Modelling of natural attenuation processes in groundwater using adaptive and parallel numerical methods

Ian Andrew Watson

Submitted for the degree of

Doctor of Philosophy

Department of Civil and Structural Engineering

University of Sheffield

January 2004

Modelling of natural attenuation processes in groundwater using adaptive and parallel numerical methods

Ian Andrew Watson

Summary

Biodegradation is an important process contributing to the natural attenuation (NA) of organic contaminants in groundwater. A numerical model was created to describe anaerobic phenol biodegradation data from an aquifer-derived laboratory scale microcosm. The dynamic behaviour of the system was simulated by considering a two-step syntrophic biodegradation model with fermentation and respiration steps, both simulated kinetically, and with hydrogen and acetate as intermediate species, and additionally, other geochemical reactions including aqueous speciation, surface complexation, mineral dissolution and precipitation. The model suggested microbial competition between respiration processes using different electron acceptors was important. In contrast, a partial equilibrium approach, considering only thermodynamics, and not kinetics, for respiration, did not explain the data.

The laboratory scale biodegradation model was transferred to a field scale reactive transport model of the phenol plume at Four Ashes, UK. The effects of acclimatisation, toxicity, and bioavailability on microbial kinetics were considered. The simulations suggest that plume core processes are much more important than previously thought, possibly with a greater impact than plume fringe processes.

The field scale model was computationally demanding due to the biogeochemical complexity. Two strategies for dealing with high computational demands are (i) parallel processing, where the workload is shared between multiple processors, and (ii) locally adaptive remeshing, where a refined area of the grid tracks moving plume fringes through the domain. A new code was developed using the partial differential equation software toolbox, UG, and tested against other biodegradation simulators. The relative efficiency of parallel, adaptive methods for multi-species biodegradation simulations was measured.

It appears, in general, that relatively complex models are required for the realistic, quantitative assessment of NA at field scale, and that parallel, adaptive numerical methods provide appropriate efficiency benefits for such simulations.

Acknowledgements

I gratefully acknowledge the help of the following people.

Firstly, I thank my supervisors Steven Banwart and Roger Crouch for their valuable guidance and encouragement throughout this thesis. I also thank Sascha Oswald who played an important role early in the project in determining the direction of the research and in setting up the collaboration with the University of Heidelberg. Despite returning to Switzerland and his new family he has continued to provide support.

The research was sponsored financially by Engineering and Physical Sciences Research Council (EPSRC) as part of a Platfrom Grant awarded to the Groundwater Protection and Restoration Group (GPRG) with the aim of building up groundwater modelling expertise in the UK. Further sponsorship was from the Environment Agency in the form of a CASE award. Funding for visits to Germany was from the Platform Grant and then from GPOLL, an European Science Foundation initiative, and finally from the British Council in the form of a British German Academic Research Collaboration (ARC) programme.

I thank Uli Mayer of University of British Columbia, Canada who kindly let me use his reactive transport code, MIN3P, and provided help and advice with modelling and writing up the results.

I thank Wolfgang Schäfer and Peter Bastian, of IWR, University of Heidelberg, Germany for making me feel welcome and providing much help with the research through their kind collaboration. The other members of IWR were also very friendly and patient with my lack of German language skills, especially Thimo Neubauer. In particular, I wish to thank Peter Bastian for providing me with his code UG, and spending large amounts of time with me to fix problems I encountered in the development work, although he is very busy.

I would also like to thank the members of GPRG who have been friendly and helpful throughout my time here.

Finally, I thank my wife, Christine, who agreed to marry me even though I was giving up my job with Corus to be a PhD student for three years, and has been very supportive throughout the whole time.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Reactive transport modelling of groundwater systems has become an important field of research during recent years (e.g. Zheng and Bennett 1995, Steefel and MacQuarrie 1996, Cirpka and Helmig 1997, Abrams et al. 1998, Hunter et al. 1998, Schäfer et al. 1998, MacQuarrie and Sudicky 2001, Saaltink et al. 2001, Brun and Engesgaard 2002, Mayer et al. 2002, Prommer et al. 2003). One reason for this is that it is a potentially powerful tool in understanding processes which contribute to the cleansing of contaminated groundwaters. At the field scale the risk arising from groundwater contamination may be reduced by natural processes (natural attenuation) (Carey et al. 2000, Lerner et al. 2000), or by various engineered remediation methods (Page 1997, Fetter 1999, Schäfer 2001, Wagner et al. 2002). Biodegradation is a major process contributing to removal of contaminant mass in plumes of groundwater pollution, and at most field sites is a very significant natural attenuation (NA) process (Smets and Pritchard 2003). Contaminant removal by biodegradation is also preferable to contaminant retardation by, for example, sorption processes since the mass may be transformed to non-toxic products, as against being stored up for possible later release into the flowing groundwater.

Biodegradation occurs due to the activity of microbial populations in the subsurface. Energy for cell growth and maintenance is supplied by the release of chemical energy during fermentation or respiration reactions (Madigan et al. 2000). In fermentation the contaminant is split into smaller molecules, including low molecular weight organic compounds, e.g. phenol may ferment into hydrogen, carbon dioxide, acetate and other volatile fatty acids. In respiration, a microbial cell transfers electrons from a primary electron donor, e.g. phenol, hydrogen, acetate, etc., to a terminal electron acceptor (TEA) (Madigan et al. 2000). TEAs in groundwater are typically limited to dissolved oxygen, nitrate, sulphate and carbon dioxide, and minerals comprising manganese(IV) and iron(III) fractions. Figure 1.1 illustrates a typical pattern of TEA zonation in a contaminant plume which might be observed. The pattern is generally thought to result from the energy available to microbial populations from each TEA. Thus, oxygen gives the most energy and is used first at the fringes of the plume, while methanogenesis yields much less energy and occurs deep in the plume core where other TEAs are exhausted.

Chappelle and Lovely (Lovley and Goodwin 1988, Chapelle et al. 1995, Lovley and Chapelle 1995) found that the intermediate species hydrogen was potentially important for diagnosing which anaerobic respiration process was dominant. Hydrogen concentration progressively increased as the dominant TEA changed from iron reduction to sulphate reduction to methanogensis. The observed sequential pattern of respiration was explained in terms of one

microbial population lowering the hydrogen concentration to below the threshold for use by the next population, such that the latter are competitively excluded by the former (Lovley and Goodwin 1988). Since hydrogen has a turn- over time of minutes, it represents the present microbial activity, and not a historical record.

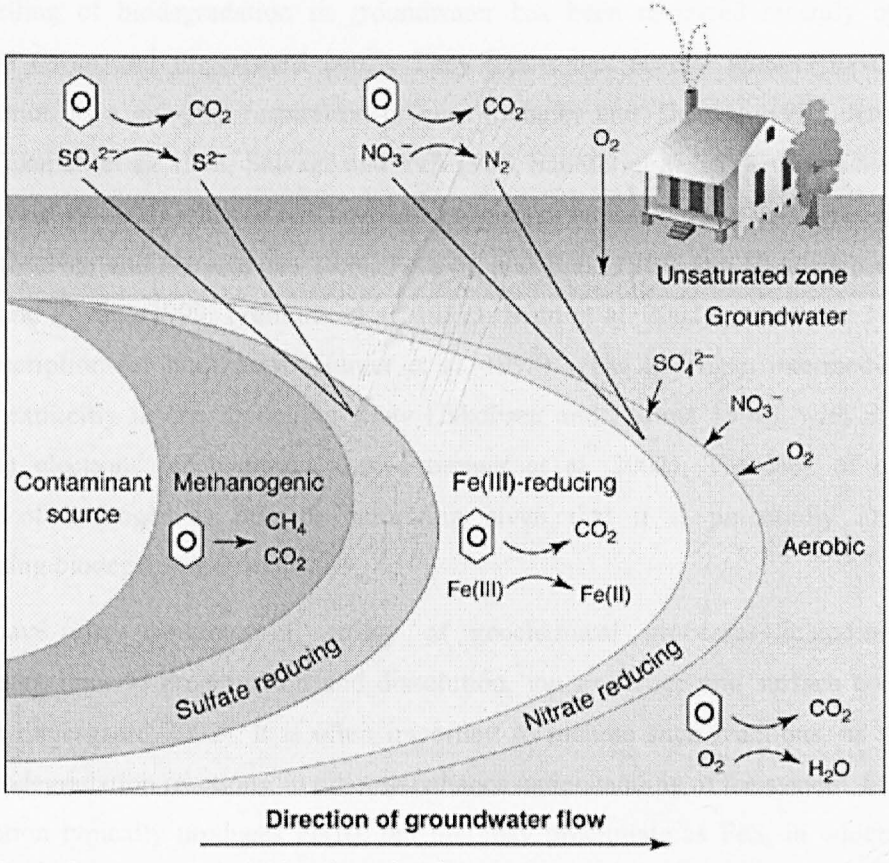

Figure 1.1 Typical conceptual model for plume development.
An organic source enters an aerobic aquifer and different redox zones develop. In this case O_2, NO_3^-, SO_4^{2-} are shown entering the plume by dispersive mixing at the fringe. Fe(III) reduction occurs where the plume is advected over Fe(III) bearing minerals, and where it is energetically favorable. Methane is produced in the interior of the plume where other TEAs are exhausted (from Lovley 2001).

Subsequently, field sites were found with overlapping TEA zones and hydrogen concentration was not diagnostic for them (Christensen et al. 2000). A new modelling approach was developed called the partial equilibrium approach (McNab and Narasimhan 1994, Hoehler et al. 1998, Jakobsen et al. 1998, Jakobsen and Postma 1999). This two-step approach includes a rate limiting kinetic step, fermentation, followed by a fast step, respiration which approaches thermodynamic equilibrium. The advantage is that overlapping TEA zones can be simulated. The approach assumes that thermodynamics is the principal control on the microbial consumption of TEAs. However, this may not always be the case since although a reaction may be thermodynamically favourable, the subsurface microbial population may not necessarily possess the enzymes to complete the reaction. Enzymatic and other microbial factors may mean that the fastest reaction is not the same as the most thermodynamically favourable one. Often in groundwater, redox disequilibrium is found, i.e. different redox couples predict different redox

states (Christensen et al. 2000). This means in some applications of the partial equilibrium model it is necessary to decouple the redox pairs such as nitrate/ammonia or carbon dioxide/methane (Brun and Engesgaard 2002), and perhaps use a kinetic relationship instead (e.g. Hunter et al. 1998).

The modelling of biodegradation in groundwater has been reviewed recently by Brun & Engesgaard (Brun and Engesgaard 2002). They report that several authors have simulated biodegradation as a one-step respiration process (Schafer and Therrien 1995, Abrams et al. 1998, Chilakapati et al. 1998, Salvage and Yeh 1998, Schäfer et al. 1998, Tebes-Stevens et al. 1998, Mayer et al. 2001) while others have used two steps typically using the partial equilibrium approach (McNab and Narasimhan 1994, Keating and Bahr 1998, van Breukelen et al. 1998, Jakobsen and Postma 1999, Prommer et al. 1999a, Brun et al. 2002), but in one case using a kinetic description for both steps (Hunter et al. 1998). The hydrogen intermediate is only simulated explicitly in one modelling study (Jakobsen and Postma 1999), with most studies substituting electrons for hydrogen (see Prommer et al. 2002). The lack of quantitative modelling of hydrogen is perhaps surprising given that it is potentially important in understanding biodegrading systems.

Models have also contained a variety of geochemical processes including aqueous complexation, mineral precipitation and dissolution, ion exchange and surface complexation (Brun and Engesgaard 2002). It is often important to include such reactions, as well as the primary biodegradation reactions, in order to enhance understanding of the system. For instance, iron reduction typically produces Fe(II), but this may precipitate as FeS, in which case it is possible to underestimate the amount of iron reduction if the only data available is for dissolved Fe(II) (Christensen et al. 2000, Schäfer 2001). At present, it is not clear if sufficient geochemical processes are generally included in field studies to give a realistic appreciation of the natural attenuation processes.

In order to address the issue of improving understanding of biodegradation processes, a numerical model may need to include a complex multi-species chemical model in a highly refined field scale transport model. Such a numerical model is computationally very demanding and so optimum numerical methods should be applied. One area discussed in the literature is the efficiency of fully coupled reactive transport models compared with split operator approaches which solve the reactions and transport sequentially (Yeh and Tripathi 1989, Cirpka and Helmig 1997, Saaltink et al. 2001). While the split operator approach can save time and memory, there are truncation errors introduced by the splitting. Recently, the fully coupled approach has been found to be more robust and possibly even more efficient in large, chemically difficult cases.

Other ways to save time on large numerical problems include adaptively refining the grid only where it is necessary, and using multiple processor computers (Barry et al. 2002). While both these methods have been applied in the field of fluid dynamics as a whole, there has been little

reported for reactive transport in porous media. Local adaptive remeshing has the potential to aid interpretation of plume fringe processes through progressively refining the mesh at the fringe as it migrates. This might be desirable if a plume has a narrow fringe with sharp gradients. Extra refinement at the fringe may give increased insight into the processes of dispersive mixing of an electron donor with a TEA and their microbially mediated reaction together. Adaptive remeshing for reactive groundwater problems has been limited to an ion exchange example (Wolfsberg and Freyberg 1994), and a study of biodegradation (Wagner et al. 2002). Parallel processing on digital computers has the potential to speed up solution times proportional to the number of CPUs used, but with a limitation imposed by the time spent communicating data between different processors. Chemically complex models may be well suited to parallelisation, especially if the split-operator approach is used (Barry et al. 2002). Since parallel machines (e.g. clusters of PCs) have only recently become more affordable, and special coding is needed to allow parallelisation, little has so far been reported in the groundwater reactive transport literature (Schäfer et al. 1998). The combination of parallel and adaptive methods in a 2D/3D reactive transport groundwater application is, as far as the author is aware, as yet unreported in the literature.

1.2 Objectives

In order to contribute to research efforts in the areas mentioned above the objectives of this thesis are as follows.

1. To develop a conceptual and numerical model of biodegradation processes at laboratory scale. To include as much detail as necessary to describe the reacting system, including explicit inclusion of intermediate species hydrogen. To use the model to make a comparison between kinetic and partial equilibrium approaches to understanding biodegradation.

2. To apply the laboratory conceptual model to a realistic field scale example, and assess the applicability of including extra complexity in terms of biogeochemical processes. To assess whether there are particular processes which are important, but are not routinely addressed by current modelling practice.

3. To develop a new numerical modelling tool which is better suited to simulating multi-dimensional reactive transport in groundwater with significant biogeochemical complexity, than current state of the art models. To examine the benefits and limitations of adaptive remeshing around plume fringes, and parallel processing, for fully coupled reactive transport problems.

1.3 Scope of work

The first phase of work was to fulfil the first objective. A detailed dataset of a microcosm in which phenol degraded via a number of TEAs over 600 days was provided by Youxian Wu (Wu 2002) at the Groundwater Protection and Restoration Group (GPRG), University of Sheffield. Figure 1.2 illustrates the microcosm set-up schematically. The microscosm was designed to simulate conditions at the Four Ashes, UK, phenolics plume site (Williams et al. 2001). Two numerical codes were considered for the modelling exercise: TBC (Schäfer et al. 1998) and MIN3P (Mayer et al. 2002). Both are reactive transport codes. Although the microcosm only required modelling of reactions, and not transport, it was felt appropriate for the author to gain experience with state of the art reactive transport codes in view of the other objectives. TBC was designed for modelling biodegradation with biomass growth and decay. MIN3P had previously been applied to the Four Ashes field site as a one step, respiration model without biomass growth (Mayer et al. 2001). It was found MIN3P could be adapted to model biomass growth in a two step kinetic model and that its unsaturated flow module could be used to simulate sampling loss from the microcosm, which gave MIN3P the advantage over TBC. MIN3P's developer, Uli Mayer, University of British Columbia, Canada, provided collaborative advice on the use of the code. Thus MIN3P was used to develop the numerical and conceptual model describing the processes occurring in the microcosm.

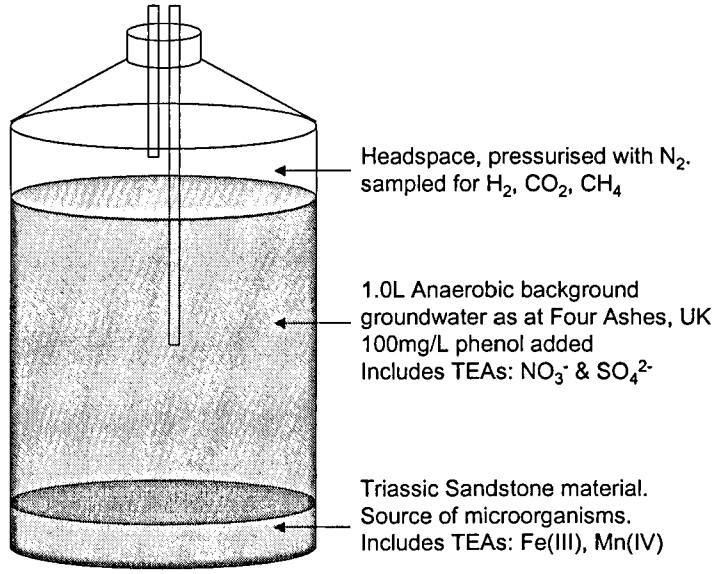

Figure 1.2 Schematic picture of the microcosm simulated in this work.

As far as the author is aware, such a detailed model for a groundwater degradation microcosm has not been completed before. The work showed that intermediate species, hydrogen and acetate, could be described by a fully kinetic two-step reaction system, and that dissolved Fe^{2+} could be understood in terms of FeS precipitation and surface complexation reactions.

The next phase of work was to develop and test a new reactive transport model including tools for adaptive remeshing and parallel processing. State of the art reactive transport models,

including MIN3P and TBC, do not at present include these tools, partly because the coding effort can be considerable. A collaboration was set up between GPRG and Peter Bastian of IWR, University of Heidelberg, Germany, also including his colleagues Dr. Wolfgang Schäfer and Thimo Neubauer. Peter Bastian is a co-developer of a software platform called UG (Unstructured Grids) (Bastian et al. 1997a). The software platform is designed as a toolbox for solving partial differential equations in general. It provides tools for grid management, efficient data storage, solvers for linear and non-linear systems, including fast multigrid solvers, and different discretisation schemes, all available in parallel. These items form the UG library and the user then implements specific problems into this framework which allows the tools to be used. For example, a multiphase flow problem in UG demonstrated the effectiveness of using adaptive remeshing in parallel (Bastian et al. 2000). One example of the use of UG to simulate biodegradation kinetics in groundwater using adaptive remeshing on a single processor exists (Wagner et al. 2002). However, the basis for the development work done here was a radioactive decay and transport benchmark model (Bastian and Lang 2002). This provided a general source code, in C, for multispecies reactive transport in groundwater which could be modified by the present author with a moderate coding effort to include biodegradation kinetics and thermodynamic equilibrium reactions. Due to the lack of documentation and the large size of the UG code this task required support from IWR staff. Several visits to Heidelberg were made initially to set up the collaboration, and later to discuss and solve problems encountered. In this way, the biodegradation application in UG was completed, meeting part of objective 3. Verification of the code was performed by comparison with three literature examples of biodegradation and transport in 1D and 2D (Schäfer et al. 1998, Mayer et al. 2001, Huang et al. 2003).

The next phase of work was prescribed by objective 2, above. The UG code was used to implement the microcosm conceptual model into a field scale model of the plume of phenolics at Four Ashes, West Midlands, UK (Williams et al. 2001). Since the rates in the microcosm are much faster than rates in the plume, the microcosm rate parameters could not be used to predictively model the field site. Instead, the model was calibrated against existing published datasets for multilevel sampling wells (Thornton et al. 2001b). In addition to the reactive model from the microcosm, the effect of homogeneous versus heterogeneous anisotropic flow fields, and of variable source zone concentrations, was investigated. The plume had been modelled previously by Uli Mayer with MIN3P (Mayer et al. 2001), so the new UG model sought to increase the level of realism by including more chemistry in multi-dimensional models. Another aim was to identify any processes that are important and yet are in general not included in studies of biodegrading plumes.

In order to complete the field scale modelling, much use was made of the adaptive and parallel tools included in UG. Most of the work was completed on a small, 8 processor Beowulf cluster

of Linux PCs in Sheffield to which the author was fortunate to have exclusive access. Access was also granted to HELICS, a 512 processor Linux cluster at Heidelberg, and some work was completed on part of that cluster through a queuing system. These computational facilities allowed a more realistic set of physical and chemical processes to be included in the model. The work revealed that several of the important factors for the microcosm were also important at the field scale. In particular, the kinetic two-step model was useful in understanding factors affecting fermentation and respiration processes, particularly the role of toxic inhibition by phenolics. The inclusion of surface complexation increased the amount of iron reduction necessary, which had a significant impact on the mass turnover of the phenolics. The spatial and temporal change of biodegradation rates due to long term acclimatisation was important in understanding concentration profiles at multilevel sampling boreholes.

The final part of the work was to assess the usefulness of UG's numerical methods in achieving a more realistic field model in practical amounts of time, as in the third objective, above. Adaptive remeshing was tested on a single processor using mulitspecies, multidimensional models of the Four Ashes site. A uniform grid version of the model was used to test the efficiency of parallel processing. Then the combined use of parallel and adaptive methods was tested. In addition, a sparse matrix storage scheme available in the UG toolbox was tested. This method stores only non-zero entries in the Jacobian matrix which in multicomponent models saves significant memory and time (Neuss 2002). All these methods were found to be useful in improving efficiency without compromising accuracy of results, especially for larger multispecies models. Parallel processing gives speed benefits which scale well with number of processors used. The adaptive remeshing showed some limitations: when simulating only a few species there was a significant overhead in adapting the grid in parallel, but not on a single processor. When the core of a plume has reacting species and chemical gradients, use of adaptive coarsening within the plume core requires significantly more iterations to solve the fully coupled system and can limit the benefits of using less mesh nodes. The sparse matrix storage typically saved 90% memory and 20% runtime.

1.4 Structure of Thesis

The structure of this thesis is this introduction, three separate pieces of research work, conclusions, and appendices. The introduction, Chapter 1, briefly describes the research background, the objectives, and the scope of the work completed which meets the objectives and provides original contributions to the research effort. Chapters 2 to 4 describe work meeting objectives 1 to 3 respectively. Each of these chapters forms a self-contained and complete piece of research, including background, methodology, results and discussion, which has been written as a manuscript for publication in a peer reviewed journal. They are as follows:

Chapter 2 is published as:

> Watson, I. A., S. E. Oswald, K. U. Mayer, Y. X. Wu, and S. A. Banwart. 2003. *Modeling kinetic processes controlling hydrogen and acetate concentrations in an aquifer-derived microcosm.* Environmental Science & Technology **37**:3910-3919.

This paper has been reformatted for this thesis, but is otherwise the same as the published version. A copy in the journal style is available on the CDROM.

Chapter 3 is a manuscript submitted as:

> Watson, I. A., S. E. Oswald, R. S. Crouch, S. F. Thornton, and S. A. Banwart. 2004. *Insights gained through multispecies field scale reactive transport modelling driven by biodegradation complexity.* Water Resources Research:submitted.

Chapter 4 is a manuscript submitted as:

> Watson, I. A., S. E. Oswald, R. S. Crouch, and P. Bastian. 2004. *Advantages of using locally adaptive remeshing and parallel processing in modelling biodegradation in groundwater.* Advances in Water Resources:submitted.

In all three cases the present author conducted the research work and authored the paper manuscripts, with co-authors providing advisory or supervisory roles. The reader of this thesis will find a limited amount of repetition of introductory material in the chapters 2 – 4, which was necessary for the papers to be read independently. The literature review has been effectively split between the three manuscripts, and is therefore only briefly covered in this introductory chapter.

Chapter 5 contains the overall conclusions of the research and the recommendations for future research.

The appendices are as follows.

Appendix A. Conference paper which gives related but different data to chapter 2

Appendix B. Conference paper which gives related but different data to chapter 4

Appendix C. Tables of parameters used in UG code test cases.

Appendix D. User guide to biodegradation in UG: script and parameter code for 2 examples

Appendix E. CDROM contents.

2 Modelling kinetic processes controlling hydrogen and acetate concentrations in an aquifer-derived microcosm.

IAN A. WATSON[1], SASCHA E. OSWALD[1], K. ULRICH MAYER[2], YOUXIAN WU[3] AND STEVEN A. BANWART[1]

1. Groundwater Protection and Restoration Group, Department of Civil and Structural Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, United Kingdom. 2. Department of Earth and Ocean Sciences, University of British Columbia, 6339 Stores Road, Vancouver, BC, V6T 1Z4, Canada. 3. Department of Civil and Environmental Engineering, University of Missouri, Columbia, Missouri, USA

Abstract

Groundwater contaminants may degrade via fermentation to intermediate species, which are subsequently consumed by terminal electron accepting processes (TEAPs). A numerical model of an aquifer-derived laboratory microcosm is developed to simulate the dynamic behaviour of fermentation and respiration in groundwater by coupling microbial growth and substrate utilisation kinetics with a formulation that also includes aqueous speciation and other geochemical reactions including surface complexation, mineral dissolution and precipitation. The model is used to test approaches that currently make use of $H_{2(aq)}$ to diagnose prevalent TEAPs in groundwater. Competition between TEAPs is integral to the conceptual model of the simulation, and the results indicate that competitive exclusion is significant, but with some overlap found in the temporal sequence of TEAPs. Steady state $H_{2(aq)}$ concentrations observed during different TEAPs do not differ significantly. The results are not consistent with previous applications of the partial equilibrium approach since most TEAP redox pairs exhibit free energies that suggest a particular process is able to proceed, yet observations here show that this process does not proceed.

2.1 Introduction

Intermediate species of biodegradation pathways for organic pollutants provide important diagnostic tools for assessment of groundwater contamination plumes. Their quantitative analysis gives confidence in the management of remediation technologies, such as natural attenuation, by providing additional lines of evidence for site conceptual models of reactive processes (Chapelle et al. 1995, Albrechtsen et al. 1999, Jakobsen and Postma 1999, Christensen et al. 2000, Thornton et al. 2001b) and data for reactive transport modelling (Jakobsen and Postma 1999). The presence of intermediate species, particularly $H_{2(aq)}$ and low molecular weight organic acids, is explained with the concept of syntrophic biodegradation, reviewed recently by Christensen et al. (Christensen et al. 2000) and Chapelle (Chapelle 2000).

Organic compounds are degraded in three sequential steps: hydrolysis, fermentation and respiration (Christensen et al. 2000). In this study we focus on the latter two steps, since hydrolysis does not occur for phenol, a low molecular weight organic compound considered here, that acts as a primary fermentation substrate. Fermentation produces significant amounts of intermediate species including hydrogen, acetate and volatile fatty acids (VFAs). Subsequently, during respiration, these species are consumed to give mineralised products. Typically the $H_{2(aq)}$ or acetate would act as electron donors while one or more of nitrate, sulphate, Fe(III), Mn(III,IV), or inorganic carbon species act as terminal electron acceptors (TEAs). The interaction and relative rates of the fermentation and terminal electron accepting processes (TEAPs) controls the concentrations of $H_{2(aq)}$, acetate and VFAs.

Former work showed that groundwater systems that are geochemically at a steady state, had specific, concentrations of $H_{2(aq)}$ that were characteristic for the dominating TEAP (Lovley and Goodwin 1988). This was proposed to be due to efficient competitive exclusion between microbial populations that utilise specific TEAs (Christensen et al. 2000). An important implication of this competitive exclusion (CE) approach to the interpretation of $H_{2(aq)}$ concentrations was that a single TEAP should dominate locally at a particular location within a contaminant plume. Thus, using CE the $H_{2(aq)}$ concentration could be used to identify the dominant redox process in any part of a plume. In related work acetate and VFAs did not show useful characteristic concentrations for TEAPs (Vroblesky et al. 1997), so the CE approach did not provide a useful diagnostic capability using these system data.

Subsequently, several field sites were identified with concomitant TEAPs, i.e. two or more TEAPs locally active at a given $H_{2(aq)}$ concentration. This apparent discrepancy with the assumption of competitive exclusion gave rise to the concept of partial equilibrium (PE). In the PE approach, $H_{2(aq)}$ is assumed to be the main energy source for the TEAPs and therefore its concentration is used together with other geochemical parameters to calculate the available free energy (ΔG_r); i.e., the extent of disequilibrium for the TEAP, written as a stoichiometric electron transfer reaction. The available free energy for each TEAP is then compared with the respective threshold energy requirements (ΔG_T, a ΔG value <0) for the reaction to proceed. All TEAPs characterised by an available free energy of reaction that is lower than the threshold value ($\Delta G_r < \Delta G_T$) are presumed to be active; thus providing a diagnostic test to identify which TEAPs are active within the plume.

A recent review by Banwart and Thornton (Banwart and Thornton 2003) outlines clear differences in the conceptual models underlying the CE and PE approaches. The CE approach is based on microbial ecological considerations. Here, physiologically distinct populations utilise TEAs with characteristic rates of $H_{2(aq)}$ consumption that are significantly different for each population. Dominant TEAPs thus arise by ecological selection of populations that are capable of the most rapid $H_{2(aq)}$ utilisation, and thus the greatest growth in population numbers

and overall activity. As the associated TEA is consumed, it eventually becomes limiting to $H_{2(aq)}$ utilisation kinetics. The rate then decreases with TEA concentration until the population no longer competes effectively and another TEAP establishes dominance. The PE approach arises from consideration of geochemical thermodynamics where some solutes and phases in a system have reached chemical equilibrium while others remain at disequilibrium (Nordstrom and Munoz 1994, p12). For syntrophic biodegradation, fermentation is presumed to proceed slowly as an irreversible process, while the subsequent TEAPs are sufficiently fast that the threshold value for the free energy is approached as a limiting value that approximates chemical equilibrium. This approach to equilibrium exhibits time scales that are far shorter than characteristic time scales for plume dynamics. Thus, even for a non-equilibrium system such as a biodegrading, transient-state plume of organic contaminants, the $H^+/H_{2(aq)}$ and TEA redox couples are assumed to approach local equilibrium. In the PE approach, $H_{2(aq)}$ concentrations are dictated by the equilibrium redox buffer system described by the reacting minerals and solutes. In this case, considerations of microbial physiology and ecology and the resulting relative rates of microbial processes are not a controlling factor; the only caveat is that they are sufficiently fast that they approach equilibrium.

The Four Ashes field site, UK, is an example with simultaneous TEAPs and far higher than expected $H_{2(aq)}$ concentrations which were not consistent with any previously reported values for particular TEAPs (Thornton et al. 2001b). The field site was therefore not amenable to quantitative interpretation by the CE approach. The PE approach showed that sulphate reduction and methanogenesis were active in areas where the free energy of reaction was below previously reported values for the corresponding threshold energies. However, if these reaction energies were assumed to be indicative of threshold values at the site, then these values were significantly lower than other published values, suggesting that quantitative application of PE for diagnostic purposes is not straight forward. The high $H_{2(aq)}$ concentrations at this site were explained by possible toxic inhibition of the TEAPs causing them to be much slower than usual (Thornton et al. 2001b, Wu 2002), a site specific characteristic that is not considered in the CE and PE approaches. Thus this field case is potentially problematic for the generality of the underlying conceptual assumptions used in the two established approaches.

In this work, a numerical model is used to investigate the dynamic behaviour of syntrophic biodegradation in an anaerobic laboratory system derived from the Four Ashes plume. The model applies microbial growth and substrate utilisation kinetics to fermentation and subsequent respiration processes. The state variables are concentrations of reactants and products. Simulation of system dynamics result from the integration over time of empirical rate laws based on Monod kinetics for the stoichiometric redox reactions that describe TEAPs. The aim is to develop a more fundamental description of process dynamics where temporal changes are simulated by rate laws with fixed parameter values that can subsequently be compared with

values from similar systems. This is coupled with a relatively complete description of aqueous speciation and other geochemical reactions including adsorption, mineral dissolution and precipitation.

The representation of the existing conceptual models, which generally focus on interpretation of steady-state (CE) or partial equilibrium (PE) conditions, is extended to consider the behaviour of biodegradation intermediates such as $H_{2(aq)}$ and acetate in a dynamic reaction system where we consider irreversible biodegradation processes. Our fully kinetic model (FKM) is formulated based on observations and results in an empirical quantification of the extent to which TEAPs are occurring either concomitantly or sequentially, i.e. how effective competitive exclusion processes are. In this way the CE approach is tested with regard to the validity of its underlying conceptual assumptions, and to its usefulness in treating reaction intermediates, such as $H_{2(aq)}$, as quantitative diagnostic aids to assess TEAPs during biodegradation in groundwater. Our FKM does not simulate TEAPs as controlled by thermodynamic equilibrium. However, the PE approach is subsequently applied to the measured data and its benefits and limitations are discussed.

2.2 *Method and Modelling Procedures*

Dataset description

The chosen dataset was collected from a 600 day duration microcosm experiment which comprised anaerobic phenol degradation in the presence of several TEAPs with hydrogen and acetate as intermediates (Wu 2002). The microcosm was initially created to further investigate microbial processes in groundwater observed in a phenolics plume at the Four Ashes site (Thornton et al. 2001b). A 1.25L microcosm was set up with sterile synthetic groundwater, as per composition of background groundwater outside the plume of phenolics. 200g anaerobic sandstone material, from within the plume area, provided the source of microorganisms, as well as iron and manganese bearing minerals. Resazurine was added at a final concentration of 2 mg/l as a redox indicator, which exhibits various colours at different redox potentials (Eh): blue-brown at Eh >150 mV, pink at -50 - +150, and colourless at < -50. The microcosm was then flushed with pure N_2 gas through sterile 0.2 μm filter for 30 minutes after 3 vacuum-N_2 filling cycles until the liquid colour inside the microcosm became pink. A killed cell control (autoclaved 3 times at 121°C and 1.05 kg/cm^2 for 20 min each) was prepared as above.

1mM phenol was added at the start of the experiment, and a further 0.3mM was added after 470 days when the initial phenol had all been consumed. Aqueous samples were taken at regular intervals throughout the 600 days and the headspace gas was sampled regularly between days 120 and 600. Liquid samples were passed through 0.2 μm membrane filter and either analysed

immediately or frozen at -20°C and stored for subsequent analysis. Analyses of phenolic compounds were done by reverse-phase gradient-elution HPLC (High Performance Liquid Chromatography). The apparatus consisted of two Isco Model 2350 pumps equipped with an Isco Chemresearch System controller, two Pye-Unicam 4020 Model UV variable wavelength detectors set at a wavelength of 280 nm and 266 nm, respectively, and a Jones-Genesis C18 chromatography column (250 mm x 4 um) and C18 guard column. The sample injection loop was 5 µl. The eluent was an acetonitrile: 1% acetic acid mixture with 40: 60 volume ratio, fed at a flow rate of 1.0 ml min^{-1}.

Ions including sulphate and nitrate were analysed by Dionex 2000 IC (ion chromatography). Anions were separated using an AS14 column (Dionex Corporation) with AG14 guard column (Dionex Co.). Cations were analysed on a CS12 column with a CG12 guard column (Dionex Co.). The instrument was incorporated with both anion and cation micro-membrane suppressers and a conductivity detector. The eluent for anion analysis was 3.5 mM Na_2CO_3 + 1.0 mM $NaHCO_3$ at a flow rate of 1.2 ml/min, and 25 mM H_2SO_4 was used as the regenerant. The cation eluent was 18 mM methanesulphonic acid at a flow rate of 1 ml/min. Some dissolved elements such as Fe(II) and Mn(II) were analysed by ICP-OES (Inductively Coupled Plasma-Optical Emission Spectrometry) with Spectra Analytical Instruments Spectroflame M120E. The samples were acidified with concentrated HNO_3 to pH <1 immediately after membrane filtration.

Methane and carbon dioxide contents in headspace were determined using a Varian model 3400 GC (Varian Associates, Norwalk, CT, USA) equipped with a flame ionization detector (FID). Methane and carbon dioxide were separated at 140°C on 80/100 Carbonsphere in a stainless-steel column (1.83m x 3.2 mm id) using nitrogen (40 ml/min) as carrier. The temperatures of the injector and detector were 340°C and 250°C, respectively. Gas-phase hydrogen concentration was measured using a Trace Analytical RG3 Reduction gas analyzer (Trace Analytical, Menlo Park, CA, USA). Hydrogen was analyzed at 165°C on 60/80 Spherocarb in a stainless-steel column (0.92 m x 3.2 mm id) using nitrogen (30 ml/min) as carrier. The reduction gas detector was operated at a temperature of 250°C. The pressure transducer described by Salanitro and Diaz (Salanitro and Diaz 1995) was used to check the total pressure in microcosms. The concentrations of dissolved gases were calculated according to Henry's Law partitioning (Stumm and Morgan 1996). Mass of $H_{2(g)}$ removed from the headspace during sampling was determined by the calculated $PH_{2(g)}$ and the sample volume. Headspace overpressure was maintained by replacing the sampled gas volume with an equal volume of $N_{2(g)}$ at atmospheric pressure.

The microcosms remained colourless (active) and pink (control) throughout the experiment. In the active microcosm, hydrogen and acetate production coincided with loss of phenol, and are assumed to be produced by fermentation of the phenol, and to be subsequently used by the

TEAPs. The observed sequence of the TEAPs is denitrification, days 0 to 60, followed by iron and manganese reduction, days 50 to 350. Sulphate reduction occurs in a relatively brief period, days 100 to 200, concomitantly with iron and manganese reduction. Significant methanogenesis is observed only after cessation of all other TEAPs, days 400 to 600. Parallel experiments (Wu 2002) demonstrated that elevated $H_2(g)$ concentrations in microcosm headspace were constant over periods of several months with no loss by diffusion from the reactor.

2.2.1 Numerical model

The MIN3P numerical simulation code (Mayer et al. 2002) was used for this work. It is a reactive transport model with capabilities including equilibrium and kinetic reactions involving aqueous, gaseous and solid species, together with saturated and unsaturated flow and transport in three dimensions. The model has previously been applied to the Four Ashes groundwater plume (Mayer et al. 2001), but this work neither considered growth of microbial populations nor coupled syntrophic species.

2.2.2 Conceptual reaction pathway

The field scale modelling previously completed with MIN3P (Mayer et al. 2001) described a reaction pathway whereby phenol is mineralised directly to carbon dioxide by each of the TEAPs, denitrification, manganese and iron oxide reduction, sulphate reduction, and methanogenesis. In the current work, however, the syntrophic biodegradation concept was applied as illustrated in Figure 2.1 and explained above. Both fermentation and TEAP steps are formulated with empirical kinetic expressions to allow the simulation of the dynamic features of system behaviour. There is no a-priori assumption of which step is rate limiting; fermentation can be either faster or slower than TEAPs. There is no assumption about which TEAPs occur exclusively or concomitantly – the kinetic model is flexible enough to accommodate complete, partial, or negligible competitive exclusion between TEAPs. The fermentation stoichiometry (see Table 2.2 below) is simplified to only produce $H_{2(aq)}$, acetate, and biomass, and ignores other possible metabolic products such as CO_2 or other VFAs.

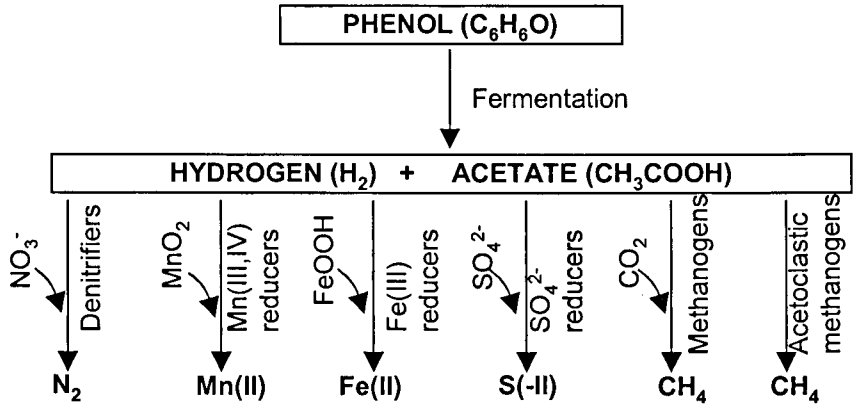

Figure 2.1 Conceptual model of the microbial reaction pathway.

2.2.3 Carbon balance

The carbon initially present in the microcosm as phenol was assumed to be transformed to aqueous and gaseous products, part of which were lost due to sampling, and also biomass. Biomass was not measured directly, but because other species in the system carbon balance are measured, this allows the biomass produced to be deduced but not independently verified. Results were otherwise verified by testing against the observed alkalinity balance (and pH) and the system electron balance. In this manner, biomass was quantified and found to be an important part of the system carbon balance. The sampling loss of aqueous phase and consequent headspace growth within the microcosm were simulated using the unsaturated flow facility of MIN3P and setting an outflow of water equal to the volumes sampled. The carbon balance between the aqueous and gaseous phases was simulated using equilibrium partitioning according to Henry's Law constants. Aqueous complexation was included for inorganic carbon species, which is done in MIN3P via a standard thermodynamic database based on the WATEQ4F database (Mayer et al. 2002). The experimental temperature was 20°C and MIN3P corrects for this using the Van't Hoff equation (Mayer et al. 2002). The organic carbon balance, including degradation and biomass growth, was simulated using biodegradation kinetics.

2.2.4 Biodegradation kinetics

Microbially mediated fermentation and TEAPs were all conceptually modelled as irreversible redox reactions and were mathematically described by empirical kinetic rate laws based on the Monod expression. Many studies on microbial physiology and kinetics are based on observations from pure cultures. By the complex nature of the microbial ecology represented by the indigenous microbes in an aquifer it is inevitable that kinetic descriptions of microbial processes have a strong empirical component. Thus, the FKM simulation uses a standard empirical multiple Monod kinetic formulation, which is provided in MIN3P.

$$r_S = \frac{dS}{dt} = -k_{\max} \cdot X \cdot \frac{S}{(K_{M\_S} + S)} \cdot \frac{TEA}{(K_{M\_TEA} + TEA)} \cdot \frac{K_I}{(K_I + I)} \tag{2.1}$$

r_S is the rate of consumption of substrate S, [mol/L/s]; S is the substrate concentration [mol/L]; k_{max} is the maximum substrate utilisation rate [1/s]; X is the biomass concentration [mol/L]; K_{M_S} is the Monod half saturation constant with respect to substrate S [mol/L]; TEA is the concentration of the particular TEA [mol/l]; K_{M_TEA} is the Monod half saturation constant with respect to the TEA [mol/l]; K_I is the inhibition constant [mol/l]; I is the concentration of an inhibiting substance [mol/l]. MIN3P allows multiple Monod and inhibition terms to be included in equation (2.1).

2.2.5 Biomass

The biomass term, X, was considered to be stoichiometrically equivalent to hydrated carbon (CH_2O). Several separate biomass "populations" for fermentation and individual TEAPs were included explicitly as a product of the redox reactions. This was found to be necessary to reproduce the observed results. Kinetic behaviour is averaged over physiological classes of organisms that are able to use a particular TEAP. When modelling this complex mixed population, we do not attempt to model individual strains of organisms, but groups of organisms and their average enzymatic activity with respect to particular biodegradation reactions. The model assumes a completely mixed reactor i.e. that compartmentalisation of biomass with slow mass transfer between biofilms does not occur and that chemical or biomass activity gradients are negligible. Compartmentalisation requires greater conceptual and mathematical complexity that was neither necessary, nor could be justified to describe the system in this study.

MIN3P simulates growth by including biomass as a product in the reaction stoichiometries, according to the molar biomass yield, Y = biomass produced [mol CH_2O] / substrate consumed [mol]. This requires different stoichiometries for different values of Y. Each stoichiometric reaction was balanced according to C,H,O, electrons, and TEA species, so each reaction is mass and charge conservative. The values for Y in each reaction were constrained in two ways. Firstly, the values were adjusted in order to close the system carbon and electron balance for the TEAPs and the system as a whole. Secondly, the proposed stoichiometry was used to calculate the approximate free energy of reaction, ΔG_r, by using typical activities in the microcosm reactor and the approximate energy requirements needed to synthesise biomass (Chapelle 2000). All values for this showed that all proposed reactions were thermodynamically favourable, i.e. $\Delta G_r < 0$. This thermodynamic constraint on the calibrated Y values is consistent with our conceptual approach to describe biodegradation as an irreversible process that does not approach thermodynamic equilibrium, and thus differs from the conceptualised equilibrium for TEAPs in the PE approach.

The model was simplified for hydrogen consuming TEAPs (H_2 TEAPs) so that no biomass growth resulted from the stoichiometry, and the X term was omitted from the rate expression. This was possible because compared with phenol or acetate the H_2 substrate has less energy to

offer to any growing organisms so that any growth would consequently be relatively slow and make little difference to the system carbon balance or reaction rates.

It was assumed that biomass was growing and decaying in the microcosm. However, it was found to be unnecessary to explicitly parameterise the model for biomass decay, so for simplicity biomass dynamics are simulated with stoichiometric growth equations, which represent only net increase in biomass. In the absence of an explicit mathematical description of decay processes in this closed system, the kinetic descriptions will lead to accelerating growth followed by slowing and cessation of the growth rates as the concentration of substrate, or TEA, decreases to limiting concentrations and approaches zero.

2.2.6 TEAP competition mechanisms

The observed sequence of TEAPs, stated above, is assumed to occur as TEAPs compete, more or less successfully, for available substrates, i.e. H_2 and acetate (Chapelle 2000). The FKM incorporates two processes which allow TEAPs to compete for substrate, and these have been implemented mathematically as the biomass growth and inhibition terms as in equation (2.1).

The biomass growth process may give rise to competition since the reaction rate is proportional to the population biomass, X in equation (2.1). Thus, if one population is relatively much larger than a competing population, its rate will be proportionally faster and the smaller population will be out-competed. A further modelling refinement is given by the selection of the initial amounts of biomass, which effectively determines the initial reaction rates, and together with the growth rate constants determines the time necessary for the population to grow sufficiently to become competitive. The sensitivity of this parameter was tested and is presented below. As described earlier, a TEAP with more favourable thermodynamics could potentially have a higher biomass yield. This allows a more rapidly increasing degradation rate since it is proportional to biomass, and so thermodynamics has an indirect impact on the Monod kinetic description of how efficiently TEAPs compete for substrate. However, the FKM remains distinct from the PE approach where thermodynamic equilibrium of TEAPs is enforced.

The FKM uses empirical inhibition terms, which allow TEAPs to be retarded, i.e. slowed, in the presence of other potentially competing TEAPs, with these TEAPs being indicated by their TEA. For example, the denitrifying TEAP is observed to have a competitive advantage over the iron reduction TEAP. This advantage may be due to more efficient substrate utilisation, perhaps due to more favourable thermodynamics or other microbial physiological reasons (Chapelle 2000), but the advantage will only exist so long as sufficient nitrate remains. In the mathematical formulation (2.1) the iron reduction TEAP rate is inhibited by nitrate. Thus, while nitrate is present, the iron reduction rate is suppressed, and this gives the nitrate reduction TEAP the observed competitive advantage. This does not imply any toxicity effects of the inhibiting

TEA. Instead, the inhibition term is an empirical way of including a better efficiency of one process with respect to another in the model.

By considering these competition processes it becomes possible to represent and investigate the dynamic, as opposed to steady-state, behaviour of intermediate and other species. Dynamic behaviour occurs, for example during a transition from one dominating TEAP to another. The resulting TEAP rates should also allow a quantification of the extent to which competitive exclusion is effective. Notably, both CE and PE approaches assume steady-state conditions, so it is of interest to assess the timescale within which steady-state conditions might be restored. Therefore, pseudo-first-order rate constants were calculated for the simulated substrates in order to quantify characteristic time scales to approach steady-state conditions.

2.2.7 Electron balance

The system electron balance was achieved by considering the initial electron donor, phenol, and the TEAs nitrate, sulphate, $FeOOH_{(s)}$ and $MnO_{2(s)}$, as well as the products of fermentation processes (cf.Thornton et al. 2001a). Each stoichiometric reaction was balanced to conserve electrons and so the system electron balance is inherent in the simulation. The ratio of TEAs in the balance depends on the contributions of each reaction throughout the experiment. Since oxygen was not present in the microcosm, and there was no significant gypsum present in the aquifer material (Thornton et al. 2001b) these do not contribute to the electron balance. Aqueous Fe(II) and Mn(II) concentrations were constrained by considering precipitation of mineral phases, $FeS_{(s)}$, $FeCO_{3(s)}$, $MnCO_{3(s)}$, and sorption of the metal cations onto surface sites of the aquifer sediment in the microcosm through surface complexation reactions. A standard kinetic formulation for mineral precipitation (Mayer et al. 2002) and equilibrium formulation for the surface complexation reactions was used (see Table 2.1). As with many modelling approaches for complex natural systems we rely strongly on parameter values taken from idealised model laboratory systems (Table 2.1) and test whether the results are consistent with the experimental observations. Such an approach is routinely applied to complex geochemical systems including adsorption of contaminants on soils (Banwart 1997) and on aquifer rock (Banwart 1999). The surface area parameter used for surface complexation was constant and was calibrated, with sensitivity testing of simulation results to its value reported below. Reaction of reduced compounds with O_2 is not considered since O_2 was excluded from the microcosm for the duration of the experiment.

Table 2.1 Surface complexation reaction thermodynamic data.

Surface Complexation Reaction	Log K
$>FeOH_{(w)} + H^+ = >FeOH_2^+{}_{(w)}$	7.29
$>FeOH_{(w)} = >FeO^-{}_{(w)} + H^+$	-8.93
$>FeOH_{(w)} + Fe^{2+} = >FeOFe^+{}_{(w)} + H^+$	-2.98
$>FeOH_{(w)} + Fe^{2+} + H_2O = >FeOFeOH_{(w)} + 2H^+$	-11.55
$>FeOH_{(w)} + Mn^{2+} = >FeOMn^+{}_{(w)} + H^+$	-3.50

Thermodynamic data used in MIN3P is taken from WATEQ4F database as used in PHREEQC2 (Parkhurst and Appelo 1999). K is the equilibrium coefficient.

2.2.8 Calibration

The model was calibrated using a step-wise trial-and-error approach starting by adjusting the parameters of the earliest reactions observed during the experiment, then proceeding to the parameters of the following reactions. The apparent sensitivity of the simulated behaviour to parameter values was also recorded during the calibration procedure, described later.

2.3 Results and Discussion

Figure 2.2 illustrates that the model results demonstrate well the essential trends in the observed changes in species concentrations during the experiment. The primary electron donor (phenol) and TEAs (nitrate and sulphate) compare well, as do the end products: total inorganic carbon (TIC), methane, Fe^{2+}, and Mn^{2+} (Figure 2.2A, B, C). The intermediate species, hydrogen and acetate both follow the same general pattern as the experimental data (Figure 2.2E,G). The relatively good agreement between observed and simulated solute concentrations supports the conceptual model for syntrophic biodegradation of phenol and competitive exclusion between some TEAPs, and concomitant substrate utilisation for others. The simulated TEAP rates allow a quantification of the extent to which TEAPs exhibit competitive exclusion or concomitant behaviour. An important conclusion is that, at least to some degree, invoking competitive exclusion was necessary in the mathematical formulation to describe this dataset.

2.3.1 Kinetic microbial reactions

Table 2.2 presents the final calibrated set of 13 balanced kinetic redox reactions. Five microbial populations are simulated, which consist of fermenters (CH_2O_F), nitrate reducers (CH_2O_N), iron and manganese oxide reducers (CH_2O_I), sulphate reducers (CH_2O_S), and acetoclastic methanogens (CH_2O_M). Iron and manganese oxide reducers are grouped together because the observed aqueous metabolic products Fe^{2+} and Mn^{2+} appear to follow the same temporal trend (Figure 2.2C). Biomass population growth was simulated for the heterotrophic, acetate and phenol consuming, reactions but not for the autotrophic $H_{2(aq)}$ consuming reactions. This was because the calculated ΔG_r for each of the heterotrophic reactions was typically negative enough

to allow significant biomass yield, while this was not the case for the autotrophic utilisation of $H_{2(aq)}$ as an energy source.

Figure 2.2D shows the simulated biomass population growth. As well as forming a significant part of the overall carbon balance, it strongly influences the timing of the heterotrophic reactions. The simulated fermenters exhibit growth for the whole duration, while the simulated TEAP populations grow sequentially roughly in the order: denitrifiers, sulphate reducers and metal oxide reducers, and lastly acetoclastic methanogens (reactions F0, A1, A4, A2&A3, A5 respectively). Table 2.3 presents the parameter set, which provided the best simulated representation of the experimental data.

Table 2.2 Stoichiometries and biomass yields for kinetic redox reactions

No.	Stoichiometry	Biomass Yield, Y
F0	$C_6H_6O + 5H_2O \rightarrow 2.4CH_3COOH + 1.2CH_2O_F + 2H_2$	20%
A1	$CH_3COOH + 1.28NO_3^- + 0.64H_2O \rightarrow 1.6CO_3^{2-} + 0.4CH_2O_N + 0.64N_2 + 1.92H^+$	20%
A2	$CH_3COOH + 1.6MnO_2 + 1.6H^+ \rightarrow 0.8CO_3^{2-} + 1.2CH_2O_I + 1.6Mn^{2+} + 1.6H_2O$	60%
A3	$CH_3COOH + 3.2FeOOH + 4.8H^+ \rightarrow 0.8CO_3^{2-} + 1.2CH_2O_I + 3.2Fe^{2+} + 4.8H_2O$	60%
A4	$CH_3COOH + 0.95SO_4^{2-} \rightarrow 1.9CO_3^{2-} + 0.1CH_2O_S + 0.95HS^- + 2.85H^+$	5%
A5	$CH_3COOH + 0.6H_2O \rightarrow 0.6CO_3^{2-} + 0.8CH_2O_M + 0.6CH_4 + 1.2H^+$	40%
H1	$H_2 + 0.4NO_3^- + 0.4H^+ \rightarrow 1.2H_2O + 0.2N_2$	-
H2a	$H_2 + MnO_{2,MR} + 2H^+ \rightarrow 2H_2O + Mn^{2+}$	-
H2b	$H_2 + MnO_{2,LR} + 2H^+ \rightarrow 2H_2O + Mn^{2+}$	-
H3a	$H_2 + 2FeOOH_{MR} + 4H^+ \rightarrow 4H_2O + 2Fe^{2+}$	-
H3b	$H_2 + 2FeOOH_{LR} + 4H^+ \rightarrow 4H_2O + 2Fe^{2+}$	-
H4	$H_2 + 0.25SO_4^{2-} + 0.25H^+ \rightarrow H_2O + 0.25HS^-$	-
H5	$H_2 + 0.25CO_3^{2-} + 0.5H^+ \rightarrow 0.75H_2O + 0.25CH_4$	-

Biomass yield defined as (no. mols C as CH_2O) / (no. mols C as substrate), and is calibrated. Note that for TEAPs utilising H_2 the metal oxides are split into two phases: MR – More Reactive phase, LR – Less Reactive phase. H_2 utilising TEAPs do not produce biomass (CH_2O) because the calculated ΔG_r was too small to allow a significant yield when compared with the acetate utilising TEAPs.

Table 2.3 Rate determining parameters for simulation of kinetic redox reactions, parameters as in (2.1)

	Substrate, S	*TEA*	Biomass,X	k_{max}	$X_{INITIAL}$	Half saturation constants		Inhibition terms	Calculated rate constant	Calculated half life
				[s^{-1}]	[mol/l]	K_{M_S} [mol/l]	K_{M_TEA} [mol/l]	I, K_I [mol/l]	k_1,[1/day]	$t_{1/2}$,[day]
Fermentation of phenol										
F0	C_6H_6O	-	CH_2O_F	$3.8\times10^{-8,a}$	$5.0\times10^{-4,a}$	$1.1\times10^{-4,b}$	-	-	1.14×10^{-2}	61
Acetate utilising reactions										
A1	CH_3COOH	NO_3^-	CH_2O_N	$2.0\times10^{-5,a}$	$1.0\times10^{-4,a}$	$5.0\times10^{-4,a}$	$8.1\times10^{-6,b}$	-	4.49×10^{-1}	1.5
A2	CH_3COOH	$Mn(IV)_{(s)}$	$CH2O_I$	$9.0\times10^{-8,a}$	$2.0\times10^{-5,a}$	$5.0\times10^{-4,a}$	-	NO_3^-, $1.6\times10^{-5,b}$	6.24×10^{-3}	111
A3	CH_3COOH	$Fe(III)_{(s)}$	$CH2O_I$	$4.1\times10^{-7,a}$	$2.0\times10^{-5,a}$	$5.0\times10^{-4,a}$	-	NO_3^-, $1.6\times10^{-5,b}$	2.85×10^{-2}	24
A4	CH_3COOH	SO_4^{2-}	CH_2O_S	$1.0\times10^{-5,a}$	$2.5\times10^{-7,a}$	$1.0\times10^{-4,a}$	$1.6\times10^{-4,b}$	NO_3^-, $1.6\times10^{-5,b}$	2.12×10^{-2}	33
A5	CH_3COOH	-	CH_2O_M	$8.0\times10^{-7,a}$	$5.0\times10^{-9,a}$	$5.0\times10^{-4,a}$	-	NO_3^-, $1.6\times10^{-5,b}$ & SO_4^{2-}, $1.6\times10^{-5,f}$	1.91×10^{-3}	363
Hydrogen utilising reactions				k_{max} [mol/L/s]						
H1	H_2	NO_3^-	-	$1.0\times10^{-7,a}$	$1.0\times10^{-7,e}$	$8.1\times10^{-6,b}$	-	-	5.72×10^{4}	1.21×10^{-5}
H2a	H_2	$Mn(IV)_{(s)}$	-	$1.1\times10^{-8,a}$	$5.0\times10^{-7,d}$	-	-	-	1.80×10^{3}	3.85×10^{-4}
H3a	H_2	$Fe(III)_{(s)}$	-	$5.0\times10^{-8,a}$	$5.0\times10^{-7,d}$	-	-	-	8.18×10^{3}	8.47×10^{-5}
H4	H_2	SO_4^{2-}	-	$9.0\times10^{-8,a}$	$1.0\times10^{-6,c}$	$1.6\times10^{-4,b}$	$FeOOH_{MR}$, $1.0\times10^{-12,a}$	5.71×10^{3}	1.21×10^{-4}	
H2b	H_2	$Mn(IV)_{(s)}$	-	$1.1\times10^{-8,a}$	$5.0\times10^{-7,d}$	-	$FeOOH_{MR}$, $1.0\times10^{-13,a}$ & SO_4^{2-}, $1.0\times10^{-5,g}$	1.83×10^{3}	3.78×10^{-4}	
H3b	H_2	$Fe(III)_{(s)}$	-	$5.0\times10^{-8,a}$	$5.0\times10^{-7,d}$	-	$FeOOH_{MR}$, $1.0\times10^{-13,a}$ & SO_4^{2-}, $1.0\times10^{-5,g}$	8.31×10^{3}	8.34×10^{-5}	
H5	H_2	TIC	-	$1.0\times10^{-7,a}$	$5.0\times10^{-6,c}$	-	$FeOOH_{LR}$, $1.0\times10^{-6,a}$	1.71×10^{3}	4.05×10^{-4}	

Notes: MR – More Reactive phase, LR – Less Reactive phase, a – calibrated, b – estimated from literature values in (Mayer et al. 2001), c – value from range in (Zehnder 1988), d - assumed value, lower than value for sulphate reduction due to expected competitive advantage of iron and manganese reduction over sulphate reduction, e - assumed value, lower than value for iron reduction due to expected competitive advantage of denitrification over iron reduction, f – assumed same value as nitrate inhibition parameter, g – assumed similar value to sulphate inhibition of acetate TEAPs. These first order pseudo rate constants, k_1 and half lives $t_{1/2}$ are calculated assuming: rate = $k_1 S$ and $t_{1/2}=ln2/k_1$ where rate is the simulated reaction rate at times when the specific reaction is considered to have been active.

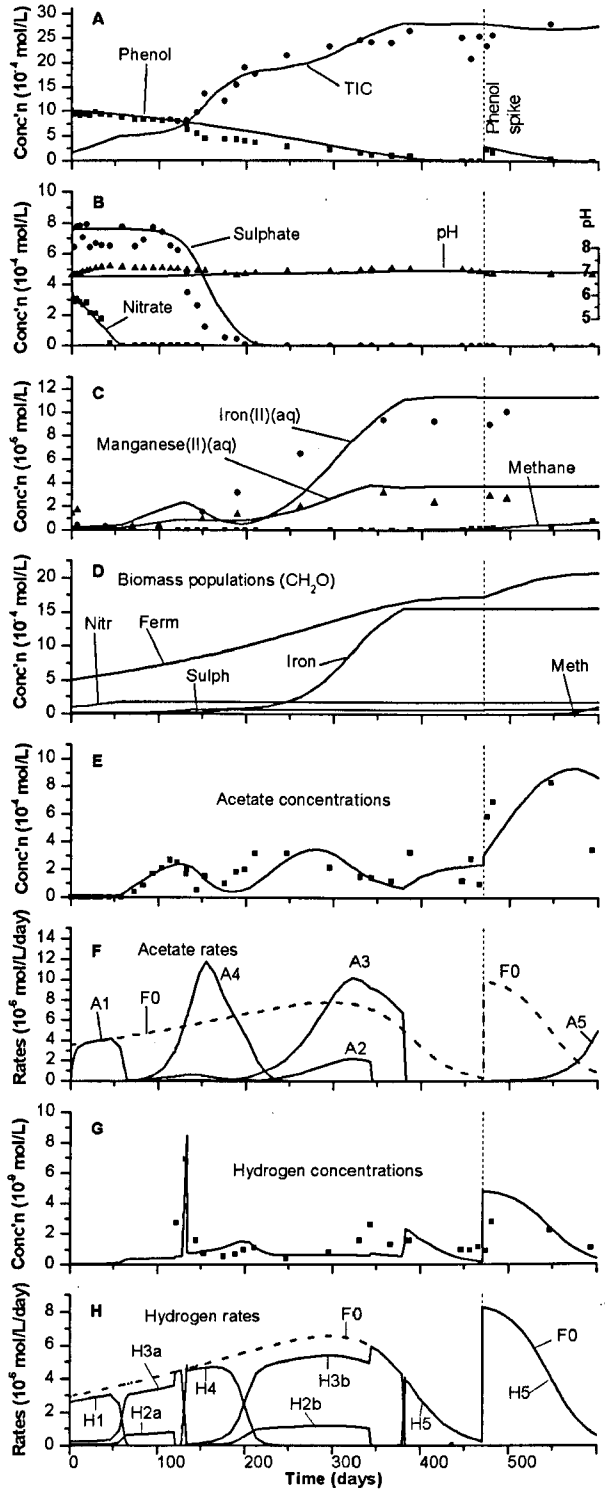

Figure 2.2 Results of microcosm simulation.
Symbols are experimental data, lines are simulated with MIN3P, respective species are indicated on the plots. The vertical line at 470 days represents the phenol spike. Key to individual abbreviations on plots: Plot A: TIC – Total Inorganic Carbon. Plot D, Biomass populations: Ferm – fermenters (CH_2O_F), Nitr – denitrifiers (CH_2O_N), Sulph - sulphate reducers (CH_2O_S), Meth – methanogens (CH_2O_M), Iron – iron and manganese reducers (CH_2O_I). Plots F,H: labels F0, A1-5, H1-5 refer to reactions in Table 2.2, reaction F0 (dashed line) shows production rates, others are consumption rates (solid lines).

2.3.2 Acetate

Acetate is produced throughout the experiment by fermentation (F0 in Table 2.2 and Figure 2.2E,F). The fermentation rate increases as the CH_2O_F population grows (Figure 2.2D) during the first year, while the substrate phenol concentration is high, and then decreases as the phenol concentration drops. After the phenol spike, the CH_2O_F population ferments the phenol more rapidly than at the start of the experiment because the population has grown. Overall, the CH_2O_F exhibits pseudo-linear growth as a combined result of the growth process and substrate limitation. The acetate produced is competitively consumed by TEAP processes (Table 2.2, A1-5) as described in the following interpretation. Initially then, nitrate reduction (Table 2.2, A1) keeps acetate concentrations low by consuming it as fast as it is produced. Reactions A2-A5 are inhibited by nitrate (Table 2.3), so they are competitively excluded until nitrate is consumed. After nitrate is consumed CH_2O_I and CH_2O_S begin to grow. Of the two the sulphate reducers are the first to have a significant effect on the acetate concentration, decreasing it rapidly, but as sulphate is soon exhausted acetate concentrations rise again after about 200 days (Figure 2.2E,F). The metal oxide reducers are effectively excluded during this period due to their small population size. After 250 days the CH_2O_I population is large enough to decrease the acetate concentration. This continues until the available metal oxides are consumed, observed to be when the Fe(II)(aq) and Mn(II)(aq) concentrations reach a plateau at about 360 days (Figure 2.2C). Meanwhile, the acetoclastic methanogens CH_2O_M have started growing, since they are no longer inhibited by sulphate or nitrate (Table 2.3, A5). Near the end of the experiment the methanogenic population has grown sufficiently to cause the final observed decrease in acetate concentration.

The numerical modelling of the temporal evolution of the acetate intermediate has required the use of six kinetic microbial growth reactions with a total of 25 parameters, some of which were taken from literature sources (see Table 2.3). This constitutes a relatively complex formulation considering the relative simplicity of the conceptual model for syntrophic biodegradation with partial to complete competitive exclusion. Since the FKM uses a more general description of biodegradation than CE or PE provide, the relatively large number of parameters is to be expected. However, only with this approach was it possible to reproduce the dynamics of acetate evolution in response to transient changes in TEAPs. CE and PE approaches could not be applied to acetate.

2.3.3 Hydrogen

Observations show that $H_{2(aq)}$ concentrations are relatively constant at c.1nM, except for a series of peaks (Figure 2.2G). The steady $H_{2(aq)}$ concentration demonstrates that the relative rates of $H_{2(aq)}$ production and consumption are relatively constant throughout the experiment, regardless of TEA being utilised. $H_{2(aq)}$ is present at much lower concentrations than acetate, i.e. nM rather

than mM, despite being produced at roughly the same rate by fermentation (F0, Table 2.2). This is a consequence of the dynamic behaviour of acetate consuming processes and demonstrates that for $H_{2(aq)}$ the total consumption rate must be very close to its production rate throughout the experiment. Inspection of Fig. 2F and 2H support this, where significant periods of the simulated experiment are characterised by negligible acetate utilisation rates while $H_{2(aq)}$ utilisation rates remain similar to fermentation rates throughout. The $H_{2(aq)}$ peaks are assumed to occur during transition from one TEAP to the next, presumably due to cessation of one process prior to the onset of the next (Hoehler et al. 1999, Christensen et al. 2000). The peaks are observed to occur at: the onset of sulphate reduction, 7nM at 125days, at the cessation of sulphate reduction, 1.1nM at 210 days, close to the cessation of iron and manganese oxide reduction, 3nM at 340 days, and finally just after the phenol spike, 3nM at 480days. As described above, sulphate reduction appears to occur within the longer period of metal oxide reduction. Consequently, it appears that a transition is made from metal oxide reduction to sulphate reduction, and this is followed by a subsequent transition back to metal oxide reduction. This behaviour is possibly explained by considering the available metal oxides to consist of two distinct parts: a more reactive fraction and a less reactive fraction. In general, metal oxides in aquifers are known to exhibit a variety of mineral phases which have different levels of reactivity and bioavailability (Roden and Zachara 1996, Christensen et al. 2000). It is proposed that the more reactive oxides are consumed prior to sulphate reduction, and the less reactive oxides are consumed after sulphate reduction. This empirical splitting of the oxides is only applied to the $H_{2(aq)}$ TEAPs and not to the acetate TEAPS. Thus while this has a large effect on the $H_{2(aq)}$ concentrations, the overall effect on the system dynamics is minimal since the acetate TEAPs show the concomitant metal oxide and sulphate reduction (Fig. 2F) which is observed at the gross scale.

During the first 60 days nitrate reduction (H1) maintains $H_{2(aq)}$ concentration at a relatively low value. Subsequently, the more reactive metal oxide fractions are consumed (H2a, H3a). At about day 150 a short transition to sulphate reduction (H4) causes the sharp $H_{2(aq)}$ peak. Sulphate reduction shows a relatively long transition to the less reactive oxide TEAPs (H2b, H3b) which gives the relatively broad second $H_{2(aq)}$ peak. The final transition to methanogenesis (H5) occurs when all the available metal oxides are effectively exhausted, and causes the third $H_{2(aq)}$ peak. The final $H_{2(aq)}$ peak is a result of the increased production rate (F0) following the phenol spike.

The modelled data supports the notion that $H_{2(aq)}$ peaks occur during TEAP transitions and that metal oxides consisting of fractions of varying reactivity may affect the TEAP sequence significantly. The first large $H_{2(aq)}$ peak demonstrates effective competitive exclusion, since it is an accumulation of $H_{2(aq)}$ occurring after the exhaustion of the $FeOOH_{MR}$ (H3a). Presumably a short lag time occurs before the onset of sulphate reduction (H4) and permits the $H_{2(aq)}$ accumulation. Examination of the TEAP rates (Figure 2.2H) shows that the third $H_{2(aq)}$ peak is

also caused by competitive exclusion and a rapid transition from one TEAP to the next. However, the second $H_{2(aq)}$ peak is caused by a gradual transition between TEAPs that overlap temporally, implying that in this case competitive exclusion is not complete.

2.3.4 Sensitivity analysis of biodegradation parameters

Simulation results have a relatively low, yet observable, sensitivity to Monod and inhibition parameters, K_M and K_I. Generally, an order of magnitude change in value is required to give observable differences, depending on the exact parameter value compared with that for S or I respectively. k_{max} is a more sensitive parameter which has a linear effect on its specific reaction rate, but a non-linear effect on the timing and rates of other ongoing and subsequent reactions due to coupling through substrate competition processes described above; a 10% change in value results in an observable shift in the results. This implies that k_{max} values are more precisely constrained than K_M and K_I; i.e. the latter two parameter values are less certain (cf. Schirmer et al. 1999).

The calibrated biomass yield parameters, Y, are moderately sensitive; simulations show that a change of about 10% in parameter value for any of them significantly changes the results in terms of the rates and timing of reactions. Therefore, the Y values are well constrained by the available experimental observations. The initial biomass parameters are relatively insensitive. Typically, the initial biomass had to be altered by an order of magnitude to significantly change the timing of reactions. This sensitivity does increase with decreasing growth rate, i.e. due to lower values for Y, or k_{max}. The values for initial biomass were not significant in terms of the overall carbon balance, so the values were simply calibrated to within an order of magnitude in order to give appropriate results. In general, it is important to note that the aim of this work is not to test particular parameter values, but rather to investigate the conceptual approaches that allow the simulation of the temporal evolution of biodegradation intermediates.

2.3.5 Carbon and electron balance

The simulation shows that the initial mass of carbon in phenol is converted to approximately 70% aqueous products (comprised of 40% sampling loss during the experiment, and 30% remaining in the microcosm), 5% gaseous products, and 25% biomass. The simulated electron balance is presented in Table 2.4 and demonstrates that a significant proportion of TEAP activity is attributed to reduction of iron and manganese oxides in the aquifer material.

Table 2.4 Simulated TEA and electron balance

TEA	Total TEA Consumption[a] Moles	Electron equivalents	Ratio of substrates consumed by each TEA	End products for TEA species.
NO_3^-	2.9×10^{-4}	1.5×10^{-3} (13%)	20% H_2 : 80% Ac	N(0): 100% N_2
$Mn(IV)_{(s)}$	3.6×10^{-4}	7.2×10^{-4} (6%)	45% H_2 : 55% Ac	Mn(II): 6% $Mn^{2+}_{(aq)}$; 94% $Mn_{(surface)}$[c]
$Fe(III)_{(s)}$	4.2×10^{-3}	4.2×10^{-3} (37%)	42% H_2 : 58% Ac	Fe(II): 2% $Fe^{2+}_{(aq)}$; 14% $FeS_{(s)}$; 84% $Fe_{(surface)}$[d]
SO_4^{2-}	6.1×10^{-4}	4.9×10^{-3} (43%)	10% H_2 : 90% Ac	S(-II): 100% $FeS_{(s)}$
CO_2	2.1×10^{-5}	1.7×10^{-4} (1%)	100% H_2[b]	C(-IV): 100% CH_4

Notes: Ac. – Acetate. a. Totals are cumulative up to day 470, i.e. not including reactions after the phenol spike, which were principally methanogenic. b. -Acetate does not use CO_2 as TEA, but methane is produced via acetoclastic methanogenesis, reaction A5, Table 2.2. c. - $Mn_{(surface)}$ includes $>FeOMn^+_{(w)}$ as in Table 2.1. d. - $Fe_{(surface)}$ includes $>FeOFe^+_{(w)}$ and $>FeOFeOH_{(w)}$ as in Table 2.1.

The biomass yields of iron and manganese reduction (A2 and A3, Table 2.2) are high, but are well constrained by several factors, which include the system carbon balance, the system electron balance, the necessity for calculated $\Delta G_r<0$ for the balanced stoichiometric growth reactions, and the mass of mineral oxides available. The measured mass available in the aquifer material corresponds to a volume fraction of $5x10^{-3}$ FeOOH and $4x10^{-5}$ MnO_2 respectively (Thornton et al. 2001b). The model consumes $1x10^{-3}$ FeOOH and $8x10^{-5}$ MnO_2 which agrees reasonably well with the measured data and also meets the constraints imposed by the system electron balance. The reaction energetics for the high biomass yield suggests that the microbes may be very efficient, and a similar high value of biomass yield for FeOOH utilisation during biodegradation of phenolic compounds in the same UK sandstone aquifer was reported by Broholm (Broholm and Arvin 2000). Moreover, lowering the biomass yield value from that used to create the stoichiometry A3 (Table 2.2) would produce a new stoichiometry which would demand a larger quantity of FeOOH. This produced much higher pH values, up to pH 12 in a number of test simulations. This would also significantly affect the timing and rates of this and of subsequent TEAPs. Thus, the model results do not support a lower biomass yield in this case due to the multiple constraints.

2.3.6 Iron(II) and manganese(II)

Figure 2.2C shows dissolved Fe(II) and Mn(II) concentrations that increase during the experiment to a final constant concentration which is significantly less than the amount of mineral Fe(III) and Mn(III,IV) that is oxidised, according to the electron balance calculations in Table 2.4. Precipitation and surface complexation of microbiologically produced $Fe(II)_{(aq)}$ and $Mn(II)_{(aq)}$, can occur generally in groundwater, and can limit dissolved concentrations of these species. The MIN3P code calculated the aqueous solution to be slightly supersaturated with respect to both siderite ($FeCO_3$) and rhodochrosite ($MnCO_3$) after about 350 days, implying that the aqueous solution could be in thermodynamic equilibrium with the carbonate mineral phases. However, inclusion of these phases led to a poor representation of dissolved $Fe(II)_{(aq)}$, $Mn(II)_{(aq)}$ and TIC data. Consequently, these carbonate minerals were not included in the model, although

their precipitation cannot be conclusively ruled out. Precipitation of $FeS_{(s)}$ was assumed to be rapid because aqueous $S(-II)_{(aq)}$ concentrations remained below analytical detection limits during the experiment. $FeS_{(s)}$ precipitation by itself was insufficient to account for all the Fe(II) mass (only 14%, see Table 2.4) or to prevent the pH rising to unrealistically high values.

The surface complexation reactions (Table 2.1) not only limited $Fe(II)_{(aq)}$ and $Mn(II)_{(aq)}$ concentrations, but also acted as a pH buffer for the simulated system. The simulated pH (Fig. 2B) shows a maximum discrepancy of 0.5 pH units from the data and this occurred at an early stage when iron and manganese reduction were not prevalent. The pH buffering was important since large amounts of iron reduction produces high alkalinity and pH. Calcite buffering was not present in this system since the amount of calcite in the rock matrix is negligible (Thornton et al. 2001b). Considering that the surface complexation parameters used were derived from synthetic systems with pure electrolytes, whereas our data is from a complex aquifer based microcosm with actual aquifer solids in a complicated aqueous medium, the simulated results are encouraging, especially since the simulated pH closely follows the observations when the adsorption of Fe(II) and Mn(II) is greatest, i.e. later in the experiment. A specific surface area of $6.9m^2/g$, the mean value for the Triassic sandstone aquifer matrix, was used for surface complexation reactions. If the surface site concentration was significantly increased there was little effect on the simulated results. However, decreasing the total concentration by 10% resulted in a significant shift towards higher pH and $Fe(II)_{(aq)}$ concentration, while an order of magnitude reduction resulted in extremely high pH and $Fe(II)_{(aq)}$ concentration. Thus, the results are sensitive to lower values of surface complexation site concentration.

Table 2.4 gives the final distribution of the Fe(II) species and phases. The simulated total mass of Fe(II) is relatively well constrained by the system electron balance, and the results for mass of dissolved $Fe(II)_{(aq)}$ are consistent with surface complexation as the main inorganic process controlling the $Fe(II)_{(aq)}$ and $Mn(II)_{(aq)}$ concentrations.

These results suggest that adsorption of $Fe(II)_{(aq)}$ and $Mn(II)_{(aq)}$ should be considered when quantifying electron balances for aquifer redox processes since the measured $Fe(II)_{(aq)}$ concentration is not likely to represent the total mass of Fe(II) (Christensen et al. 2000). The previous field scale modelling of the Four Ashes phenolics plume did not consider surface complexation reactions. If the simulation results here are indicative of field conditions then the extent of iron reduction may have been greatly underestimated at field scale by neglecting the adsorption of $Fe(II)_{(aq)}$.

2.3.7 Syntrophic biodegradation

The modelling of the acetate and $H_{2(aq)}$ intermediates supports the conceptualisation of syntrophic biodegradation as two sequential reactions. However, it does not preclude simulation of phenol biodegradation as direct mineralisation of phenol in a single step if the objective is

simply to simulate phenol mass loss (Mayer et al. 2001).

The reaction time scales quantified as half-life values in Table 2.3 indicate the characteristic time scale for each reaction to approach steady-state. The $H_{2(aq)}$ consuming TEAPs tend to return to steady-state very quickly relative to fermentation or acetate consuming TEAPs. Although it is possible to simulate the dynamic peaks of $H_{2(aq)}$, the steady-state assumption often invoked to interpret $H_{2(aq)}$ data obtained from field sites is supported by the short characteristic response times and does not appear to be restrictive. For acetate, on the other hand, a steady-state assumption would appear not to be suitable as characteristic reaction times and approach to a steady-state condition are longer. Furthermore, the results show it would be inappropriate to assume that fermentation is rate limiting with respect to acetate TEAPs, since acetate consumption rates are, at times, significantly slower than fermentation. However, competition for the acetate substrate occurs throughout the experiment.

2.3.8 Extent of competitive exclusion

The FKM used allows us to assess the extent of competitive exclusion in a dynamic system by inspection of the TEAP rates. For example, for acetate consumption, the simulated results show sulphate reduction to be the dominant TEAP from day 60 to day 220, although metal oxide TEAPs are evident during the same period (Figure 2.2F). For $H_{2(aq)}$ consumption, there is one dominant TEAP at almost all times, especially if the metal oxide reduction TEAPs are considered together (Figure 2.2H). The simulation of the observed TEAP transitions is considered to support competitive exclusion (see above). However, during the transition from sulphate reducing to metal oxide reducing TEAPs, around day 200, the two occur concomitantly for both acetate and H_2 consumption. Thus, at most times there exists one dominant reaction, and these reactions generally occur sequentially which largely supports the concept of competitive exclusion. However, the observed and simulated concomitant TEAPs support the notion that complete competitive exclusion does not always occur.

2.3.9 Comparison of CE and PE Approaches

A recent review of the CE and PE approaches (Banwart and Thornton 2003) compared the quantitative interpretation of $H_{2(aq)}$ concentrations through mathematical representations of the underlying conceptual models. The CE approach can be mathematically represented as a closed, steady-state chemical system where fermentation is described by zero-order kinetics, followed by TEAPs as irreversible reactions described by first-order kinetics with respect to $H_{2(aq)}$. The zero- and first-order rate expressions correspond to limiting forms of the single- and dual-Monod expressions, respectively. Unique values for $H_{2(aq)}$ concentration result for each TEAP if biomass concentration is constant and the Monod kinetic constants for TEAPs are sufficiently different (Banwart and Thornton 2003).

The PE approach also results from considering a closed, steady-state chemical system with fermentation as a slow irreversible process described by zero-order kinetics followed by $H_{2(aq)}$ utilisation described by first-order kinetics. The key difference in this case is that the relatively more rapid utilisation of $H_{2(aq)}$ must be a reversible process with a rapid reverse reaction such that steady-state results in chemical equilibrium according to the Principle of Detailed Balancing (Nordstrom and Munoz 1994, chapter 1) for the $H^+/H_{2(aq)}$ redox couple (Banwart and Thornton 2003). The steady-state analysis cited here applies locally on time scales of the reaction processes, and does not refer to the dynamic time scales of migrating contaminant plumes, which are much greater.

The original quantitative interpretation of competitive exclusion was that it should lead to characteristic steady-state $H_{2(aq)}$ concentrations for iron reduction (0.2 to1.0nM), sulphate reduction (1 to 6nM) or methanogenesis (7 to 20nM) (Lovley and Goodwin 1988). In this study, the observed $H_{2(aq)}$ concentrations (0.6 to 7nM) would lead to a prediction that sulphate reduction is dominant throughout the experiment. This is clearly not the case. Further, the original CE interpretation suggests that $H_{2(aq)}$ should increase with time as TEAPs are consumed. Again, this is contrary to the observed pattern of transient peaks followed by a return to steady-state $H_{2(aq)}$ values of c.1nM (Figure 2.2G). Thus, although the concepts of steady-state $H_{2(aq)}$ arising from syntrophic biodegradation and competitive exclusion are supported by the data and simulation results, the expected ranges in $H_{2(aq)}$ concentration for the respective TEAPs are not consistent with the data.

The PE approach assumption that fermentation is rate limiting and $H_{2(aq)}$ utilising TEAPs are much faster is also supported by the simulation. PE also assumes that each TEAP will reduce the $H_{2(aq)}$ concentration such that its free energy ΔG_r will be decreased to a threshold value representing the minimum energy required by the bacteria for ATP synthesis (Christensen et al. 2000). TEAPs may occur simultaneously if the respective values for ΔG_r are both below their respective thresholds. Figure 2.3 shows ΔG_r values that were calculated from the observed data using the method of Jakobsen *et al* (Jakobsen et al. 1998).

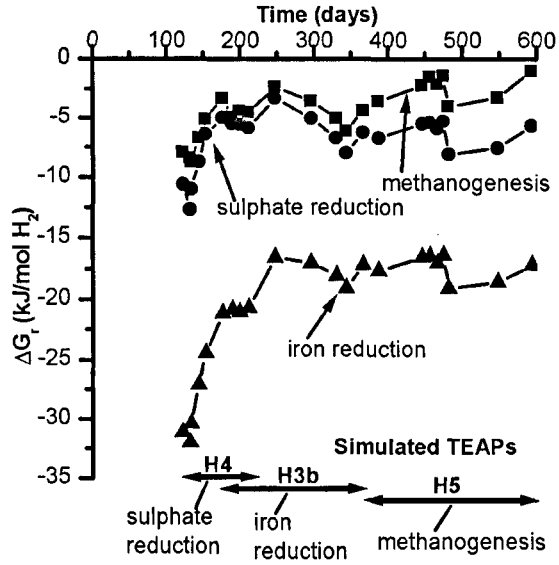

Figure 2.3 Evolution of TEAP ΔG_r with time.
ΔG_r is calculated following method in (Jakobsen et al. 1998) using the observed data and simulated concentrations of HS^-. Includes temperature correction using Van't Hoff equation to 20°C, although this typically changes the result by <1kJ/mol. For iron reduction the curve is calculated for the least stable lepidocrocite (Jakobsen et al. 1998). If the most stable lepidocrocite (Jakobsen et al. 1998) is used the curve moves up by +20kJ/mol to be partly in the positive ΔG_r region. For comparison, the range of steady-state threshold values for TEAPs are reported as: methanogenesis, –0.5 kJ/mol to –12.5 kJ/mol (Jakobsen et al. 1998, Conrad 1999, Hoehler et al. 2001), sulphate reducers, -4.3 kJ/mol to –18kJ/mol (Jakobsen et al. 1998, Hoehler et al. 2001, Thornton et al. 2001b), iron reducers, -7 kJ/mol -10.5 kJ/mol (Jakobsen et al. 1998, Jakobsen and Postma 1999).

Figure 2.3 shows that the free energy of TEAP reactions is negative. The observed $H_{2(aq)}$ data strongly influences the transient trend of the calculated values of ΔG_r. The absolute value of the FeOOH TEAP ΔG_r is uncertain since the reactivity of the iron-bearing phase is unknown (Jakobsen et al. 1998). Figure 2.3 shows data for the thermodynamically least stable form of the mineral phase lepidocrocite. A lepidocrocite phase of intermediate thermodynamic stability would increase ΔG_r by about 10kJ/mol $H_{2(aq)}$, to give a ΔG_r much closer to the calculated values for the other two TEAPs. Even considering the uncertainty of the calculations, there do not appear to be any steady-state ΔG_r values which are consistent with previously reported threshold ΔG_r values (see Figure 2.3 caption). There is no obvious relationship between calculated TEAP ΔG_r and the periods when the same TEAP is dominant (see Figure 2.3). For example, the ΔG_r of methanogenesis is most negative early on, but the process appears to be inhibited until late in the experiment, although trace methane amounts were detected which allowed the calculation of ΔG_r.

Previous data interpreted using the PE approach have often been characterised by steady ΔG_r values during the period that the associated TEAP was observed. This is not the case for sulphate reduction here. The change in ΔG_r during the observed period of sulphate loss indicates either a kinetic process proceeding towards equilibrium, or in the case of control by partial equilibrium, the threshold energy for the reaction to occur must be increasing with time. Most

TEAPs in this study exhibit free energies that, when compared to the least negative threshold values in the literature (see Figure 2.3 caption), predict the process to proceed, yet observations here do not show that they proceed. Therefore, in this microcosm the standard PE approach does not appear to be able to predict the active TEAPs. This is despite the fact that its conceptual basis, including relatively slow fermentation compared with relatively fast TEAPs and locally steady-state $H_{2(aq)}$ concentrations, is supported.

The possibility of extending the present PE approach to incorporate system dynamics cannot be excluded. The approach would require a kinetic description of the fermentation process to explain system dynamics. Further, the range of literature values for threshold ΔG_r illustrates that different systems, at quasi-steady-state, give different values. This suggests that in a dynamic system the threshold values could change with time, which is apparent for sulphate reduction in this study. A system-specific response would reduce the possibility for a predictive capability of a dynamic PE approach.

2.3.10 Model complexity

The model for syntrophic biodegradation presented in this work is more complex than either of the established approaches for diagnosing TEAPs from $H_{2(aq)}$ concentrations. In contrast with the CE competitive exclusion approach, this work does not assume a general steady-state $H_{2(aq)}$ concentration. Instead, it formulates kinetic equations based on Monod growth kinetics, which allow competing processes to occur according to model parameter values. Modelling of additional laboratory and field systems is needed in order to determine if parameter values for the Monod kinetics are general, vary predictably or are site- and system- specific. If values are site-specific, this implies that it will be necessary in the future to model each system separately in order to understand the processes occurring. Even then, although it is generally observed that $H_{2(aq)}$ concentrations do fall in a remarkably narrow and predictable range (Christensen et al. 2000), there may not exist specific ranges of $H_{2(aq)}$ for each TEAP. In contrast with the PE partial equilibrium approach the model in this work does not calculate or utilise values of ΔG_r for TEAPs. However, the model does simulate sequential utilisation of H_2 by TEAPs in spite of relatively steady ΔG_r values during much of the experiment.

Neither the CE, competitive exclusion, nor the PE, partial equilibrium, approach can quantifiably diagnose the active TEAPs for this particular dataset. This suggests that both approaches may be oversimplifying the true situation. In contrast, the fully kinetic simulation used in this work was able to represent the observed data through consideration of specific microbial processes. The FKM description is more general than CE or PE approaches, particularly in that it characterises system dynamics with net biomass growth rate through the empirical Monod kinetic formulation. In addition CE and PE cannot describe the acetate intermediate which forms an important part of the system carbon turnover.

Overall it appears that conceptual and numerical modelling of the temporal evolution of an entire biodegradation system requires a high level of complexity. This is true even though the system is relatively well characterised, is bench-scale, and does not have flow or transport. Important and necessary components of the underlying conceptual model include biomass growth, competitive exclusion of TEAPs, and mineral phase availability. Although it is quite possible that these are also important factors at field sites, they are not routinely considered in field scale numerical modelling, not only because this level of complexity renders the associated size of the numerical problem too large to be computationally tractable, but also because the supporting data is often not available.

3 Insights gained through multispecies field scale reactive transport modelling driven by biodegradation complexity.

IAN A. WATSON[1], SASCHA E. OSWALD[1], ROGER S. CROUCH[1], STEVEN F. THORNTON[1] AND STEVEN A. BANWART[1] *

1. Groundwater Protection and Restoration Group, Department of Civil and Structural Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, United Kingdom.

Abstract

Biodegradation is an important process contributing to the natural attenuation (NA) of organic contaminants in groundwater. A laboratory scale biodegradation model has been implemented into a field scale reactive transport model. The use of a two-step syntrophic biodegradation model with fermentation and respiration steps simulated kinetically, and with hydrogen and acetate as intermediate species, allowed rates to be estimated from detailed discrete field data. Important processes addressed by these simulations include: (i) long term microbial acclimatisation to the contaminant matrix, (ii) differential toxicity effects between fermentation and respiration affecting hydrogen distribution, (iii) bioavailability of FeOOH mineral electron acceptor decreasing with time, and (iv) sorption of microbially produced Fe(II) modelled with surface complexation reactions. The new findings suggest that plume core processes are much more important than previously thought, possibly with a greater impact than plume fringe processes, based on a 47 year cumulative mass balance. Considering a heterogeneous flow field and a spatially variable source zone resulted in more mass being degraded, but with no effect on the ratio of mass loss between fringe and core processes. These results indicate strongly that the choice of conceptual model for reactive processes plays a dominant role in the quantitative assessment of NA.

3.1 Introduction

Biodegradation is often the most important process contributing to the natural attenuation (NA), or engineered in situ bioremediation, of organic contaminants in groundwater (Smets and Pritchard 2003). Reactive transport modelling can be used to assess the potential for NA by quantifying biodegradation rates rates, and other relevant processes (e.g. dispersion, sorption, precipitation and dissolution reactions). A recent review of biogeochemical transport modelling approaches shows that the biodegradation process is modelled as either a single-step or a two-step process (Brun and Engesgaard 2002). The two-step approach considers fermentation of the

organic substrate to intermediate species, e.g. hydrogen and acetate, followed by respiration where the intermediates are consumed by terminal electron acceptors (TEAs). The fermentation is overall rate limiting and is thus treated with kinetic expressions. The respiration, or terminal electron accepting processes (TEAPs), is considered to occur much faster and, in the partial equilibrium approach, is treated as being close to thermodynamic equilibrium (McNab and Narasimhan 1994, Jakobsen and Postma 1999, Prommer et al. 1999b, Brun and Engesgaard 2002). Alternatively the TEAP step can be treated kinetically which notably does not assume that the microbial TEAPs are reversible (Hunter et al. 1998, Curtis 2003).

In addition to the primary biodegradation reactions, it is generally desirable that models can account for secondary inorganic reactions, including aqueous complexation, sorption, and mineral precipitation and dissolution (Hunter et al. 1998, Mayer et al. 2001, Brun and Engesgaard 2002) since these can all affect concentrations of species measured in the field, and can have an effect on mass turnover estimates in assessing NA (Banwart and Thornton 2003). Besides the chemical complexity, i.e. primary and secondary reactions, site-specific physical factors such as dispersive mixing, heterogeneity or source zone distribution can have important effects on NA potential.

As well as reactive transport models, laboratory scale experiments may be used to demonstrate the potential for NA at field sites (e.g. Bekins et al. 1993, Nielsen et al. 1995, Broholm and Arvin 2000, Pickup et al. 2001). In recent work, a phenol degrading microcosm was simulated using a fully kinetic two-step approach focusing on hydrogen and acetate dynamics to extract TEAP rates (Watson et al. 2003b). The work highlighted several conclusions which were important in understanding the microcosm reactive system, including: (i) hydrogen concentration was useful in determining the relative rates of fermentation and respiration, (ii) microbial population activity changed with time, (iii) bioavailability of FeOOH and MnO_2 appeared to vary with time, and (iv) aqueous Fe^{2+} and Mn^{2+} behaviour could be understood in terms of surface complexation equilibria. The field site providing the innocula for the microcosm had been studied at the field scale with reactive transport modelling (Mayer et al. 2001). However, this initial interpretation of the field site did not incorporate any of the factors, above, which were important for understanding the microcosm system. Each of these factors could potentially be important at field sites, but are not routinely considered in NA assessments.

In this study the above mentioned field site provides hydrogeological and hydrochemical data that are applicable for NA assessments in general. The conceptualisation of the site is extended to include the additional complexity used to simulate reactive processes in the microcosm. The new conceptual model describes fermentation linked with TEAPs through intermediate species such as hydrogen and acetate, as well as relevant secondary inorganic reactions. This is implemented into a mathematical framework for reactive transport modelling to identify if the interpretation of plumes may be altered significantly by applying the new understanding gained

from the microcosm. In addition, the potential toxic effects of the contaminant matrix on fermentation and TEAPs, especially in the plume core where the highest concentrations exist, are considered. The site is contaminated with phenolics which are a common contaminant and may show toxic effects at higher concentrations (e.g. Arvin and Flyvbjerg 1992, Bekins et al. 1993, Nielsen et al. 1995, King et al. 1999, Lerner et al. 2000).

The initial field-scale model of the plume demonstrated that biodegradation reactions contributed to the spatial variability of observed plume concentrations (Mayer et al. 2001). However, this variability may also be due to other factors such as a spatially and temporally variable source zone, or a heterogeneous flow field. Consequently, these two physical factors are also included in the new conceptual and mathematical models. Simulations are completed both in 2D and in 3D. The new models are assessed by examining the reaction rates, spatial distributions of solute concentrations, and plume scale mass balance, and are compared with the initial, simpler, field model, and the microcosm degradation model. The value of using a more complex model is discussed in terms of the extra process understanding gained. The general applicability, and importance, to NA sites of the significant biogeochemical reactions, and physical processes, are considered.

3.2 Modelling Methods

3.2.1 Field Site description.

This modelling study was driven by detailed data obtained at the Four Ashes, UK, field site where a plume of phenolic compounds has been undergoing natural attenuation for approximately 50 years in a consolidated sandstone aquifer (Williams et al. 2001). Figure 3.1 illustrates some of the main features of the plume. The groundwater flows west with an average velocity of 10m/yr. The source zone is poorly characterised as it is under an operational industrial site. However, the source is known to have contained phenols (primarily phenol, cresols and xylenols) and mineral acids (Williams et al. 2001). The most detailed geochemical data for the plume is provided by two boreholes equipped with sample ports with a vertical spacing of 1 metre. These multi-level samplers (MLS), which approximately lie on a flow line, are BH59, 130m from the source, and BH60, 350m from the source (Thornton et al. 2001b). Annual and quarterly monitoring of these wells, including published 1998 data (Thornton et al. 2001b) and subsequent annual monitoring (data as yet unpublished), show consistent vertical profiles, which confirm the plume to be 20m thick and plunging at an angle of 3° to the west. Other site monitoring data suggests that the plume dimensions are 500m long and 130m wide (Thornton et al. 2001a), but a significant volume of the plume remains sampled by a network of monitoring wells fitted with long screens (3-10m length). Since the aim of this study is to apply the detailed microcosm model for reactive processes to the field scale,

comparisons of simulated and observed profiles are made at the MLS locations where the greatest detail in chemical data is available.

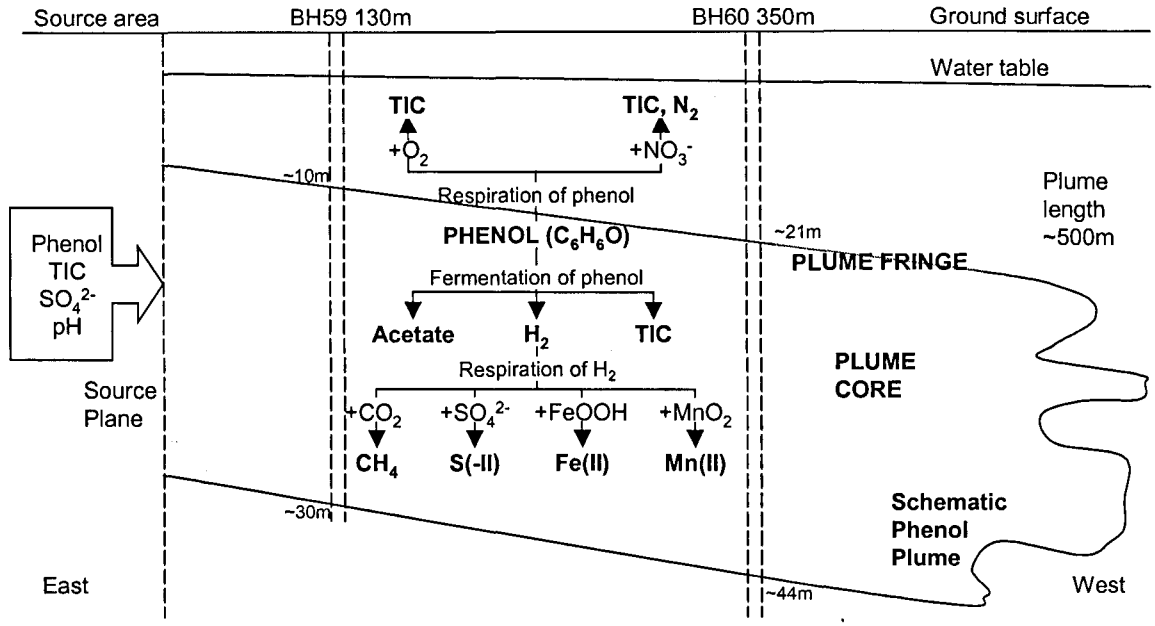

Figure 3.1 Conceptual figure of plume layout and reaction paths

3.2.2 Laboratory microcosm modelling.

A two year microcosm, innoculated with aquifer material and groundwater from the site, was interpreted conceptually and numerically modelled in considerable detail using biodegradation kinetics (Watson et al. 2003b). The study illustrated that, in order to understand the geochemical development of the system with time, a number of coupled processes had to be incorporated. These included fermentation reactions producing hydrogen and acetate, temporal variation of biomass activity, sorption of microbially produced Fe(II) and Mn(II), and splitting the pool of metal oxide minerals into "more" and "less" bioavailable fractions. These are not routinely considered in field scale biodegradation modelling studies. Thus, this study aims to implement these concepts at the field scale to identify their importance for the interpretation of plume data.

3.2.3 Previous field scale modelling.

A field scale numerical model of the plume has been completed by Mayer *et al* (2001). The plume core processes were simulated using two 1D flowlines with detailed chemistry, based upon the MLS data available at that time. The plume fringe processes were simulated using a 3D simulation with simplified chemistry, considering only denitrification by using just three species. The model results were combined to give an overall mass balance suggesting that in 47 years 2% of the phenol added had been degraded. This analysis was made without including the processes which the recent microcosm work subsequently suggested as being important.

3.2.4 Model formulation.

The new field-scale model formulation, described below, includes a substantially different reactive biodegradation concept to the previous model (Mayer et al. 2001), based upon the important processes identified in the microcosm study. In addition, the effects of phenol toxicity, a heterogeneous flow field, and a spatially, and temporally, variable planar source term, which have not been simulated previously, are investigated.

3.2.5 Biodegradation Chemistry.

Table 3.1 shows the stoichiometry of reactions used and Figure 3.1 illustrates the reaction pathways. Reactions 1 and 2 are the same as the previous field model (Mayer et al. 2001) and represent reactions occurring at the plume fringe through dispersive mixing processes. Reactions 3 to 8 are similar to those used in the microcosm model (Watson et al. 2003b) and represent syntrophic, i.e. coupled fermentation and respiration, biodegradation processes in the plume core. The fermentation of phenol was represented as two reactions degrading phenol to acetate (reaction 3, Table 3.1) or Total Inorganic Carbon (TIC) (reaction 4, Table 3.1). Mayer *et al* (2001) combined the principal phenol species, phenol, cresols, and, xylenols according to their electron demand to oxidise phenol-derived carbon, giving total phenolics as phenol. This approach has been adopted here so, henceforth, phenol, C_6H_6O, is synonymous with total phenolics.

Table 3.1 Stoichiometry of kinetic biodegradation reactions

No.	Type	Stoichiometry
1	Aerobic degradation	$C_6H_6O + 7O_2 + 3H_2O \rightarrow 6HCO_3^- + 6H^+$
2	Denitrification	$C_6H_6O + 28/5NO_3^- + 1/5H_2O \rightarrow 6HCO_3^- + 2/5H^+ + 14/5N_2$
3	Fermentation	$C_6H_6O + 5H_2O \rightarrow 3CH_3COOH + 2H_2$
4	Fermentation	$C_6H_6O + 17H_2O \rightarrow 6HCO_3^- + 6H^+ + 14H_2$
5	Manganese reduction	$H_2 + MnO_2 + 2H^+ \rightarrow 2H_2O + Mn^{2+}$
6	Iron reduction	$H_2 + 2FeOOH + 4H^+ \rightarrow 4H_2O + 2Fe^{2+}$
7	Sulphate reduction	$H_2 + 0.25SO_4^{2-} + 0.25H^+ \rightarrow H_2O + 0.25HS^-$
8	Methanogenesis	$H_2 + 0.25CO_3^{2-} + 0.5H^+ \rightarrow 0.75H_2O + 0.25CH_4$

Both fermentation and respiration process rates are modelled using the same empirical Monod kinetics as formulated in the previous models (Mayer et al. 2001, Watson et al. 2003b). Equation (3.1) shows the general form of the rate expression.

$$r_S = \frac{dS}{dt} = -k_{\max} \cdot \frac{S}{(K_{M\_S} + S)} \cdot \frac{TEA}{(K_{M\_TEA} + TEA)} \cdot \frac{K_I}{(K_I + I)} \qquad (3.1)$$

Where r_S is the rate of consumption of substrate S, [mol/L/s]; S is the substrate concentration [mol/L]; $k_{\max}$ is the maximum substrate utilisation rate [mol/L/s]; K_{M_S} is the Monod half saturation constant with respect to substrate S [mol/L]; TEA is the concentration of the

particular TEA [mol/l]; K_{M_TEA} is the Monod half saturation constant with respect to the TEA [mol/l]; K_I is the inhibition constant [mol/l]; I is the concentration of an inhibiting substance [mol/l], which could be the substrate itself, a competing TEA, or a toxic species. Multiple Monod and inhibition terms can be included in equation (3.1).

Microbial biodegradation activity is intrinsically included in the k_{max} term and growth is not explicitly modelled in this formulation. The reason for this is that unpublished MLS data shows that solute concentrations are pseudo steady-state, i.e. showing a small variation around a mean value, over four years of sampling, so it is assumed that the microbial population activity is also steady-state in the two boreholes during this period. However, microbial activity might be expected to vary over the ~50 year lifetime of the plume.

Thus, there may be spatial and temporal variation in microbial activity due to other factors such as gradual acclimatisation to the high phenol concentrations during the ~50 year history of the plume. Since the data is limited to two vertical profiles determined over a four year period this spatial and temporal variability is implemented in the mathematical model in a simple way depending on the position within the plume at any given time.

$$X_{\text{Plume back}} < (\text{Velocity}) \times (\text{Time} - \text{Time}_{\text{change}}) <= X_{\text{Plume front}} \quad (3.2)$$

Where X is the location within the plume measured from the source along the plume travel direction. The plume front represents the first portion of the plume flowing into uncontaminated aquifer, while the plume back is the subsequent part of the plume, first entering the aquifer at $\text{Time}_{\text{change}}$. After $\text{Time}_{\text{change}}$, the boundary between plume front and back travels forward at the groundwater flow velocity. The final simulation time was 47 years to allow comparisons with the previous field scale modelling, above. $\text{Time}_{\text{change}}$ was chosen as 25 years, such that at the end of the simulation, 47 years, BH60 lies in the plume front, and BH59 in the plume back. This $\text{Time}_{\text{change}}$ also corresponds to a change in the source term, see below. The only parameter varied according to this simplistic plume location criterion was k_{max}, since this, by itself, allows spatial and temporal variation of microbial activity to be simulated.

The inhibition terms K_I may represent either inhibition of one process by another, through microbial competition (Mayer et al. 2001, Watson et al. 2003b) or toxic effects due to a toxic substance, e.g. phenol (Mayer et al. 2001). Evidence from the field site suggests that the high concentration of phenols is likely to inhibit microbial activity due to toxic effects on the biodegradation processes (Lerner et al. 2000). However, fermentation reactions appear to be more robust than respiration reactions in the presence of phenol (Wu 2002). Thus, intermediate species, hydrogen and acetate, accumulate to higher concentrations in the groundwater than are generally observed in plumes from other contaminated aquifers (Thornton et al. 2001b, Watson et al. 2003b). Therefore, the model is formulated to allow different K_I values for respiration and fermentation processes producing and consuming hydrogen. Respiration of acetate is assumed

in the model to be absent due to toxic inhibition. Table 3.2 summarises all the Monod and inhibition parameters used in the model.

Table 3.2 Rate determining parameters for simulation of kinetic redox reactions, K_M and K_I parameters as in eq. (3.1)

			Half saturation constants		Inhibition terms		
	Substrate, *S*	*TEA*	K_{M_S} [mol/l]	K_{M_TEA} [mol/l]	$K_I O_2$ [mol/l]	$K_I NO_3$ [mol/l]	$K_I C_6H_6O$ [mol/l]
1	C_6H_6O	O_2	1.1×10^{-4}[a]	3.1×10^{-6}[a]	-	-	-
2	C_6H_6O	NO_3^-	1.1×10^{-4}[a]	8.1×10^{-6}[a]	6.2×10^{-6}[a]	-	-
3	C_6H_6O	-	1.1×10^{-4}[b]	-	3.1×10^{-5}[a]	1.6×10^{-5}[a]	6.0×10^{-2}[c]
4	C_6H_6O	-	1.1×10^{-4}[b]	-	3.1×10^{-5}[a]	1.6×10^{-5}[a]	6.0×10^{-2}[c]
5	H_2	$Mn(IV)_{(s)}$	5.0×10^{-7}[b]	-	3.1×10^{-5}[a]	1.6×10^{-5}[a]	4.0×10^{-2}[c]
6	H_2	$Fe(III)_{(s)}$	5.0×10^{-7}[b]	-	3.1×10^{-5}[a]	1.6×10^{-5}[a]	4.0×10^{-2}[c]
7	H_2	SO_4^{2-}	1.0×10^{-6}[b]	1.6×10^{-3}[d]	3.1×10^{-5}[a]	1.6×10^{-5}[a]	4.0×10^{-2}[c]
8	H_2	TIC	5.0×10^{-6}[b]	-	3.1×10^{-5}[a]	1.6×10^{-5}[a]	4.0×10^{-2}[c]

Notes: reaction numbers as in Table 3.1 a – same as in Mayer *et al* (Mayer et al. 2001) b – same as in Watson *et al* (Watson et al. 2003b) c – calibrated, value in (Mayer et al. 2001) of $K_I\ C_6H_6O = 6.4\times10^{-3}$. d – calibrated, value in (Mayer et al. 2001) of $K_{M\_SO4} = 1.6\times10^{-3}$

Table 3.3 Background and source concentrations

Mobile aqueous species concentrations in the line source as mol/l_H_2O			
Species	Background (mol/l)	Source, 0 to 25 years Z range[a] (m): C range[b] (mol/l)	Source, 25 to 47 years Z range[a] (m): C range[b] (mol/l)
C_6H_6O	3.4×10^{-10}	10→20m: 3.4×10^{-10}→1.11×10^{-1}	10→19m: 3.4×10^{-10}→8.3×10^{-2}
		20→30m: 4.8×10^{-2}→3.4×10^{-10}	19→30m: 3.2×10^{-2}→3.4×10^{-10}
SO_4^{2-}	6.7×10^{-4}	10→24m: 2.0×10^{-3}→7.0×10^{-3}	10→26m: 1.5×10^{-3}
		24→30m: 7.0×10^{-3}	26→30m: 2.5×10^{-3}→6.7×10^{-4}
TIC	2.3×10^{-3}	10→30m: 1.3×10^{-2}→2.1×10^{-2}	10→30m: 1.1×10^{-2}→3.6×10^{-2}
pH	6.9	10→30m: 5.5→5.1	10→30m: 6.25

Other mobile aqueous species	
Species	Background (mol/l)
O_2	2.9×10^{-4}
NO_3^-	1.7×10^{-3}
H_2	1.0×10^{-10}
CH_3COOH	1.0×10^{-6}
Fe^{2+}aq	8.9×10^{-7}
Mn^{2+}aq	1.2×10^{-7}
HS-	3.0×10^{-10}
CH_4	6.2×10^{-10}
OH-	Equil
$CO3^{-2}$	Equil
H_2CO_3	Equil

Immobile species	
Minerals	mol/kg_solid
FeOOH	9.8×10^{-2}
MnO_2	8.7×10^{-4}
FeS	0, Equil
Surface species	**mol/l_H_2O**
TOTXFeOH	1.39×10^{-2}[c]
$XFeOH_2$+	Equil
XFeO-	Equil
XFeOFe+	Equil
XFeOMn+	Equil
XfeOFeOH	Equil

Notes : a – Z range represents height in domain (0-40m), not depth. The dirichlet type inflow boundary is applied along the specified segments of the inflow boundary. Unspecified Z ranges use the background concentration.

b – C range specifies the concentration values at the ends of each boundary segment, with intermediate points being linearly interpolated.

c - surface adsorption capacity same as used in microcosm modelling (Watson et al. 2003b).

Equil – Equilibrium species whose concentration is computed by the program. The thermodynamic equilibrium constants used are the same as in the microcosm work (Watson et al. 2003b).

TIC - total inorganic carbon, equivalent to TOTAL(HCO_3^-) or sum of CO_3^{-2},H_2CO_3 HCO_3^-,
TOTXFeOH - represents surface complexation capacity i.e. sum of all XFeO...type species.

3.2.6 Equilibrium chemistry.

Several inorganic reactions were included as thermodynamic equilibria (see Table 3.3). Aqueous complexation considered pH, CO_3^{2-}, H_2CO_3, HCO_3^-, OH^- with activity corrections for changes in groundwater composition calculated using the Davies equation (Stumm and Morgan 1996). Precipitation and dissolution of minerals included $FeS_{(s)}$. As the microcosm model suggested that surface complexation may account for a significant proportion of the Fe(II) produced, the same surface complexation reactions and adsorption capacity were applied (Watson et al. 2003b). In order to estimate the relevance of these surface reactions, the field case was simulated both with and without them.

3.2.7 Source zone.

The source zone is simulated using Dirichlet boundary conditions at the inflow boundary. The only species entering the model domain with different concentrations from the background water are phenol, SO_4^{2-}, pH and TIC (see Figure 3.1). Table 3.3 details the spatial and temporal distribution of source concentrations used as well as the background, i.e. initial, concentrations for all species. Mayer *et al* (2001) indicated that the TIC added is due to phenol degradation in the unsaturated zone which is much quicker than in the saturated zone. In that model the boundary concentrations were changed after 25 years since the MLS data suggests that the source term had changed, and this feature is retained in the present model. The 2D source is a 20m thick vertical line source which, for each source species, is split into segments which approximately describe the concentration gradients observed in the vertical profiles at the boreholes. This method will give a closer approximation to the integrated plume mass balance than was achieved formerly by using two 1D flowlines (Mayer et al. 2001).

Modelling was also performed in 3D, partly with the aim of investigating the effect of lateral variability in source zone phenol concentration. Only half the width of the plume was simulated for computational reasons, so the source plane was 20m thick and 65m wide. The same vertical profiles used in 2D were applied across the width of the source plane. The real transverse variability in the source term cannot be estimated as detailed transverse borehole data is currently unavailable. For this reason, an example stochastic realisation of the plume source plane was created as follows. At the plume centreline the 2D profile was unaltered. From 6.5m to 65m from the centreline the 2D profile was used as the mean value for a superimposed stochastic spatial field. The Gaussian stochastic source term was only applied to phenol and the parameters are given in Table 3.4. The variance was chosen to give approximately one order of magnitude range around the mean concentration. Former work using Monte Carlo techniques

suggested that typical source term size might be 6-7m vertically (Huang 2001). Therefore, a pragmatic choice of 6.5m horizontal correlation length was made.

Table 3.4 Physical parameters used for reactive transport simulations

Parameter	Value
Effective porosity (-)	0.125[a]
Hydraulic conductivity, K_x (m/s)	8.4×10^{-6} [a]
Hydraulic gradient (-)	4.2×10^{-3} [a]
Longitudinal dispersivity (m)	1.0[a]
Vertical transverse dispervity (m)	4.0×10^{-4} [a]
Horizontal transverse dispervity (m) (3D only)	1.0×10^{-3} [a]
Final solution time (year)	47[a]
Max time step size (year)	0.05[b]
Domain size, length X (m)	750[a]
Domain size, height Z (m)	40[a]
Domain size, width Y (m) (3D only)	90[a]
Max elements in X	256[c]
Max elements in Z	64[c]
Max elements in Y (3D only)	64[c]
Parameters for heterogeneous anisotropic 2D flow field	
Correlation length in x (m)	10
Correlation length in z (m)	1
Variance applied to K_x	0.1
Anisotropy K_x / K_z	100
Parameters for stochastic 3D source of C_6H_6O	
Correlation length in y (m)	6.5
Correlation length in z (m)	1
Variance applied to C_6H_6O source	0.1

Notes: a – same as in Mayer *et al* (Mayer et al. 2001)
b - adaptive time stepping used up to this maximum dt
c - local adaptive remeshing was applied up to the most refined multigrid level 6 which has 2^6 =64 times more elements than level 0.

3.2.8 Flow field.

The previous field-scale model used a simple uniform flow field aligned to the model grid. Since the plume is plunging (Figure 3.1), the model domain was inclined parallel to the plume and no-flow boundaries were used at the top and bottom of the domain (Mayer et al. 2001). This strategy has been adopted here also. Recharge is not explicitly included, nor is a well pumping at 80m depth 2km to the west, but these are assumed to be the main reason for the dipping flow field. While plume density effects may also contribute to the plunge, these are not considered in this study. There have been no significant pumping wells in the vicinity and it is assumed the flow field has been steady throughout the 47 year simulation period. The parameters for physical flow and transport are given in Table 3.4.

Models were run with the uniform flow field in 2D and 3D. In order to test the effect of a variable flow field on the reactive system, an example anisotropic heterogeneous 2D

permeability field was created. Hydraulic conductivity data from the consolidated Triassic sandstone aquifer at the Four Ashes site shows a range of 0.4 to 1m/day (Mayer et al. 2001). Thus, for this example, stochastic parameters (see Table 3.4) were chosen to give only a slight degree of heterogeneity, with a permeability range of about one order of magnitude around the mean. The resulting flow field has a horizontal velocity component, q_x which is 20 or more times larger than the vertical component q_z.

3.2.9 Modelling procedure.

In order to obtain a good estimate of the plume mass balance it was necessary to calibrate the model to the measured concentrations of the phenol, the intermediate species and the TEAP reactants and products at the MLS locations. The source term and rate parameters were calibrated manually using the 2D uniform flow model. The calibration was completed against the profiles from BH59 and BH60 (Figure 3.1), sampled over a four year period. However, only the sampling data published in 1998 (Thornton et al. 2001b) is reproduced here. A sensitivity analysis performed on the parameters in the 2D uniform flow model is presented elsewhere (Watson et al. 2004a). The calibrated model was used as a base for the example 2D variable flow field and the 3D variable source models, in order to test the effect of these factors on the plume mass balance. Non-reactive models were also run for comparison.

The aim of the 3D modelling was to quantify the plume mass balance in the most realistic way possible, given the lack of transverse borehole data. Modelling the third dimension allows the effects of transverse horizontal dispersion, and consequent reactions at the fringe, to be evaluated. In some cases this is critical to the understanding of the plume shape and mass balance (e.g. Molson et al. 2002). In this study, enhanced computational power and especially the use of parallel processing (Watson et al. 2004a) allows the inclusion of a realistic set of kinetic reactions representing both plume core and fringe in the 3D model. In contrast, the former work was limited by the high computational demands of 3D simulations such that only one reaction, denitrification, and three species (Phenol, NO_3^-, TIC) were included (Mayer et al. 2001). However, some simplification was still required in the present study to make a computationally manageable 3D model. In the 2D model the secondary, equilibrium reactions allow the concentrations of species to be interpreted, but it is only the primary, kinetic reactions which affect the global mass balance of the plume. For example, the global mass balance of Fe(II) produced by primary reactions is unaffected by the secondary reactions which distribute Fe(II) between dissolved Fe^{2+}, precipitated FeS, or surface complexation species such as $XFeOFe^+$. Therefore, the 3D model uses 11 species to simulate exactly the same 8 kinetic biodegradation reactions (Table 3.1) as in the 2D case. Thus, the 3D plume mass balance can be compared directly with the 2D mass balance.

3.2.10 Numerical code description.

The numerical simulator used for this study was UG (Unstructured Grids) (Bastian et al. 1997a) which is a general partial differential equation solver. An existing UG groundwater flow and transport finite volume application (Bastian and Lang 2002) was modified to include general chemical reactions typically used when simulating biodegrading plumes. These processes included Monod kinetics, NAPL dissolution, aqueous complexation, mineral precipitation and dissolution, and ion exchange. The non-linear multi-component reactive transport system is solved with the globally implicit method (Yeh and Tripathi 1989, Saaltink et al. 2000, Mayer et al. 2002) using the Newton-Raphson method. The linearised system is solved with the multigrid solver using a bi-conjugate gradient stabilised linear solver with Gauss-Seidel smoothing. Thermodynamic equilibrium is solved simultaneously with the transport equations forming a differential algebraic equation (DAE) system (Yeh and Tripathi 1989). A number of test cases have verified the new capability and are reported in detail separately (Watson et al. 2004a). Here, the presented simulations used first-order upwind vertex centred finite volume spatial discretisation, with implicit time stepping. The code is capable of local adaptive remeshing and parallel processing and these numerical methods were utilised in the simulations presented. The efficiency gains provided by these methods made tackling the large 2D and 3D multi-species problems more practicable and thus contributed to the understanding gained in this work. The application of these numerical methods to the simulations is presented separately (Watson et al. 2004a).

3.3 Results and Discussion

The biogeochemical processes previously found to be important in understanding the microcosm experiments (Watson et al. 2003b) were found to be important in simulating the field-scale system. These processes included fermentation and respiration reactions, temporal variation of biomass activity, sorption of microbially produced Fe(II) species, and consideration of the bioavailability of metal oxides. In addition, phenol toxicity towards fermentation and respiration, a variable flow field, and a variable source term all affect the results.

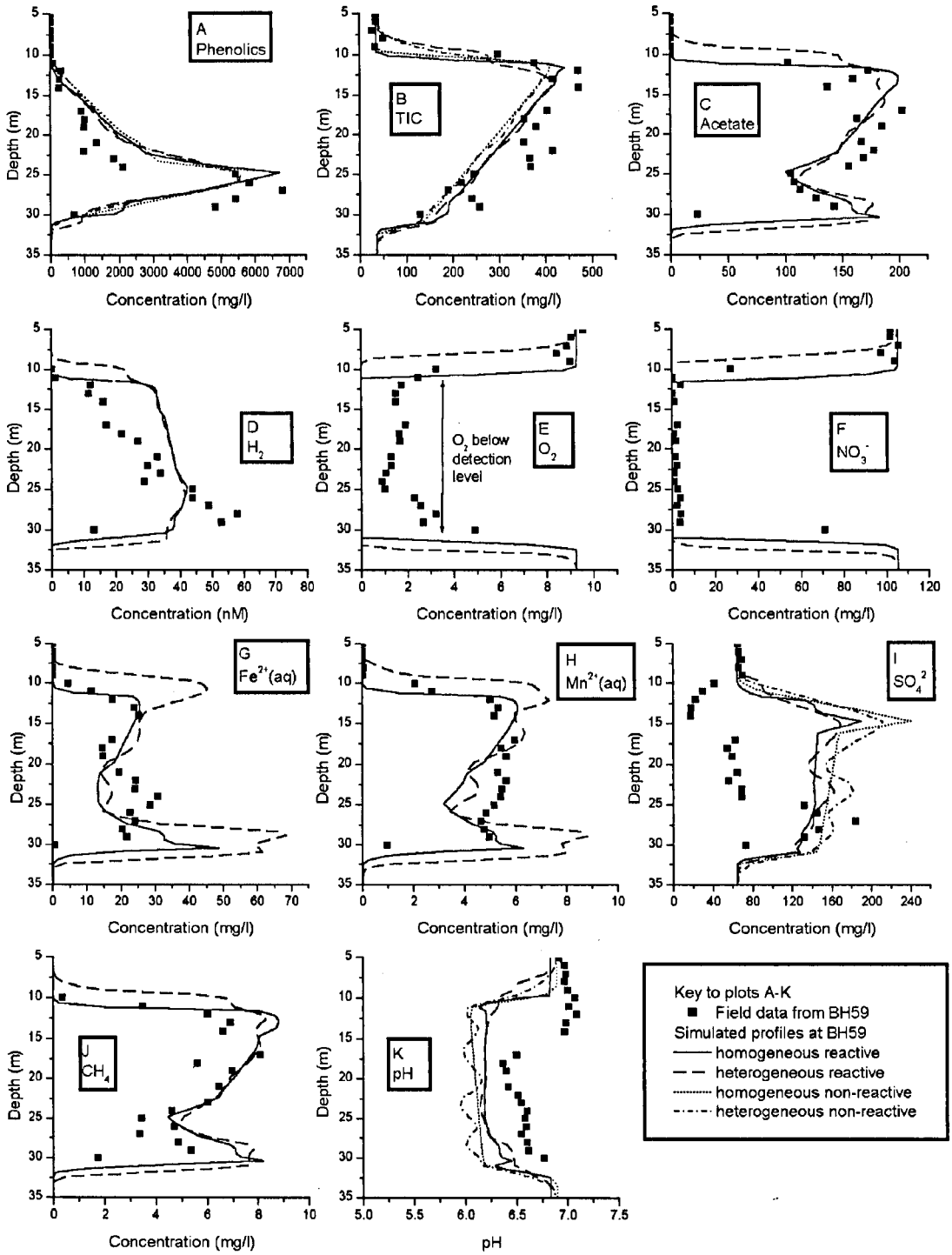

Figure 3.2. BH59 concentration profiles showing one set of observed data, and simulation results from homogeneous and heterogeneous flow models with both reactive and non-reactive cases after 47 years.

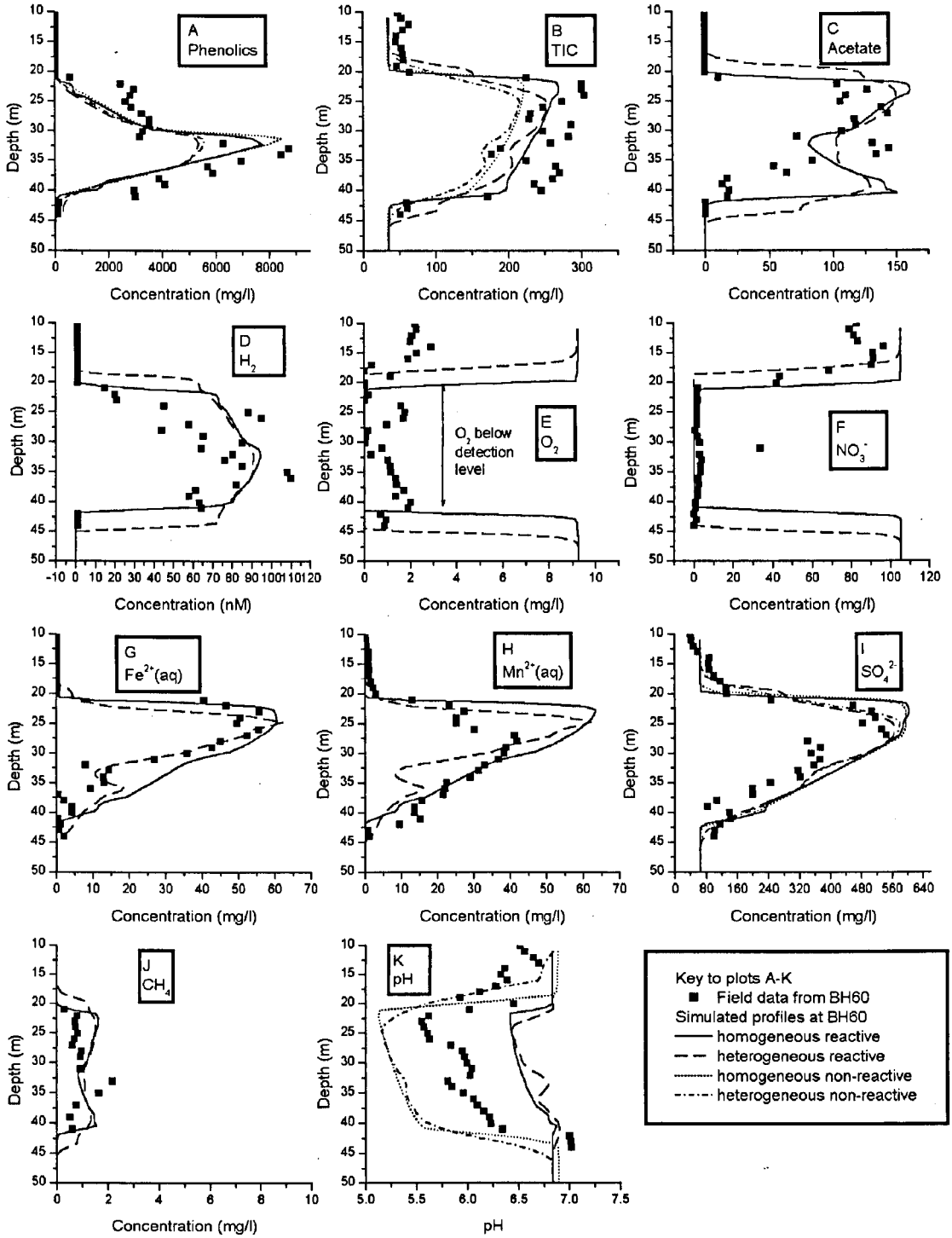

Figure 3.3. BH60 concentration profiles showing one set of observed data, and simulation results from homogeneous and heterogeneous flow models with both reactive and non-reactive cases after 47 years.

3.3.1 2D results overview

Figure 3.2 and Figure 3.3 show the concentration profiles of the 2D simulations compared with the published, observed data for 11 analytes at both MLS locations. The simulations are generally in good agreement with the observed field data, demonstrating that the conceptual model of syntrophic biodegradation developed at laboratory scale (Watson et al. 2003b) can describe these processes at field scale. A comparison of the reactive and non-reactive simulations is plotted for the source term species (phenol, TIC, pH, SO_4^{2-}), but not for other species which remained at background concentrations in the non-reactive case. The similarity between reactive and non-reactive profiles for phenol demonstrates that only a small proportion of the phenol entering the saturated zone is degraded by biodegradation, which is supported by other studies of the plume (Mayer et al. 2001, Thornton et al. 2001a, Thornton et al. 2001b). Concentrations of O_2, NO_3^-, SO_4^{2-} decrease and Fe^{2+}, Mn^{2+} and CH_4 increase in the plume compared with the uncontaminated aquifer, demonstrating the activity of six different TEAPs in the plume. H_2, TIC and acetate concentrations increase within the plume, which further supports the conceptual model of syntrophic biodegradation occuring at the plume scale. The relatively high observed H_2 concentrations are reproduced, thus supporting the use of a fully kinetic description of fermentation and respiration processes. The highest H_2 concentration occurs in the plume core where the phenol concentration is highest. This trend has been reproduced by using inhibition terms, c.f. (3.1), representing phenol toxicity to fermentation and respiration processes and supports the use of the Monod kinetics mathematical formulation of the conceptual model.

Homogeneous and heterogeneous flow-field simulations are plotted and show broadly the same results except for some oscillations and a slightly broader, more dispersed plume in the heterogeneous flow-field. This shows that the principal control on the distribution of plume analytes is likely to be the spatially and temporally variable boundary condition, representing the variable source zone, and the biogeochemical reactions, rather than the spatial variability of the flow-field within the plume. The fringe TEAs O_2 and NO_3^- are reproduced with a sharp gradient. O_2 is difficult to measure in the field with probes, due to interference from the contaminant matrix, and has previously been assumed to be zero inside the plume according to detailed experimental results (Thornton et al. 2001b). The fringe species show that the plume is wider in the case of a heterogeneous flow field, which increases the apparent transverse dispersion.

The dissolved metal species, Fe^{2+} and Mn^{2+}, are reproduced using spatially variable degradation rates and surface complexation reactions. These TEAPs required higher rates in the front part of the plume, Table 3.5, interpreted to be due to initially higher bioavailability of mineral oxidants which decreases with time. The plume mass balance, discussed later, shows that iron reduction

is the most important TEAP, suggesting that it is necessary to understand the processes controlling Fe^{2+} concentration, including oxidant bioavailability and sorption processes, to correctly interpret the Fe dynamics in the system. The SO_4^{2-} and CH_4 are also simulated with spatially variable reaction rates with higher rates at the back of the plume, Table 3.5, interpreted to be due to acclimatisation of microorganisms over long time periods. The SO_4^{2-} in BH59 was observed to be more variable between sampling rounds than other analytes, and since it was calibrated to the full dataset there are apparent differences between the calibrated concentration profiles and the former published data. The pH is generally within half a pH unit, and is largely controlled by the source term at BH59 and by surface complexation reactions at BH60. These results suggest that mineral phase oxidant bioavailability, microbial acclimatisation and surface complexation processes, which are general features of the aquifer geochemical environment, are potentially important in assessing natural attenuation sites in general.

Table 3.5 Rate parameters, k_{max}, for this and other studies

	Values of k_{max} used in this paper				k_{max} in other modelling studies based on Four Ashes site	
	Substrate, *S*	Plume front [mol_*S*/l/s]	Plume back [mol_*S*/l/s]	Ratio Back/front	Field scale Mayer *et al* model (Mayer et al. 2001) [mol_*S*/l/s]	Laboratory scale Watson *et al* model (Watson et al. 2003b) [mol_*S*/l/s]
1	C_6H_6O	4.0×10^{-10a}	4.0×10^{-10a}	1	4.0×10^{-10}	-
2	C_6H_6O	4.0×10^{-11a}	4.0×10^{-11a}	1	4.0×10^{-11}	-
3	C_6H_6O	8.4×10^{-13b}	2.7×10^{-12b}	3.2	-	3.8×10^{-11c}
4	C_6H_6O	2.1×10^{-13b}	2.5×10^{-13b}	1.2	-	-
5	H_2	9.8×10^{-12b}	4.2×10^{-12b}	0.43	1.2×10^{-11d}	1.1×10^{-8}
6	H_2	2.2×10^{-11b}	1.4×10^{-11b}	0.64	6.8×10^{-12d}	5.0×10^{-8}
7	H_2	8.0×10^{-12b}	1.6×10^{-10b}	20	5.0×10^{-11d}	9.0×10^{-8}
8	H_2	2.8×10^{-11b}	8.4×10^{-10b}	30	4.0×10^{-10d}	1.0×10^{-7}

Notes: reaction numbers as in Table 3.1

a – same as in Mayer *et al* (Mayer et al. 2001)

b - calibrated

c - calculated from microcosm parameters, k_{max} multiplied by an assumed fermenter biomass of 1×10^{-3}mol/l (Watson et al. 2003b)

d – calculated by converting stated k_{max} in terms of phenol degradation to H_2 degradation and correcting for different Monod and inhibition terms using typical values: H_2 1×10^{-7}mol/l, phenol 5×10^{-2} mol/l.

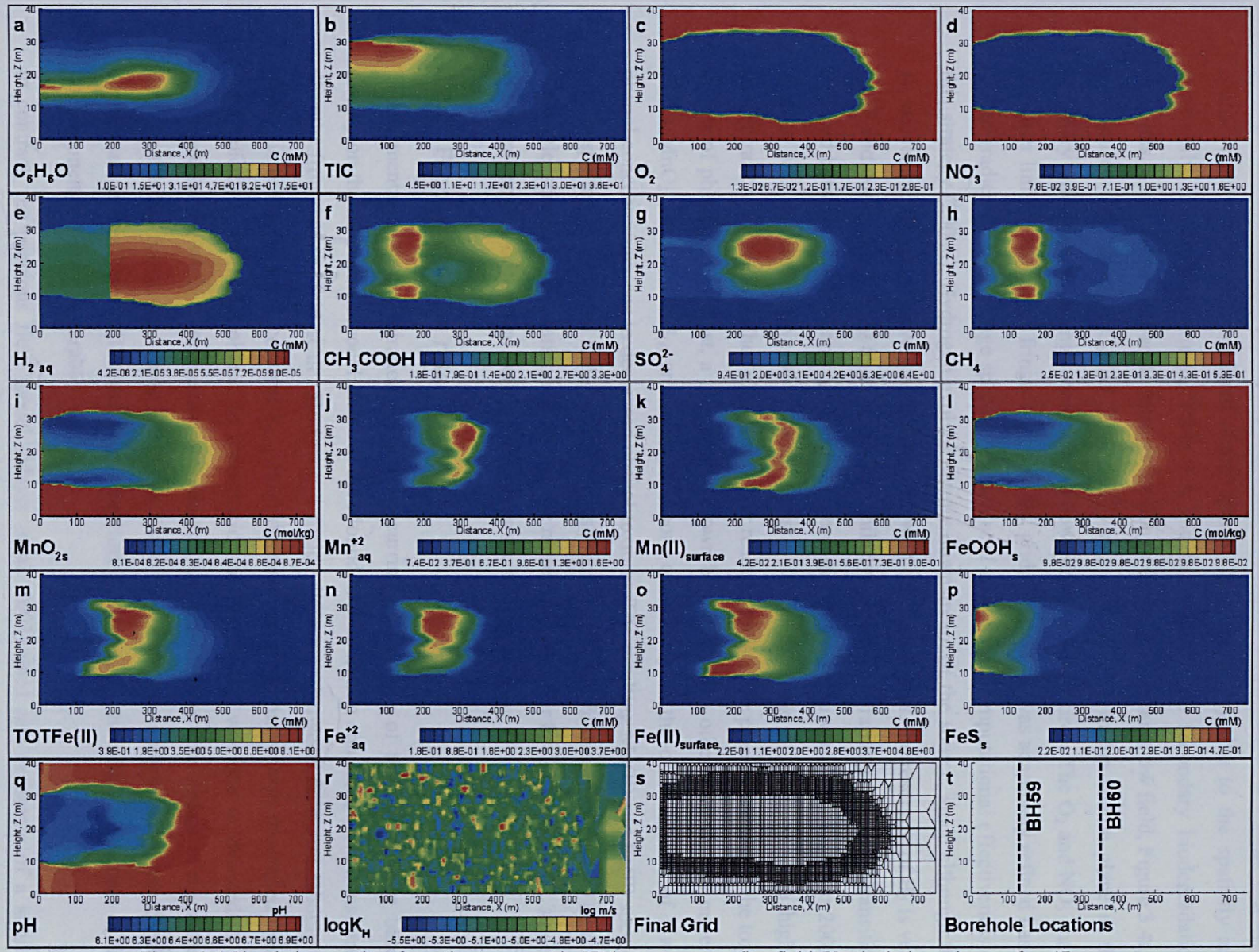

Figure 3.4 Simulation results of reactive transport in 2D heterogeneous flowfield. Vertical plane shown after 47 years

Figure 3.4 shows the contour plots for simulated reactive species in the heterogeneous flow-field, visualising the plume in a vertical plane. It illustrates plume development using the conceptual model derived principally from the microcosm model. There is a large degree of spatial variability in the simulated plume species. This is largely due to the spatially and temporally varying source term, Table 3.3, and the primary and secondary biodegradation reactions, while the smaller-scale features are due to the heterogeneous flow field, Figure 3.4r,s. The source term species, C_6H_6O, TIC, SO_4^{2-}, pH, are anchored to the source plane at the boundary while the remaining species evolve downstream of the source. The O_2 and NO_3^- mark the outer limit of the plume fringe. Note that the grid, Figure 3.4t, was adaptively refined based on the gradient of these fringe species; Watson *et al* showed the computational effectiveness of this strategy (2004a). Hydrogen evolves rapidly downstream of the source, since although the H_2 concentrations are relatively high compared with other field sites, the reservoir of H_2 is very small and does not take long to accumulate. Similarly, H_2 responds rapidly to the rate transition imposed between the front and back of the plume, giving the sharp vertical interface at 200m. Acetate and methane evolve with higher rates at the back of the plume leading to higher concentrations, Table 3.5. They also show lower reaction rates in the plume core where the toxic effects of phenol are greatest, a feature also shown by many of the other species. The mineral TEAs $MnO_{2(s)}$ and $FeOOH_{(s)}$ are consumed more extensively in the outer regions of lower phenol concentration, due to decreased phenol toxicity. The resulting Mn(II) and Fe(II) species are transported, precipitated and sorbed. $FeS_{(s)}$ is precipitated close to the source zone. The fast, reversible, surface complexation reactions cause more retardation for Fe^{2+} than for Mn^{2+}, since a greater proportion of the total Fe(II) is present as surface species.

3.3.2 Biodegradation rates

The fermentation and respiration reactions occurring in the core of the plume were based closely on the microcosm model. Most of the Monod half saturation constants were transferred directly from lab to field scale. However, a major difference between field and laboratory studies is seen in the rate constant k_{max}, since degradation in the microcosm proceeded much more quickly than in the plume. In fermentation of phenol k_{max} was one to two orders of magnitude larger, while in respiration of hydrogen k_{max} was three or four orders of magnitude larger in the microcosm (see Table 3.5). Thus, while it appears to be appropriate to transfer the conceptual and mathematical model from laboratory to field scale the actual magnitude of the rate parameters is not transferable.

In contrast, the previous field model (Mayer et al. 2001) used rate constants of a similar magnitude to this study. The main difference is that the previous study applied one value of k_{max} throughout the plume while this study applies separate parameters at the front and back of the

plume (see Table 3.5). This was implemented because the microcosm study suggested that microbially controlled rates could change with time. The calibrated rates of phenol fermentation and respiration of SO_4^{2-} and CO_2 (reactions 3, 4, 7, and 8, Table 3.5) are up to 30 times faster at the back than the front of the plume. This is interpreted to represent higher microbial activity in parts of the aquifer that have been exposed to the high concentrations of phenol in the plume for a much longer amount of time, i.e. acclimatisation. The acclimatisation on the time scale of decades could not be assessed through modelling the two year data set available from the microcosm study. However, the idea is supported by microbiological studies based on the two MLS locations, which show slightly elevated microbial activity at BH59, the back of the plume (Pickup et al. 2001). At this field site the acclimatisation may be enhanced by the observed lowering of phenol concentrations in the back of the plume at BH59. The phenol inhibition term included in the plume core reactions (see reactions 3-8, Table 3.2) decreases the reaction rate with increasing phenol concentrations. By itself, the inhibition term gave a ratio of rates at the back and the front of less than 1.5 which did not reproduce the observed data. This suggests that the long term acclimatisation processes are more complex than the inhibiting, or toxic, effects which the standard empirical inhibition kinetics describe. The calibrated rates suggest that acclimatisation makes a significant difference as the plume evolves, showing that maximum rates, increasing with exposure time, have to be applied rather than uniform rates.

The calibrated k_{max} values for iron and manganese reduction are higher at the front than the back (Table 3.5). In the microcosm modelling, it was shown that the mineral oxides could exist in more and less bioavailable forms, and this is in general supported by findings from other workers (Roden and Zachara 1996, Christensen et al. 2000). Applying this concept to a plume, the most bioavailable mineral fractions would be expected to be consumed first at the advancing front of the plume and also to be consumed at a higher rate than less bioavailable fractions. This expected pattern is achieved through the particular kinetic coefficients used (Table 3.5). It is suggested that the population of mineral oxide reducing microbes acclimatise with time, as observed with the other plume core reactions, yet overall the bioavailability is the dominant factor for these TEAPs. The microcosm concept of mineral oxidant bioavailability being an important control on the plume reactions is, therefore, supported by the calibrated model of the plume.

3.3.3 Hydrogen and toxicity.

Hydrogen plays an important role in syntrophic biodegradation (Christensen et al. 2000). Watson *et al* (2003b) discussed this at length in relation to the microcosm modelling. The partial equilibrium approach (McNab and Narasimhan 1994, Jakobsen and Postma 1999, Brun and Engesgaard 2002), which makes use of the hydrogen concentrations, was previously applied to the Four Ashes plume (Thornton et al. 2001b). That work concluded that threshold energy levels for TEAPs were lowered, probably due to toxic effects from the phenol, resulting

in elevated hydrogen concentrations compared with other documented sites. This suggested that the respiration processes were affected more by the toxicity than the fermentation processes (Thornton et al. 2001b). Laboratory work has also indicated that fermentation is more robust than respiration in the presence of high phenol concentrations (Wu 2002).

In the present fully kinetic model, hydrogen forms a central part of the reaction scheme (Figure 3.1), and its concentration is determined by the relative rates of fermentation and respiration, as was the case with the microcosm (Watson et al. 2003b). In order to account for the toxic effects of phenol, the fermentation reactions have a higher phenol inhibition constant than the respiration reactions (compare K_I values for reactions 3 and 4 with 5-8 in Table 3.2), implying that fermentation is less affected by high phenol concentrations than respiration. This results in hydrogen profiles which follow the trend of the phenol i.e. higher phenol gives higher hydrogen, as illustrated in Figure 3.2a, d, and Figure 3.3a, d. Thus, the trend of the hydrogen profiles can be simulated using this simple conceptual treatment, and this adds support to the idea that fermentation is more robust than respiration.

The fact that the reaction system depends strongly on phenol concentration is reflected in many of the other species results (see Figure 3.4). For example, the contour plot of acetate (Figure 3.4f) shows that acetate production has been suppressed in the centre of the plume where the phenol concentrations are highest (Figure 3.4a). Similar patterns exist for TEAP products Fe^{2+}, Mn^{2+}, and CH_4 (Figure 3.4h, j, m). Profiles which show observed data that appear to support this concept include acetate and CH_4 in BH59 (Figure 3.2c, j).

The contour plot for hydrogen (Figure 3.4e) has a clear division between the front and back of the plume, see equation (2), which is due to the separate rates applied at the front and back (see Table 3.5) in order to calibrate the two boreholes. Note that the interface has moved forward with the velocity of the plume and is shown at its final position at 47 years. The reason that the simulated interface stays sharp, and has not been affected by dispersion, is that the hydrogen reservoir is very small and it is turned over in a matter of minutes (Lovley and Goodwin 1988). This interface is a consequence of the unrealistic sudden change in simulated reaction rates from front to back. A more gradual change of rates would give a potentially more realistic gradual change in hydrogen, but the two MLS profiles do not yield information on the rates in the intervening space so, for simplicity, the simulation used two discrete reacting zones at the front and back of the plume.

The consideration of biodegradation parameters as a function of plume position has aided the understanding of this plume according to acclimatisation and bioavailability processes which are empirically simulated in the lumped k_{max} parameter. These ideas originated with the microcosm model and could form important aspects of conceptual models for other field scale plume studies.

3.3.4 Iron(II) and Manganese(II)

It is well known that the Fe(II) measured in solution as Fe^{2+} does not necessarily represent the full amount of Fe(II) created by microbial iron reduction in a plume, since processes such as precipitation and sorption act to decrease the amount in solution (Christensen et al. 2000). The previous field scale model included $FeS_{(s)}$ precipitation, but not surface complexation. In that model, a ten-fold increase of $FeOOH_{(s)}$ reduction rate was found to give a very high pH (Mayer et al. 2001). Based on experiences with the microcosm, the inclusion of surface complexation reactions permits a higher $FeOOH_{(s)}$ reduction rate, without a large increase in alkalinity due to proton release upon Fe^{2+} adsorption (Watson et al. 2003b).

The simulations showed that when surface complexation was included, the $FeOOH_{(s)}$ reduction rate was four times faster, although the field data could be explained with and without these reactions. This contributes towards $FeOOH_{(s)}$ being the dominant TEAP, and towards the apparent dominance of core degradation processes over fringe degradation processes, discussed in the mass balance, below. The contour plots illustrate the quantity and distribution of Fe(II) species (Figure 3.4m,n,o,p). Over half the total mass of Fe(II) is held on surface sites, most of the rest is Fe^{2+} and a small amount is precipitated $FeS_{(s)}$. This sorption leads to retardation of the Fe^{2+}, and at the back of the plume where the $FeOOH_{(s)}$ reaction rate is lower, as discussed above, desorption contributes about 1/3 of the aqueous Fe^{2+} concentration. The reproduction of the observed Fe^{2+} profiles (Figure 3.2g, Figure 3.3g) is better than was achieved with the former model, lending support to the use of the microcosm conceptual model. The Mn(II) is less strongly affected by surface complexation with less than half the mass being on surface sites, and with less retardation of Mn^{2+} than Fe^{2+}. The simulated pH is slightly high at BH60 (Figure 3.3k), but the contour plot shows that a lower pH zone, originating from the source, is about to enter BH60 from upstream (Figure 3.4q).

The relatively complex behaviour of Fe(II) resulting from the use of the microcosm model is not unexpected, yet this confirms that care must be taken in interpreting field measurements of Fe^{2+}; as that does not represent the full amount of $FeOOH_{(s)}$ reduction. This complexity is partly due to interaction with aquifer solids, which means that Fe^{2+} has a history and does not necessarily represent the in-situ biodegradation processes. On the other hand, hydrogen, discussed above, does represent in-situ processes due to its very rapid turnover. The present model has allowed understanding of the controlling factors affecting both iron(II) and hydrogen and together this has given an improved understanding of processes at field scale.

3.3.5 Heterogeneous flow simulation

The example Gaussian stochastic hydraulic conductivity field for the aquifer model domain is shown in Figure 3.4r. Since only a small degree of heterogeneity was introduced into the model, and reaction parameters were otherwise the same as the homogeneous flow field, the effect on

the simulated concentrations has been small. The profiles (Figure 3.2, Figure 3.3) and contour plots (Figure 3.4) both show fingering due to the spatially variable advective velocity. The contour plot for the homogeneous flow (not shown) does not look significantly different from the heterogeneous flow model (Figure 3.4). The heterogeneous plume is also thicker than the homogeneous flow plume due to the vertical transverse spreading caused by the flow field. The difference in mass balance is discussed below.

3.3.6 Non reactive models

Non-reactive simulation results are plotted for the source species, phenol, TIC, SO_4^{2-}, and pH (see Figure 3.1) as profiles in Figure 3.2 and Figure 3.3. These profiles generally show little difference from the reactive simulations, which show that even though the reactions have occurred for 47 years they have had a small impact on the overall system. The decrease in phenol due to reaction is very small, the increase in TIC is small, and the decrease in SO_4^{2-} is small. Note that only one measured data set is plotted in the profiles, and that SO_4^{2-} was calibrated against additional data that is not shown. Isotopic evidence forSO_4^{2-} reduction was used to estimate the non-reactive source term for SO_4^{2-} at BH59 (Spence et al. 2001), and this data was considered along with the multiple field sampling data (not shown).

3.3.7 3D Results

The results of the 3D model created by extending the source zone of the 2D model laterally and including the horizontal dispersive fringe are presented in the mass balance, below. The variable phenol source zone model is presented in Figure 3.5, which also illustrates the resulting plumes for phenol and acetate as isosurfaces. A small amount of fingering of the plume is apparent in the simulations. This is not caused by the flowfield, which in the 3D simulations was kept uniform, but solely by the spatially variable phenol source.

3.3.8 Mass Balance

Table 3.6 shows aspects of the mass balance simulated in the various models which have been discussed briefly above. The previous field model data is shown for comparison.

Table 3.6 Mass balance comparison between field models.

		2D models			3D models		
		2D with homo-geneous flow field, 25 years	2D with homo-geneous flow field, 47 years	2D with hetero-geneous flow field	3D with deter-ministic source	3D with stochastic source	3D (Mayer et al. 2001)
Total mass degraded (mol phenol)		5.18×10^{2}	1.47×10^{3}	1.66×10^{3}	1.18×10^{5}	1.18×10^{5}	1.30×10^{5}
Total mass degraded per m plume width (mol phenol)		5.18×10^{2}	1.47×10^{3}	1.66×10^{3}	1.82×10^{3} [a]	1.82×10^{3} [a]	1.00×10^{3}[b]
Total mass degraded relative to total added (wt. %)		2.3	4.2	5.1	5.3	5.0	2.2
Balance of carbon in products (% of mol C degraded)	TIC	62.6	41.5	42.6	45.0	45.2	97.3
	Acetate	36.7	56.1	55.1	53.0	52.8	0
	CH_4	0.7	2.4	2.3	2.0	2.0	2.7
Relative contribution of each TEA to total degradation (%)	O_2	6.7	4.1	4.2	5.0	5.0	6.5
	NO_3^-	47.4	28.9	29.9	31.3	31.5	47.5
	MnO_2	12.1	13.7	13.5	13.8	13.7	5.6
	FeOOH	26.7	30.7	30.3	30.8	30.7	4.9
	SO_4^{2-}	3.2	8.2	7.9	6.9	6.9	19.3
	CH_4	3.9	14.4	14.1	12.2	12.2	16.2
Degradation at fringe (%)		54.1	33.0	34.2	36.3	36.5	54.0
Degradation in core (%)		45.9	67.0	65.8	63.7	63.5	46.0

Notes: All figures based on global masses cumulatively reacted in 47 years unless stated otherwise
a - plume width 65m. b - plume width 130m

Most of the mass balances cover the entire simulation period of 47 years, but the 2D homogeneous flow simulation mass balance is also given at 25 years and therefore is only affected by the first set of boundary conditions (Table 3.3) and the rates for the front of the plume (Table 3.5).

The present simulations estimate that 4-5% of the phenol added is degraded, that is roughly double the previous prediction of 2.2% (Mayer et al. 2001). This proportion is much closer when the figures at 25 years are examined, 2.3%, which indicates that the accelerated degradation rates at the back of the plume play a large role in increasing the mass degraded. Although this may work in favour of using a monitored natural attenuation strategy for the site since the degradation rate appears to have increased the mass turn-over rate with time, it is clear that biodegradation of the phenols is extremely slow at this site.

The plume carbon balance shows that over 50% of the degraded carbon is present as acetate. This species was omitted in the previous work, and partly explains the previously lower estimate for the amount of phenol degradation.

The contribution of the electron acceptors to degradation shows that nitrate and $FeOOH_{(s)}$ are the most important in the simulations. In contrast, in the previous work $FeOOH_{(s)}$ was the least important electron acceptor. This difference is primarily due to the inclusion of surface complexation reactions which required a four-fold increase in FeOOH reduction rate, and illustrates clearly the potential importance of surface complexation reactions, or other reactions such as mineral precipitation that remove Fe(II) from solution. These processes are general in aquifers and their potential importance at candidate sites for NA should be considered. The importance of the $FeOOH_{(s)}$ TEAP was also shown in another recent NA modelling study, which found that $FeOOH_{(s)}$ was the most important TEA for xylene degradation under natural gradient conditions (Schäfer 2001). The study also commented on the potential importance of bioavailability of $FeOOH_{(s)}$ at that site. The present work demonstrates the difference that considering bioavailability can make to kinetic rates (see above).

Former work at the Four Ashes site suggested that NA was limited because toxic effects of high phenols concentration limited fermentation and associated TEAPs in the core, and because very low transverse mixing limited the fringe degradation processes (Lerner et al. 2000). To ascertain if it is the fringe or core processes which are most likely to reduce the plume mass, it is useful to examine the mass balance. In the present model degradation at the plume fringe accounts for 35% of the total, with 65% degrading in the plume core (see Table 3.6). This contrasts with the former modelling work which had 54% degrading at the fringe (Mayer et al. 2001). In fact, previous studies of the plume have concluded that the fringe degradation processes are the most important since plume core degradation processes are limited by the toxic effects of phenolics (Lerner et al. 2000, Pickup et al. 2001, Thornton et al. 2001a, Thornton et al. 2001b, Williams et al. 2001). The present study shows that, despite toxicity effects, the plume core is the most important area for phenol mass turnover, and that the fringe is limited by the relatively small dispersive mixing. This finding may have implications for other NA studies in contaminated aquifers, especially those involving phenols, or other potentially toxic contaminants, and limited by dispersive mixing of dissolved TEAs into the plume fringe.

Two other modelling studies of natural attenuation also report the importance of plume core processes with c.a.60% degradation attributed to iron reduction, manganese reduction, sulphate reduction and methanogenesis, with c.a.40% due to aerobic and denitrifying activity (Essaid et al. 1995, Schäfer 2001). In the present study, the increased importance of the core processes is due to the increased fermentation rates, primarily at the back of the plume.

Comparison of the homogenous and heterogeneous flow cases shows some small but predictable differences. As mentioned above, the plume is bigger in the heterogeneous case, implying that both the fringe surface area and the core volume will be bigger. Thus the total mass degraded increases from 4.2% to 5.1% due to the increased reactions in both fringe and core. The fraction of degradation at the fringe increases only by 1% to 34.2% (see Table 3.6).

Note that the reaction model and parameters for the fringe processes were kept the same as the previous work (Mayer et al. 2001). However, it was found in the sensitivity study (see (Watson et al. 2004a) for details) that increasing the reaction rates at the fringe made little difference, rather the limiting factor affecting mass loss at the fringe was dispersion, which was determined independently in another study of the site (Huang 2001). A more heterogeneous flow field than the one used might make the fringe more important still, especially if extensive fingering created significantly larger surface areas without significant core volume increase.

In the 3D cases the effect on overall degradation of using a stochastic source rather than a deterministic one is relatively minor. The mass of phenol degraded is the same in both deterministic and stochastic sources (see Table 3.6). However, the distribution in the source plane means that more phenol mass is input in the stochastic case so the percentage of phenol degraded decreases slightly. The lack of a difference is perhaps surprising given that some fingering is apparent in Figure 3.5. Other trial simulations (not shown) suggested that a greater degree of heterogeneity in the source could lead to more fingering, and more reaction, principally through more reaction at the nose of the plume as fingers develop.

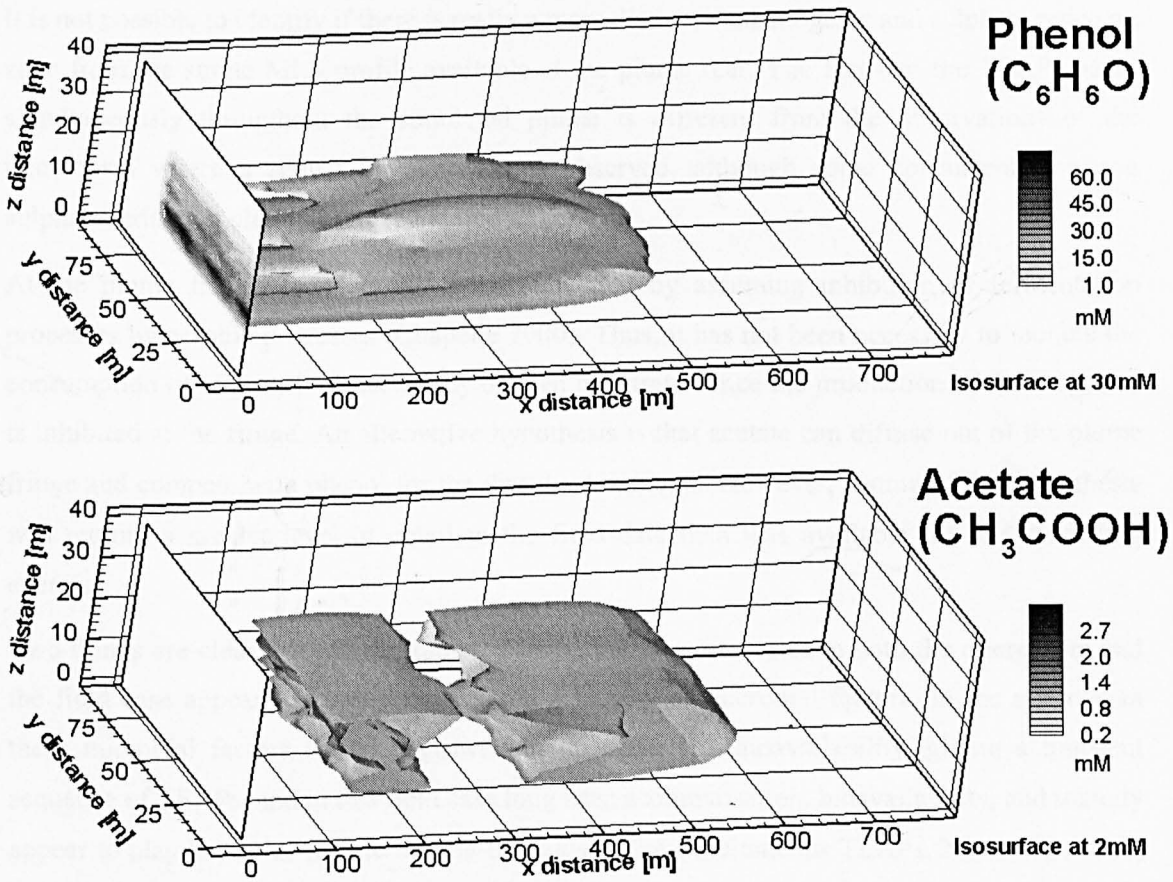

Figure 3.5 Results of 3D model showing contours on variable source plane and resulting fingering plume isosurfaces for phenolics and acetate after 47 years.

Overall, using a heterogeneous flow field or a stochastic source zone alter the mass balance in favour of the fringe processes, but the difference is small compared with the resulting changes to the fermentation rate in the plume core. In the light of this, the conceptual ideas of syntrophic processes, in which rates are constrained by the data, become all the more important.

3.3.9 TEAP patterns

It is worth comparing qualitatively the results with an early conceptual model of the redox zonation of the plume (Williams et al. 2001) based on observations at long screen wells. The redox zones extend from the source in order as CH_4, SO_4^{2-}, FeOOH, MnO_2, NO_3^-, O_2. At the Four Ashes site the redox zones are not exclusive, but are strongly overlapping. Free energy calculations using measured data for the TEAPs also suggest that all the plume core TEAPs are thermodynamically favourable and concomitant TEAPs are observed (Thornton et al. 2001b). The simulation results show that at the back of the plume CH_4 and SO_4^{2-} are the dominant TEAPs, while at the plume front the FeOOH and MnO_2 are dominant (see Table 3.5), and at the fringe O_2 and NO_3^- dominate. In addition, the simulation shows all the plume core TEAPs to be concomitantly active in both MLSs, which is in agreement with the previous conceptual model. It is not possible to identify if there is really a more distinct methanogenic and sulphate reducing zone from the single MLS profile available at the plume rear. The fact that the TEAPs occur simultaneously throughout the simulated plume is different from the observations of the microcosm where a sequential pattern was observed, although some concurrent iron and sulphate reduction clearly occurred at one stage.

At the plume fringe, the data has been simulated by assuming inhibition of fermentation processes by aerobic processes (Chapelle 2000). Thus, it has not been necessary to include the consumption of hydrogen or acetate by oxygen or nitrate, since the production of these species is inhibited at the fringe. An alternative hypothesis is that acetate can diffuse out of the plume fringe and compete with phenol for the dissolved oxidants. However, testing of this hypothesis will require a greater level of detail in the field data than was available for this modelling exercise.

Two things are clear from these studies: Firstly, the observed rates in both the microcosm and the field case appear to be dependent upon a variety of microbial factors. In the microcosm these microbial factors included growth, competition and bioavailability, giving a transient sequence of TEAPs, and in this field case long term acclimatisation, bioavailability, and toxicity appear to play key roles in determining the rates of the simultaneous TEAPs. Note that in both cases sufficient free energy appears to be available but its relative magnitude does not seem to exert a control on the rates. This supports the finding in the microcosm that a fully kinetic model may be a better way to describe microbial process than a partial equilibrium approach which does not model microbial activities in the TEAP step (see (Watson et al. 2003b) for

discussion). Secondly, the Monod kinetic framework used is flexible so that in general TEAPs will occur simultaneously at a given location, but inhibition terms can be added to simulate competition processes and possible exclusion (Schäfer et al. 1998, Mayer et al. 2001, Curtis 2003, Watson et al. 2003b). Thus the Monod kinetic framework was able to increase understanding of the microbial processes mentioned in the microcosm and the field case. These points support the general applicability of the Monod kinetic model, although case-specific modifications may be required if sufficient data is available to constrain the conceptualisation and parameter values.

3.4 Conclusions

Several important ideas resulting from the microcosm work have now been considered at the field scale. Transferring the microcosm model to field scale has resulted in a new fully-kinetic, two-step syntrophic biodegradation model for plumes. This model has generally reproduced observed detailed MLS profiles at a field site, and the pattern of concomitant TEAPs in the plume core. In examining the similarities and differences between the laboratory and field case, it has been important to have a consistent conceptual framework, yet with flexibility to include differing parameter values as required by the different cases, and the fully kinetic model has met these needs. It was found that while the microcosm conceptual reaction model was transferrable to the field scale, the values of the rate parameters were not, since the reactions are much slower in the field.

In both field and laboratory cases, the microbial activities of both fermenters and TEAPs change with time, and space, due to processes such as growth, bioavailability, acclimatisation, and toxic effects. An important result of considering acclimatisation, and consequent increase of rates with exposure time, is that the core reactions turned over more contaminant mass than the fringe reactions, which was not expected from former studies of the plume (Mayer et al. 2001, Thornton et al. 2001a). It appears, in general, that reactive transport models used for NA assessment should consider such temporal and spatial changes.

The importance of the $FeOOH_{(s)}$ TEAP should be considered carefully. Failure to consider surface complexation, or other processes that remove Fe(II) from solution, could result in an underestimation of $FeOOH_{(s)}$ TEAP importance. In this example, considering surface complexation meant that a fourfold increase in the rate of $FeOOH_{(s)}$ reduction was required. In contrast, if the limited bioavailability of $FeOOH_{(s)}$ is not accounted for, then the importance of the $FeOOH_{(s)}$ TEAP may be overestimated. In general, these findings indicate closer attention is needed to this aspect of biodegradation in contaminated systems.

Including additional physical detail in the system, by including a heterogeneous anisotropic flow field and a spatially variable source zone, increased the mass turnover by increasing the plume volume and surface area of the fringe. However, the ratio between fringe degradation and

core degradation changed only slightly, suggesting that the reactive processes exert an important control on mass turnover at this site. The detailed results from considering reactive processes in the laboratory microcosm, were supported at field scale, and suggest these factors should be more generally considered at candidate sites for NA applications.

4 Advantages of using adaptive remeshing in parallel for modelling biodegradation in groundwater

I.A. Watson[1], R.S. Crouch[1], S.E.Oswald[1], P. Bastian[2]

[1]*University of Sheffield, i.watson@sheffield.ac.uk, r.crouch@sheffield.ac.uk, s.oswald@sheffield.ac.uk, Sheffield, UK*

[2]*University of Heidelberg, Peter.Bastian@iwr.uni-heidelberg.de, Heidelberg, Germany*

Abstract

Biodegrading plumes in groundwater are often typified by relatively reactive zones around the fringes and less reactive zones in the core. A high degree of refinement is required at the fringes if a model is to be of use in improving the conceptual understanding of plumes. Two strategies for dealing with the potentially high computational demands are (i) parallel processing, where the workload is shared between multiple processors, and (ii) locally adaptive remeshing, where a refined area of the grid tracks the moving plume fringes through the domain. The partial differential equation toolbox, UG (Unstructured Grids) offers advanced numerical tools including adaptive remeshing, sparse matrix storage schemes, and multigrid solvers. It embraces many of these features within a parallel processing environment. This paper reports on a recent development of UG to simulate field scale reactive biogeochemistry including Monod kinetics, NAPL dissolution, mineral precipitation/dissolution and ion exchange. The non-linear multicomponent reactive transport system is solved with the fully coupled method. Test cases have been used for verification of the new capability. The paper illustrates an application to a 3D field site. It is demonstrated that both adaptive remeshing and sparse matrix storage in parallel processing can improve efficiency and in turn facilitate the incorporation of a more complex set of species and reactions such that understanding of plume processes is enhanced.

4.1 Introduction

Studies on natural attenuation and engineered biodegradation at field sites of groundwater contamination often produce detailed data sets which have the potential to improve understanding of the processes occurring (e.g. Thornton et al. 2001b). Biodegrading plumes in groundwater will in general have significant and distinct reaction processes occurring around the fringe and in the core of the plume (Watson et al. 2004b). The plume fringe may often be

characterised by steep concentration gradients over a narrow region. In order to use numerical modelling to understand processes in such cases a higher level of mesh refinement will be required at the narrow fringe than might be necessary within the broader plume core. The numerical model should ideally include many species covering the effects of primary degradation and secondary inorganic reactions (Hunter et al. 1998). Overall the desire to include sufficient spatial resolution and many species in a field scale model produces potentially very large numerical problems. Two potential strategies for dealing with the high computational demands arising from the need for more highly refined models are (i) parallel processing and (ii) locally adaptive remeshing (Barry et al. 2002). These techniques have rarely been applied to groundwater biodegradation models.

In parallel processing the workload is shared between multiple processors, for example on an array of Linux PCs which can be built into a Beowulf cluster relatively cheaply. The speed up gained by using this form of super computer depends not only on the number of processor nodes available, but the amount of communication of processors with each other in any specific problem. Reactive transport codes have to be specially written for parallel applications. Examples of the use of parallel processing in reactive transport modelling include models which apply operator splitting and solve the transport and chemistry separately, in which case it is relatively easy to parallelise the chemical step (e.g. Schäfer et al. 1998).

Locally adaptive remeshing is a modelling strategy where a refined area of the grid tracks the moving plume fringes through the domain. Thus, the reactive fringe may be simulated more accurately and efficiently, while the core of the plume is simulated on a coarser grid. The algorithm requires some form of specified error indicator, for example related to high concentration gradients, which tells the model where to refine and coarsen the mesh. A more precise analytical error estimator may not be available for general reactive transport problems due to their highly non-linear nature, thus empirical error indicators are used (Wagner et al. 2002). While the approach has been used in a wide variety of computational problems, recent reviews show there has been relatively little reported in the reactive transport literature (Mansell et al. 2002, Kanney et al. 2003). Exceptions are an application to ion exchange (Wolfsberg and Freyberg 1994) and an application to chlorinated solvents biodegradation (Wagner et al. 2002). The use of adaptive remeshing combined with parallel processing has been demonstrated for multiphase flow problems (Bastian et al. 2000), but are just emerging for reactive transport problems (Mansell et al. 2002).

The aims of this study are (i) to develop and verify a flexible groundwater biodegradation code which includes numerical tools to facilitate large scale, chemically complex simulations, (ii) to use the model to assess the usefulness of adaptive remeshing and parallel processing for realistic biodegradation problems.

The numerical simulator used for this study was UG (Unstructured Grids) (Bastian et al. 1997a), a partial differential equation toolbox which offers advanced numerical tools including adaptive remeshing, multigrid solvers, sparse matrix storage, and a variety of tools for different discretisations. It embraces many of these features within a parallel processing environment. An existing UG application for solving subsurface reactive transport in a fully coupled manner formed the basis of this work (Bastian and Lang 2002). The application's capability for chemical reactions was significantly extended to allow a flexible range of processes to be simulated as appropriate for biodegradation studies. Processes included kinetic processes such as organic degradation, biomass growth, non-aqueous phase liquid (NAPL) dissolution, and equilibrium processes including aqueous complexation, mineral dissolution and precipitation, ion exchange and surface complexation. Thus, the code's reaction capability is comparable to state of the art multi-dimensional (3D) reactive transport simulators (e.g. Schäfer et al. 1998, Brun and Engesgaard 2002, Mayer et al. 2002, Prommer et al. 2003). The new code is verified by comparison with typical biodegradation examples from the literature which serve as test cases.

In addition, the new code possesses the advanced numerical features of UG, i.e. adaptive remeshing, sparse matrix storage, and parallel processing, which are generally lacking in published reactive transport codes. A realistic plume modelling study (Watson et al. 2004b) has been used as a basis to assess these advanced numerical tools with regard to efficiency and accuracy, and in the context of multi-species, fully coupled reactive transport modelling. It is demonstrated that these tools can improve efficiency and the relative benefits and limitations are discussed. These techniques will facilitate the incorporation of a more complex set of species and reactions such that understanding of plume processes is enhanced.

4.2 Model Formulation

An existing UG groundwater flow and transport application (Bastian and Lang 2002) was used as the basis for the work. This comprised a multi-component advection dispersion framework with a non-linear source-sink term to accommodate kinetic reactions. The system was solved with a fully coupled approach. This source code, written in C, was modified to include generalised chemical reactions typically used when simulating biodegrading plumes. These included kinetic and equilibrium processes as described by the following governing equations and test cases.

4.2.1 Governing Equations

Several previous works have given exemplified descriptions of how to include kinetic and equilibrium chemistry in a transport model (Yeh and Tripathi 1989, Steefel and MacQuarrie 1996, Saaltink et al. 1998) so this methodology is only briefly summarised here. The governing equations for the coupled reactive transport system are now given. Here, one mobile phase is

considered. The advection dispersion reaction (mass conservation) equation is used in the following general form.

$$\frac{\partial M(C_i)}{\partial t} = L(C_i) + SS_i \tag{4.1}$$

Where C_i is the concentration of the ith species that is being solved for, and takes various units as described below. $M(C_i)$ is the mass storage function that expresses C_i in units of mass/bulk volume [mol/m^3_{bulk}]. t is time [s]. $L(C_i)$ is the advection dispersion differential operator [mol/m^3_{bulk}/s]. SS_i is the source-sink term which is in general a non-linear function of all the species concentrations [mol/m^3_{bulk}/s]. In the case of species in the mobile, aqueous, phase the units of C_i are chosen as [mol/m^3_{mob}], and so $M(C_i)$ and $L(C_i)$ are given by:

$$M(C_i) = n_{mob} C_i \qquad \text{for mobile species} \tag{4.2}$$

$$L(C_i) = \nabla \cdot (D\nabla C_i - qC_i) \qquad \text{for mobile species} \tag{4.3}$$

Where n_{mob} is the volume fraction of the mobile phase, i.e. the effective porosity [m^3_{mob}/m^3_{bulk}]. D is the second order hydrodynamic dispersion tensor [m^2/s]. q is the Darcy flux vector [m^3_{mob}/m^2_{bulk}/s]. In this formulation all the terms in (4.1) have units in terms of bulk volume which facilitates the use of inter-phase kinetic reactions. In the case of all immobile species the transport operator in (4.1) is zero:

$$L(C_i) = 0 \qquad \text{for immobile species} \tag{4.4}$$

Immobile species can have their concentrations, C_i, defined with various units to follow appropriate conventions, but the units of $M(C_i)$ must always be mol/m^3_{bulk}. Some examples of $M(C_i)$ for various types of immobile species follow. For the case of immobile mineral species in the solid matrix phase with units chosen as [mol/kg_{solids}], $M(C_i)$ is:

$$M(C_i) = n_s \rho_s C_i \qquad \text{for mineral species} \tag{4.5}$$

Where n_s is the volume fraction of the solids phase, [m^3_{solids}/m^3_{bulk}] and ρ_s is the density of the solids [kg_{solids}/m^3_{solids}]. In the case of immobile surface complexation species with units chosen as [mol/m^3_{mob}], $M(C_i)$ is:

$$M(C_i) = n_{mob} C_i \qquad \text{for surface complexation species} \tag{4.6}$$

In the case of ion exchange species using the Gaines-Thomas convention (Appelo and Postma 1993) the concentrations are expressed as equivalent fractions [meq per 100g sediment/cation exchange capacity]

$$M(C_i) = \frac{n_s \rho_s C_i}{100 z_i} \qquad \text{for ion exchange species} \tag{4.7}$$

where z_i is the charge of the exchanged cation. Other immobile phases may also be inserted into the flexible model interface. For example, one of the test cases below includes a biophase to simulate mass transfer of nutrients into a biofilm, as well as a non-aqueous phase liquid (NAPL) source (see case B).

4.2.2 Kinetic reactions

The source sink term can include physical sources or sinks of mass, but the discussion here focuses on how chemical reactions contribute to the interspecies mass fluxes. In a system with k kinetic reactions each reaction has a certain calculated rate, R_k, computed in terms of the bulk volume [mol/m$^3_{bulk}$/s]. For any one species SS_i may then be calculated using the rates and the stoichiometric coefficients, s_{ik}, of the species in the reactions. For example if the kth kinetic reaction is:

$$s_{1k}C_1 + s_{2k}C_2 \xrightarrow{R_k} s_{3k}C_3 \tag{4.8}$$

then SS_i for species i is given by:

$$SS_i = \sum_k s_{ik} R_k \tag{4.9}$$

The interface is flexible such that reaction rates, R_k, may be specified as zero order, first order, or some non-linear combination of any of the concentrations, C_i. Since the applications under consideration focus on biodegradation, much use is made of the empirical, non-linear Monod kinetics:

$$R_k = n_{mob} k_{\max} \cdot \prod \frac{C_i}{(K_M + C_i)} \cdot \prod \frac{K_I}{(K_I + C_i)} \tag{4.10}$$

k_{max} is the maximum rate constant [mol/m$^3_{mob}$/s]. K_M is the Monod half saturation constant used in the Monod term(s). K_I is the inhibition constant used in the inhibition term(s). K_M and K_I may vary between different reactions and different species, C_i. Variations of this formulation exist. For example, if biomass growth and decay are simulated, the rate may also be proportional to a biomass concentration, C_i, in which case the dimensions of k_{max} are altered such that the overall dimension of R_k is unchanged (e.g. Schäfer et al. 1998).

4.2.3 Equilibrium reactions

Certain chemical reactions may be reversible, and may have forward and backward rates that relatively quickly bring the reacting system into a state of thermodynamic equilibrium. In the case that equilibrium is achieved more rapidly than the timescale of the transport processes then a local equilibrium assumption can be applied, and an algebraic equation can be used. For example:

$$C_4 \Leftrightarrow 2C_2 + C_3 \tag{4.11}$$

In this example, C_2 and C_3 represent primary species such as in the above kinetic example (4.8). Kinetic reactions should only be formulated using primary species. C_4 represents a secondary species made of a linear combination of primary species (4.11), and whose concentration can be determined from the primary species concentrations by means of the mass action law (Stumm and Morgan 1996), which in this case gives:

$$\frac{a(C_2)^2 \cdot a(C_3)}{a(C_4)} - K = 0 \tag{4.12}$$

$a()$ represents the activity of the species and is calculated using the Davies equation (Stumm and Morgan 1996) which involves determining an activity coefficient and multiplying by the species concentration. K is the equilibrium constant.

In terms of the whole reactive transport equation system, each secondary species has an algebraic equation (4.12) associated with it, and each primary species has a mass conservation equation (4.1). In order to conserve the mass of the secondary species their mass is linearly combined with that of the primary species according to the reaction stoichiometries to create component concentrations, $TOTC_i$ (Steefel and MacQuarrie 1996, Saaltink et al. 1998). Continuing the example in (4.11) components are defined as:

$$TOTC_2 = C_2 + 2C_4; \; TOTC_3 = C_3 + C_4 \tag{4.13}$$

Or, in more general terms:

$$TOTC_i = \sum_j u_{ij} M(C_i) \tag{4.14}$$

Where u_{ij} contains the information on summation of secondary species into total components (Saaltink et al. 1998). If $i=j$ then $u_{ij} = 1$, e.g. $u_{22} = 1$ for the above reaction; if j is another primary species $u_{ij} = 0$, e.g. $u_{23} = 0$; and if j is a secondary species u_{ij} is the stoichiometric coefficient contributing to component i, e.g. $u_{24} = 2$. If i is a secondary species then $u_{ij} = 0$. The function $M()$ allows the components to be made up of species from different phases, e.g. the component TOTFe contains the mass of the mobile Fe primary species and may also contain mass from an immobile secondary species such as the mineral FeS, giving TOTFe = Fe + FeS. These total concentrations are then utilised in the mass conservation equation (4.1) which becomes:

$$\frac{\partial}{\partial t} \sum_j u_{ij} M(C_i) - \sum_j u_{ij} L(C_i) - SS_i = 0 \quad \text{for primary species} \tag{4.15}$$

This transformation means that the number of mass conservation, partial differential equations is equal to the number of primary species, or components. Note that immobile secondary

species, e.g. the mineral FeS, would not be transported since $L(C_i) = 0$. Only the mobile part of a component is transported. Equation (4.15) is effectively an advection dispersion reaction equation in terms of components.

The complete system of equations is established from the above set (4.15) together with a set of algebraic equations, one for each secondary species, of the general form:

$$\prod_i a(C_i)^{s_i} - K = 0 \qquad \text{for secondary species} \qquad (4.16)$$

where s_i is the stoichiometric coefficient used in the mass action law, see (4.12) for example.

In the work reported here, equilibrium has been implemented for types of reactions including aqueous complexation, mineral precipitation and dissolution, ion exchange and surface complexation. In the case of minerals, their activity is unity (Stumm and Morgan 1996) The mass action equation is omitted locally in the case where no mineral is present and the solution is under-saturated with respect to that mineral (cf. Schäfer et al. 1998) (Leeming et al. 1998). In the case of ion exchange, activity is replaced with equivalent fraction according to the Gaines-Thomas convention (Appelo and Postma 1993). For surface complexation species, the activity coefficient is unity (Stumm and Morgan 1996).

4.2.4 Solution approach

The complete description represents a differential algebraic equation (DAE) system consisting of partial differential equations (PDE) for transport and kinetic reactions of components (see (4.15)) and algebraic equations (AE) for secondary species at thermodynamic equilibrium (see (4.16)). The unknowns are the species concentrations, C_i.

The UG application which forms the basis of this work uses the global linearisation strategy to solve the fully coupled reactive transport equations by adopting an Eulerian approach (Bastian and Lang 2002). Thus, the split-operator approach which solves chemistry and transport separately (Yeh and Tripathi 1989) was not applied in this work. Full coupling has become the most commonly used approach in water resources (Steefel and MacQuarrie 1996, Kanney et al. 2003). Recent work suggests that for chemically complex systems, full coupling may be more robust and may run faster than split-operator approaches (Cirpka and Helmig 1997, Saaltink et al. 2001, Kanney et al. 2003). Full coupling also avoids truncation errors due to the operator splitting process (Valocchi and Malmstead 1992). One method of solving the coupled system is the direct substitution approach (DSA) where the secondary equilibrium species are eliminated through substitution into the transport equations (Yeh and Tripathi 1989, Saaltink et al. 1998, Mayer et al. 2002). This reduces the number of unknowns to the number of components, and is desirable in cases where a significant number of equilibrium reactions are considered, since the equation size will be smaller. In this work, the focus has been on biodegradation reactions

which are mainly kinetic so the savings expected by using DSA would be relatively minor. Thus the DAE form was used as this was significantly easier to implement into UG.

For realistic field scale biodegradation studies, the DAE strategy requires stable solvers (Cirpka and Helmig 1997) and efficient matrix storage and handling procedures. The complete system is discretised in space and time using common strategies, as described now.

Spatial discretisation in this application uses a standard vertex centred finite volume method (Bastian and Helmig 1999, Bastian and Lang 2002), and unless otherwise noted, a first order upwind scheme was applied. The spatial refinement adaption technique is described below.

The chemical reactions can form a very stiff equation system, especially when considering mobile-immobile interactions (Leeming et al. 1998). Thus an adaptive time stepping strategy is important (Leeming et al. 1998, Barry et al. 2002). Discretisation in time used the implicit, stiff backward difference formula (BDF) solver. The size of the time step was controlled by the non-linear solver in a simple way, by scaling the time step size up to a given maximum, and halving the time step if convergence was not achieved after a given number of non-linear iterations.

The non-linear system was linearised with the Newton-Raphson method; the Jacobian matrix (Steefel and MacQuarrie 1996) being created by numerical differentiation. A line search method was included in order to locate the bowl of quadratic convergence more rapidly.

The linearised system can be very large for multi-component multi-dimensional problems. UG offers an advanced multigrid method which works on a sequence of successively finer meshes with solutions on coarser grids filtering out longer wavelength errors and thus allowing a more rapid convergence at the finest grid level (Wesseling 1992, Bastian et al. 1997a, Bastian and Helmig 1999). The multiplicative geometric multigrid method with Gauss-Seidel smoothing is used as a preconditioner in a bi-conjugate gradient stabilised linear solver (cf. Neuss 2002, Wagner et al. 2002).

The fully coupled DAE system creates a potentially very large Jacobian matrix which can use large amounts of memory (Yeh and Tripathi 1989, Cirpka and Helmig 1997, Saaltink et al. 2001). UG uses an efficient block matrix storage scheme (Bastian et al. 1997a, Neuss 2002) where matrix blocks represent coupling between nodes. In multicomponent systems the blocks themselves may be sparse due to limited coupling between chemical species. The UG sparse matrix approach only stores the non-zero entries, the user having identified the sparsity pattern in advance based upon the chemical reaction scheme (Neuss 2002, Wagner et al. 2002). The benefit of this method is assessed using a realistic example problem below.

The UG toolbox is designed to execute computational tasks either sequentially, on one processor, or in parallel, where the workload is shared between multiple processors (Bastian et al. 1997a, Lang 2000). Many of the features of UG are able to exploit a parallel architecture. Trials showed that for a realistic biodegradation problem in 2D, with ten to twenty species,

computational times became impractical (for example more than a week) when using a single processor. Thus, use was made of the parallel processing capabilities of UG. For much of this work a Beowulf cluster of 8 processors was utilised. The cluster consisted of 4 AMD 1900XP Athlon dual processor nodes with 2GB memory each and connected with a 100 Megabit ethernet network. The workload is divided between the processors by a load balancing algorithm. Inertial recursive bisection (IRB) was used to divide the grid into similar sized processor domains while attempting to minimise the area of the boundaries between processors (Bastian et al. 1997b). Nodes that lie near a processor boundary may require exchange of information from nodes in the adjacent processor domain. This communication step is often the main task limiting the efficiency of a parallel application. Since in this work the multispecies reactive transport simulations focus on relatively complex biodegradation processes, the computational load is expected to be largely associated with coupled chemistry at the nodes rather than transport between the nodes. Therefore it was expected that there might be a relatively high amount of processing within domains compared with communication between them. Therefore the speed benefit should scale well with the number of processors used. The scalability of an example realistic field problem is assessed later.

UG's multigrid approach makes it well suited to local adaptive refinement methods (Bastian et al. 1997a, Haefner and Boy 2003). Since the multigrid already operates on a series of successively more refined grid levels, it is simple to envisage that higher grid levels can be applied locally in the domain – the *h*-method of adaption. Further, the extent and position of the local refinement can move as the simulation progresses, as determined by an error estimator or error indicator. With regard to grid topology, UG decides how elements should be refined or coarsened in a consistent manner (Bastian et al. 1997a, Bastian 1998). Since formal error estimators for highly non-linear typical biodegradation problems cannot be found easily, if at all, and are very costly (Wagner et al. 2002), a pragmatic choice of error indicator was made instead. A flexible indicator package was used which allowed refinement to be made based upon the absolute values of any of the chemical species, or upon their spatial gradients (difference in value across an element), or upon the nodes position in the grid, or a combination of all these. In this work, the aim was to refine on the plume fringe, defined principally as the zone where oxygen and nitrate were consumed. Thus an example of an error indicator could be to refine where the oxygen gradient became greater than 1% of the background oxygen concentration, but only to coarsen (de-refine) the grid where the nitrate concentration fell below a very low value (for example 10^{-6}mM).

As mentioned earlier relatively little has been reported on the use of adaptive refinement with reactive transport problems (Wolfsberg and Freyberg 1994, Mansell et al. 2002, Wagner et al. 2002). Wolfsberg used refined subgrids within a uniform grid and applied it to an ion exchange problem with 3 mobile and 3 sorbed species (Wolfsberg and Freyberg 1994). Wagner *et al*

(2002) used UG with a solution procedure similar to this work, and applied it to chlorinated solvents biodegradation using 9 kinetic species, but parallel processing was not used.

The effectiveness of adaptive remeshing in conjunction with parallel processing has been demonstrated for typical multiphase flow problems (Bastian et al. 2000), but with no more than 3 species. In the work reported here the use of adaption together with parallel processing of chemically complex systems with greater than ten species is assessed.

4.3 Test Cases

Several test cases have been completed using the new UG application in order to verify it by comparison with existing published simulations of biodegradation problems. Comparisons are made against codes which use either full coupling or split-operator approaches. Table 4.1 summarises all the simulations used in this paper to test the new UG application, and to test the numerical tools offered by UG. The test cases demonstrate the flexibility of this modified UG application in terms of chemical processes and phases that can be included. Some other ancillary points are made as well.

Table 4.1 Summary of model simulations in this paper.

Case	Ref.	Dim.	No. of species	Scale	Location	Testing	Numerical test performed
A	p	2D	2	Lab.	box	UG-RT3D	Adap
B	q	1D	12	Lab.	column	UG-TBC	
C	r	1D	24	Field	F.A.	UG-MIN3P	
D	-	2D	1	Field	F.A.		Par, Adap, PA
E	s	2D	24	Field	F.A.		Sparse, PA
F	s	3D	11	Field	F.A.		Par

Notes. Ref. - reference. p - (Huang et al. 2003). q - (Schäfer et al. 1998). r - (Mayer et al. 2001). s - (Watson et al. 2004b). Dim. – dimensions of simulation. Lab. – laboratory. F.A.- Four Ashes, UK field site. UG-*** UG was tested against published results from the codes listed. Par. – parallel processing tested with fixed uniform grid. Sparse – sparse matrix storage scheme tested. Adap. – adaptive remeshing tested on a single processor. PA – parallel adaptive remeshing tested.

4.3.1 Case A: 2D plume visualisation experiment.

This test case is based upon the simulation used by Huang *et al (2003)* which had the following characteristics. The experiment involved aerobic degradation of acetate in a 2D sand box with an indicator dye giving the spatial distribution of oxygen remaining in the porous media. Water flows steadily through the 150mm long box. The experiment ran for 1500 minutes. The simulation has two aqueous components, acetate and oxygen, with one kinetic reaction simulated between them using double Monod kinetics (see (4.10)). At the end of the simulation a steady state was reached.

In the original paper *(Huang et al. 2003)*, the RT3D code solved this system using operator splitting and the sequential iterative approach, and the TVD solver was applied. Here UG solves

the fully coupled DAE with an implicit, upwind method (as described above). Appendix C, Table 9.1 to Table 9.3 show all the model specific parameters and equations used in this simulation. In UG, steady state was observed to have been reached after 300 to 440 minutes so the simulations were completed at 500 minutes. These relatively simple simulations were completed on a single processor.

Figure 4.1 shows a 2D plan illustration of the half the plume which was simulated. Figure 4.2 compares the UG results to the published results *(Huang et al. 2003)* along the same transect (X = 76mm) as in the original publication. UG compares well with the RT3D simulation which was closest to the experimental data. This work has not attempted to fit the experimental data better, but rather to compare the accuracy of the numerical algorithms. An UG simulation was also undertaken using central differences instead of upwind spatial discretisation but this did not lead to any significant difference in the results.

This is a simple benchmark case, demonstrating UG's fully coupled solver to give comparable results to RT3D.

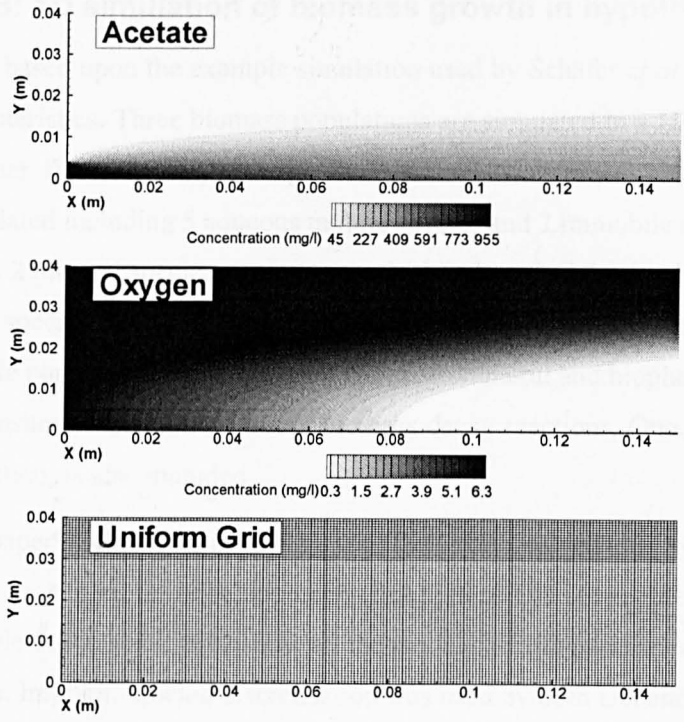

Figure 4.1 Case A 2D contour plots of acetate and oxygen at steady state after 500 minutes simulated with UG on the uniform grid.

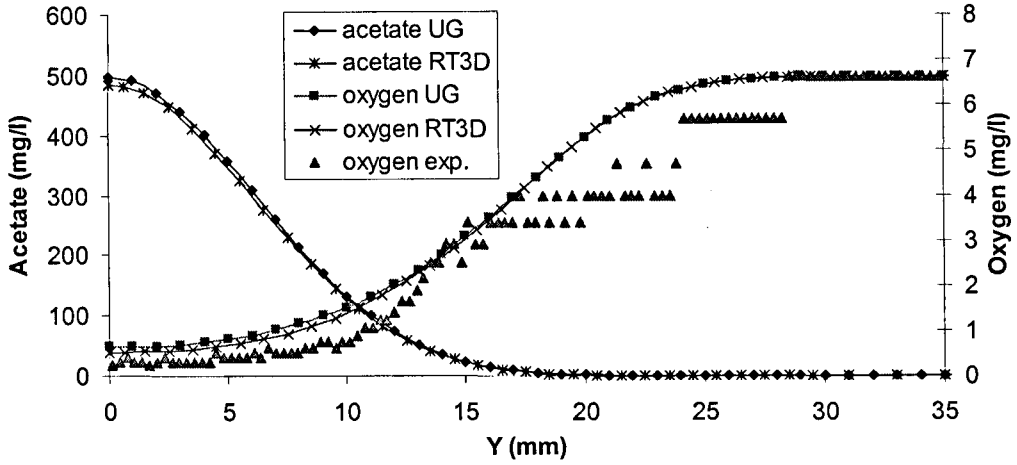

Figure 4.2 Case A UG results vs. RT3D results and experimental data plotted on a Y transect across plume at x=76mm (about half the plume length) after steady state has been attained.

4.3.2 Case B: 1D simulation of biomass growth in hypothetical column

This test case is based upon the example simulation used by Schäfer *et al* (1998) which had the following characteristics. Three biomass populations are simulated in a 1D hypothetical column experiment. Water flows steadily through the 1m long column for 100 days. A total of 12 species are simulated including 5 aqueous mobile species and 7 immobile species which include 1 NAPL species, 2 mineral species, 1 nutrient in the biophase and 3 microbial populations in the biophase. These species interact via 8 kinetic reactions which include linear exchange between NAPL and mobile pore water, linear exchange between mineral and biophase, 3 biomass growth and substrate consumption reactions and 3 biomass decay reactions. One equilibrium reaction, mineral precipitation, is also included.

In the original paper (Schäfer et al. 1998), the TBC code solved this system using operator splitting and the sequential iterative approach (SIA). Here UG solves the fully coupled DAE. Appendix C, Table 9.4 to Table 9.10 show all the model specific parameters and equations used in this simulation. Implicit, upwind discretisation was used by both UG and TBC.

Figure 4.3 shows that the UG results compared very closely to the TBC results obtained by using 20 iterations between the transport and chemistry. When the TBC model was run with only three iterations per time step significant operator splitting errors were revealed; the main difference being an overestimation of the amount of sulphate reduction. The fully coupled method of the UG application does not suffer from operator splitting errors.

The agreement between these 1D TBC and UG simulations demonstrates that UG is capable of simulating reactive transport with biogeochemical reactions including phase exchange reactions, biomass growth and decay, and equilibrium mineral precipitation. Runtimes were not compared

in this exercise since the models were run on different machines and the grids were different because UG used a 2D grid refined in one direction only.

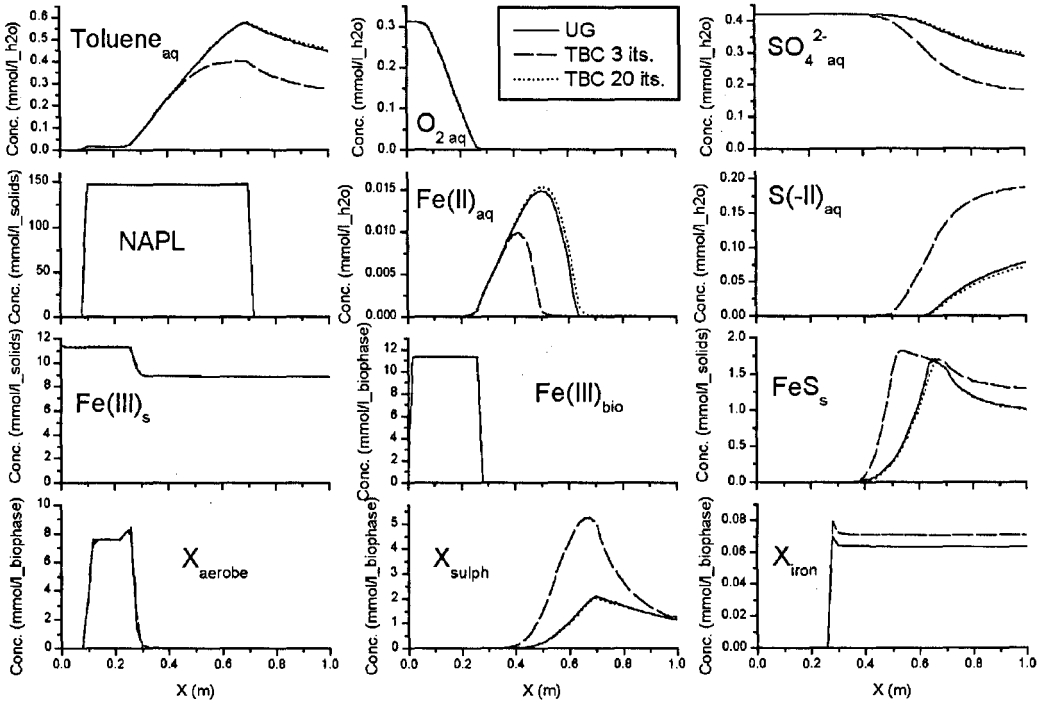

Figure 4.3 Case B profiles along column at t=100days.
Shows UG results (solid lines), TBC results with 3 iterations (dashed lines), and TBC results with 20 iterations (dotted lines) between transport and chemistry each time step.

4.3.3 Case C: 1D flowline field simulation, Four Ashes site

This test case is based upon the published simulation of Mayer *et al* (Mayer et al. 2001) of "Flowline I" in the phenolics plume at the Four Ashes field site. In Flowline I, groundwater flows steadily along the 750m 1D domain for 47 years. At the source, phenolics enter the groundwater and these are consumed by multiple electron acceptors which are present in the aquifer. The simulation shows the evolution of the plume geochemistry as the phenolics, expressed as phenol, are biodegraded. Overall 24 chemical species are simulated, including 16 aqueous mobile species, 6 aquifer minerals, and 2 ion exchange surface species. Reactions include 6 Monod kinetic degradation reactions and 8 thermodynamic equilibrium reactions, including 3 homogeneous aqueous complexation, 4 heterogeneous mineral dissolution-precipitation, and 1 ion exchange reaction. The phenol source concentration decreases after 25 years. Appendix C, Table 9.11 to Table 9.14 show the parameters.

The published simulation was performed with MIN3P (Mayer et al. 2002) which employs full coupling using DSA, whereas UG solves the DAE with no substitution. While both codes use the finite volume method, MIN3P uses the van Leer flux limiter discretisation, whereas in this case UG was set up to use standard central differences spatially.

Figure 4.4 shows that the aqueous, mineral and ion exchange species are generally well reproduced by UG compared with the MIN3P simulation (Mayer et al. 2001). The aqueous species, Fe^{2+} and Mn^{2+}, are strongly controlled by equilibrium with mineral phases; UG calculates activity coefficients using the Davies equation, while MIN3P uses the extended Debye-Huckel equation (Stumm and Morgan 1996). Additional investigations (not shown) showed the activity calculation methods to be the principal cause of the observed minor difference between the simulations.

These results suggests that, in this case, central differences provides similar results to the van Leer discretisation. Once again, simulation times were not compared since they were not run on the same computer.

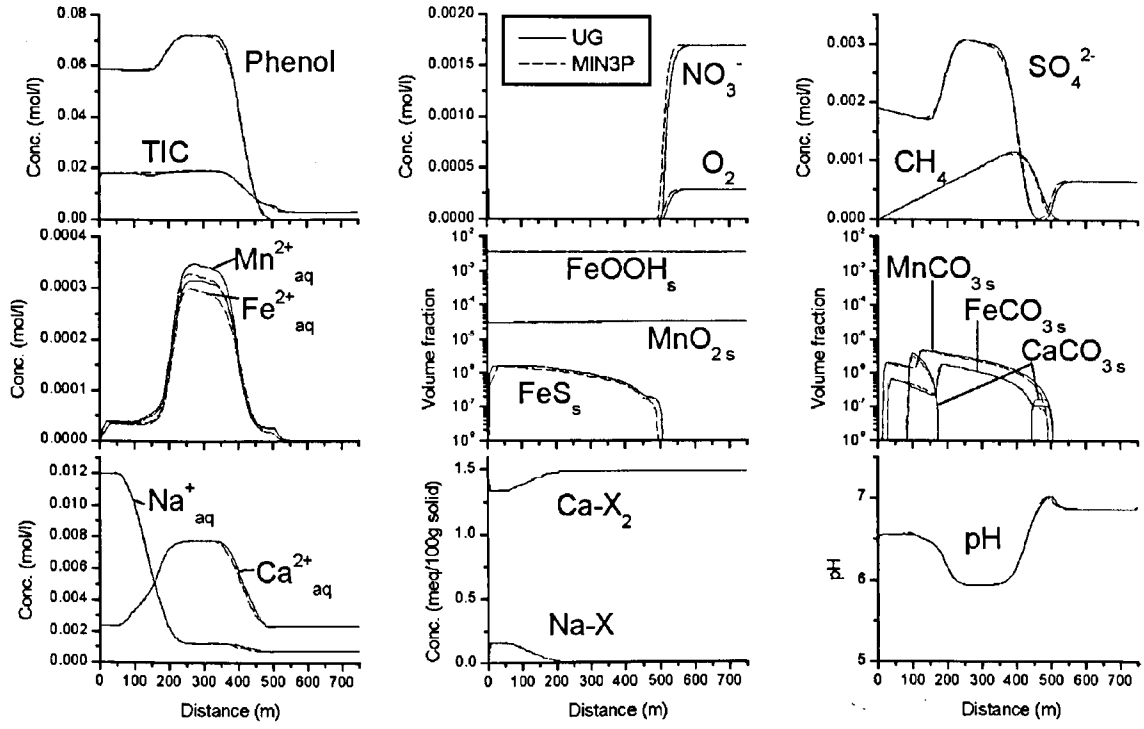

Figure 4.4 Case C profile along flowline at 47 years. Shows UG results (solid lines) and MIN3P results (dashed lines) for aqueous, mineral and ion exchange species.

4.4 *Multi-dimensional field scale simulations*

The following three models are all based on the phenolics plume at Four Ashes, UK, as is case C. These are used to assess the performance, with respect to biodegradation in particular, of some of the numerical strategies available in UG, including parallel processing, sparse matrix

storage, and adaptive remeshing, see Table 4.1 for a summary. All the simulations used upwind discretisation which avoids the non-physical oscillations that can be induced by central differencing (see case C), but does result in some extra numerical dispersion. Rather more accurate discretisation schemes, like discontinuous Galerkin methods (Shu et al. 2000), or Eulerian-Lagrangian Localised Adjoint Methods (ELLAM) (Celia et al. 1990), are under consideration, but such techniques are not trivial to implement in the complex multi-species simulations tackled here (Cirpka and Helmig 1997).

4.4.1 Case D: 2D single species non-reactive transport

This simple one-species model uses the same physical flow parameters as case C. The 2D vertical domain is 40m high by 750m long. Only phenol is considered, being introduced as a constant source of concentration of 70mM, along a line source 20m long placed centrally on the inflow boundary. Although case D is a symmetric plume, the whole plume is simulated to give consistency with the non-symmetric plumes of cases E and F.

4.4.2 Case E: 2D multi-species realistic biodegradation

Case E corresponds to the 2D model with a uniform flow field of the Four Ashes site as presented in Watson *et al* (2004b); physical and chemical simulation parameters are given there. The 2D vertical domain is the same as in case D. The model has 24 species simulated by 8 kinetic reactions. Phenol is again the primary organic contaminant, but, unlike case D, the phenol source is spatially variable and changes in time, giving a more realistic plume morphology. The principal difference between the kinetic chemistry used here and in case C is that here a two step reaction process is implemented. This simulates the fermentation of phenol to intermediate species, hydrogen and acetate, followed by the further consumption of hydrogen by respiration reactions. This leads to the differing parameter set reported in Watson *et al* (2004b), including several secondary equilibrium species: aqueous complexes, minerals and surface complexation species. The advantage of including these extra species is that the fate of microbially produced Fe(II) can be simulated as it can be present in the mobile aqueous phase or the immobile minerals or surface complexes (see (Watson et al. 2004b) for discussion). Figure 4.5 shows selected results for case E with a locally adapted mesh.

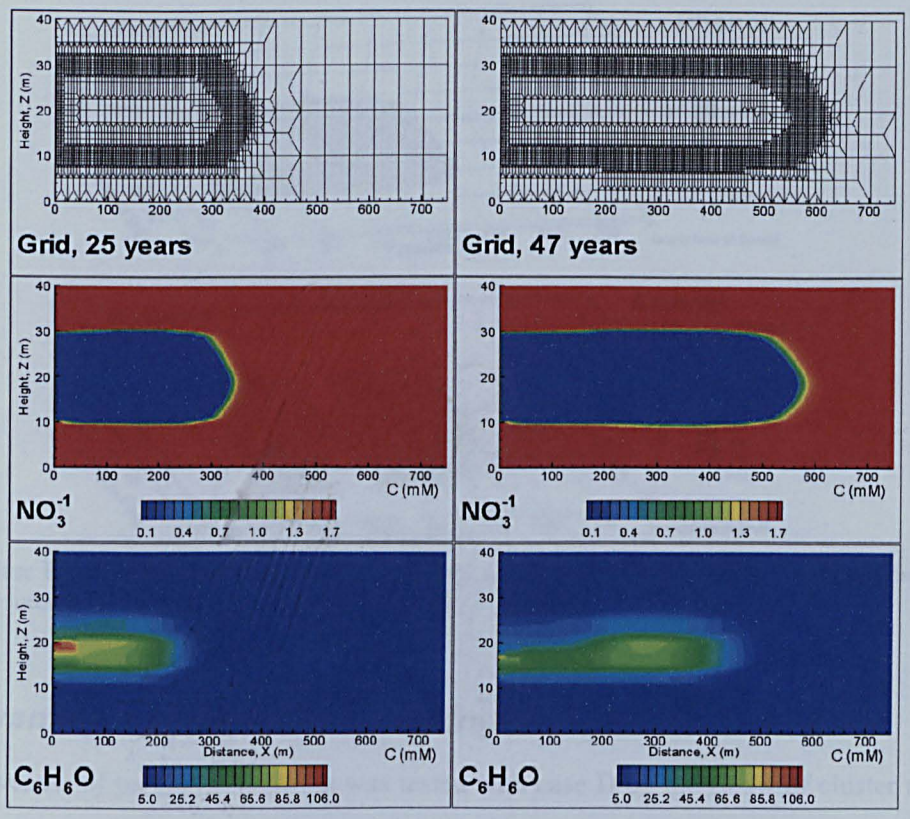

Figure 4.5 Case E results for NO_3^{-1} and C_6H_6O at 25 and 47 years, simulated with an adaptively refined and coarsened mesh.

4.4.3 Case F: 3D multi-species realistic biodegradation

Case F corresponds to the 3D model with a deterministic source zone of the Four Ashes site reported in Watson *et al* (2004b). Case F uses the same 8 kinetic reactions as case E, but only 11 species are simulated. Equilibrium chemical species are omitted for simplicity since the 3D problem is more demanding in terms of size than 2D. The domain is 90m wide and has the same height and length as cases D and E. Figure 4.6 illustrates selected results for case F.

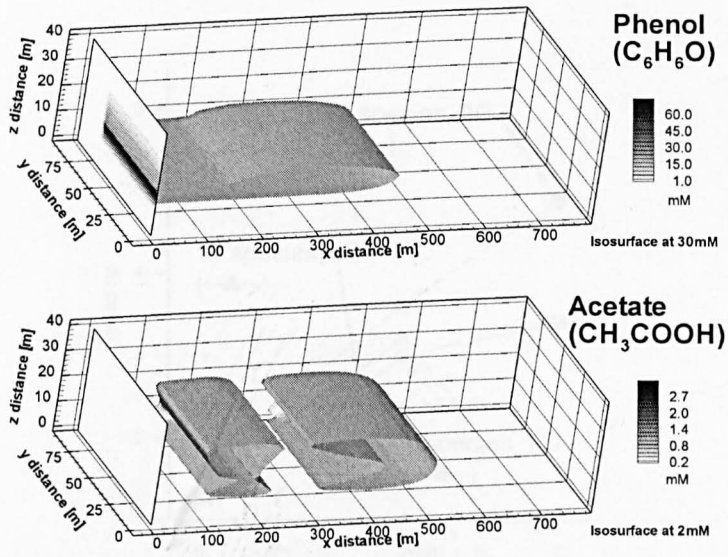

Figure 4.6 Case F results for phenol and acetate as a selected isosurface at 47 years. The inflow boundary plane is contoured with the phenol source concentration.

4.5 *Advantages of parallel processing*

The effectiveness of parallel processing was tested with case D on the Beowulf cluster using a uniformly refined grid (no adaption). Table 4.2 and Figure 4.7 show the results in terms of the speed-up gained as more processors were used. In increasing from 4 to 6 processors the efficiency (speed-up / number of processors) appears to drop from 0.75 to 0.57. While the observed low efficiency of using 2 processors is due in part to the doubling of the number of linear iterations taken, the drop in performance with 4 and 6 processors appears to be due to the decrease in processor domain size, and increase in communication across processor boundaries.

Table 4.2 Parallel efficiency tests using case D: single species

No. of processors	Nodes per processor	Linear iterations	Runtime (mins)	Speed up	Efficiency
1	66177	1034	95	1	-
2	33165	2034	62	1.5	0.75
4	16659	2036	32	3.0	0.75
6	11508	2036	28	3.4	0.57

Case D has one component in 2D, here simulated on uniform level 7 grid.
Simulations were run on the Beowulf cluster (see text for details).
NB. earlier tests in 2D with 19 species gave 5 times speed up with 6 processors.

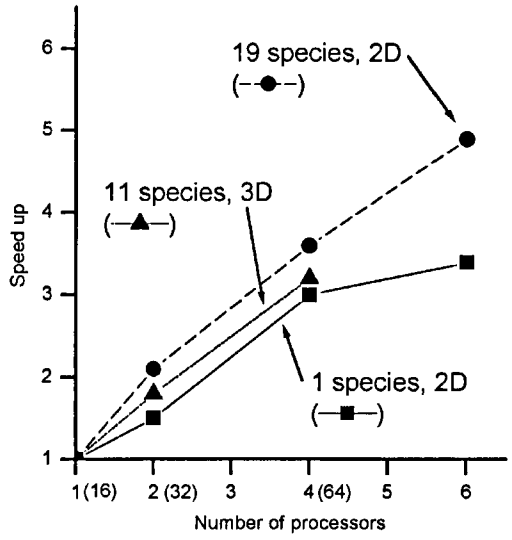

Figure 4.7 Parallel speed up for single and multi-species simulations.
Plots are: 1 species, 2D, case D on Beowulf cluster, see Table 2; 11 species, 3D, case F on HELICS (no. of processors in brackets), see Table 3; 19 species, 2D on Beowulf (Watson et al. 2003a)

In contrast to this single species model, earlier work on a 2D model similar to case E performed notably better (Watson et al. 2003a), see Figure 4.7. That problem simulated 19 species at 4257 mesh nodes on the Beowulf system and showed that using 6 processors could achieve a speed up of 4.9 corresponding to an efficiency of 0.82.

Interestingly, the multi-species reactive model appears to scale better than the single species, non-reactive model. To assess whether this multi-species scalability is maintained at a higher number of processors, a different and larger parallel machine was used to undertake additional test simulations for case F. The HEidelberg LInux Cluster System (HELICS) at the Interdisciplinary Centre for Scientific Computing (IWR) of the University of Heidelberg (see http://helics.iwr.uni-heidelberg.de) was used. HELICS has 512 AMD Athlon MP processors with clock speeds of 1.4GHz, connected via a high-speed Myrinet2000 network with a point-to-point throughput of 2 Gigabit per second between two computing nodes. The system has a total of 512 Gigabytes of distributed memory.

The 3D multi-species case F showed that increasing the number of processors from 16 to 64 gave an efficiency of 0.8 relative to the 16 processor simulation (Table 4.3 and Figure 4.7). Thus, the test simulations appear to show that multi-species problems have better parallel scaling properties than single species problems, and that multi-species problems scale well up to a high number of processors.

Table 4.3 Parallel Runs made on case F : multispecies

No. of processors	Nodes per processor	Linear iterations	Runtime (mins)	Speed up	Efficiency
16	1443	8353	530	1	-
32	800	8334	289	1.8	0.9
64	476	8338	164	3.2	0.8

Case F has 11 species in 3D, here simulated on uniform level 4 grid giving 18785 unique nodes in total (~207,000 degrees of freedom). Simulations were run on the HELICS cluster (see text for details).

For the multi-species simulations a large amount of work has to be done by the Gauss-Seidel smoother in the multigrid solver which has to invert each of the diagonal blocks of the Jacobian matrix. A diagonal block represents the mass storage and source sink terms in the governing equation (4.1), i.e. the coupled reactive chemistry at a node, while the off-diagonal blocks represent the transport terms between nodes. Thus, the bulk of the work is done on the processors solving the chemistry, and not in communicating between processors across the network, and this is the reason for the rather impressive scaling observed. In contrast, single species problems have a far higher proportion of work to do across processor boundaries. Thus, when the boundaries become more important as processor number increases, the efficiency will decrease.

4.6 Advantages of sparse matrix storage approach

Case E, realistic multi-species biodegradation, was used to test the efficiency of the sparse matrix approach for storing only non-zero entries (Neuss 2002), explained above. The model has 24 species which are highly coupled in the diagonal blocks of the Jacobian matrix, but the off diagonal blocks are highly sparse, especially since all the immobile and equilibrium species are not transported, and have rows of zeros.

Table 4.4 shows the effect of storing either full or sparse matrices in the diagonal or off-diagonal blocks. The numerical experiments were conducted on a uniform grid in parallel on the Beowulf machine. Simply switching the diagonal blocks between sparse or full storage does not make a large difference in terms of memory or runtime. However, comparison of the sparse matrix approach to the full storage of the blocks shows that only a twentieth of the memory is needed and 28% runtime is saved. Similar savings were also observed for other examples, but are not presented here. As shown in the original paper (Neuss 2002) the savings in general increase with the number of components. The substantial memory savings may mean that certain problems can be run on a single machine, without needing to use a cluster, as already reported by Wagner et al (2002). This is an important method to save time and memory in multi-species problems such as biodegradation problems.

Table 4.4 Sparse matrix savings using case E

Diag Blocks	Off-diag Blocks	Memory (MBpp)		Run time (mins)	
Full	Full	633	100%	4600	100%
Full	Sparse	62	9.8%	3635	79%
Sparse	Sparse	34.7	5.5%	3316	72%

Simulations were run on 6 processors of the Beowulf cluster with a uniform grid level 6. Case E has 24 species. Memory usage expressed as MB per processor (pp)

4.7 Advantages of adaptive remeshing

In this section the 2D models described above are used to assess the local adaptive remeshing facility in UG in terms of its usefulness for realistic field scale problems. A variety of pragmatically chosen *a posteriori* error indicators are chosen to give differing refinement patterns. The accuracy and efficiency of these simulations are compared to uniformly refined models run sequentially (on one processor) or in parallel.

4.7.1 Adaptive remeshing with case A, sequential processing

Case A, dual Monod kinetic degradation, was used as the basis for exploring various error indicators for adapting the mesh. The indicators, d, for local adaption were based upon the concentration difference across a cell expressed here as a percentage of the maximum concentration for that species in the whole domain (see Table 4.5). Thus, if the concentration difference exceeded d, local refinement of that element would occur. Maximum levels for refinement and minimum levels for coarsening (de-refinement) were also set. All the simulations were completed on a single processor. Adaptive simulations were compared to the uniform simulation in terms of the number of nodes at the end of the simulation, the total number of iterations required by the multigrid solver, and the run time. The accuracy was assessed in terms of the maximum absolute concentration difference between the adapted solution and the uniform solution (explained in Table 4.5). For each choice of gradient indicator, d, the effect of refinement was first tested, then coarsening was switched on additionally.

Table 4.5 Results of sequential adaptive 2D simulations with case A

Refinement indicator	Coarsening indicator	Final nodes	Linear iterations	Run time (mins)	Max. Conc. Error (O_2)
- (uniform mesh)	-	16705	1272	45	0%
$d > 1\%$	-	12034	2320	41	0.56%
$d > 1\%$	$d < 0.5\%$	11753	2325	40	0.59%
$d > 5\%$	-	8995	2322	31	7.3%
$d > 5\%$	$d < 2.5\%$	6919	2384	26	8.2%
$d > 10\%$	-	4648	2043	14	29.7%
$d > 10\%$	$d < 5\%$	3514	2049	11	30.5%

Simulations were run on 1 processor of the Beowulf cluster. Indicators: d is difference in value across any cell, parameter given as percentage of maximum concentrations, i.e. 1000mg/l acetate and 6.6 mg/l oxygen. Refinement was up to level 6 and coarsening was allowed down to level 1, the coarsest being level 0. One grid adaption was allowed every time step. Maximum Concentration Error is defined as max(| (adaptive solution) - (uniform solution) |) and is given as percentage of maximum concentration of oxygen. Acetate error was always smaller than oxygen error, and is not shown.

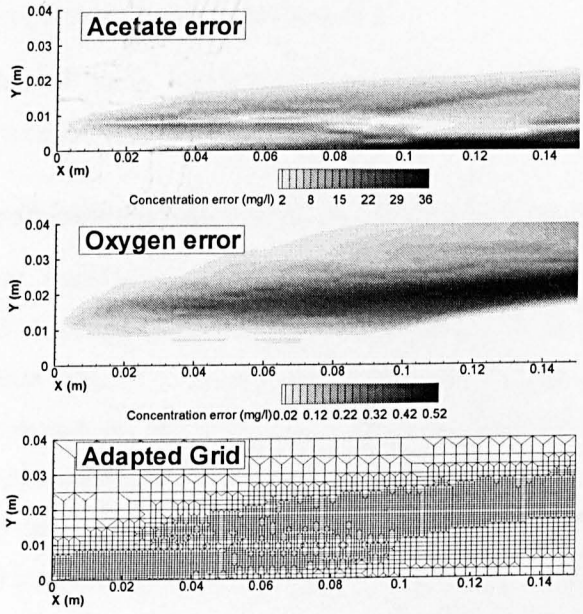

Figure 4.8 Concentration errors for adaptive simulation of test case A using 5% gradient indicator for refinement and 2.5% gradient indicator for coarsening.
Contours of absolute concentration error (adaptive grid solution – uniform grid solution) are shown (units: mg/l); c.f. Figure 4.1 for uniform grid solution.

Adaption with the 1% gradient indicator gives approximately 10% decrease in run time, and only a slight loss of accuracy (Table 4.5). In this case only a few cells were actually coarsened so there is minimal difference between switching coarsening on or not. Adaption using 5% gradient indicator gives a further 20% decrease in run time, but now the concentration error is greater than the 5% indicator which was applied. However, it is noted that applying coarsening results in a good improvement in run time and only a small increase in error; Figure 4.8 shows the error map and final mesh for this case. Adaption using 10% gradient indicator gives a further significant improvement in run time with up to 75% saved compared with a uniform mesh, but now the error is much larger than 10%. This is unacceptable for all but preliminary simulations; however, as such these could be performed four times faster and with a fifth of the

memory compared to the uniform grid simulation. Thus, coarsening the grid inside the plume consistently improved the efficiency in this example. The coarsened models used slightly more iterations than those which were only refined, but since many nodes were saved, the overall workload was reduced.

Overall it appears that the 1% gradient indicator is a good pragmatic choice for this kind of conceptual model, i.e. 2D dual Monod kinetics. The consequent error should be less than 1% which is acceptable, given that dual Monod parameters are often poorly known and case specific. Although 1% indicator works well for this test example, this does not imply that it will work for other reactive transport cases. However, this example is typical of many groundwater contamination scenarios where dissolved oxygen, although present in much smaller concentrations than the contaminant, is the primary electron acceptor for the contaminant biodegradation, and requires a detailed understanding.

The adaptive simulations show that a judicious choice of indicator can improve model efficiency with only a small loss of accuracy compared with a uniform grid.

4.7.2 Adaptive remeshing with case D, parallel processing

Case D, single species, non-reactive transport, was used to assess the applicability of adaptive remeshing in parallel. Gradient error indicators were applied to the advancing phenol plume giving various adaption patterns. The simulations were completed on six processors on the Beowulf cluster. The model was not set up to re-balance the load following grid adaption. This means that as the plume moves through the domain some processors have more refined grids than others and will have more work to do. Thus some of the efficiency of the parallel processing is lost, but also some time is saved by not re-balancing the load.

Table 4.6 summarises the efficiency gains of adaption. On one processor, with one grid adaption per time step applying both refinement and coarsening gave run time savings of 75%. The coarsened versions were solved quicker than the refined-only versions. A few more iterations were taken by the coarsened versions, but many nodes were saved which reduced run time. Since the initial grid was at level 2, and the most refined at level 7, it was felt appropriate to allow 5 adaptions per time step (each adaption locally changes the grid refinement by at most one level), so that sharp fronts could be locally refined fully and rapidly. The results in Table 4.6 show that on a single processor using 5 adaptions rather than 1 adaption per step adds a large computational overhead, roughly doubling the simulation time. In parallel, this adaptive overhead appears to become much worse, and the adaptive simulations actually ran significantly slower than the uniformly meshed ones (see Table 4.6). Thus, it appears that the adaptive process requires a significant amount of communication effort between processors which slows down this single species simulation. It is suggested that reducing the adaption to, say, 1 adaption per 5 time steps would improve efficiency, but the accurate capturing of sharp fronts would be

compromised. However, it is shown later that the adaptive overhead is not as pronounced in multi-species simulations, so multiple adaptions per step become acceptable.

Table 4.6 Run times of parallel adaptive 2D single species simulations with case D

Uni.	Rfn.	Crs.	Final Nodes	Linear iterations	Run time (mins)			
					1 proc, 1 adap	1 proc, 5 adap	6 proc, 1 adap	6 proc, 5 adap
7	-	-	66177	2036	95	95	28	28
-	7	-	27206	2037	44	85	38	94
-	7	6	14312	2038	27	53	27	69
-	7	5	12227	2038	25	48	25	65
6	-	-	16705	1034	24	24	7	7

Simulations were run on the Beowulf cluster. Uni. – maximum uniform level. Rfn. – most refined grid level. Crs. – lowest level grid coarsened to following refinement. In adaptive cases the initial grid is at level 2. All refinements made with d >1% (=0.7mM phenol) and coarsening if d <0.5%. proc is the number of processors used, and adap is the number of grid adaptions allowed per time step.

Table 4.7 Accuracy of parallel adaptive 2D single species simulations with case D

Uni.	Rfn.	Crs.	Max. Conc. Error	Plume Length Error	Adaptive Mass Error
7	-	-	8.4%	3.4%	-
-	7	-	7.72%	17.9%	0.04%
-	7	6	7.74%	17.9%	0.05%
-	7	5	7.74%	17.9%	0.06%
6	-	-	12.4%	7.9%	-
-	6	-	12.1%	20.5%	0.03%
-	6	5	12.1%	20.4%	0.05%

Simulations were run on 6 processors of the Beowulf cluster. Uni. – maximum uniform level. Rfn. – most refined grid level. Crs. – lowest level grid coarsened to following refinement. In adaptive cases the initial grid is at level 2. All refinements made with d>1% (=0.7mM phenol) and coarsening if d<0.5%. Maximum concentration error is max(|numerical solution - analytical solution|) measured along plume centreline.
Plume length is the distance travelled by the 0.1mM phenol contour (Thornton et al. 2001a). Plume length error is the proportion that the simulated contour has travelled above the analytical solution (517.5meters travelled). Adaptive mass error is cumulative (total) error in global mass introduced by changing the grid, normalised to maximum mass.

The accuracy of the simulations was assessed in a variety of ways including concentrations, plume length and plume mass (see Table 4.7). Since an analytical solution is available for this constant line source problem (Domenico and Schwartz 1998) this was used to compare the simulated concentrations. The maximum concentration error along the plume centreline, Table 4.7, shows the coarser grids to be the least accurate. The adapted grids, surprisingly, appeared to be more accurate than uniform grids using this measure. However, Figure 4.9 illustrates that the adapted grids are only locally more accurate than the uniform grids. The principle reduction in concentration error is due to using an extra level of refinement (see Table 4.7), although this does not appear to be a particularly significant error reduction. Figure 4.10 shows the

concentration error of the level 7 adaptive mesh models relative to the level 7 uniform model. The maximum error is approximately 1% of the maximum phenol concentration, and this arises from using the 1% gradient indicator (see Table 4.7). This is similar to the observations made with the sequential adaptive two species example above.

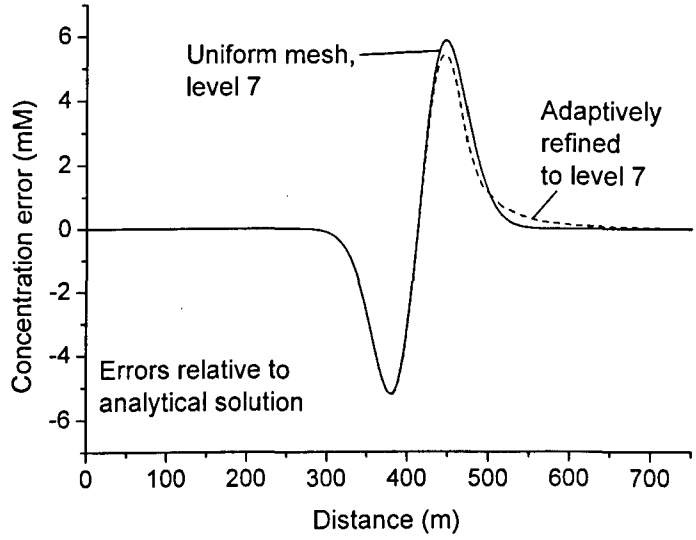

Figure 4.9 Concentration error along plume centreline of case D. Adaptive and uniform grid solution errors shown relative to analytical solution.

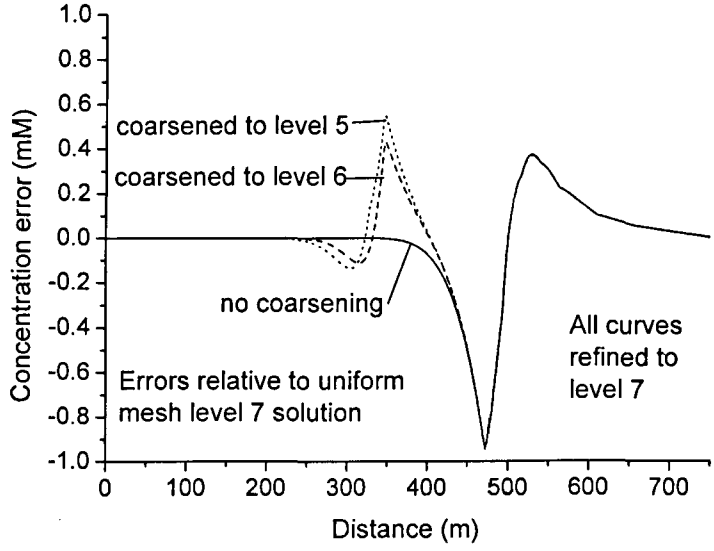

Figure 4.10 Concentration error along plume centreline of case D. Error of adapted simulations is shown relative to uniform solution: (adaptive grid solution – uniform grid level 7 solution)

Figure 4.10 shows that all the simulations have the same concentration error ahead of the front of the plume, but that coarsening inside the plume gives an additional source of error. However, the coarsening error is localised inside the plume where the coarsening occurs and does not appear to produce effects elsewhere in the domain. Thus, in examining the error in simulated plume length (see Table 4.7) there is negligible difference between the plume lengths of the adapted simulations with or without coarsening. The plume length errors are about 20% for the adapted grids, but only 3.4% for the finest uniform grid (Table 4.7). The poorer performance of the adapted grids here is attributed to the use of a 1% (0.7mM phenol) gradient indicator when the plume length is defined at the 0.1mM phenol contour which, therefore, cannot be in the most refined region of the adapted grid. The plume width errors, based on the same 0.1mM contour, on a section transverse to the plume were found to be slightly less than the stated plume length errors.

Finally, it is noted that some error is added by the actual adaption process itself, i.e. migrating mass from one grid to a newly adapted one. This adaptive mass error was found to be relatively small with only around 0.05% added to the total global mass of phenol throughout the simulation, see Table 4.7. A similarly small adaptive mass error was found when simulating case A with an adaptive mesh, above.

4.7.3 Adaptive remeshing with case E, parallel processing

This section aims to illustrate the benefits of applying a parallel adaptive modelling tool to a relatively realistic biodegradation model. This has not been reported in the literature, although UG has been used to apply parallel adaptive multigrid methods to multiphase problems (Bastian et al. 2000). UG has also been used to simulate biodegradation in groundwater with adaptive multigrid methods on a single processor (Wagner et al. 2002).

The large number of species used, presents a wide choice of potential error indicators. Adaption based on fringe processes is desirable since this is often the most active part of a plume and may be the part where the highest resolution field data is collected. The plume has reactions occurring at the fringe and in the core (Watson et al. 2004b). Refining the whole plume, without coarsening, should give accurate results as was seen in the previous adaptive examples, above. However, the inside of the plume is characterised by many species with gradients caused by the source term (e.g. phenol) and by the reactions (e.g. Fe(II)), so it is not clear that adaptive coarsening inside the plume will be beneficial. In order to test the usefulness of adaptive refinement of the reactive fringe, the following indicators were chosen. For adaptive refinement, the oxygen was known to be the species reacting at the outer edges of the plume fringe, so a gradient indicator of 1% O_2 was chosen to ensure refinement to the edges of the plume. For coarsening, it was decided to define the core of the plume as being surrounded by a very low nitrate concentration contour. Thus, the error indicator allowed coarsening for values of nitrate

less than 10^{-4} mM. All the simulations were completed on six processors on the Beowulf cluster. As with case D, above, the model was not set up to re-balance the load following grid adaption.

The results are shown in Table 4.8. In this multispecies example the fastest solutions are obtained by the coarsened grids, followed by the refined grids and the uniform grids. Using adaption saves 50% of the run time in this example, which is significantly better than the single species parallel adaptive example above (case D, Table 4.6). Thus in this case, the overhead of grid adaption is not so great as to compromise the efficiency of the adaptive method. This is because the system is larger and a proportionately longer time is spent solving the system than adapting the grid. It is also noted that while coarsening saves nodes, it also causes more iterations to be taken. Table 4.8 shows that coarsening to level 4 rather than level 5 uses less nodes but takes more iterations and overall no time is saved. The extra iterations are assumed to be due to the high number of species in the plume core which are reacting and exhibit concentration gradients. This follows the pattern of the above adaptive examples, where with one species, coarsening caused negligible extra iterations (case D, Table 4.6), two species caused several extra iterations (case A, Table 4.5) and multiple species now causes many extra iterations (case E, Table 4.8). To summarise the efficiency of parallel multispecies adaptivity, refinement is more efficient than using a uniform grid, and plume fringe coarsening may be more efficient depending on the balance of nodes saved and extra iterations needed to solve plume core processes on a coarser grid.

Table 4.8 Results of parallel adaptive 2D multi-species simulations with case E

			Efficiency			Global mass			
Uni.	Rfn.	Crs.	Final nodes	Linear iterations	Runtime hours	Phenol mols	O_2 mols	NO_3^{2-} mols	FeII mols
6	-	-	16705	15895	55.3	100.0%	100.0%	100.0%	100.0%
-	6	-	8372	14694	36.0	100.0%	100.0%	100.0%	100.0%
-	6	5	5224	16500	28.3	100.2%	97.8%	97.9%	107.2%
-	6	4	4900	17004	28.3	100.4%	97.4%	97.6%	107.7%
5	-	-	4257	9598	9.5	100.4%	95.6%	96.7%	105.2%
-	5	-	2676	9587	7.6	100.4%	95.6%	96.7%	105.2%
4	-	-	1105	6086	2.2	101.4%	86.8%	93.6%	109.5%

Simulations were run on 6 processors of the Beowulf cluster. Uni – maximum uniform level. Rfn – most refined grid level. Crs – lowest level grid coarsened to following refinement. Four adaptions were allowed each time step. Total Global Masses are given at 47 years, and are normalised to the mass of the level 6 uniform results, assumed to be most accurate.

In assessing the accuracy of the reactive transport simulations, the global mass balance is important. The effects of various grids on the global mass of phenol, oxygen, nitrate and iron(II) is shown in Table 4.8. The refined models are as accurate as the uniform models. The coarsened models add extra mass error relative to the uniform models. This is partly because the plume grows slightly-bigger when using the coarsened grid which means ~2% more oxygen and nitrate

are used at the fringe and also ~7% more iron(II) is produced in the enlarged plume core (Table 4.8). The plume growth is due to extra numerical dispersion on the coarser grid.

The grid adaption process causes a small mass error, see Table 4.9. For the refined only model the oxygen and nitrate had the highest adaption mass error at only 0.01%. This might be expected since they exhibit gradients at the fringe where the refinement is occurring. Conversely, for the coarsened version the adaption mass error was 0.4% for phenol, presumably due to the high phenol gradients throughout the plume core, but especially where coarsening occurred inside the fringe.

Table 4.9 Adaptive mass errors for multiple species simulations with case E

Uni.	Rfn.	Crs.	Adaptive mass error	Phenol reacted
6	-	-	-	4.1%
-	6	-	0.01% (oxygen) 0.005% (nitrate)	4.1%
-	6	5	0.4% (phenol)	4.2%

Adaptive mass error is total mass added by the grid adaption process, and is normalised to maximum value of mass. Only the highest error species are shown.

The impact of these mass errors on the overall prediction of mass of phenol degraded is slight. Table 4.9 shows that while both uniform and refined versions predict 4.1%, the coarsened version predicts 4.2%. Thus overall, the errors introduced by coarsening are not critical for this particular example. If better control over coarsening mass errors is required, then a more sophisticated interpolation scheme could be applied (e.g. Paul 2003). The mass error introduced by the actual simulation of reactions, i.e. kinetic reactions transferring mass from one component to another, was assessed by comparison of reactive to non-reactive models on uniformly refined grids. The cumulative reaction mass error over >1000 time steps was found to be 0.04%. Again, this does not significantly affect the prediction of 4.1% phenol degradation.

Overall, the accuracy of adaptive refinement is as good as a uniform grid. Adaptive coarsening introduces more error both in terms of adaption mass error and extra numerical dispersion. However, the actual impact of these errors on the predicted contaminant removal is small, at least for this example.

Multispecies, parallel adaptive refinement of a biodegrading plume is as accurate and more efficient than using a uniform grid for the example considered. Coarsening inside the plume, however, appeared to add slightly more error and the efficiency benefits were limited by the increase in the number of iterations needed by the coarsened multispecies system. It appears that coarsening may only be useful for rough preliminary simulations.

4.7.3.1 Parametric study of case E

Thus far, the mass errors considered have been related to the numerical method chosen, and not to the conceptual model itself. Many of the parameters in case E were taken from literature values, but some calibration to detailed field observations was necessary. Parameter sensitivity was studied by taking the model parameters and re-running the model with each parameter first doubled and then halved. Since a total of 46 simulations were necessary, they were run on a relatively coarse grid, level 3, which had been observed to give similar results to more refined simulations in terms of calibrated concentrations and mass turnover, but in a much shorter time. This coarse grid study was sufficient to reveal which chemical parameters were most sensitive.

Table 4.10 Sensitivity study of case E, showing selected typical results for relative changes in reacted masses for doubling of parameter values

Parameter	Reaction	Phenol	NO_3^-	H_2	FeII	MnII	SO_4^{2-}	CH_4
k_{max}	Fermentation	1.61	0.98	2.43	1.91	1.91	2.06	2.12
k_{max}	Fe(III) red'n	1.00	1.00	0.58	1.28	0.64	0.76	0.78
k_{max}	CH4 prod'n	1.00	1.00	0.84	0.91	0.92	0.78	1.49
k_{max}	SO_4^{2-} red'n	1.00	1.00	0.89	0.94	0.94	1.72	0.84
K_{M_SO4}	SO_4^{2-} red'n	1.00	1.00	1.04	1.02	1.02	0.75	1.07
K_{M_H2}	SO_4^{2-} red'n	1.00	1.00	1.07	1.03	1.03	0.56	1.11
K_{I_phenol}	All TEAPs	1.00	1.00	0.76	1.00	1.00	1.00	0.99
k_{max}	NO_3^- red'n	1.01	1.00	1.00	1.00	1.00	1.01	1.01
K_{M_NO3}	NO_3^- red'n	1.00	1.00	1.00	1.00	1.00	1.01	0.99
α_{TV}	-	1.05	1.14	1.01	1.00	1.00	1.01	1.00

In all 23 parameters were doubled. red'n – reduction; prod'n – production. Numbers are total reacted mass of species in moles normalised to the reacted mass given by the base case. For the original parameter values see (Watson et al. 2004b). All simulations were done on level 3 uniform grid which, during calibration, was observed to give similar results to more refined grids, yet saved a lot of time. Halving the parameters (not shown) typically gave similar magnitude of change but in opposite direction from doubling.

Table 4.10 shows selected results giving typical patterns which emerged from the sensitivity study in terms of the change in reacted mass of species for parameter doubling. In general the results are most sensitive to changes in the k_{max} parameter in (4.10) and less sensitive to other parameters such as K_M or K_I. The most sensitive parameter for the whole model is the k_{max} rate parameter for fermentation of phenol to hydrogen, for which parameter doubling approximately doubles the reacted mass of most species. Most of the parameters shown relate to plume core reactions, and have little effect on nitrate which reacts at the fringe. The mass of nitrate reacted is insensitive to denitrification reaction parameters, k_{max} and K_M, but is sensitive to transverse dispersivity, α_{TV}, see Table 4.10, and to a lesser extent longitudinal dispersivity. This shows that the fringe reactions are limited by dispersive mixing and not by the reaction rates, i.e. fringe reactants react immediately after dispersive mixing. This is true even on the coarse grid which includes more numerical dispersion than more highly refined grids. Overall, the plume core

processes are controlled by the fermentation rate, while the fringe processes are controlled by dispersive mixing processes.

In all cases the change in global mass closely followed the change in concentration at the observation boreholes. Thus, large mass changes were associated with large concentration changes so that the model was no longer calibrated. There were no instances where the model maintained calibrated concentrations while giving large mass changes. Therefore, the mass error associated with choice of parameter values is relatively small, since a mass error can only occur while the model concentrations remain calibrated. These model parameter mass errors are less than 1% in all cases making them slightly more significant than other mass errors considered above, for example the grid adaption error.

4.8 Conclusions

A reactive transport model has been developed based on an existing UG application in order to model relatively complex, realistic 2D and 3D biodegradation in groundwater problems. A fully coupled approach is used to solve the differential algebraic equation system resulting from mixed kinetic and equilibrium chemical reactions. Three test cases were completed in order to verify the new reactive facility in the code. The UG application compared well with published examples which made use of both split-operator and fully coupled strategies. Tools from the UG toolbox were assessed as to their efficiency and accuracy as applied to typical biodegradation problems, with a special focus on the Four Ashes, UK field site.

Parallel processing was found to give very good scaling properties for multispecies systems, even at moderately high numbers of processors. The sparse matrix storage scheme was applied and found to make savings of 95% memory and 28% runtime.

Local adaptive remeshing in conjunction with parallel processing was assessed in terms of accuracy and efficiency. Parallel adaptive remeshing was successfully applied for a realistic multispecies biodegradation problem. The adaption process gave some overheads in terms of computational time, and these were most apparent when modelling a single species in parallel, less apparent on a single processor, and unimportant for a large multispecies problem. Local adaptive refinement saved 35% of the runtime for a realistic multispecies example and was found to be nearly as accurate as using a uniform grid. In general, a pragmatic choice of 1% gradient error indicator for refinement gave results with an accuracy in terms of concentration of about 1% compared with a uniform fine grid. Local adaptive coarsening inside the plume saved an additional 20% to 30% in runtime. However, the savings were limited in the multispecies case by the requirement for more iterations on the coarsened grid. Furthermore, the accuracy of the coarsened grid solution was slightly diminished compared to the refined grid. This was due to grid adaption error as well as extra numerical dispersion resulting from using a coarser grid. However, the parametric study demonstrates that the largest area of uncertainty probably still

resides in the choice of parameter values rather than in the use of the adaptive strategy. Overall the parallel adaptive remeshing of a multispecies problem is an efficient flexible strategy where the user may opt for either a quicker, but less accurate solution, or a solution with similar accuracy to a uniform grid but taking less time. A parametric study of an example model also demonstrated that reaction rates controlled the plume core processes, and dispersive mixing controlled the plume fringe processes. This confirms that it may often be desirable to refine a model more highly at the fringe where a finer grid is more useful in limiting numerical dispersion. Additionally there may be more highly resolved field data available at the fringe which requires finer resolution to allow process interpretation.

The overall speed-up in using 64 processors with adaptive remeshing and sparse matrix storage is estimated at 150 times. Thus the combined use of efficient numerical strategies such as those available in UG helps in the development of larger, field scale, multi-species models which are necessary to understand reactive processes in biodegrading plumes in groundwater. It is argued that even with the doubling of processor speeds every 18 months, and reduction in costs for on-chip memory, realistic, detailed, 3D multi-species reactive transport simulations can only be performed on multi-processor computing platforms if solutions are required within hours rather than days or weeks. Sparse storage schemes and multigrid solvers are valuable techniques to gain efficiency, and adaptive meshing can lead to further time benefits, but it is the ability to operate on multiple processors that leads to the greatest time savings.

5 Conclusions

The research objectives outlined in chapter 1 have been met, the results having been described in chapters 2 to 4.

The numerical modelling undertaken has demonstrated that biodegradation systems at laboratory and field scale can be understood in terms of a kinetic two-step syntrophic biodegradation model. The use of Monod kinetics has provided a flexible framework which allowed the consideration of several microbial kinetic factors, including microbial growth, competitive exclusion, acclimatisation, toxic inhibition, and bioavailability of mineral electron acceptors. Such microbial kinetic factors were found to be important in understanding both laboratory and field cases; in particular, they allowed a description of intermediate species, hydrogen and acetate. In contrast, in the laboratory case, it was found that considering only the free energy of respiration processes, as in the partial equilibrium approach, was not sufficient to explain system dynamics (see Section 2.3.9).

The laboratory system model required some additional geochemical processes to explain the fate of Fe(II) created by iron reduction. Modelling results showed that precipitation and surface complexation reactions act to decrease the concentration of dissolved Fe^{2+} (see Section 2.3.6). Applying these processes to the field scale plume increased the estimated rate of iron reduction such that it became the most important electron accepting process in terms of plume mass turnover (see Section 3.3.4). A similar, but lesser, effect was also observed for manganese reduction.

In transferring the detailed microcosm model to the field scale, a plume model was created with considerably more detail than is normally applied in natural attenuation assessments, and consequently the field data was described in a more realistic manner. The use of the conceptual model with a high level of complexity appears to have been appropriate, in order to allow comparisons to the detailed laboratory and field data sets available. While the conceptual model was transferable to the field scale, the values of the rate constants were not, since the reactions are much slower in the field (see Section 3.3.2). Sensitivity studies showed that many of the Monod kinetic parameters, e.g. half saturation constants, could be taken from literature for use in both laboratory and field cases, but since the results were most sensitive to the rate constants, calibration to available data was appropriate (see Section 2.3.4. and 4.7.3) Other geochemical parameters, such as adsorption capacity could be transferred directly to the field case.

A major finding from the new detailed field model is that the plume core degradation appears to be more important than plume fringe degradation (see Section 3.3.8). This conclusion held even when considering aquifer heterogeneity, and temporal and spatial variability of the source term. Overall the use of a more detailed, realistic biodegradation model has led to a greater understanding of the modelled systems, and in the field case has illustrated that there may be

more potential for NA than was previously estimated by models with less biogeochemical complexity.

The simulation of the detailed field model was computationally expensive. The development of the reactive transport model in UG provided a simulator, which offered efficient numerical methods that are not commonly available in state of the art reactive transport codes (see Section 4.1). These methods included local adaptive remeshing, sparse matrix storage, a highly efficient multigrid solver, and parallel processing capabilities. A fully coupled approach was used to solve the differential algebraic equation system resulting from mixed kinetic and equilibrium chemical reactions (see Section 4.2). Three test cases were completed in order to verify the new reactive facility in the code. The new UG application compared well with published examples which made use of both split-operator and fully coupled strategies (see Section 4.3). The application of parallel adaptive methods to multi-species biodegradation problems is new. Results showed that parallelisation was most efficient in multi-species problems (see Sections 4.5 and 4.7.3). Adaptive remeshing was useful in tracking the plume fringe zone, without the need for a uniformly fine grid (see Section 4.7). Pragmatically chosen error indicators provided good accuracy for the adaptive mesh solutions compared with results from uniform meshes. There was an overhead associated with adapting the grid, especially in parallel, but this became relatively insignificant in multi-species simulations. Coarsening the grid inside the plume showed limited efficiency in multi-species problems, apparently due to gradients and reactions occurring in the plume core, which meant more iterations were needed when using a coarsened grid. Sparse matrix storage also provided significant gains in efficiency (see Section 4.6). The overall speed-up in using 64 processors with adaptive remeshing and sparse matrix storage is estimated to be 150 times, compared with a standard approach on a single processor. Using such a system, the run time can be reduced from weeks or months (for 3D analyses) to less than a day. Thus, the combined use of efficient numerical strategies such as those available in UG helps in the development of more realistic, field scale, multi-species models which are necessary to understand reactive processes in biodegrading plumes in groundwater, and to make improved assessments of the potential for natural attenuation.

5.1 Recommendations for future work

The work has shown that using more realistic models has led to better quantification and understanding of biodegradation systems. Therefore, it is recommended that contaminated sites should be modelled with a greater degree of biogeochemical realism than is commonly used. The particular focus should be on the temporal and spatial variation of reaction rates, and on processes which affect the distribution of Fe(II). Both these factors were important in evaluating the plume scale mass balance, but are not routinely taken into account. Such future simulations

would benefit from experimental laboratory and field studies focusing on these issues, and providing additional data to support detailed modelling studies. It is worth considering simulating both laboratory and field scale biodegradation systems, as was done in this work, demonstrating that much process understanding can be gained. Future numerical modelling studies would also benefit from inclusion of measured data on microbial populations. Simulation of further field sites would provide more parameter sets, and sensitivity studies should be used to show which parameters need to be known accurately on a site specific basis, and where literature values for parameters are appropriate. An advantage of making extensive use of numerical modelling to understand processes, and interpret data is the relatively low cost. For example, the costs involved in the research described in this thesis are <£70K, whereas the costs of collecting the field data at the Four Ashes site were ~£7M. The result of such future numerical modelling activities may be a greater capability for predictive modelling of natural attenuation at field sites.

From a numerical point of view progress may be made in the following areas. A fully coupled direct substitution approach could be developed, and may be more efficient than the differential algebraic equation approach, especially in cases where many equilibrium reactions are considered. Further, the operator splitting, sequential iterative approach could be implemented in UG, and comparisons made with fully coupled approaches in terms of efficiency and robustness in parallel, adaptive applications. It is recommended to continue using the methods of parallel processing, adaptive remeshing, and sparse matrix storage. With regard to adaptive remeshing, improvements could be made by reducing the associated computational overhead, possibly by adapting the mesh only after several time steps, and not every step, although some additional error might be expected. The code could be set up to rebalance the load on a parallel machine following grid adaptions, and the efficiency gains and overheads could be assessed with regard to biodegradation modelling. The choice of discretisation scheme may also be an area for future work. For example, the modern scheme ELLAM, which is presently being developed in UG by IWR, University of Heidelberg, is locally mass conservative, with little numerical dispersion, although fast reactions may limit the time step size, thus the effectiveness of the scheme remains to be tested for realistic biodegradation models. This work has focused on the saturated zone, but at some sites it may be necessary to model reactive transport in the unsaturated zone as well, forming a multi-phase model. It is a significant undertaking to extend the UG framework to include multi-phase reactive transport to enable simulation of the partially saturated zone. This work forms part of a proposal for future studies at the University of Sheffield.

In the future, it is likely that access to larger, faster clusters of micro-processors will increase, and simulation of field scale problems with realistic biogeochemistry, and appropriate physical heterogeneities, will become increasingly practical. Thus, it is important that appropriate

conceptual models for field scale biodegradation, as well as numerical methods, continue to be investigated.

6 References

Abrams, R. H., K. Loague, and D. B. Kent. 1998. Development and testing of a compartmentalized reaction network model for redox zones in contaminated aquifers. Water Resources Research **34**:1531-1541.

Albrechtsen, H. J., P. L. Bjerg, L. Ludvigsen, K. Rugge, and T. H. Christensen. 1999. An anaerobic field injection experiment in a landfill leachate plume, Grindsted, Denmark 2. Deduction of anaerobic (methanogenic, sulfate-, and Fe(III)-reducing) redox conditions. Water Resources Research **35**:1247-1256.

Appelo, C. A. J., and D. Postma. 1993. Geochemistry, groundwater and pollution. Balkema.

Arvin, E., and J. Flyvbjerg. 1992. Groundwater Pollution Arising from the Disposal of Creosote Waste. Journal of the Institution of Water and Environmental Management **6**:646-652.

Banwart, S. A. 1997. Aqueous speciation at the interface between geological solids and groundwater. Pages 245-287 *in* I. Grenthe and I. Puigdomenech, editors. Modelling in aquatic chemistry. Nuclear Energy Agency of the Organisation for Economic Co-operation and Development, OECD, Paris.

Banwart, S. A. 1999. Reduction of iron(III) minerals by natural organic matter in groundwater. Geochimica Et Cosmochimica Acta **63**:2919-2928.

Banwart, S. A., and S. F. Thornton. 2003. The geochemistry and hydrology of groundwater bioremediation by natural attenuation. Pages 93-138 *in* I. M. Head, I. Singleton, and M. G. Milner, editors. Bioremediation : a critical review. Horizon Scientific Press, Wymondham.

Barry, D. A., H. Prommer, C. T. Miller, P. Engesgaard, A. Brun, and C. Zheng. 2002. Modelling the fate of oxidisable organic contaminants in groundwater. Advances in Water Resources **25**:945-983.

Bastian, P. 1998. Load balancing for adaptive multigrid methods. Siam Journal on Scientific Computing **19**:1303-1321.

Bastian, P., K. Birken, K. Johannsen, S. Lang, N. Neuß, H. Rentz-Reichert, and C. Wieners. 1997a. UG - A flexible software toolbox for solving partial differential equations. Comput Visual Sci **1**:27-40.

Bastian, P., K. Birken, K. Johannsen, S. Lang, V. Reichenberger, C. Wieners, G. Wittum, and C. Wrobel. 2000. Parallel solution of partial differential equations with adaptive multigrid methods on unstructured grids. Pages 496-508 *in* High Performance Computing in Science and Engineering '99. SPRINGER-VERLAG BERLIN, Berlin.

Bastian, P., and R. Helmig. 1999. Efficient fully-coupled solution techniques for two-phase flow in porous media - Parallel multigrid solution and large scale computations. Advances in Water Resources **23**:199-216.

Bastian, P., and S. Lang. 2002. Couplex benchmark computations with UG. Computational Geosciences:accepted.

Bastian, P., S. Lang, and K. Eckstein. 1997b. Parallel adaptive multigrid methods in plane linear elasticity problems. Numerical Linear Algebra with Applications **4**:153-176.

Bekins, B. A., E. M. Godsy, and D. F. Goerlitz. 1993. Modeling Steady-State Methanogenic Degradation of Phenols in Groundwater. Journal of Contaminant Hydrology **14**:279-294.

Borden, R. C., and P. B. Bedient. 1986. Transport of Dissolved Hydrocarbons Influenced by Oxygen- Limited Biodegradation .1. Theoretical Development. Water Resources Research **22**:1973-1982.

Broholm, M. M., and E. Arvin. 2000. Biodegradation of phenols in a sandstone aquifer under aerobic conditions and mixed nitrate and iron reducing conditions. Journal of Contaminant Hydrology **44**:239-273.

Brun, A., and P. Engesgaard. 2002. Modelling of transport and biogeochemical processes in pollution plumes: literature review and model development. Journal of Hydrology **256**:211-227.

Brun, A., P. Engesgaard, T. H. Christensen, and D. Rosbjerg. 2002. Modelling of transport and biogeochemical processes in pollution plumes: Vejen landfill, Denmark. Journal of Hydrology **256**:228-247.

Carey, M. A., J. R. Finnamore, M. J. Morrey, and A. Environment. 2000. Guidance on the assessment and monitoring of natural attenuation of contaminants in groundwater.
Celia, M. A., T. F. Russell, I. Herrera, and R. E. Ewing. 1990. An Eulerian-Lagrangian Localized Adjoint Method for the Advection-Diffusion Equation. Advances in Water Resources **13**:187-206.
Chapelle, F. H. 2000. Ground-water microbiology and geochemistry, 2nd edition. J. Wiley, New York.
Chapelle, F. H., P. B. McMahon, N. M. Dubrovsky, R. F. Fujii, E. T. Oaksford, and D. A. Vroblesky. 1995. Deducing the Distribution of Terminal Electron-Accepting Processes in Hydrologically Diverse Groundwater Systems. Water Resources Research **31**:359-371.
Chilakapati, A., T. Ginn, and J. Szecsody. 1998. An analysis of complex reaction networks in groundwater modeling. Water Resources Research **34**:1767-1780.
Christensen, T. H., P. L. Bjerg, S. A. Banwart, R. Jakobsen, G. Heron, and H. J. Albrechtsen. 2000. Characterization of redox conditions in groundwater contaminant plumes. Journal of Contaminant Hydrology **45**:165-241.
Cirpka, O. A., and R. Helmig. 1997. Comparison of approaches for the coupling of chemistry to transport in groundwater systems. Pages 102-120 *in* R. Helmig, W. Jager, W. Kinzelbach, P. Knabner, and G. Wittum, editors. Notes on Numerical Dynamics 59. Vieweg Verlag Braunschweig.
Conrad, R. 1999. Contribution of hydrogen to methane production and control of hydrogen concentrations in methanogenic soils and sediments. Fems Microbiology Ecology **28**:193-202.
Curtis, G. P. 2003. Comparison of approaches for simulating reactive solute transport involving organic degradation reactions by multiple terminal electron acceptors. Computers & Geosciences **29**:319-329.
Domenico, P. A., and F. W. Schwartz. 1998. Physical and chemical hydrogeology, 2nd ed edition. Wiley, New York Chichester.
Essaid, H. I., B. A. Bekins, E. M. Godsy, E. Warren, M. J. Baedecker, and I. M. Cozzarelli. 1995. Simulation of aerobic and anaerobic biodegradation processes at a crude oil spill site. Water Resources Research **31**:3309-3327.
Fetter, C. W. 1999. Contaminant hydrogeology, 2nd edition. Prentice-Hall International (UK), Upper Saddle River, NJ London.
Frind, E. O., W. H. M. Duynisveld, O. Strebel, and J. Boettcher. 1990. Modeling of Multicomponent Transport with Microbial Transformation in Groundwater - the Fuhrberg Case. Water Resources Research **26**:1707-1719.
Haefner, F., and S. Boy. 2003. Fast transport simulation with an adaptive grid refinement. Ground Water **41**:273-279.
Hoehler, T. M., D. B. Albert, M. J. Alperin, and C. S. Martens. 1999. Acetogenesis from CO2 in an anoxic marine sediment. Limnology and Oceanography **44**:662-667.
Hoehler, T. M., M. J. Alperin, D. B. Albert, and C. S. Martens. 1998. Thermodynamic control on hydrogen concentrations in anoxic sediments. Geochimica Et Cosmochimica Acta **62**:1745-1756.
Hoehler, T. M., M. J. Alperin, D. B. Albert, and C. S. Martens. 2001. Apparent minimum free energy requirements for methanogenic Archaea and sulfate-reducing bacteria in an anoxic marine sediment. Fems Microbiology Ecology **38**:33-41.
Huang, W. 2001. The Role of Transverse Mixing of Electron Acceptors and Carbon Substrates in Natural Attenuation. PhD. Sheffield, Sheffield.
Huang, W. E., S. E. Oswald, D. N. Lerner, C. C. Smith, and C. M. Zheng. 2003. Dissolved oxygen imaging in a porous medium to investigate biodegradation in a plume with limited electron acceptor supply. Environmental Science & Technology **37**:1905-1911.
Hunter, K. S., Y. F. Wang, and P. Van Cappellen. 1998. Kinetic modeling of microbially-driven redox chemistry of subsurface environments: coupling transport, microbial metabolism and geochemistry. Journal of Hydrology **209**:53-80.
Jakobsen, R., H. J. Albrechtsen, M. Rasmussen, H. Bay, P. L. Bjerg, and T. H. Christensen. 1998. H-2 concentrations in a landfill leachate plume (Grindsted, Denmark): In situ

energetics of terminal electron acceptor processes. Environmental Science & Technology **32**:2142-2148.

Jakobsen, R., and D. Postma. 1999. Redox zoning, rates of sulfate reduction and interactions with Fe-reduction and methanogenesis in a shallow sandy aquifer, Romo, Denmark. Geochimica Et Cosmochimica Acta **63**:137-151.

Kanney, J. F., C. T. Miller, and D. A. Barry. 2003. Comparison of fully coupled approaches for approximating nonlinear transport and reaction problems. Advances in Water Resources **26**:353-372.

Keating, E. H., and J. M. Bahr. 1998. Reactive transport modeling of redox geochemistry: Approaches to chemical disequilibrium and reaction rate estimation at a site in northern Wisconsin. Water Resources Research **34**:3573-3584.

King, M. W. G., J. F. Barker, J. F. Devlin, and B. J. Butler. 1999. Migration and natural fate of a coal tar creosote plume; 2. Mass balance and biodegradation indicators. Journal of Contaminant Hydrology **39**:281-307.

Lang, S. 2000. Parallele Numerische Simulation instationärer Probleme mit adaptiven Methoden auf unstrukturierten Gittern. Universität Stuttgart.

Leeming, G. J. S., K. U. Mayer, and R. B. Simpson. 1998. Effects of chemical reactions on iterative methods for implicit time stepping. Advances in Water Resources **22**:333-347.

Lerner, D. N., S. F. Thornton, M. J. Spence, S. A. Banwart, S. H. Bottrell, J. J. Higgo, H. E. H. Mallinson, R. W. Pickup, and G. M. Williams. 2000. Ineffective natural attenuation of degradable organic compounds in a phenol-contaminated aquifer. Ground Water **38**:922-928.

Lovley, D. R. 2001. Bioremediation - Anaerobes to the rescue. Science **293**:1444-1446.

Lovley, D. R., and F. H. Chapelle. 1995. Deep Subsurface Microbial Processes. Reviews of Geophysics **33**:365-381.

Lovley, D. R., and S. Goodwin. 1988. Hydrogen Concentrations as an Indicator of the Predominant Terminal Electron-Accepting Reactions in Aquatic Sediments. Geochimica Et Cosmochimica Acta **52**:2993-3003.

MacQuarrie, K. T. B., and E. A. Sudicky. 2001. Multicomponent simulation of wastewater-derived nitrogen and carbon in shallow unconfined aquifers I. Model formulation and performance. Journal of Contaminant Hydrology **47**:53-84.

MacQuarrie, K. T. B., E. A. Sudicky, and E. O. Frind. 1990. Simulation of Biodegradable Organic Contaminants in Groundwater .1. Numerical Formulation in Principal Directions. Water Resources Research **26**:207-222.

Madigan, M. T., J. M. Martinko, J. Parker, and T. D. Brock. 2000. Brock's biology of microorganisms, 9th edition. Prentice Hall International, London.

Mansell, R. S., L. Ma, L. R. Ahuja, and S. A. Bloom. 2002. Adaptive Grid Refinement in Numerical Models for Water Flow and Chemical Transport in Soil: A Review. Vadose Zone J **1**:222-238.

Mayer, K. U. 1999. A numerical model for multicomponent reactive transport in variably saturated porous media. PhD. University of Waterloo, Waterloo, ON, Canada.

Mayer, K. U., S. G. Benner, E. O. Frind, S. F. Thornton, and D. N. Lerner. 2001. Reactive transport modeling of processes controlling the distribution and natural attenuation of phenolic compounds in a deep sandstone aquifer. Journal of Contaminant Hydrology **53**:341-368.

Mayer, K. U., E. O. Frind, and D. W. Blowes. 2002. Multicomponent reactive transport modeling in variably saturated porous media using a generalized formulation for kinetically controlled reactions. Water Resources Research **38**:1174-1194.

McNab, W. W., and T. N. Narasimhan. 1994. Modeling Reactive Transport of Organic-Compounds in Groundwater Using a Partial Redox Disequilibrium Approach. Water Resources Research **30**:2619-2635.

Molson, J. W., J. F. Barker, E. O. Frind, and M. Schirmer. 2002. Modeling the impact of ethanol on the persistence of benzene in gasoline-contaminated groundwater. Water Resources Research **38**:art. no.-1003.

Molz, F. J., M. A. Widdowson, and L. D. Benefield. 1986. Simulation of Microbial-Growth Dynamics Coupled to Nutrient and Oxygen-Transport in Porous-Media. Water Resources Research **22**:1207-1216.

Murphy, E. M., and T. R. Ginn. 2000. Modeling microbial processes in porous media. Hydrogeology Journal **8**:142-158.

Neuss, N. 2002. A new sparse-matrix storage method for adaptively solving large systems of reaction-diffusion-transport equations. Computing **68**:19-36.

Nielsen, P. H., H. J. Albrechtsen, G. Heron, and T. H. Christensen. 1995. In-Situ and Laboratory Studies On the Fate of Specific Organic- Compounds in an Anaerobic Landfill Leachate Plume .1. Experimental Conditions and Fate of Phenolic-Compounds. Journal of Contaminant Hydrology **20**:27-50.

Nordstrom, D. K., and J. L. Munoz. 1994. Geochemical thermodynamics, 2nd edition. Blackwell Scientific, Boston.

Page, G. W. 1997. Contaminated sites and environmental cleanup : international approaches to prevention, remediation, and reuse. Academic Press, San Diego, Calif.; London.

Parkhurst, D. L., and C. A. J. Appelo. 1999. User's guide to PHREEQC (version 2)- A computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations. Water-Resources Investigations Report 99-4259, USGS, Denver, Colorado.

Paul, M. 2003. Simulation of Two-Phase Flow Processes in Heterogeneous Porous Media with Adaptive Methods. Universität Stuttgart.

Pickup, R. W., G. Rhodes, M. L. Alamillo, H. E. H. Mallinson, S. F. Thornton, and D. N. Lerner. 2001. Microbiological analysis of multi-level borehole samples from a contaminated groundwater system. Journal of Contaminant Hydrology **53**:269-284.

Prommer, H., D. A. Barry, and G. B. Davis. 1999a. A one-dimensional reactive multi-component transport model for biodegradation of petroleum hydrocarbons in groundwater. Environmental Modelling & Software **14**:213-223.

Prommer, H., D. A. Barry, and G. B. Davis. 2002. Modelling of physical and reactive processes during biodegradation of a hydrocarbon plume under transient groundwater flow conditions. Journal of Contaminant Hydrology **59**:113-131.

Prommer, H., D. A. Barry, and C. Zheng. 2003. MODFLOW/MT3DMS-based reactive multicomponent transport modeling. Ground Water **41**:247-257.

Prommer, H., G. B. Davis, and D. A. Barry. 1999b. Geochemical changes during biodegradation of petroleum hydrocarbons: field investigations and biogeochemical modelling. Organic Geochemistry **30**:423-435.

Roden, E. E., and J. M. Zachara. 1996. Microbial reduction of crystalline iron(III) oxides: Influence of oxide surface area and potential for cell growth. Environmental Science & Technology **30**:1618-1628.

Saaltink, M. W., C. Ayora, and J. Carrera. 1998. A mathematical formulation for reactive transport that eliminates mineral concentrations. Water Resources Research **34**:1649-1656.

Saaltink, M. W., J. Carrera, and C. Ayora. 2000. A comparison of two approaches for reactive transport modelling. Journal of Geochemical Exploration **69**:97-101.

Saaltink, M. W., J. Carrera, and C. Ayora. 2001. On the behavior of approaches to simulate reactive transport. Journal of Contaminant Hydrology **48**:213-235.

Salanitro, J. P., and L. A. Diaz. 1995. Anaerobic Biodegradability Testing of Surfactants. Chemosphere **30**:813-830.

Salvage, K. M., and G.-T. Yeh. 1998. Development and application of a numerical model of kinetic and equilibrium microbiological and geochemical reactions (BIOKEMOD). Journal of Hydrology **209**:27-52.

Schäfer, D., W. Schäfer, and W. Kinzelbach. 1998. Simulation of reactive processes related to biodegradation in aquifers - 1. Structure of the three-dimensional reactive transport model. Journal of Contaminant Hydrology **31**:167-186.

Schäfer, W. 2001. Predicting natural attenuation of xylene in groundwater using a numerical model. Journal of Contaminant Hydrology **52**:57-83.

Schafer, W., and R. Therrien. 1995. Simulating transport and removal of xylene during remediation of a sandy aquifer. Journal of Contaminant Hydrology **19**:205-236.

Schirmer, M., B. J. Butler, J. W. Roy, E. O. Frind, and J. F. Barker. 1999. A relative-least-squares technique to determine unique Monod kinetic parameters of BTEX compounds using batch experiments. Journal of Contaminant Hydrology **37**:69-86.

Shu, C.-W., B. Cockburn, and G. Karniadakis. 2000. Discontinuous Galerkin methods : theory, computation, and applications. Springer, Berlin ; London.

Smets, B. F., and P. Pritchard. 2003. Elucidating the microbial component of natural attenuation. Current Opinion in Biotechnology **14**:283-288.

Spence, M. J., S. H. Bottrell, S. F. Thornton, and D. N. Lerner. 2001. Isotopic modelling of the significance of bacterial sulphate reduction for phenol attenuation in a contaminated aquifer. Journal of Contaminant Hydrology **53**:285-304.

Steefel, C. I., and K. T. B. MacQuarrie. 1996. Approaches to modeling of reactive transport in porous media. Pages 83-130 *in* P. Lichtner, C. I. Steefel, and E. H. Oelkers, editors. Reactive transport in porous media. Mineralogical Society of America, Washington, DC.

Stumm, W., and J. J. Morgan. 1996. Aquatic chemistry : chemical equilibria and rates in natural waters, 3rd edition. Wiley, New York Chichester.

Tebes-Stevens, C., A. J. Valocchi, J. M. VanBriesen, and B. E. Rittmann. 1998. Multicomponent transport with coupled geochemical and microbiological reactions: model description and example simulations. Journal of Hydrology **209**:8-26.

Thornton, S. F., D. N. Lerner, and S. A. Banwart. 2001a. Assessing the natural attenuation of organic contaminants in aquifers using plume-scale electron and carbon balances: model development with analysis of uncertainty and parameter sensitivity. Journal of Contaminant Hydrology **53**:199-232.

Thornton, S. F., S. Quigley, M. J. Spence, S. A. Banwart, S. Bottrell, and D. N. Lerner. 2001b. Processes controlling the distribution and natural attenuation of dissolved phenolic compounds in a deep sandstone aquifer. Journal of Contaminant Hydrology **53**:233-267.

Valocchi, A. J., and M. Malmstead. 1992. Accuracy of Operator Splitting for Advection-Dispersion- Reaction Problems. Water Resources Research **28**:1471-1476.

van Breukelen, B. M., C. A. J. Appelo, and T. N. Olsthoorn. 1998. Hydrogeochemical transport modeling of 24 years of Rhine water infiltration in the dunes of the Amsterdam Water Supply. Journal of Hydrology **209**:281-296.

Vroblesky, D. A., P. M. Bradley, and F. H. Chapelle. 1997. Lack of correlation between organic acid concentrations and predominant electron-accepting processes in a contaminated aquifer. Environmental Science & Technology **31**:1416-1418.

Wagner, C., W. Schäfer, W. Jäger, G. Wittum, and E. Kohlmeier. 2002. Simulating a Field Case of Reactive Chloroethene Transport. Computing and Visualization in Science **5**:149 - 163.

Watson, I. A., P. Bastian, R. S. Crouch, and S. E. Oswald. 2003a. Multicomponent simulation of biodegradation applying locally adaptive remeshing and parallel processing. Pages 291-295 *in* E. Poeter, C. Zheng, M. C. Hill, and J. Doherty, editors. MODFLOW and more 2003. Understanding through modeling. International groundwater modeling center, Denver, Colorado, US.

Watson, I. A., S. E. Oswald, R. S. Crouch, and P. Bastian. 2004a. Advantages of using locally adaptive remeshing and parallel processing in modelling biodegradation in groundwater. Advances in Water Resources:submitted.

Watson, I. A., S. E. Oswald, R. S. Crouch, S. F. Thornton, and S. A. Banwart. 2004b. Insights gained through multispecies field scale reactive transport modelling driven by biodegradation complexity. Water Resources Research:submitted.

Watson, I. A., S. E. Oswald, K. U. Mayer, Y. X. Wu, and S. A. Banwart. 2003b. Modeling kinetic processes controlling hydrogen and acetate concentrations in an aquifer-derived microcosm. Environmental Science & Technology **37**:3910-3919.

Wesseling, P. 1992. An introduction to multigrid methods. Wiley, Chichester.

Williams, G. M., R. W. Pickup, S. F. Thornton, D. N. Lerner, H. E. H. Mallinson, Y. Moore, and C. White. 2001. Biogeochemical characterisation of a coal tar distillate plume. Journal of Contaminant Hydrology **53**:175-197.

Wolfsberg, A. V., and D. L. Freyberg. 1994. Efficient Simulation of Single-Species and Multispecies Transport in Groundwater With Local Adaptive-Grid Refinement. Water Resources Research **30**:2979-2991.

Wu, Y. 2002. Assessing the role of fermentation in intrinsic biodegradation of organic pollutants in groundwater. PhD. University of Sheffield, Sheffield.

Yeh, G. T., and V. S. Tripathi. 1989. A Critical-Evaluation of Recent Developments in Hydrogeochemical Transport Models of Reactive Multichemical Components. Water Resources Research **25**:93-108.

Zehnder, A. J. B. 1988. Biology of Anaerobic Microorganisms. Wiley.

Zheng, C., and G. D. Bennett. 1995. Applied contaminant transport modeling : theory and practice. Van Nostrand Reinhold, New York London.

7 Appendix A

Reproduction of conference paper containing extra modelling results, in addition to those presented in chapter 2.

Watson, I. A., S. Oswald, and S. A. Banwart. 2002. Modelling natural attenuation processes of phenol degradation in groundwater. Pages 827-834 *in* Computational Methods in Water Resources, Vols 1 and 2, Proceedings. Elsevier Science Bv, Amsterdam.

Modelling Natural Attenuation Processes of Phenol Degradation in Groundwater.

I. A. Watson, S. Oswald, S. A. Banwart.

Abstract

The ability to model subsurface natural attenuation (NA) processes at the field scale is desirable in order to facilitate understanding and management of contaminated sites. These field scale reactive transport models form numerically large systems, so it is important to use the simplest conceptual models for the reactive part. Currently, conceptual models of varying complexity are used. This work demonstrates that a full understanding of NA processes requires a very complex conceptual model. This is achieved by numerical modelling of a batch microcosm, in which phenol is degraded by a range of anaerobic processes typical of redox plumes. Microcosm carbon balance, redox balance, and temporal sequence of reactions were simulated. The conceptual model included aqueous and gaseous speciation, biomass growth using Monod kinetics, surface complexation, and precipitation. The redox reactions included fermentation to hydrogen and acetate intermediates, denitrification, manganese, iron and sulphate reduction, and methanogenesis. Field scale conceptual models which include transport processes as well are likely to be even more complex. The batch simulation offers the user a tool to assess what simplifications can be justified for the field scale conceptual model.

7.1 Introduction

Natural attenuation (NA) of subsurface contaminants can be achieved by a variety of different processes including biodegradation, sorption and precipitation (Lerner et al. 2000). The ability to model these processes and simulate full scale field sites is desirable in order to facilitate understanding and management of contaminated sites. Much progress has been made in formulating mathematical and numerical models which aim to reproduce both transport and reactions of species in groundwater (see review in Steefel and MacQuarrie 1996). Today, a variety of numerical algorithms exist which have different levels of efficiency. However, it will always be the case that the least demanding numerical simulation is the one which uses the simplest, most concise conceptual model. A concise conceptual model is considered here to be the one which uses the least number of chemical species and reactions, in short, the least complexity, but which still allows the modelling objectives to be fulfilled.

Biodegradation is a good example of where several levels of conceptual complexity are possible. Perhaps the simplest are zero and first order degradation expressions (e.g. Frind et al. 1990). Also conceptually simple is instantaneous biodegradation, where any mixing of contaminant electron donor with terminal electron acceptor (TEA) results in instant completion of the redox reaction (Borden and Bedient 1986). None of these approaches account for the biomass which is responsible for the degradation. To account for this Monod kinetics can be implemented. A typical use of Monod kinetics is the Double Monod formulation where the rate is limited by the amounts of electron donor and TEA present, and is proportional to the biomass concentration, through two Monod terms and a first order term respectively (e.g. MacQuarrie et al. 1990). In fact, Monod kinetics is very flexible: The biomass term may be omitted, in which case a steady state population is assumed (e.g. Mayer et al. 2001). Terms for biomass decay, for maximum biomass concentration, or inhibition terms may be included or omitted (e.g. Schäfer et al. 1998). Biofilm models consider the biomass as a separate phase which receives contaminants by mass transfer from the mobile porewater (e.g. Molz et al. 1986). More recently it has been suggested that models should account for degradation by both attached and suspended populations, and the exposure time to the contaminant (Murphy and Ginn 2000). Overall, it can be seen that it is possible to choose quite a complex conceptual model for biodegradation.

It is not always possible to determine in advance the level of sophistication required for a conceptual model. In addition, computational restrictions will significantly limit the complexity in two and three dimensions. Take, for example, the plume of phenolic compounds at Four Ashes, UK (Lerner et al. 2000) which is 500m long, 20m thick and has a sharp fringe less than 2m thick. One dimensional flow line modelling was able to include considerable detail of the internal plume geochemistry. However, in three dimensions the system size had to be significantly reduced to only one redox reaction (Mayer et al. 2001). The relatively poor geochemical 'resolution' of the model at the plume fringe meant that the insight which could be gained on the natural attenuation processes occurring in this vital, active part of the plume was limited.

Assuming that the ultimate aim of a hypothetical reactive transport model is to be able to reproduce the key processes at the Four Ashes plume fringe, it is necessary to know what conceptual simplifications can be made. What is the simplest description of biodegradation that can be used? Do intermediate species, hydrogen and acetate, have an important role? To what extent are aqueous complexation, sorption, precipitation and degassing required?

This work aims to demonstrate the level of complexity which might be required to allow plume fringe simulation. This will be achieved through developing the conceptual model for a laboratory microcosm derived from the Four Ashes site, and testing it with a numerical model. Although the microcosm has no flow the system of chemical reactions are quite complicated. Therefore, this work has implications for reactive modelling of field cases, but not for transport modelling. The principal modelling objectives are to simulate the carbon balance, the TEA balance, and the time lags of the terminal electron accepting processes (TEAPs), and to achieve this using concepts which are currently applied in reactive transport modelling. As far as the authors are aware, this is the first time such a detailed numerical model has been applied to laboratory data based on biodegradation processes in groundwater.

7.2 Method and Modelling Procedures

The chosen dataset was collected from a 600 day long microcosm experiment which comprised anaerobic phenol degradation in the presence of several TEAs with hydrogen and acetate intermediates (Wu, in preparation). Anaerobic sediment from the Four Ashes field site provided the source of bacteria, as well as iron and manganese bearing minerals. The TEAPs were denitrification, sulphate reduction, iron reduction, manganese reduction, and methanogenesis from both acetate and hydrogen. Hydrogen and acetate are assumed to be produced by fermentation of the phenol.

Simulating the system carbon balance was a central objective of the work. The carbon initially present in the microcosm as phenol was assumed to be transformed to aqueous and gaseous products, and biomass. Of these, the aqueous and gaseous species were analysed throughout the course of the experiment. Thus, a significant amount of carbon was lost from the microcosm through sampling activities, and the headspace in the reactor grew with time. Biomass was not measured, but was estimated from the difference between the initial and final measured carbon, allowing for the sampling loss. Therefore, biomass was determined to be an important part of the system carbon balance.

Since the work aims to apply conceptual models which are currently used in field scale models, numerical models capable of simulating Monod type kinetic reactions in three dimensions were considered. Therefore, two reactive transport models were considered for the modelling: TBC (Transport, Biochemistry, Chemistry), a sequential iterative model capable of biofilm population modelling (Schäfer et al. 1998); and MIN3P, a globally implicit model capable of saturated and unsaturated reactive transport modelling (Mayer 1999). Conceptually, both biomass and unsaturated zone were important considerations for simulating the microcosm's biomass growth and headspace development respectively. A simple restriction in this case was that the smallest grid in MIN3P comprises three nodes, while TBC cannot run with less than twenty seven nodes. As a result MIN3P solved the complex multicomponent system quicker than TBC. Further, MIN3P has previously been applied to the Four Ashes field site (Mayer et

al. 2001). Therefore, MIN3P was chosen for the main part of the modelling, and TBC was used for comparison only (results not presented here).

The MIN3P model of the batch experiment was set up as a one dimensional unsaturated model with a constant outflow in order to reproduce the volumes of water lost via sampling, and the consequent increase of the gaseous headspace. The model was restarted at 470 days to allow for a spike of phenol which was added to the microcosm at this time.

The redox reactions were implemented using the Monod kinetic formulation available in MIN3P.

$$r_S = \frac{dS}{dt} = -k_{\max} \cdot X \cdot \frac{S}{(K_{M\_S} + S)} \cdot \frac{K_I}{(K_I + I)} \quad (1)$$

r_S is the rate of consumption of substrate S, [mol l^{-1} s^{-1}]; S is the substrate concentration [mol l^{-1}]; k_{max} is the maximum substrate utilisation rate [mol l^{-1} s^{-1}]; X is the biomass concentration [mol l^{-1}]; K_{M_S} is the Monod half saturation constant with respect to substrate S [mol l^{-1}]; K_I is the inhibition constant [mol l^{-1}]; I is the concentration of an inhibiting substance [mol l^{-1}]. The model actually allows multiple Monod and inhibition terms to be included multiplicatively in expression (1).

The biomass term, X, is considered to be stoichiometrically CH_2O and is included as a product in the redox reactions which are stoichiometrically balanced, according to the molar biomass yield. The resulting reactions were also checked to ensure they were thermodynamically consistent, i.e. an overall negative Gibbs free energy was produced. For reactions where hydrogen was the electron donor, significant production of biomass led to a positive production of free energy, and is therefore not possible. A steady state assumption of constant biomass concentration was made for these autotrophic reactions and biomass terms were not included in the mass balance or rate equations. Table 1 presents the equation and parameter set which allowed the best model fit to the experimental data.

Both the growth and steady state reactions are capable of simulating time lags. The steady state reactions can be delayed through use of the inhibition terms, for example hydrogenotrophic methanogenesis is inhibited by the presence of iron(III), and is therefore delayed until sufficient iron reduction has occurred. Growth reactions can also be delayed using inhibition kinetics. However, in this simulation growth reactions are primarily delayed by setting a small initial biomass such that the reaction rate will not be significant until sufficient time has passed that the newly grown biomass causes a faster rate.

The growth reactions do not include terms for the decay of bacteria, nor is there any upper limit placed on the biomass, since these facilities were not required for this batch simulation. The biomass species and the iron(III) and manganese(IV) minerals were actually simulated as aqueous phases for ease of implementation of the kinetic reactions. Clearly these items would require more attention in a transport model.

In broad terms two conceptual models of the reaction paths were considered. The first, and simpler approach was based on the field scale modelling previously completed with MIN3P (Mayer et al. 2001). It comprises a mainly one step reaction path where phenol is mineralised by each of the TEAPs. In addition, phenol is fermented to acetate and hydrogen which are in turn converted to methane. The second, more sophisticated, approach comprises a two step reaction path where phenol is first fermented to hydrogen and acetate. Subsequently, these two substrates act as electron donors for the TEAPs, as well as for methanogenesis. The results presented here are a hybrid of the two approaches, but closer to the first of the two.

Determining the quantity of iron reduction which has occurred in a system is often problematic (Christensen et al. 2000), since the measured amount of aqueous iron does not include iron which has precipitated or sorbed. In this system, it is likely that FeS precipitates rapidly following sulphate and iron reduction. This is included in the model. It is possible that $FeCO_{3(s)}$ and $MnCO_{3(s)}$ also precipitate, but as there is no specific experimental evidence for this these phases are not considered. MIN3P does, however, calculate the saturation indices of these

minerals, without including them explicitly. Sorption of iron onto the aquifer material was simulated using surface complexation reactions. As well as limiting the amount of iron in solution, the surface complexation reactions acted as a pH buffer for the simulated system.

Table 1. Stoichiometries and parameters for kinetic redox reactions

	Reaction Stoichiometry	Yield
1.	$C_6H_6O + 3.68NO_3^- + 1.16H_2O \rightarrow 3.6CO_3^{2-} + 2.4CH_2O_n + 1.84N_2 + 3.52H^+$	40%
2.	$C_6H_6O + 5H_2O \rightarrow 2.4CH_3COOH + 1.2CH_2O_f + 2H_2$	20%
3.	$C_6H_6O + 2.3SO_4^{2-} + 3H_2O \rightarrow 3.6CO_3^{2-} + 2.4CH_2O_s + 2.3HS^- + 4.9H^+$	40%
4.	$CH_3COOH + 1.6MnO_2 + 1.6H^+ \rightarrow 0.8CO_3^{2-} + 1.2CH_2O_i + 1.6Mn^{2+} + 1.6H_2O$	60%
5.	$CH_3COOH + 3.2FeOOH + 4.8H^+ \rightarrow 0.8CO_3^{2-} + 1.2CH_2O_i + 3.2Fe^{2+} + 4.8H_2O$	60%
6.	$CH_3COOH + 0.6H_2O \rightarrow 0.6CO_3^{2-} + 0.8CH_2O_m + 0.6CH_4 + 1.2H^+$	40%
7.	$H_2 + 2FeOOH + 4H^+ \rightarrow 4H_2O + 2Fe^{2+}$	-
8.	$H_2 + 0.25CO_3^{2-} + 0.5H^+ \rightarrow 0.75H_2O + 0.25CH_4$	-

Reaction rate terms and parameters

	k_{max} [s^{-1}]	X [mol L^{-1}]	Half saturation constants S, $K_{M\ S}$ [mol L^{-1}]		Inhibition I, K_I [mol L^{-1}]	
1.	$3.0x10^{-7}$	CH_2O_n	NO_3^- $8.1x10^{-6}$	C_6H_6O $1.1x10^{-4}$	-	
2.	$3.2x10^{-8}$	CH_2O_f	-	C_6H_6O $1.1x10^{-4}$	NO_3^- $1.6x10^{-5}$	
3.	$3.5x10^{-7}$	CH_2O_s	SO_4^{2-} $1.6x10^{-4}$	C_6H_6O $1.1x10^{-4}$	NO_3^- $1.6x10^{-5}$	
4.	$7.0x10^{-8}$	CH_2O_i	-	CH_3COOH $5.0x10^{-4}$	NO_3^- $1.6x10^{-5}$	
5.	$2.1x10^{-7}$	CH_2O_i	-	CH_3COOH $5.0x10^{-4}$	NO_3^- $1.6x10^{-5}$	
6.	$8.0x10^{-7}$	CH_2O_m	-	CH_3COOH $5.0x10^{-4}$	NO_3^- $1.6x10^{-5}$	SO_4^{2-} $1.6x10^{-5}$
7.	$1.0x10^{-8}$	-	-	H_2 $5.0x10^{-7}$	-	
8.	$1.0x10^{-7}$	-	-	H_2 $5.0x10^{-6}$	FeOOH $1.0x10^{-5}$	

Using all the concepts above the model was calibrated manually in a step wise fashion starting by adjusting the parameters of the earliest reactions. The calibration parameters for the redox reactions were the biomass yield, the initial biomass, and the maximum rate coefficient. Since the rate coefficient and Monod parameters are closely linked there is no need to adjust both, so Monod parameters were generally kept the same as for the original field simulation (Mayer et al. 2001).

7.3 Results

Figure 1 compares the model results to the data and illustrates the quality of the fit. Overall, the fit to the data is considered to be very good. The primary electron donor, phenol, and nitrate and sulphate TEAs match well, as do the end products: TIC, methane, iron and manganese. Of the intermediate species, acetate follows the same broad pattern as the experimental data, but the simulation of the hydrogen fails to capture the experimental peaks in concentration. With respect to the carbon balance, the simulation shows that the initial mass of carbon in phenol is converted to approximately 30% aqueous products, 5% gaseous products, 40% sampling loss, and 25% biomass.

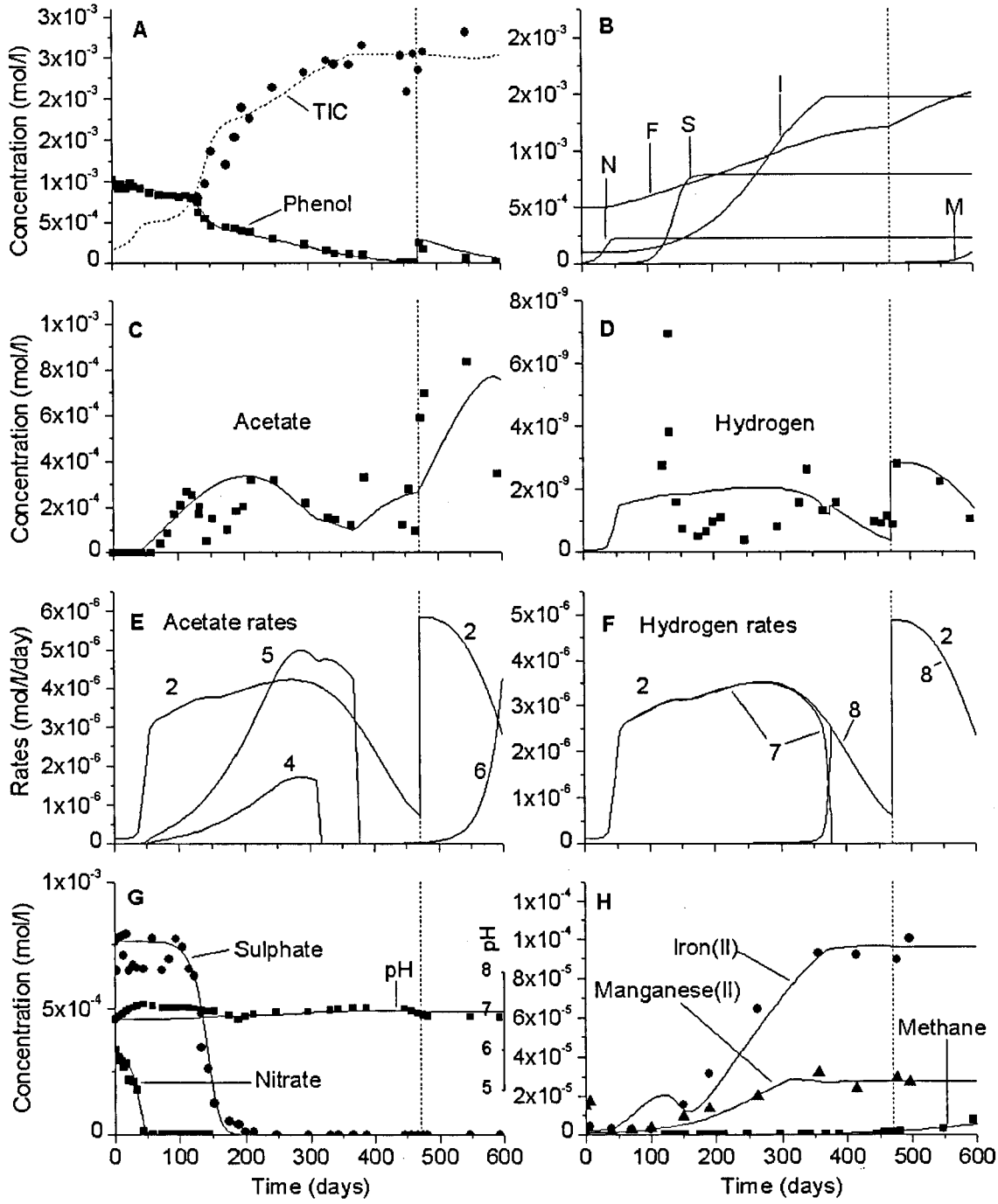

Figure 1. Results of microcosm simulation. Points are experimental data, lines are simulated with MIN3P. The vertical line at 470 days represents the phenol spike. Plot A: TIC – Total Inorganic Carbon; Numbers refer to reaction numbers in Table 1. Plot B: F – fermenters, N – denitrifiers, S - sulphate reducers, M – methanogens, I – iron and manganese reducers. Plots E, F: numbers refer to reactions in Table 1, reaction 2 shows production rates, others are consumption rates.

Figure 1B shows the simulated biomass population growth. As well as forming a significant part of the overall carbon balance, it controls the timing of the reactions. The simulated sequence of growth is roughly: denitrifiers, fermenters, sulphate reducers and iron reducers, and lastly methanogens.

Figure 1E shows the rate of production and destruction of acetate and can be used to explain the form of modelled evolution of acetate. At first the reactions are inhibited by the presence of nitrate, but once nitrate is consumed (by reaction 1) fermentation of phenol begins to produce acetate (reaction 2). Shortly afterwards iron reducers begin to grow on the newly produced acetate by reducing both iron and manganese (reactions 4 and 5). Since the iron reducers grow

quickly, the acetate consumption rate increases and after 200 days the acetate concentration begins to fall. At about 250 days the acetate production rate begins to fall since the phenol concentration has decreased to a level which limits the fermentation rate. This is simulated with the Monod term in reaction 2. The simulation provides only a finite reservoir of manganese and iron oxides as TEAs, and by 375 days these are exhausted. With no significant consumption of the acetate the concentration rises again. The addition of a spike of phenol at 470 days sharply increases the rate of production and the concentration of acetate. After the spike the growth of the acetoclastic methanogens (reaction 6) allows them to consume acetate increasingly rapidly. This results in the final decrease of acetate concentration.

The TEA balance clearly shows that a significant amount of the carbon substrate is degraded by iron and manganese. In the model the total mass reacted is $4.2x10^{-3}$ mol iron as goethite (FeOOH) and $3.6x10^{-4}$ mol manganese as pyrolusite (MnO_2). Expressing these as volume fractions of the solid aquifer material present gives $1.1x10^{-3}$ goethite and $8.3x10^{-5}$ pyrolusite. The corresponding values for the aquifer are $5.2x10^{-3}$ and $4.2x10^{-5}$ respectively (Thornton et al. 2001b), which corresonds well with the results. Interestingly, the field results indicate that very little of the total mineral oxides in the plume have been utilised. The high biomass yield used (Table 1) means that the bacteria are simulated to be very efficient in their use of TEAs – a lower yield would lead to a higher TEA requirement. Since the actual mass of the minerals is approximately known, and the TEAs required to degrade the substrate can be estimated, the biomass yield has been constrained.

Modelled dissolved iron(II) and manganese(II) are prevented from rising too high due to surface complexation reactions, and to a lesser extent, by rapid precipitation of iron sulphide. A specific surface area of $6.9m^2/g$ was necessary to control dissolved iron concentrations. Both dissolved iron and manganese appear to rise to a plateau at about 350 days. This is modelled as the exhaustion of the mineral phases, but MIN3P calculated both siderite ($FeCO_3$) and rhodochrosite ($MnCO_3$) to be supersaturated following this time, implying that the precipitation of these minerals is a possible explanation of the data.

7.4 Discussion & Conclusions

The results demonstrate that current conceptual models can be used to simulate a complex reactive biogeochemical system. However, the level of complexity required to achieve this is high. This has important implications for field scale reactive transport models. The fact that this level of complexity is required for a batch system suggests that a plume fringe which has a similar range of redox processes, and is affected by transport processes as well, will demand an even more sophisticated conceptual approach. To put this into context, consider that the field scale modelling at the Four Ashes site used a much simpler set of reactions, yet was unable to give sufficient resolution of the narrow plume fringe due to computational limitations (Mayer et al. 2001). It may be some time before reactive transport models are able to include the necessary biogeochemical details in full scale simulations where high resolution is needed.

This work has shown it is possible to simulate a great deal of detail for a batch reactor. In NA models this detail may not always be required, so it is useful to know where a conceptual model can be justifiably simplified. For example, if an NA model does not require intermediate species to be modelled, then these redox reactions could be omitted. If biomass is considered to be at a steady state, then there may be no need to model biomass. If iron reduction is not significant, then surface complexation and precipitation of the iron(II) product are not needed. Therefore, the batch simulation allows the user to explore which simplifications are appropriate for a given scenario.

8 Appendix B

Reproduction of conference paper containing extra modelling results, in addition to those presented in chapter 4.

Watson, I. A., P. Bastian, R. S. Crouch, and S. E. Oswald. 2003. Multicomponent simulation of biodegradation applying locally adaptive remeshing and parallel processing. Pages 291-295 *in* E. Poeter, C. Zheng, M. C. Hill, and J. Doherty, editors. MODFLOW and more 2003. Understanding through modeling. International groundwater modeling center, Denver, Colorado, US.

Multicomponent simulation of biodegradation applying locally adaptive remeshing and parallel processing.

I.A. Watson, P. Bastian R.S. Crouch, S.E.Oswald

Abstract

Biodegrading plumes in groundwater are often typified by relatively reactive zones around the fringes and less reactive zones in the core. A high degree of refinement is required at the fringes if a model is to be of use in improving the conceptual understanding of plumes. Two strategies for dealing with the potentially high computational demands are (i) parallel processing, where the workload is shared between multiple processors, and (ii) locally adaptive remeshing, where a refined area of the grid tracks the moving plume fringes through the domain. The partial differential equation toolbox, UG (Unstructured Grids) offers advanced numerical tools including adaptive remeshing and multigrid solvers. It embraces many of these features within a parallel processing environment. This paper reports on a recent development of UG to simulate field scale reactive biogeochemistry including Monod kinetics, NAPL dissolution, mineral precipitation/dissolution and ion exchange. The non-linear multicomponent reactive transport system is solved with the globally implicit method. Test cases have been used for verification of the new capability. The paper illustrates an application to a 3D field site. It is demonstrated that both adaptive remeshing and parallel processing can improve efficiency and in turn facilitate the incorporation of a more complex set of species and reactions such that understanding of plume processes is enhanced.

8.1 INTRODUCTION

Studies on natural attenuation and enhanced biodegradation at field sites of groundwater contamination often produce detailed data sets which have the potential to improve understanding of the processes occurring (Thornton et al. 2001b). Biodegrading plumes in groundwater are often typified by relatively reactive zones around the fringes and less reactive zones in the core. This may occur, for example, when more energetic electron acceptors such as oxygen or nitrate are rapidly consumed at the plume fringes before they have time to disperse into the core of the plume. Reactive processes which do not depend on dispersive mixing then commonly dominate in the core and may include iron reduction, with the iron sourced from the aquifer solids, and fermentation and methanogenic processes. Numerical models applied to such sites should be able to include sufficient detail to allow the varying reactive transport proceses to be resolved. Therefore, a high degree of model refinement is required at the fringes, but not necessarily in the plume core. Since field sites are typically much larger than the scale of processes of interest such as the reactive fringe, this leads to computationally large problems.

Two potential strategies for dealing with the high computational demands arising from the need for more highly refined models are (i) parallel processing and (ii) locally adaptive remeshing. In parallel processing

the workload is shared between multiple processors, for example on an array of PCs. This approach has been applied to groundwater reactive transport (e.g. Schäfer et al. 1998), but is now becoming more common as clusters of PCs can now be built relatively cheaply. The speed up gained by using this form of super computer depends not only on the number of processor nodes available, but the amount of communication of processors with each other in any specific problem. Reactive transport codes have to be specially written for parallel applications, because calculations cannot be done on processors independently, opposed to a purely local reaction computation without transport.

Locally adaptive remeshing is a modelling strategy where a refined area of the grid tracks the moving plume fringes through the domain. Thus, the reactive fringe may be simulated more accurately and efficiently, while the core of the plume is simulated on a coarser grid. The algorithm requires some form of specified error indicator, for example related to high concentration gradients, which tells the model where to refine and coarsen the mesh. While the approach has been used in a wide variety of computational problems there has been relatively little reported in the field of reactive transport modelling (Mansell et al. 2002, Wagner et al. 2002).

The work presented here is based upon the partial differential equation toolbox, UG (Unstructured Grids) which offers advanced numerical tools including adaptive remeshing, multigrid solvers (Bastian et al. 1997a), sparse matrix storage, and a variety of tools for different discretisations. It embraces many of these features within a parallel processing environment. UG has been recently developed to simulate field scale reactive biogeochemistry. The paper illustrates an application to a realistic plume modeling study which has been simulated using both 2D and 3D domains. It is demonstrated that both adaptive remeshing and parallel processing can improve efficiency and the relative benefits and limitations are discussed. These techniques will facilitate the incorporation of a more complex set of species and reactions such that understanding of plume processes is enhanced.

8.2 MODEL DEVELOPMENT

An existing UG groundwater flow and transport application was modified to include chemical reactions typically used when simulating biodegrading plumes. These processes included Monod kinetics, NAPL dissolution, aqueous complexation, mineral precipitation / dissolution and ion exchange. The non-linear multicomponent reactive transport system is solved with the globally implicit method using the Newton-Raphson method. The linearised system is solved with the mutligrid solver which applies a v-cycle over grid levels 0 to 5 (level 0 being the coarsest grid and level 5 being refined 5 times to form the finest grid) using a bi-conjugate gradient stabilised linear solver with Gauss-Seidel smoothing. Thermodynamic equilibrium is solved simultaneously with the transport equations forming a differential algebraic equation (DAE) system (Yeh and Tripathi 1989). Several test cases have been used for verification of the new capability. One of these was to reproduce chemical reactions along a field scale 1D flow line described in detail by (Mayer et al. 2001), and this forms the basis of the 2D and 3D simulations presented here. In this paper the presented simulations used first order upwind vertex centred finite volume spatial discretisation, with implicit time stepping.

8.3 MULTICOMPONENT REACTIVE TRANSPORT MODELLING IN 2D

The 2D simulations presented here use the parameter set described by Mayer (Mayer et al. 2001) as Flowline I at the Four Ashes, UK site. The site is contaminated with phenolics which are relatively persistent in the consolidated sandstone aquifer. A vertical section in the centerline of the plume is simulated on a 750m long by 20m deep domain with a 10m long constant concentration line source at the inflow boundary. The flow field is uniform and dispersivity is 1m longitudinally and 4mm in the transverse vertical direction. 47 years are simulated with a maximum step size of 0.1yr. The model includes 19 chemical species: Phenol (C_6H_6O), O_2, NO_3^-, N_2, SO_4^{2-}, S^{2-}, $FeOOH_{(s)}$, Fe^{2+}, $MnO_{2(s)}$, Mn^{2+}, CH_4, H^+, OH^-, HCO_3^-, H_2CO_3, CO_3^{2-}, $FeS_{(s)}$, $FeCO_{3(s)}$, $MnCO_{3(S)}$. The inflow concentration of phenol changes after 25 years. The chemical system includes six Monod kinetic reactions for phenol consumption by the electron acceptors, and six thermodynamic equilibrium reactions, including three precipitating minerals.

8.3.1 Local Adaptive Remeshing

The local adaptive remeshing strategy was tested with respect to the benefits gained in going to an adaptively refined mesh from a coarser mesh. The benefits are measured in terms of accuracy and efficiency. Accuracy was measured simply by assessing the longitudinal numerical

dispersion in each case. Since the example model has a high concentration of phenol and relatively low background concentration of oxygen, a small amount of phenol travelling further due to numerical dispersion reacts with oxygen and causes the oxygen front to advance significantly. The relative position of the oxygen front is therefore the most sensitive measure of the accuracy. The following results show the O_2 finge dispersion relative to the simulation considered to be most accurate. Efficiency was assessed by means of an empirical index calculated by multiplying the total degrees of freedom at the end of the simulation by the total number of linear iterations used by the linear solver within Newton's method. The index was normalised to that obtained with a uniformly fine grid.

Table 1. Summary of 2D local adaptive remeshing results

No.	Refinement criteria	Coarsening criteria	O_2 fringe dispersion	Efficiency Index
1	Fine Uniform (level 5)	None	0 m	1
2	Coarse Uniform (level 4)	None	65 m	0.18
3	10% phenol,O_2	None	4 m	0.67
4	1% phenol, O_2	None	0.1 m	0.72
5	1% phenol, O_2	0.5% phenol, O_2	15 m	1.02
6	1% phenol, O_2, Fe^{2+}	0.5% phenol, O_2, Fe^{2+}	7 m	1.08

Several simulations were run with a variety of different error indicators used. Two uniform meshes were used, refined to level 4 (64 by 16 elements) and level 5 (128 by 32 elements). The error indicators were chosen based on concentration gradients of chosen species, for example O_2 which always marks the outer limit of the plume fringe, so there should be no need to refine the mesh outside this. In Table 1 the adaption criteria are related to the maximum concentrations of the species. Thus, if the criteria is 1% O_2 the grid is refined each time step so that every cell with a difference in O_2 concentration across the cell of more than 1% of the maximum O_2 concentration is refined. Cells are refined in both x and y directions and not anisotropically depending on the direction of the gradient. Coarsening criteria function in the same way, and were applied to some of the simulations.

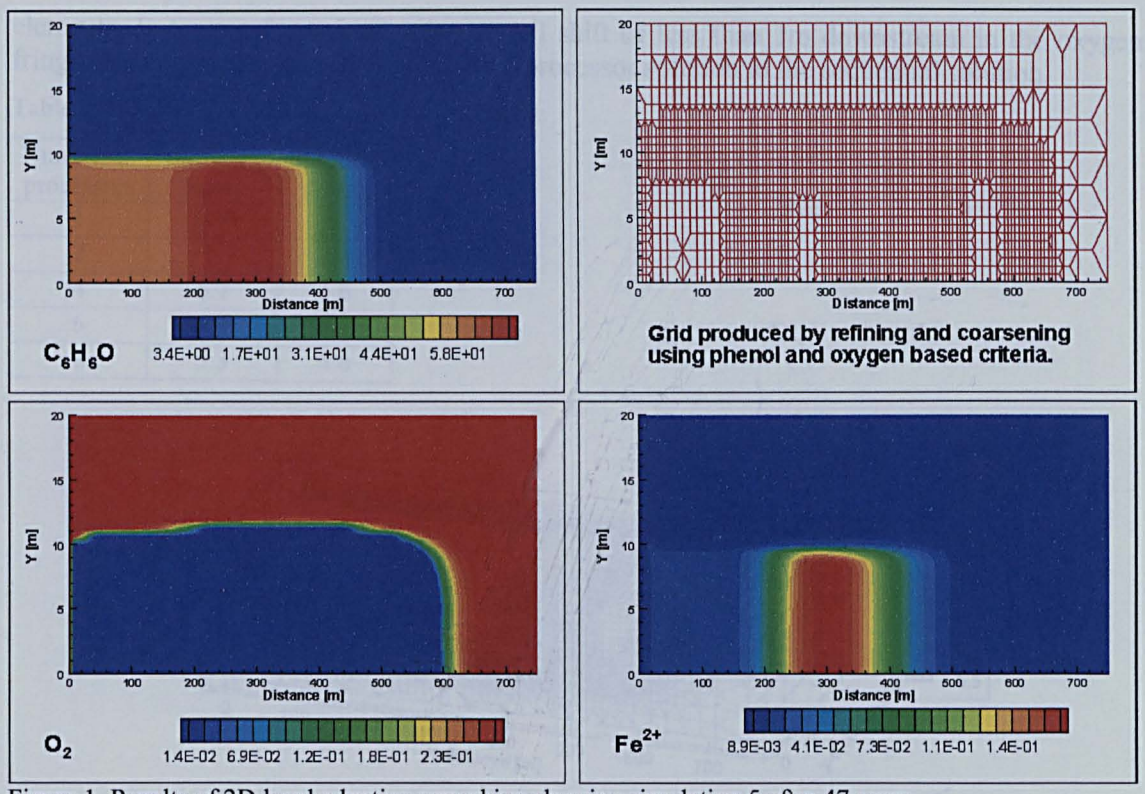

Figure 1. Results of 2D local adaptive remeshing showing simulation 5 after 47 years.

The results, see Table 1, are compared relative to the fine uniform mesh simulation no.1 which had the finest mesh and was therefore the most accurate simulation. The coarse uniform mesh simulation (no.2) shows a significant amount of numerical dispersion of the O_2 fringe relative to simulation no.1. It was found that refining the entire plume with no coarsening was more efficient than the uniform model. Tighter refinement criteria resulted in higher accuracy at the expense of efficiency, see simulations 3 and 4. Figure 1 illustrates the grid and three concentration profiles obtained in simulation 5 which allowed coarsening. Since the grid is only adapted based upon phenol and O_2 the internal plume of Fe^{2+}, produced by iron oxide reduction clearly has steep gradients in coarsened regions. This resulted in a less efficient and less accurate simulation than the uniformly fine model. Thus, in simulation 6 the adaption criteria were extended to include Fe^{2+}. The result was a slight improvement in accuracy, but the efficiency was still poor. The resulting grid actually had very little coarsening, and although it visually looked similar to that of simulation 4, it was significantly less accurate and efficient. To summarise, adaptive refinement provides a good way of improving accuracy of a simulation without the expense of going to a uniformly finer mesh. However, in this non-linear multicomponent reactive transport example even a small amount of coarsening inside the plume was found to be detrimental to the simulation.

8.3.2 Parallel Processing

The same example problem as above was run on a Beowulf cluster consisting of four dual processor Athlon 1800 machines. UG is written to be compiled as either sequential or parallel code, and in this case the code was compiled with message passing interface, MPI. UG implements a load balancing algorithm to minimise the area of the processor boundaries. The example run was the same as simulation 1, Table 1, i.e. a fine uniform grid, level 5. The results show that the parallel processing scales well, although it is not linear. The scaling is slightly poorer when the maximum 7 processors were used. This may be due to the location of the processor boundaries causing an observed increase in the number of linear iterations performed. The five times speed up achieved implies that the parallel fine uniform solution can be completed in the same time as the sequential coarse uniform solution which has four times less

elements. In terms of accuracy only a small shift of less than 1m downstream in the oxygen fringe was observed when comparing the 7 processor solution to the sequential solution.

Table 2. Results of parallel runs of 2D simulation

Number of processors	Runtime, hours	Speed up factor
1	24.0	1.0
2	11.7	2.1
4	6.7	3.6
6	4.9	4.9
7	5.3	4.6

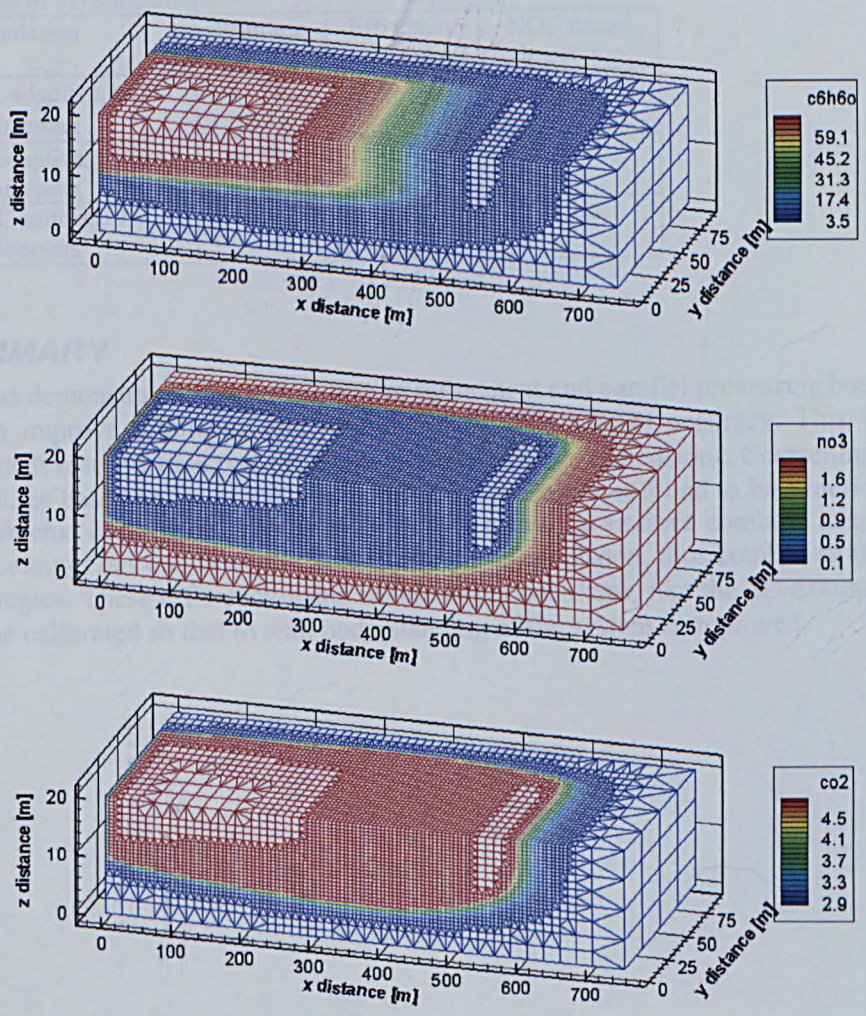

Figure 2. 3D results with adaptive remeshing

8.4 MULTICOMPONENT REACTIVE TRANSPORT MODELLING IN 3D

The 3D example is very similar to the 2D example described above, but the plume now has a 65m wide by 10m deep plane source (Mayer et al. 2001). A maximum of 32 elements across the 90m wide domain are used at grid level 5. The chemistry is simplified greatly to four species, C_6H_6O, NO_3^-, HCO_3^-, H^+, with only one Monod reaction. Note that the principle electron acceptor is now nitrate which actually represents the sum of the electron equivalents in both oxygen and nitrate. There is no change in the line source concentrations with time. Two adaptive simulations were performed, with and without coarsening, both refined using a 1%

phenol, NO_3^- criteria. These were compared to a parallel uniform model in terms of absolute runtime, efficiency index, and accuracy, measured by the position of the nitrate fringe relative to the adaptive non-coarsening results which were assumed to be the most accurate of the three simulations.

Figure 2 shows the results of the adaptive coarsened model. On the vertical section the mesh pattern is similar to the 2d case except that the lack of a step in the phenol input concentration allows the 3D model to coarsen more extensively. Despite this extra coarsening, Table 3 shows that the coarsened simulation is still less efficient and also less accurate than the non-coarsened model. The efficiency index normalised to the uniform results is 0.72 for the adaptive, non-coarsening results which is the same as for 2D simulation 4 (see Table 1). Thus for these particular examples the adaptive savings are the same in 2D in 3D. Table 3 shows that the parallel simulation ran quicker with only a marginal loss of accuracy.

Table 3. Results of 3D simulations

Simulation	Runtime	Efficiency Index	NO_3^- fringe dispersion
Sequential, adaptive, with coarsening	51.9hr	0.77	12.5m
Sequential, adaptive, without coarsening	46.6hr	0.72	0m
Parallel, uniform, 7 processors	19.3hr	1	0.5m

8.5 SUMMARY

This work has demonstrated that local adaptive refinement and parallel processing both have the capability to improve efficiency and runtimes with little loss of accuracy. This is true for mulitcomponent reactive transport problems in two and three dimensions. Coarsening the mesh inside the plume based on concentration gradient criteria was not found to be beneficial for the example problems considered. This module of UG cannot at present combine adaptivity and parallel processing, but this is expected to become available soon, thus combining the benefits of both strategies. These efficiency gains enable larger problems, like the 2D example with 19 species, to be calibrated so that overall understanding of the system is improved.

9 Appendix C

Parameters and equations used for test cases.

Table 9.1 Case A. Discretisation and hydraulic parameters

Parameter	Description	Value
X	2D domain length	0.150 m
Y	2D domain width[a]	0.040 m
Δx	Discretised cell size, uniform grid	0.75mm
Δy	Discretised cell size, uniform grid	0.625mm
T_end	Time at simulation end	500 minutes
Δt	Time step size, constant	1 minute
v	Linear pore water velocity along X	1.25×10^{-5} m/s
α_L	Longitudinal dispersivity	2.5×10^{-3} m
α_T	Transverse dispersivity	1.67×10^{-4} m
D^*	Effective pore diffusion coefficient	8.67×10^{-10} m^2/s [b]
n_{mob}	Mobile phase volume fraction	0.49 m^3 mobile phase / m^3 bulk aquifer

a only half the plume is simulated since it is symmetrical
b the D^* value is the mean of diffusion values for oxygen (1.15×10^{-9}) and acetate (5.26×10^{-10}). Note hydrodynamic dispersion, $D = \alpha v + D^*$.

Table 9.2 Case A. Chemical species

Mobile, aqueous species	Initial (background) (mg/l)	Inflow boundary condition (mg/l)
Acetate	0	At 0.0m<Y<0.004m : 1000 Elsewhere : 0
O_2	6.64	6.64

Table 9.3 Case A. Kinetic Reaction

Aerobic biodegradation of acetate (Monod kinetics) (without biomass growth)	
Reaction	Acetate + $ST.O_2$ → products[a]
ST	0.39 mg O_2 / mg Acetate
k_{max}	2.60×10^{-4} 1/s
X_{bio}	10 mg/l water
MT_1	Acetate / (K_{M_ac} + Acetate)
K_{M_ac}	2.1 mg Acetate / l water
MT_2	O_2 / (K_{M_O2} + O_2)
K_{M_O2}	0.01 mg O_2 / l water
R_k	$R_k = n_{mob}.k_{\max}.X_{bio}.MT_1.MT_2$ [mg/s/l bulk]
Source/sink: Acetate	$\frac{n_{mob}\partial Acetate}{\partial t} = -1.R_k$
Source/sink: O_2	$\frac{n_{mob}\partial O_2}{\partial t} = -ST.R_k$

a - products are not simulated.
ST is the stoichiometric coefficient, cf. s_{ik} in (9).
MT is a Monod term of general form $C/(K_M + C)$, see (10).

Table 9.4 Case B. Discretisation, hydraulic parameters and physical properties

Parameter	Description	Value
X	1D domain length	1.0 m
Δx	Discretised cell size, uniform	0.02 m
T_end	Time at simulation end	100 days
Δt	Time step size, constant	0.02 day
v	Linear pore water velocity	1 m/day
α_L	Longitudinal dispersivity	0.01 m
n_{mob}	Mobile phase volume fraction	0.25 m^3 mobile phase / m^3 bulk aquifer
n_{bio}	Biophase volume fraction	0.01 m^3 biophase / m^3 bulk aquifer
n_{mat}	Aquifer solids (matrix) volume fraction	0.74 m^3 aquifer solids / m^3 bulk aquifer
ρ_s	Density of aquifer solids	2650 kg / m^3

Table 9.5 Case B. Chemical species

Mobile, aqueous species	Initial (background) and inflow boundary condition
Tol_{mob}	0 mol OC_{tol} / m^3 water
O_2	0.313 mol / m^3 water
SO_4^{2-}	0.42 mol / m^3 water
S^{2-}	10^{-6} mol / m^3 water
Fe^{2+}	10^{-6} mol / m^3 water

Immobile species	Initial (background)
Tol_{NAPL}	At 0.1m<x<0.7m : 198 mol / m^3 solids
	Elsewhere : 0 mol / m^3 solids
$Fe(III)_s$	11.5 mol / m^3 solids
FeS_s	0 mol / m^3 solids
$Fe(III)_{bio}$	10^{-3} mol / m^3 biophase
X_{aerobe}	10^{-3} mol / m^3 biophase
$X_{sulphate}$	10^{-3} mol / m^3 biophase
X_{iron}	10^{-3} mol / m^3 biophase

OC_{tol} - toluene expressed as moles of organic carbon.
1 mol/m^3 is equivalent to 1 mmol/l.

Table 9.6 Case B. Kinetic reactions: Phase exchange

Toluene Dissolution	
Reaction	$Tol_{NAPL} \rightarrow Tol_{mob}$
R_1	$R_1 = \gamma(C_{SOL} - Tol_{mob})$ [mol/day/m^{3}bulk]
γ	0.01[a] 1/day
C_{SOL}	38 mol OC_{Tol} / m^3 water {500 mg toluene / l}
Source/sink: Tol_{NAPL}	$\frac{n_{mat}\partial Tol_{NAPL}}{\partial t} = -1.0R_1$
Source/sink: Tol_{mob}	$\frac{n_{mob}\partial Tol_{mob}}{\partial t} = 1.0R_1$

$Fe(III)_s$ dissolution to biophase	
Reaction	$Fe(III)_s \rightarrow Fe(III)_{bio}$
R_2	$R_2 = \delta(Fe(III)_s - Fe(III)_{bio})$ [mol/day/m^{3}bulk]
δ	0.002 1/day
Source/sink: $Fe(III)_s$	$\frac{n_{mat}\partial Fe(III)_s}{\partial t} = -1.0R_2$
Source/sink: $Fe(III)_{bio}$	$\frac{n_{bio}\partial Fe(III)_{bio}}{\partial t} = 1.0R_2$

a. It was found that the NAPL dissolution rate, γ, had been published incorrectly and that a value twice that published had to be used to reproduce the published concentration profiles (Schäfer et al. 1998).

Table 9.7 Case B. Aerobic growth and decay on toluene

Aerobic biomass growth on toluene (Monod kinetics)	
Reaction	$(1/Y)Tol_{mob} + ST(1\text{-}Y)(1/Y).O_2 \rightarrow X_{aerobe}$
Y	0.1 mol X_{aerobe} / mol Tol_{mob}
ST	1.29 mol O_2 / mol Tol_{mob}
R_3	$R_3 = n_{bio}.k_{max}.X_{aerobe}.MT_1.MT_2$ [mol/day/m^{3}bulk]
k_{max}	5.0 1/day
MT_1	$Tol_{mob} / (K_{M\_tol} + Tol_{mob})$
K_{M_tol}	0.15 mol Tol_{mob}/m^{3}water
MT_2	$O_2 / (K_{M\_O2} + O_2)$
K_{M_O2}	0.003 mol O_2 /m^{3}water
Source/sink: Tol_{mob}	$\frac{n_{mob}\partial Tol_{mob}}{\partial t} = -1/Y.R_3$
Source/sink: O_2	$\frac{n_{mob}\partial O_2}{\partial t} = -ST(1-Y)(1/Y).R_3$
Source/sink: X_{aerobe}	$\frac{n_{bio}\partial X_{aerobe}}{\partial t} = 1.0R_3$

Aerobic biomass decay	
Reaction	X_{aerobe} first order decay
R_4	$R_4 = n_{bio}.k_{dec}.X_{aerobe}$ [mol/day/m^{3}bulk]
k_{dec}	0.5 1/day
Source/sink: X_{aerobe}	$\frac{n_{bio}\partial X_{aerobe}}{\partial t} = -1.0R_4$

Y – Yield factor for biomass growth. ST - Stoichiometric coefficient
MT - Monod term. IT – Inhibition term

Table 9.8 Case B. Sulphate reducers growth and decay on toluene

Sulphate reducer biomass growth on toluene (Monod kinetics)	
Reaction	$(1/Y)\mathrm{Tol_{mob}} + ST(1\text{-}Y)(1/Y).\mathrm{SO_4^{2-}} \rightarrow ST(1\text{-}Y)(1/Y).\mathrm{S^{2-}} + \mathrm{X_{sulphate}}$
Y	0.02 mol $\mathrm{X_{sulphate}}$ / mol $\mathrm{Tol_{mob}}$
ST	0.643 mol $\mathrm{SO_4^{2-}}$ / mol $\mathrm{Tol_{mob}}$
R_5	$R_5 = n_{bio}.k_{\max}.X_{sulphate}.MT_1.MT_2.IT_1$ [mol/day/m^{3}bulk]
k_{max}	1.0 1/day
MT_1	$\mathrm{Tol_{mob}} / (K_{M\_tol} + \mathrm{Tol_{mob}})$
K_{M_tol}	0.15 mol $\mathrm{Tol_{mob}}$ / m^{3}water
MT_2	$\mathrm{SO_4^{2-}} / (K_{M\_SO4} + \mathrm{SO_4^{2-}})$
K_{M_SO4}	1.25 mol $\mathrm{SO_4^{2-}}$ / m^{3}water
IT_1	$\mathrm{K_{I\_O2}} / (K_{I\_O2} + \mathrm{O_2})$
K_{I_O2}	0.003 mol $\mathrm{O_2}$/m^{3}water
Source/sink: $\mathrm{Tol_{mob}}$	$\frac{n_{mob}\partial Tol_{mob}}{\partial t} = -1/Y.R_5$
Source/sink: $\mathrm{SO_4^{2-}}$	$\frac{n_{mob}\partial SO_4^{2-}}{\partial t} = -ST(1-Y)(1/Y).R_5$
Source/sink: $\mathrm{S^{2-}}$	$\frac{n_{mob}\partial S^{2-}}{\partial t} = ST(1-Y)(1/Y).R_5$
Source/sink: $\mathrm{X_{sulphate}}$	$\frac{n_{bio}\partial X_{sulphate}}{\partial t} = 1.0R_5$

Sulphate reducer biomass decay	
Reaction	$\mathrm{X_{sulphate}}$ first order decay
R_6	$R_6 = n_{bio}.k_{dec}.X_{sulphate}$ [mol/day/m^{3}bulk]
k_{dec}	0.1 1/day
Source/sink: $\mathrm{X_{sulphate}}$	$\frac{n_{bio}\partial X_{sulphate}}{\partial t} = -1.0R_6$

Table 9.9. Case B. Iron reducers growth and decay on toluene

Iron reducer biomass growth on toluene (Monod kinetics)	
Reaction	$(1/Y)Tol_{mob} + ST(1-Y)(1/Y).Fe(III)_{bio} \rightarrow ST(1-Y)(1/Y).Fe^{2+} + X_{iron}$
Y	0.03 mol X_{iron} / mol Tol_{mob}
ST	5.14 mol $Fe(III)_{bio}$ / mol Tol_{mob}
R_7	$R_7 = n_{bio}.k_{max}.X_{iron}.MT_1.MT_2.IT_1$ [mol/day/m^{3}bulk]
k_{max}	1.7 1/day
MT_1	$Tol_{mob} / (K_{M\_tol} + Tol_{mob})$
K_{M_tol}	0.15 mol Tol_{mob} / m^{3}water
MT_2	$Fe(III)_{bio} / (K_{M\_iron} + Fe(III)_{bio})$
K_{M_iron}	0.0018 mol $Fe(III)_{bio}$ / m^{3}biophase
IT_1	$K_{I\_O2} / (K_{I\_O2} + O_2)$
K_{I_O2}	0.003 mol O_2/m^{3}water
Source/sink: Tol_{mob}	$\frac{n_{mob}\partial Tol_{mob}}{\partial t} = -1/Y.R_7$
Source/sink: $Fe(III)_{bio}$	$\frac{n_{bio}\partial Fe(III)_{bio}}{\partial t} = -ST(1-Y)(1/Y).R_7$
Source/sink: Fe^{2+}	$\frac{n_{mob}\partial Fe^{2+}}{\partial t} = ST(1-Y)(1/Y).R_7$
Source/sink: X_{iron}	$\frac{n_{bio}\partial X_{iron}}{\partial t} = 1.0R_7$

Iron reducer biomass decay	
Reaction	X_{iron} first order decay
R_8	$R_8 = n_{bio}.k_{dec}.X_{iron}$ [mol/day/m^{3}bulk]
K_{dec}	0.17 1/day
Source/sink: X_{iron}	$\frac{n_{bio}\partial X_{iron}}{\partial t} = -1.0R_8$

Table 9.10 Case B. Thermodynamic equilibrium reactions

Precipitation / dissolution of iron sulphide	
Reaction	$FeS_s \Leftrightarrow Fe^{2+} + S^{2-}$ logK = -18.1
Mass action law	$a(Fe^{2+}) \times a(S^{2-}) = 10^{-18.1}$

$a(Concentration)$ - activity of species. Here the conversion uses unit activity coefficients and a factor of 10^{-3} to convert concentrations from mol/m^3 to mol/l

Table 9.11 Case C. Discretisation, hydraulic parameters and physical properties

Parameter	Description	Value
X	1D domain length	750 m
Δx	Discretised cell size, uniform level 6	5.86 m
T_end	Time at simulation end	47 years
Δt	Time step size, variable	0.1 years maximum
v	Linear pore water velocity	2.78×10^{-7} m/s
α_L	Longitudinal dispersivity	1 m
n_{mob}	Mobile phase volume fraction	0.125 m^3 mobile phase / m^3 bulk aquifer
n_{mat}	Aquifer solids (matrix) volume fraction	0.73 m^3 aquifer solids / m^3 bulk aquifer
ρ_s	Density of aquifer solids	2644 kg / m^3 solids

v is calculated from: $v = q / n_{mob}$ = 3.48 x 10^{-8} m/s / 0.125 = 8.77 m/y ; q is the Darcy flux, 1.09 m/y.

Table 9.12 Case C. Chemical species

Mobile, aqueous species	Initial (background) (mol/l water)	Source T = 0-25years (mol/l water)	Source T = 25-47 years (mol/l water)
C_6H_6O [a]	3.4×10^{-10}	7.3×10^{-2}	5.9×10^{-2}
O_2	2.9×10^{-4}	2.9×10^{-4}	2.9×10^{-4}
NO_3^-	1.7×10^{-3}	1.7×10^{-3}	1.7×10^{-3}
N_2	3.6×10^{-10}	3.6×10^{-10}	3.6×10^{-10}
Fe^{2+}	8.9×10^{-7}	8.9×10^{-7}	8.9×10^{-7}
Mn^{2+}	1.2×10^{-7}	1.2×10^{-7}	1.2×10^{-7}
SO_4^{2-}	6.7×10^{-4}	3.4×10^{-4}	1.9×10^{-4}
HS^-	3.0×10^{-10}	3.0×10^{-10}	3.0×10^{-10}
CH_4	6.2×10^{-10}	6.2×10^{-10}	6.2×10^{-10}
H^+	1.41×10^{-7} [b]	3.75×10^{-6} [c]	3.62×10^{-7} [d]
HCO_3^-	2.27×10^{-3} [e]	2.28×10^{-3} [f]	9.76×10^{-3} [g]
CO_3^{2-}	1.19×10^{-6} [e]	5.63×10^{-8} [f]	2.16×10^{-6} [g]
H_2CO_3	6.30×10^{-4} [e]	1.37×10^{-2} [f]	6.07×10^{-3} [g]
OH^-	8.91×10^{-8} [b]	3.75×10^{-9} [c]	3.62×10^{-8} [d]
Ca^{2+}	2.4×10^{-3}	2.4×10^{-3}	2.4×10^{-3}
Na^+	7.8×10^{-4}	1.2×10^{-2}	1.2×10^{-2}

Immobile species	Type	Initial (background)	Unit
$FeOOH_s$	Mineral	9.82×10^{-2} [h]	mol / kg solids
$MnO_{2\,s}$	Mineral	8.73×10^{-4} [i]	mol / kg solids
FeS_s	Mineral	1×10^{-13}	mol / kg solids
$FeCO_{3\,s}$	Mineral	1×10^{-13}	mol / kg solids
$MnCO_{3\,s}$	Mineral	1×10^{-13}	mol / kg solids
$CaCO_{3\,s}$	Mineral	1×10^{-13}	mol / kg solids
Na-X	ion exchange	0.01[j]	meq / 100g solids
Ca-X_2	ion exchange	1.49[j]	meq / 100g solids

a. Total phenolics expressed as phenol
b. pH = 6.9 c. pH = 5.5 d. pH = 6.5
e. TIC = 2.9×10^{-3} f. TIC = 1.6×10^{-2} g. TIC = 1.6×10^{-2}
h. (5.5g_Fe/kg_solids) / (56g/mol) i. (48mg_Mn/kg_solids) / (55g/mol)
j. cation exchange capacity, CEC = 1.5 meq/100g solids

Table 9.13 Case C. Kinetic reactions

Aerobic degradation of phenol (Monod kinetics)	
Reaction	$C_6H_6O + 7O_2 + 3H_2O \rightarrow 6HCO_3^- + 6H^+$
R_1	$R_1 = n_{mob}.k_{max}.MT_{c6h6o}.MT_{O2}$
k_{max}	3.98×10^{-10} mol/l water /s
$K_{M\text{-}c6h6o}$	1.064×10^{-4} mol/l water
$K_{M\text{-}O2}$	3.125×10^{-6} mol/l water

Phenol Denitrification	
Reaction	$C_6H_6O + 28NO_3^- + H_2O \rightarrow 30HCO_3^- + 2H^+ + 14N_2$
R_2	$R_2 = n_{mob}.k_{max}.MT_{c6h6o}.MT_{NO3}.IT_{O2}$
k_{max}	3.98×10^{-11} mol/l water /s
$K_{M\text{-}c6h6o}$	1.064×10^{-4} mol/l water
$K_{M\text{-}NO3}$	8.064×10^{-6} mol/l water
$I\text{-}O2$	6.2×10^{-6} mol/l water

Phenol Iron reduction	
Reaction	$C_6H_6O + 28FeOOH + 50H^+ \rightarrow 6HCO_3^- + 28Fe^{2+} + 39H_20$
R_3	$R_3 = n_{mob}.k_{max}.MT_{c6h6o}.MT_{SO4}.IT_{O2}.IT_{NO3}.IT_{c6h6o}$
k_{max}	3.16×10^{-13} mol/l water /s
$K_{M\text{-}c6h6o}$	1.064×10^{-4} mol/l water
$K_{M\text{-}SO4}$	2.400×10^{-4} mol/l water
$K_{I\text{-}O2}$	3.125×10^{-5} mol/l water
$K_{I\text{-}NO3}$	1.6×10^{-5} mol/l water
$K_{I\text{-}c6h6o}$	6.382×10^{-3} mol/l water

Phenol Manganese reduction	
Reaction	$C_6H_6O + 14MnO_2 + 22H^+ \rightarrow 6HCO_3^- + 14Mn^{2+} + 11H_20$
R_4	$R_4 = n_{mob}.k_{max}.MT_{c6h6o}.IT_{O2}.IT_{NO3}$
k_{max}	6.31×10^{-14} mol/l water /s
$K_{M\text{-}c6h6o}$	1.064×10^{-4} mol/l water
$K_{I\text{-}O2}$	3.125×10^{-5} mol/l water
$K_{I\text{-}NO3}$	1.6×10^{-5} mol/l water

Phenol Sulphate reduction	
Reaction	$C_6H_6O + 3.5SO_4^{2-} + 3H_2O \rightarrow 6HCO_3^- + 3.5HS^- + 2.5H^+$
R_5	$R_5 = n_{mob}.k_{max}.MT_{c6h6o}.MT_{SO4}.IT_{O2}.IT_{NO3}.IT_{c6h6o}$
k_{max}	1.259×10^{-12} mol/l water /s
$K_{M\text{-}c6h6o}$	1.064×10^{-4} mol/l water
$K_{M\text{-}SO4}$	1.600×10^{-4} mol/l water
$K_{I\text{-}O2}$	3.125×10^{-5} mol/l water
$K_{I\text{-}NO3}$	1.6×10^{-5} mol/l water
$K_{I\text{-}c6h6o}$	6.382×10^{-3} mol/l water

Phenol Methanogenesis	
Reaction	$C_6H_6O + \frac{13}{2} H_20 \rightarrow \frac{5}{2} HCO_3^- + \frac{7}{2} CH_4 + \frac{5}{2} H^+$
R_6	$R_6 = n_{mob}.k_{max}.MT_{c6h6o}.IT_{O2}.IT_{NO3}$
k_{max}	2.51×10^{-13} mol/l water /s
$K_{M\text{-}c6h6o}$	1.087×10^{-3} mol/l water
$K_{I\text{-}O2}$	3.125×10^{-5} mol/l water
$K_{I\text{-}NO3}$	1.6×10^{-5} mol/l water

Table 9.14 Case C. Equilibrium reactions

Aqueous complexation reactions			Log K
OH^-	H_2O	$\Leftrightarrow H^+ + OH^-$	-13.998
H_2CO_3	$H_2CO_{3(aq)}$	$\Leftrightarrow H^+ + HCO_3^-$	-6.351
CO_3^{2-}	HCO_3^-	$\Leftrightarrow H^+ + CO_3^{2-}$	-10.330
Mineral precipitation/dissolution reactions			Log K
Mackinawite	$FeS + H^+$	$\Leftrightarrow Fe^{2+} + HS^-$	-4.648
Siderite	$FeCO_3 + H^+$	$\Leftrightarrow Fe^{2+} + CO_3^{2-}$	-0.12
Rhodochrosite	$MnCO_3 + H^+$	$\Leftrightarrow Mn^{2+} + CO_3^{2-}$	-0.08
Calcite	$CaCO_3 + H^+$	$\Leftrightarrow Ca^{2+} + CO_3^{2-}$	1.855
Ion exchange reactions			K
Na\Ca exchange	$Na^+ + \frac{1}{2}\,Ca\text{-}X_2$	$\Leftrightarrow \frac{1}{2}\,Ca^{2+} + Na\text{-}X$	0.4

Activity coefficients calculated with the Davies equation.
Mineral activities are equal to unity.
Ion exchange uses the Gaines-Thomas convention.

10 Appendix D

User Guide to Biogeochemical modelling in groundwater with UG

10.1 Introduction

This user guide is intended to cover and explain the C source code and UG script files which were developed in order to complete the work described in the PhD thesis of Ian Watson. It is not intended to provide a full description of UG, but just a small part relevant to biogeochemical modelling. Readers should first familiarise themselves with UG through use of the UG Tutorial (available on the internet at UG home page) and the Course documentation (written by Peter Bastian and provided on the CDROM) which covers the various groundwater applications in the course directory.

Course includes interfaces to multiphase flow and linear/non-linear reactive transport applications. For example, the published couplex benchmark work is included in the course directory.

10.2 CDROM

The UG code used is provided on the CDROM.

Most of it is the same as the internet download version, the exception being the "course" directory which was provided by Peter Bastian.

The directories of particular interest are in

`UG/course/*`

Documentation provided on the CD includes copies of the UG Tutorial and the Course documentation.

10.3 Discretisation scheme

The course directory includes several interfaces to use a variety of discretisation schemes. These are the problem class libraries in the pclib directory

`UG/course/pclib/*`

Including for example the boxnlsys discretisation which uses the box method (standard vertex centered finite volume) for a system of unknowns (mobile or immobile chemical species) with non-linearites in the storage and source-sink terms, and the linear advection-dispersion operator. This is in

`UG/course/pclib/transport/boxnlsys.c`

We will not dwell on the problem classes here but focus on the implementation of the parameter files, and the running of the model with the UG script file. These files are the application files and are in:

`UG/course/appl/`

The UG tutorial describes how to introduce a new application directory such that it is compiled into the course executable. In general this invloves small additions to the main() function in

`UG/course/appl/course.c`

The make files in each of the directories also need changing.

Applications created by the present author were:

`UG/course/appl/equilibrium/*`
`UG/course/appl/fourash/*`
`UG/course/appl/visplume/*`
`UG/course/appl/test1d/*`
`UG/course/appl/fourash2d/*`
`UG/course/appl/fourash3d/*`

The last and most recent two contain the example application files which are described in detail below.

10.4 Compilation and running

In order to compile the applications:
For a 2D sequential processing model running UG in the UG shell do this
Set the environment variables PATH and UGROOT as described in the Tutorial

```
> ugclean all
> ugconf 2 PC XIF SEQ NOCHACO
> ugmake ug
> ugmake ug ar
```

The ar option reruns the archiving of object files into library files and is sometimes necessary if errors are reported in the next step.

```
> cd UG/course/appl
> make clean
> make
```

If compilation returns true you can now run:

```
> course2d
```

The UG shell (using the X interface) will start up. UG can also be run in the normal shell by:

```
> course2d -ui c
```

This may allow the program to run quicker if there is a lot of screen output.
Now to execute a script file to run the example model you can type

```
> ex fourash_2d_01.scr
```

Once the script is finished you can run more UG commands or scripts interactively.

Now for a 3D parallel processing example on a cluster of PCs

```
> ugclean all
> ugconf 3 PCCLUSTER MPI SIF CHACO
> ugmake ug
```

The SIF gives the standard interface, i.e. no UG shell, while CHACO compiles the load balancing software. MPI means that MPI message passing interface will be used.

```
> cd UG/course/appl
> make clean
> make
```

The course3d executable should now be available. In order to start UG on 4 processors, and to begin running the script immediately:

```
> mpirun -np 4 course3d fourash_3d_01.scr -noquit
```

The -noquit UG option allows the program to remain open for interactive work once the script has been executed.

10.5 Example of 2D Flow and Transport UG Script

The example 2D script is examined first.
The example script file is called:
`UG/course/appl/fourash2d/example_2d_transport.scr`
In order to run it as described above copy the files contents into:
`UG/course/appl/fourash2d/fourash_2d_transport_01.scr`

The file is now printed with comments following each block of script.

```
  1:   ##############################################################
  2:   # example_2d_transport.scr
  3:   #
  4:   # Ian Watson began Nov 2003
  5:   #
  6:   # example script to model
  7:   #   heterogeneous flow field and
  8:   #   reactive transport using non-linear Monod kinetics and
  9:   #   thermodynamic equilibrium
 10:   #   in parallel
 11:   #   using sparse matrix approach
 12:   #   with an adaptively refined & coarsened grid
 13:   #
 14:   #   the example is the same as the heterogeneous 2D model
 15:   #   presented in the PhD thesis by Ian Watson
 16:   ##############################################################
 17:
```

```
18:  ##########################################################
19:  # some constants
20:  ##########################################################
21:
22:  BASELEVEL       = 0;
23:  MAXLEVEL        = 2;                        # finest level
24:  TOPLEVEL        = 6;                        # adaptively refine to here
25:
26:  ADAPT           = 1;                        #adapt the grid
27:  SOLVEFLOW       = 1;                        #solve the flow field (change flag in c code
     also)
28:  ONEDIM          = 0;                        #for a 1D flowline simulation (change flag in c
     code also)
29:
30:  if (conf:parallel) {
31:      LBLEVEL        = 2;                                  # parallel load balance level
32:      if (ONEDIM)    ADAPT = 0;
33:      if (ADAPT==0) MAXLEVEL  = TOPLEVEL;                        # not adaptive
34:  }
35:
```

BASELEVEL is the coarsest grid of the multigrid, MAXLEVEL is used as the maximum grid level to which refinement is carried out uniformly, TOPLEVEL is the highest level adaptively refined to. The next flags allow grid adaption, flow solving (flow velocities can be simply provided and not calculated) and 1D solution. LBLEVEL is the level at which the grid is distributed to other processors in parallel.

```
36:  DOGRAPHICS          = 0;                          # plot results
37:  DOFILM              = 0;                          # generate film
38:  FILMINC         = 20;                             # every timestep ?
39:  DOCONCFILE      = 0;                              # make concentration file
40:  CONCFILEINC     = 500;                            # conc file interval
41:  DOTECPLOT       = 1;
42:  TECPLOTINC      = 10000;                          #step increment
43:  TECPLOTTIMEINC = 5;                               #time increment
44:
45:  VERSION =    "_example";
46:  up1     =    "../"
47:  outdir =     "fourash2d/out2d/"   #ensure this dir exists
48:  set basename  fourash2dt_01@VERSION;
49:
50:  DOT =       ".";
51:  LOG =       ".log";
52:  CONCS =     ".concs";
53:  GRID =      ".grid";
54:  DISC  =     "bf";
55:  set logname      @basename@DOT@MAXLEVEL@DOT@TOPLEVEL@LOG;
56:  set concsname    @basename@DOT@MAXLEVEL@DOT@TOPLEVEL@CONCS;
57:  set film         @basename@DOT@MAXLEVEL@DOT@TOPLEVEL;
58:  set tecname      @basename@DOT@MAXLEVEL@DOT@TOPLEVEL;
59:  set gridname     @basename@DOT@MAXLEVEL@DOT@TOPLEVEL@GRID;
60:
61:  ctr = 0;                    #for film frames
62:  ctrc =0;                    #for concentration file
63:  ctrtec =0;
64:  tecfileno = 0;
65:  TECPLOTTARGETTIME = TECPLOTTIMEINC;
66:  frameno = 0;
67:  set filename ian;
68:  PPM = "ppm";
69:  DAT = "dat";
70:  DARCY = "darcy"; # for darcy avs output
71:  FLOW = "flow";
72:
```

Sets various file names and output flags.

```
73:  #Upwinding#####################################
74:  ALPHA = 1.0;       #0.0 for central differencing, 1.0 for fully upwind
75:
76:  #timestep control#################################
77:  MINUTE= 60.;
78:  DAY  = 24.*60.*60.;       # for time stepping
79:  YEAR = 365.*DAY;
80:  STEP = 1;
81:  TIME = 0.0;              # initial time
82:  TIME_MINS = TIME/MINUTE;
83:  TIME_DAYS = TIME/DAY;
84:  TIME_YEAR = TIME/YEAR;
85:
86:  END    = 47*YEAR;
87:  STEPS  = 10;  #10000;        # max no. of time steps
88:
89:
90:  DT       = 0.005*YEAR;
91:  DTMAX    = 0.05*YEAR;       # largest  timestep
92:  DTMIN    = 0.01* DT;        # smallest timestep
93:  DTSCALE = 1.1892;          # 2^0.25 to restore dt in 4 steps
94:
```

Sets discretisation scheme upwinding parameter, alpha, and time step control parameters.

```
 95:
 96:    # limits for newton solver#################
 97:    ABSLIMIT       = 5.E-7;
 98:    RED            = 5.E-5;
 99:    LRED           = 1.E-4;
100:
```

The convergence tolerances used.

```
101:    #Chemistry#################################
102:    sol= "sol";
103:
104:    MOBILE     = 15;
105:    TOTSPECIES = 24;
106:
107:    C0 = "c6h6o";              #a
108:    C1 = "o2";                 #b
109:    C2 = "no3-m1";             #c
110:    C3 = "h2";                 #d
111:    C4 = "fe-p2";              #e
112:    C5 = "mn-p2";              #f
113:    C6 = "so4-m2";             #g
114:    C7 = "hs-m1";              #h
115:    C8 = "ch4";                #i
116:    C9 = "ch3cooh";            #j
117:    C10 = "hp1";               #k
118:    C11 = "hco3-m1";           #l
119:    C12 = "ohm1";              #m
120:    C13 = "h2co3";             #n
121:    C14 = "co3-m2";            #o
122:    C15 = "feooh-s";           #p
123:    C16 = "mno2-s";            #q
124:    C17 = "fes-s";             #r
125:    C18 = "xfeoh";             #s
126:    C19 = "xfeoh2-p1";         #t
127:    C20 = "xfeo-m1";           #u
128:    C21 = "xfeofe-p1";         #v
129:    C22 = "xfeofeoh";          #w
130:    C23 = "xfeomn-p1";         #x
131:
132:    # NB don't use "_" in species names in script
133:    # ensure order is same as in c code
134:
```

The chemical species names are listed for convenience in outputting results. Numbers of mobile and total species and the order of species must be same as in the C code. The letters are those used in the format, see below.

```
135:    #              a b c d e f g h i j k l m    n    o    p q r    s    t    u    v    w
        x
136:    NEWTON_SCALE = "1:1:1:1:1:1:1:1:1:1:1:1:1e-10:1e-10:1e-10:1:1:1e-10:1e-5:1e-10:1e-10:1e-
        10:1e-10:1e-10";
137:    #keep all scale string on one continuous line!
138:
```

This can be used to give the equilibrium reactions less weight in the residual and helps the solver performance. Kinetic species (primary components) have a scale of 1 so that mass is conserved between them (equilibrium equations do not carry any mass).

```
139:    ##########################################################
140:    # for refinement
141:    ##########################################################
142:
143:    #       RefineIf  = 1;
144:    #       CoarsenIf = 0;
145:    #       RefineIf  = "r<5&&((d|c6h6osol|>0.7))";
146:    #       CoarsenIf = "l>1&&((d|c6h6osol|<0.35))";
147:
148:    #       InitialRefineIf  = "r<5&&((d|c6h6osol|>0.7)||(d|o2sol|>0.003)||(d|fe-
        p2sol|>0.004))";
149:    #       RefineIf         = "r<5&&((d|c6h6osol|>0.7)||(d|o2sol|>0.003)||(d|fe-
        p2sol|>0.004))";
150:    #       InitialCoarsenIf = "l>1&&((d|c6h6osol|<0.7)&&(d|o2sol|<0.003)&&(d|fe-
        p2sol|>0.004))";
151:    #       CoarsenIf        = "l>1&&((d|c6h6osol|<0.35)&&(d|o2sol|<0.0015)&&(d|fe-
        p2sol|>0.002))";
152:
153:    if (conf:parallel) {
154:            if (ADAPT==1) {
155:                    if (TOPLEVEL == 3) InitialRefineIf  = "r<3&&(d|c6h6osol|>0.7)";
156:                    if (TOPLEVEL == 4) InitialRefineIf  = "r<4&&(d|c6h6osol|>0.7)";
157:                    if (TOPLEVEL == 5) InitialRefineIf  = "r<5&&(d|c6h6osol|>0.7)";
158:                    if (TOPLEVEL == 6) InitialRefineIf  = "r<6&&(d|c6h6osol|>0.7)";
159:                    if (TOPLEVEL == 3) RefineIf  = "r<3&&(v|no3-m1sol|>1e-
        4)&&(d|o2sol|>0.003)";
160:                    if (TOPLEVEL == 4) RefineIf  = "r<4&&(v|no3-m1sol|>1e-
        4)&&(d|o2sol|>0.003)";
161:                    if (TOPLEVEL == 5) RefineIf  = "r<5&&(v|no3-m1sol|>1e-
        4)&&(d|o2sol|>0.003)";
162:                    if (TOPLEVEL == 6) RefineIf  = "r<6&&(v|no3-m1sol|>1e-
        4)&&(d|o2sol|>0.003)";
163:                    InitialCoarsenIf =0;
```

```
164:                    #CoarsenIf = 0;
165:                    CoarsenIf = "l>5&&((v|no3-m1sol|<1e-4)&&(d|o2sol|<0.0003))";
166:            }
167:    }
168:
```

Various indicators for refinement and coarsening. v is for value, d is for difference across an element (i.e. similar to gradient).

```
169:    ############################################################
170:    # open log file
171:    ############################################################
172:    logoff;
173:    closewindow $a;
174:    close $a;
175:
176:    logon @up1@outdir@logname;
177:
178:    ###  get heterogeneous k field
179:    if (SOLVEFLOW) rsf_read gen/scale.out;
180:
```

Remove previously open items, open log file in specified output directory, read in a file with a random spatial function for use in flow solving.

```
181:    ############################################################
182:    # data format definition  SPARSE
183:    ############################################################
184:    SPARSE=1;
185:
186:    if (SPARSE==1)
187:    {
188:    ms :SparseFormats:Jsparse;
189:    #                                abcdefghijklmnopqrstuvwx
190:    :SparseFormats:Jsparse:Dnn =  "***000*00000000000000000
191:                                   **0000000000000000000000
192:                                   ***0000000000000000000000
193:                                   ****00*000**0****0000000
194:                                   *****0**0000000*0*000**0
195:                                   ****0*0000000000*000000*
196:                                   ****00*00000000000000000
197:                                   *****0**00*000000*000000
198:                                   ****0000*000000000000000
199:                                   ***000000*00000000000000
200:                                   ************************
201:                                   ****0000*00*0**000000000
202:                                   000000000*00*00000000000
203:                                   0000000000**0*0000000000
204:                                   0000000000**00*000000000
205:                                   ****00000000000*00000000
206:                                   ****0000000000000*0000000
207:                                   0000*00*00*000000*000000
208:                                   0000000000*0000000******
209:                                   0000000000*0000000**0000
210:                                   0000000000*0000000*0*000
211:                                   0000*00000*0000000*00*00
212:                                   0000*00000*0000000*000*0
213:                                   00000*0000*0000000*0000*
214:                                   ";
215:
216:    #                                abcdefghijklmnopqrstuvwx
217:
218:    :SparseFormats:Jsparse:Tnn = " *000000000000000000000000
219:                                   0*000000000000000000000000
220:                                   00*00000000000000000000000
221:                                   000*0000000000000000000000
222:                                   0000*000000000000000000000
223:                                   00000*00000000000000000000
224:                                   000000*0000000000000000000
225:                                   0000000*000000000000000000
226:                                   00000000*00000000000000000
227:                                   000000000*0000000000000000
228:                                   0000000000*0***000000000
229:                                   00000000000*0**000000000
230:                                   000000000000000000000000
231:                                   000000000000000000000000
232:                                   000000000000000000000000
233:                                   000000000000000000000000
234:                                   000000000000000000000000
235:                                   000000000000000000000000
236:                                   000000000000000000000000
237:                                   000000000000000000000000
238:                                   000000000000000000000000
239:                                   000000000000000000000000
240:                                   000000000000000000000000
241:                                   000000000000000000000000
242:                                   ";
243:
244:    # to check the sparsity pattern, run the model with SPARSE=0
245:    # and plot the Jmatrix
246:
247:    jsp = "|Jsparse";
```

```
248:    }
249:    if (SPARSE==0)      # set jsp to blank for format, trick doesn't work in parallel
250:    {
251:    jsp = " ";
252:    }
253:
```

To use sparse matrix storage which only stores the block entries specified by * set SPARSE to 1. Also must define #SPARSE in the file boxnlsys.c and a couple of places in the ug library (UG/ug/np/algebra/ugiter.c & UG/ug/np/algebra/ugblas.c). Sparsity patterns are given for the diagonal blocks and the off-diagonal blocks of the Jacobian matrix. In each block the entries of the Jacobian are given by $\partial F_i/\partial C_j$ according to Newton's method where F_i is the i^{th} non-linear equation for the system of C_j unknowns:

$$\frac{\partial F_i}{\partial C_j}\Delta C_j = -F_i$$

In the diagonal blocks $\partial F_i/\partial C_j$ will be non-zero where mass storage terms or reaction terms cause coupling between species C_i and C_j. The off-diagonal blocks are affected only by transport of species, and are diagonal except for where secondary mobile species are summed into the primary mobile components.

```
254:    if (conf:parallel) {
255:    newformat FlowFormat
256:                            $V n1: nt 20 $M implicit(nt): mt 2 $I n1
257:                        $V n@TOTSPECIES : nt@TOTSPECIES 17 $comp abcdefghijklmnopqrstuvwx
258:                            $sub @C0    a
259:                            $sub @C1    b
260:                            $sub @C2    c
261:                            $sub @C3    d
262:                            $sub @C4    e
263:                            $sub @C5    f
264:                            $sub @C6    g
265:                            $sub @C7    h
266:                            $sub @C8    i
267:                            $sub @C9    j
268:                            $sub @C10   k
269:                            $sub @C11   l
270:                            $sub @C12   m
271:                            $sub @C13   n
272:                            $sub @C14   o
273:                            $sub @C15   p
274:                            $sub @C16   q
275:                            $sub @C17   r
276:                            $sub @C18   s
277:                            $sub @C19   t
278:                            $sub @C20   u
279:                            $sub @C21   v
280:                            $sub @C22   w
281:                            $sub @C23   x
282:                            #$M implicit(nt@TOTSPECIES):    mt@TOTSPECIES 2 $I n2;
283:                            $M implicit(nt@TOTSPECIES@jsp): mt@TOTSPECIES 2 $I n2;
284:    #need 2 matrices in parallel
285:    }
286:    else            # sequential processing format
287:    {
288:    newformat FlowFormat
289:                            $V n1: nt 20 $M implicit(nt): mt 2 $I n1
290:                        $V n@TOTSPECIES : nt@TOTSPECIES 17 $comp abcdefghijklmnopqrstuvwx
291:                            $sub @C0    a
292:                            $sub @C1    b
293:                            $sub @C2    c
294:                            $sub @C3    d
295:                            $sub @C4    e
296:                            $sub @C5    f
297:                            $sub @C6    g
298:                            $sub @C7    h
299:                            $sub @C8    i
300:                            $sub @C9    j
301:                            $sub @C10   k
302:                            $sub @C11   l
303:                            $sub @C12   m
304:                            $sub @C13   n
305:                            $sub @C14   o
306:                            $sub @C15   p
307:                            $sub @C16   q
308:                            $sub @C17   r
309:                            $sub @C18   s
310:                            $sub @C19   t
311:                            $sub @C20   u
312:                            $sub @C21   v
313:                            $sub @C22   w
314:                            $sub @C23   x
315:                            $M implicit(nt@TOTSPECIES@jsp): mt@TOTSPECIES 1 $I n2;
316:
```

```
317:    }
318:
```

Describe the data format to be used (more details in the UG tutorial and documentation). Firstly, the format of the flow, which has one unknown (hydraulic head) is given. Secondly, the format for transport of TOTSPECIES (24) is specified. The vector template is named nt24 [257, 290]. The component symbols are listed and subsymbols used to name each species (for convenience). Now if sol is the name of the soluton vector the subsymbol can be used to identify particular components, e.g. o2sol.

To change parallel script to non-sparse change the matrix $M options by using [282] instead of [283].

```
319:    ##########################################################
320:    # heap assignment
321:    ##########################################################
322:    if (conf:parallel) HEAP = "200M";
323:    else               HEAP = "350M";
324:
```

Ensure you have enough memory allocated. In parallel HEAP refers to memory per processor. If running adaptively memory use will increase towards the limit specified here as more nodes are introduced - be careful not to run out of memory mid-simulation, check first how much memory a uniform grid uses.

```
325:
326:    ##########################################################
327:    # interactive (do not use in parallel)
328:    ##########################################################
329:    #refreshon;
330:
```

Refreshon is for use with UG on-screen graphics

```
331:    ##########################################################
332:    # graphics procedures
333:    ##########################################################
334:
335:    gw = "{
336:            if (DOGRAPHICS==1)
337:            {
338:
339:                    openwindow 25 80 402 250 $n Win_Grid;
340:                    openpicture $s 1 1 400 120 $n grid;
341:                    setplotobject Grid $b 1 $n 0 $e 0 $c 1 $w a $r 0 $p 0.9;
342:                    setview $s 1 5;
343:                    zoom 0.31;
344:                    plot;
345:
346:
347:                    openwindow 25 270 402 250 $n Win_Flow;
348:
349:                    openpicture $s 1 1 400 120 $n sol_p;
350:                    setplotobject EScalar $s sol_p $m COLOR $n 20 $d 0 $f 0.0001 $t 0.9999;
351:                    setview $s 1 5;
352:                    zoom 0.31;
353:                    plot;
354:
355:                    openpicture $s 1 122 400 120 $n vel;
356:                    setplotobject EVector $e darcy $s bf $r 12 $l 2 $g 0;
357:                    setview $s 1 5;
358:                    zoom 0.31;
359:                    plot;
360:
361:
362:                    openwindow 325 200 250 515 $n Win_Conc;
363:
364:                    openpicture $s 1 1 200 100 $n @C0;
365:                    setplotobject EScalar $e nls0 $s box $d 2 $m CONTOURS_EQ $n 30 $f 1E-5 $t
        9.99999;
366:                    setview $s 2 35 $t 375 11;
367:                    zoom 1.04;
368:                    plot;
369:                    drawtext 10 80 @C0;
370:
371:                    openpicture $s 1 103 200 100 $n @C1;
372:                    setplotobject EScalar $e nls1 $s box $d 2 $m CONTOURS_EQ $n 30 $f 1E-5 $t
        9.99999;
373:                    setview $s 2 35 $t 375 11;
374:                    zoom 1.04;
375:                    plot;
376:                    drawtext 10 180 @C1;
377:
378:                    openpicture $s 1 205 200 100 $n @C2;
379:                    setplotobject EScalar $e nls2 $s box $d 2 $m CONTOURS_EQ $n 30 $f 1E-5 $t
        9.99999;
380:                    setview $s 2 35 $t 375 11;
381:                    zoom 1.04;
382:                    plot;
383:                    drawtext 10 280 @C2;
```

```
384:
385:                     openpicture $s 1 307 200 100 $n @C3;
386:                     setplotobject EScalar $e nls3 $s box $d 2 $m CONTOURS_EQ $n 30 $f 1E-5 $t
      9.99999;
387:                     setview $s 2 35 $t 375 11;
388:                     zoom 1.04;
389:                     plot;
390:                     drawtext 10 380 @C3;
391:
392:                     openpicture $s 1 409 200 100 $n @C4;
393:                     setplotobject EScalar $s @C4@sol $d 2 $m CONTOURS_EQ $n 30 $f 1E-5 $t
      9.99999;
394:                     setview $s 2 35 $t 375 11;
395:                     zoom 1.04;
396:                     plot;
397:                     drawtext 10 480 @C4;
398:
399:                     drawtext 10 10 Time:@TIME_YEAR;
400:
401:             }
402:     }";
403:
```

Create some windows and pictures of the grid, the pressure head and velocity, and concnetrations of the first 5 species. Note that species C0, C1, C2, C3 are plotted using element evaluation procedures "$e nls0 $s box" [365], while C4 uses the subsymbol method "$s @C4@sol" [393].

```
404:    plotw = "{
405:    if (DOGRAPHICS==1)
406:    {
407:             setcurrwindow Win_Flow;
408:             setcurrpicture sol_p;
409:             findrange $p;
410:             plot;
411:             setcurrpicture vel;
412:             findrange $p $z 0.1;
413:             plot;
414:    }
415:    }";
416:
417:    plotgrid = "{
418:    if (DOGRAPHICS==1)
419:    {
420:             setcurrwindow Win_Grid;
421:             setcurrpicture grid;
422:             plot;
423:    }
424:    }";
425:
426:    plotC = "{
427:    if (DOGRAPHICS==1)
428:    {
429:             setcurrwindow Win_Conc;
430:             setcurrpicture @C0;
431:             findrange $p; # $z 0.1;
432:             plot; drawtext 10 80 @C0;
433:             setcurrpicture @C1;
434:             findrange $p; # $z 0.1;
435:             plot; drawtext 10 180 @C1;
436:             setcurrpicture @C2;
437:             findrange $p; # $z 0.1;
438:             plot; drawtext 10 280 @C2;
439:             setcurrpicture @C3;
440:             findrange $p; # $z 0.1;
441:             plot; drawtext 10 380 @C3;
442:             setcurrpicture @C4;
443:             findrange $p; # $z 0.1;
444:             plot; drawtext 10 480 @C4;
445:             drawtext 10 10 Time:@TIME_YEAR;
446:             }
447:    }";
448:
```

Functions to re-plot the flow window, the grid window, and the concentrations window. The range of values plotted is reset on each function call.

```
449:    plotCprofile = "{
450:    if (DOGRAPHICS==1)
451:    {
452:
453:    }
454:    }";
455:
456:    # plot the matrix
457:    mw = "{
458:             if (DOGRAPHICS==1)
459:             {
460:                 refreshon;
461:                     openwindow 1 1 902 902 $n Win2;
462:
```

```
463:                    openpicture $s 1 1 900 900 $n mat;
464:                    setplotobject Matrix $M MAT;
465:                    setview;
466:                    plot;
467:            }
468:    }";
469:
470:    # plot the Jacobean matrix for nltransport
471:    mj = "{
472:            if (DOGRAPHICS==1)
473:            {
474:                    level 0;
475:                    refreshon;
476:                    openwindow 40 10 402 402 $n Jmatrix;
477:                    openpicture $s 1 1 400 400 $n Jmatrix;
478:                    setplotobject Matrix $M jmat;
479:                    setview;
480:                    plot;
481:            }
482:    }";
483:
```

Plot the matrices for the flow problem or the transport problem. The Jacobian is plotted on the base grid, level 0 [474], for clarity of the sparsity pattern. It is best plotted without using the sparse matrix format.

```
484:    # postscript output of the domain decomposition pattern
485:    plotparallelgrid="{
486:                    openwindow 0 0 402 180 $d ps $n @gridname;
487:                    openpicture $s 1 40 400 120 $n mesh;
488:                    setplotobject Grid $b 1 $n 0 $e 0 $c 2 $w a $r 0 $p 0.9;
489:                    setview $s 1 5;
490:                    zoom 0.31;
491:                    plot;
492:                    closewindow;
493:    }";
494:
```

Writes graphical postscript output illustrating the grid decomposition to processor domains.

```
495:    # make frames for animation
496:    frame="
497:        if (DOFILM==1) {
498:            if (ctr==@FILMINC) ctr = 0;
499:            if (ctr==0) {
500:                    cmfn @film frameno $n filename;
501:
502:                    openwindow 0 0 402 660 $d ps $n @filename;
503:
504:                    drawtext 1 1 Time:@TIME_YEAR;
505:
506:                    openpicture $s 1 40 400 120 $n mesh;
507:                    setplotobject Grid $b 1 $n 0 $e 0 $c 1 $w a $r 0 $p 0.9;
508:                    setview $s 1 5;
509:                    zoom 0.31;
510:                    plot;
511:
512:                    openpicture $s 1 162 400 120 $n C0;
513:                    setplotobject EScalar $s @C0@sol $d 2 $m CONTOURS_EQ $n 30 $f 1E-5 $t
        9.99999;
514:                    setview $s 1 5;
515:                    zoom 0.31;
516:                    plot;
517:
518:                    openpicture $s 1 284 400 120 $n C1;
519:                    setplotobject EScalar $e nls1 $s box $d 2 $m CONTOURS_EQ $n 30 $f 1E-5 $t
        9.99999;
520:                    setview $s 1 5;
521:                    zoom 0.31;
522:                    plot;
523:
524:                    openpicture $s 1 406 400 120 $n C2;
525:                    setplotobject EScalar $e nls2 $s box $d 2 $m CONTOURS_EQ $n 30 $f 1E-5 $t
        9.99999;
526:                    setview $s 1 5;
527:                    zoom 0.31;
528:                    plot;
529:
530:                    openpicture $s 1 528 400 120 $n C3;
531:                    setplotobject EScalar $e nls3 $s box $d 2 $m CONTOURS_EQ $n 30 $f 1E-5 $t
        9.99999;
532:                    setview $s 1 5;
533:                    zoom 0.31;
534:                    plot;
535:
536:                    closewindow;
537:
538:                    frameno = frameno+1;
539:                    }
540:            ctr = ctr + 1;
541:            }
542:    ";
```

```
543:
```
Similar to on screen window plotting, but writes a postscript file at regular intervals, every FILMINC timesteps.
```
544:    # data output in tecplot format
545:    do_tecplot ="{
546:                        mytecplot @outdir@tecplotname@DOT@DAT
547:                                $title @basename
548:                                $zone Concs_time=@TIME_YEAR
549:                                $e nvalue $s @C0@sol $v @C0
550:                                $e nvalue $s @C1@sol $v @C1
551:                                $e nvalue $s @C2@sol $v @C2
552:                                $e nvalue $s @C3@sol $v @C3
553:                                $e nvalue $s @C4@sol $v @C4
554:                                $e nvalue $s @C5@sol $v @C5
555:                                $e nvalue $s @C6@sol $v @C6
556:                                $e nvalue $s @C7@sol $v @C7
557:                                $e nvalue $s @C8@sol $v @C8
558:                                $e nvalue $s @C9@sol $v @C9
559:                                $e nvalue $s @C10@sol $v @C10
560:                                $e nvalue $s @C11@sol $v @C11
561:                                $e nvalue $s @C12@sol $v @C12
562:                                $e nvalue $s @C13@sol $v @C13
563:                                $e nvalue $s @C14@sol $v @C14
564:                                $e nvalue $s @C15@sol $v @C15
565:                                $e nvalue $s @C16@sol $v @C16
566:                                $e nvalue $s @C17@sol $v @C17
567:                                $e nvalue $s @C18@sol $v @C18
568:                                $e nvalue $s @C19@sol $v @C19
569:                                $e nvalue $s @C20@sol $v @C20
570:                                $e nvalue $s @C21@sol $v @C21
571:                                $e nvalue $s @C22@sol $v @C22
572:                                $e nvalue $s @C23@sol $v @C23
573:                                ;
574:                        }
575:    ";
576:
```
Specify the tecplot output format for all the species, $v gives the variable (species) name. The command mytecplot is set up in UG/course/appl/gen/mytecplot.c, which is a copy of the original UG/ug/ui/tecplot.c.
```
577:    outputtecplot = "{
578:        if (DOTECPLOT==1) {
579:            if (ctrtec==@TECPLOTINC) ctrtec = 0;
580:            if (TIME_YEAR>=@TECPLOTTARGETTIME)
581:                    {
582:                    ctrtec = 0;
583:                    TECPLOTTARGETTIME = TECPLOTTARGETTIME + TECPLOTTIMEINC;
584:                    }
585:
586:                    if (ctrtec==0)
587:                    {
588:                            cmfn @tecname tecfileno $n tecplotname;
589:                            @do_tecplot;
590:
591:                            tecfileno = tecfileno + 1;
592:
593:                            #now print concentration file interpolated to a regular grid:
594:                            if (conf:parallel==0) printconcfile sol $file @outdir@concsname
595:                                                    $time @TIME_YEAR
596:                                                    $from 0 0 $to 750 20 $dx 7.5 $dy 0.625;
597:
598:                    }
599:            ctrtec = ctrtec + 1;
600:            }
601:    }";
```
Output tecplot file every TECPLOTINC steps or every TECPLOTTIMEINC years. Here the printconcfile command has also been included to output data interpolated to a regular grid with spacing dx by dy. Printconcfile command is found in UG/course/appl/gen/mycommands.c along with some other general script commands used in this example script.
```
602:
603:    concfile="
604:        if (DOCONCFILE==1) {
605:            if (ctrc==@CONCFILEINC) ctrc = 0;
606:            if (ctrc==0)
607:            {
608:                    if (conf:parallel==0) printconcfile sol $file @outdir@concsname
609:                                            $time @TIME_YEAR
610:                                            $from 0.076 0 $to 0.076 0.040 $dx 0.001 $dy
        0.001;
611:            }
612:            ctrc = ctrc +,1;
613:    }
614:    ";
615:
```
Another example of the use of printconcfile command.

```
616:    ########################################################
617:    # create multigrid
618:    ########################################################
619:    if (SOLVEFLOW)  new mg $b FAflow1 $f FlowFormat $h @HEAP;
620:    else            new mg $b fa2dt_01 $f FlowFormat $h @HEAP;
```

The newly created multigrid can point to the boundary problem for either flow Faflow1 [619], or transport fa2dt_01 [620]. Faflow1 is set up in the file: UG/course/appl/fourash/fourash_flow1.c

```
621:
622:    ex four_quadrilaterals.mesh
623:    # ex fourash_1b.mesh;             #  gives a grid of 8 triangles
624:    # makegrid $C $D 10 $h 20.0;      #  gives a grid of size h
625:
626:    fixcoarsegrid;
627:
```

A script is executed to introduce nodes and elements in the domain to form four rectangular elements – see UG/course/appl/visplume/four_quadrilaterals.mesh. Makegrid makes an automatic triangular grid of size $h 20.0 meters. Beware of numerical dispersion when using triangles which are not aligned with the flow direction.

```
628:    ########################################################
629:    ########################################################
630:    # DEFINE NUMPROCS
631:    ########################################################
632:    ########################################################
633:
634:    ########################################################
635:    # groundwater flow
636:    ########################################################
637:    #flow velocity may be given or calculated (solved)
638:    if (SOLVEFLOW)
639:    {
640:    #set up parameter and discretisation objects
641:    npcreate flow $c FAFLOW;
642:    npinit flow;
643:
644:    npcreate bf $c bf;
645:    npinit bf $A MAT/mt $x sol_p/nt $b rhs_p/nt $param flow;
646:
```

The class FAFLOW is set up in the flow parameter file: UG/course/appl/fourash/ fourash_flow1.c. The file specifies permeability and boundary head values. If a heterogeneous flow field is required , using the file read in at [179], then #define HETERO_K_FIELD flag must be set in that file.

The boxflow discretisation object is created [644] using the class, bf, defined in file UG/course/pclib/water/boxflow.c. Matrix and vector templates, /mt and /nt, are specified [645] since they differ from those used by transport.

```
647:    # set up solver
648:    npcreate smooth_w $c gs; #$c ssor; #$c ilu;
649:    npinit smooth_w $damp 1.0;
650:
651:    npcreate base_w $c ex;
652:    npinit base_w;
653:
654:    npcreate basesolver_w $c ls;
655:    npinit basesolver_w $red 1.0E-4 $m 50 $I base_w $display no $abslimit 0;
656:
657:    npcreate transfer_w $c transfer;
658:    npinit transfer_w $x sol_p;     # $S 2.0;
659:
660:    npcreate lmgc_w $c lmgc;
661:    npinit lmgc_w $S smooth_w smooth_w basesolver_w $T transfer_w $n1 1 $n2 1 $g 1;
662:
663:    npcreate mgs_w $c bcgs;
664:    npinit mgs_w $A MAT $x sol_p $b rhs_p $m 50 $red 1.0E-8
665:              $abslimit 1.0E-80 $I lmgc_w $display full;
666:
667:    npcreate amgs $c amg;
668:    npinit amgs $A MAT $x sol_p $b rhs_p $red 1.0E-8 $abslimit 1.0E-15 $display full
669:            $alpha 0.4 $beta 1.0E-3 $minc 4 $maxc 6 $maxd 4  $dependency sym
670:            $maxcon 15 $vc 1 $dt 20 $ct 200 $cr 1.3 $major -1
671:            $scale 0 $vs 1 $solver cg $prec mgc $it 80 $sm ssor $n1 2 $n2 2 $g 1 $omp 1.8
672:            $oms 1.0
673:            $csm ex $cit 100 $cred 1.0E-3;
674:
675:    }
676:
```

This sets up the multigrid solver for the flow problem. The smoother should use Gauss-Seidel (gs) if the sparse matrix format is used.

```
677:    ########################################################
678:    # transport problem
679:    ########################################################
680:
```

```
681:    #set up parameter and discretisation objects
682:    npcreate trans $c FA2DT_01;
683:    if (SOLVEFLOW) { npinit   trans $flow bf;  }
684:    else           { npinit   trans;           }
685:
686:    npcreate box $c boxnlsys;
687:    npinit box $param trans $alpha @ALPHA $C sol/nt@TOTSPECIES $mobile @MOBILE $Uoption 1;
688:
```

The parameter class FA2DT_01 is set up in the file: UG/course/appl/fourash2d/fourash2d_transport_01.c, described below. If the flow is solved the paramter object "trans" has to be informed where to find information on the flow velocity [683]. The boxnlsys discretisation class created [686] is found in file UG/course/pclib/transport/boxnlsys.c. It is initialised with the upwinding parameter, alpha, the vector template for concentrations, the total number of mobile species, and the Uoption flag which allows thermodynamic equilibrium reactions to be included.

```
689:    #set up solver
690:    npcreate smooth1 $c gs;
691:    npinit smooth1; # $autodmp 1;
692:
693:    npcreate base1 $c ls;
694:    npinit base1 $red 0.001 $m 100 $I smooth1 $display no;
695:
696:    npcreate transfer $c transfer;
697:    npinit transfer $x sol;
698:
699:    npcreate lmgc $c lmgc;
700:    npinit lmgc $S smooth1 smooth1 base1 $T transfer $n1 2 $n2 2 $g 1;
701:
702:    npcreate mgs $c bcgs;
703:    npinit mgs $m 25 $abslimit @ABSLIMIT $I lmgc $display no;
704:
705:    npcreate newton $c newton;
706:    npinit newton $abslimit @ABSLIMIT $red @RED $T transfer $S mgs
707:                  $rhoreass 0.8 $lsteps 8 $maxit 15 $line 2 $linrate 0
708:                  $lambda 1.0 $divfac 1.0E100 $linminred @LRED $display full
709:                  $scale n:@NEWTON_SCALE
710:                  $J jmat/mt@TOTSPECIES;   # must use J option with sparse
711:
712:    npcreate ts $c bdf;
713:    npinit ts $y sol $A box $S newton $T transfer
714:              $baselevel @BASELEVEL $order 1 $predictorder 0 $nested 0
715:              $dtstart @DT $dtmin @DTMIN $dtmax @DTMAX $dtscale @DTSCALE
716:              $rhogood 1.0E-7 $display full;
717:
```

This sets up the multigrid solver for the transport problem. The smoother should use Gauss-Seidel (gs) if the sparse matrix format is used. The linearised system is solved by a bi-conjugate gradient solver (bcgs) which uses multigrid v cycles. Linearisation is done by the newton method, and the time solver is set up to use backward difference formula, bdf.

```
718:    # Create refinement procedure
719:    if (ADAPT) {
720:    npcreate igr $c igr;
721:    npinit igr $refineif @InitialRefineIf  $coarsenif @InitialCoarsenIf;
722:    }
```

The indicator for grid refinement (igr) is created with values set earlier for adaptive simulations.

```
723:
724:    ##########################################################
725:    # refine mesh up to BASELEVEL
726:    ##########################################################
727:
728:    @gw;
729:    j = 0;
730:    repeat {
731:            if (conf:parallel) {
732:              if (j==LBLEVEL) ex lb4.scr;
733:            }
734:            if (j == BASELEVEL) break;
735:            refine $a;
736:            j = j+1;
737:    }
738:    lexorderv ru;
739:
```

This refinement procedure is intended for BASELEVELs not equal to zero. In the present set up this is not used.

```
740:    ##########################################################
741:    # refine mesh to MAXLEVEL
742:    ##########################################################
743:    ##########################################################
744:    if (SOLVEFLOW)
745:    {
746:            clear sol_p $a $v 0.0;
747:            repeat
```

```
748:            {
749:                    print "LEVEL ", j;
750:                    if (conf:parallel) { if (j==LBLEVEL) ex lb4.scr; }
751:                    npexecute bf $a;
752:                    npexecute mgs_w $i $d $r $s $p;
753:                    @plotw;
754:                    if (j == MAXLEVEL) break;
755:                    j = j+1;
756:                    refine $a;
757:                    lexorderv ru;
758:                    npexecute transfer_w $N;
759:            }
760:            npexecute transfer_w $P; # project solution sol_p to coarse grid
761:
762:            #avs @basename@DOT@DARCY $nv darcy $s bf;
763:
764:            mytecplot @outdir@basename@DOT@FLOW@DOT@DAT
765:                                    $title @basename@FLOW
766:                                    $zone Flowfield
767:                                    $e darcy0 $s bf     $v q_x
768:                                    $e darcy1 $s bf     $v q_z
769:                                    $e nvalue $s sol_p $v head
770:                                    $e K      $s flow  $v K
771:                                    ;
772:    }
```

The grid is refined uniformly up to MAXLEVEL. Flow is solved on each level [751-752] and plotted [753] as refinement [756] proceeds. [750] calls the load balancing script when LBLEVEL is reached. The load balancing script is at UG/course/appl/lb4.scr. The avs format can output vector data ($nv darcy) while tecplot output is implemented to output scalars, so the darcy0 and darcy1 element evaluation procedures are used [767-768]. These evaluation procedures are set up in UG/course/pclib/water/water.c.

```
773:    else          #flow is not solved
774:    {
775:            repeat
776:            {
777:                    print "LEVEL ", j;
778:                    if (j==LBLEVEL) { if (conf:parallel) ex lb4.scr; }
779:                    if (j == MAXLEVEL) break;
780:                    j = j+1;
781:
782:                    if (ONEDIM){
783:                            mark blue 0 $a;
784:                            #mark blue 0 $y 0.005;
785:                            refine;
786:                    }
787:                    else
788:                            refine $a;
789:
790:            }
791:                    lexorderv ru;
792:    }
793:
794:    glist;          # list all multigrid levels
795:    heapstat;       # summarise memory useage
796:
```

The refinement procedure is similar if no flow is solved. If a one dimensional model is required then the refinement rule "blue" [783] can be used to obtain refinement in one direction only, here parallel to x.

```
797:    ##########################################################
798:    ##########################################################
799:    # COMPUTE CONTAMINANT TRANSPORT
800:    ##########################################################
801:    ##########################################################
802:    # Initialise
803:    ##########################################################
804:
805:    if (SOLVEFLOW) configure FAflow1 $p fa2dt_01;   # switch to transport problem for BCs
806:
807:    npexecute ts $pre $init;
808:    npexecute box $clear;
809:
810:    if (ADAPT){             # initial refinement from maxlevel to toplevel.
811:            count = TOPLEVEL - MAXLEVEL;
812:            repeat {
813:                    if (count<=0) break;
814:                    npexecute igr;
815:                    adapt $t;
816:                    glist;
817:                    if (SOLVEFLOW) npexecute transfer_w $N; # interpolate flow to new grid
        (necessary)
818:                    npexecute transfer $N;  # interpolate concs to new grid (necessary)
819:                    count = count-1;
820:                    }
821:            npexecute ts $pre $init;
822:            npexecute box $clear;
```

```
823:            }
824:    npexecute box $mass $time 0;
825:    @plotC;
826:    @plotgrid;
827:    @outputtecplot;
828:    if (conf:parallel) { @plotparallelgrid; }
829:    else {
830:    @frame;
831:    @concfile;
832:    }
833:    heapstat;
834:
835:    TOTAL_SOLVER_RUNTIME =0; TOTAL_RUNTIME = 0;
836:    BVP_SWITCH=0;
837:    TIME_OLD = TIME;
838:
```

In the initialisation the grid is adapted from MAXLEVEL to TOPLEVEL using the initial refinement indicators. The refinement area is limited to near the inflow boundary for this problem. The simulation is now ready to start the time loop. The initial global mass of all the species is output [824] – the function is set up in UG/course/pclib/transport/boxnlsys.c.

```
839:    ##########################################################
840:    # Loop over time steps
841:    ##########################################################
842:
843:    repeat
844:    {
845:            resetclock;
846:            print "TIMESTEP: ", STEP, " StartTIME: ", @TIME_YEAR;
847:
```

Begin the time loop with the repeat script command. The clock is reset to measure CPU time.

```
848:    ### Locally refine an area and recompute flow#####################
849:            if (ADAPT){
850:                    # allow refinement from maxlevel to toplevel each time step.
851:                    count = TOPLEVEL - MAXLEVEL;
852:                    repeat{
853:                            if (count<=0) break;
854:                            npexecute igr;
855:                            adapt $t;
856:                            if (SOLVEFLOW) npexecute transfer_w $N; # interpolate flow to new
    grid (necessary)
857:                            npexecute transfer $N;  # interpolate concs to new grid
    (necessary)
858:                            count =count-1;
859:                    }
860:                    glist;
861:                    @plotgrid;
862:                    heapstat;
863:                    if (STEP==1) npinit igr $refineif @RefineIf  $coarsenif @CoarsenIf;
864:
865:                    # solve flow on new grid##############
866:                    if (SOLVEFLOW){
867:                    configure  FAflow1 $p FAflow1;   # switch to flow problem for BCs
868:                    npexecute bf $a;
869:                    npexecute mgs_w $i $d $r $s $p;
870:                    @plotw;
871:                    if (TIME_YEAR<=25) configure FAflow1 $p fa2dt_01;   # switch to later bvp
872:                    else               configure FAflow1 $p fa2dt_01_b;     # switch to
    transport problem for BCs
873:                    }
874:            }
875:
```

The grid is adapted between MAXLEVEL and TOPLEVEL each time step. The refinement indicator is reinitialised after the first step [863]. Since the grid changes the flow needs to be re-solved each time step. The interpolation of heads to the new grid serves as an initial guess for the flow solution. The problem is configured to the boundary conditions of the flow problem [867], and the problem is assembled [868] and solved [869]. The problem is now configured to the transport problem ready for the next step. Two transport problems are defined in the C code allowing boundary conditions to change at 25 years. All the flow and transport problems must be defined on the same domain for consistency.

```
876:    # solve transport on new grid for next time step#####################
877:            # time solver
878:            if (STEP==1) npexecute ts $bdf1;
879:            else  {
880:                    if (TIME >= END - DT_LAST*DTSCALE ) {   #adjust dt to reach endtime exactly
881:                            DT = END - TIME;
882:                            npexecute ts $dt @DT $bdf1 ;
883:                            }
884:                    else {
885:                            if (BVP_SWITCH==1) {       #set small dt after boundary change
886:                                    DT = DTMIN*17;
887:                                    npexecute ts $dt @DT $bdf1 ;
888:                                    BVP_SWITCH= 2;
889:                            }
```

```
890:                            else    npexecute ts  $bdf1 ;
891:                    }
892:            }
893:            # bdf1 updates TIME script variable automatically
894:            TIME_YEAR = TIME/YEAR;
895:            DT_LAST = (TIME - TIME_OLD) ;
896:            DT_LAST_YEAR = DT_LAST/YEAR;
897:            TIME_OLD = TIME;
898:            readclock; TOTAL_SOLVER_RUNTIME = TOTAL_SOLVER_RUNTIME + CLOCK;
899:            print "END OF TIMESTEP: ", STEP, " newTIME: ", @TIME_YEAR, " lastDT: ",
       @DT_LAST_YEAR, " STEP RUNTIME: ", CLOCK;
900:
```

Solve the transport problem by time stepping. Normally bdf1 is executed [890]. Alternatively, if the boundary condition has just changed dt can be reset to a small value using the $dt option [887], or dt can be adjusted to reach END time precisely [882]. BDF will increase DT by multiplying by DTSCALE following succesful solution, and will reduce DT by half if the maximum number of newton iterations, set to 15 [707], is used without convergence. DT is maintained within the limits DTMIN and DTMAX [715]. The program uses time units of seconds and writes the updated value of TIME to the script variable (see file UG/ug/np/procs/bdf.c). This is converted to time in years for output purposes [894].

```
901:            minmax sol;
902:            @plotC;
903:            @plotCprofile;
904:            @frame;
905:            @concfile;
906:            @outputtecplot;
907:            npexecute box $mass $time @TIME_YEAR;
908:
```

Output varicus data to the log file and tecplot files, and plot graphics. The minmax sol [901] command lists all the species concentrations in terms of their minimum and maximum values and can be useful to spot errors at runtime. The global species masses are printed to the log file [907]. This can also be printed following the grid adaption to check for mass errors.

```
909:            if (STEP==STEPS) break;
910:            STEP = STEP+1;
911:
912:            if (BVP_SWITCH==0)
913:            if (TIME_YEAR>=25) {
914:                    if (SOLVEFLOW) configure FAflow1 $p fa2dt_01_b;   # switch to later bvp
915:                    else           configure fa2dt_01 $p fa2dt_01_b;
916:                    BVP_SWITCH=1;
917:                    }
918:
919:            readclock; TOTAL_RUNTIME = TOTAL_RUNTIME + CLOCK;
920:
921:            if (TIME>=END) break;
922:    }
923:
```

The BVP_SWITCH is used to flag the change of boundary condition occuring at 25years.
Stop time loop when maximum STEPS or END time are reached.

```
924:    ##########################################################
925:    # Postprocessing
926:    ##########################################################
927:    @plotCprofile;
928:    minmax sol;
929:    #value sol $pos 200 20; # get some point values
930:
931:    end = ".end";
932:    set tecplotname  @tecname@end;
933:    @do_tecplot;
934:    if (SOLVEFLOW)    mytecplot @outdir@basename@DOT@FLOW@end@DOT@DAT
935:                                        $title @basename@FLOW
936:                                        $zone Flowfield
937:                                        $e darcy0 $s bf     $v q_x
938:                                        $e darcy1 $s bf     $v q_z
939:                                        $e nvalue $s sol_p $v head
940:                                        $e K      $s flow  $v K
941:                                        ;
942:
943:    if (conf:parallel==0) printconcfile sol $file @outdir@concsname
944:                                                $time @TIME_YEAR
945:                                                $from 0 0 $to 750 20 $dx 7.5 $dy 0.2;
946:
947:    heapstat;
948:            print "TOTAL_RUNTIME ", TOTAL_RUNTIME, " s";
949:            print "TOTAL_SOLVER_RUNTIME ", TOTAL_SOLVER_RUNTIME, " s";
950:
951:    ##########################################################
952:    #close log file
953:    ##########################################################
954:    logoff;
```

At the end the final results are output to tecplot [933], along with a selection of other outputs.

10.6 Example of 2D Transport Parameter c code

The following code is the parameter code used in the above script for transport. It provides functions which are required by the boxnlsys interface.
The example parameter source code file is called:
`UG/course/appl/fourash2d/example_2d_transport.c`
In order to run it as described above copy the files contents into:
`UG/course/appl/fourash2d/fourash_2d_transport_01.c`

```
1:  /****************************************************************************/
2:  /*                                                                          */
3:  /* File:      fourash_2d_transport_01.c                                     */
4:  /*                                                                          */
5:  /* Purpose:   example multiple species contaminant transport                */
6:  /*                                                                          */
7:  /* Author:    Ian Watson                                                    */
8:  /*            GPRG                                                          */
9:  /*            University of Sheffield                                       */
10: /*            i.watson@sheffield.ac.uk                                      */
11: /*                                                                          */
12: /* History:   14 NOV 2003 begin                                            */
13: /*                                                                          */
14: /* Remarks:                                                                 */
15: /*                                                                          */
16: /****************************************************************************/
17:
18:
19: /* system includes */
20: #include <stddef.h>
21: #include <math.h>
22: #include <stdlib.h>
23: #include <stdio.h>
24: #include <string.h>
25:
26: /* ug library */
27: #include "compiler.h"
28: #include "evm.h"        /* for V_COPY,M_COPY */
29: #include "gm.h"
30: #include "misc.h"
31: #include "cmdint.h"
32: #include "commands.h"  /* for GetCurrentMultigrid (Quickhack) */
33: #include "ugstruct.h"
34: #include "initug.h"
35: #include "scan.h"       /* for ReadArgvVecDesc */
36: #include "std_domain.h"
37: #include "general.h"
38: #include "udm.h"        /* for VECDATA_DESC */
39: #include "ugdevices.h"
40: #include "shapes.h"
41:
42: /* problem class library */
43: #include "transport.h"
44: #include "water.h"
45: #include "fvsources.h"
46:
47: /* RCS string */
48: static char RCS_ID("$Header: /home/cvsroot/course/appl/solute/s2.c,v 1.5
49:  2000/06/29 09:24:33 peter Exp $",UG_RCS_STRING);
50:
```

Up to now the file is similar to other UG files, with the problem class library header files being specific to the course applications.

```
51:
52: /****************************************************************************/
53: /****************************************************************************/
54: /* Chemical system:
55: two step phenol degradation via full range of TEAPs
56: */
57: /* units: [mol/m^3_h2o] for mobile, aqueous phase,                          */
58: /* implements a NP_NLS_TRANSPORT_PARAM                                      */
59: /****************************************************************************/
60: /****************************************************************************/
61:
62: #define SOLVEFLOW
63: #undef ONEDIM
64:
```

Here SOLVEFLOW is defined, but not ONEDIM – so the simulation will be fully 2D and solve the flow field.

```
65:
66: #define c6h6o    0
```

```
67:   #define o2       1
68:   #define no3_m1   2
69:   #define h2       3
70:   #define fe_p2    4
71:   #define mn_p2    5
72:   #define so4_m2   6
73:   #define hs_m1    7
74:   #define ch4      8
75:   #define ch3cooh  9
76:   #define h_p1     10
77:   #define hco3_m1  11
78:   #define oh_m1    12
79:   #define h2co3    13
80:   #define co3_m2   14
81:   /* define immobile phases last */
82:   #define feooh_s  15
83:   #define mno2_s   16
84:   #define fes_s    17
85:
86:   #define xfeoh     18
87:   #define xfeoh2_p1 19
88:   #define xfeo_m1   20
89:   #define xfeofe_p1 21
90:   #define xfeofeoh  22
91:   #define xfeomn_p1 23
92:
```

These compiler directive #defines allow the use of species names instead of numbers in defining reactions and boundary conditions – which is more transparent than using the numbers.

```
93:
94:   #define NSPECIES 24
95:   #define NMOBSPECIES 15
96:   /* must also define no. of mobile species in script*/
97:   /* the script variable sets the absolute max no. mobile species*/
98:   #define NNAPL         0
99:   #define NMINERALS     3
100:  #define NBACTERIA     0
101:  #define NSURFCOMPLEX  6
102:
103:  #define NRATES 8
104:
```

Set compiler directives for numbers of species and number of each type of species, and the number of kinetic reactions used.

```
105:  /*known concentrations [mol/m3] for ionic strength calculation*/
106:  #define C_ca_p2 2.4
107:  #define C_mg_p2 0.28
108:  #define C_na_p1 12.
109:  #define C_cl_m1 5.5
110:
111:  #define CSCALE 1. /*units conversion: 1. use mol/m^3, 1000. use mol/l*/
112:
113:
114:  #define PORO_WATER 0.125
115:  #define PORO_BIO   0.00    /*biofilm volume/bulk vol*/
116:  #define PORO_MAT   0.73    /*solid matrix vol / bulk vol*/
117:
118:  #define RHO_S 2644    /* rho_s [kg_solid/m^3_solid] */
119:
120:
```

Some constants which can be used later on.

```
121:  /*#define MONOD(S,K) (S/(S+K))*/
122:  /*#define INHIB(S,K) (K/(S+K))*/
123:
124:  typedef struct
125:  {
126:          NP_NLS_TRANSPORT_PARAM tp;              /* derive from this class          */
127:
128:          /* configuration */
129:          NP_WATER *flow;                             /* the flow module exporting velo
              */
130:
131:  } NP_FLOW_TRANSPORT;
132:
```

This is to extend the transport parameter class to include a flow parameter class as well.

```
133:
134:  static DOUBLE porosity (void)
135:  {
136:          return 0.125;
137:  }
138:
139:  /*normal Monod term*/
140:  static DOUBLE Monod (DOUBLE S, DOUBLE K)
141:  {
142:          return (S/(S+K));
143:  }
144:
145:  /*linearised Monod term in the neagative quadrant*/
146:  static DOUBLE Monod_a (DOUBLE S, DOUBLE K)
```

```
147:    {
148:            if (S>=0.0)
149:            return (S/(S+K));
150:            else
151:            return (S/K);
152:    }
153:
154:    /*Monod term returning ZERO in negative quadrant*/
155:    static DOUBLE Monod_b (DOUBLE S, DOUBLE K)
156:    {
157:            if (S>=0.0)
158:            return (S/(S+K));
159:            else
160:            return 0.0;
161:    }
162:
163:    /*Monod Threshold term to shut down reaction at low concentrations*/
164:    /*and avoid negative concentrations*/
165:    static DOUBLE Threshold (DOUBLE S, DOUBLE K, DOUBLE n)
166:    {
167:            return pow((S/(S+K)),n);
168:    }
169:
170:    /*regularise the double monod formulation when both <0.*/
171:    /*for use with two linearised Monod-a terms*/
172:    static DOUBLE reg_a (DOUBLE C1, DOUBLE C2)
173:    {
174:            if (C1<0.0 && C2<0.0)
175:            return -1.0;
176:            else
177:            return 1.0;
178:    }
179:
180:    /*Monod Inhibition term  linearised in the negative quadrant*/
181:    static DOUBLE Inhib_a (DOUBLE I, DOUBLE Ki)
182:    {
183:            if (I>=0.0)
184:            return (Ki/(I+Ki));
185:            else
186:            return (1.0 - I/Ki);
187:    }
188:
189:    /*Monod Inhibition term set to unity in the negative quadrant*/
190:    static DOUBLE Inhib_b (DOUBLE I, DOUBLE Ki)
191:    {
192:            if (I>=0.0)
193:            return (Ki/(I+Ki));
194:            else
195:            return 1.0 ;
196:    }
197:
```

Various functions developed for use in Monod kinetic biodegradation.

```
198:    /*Mineral equilibrium similar to method of Leeming & Mayer 1998*/
199:    static DOUBLE Mineral_equil ( DOUBLE IAP, DOUBLE Ksp, DOUBLE Cmineral,
200:                                  DOUBLE  min_IAP_species, DOUBLE weight)
201:    {
202:            if ( min_IAP_species < 1.e-9) /*deal with -ve aqueous ion concs*/
203:                    return  min_IAP_species -  1.e-7;
204:            else
205:            {
206:                    if (Cmineral < -1.e-9) {  /*deal with -ve mineral concs*/
207:                            return  Cmineral - 1.e-13 ;
208:                    }
209:                    else
210:                            if (Cmineral > 1.e-9) /*for equil pptn or dissln*/
211:                                    return  (IAP / Ksp  -1.0)*weight;
212:                            else
213:                            {
214:                                    if ( IAP/Ksp  > 1.0) /*for precipitation*/
215:                                            return  (IAP / Ksp  -1.0)*weight;
216:                                    else { /*keep small min conc when no min. and
    undesaturated*/
217:                                            return  Cmineral - 1.e-13 ;
218:                                    }
219:                            }
220:            }
221:    }
222:
```

This function returns an appropriate value for the algebraic equation of mineral precipitation or dissolution. There are several possible cases, noted in the comments, each return value is set to zero. Essentially any negative value is set to a small positive value, then if mineral is present precipitation or dissolution to equilibrium occurs, or if there is no mineral, only precipitation is allowed.

```
223:
224:
```

```
225:    static INT Umatrix (NP_NLS_TRANSPORT_PARAM *tp, DOUBLE *U)
226:    {
227:
228:    /* to define the U matrix:
229:    go down each species column j and enter the stoichiometric coefficient.
230:    for components there will only be one 1 in its column
231:    NB secondary species have all zeros on their row
232:    the diagonal will be 1 for component, or 0 for secondary sp.
233:    kinetic reactions must be written entirely in terms of primary components.
234:    for each equilibrium reaction a secondary species can be written
235:        entirely in terms of other primary components
236:
237:    see equilibrium/equil05.c for example with equilibrium calcite dissolution*/
238:
239:
240:            int i,j;
241:            /*set up assuming all are primary components*/
242:            for (i=0; i<NSPECIES; i++)
243:                    for (j=0; j<NSPECIES; j++)
244:                            if (i==j) U[i*NSPECIES+j] = 1.0;
245:                            else      U[i*NSPECIES+j] = 0.0;
246:
247:            /* treat secondary species columns*/
248:            /*oh-*/
249:            U[oh_m1*NSPECIES+ oh_m1]  = 0.0;
250:            U[h_p1*NSPECIES+ oh_m1]  = -1.0;
251:
252:            /*h2co3*/
253:            U[h2co3   *NSPECIES+ h2co3]   = 0.0;
254:            U[h_p1    *NSPECIES+ h2co3]   = 1.0;
255:            U[hco3_m1 *NSPECIES+ h2co3]   = 1.0;
256:
257:            /*co3-2*/
258:            U[co3_m2  *NSPECIES+ co3_m2] =  0.0;
259:            U[h_p1    *NSPECIES+ co3_m2] = -1.0;
260:            U[hco3_m1 *NSPECIES+ co3_m2] =  1.0;
261:
262:            /*fes*/
263:            U[fes_s   *NSPECIES+ fes_s] =  0.0;
264:            U[h_p1    *NSPECIES+ fes_s] = -1.0;
265:            U[hs_m1   *NSPECIES+ fes_s] =  1.0;
266:            U[fe_p2   *NSPECIES+ fes_s] =  1.0;
267:
268:            /*xfeoh2_p1*/
269:            U[xfeoh2_p1 *NSPECIES+ xfeoh2_p1] =  0.0;
270:            U[h_p1      *NSPECIES+ xfeoh2_p1] =  1.0;
271:            U[xfeoh     *NSPECIES+ xfeoh2_p1] =  1.0;
272:
273:            /*xfeo_m1*/
274:            U[xfeo_m1   *NSPECIES+ xfeo_m1] =  0.0;
275:            U[h_p1      *NSPECIES+ xfeo_m1] = -1.0;
276:            U[xfeoh     *NSPECIES+ xfeo_m1] =  1.0;
277:
278:            /*xfeofe_p1*/
279:            U[xfeofe_p1 *NSPECIES+ xfeofe_p1] =  0.0;
280:            U[h_p1      *NSPECIES+ xfeofe_p1] = -1.0;
281:            U[xfeoh     *NSPECIES+ xfeofe_p1] =  1.0;
282:            U[fe_p2     *NSPECIES+ xfeofe_p1] =  1.0;
283:
284:            /*xfeofeoh*/
285:            U[xfeofeoh  *NSPECIES+ xfeofeoh] =  0.0;
286:            U[h_p1      *NSPECIES+ xfeofeoh] = -2.0;
287:            U[xfeoh     *NSPECIES+ xfeofeoh] =  1.0;
288:            U[fe_p2     *NSPECIES+ xfeofeoh] =  1.0;
289:
290:            /*xfeomn_p1*/
291:            U[xfeomn_p1 *NSPECIES+ xfeomn_p1] =  0.0;
292:            U[h_p1      *NSPECIES+ xfeomn_p1] = -1.0;
293:            U[xfeoh     *NSPECIES+ xfeomn_p1] =  1.0;
294:            U[mn_p2     *NSPECIES+ xfeomn_p1] =  1.0;
295:
296:            /*feco3_s*/
297:    /*      U[feco3_s   *NSPECIES+ feco3_s] =  0.0;*/
298:    /*      U[h_p1      *NSPECIES+ feco3_s] = -1.0;*/
299:    /*      U[hco3_m1   *NSPECIES+ feco3_s] =  1.0;*/
300:    /*      U[fe_p2     *NSPECIES+ feco3_s] =  1.0;*/
301:
302:            /*mnco3_s*/
303:    /*      U[mnco3_s   *NSPECIES+ mnco3_s] =  0.0;*/
304:    /*      U[h_p1      *NSPECIES+ mnco3_s] = -1.0;*/
305:    /*      U[hco3_m1   *NSPECIES+ mnco3_s] =  1.0;*/
306:    /*      U[mn_p2     *NSPECIES+ mnco3_s] =  1.0;*/
307:
308:
309:    /*ensure secondary equilibrium species rows are zeros*/
310:    /*each secondary species must be assigned an algebraic equation*/
311:    /*ie mass action law in the source parameter, below.*/
312:    /*must also ensure that initial / boundary conditions satisfy algebraic equation.*/
313:
314:            return 0;
```

```
315:    }
316:
```

The U matrix describes how mass should be taken from secondary equilibrium species and placed into primary components. Some simple rules are given above. The U matrix is utilised in the following Adsorption (mass storage) function, and also in the transport algorithm in file boxnlsys.c. The concept is that if we have two species to solve for, e.g. H+ and OH-, we need two equations. One equation, applying the local equilibrium asumption, is the mass action law, e.g. $OH^- * H^+=10^{-14}$. The other is a mass conservation law which needs to conserve the mass of both H^+ and OH^-. Since stoichiometrically $OH^- = H_2O - H^+$ the mass of OH- can be redistributed as H_2O (often not considered since there is so much water in the water phase) and H^+. Therefore, each primary component becomes a sum of its own species and the secondary species of which it constitutes a part. Thus: $TOTH^+ = H^+ - OH^-$. This is explained in text books such as Stumm & Morgan.

The i^{th} row of the Umatrix can be found in the above code which is given as U[i*NSPECIES+j]. For example, all the secondary species specified above are summed into the $TOTH^+$ component, so they all provide an entry on the row for h_p1, and all have an entry beginnning "U[h_p1 ...". The rows for secondary species, e.g. i=OH^-, should be all zero since the mass must be moved out of the secondary species and into primary components.

```
317:
318:    static INT Adsorption (NP_NLS_TRANSPORT_PARAM *tp,
319:                                                          const ELEMENT *e, DOUBLE_VECTOR
        *corners,
320:                                                          DOUBLE *local, DOUBLE *global,
321:                                                          DOUBLE t, INT sd, INT nc,
        DOUBLE *C, DOUBLE *res)
322:    {
323:            /* give the mass storage term                                    */
324:            /* expected UNITS are [mol/m^3_bulk]                             */
325:            /* e.g. for C_aqueous with units [mol/l_h2o]                     */
326:            /* the mass term is d(poro.Rf.C)/dt so give (poro.Rf.C)          */
327:            /* Rf is retardation factor (e.g.due to linear sorption) [-]     */
328:            /* so give (  C.       1000.      poro        .    Rf )          */
329:            /*          [mol/l_h2o] [l/m^3] [m^3_h2o/m^3_bulk] [-]           */
330:            /*                                                               */
331:            /* e.g. for C_solid with units [mol/kg_solid]                    */
332:            /* d(rho_s(1-poro)C)/dt  -> give (rho_s(1-poro)C)                */
333:            /*     rho_s           *    (1-poro)          *      C           */
334:            /* [kg_solid/m^3_solid]*[m^3_solid/m^3_bulk]*[mol/kg_solid]      */
335:            /*                                                               */
336:            /* e.g. for C_ionexchanged  with units [meq/100g_solid]                  */
337:            /* d(rho_s(1-poro)C/100z)/dt  -> give (rho_s(1-poro)C/100z)              */
338:            /*   C   *        10 *    1/z * 1/1000      *   rho_s   *    (1-poro)    */
339:            /* [meq/100g_s] [100g/kg] [mmol/meq] [mol/mmol] [kg_s/m^3_s] [m^3_s/m^3_bulk] */
340:
341:
342:            INT i,j, row_i;
343:            DOUBLE U[NSPECIES*NSPECIES];
344:
345:            (tp->Umatrix)(tp,U);
346:
347:            /* multiply by U to get primary components TOTALS */
348:
349:            for (i=0;i<NSPECIES ;i++) /*loop over each component row in turn*/
350:            {
351:              row_i = i*NSPECIES;
352:              res[i] = 0.0;
353:          /*storage for mobile aqueous species, C_aq [mol/m^3_h2o if CSCALE=1]*/
354:              for (j=0;j<NMOBSPECIES ;j++)
355:                    if ( U[row_i+j] != 0.0)
356:                            res[i] += PORO_WATER * CSCALE * C[j]*U[row_i+j];
357:
358:              /*storage for immobile napl species, C_napl [mol/m^3solids]*/
359:              for (j=NMOBSPECIES  ;j<NMOBSPECIES+NNAPL ;j++)
360:                    if ( U[row_i+j] != 0.0)
361:                            res[i] += PORO_MAT * C[j]*U[row_i+j];
362:
363:              /*storage for immobile mineral species, C_min [mol/m^3solids]*/
364:              for (j=NMOBSPECIES+NNAPL  ;j<NMOBSPECIES+NNAPL+NMINERALS ;j++)
365:                    if ( U[row_i+j] != 0.0)
366:                            res[i] += RHO_S * PORO_MAT * C[j]*U[row_i+j];
367:
368:              /*storage for immobile surface complexation species, C_surf [mol/m^3_h20]*/
369:              for (j=NMOBSPECIES+NNAPL+NMINERALS ;j<NMOBSPECIES+NNAPL+NMINERALS+NSURFCOMPLEX
        ;j++)
370:                  if ( U[row_i+j] != 0.0)
371:                            res[i] += PORO_WATER * C[j]*U[row_i+j];
372:
373:
374:              /*storage for immobile biomass species [mol/m^3]*/
375:    /*        for (j=NMOBSPECIES+NNAPL+NMINERALS ;j<NSPECIES ;j++)*/
376:    /*              if ( U[row_i+j] != 0.0)*/
```

```
377:    /*                          res[i] += PORO_BIO * C[j]*U[row_i+j];*/
378:
379:              /*storage for immobile ion exchange species [meq/100g_solid]*/
380:    /*        for (j=NMOBSPECIES+NMINERALS  ;j<NSPECIES ;j++)*/
381:    /*              if ( U[row_i+j] != 0.0)*/
382:    /*                    res[i] += RHO_S*(1.0-porosity() ) * C[j]*U[i*NSPECIES+j]
        /(100.*z(j));*/
383:
384:            }
385:
386:            return 0;
387:    }
388:
389:
```

This Adsorption function represents the mass storage term in the governing mass conservation equation. The above loops proceed through each type of species, aqueous, mineral, surface complexes, etc, and use the U matrix to add appropriate mass into the primary components. The units of each type of species are always converted into mol/m^3_bulk for consistency in the summation. With this looping setup it is important that the species list given above has the species grouped according to type, as well as placing the mobile species first. For example line [369] assumes that the species indices for surface complexes follow on from the species indices of the minerals. If new species types are added then the looping here will need alteration.

```
390:
391:    static INT Darcy (NP_NLS_TRANSPORT_PARAM *tp, const ELEMENT *e,
392:                      DOUBLE_VECTOR *corners, DOUBLE *local, DOUBLE *global,
393:                      INT subdomain, INT m, DOUBLE *jw)
394:    {
395:    #ifdef SOLVEFLOW
396:            NP_FLOW_TRANSPORT *ft = (NP_FLOW_TRANSPORT *) tp;
397:    #endif
398:
399:            /* set to zero for immobile phases */
400:            if (m>NMOBSPECIES-1)  /*m is species index starting at 0 */
401:            {
402:                    jw[0]=jw[1]=0.;   /* for DIM 2 */
403:                    return 0;
404:            }
405:
406:    #ifdef SOLVEFLOW
407:            /*fetch flow solver results*/
408:            (ft->flow->Darcy)(ft->flow,e,corners,local,jw);
409:    #else
410:            /*water flux jw is not calculated, but given as permeability *head gradient
411:            8.4e-6*4.15e-3  = 3.528e-8 m/s   ie q=1.113m/yr  (v is higher, ~8m/yr)*/
412:            /*give darcy flux*/
413:            jw[0]= 3.4776e-8 ;
414:            jw[1]= 0.0 ;
415:    #endif
416:
417:            /*CSCALE is used to multiply by 1000 to convert Concentration from mol/m^3 to
        mol/l*/
418:    /*      jw[0]=   jw[0]* CSCALE;
419:            jw[1]=   jw[1]* CSCALE;
420:    */
421:
422:            return 0;
423:    }
424:
```

This function gives the Darcy flux (not the solute velocity). The mass flux of water can be given as a vector [413-414]. If the flow is solved (e.g. in a heterogeneous K field) the flow can be retrieved from the flow solver [408]. This set up requires lines [396], [124-131], [898-903] and in the script [683]. Immobile species have zero velocity.

Note to use units of mol/l_h2o for the aqueous phase CSCALE must be applied to the advection and dispersion terms as well as the storage and reaction terms.

```
425:
426:    static INT Dispersion (NP_NLS_TRANSPORT_PARAM *tp, const ELEMENT *e,
427:                           DOUBLE_VECTOR *corners, DOUBLE *local, DOUBLE *global,
428:                           INT sd, INT m, DOUBLE *out)
429:    {
430:    #ifdef SOLVEFLOW
431:            NP_FLOW_TRANSPORT *ft = (NP_FLOW_TRANSPORT *) tp;
432:    #endif
433:        DOUBLE diffmol;
434:            DOUBLE u[DIM],uabs,uabs2;
435:            DOUBLE aL,aT;
436:            DOUBLE e00,e01,e11;
437:
438:            /* set to zero for immobile phases */
439:            if (m>NMOBSPECIES-1)
440:            {
441:                    out[0] = 0.0;   out[1] = 0.0;
442:                    out[2] = 0.0;    out[3] = 0.0;
```

```
443:                    return 0;
444:            }
445:
446:            /* set D for mobile, aqueous species */
447:            /* get data */
448:            if (m==0) diffmol = 0.0 *PORO_WATER;  /* eff. pore diffusion  [m^2/s]  */
449:            else      diffmol = 0.0 *PORO_WATER;   /* eff. pore diffusion  [m^2/s]  */
450:
451:
452:            aL = 1.0;                /* dispersivity longitudinal 0.01m, in m  */
453:            aT = 0.0004;             /* dispersivity transverse vertical ,0.4mm in m  */
454:    #ifdef ONEDIM
455:            aT = 0.0;
456:    #endif
457:
458:    #ifdef SOLVEFLOW
459:            /* get velocity */
460:            (ft->flow->Darcy)(ft->flow,e,corners,local,u);
461:    #else
462:            /* give darcy velocity */
463:            u[0]= 3.4776e-8 ;
464:            u[1]= 0.0       ;
465:    #endif
466:
467:            uabs2 = u[0]*u[0]+u[1]*u[1];
468:            uabs = sqrt(uabs2);
469:            if (uabs>0.0)
470:            {
471:                    e00 = u[0]*u[0]/uabs2;
472:                    e01 = u[0]*u[1]/uabs2;
473:                    e11 = u[1]*u[1]/uabs2;    }
474:            else e00=e01=e11=0.0;
475:
476:            /* fill tensor */
477:        out[0] = diffmol + aT*uabs + uabs*(aL-aT)*e00;
478:        out[1]=out[2] =              uabs*(aL-aT)*e01;
479:        out[3] = diffmol + aT*uabs + uabs*(aL-aT)*e11;
480:
481:            /*multiply by 1000 to convert aqueous mobile Concentration from mol/l to mol/m^3*/
482:    /*      out[0] = out[0] *CSCALE;
483:            out[1] = out[1] *CSCALE;
484:            out[2] = out[2] *CSCALE;
485:            out[3] = out[3] *CSCALE;
486:    */
487:            return 0;
488:    }
489:
```

This Dispersion function calculates the 2D hydrodynamic dispersion tensor using the Darcy flux, molecular diffusion and longitudinal and transverse dispersivities.

```
490:
491:
492:    static INT Source (NP_NLS_TRANSPORT_PARAM *tp, const ELEMENT *e,
493:                       DOUBLE_VECTOR *corners, DOUBLE *local, DOUBLE *global,
494:                                                DOUBLE t, INT sd, INT nc, DOUBLE *C,
        DOUBLE *res)
495:    {
496:            /* Enter the source/sink terms including rate term  */
497:            /* positive values imply mass is lost  */
498:            /* units require [mol/m^3_bulk/s]  */
499:            /* remember to include porosity   */
500:            /* remember to convert from and to mol/litre/s   */
501:            DOUBLE s[NSPECIES][NRATES];  /* stoichiometric coeffiecients*/
502:            DOUBLE rate[NRATES];
503:            DOUBLE U[NSPECIES*NSPECIES];
504:            DOUBLE Ionic_Strength,sqrt_I,Davies_Term;
505:            DOUBLE GammaZ1,GammaZ2,GammaZ3;
506:            DOUBLE activity[NSPECIES];
507:            DOUBLE MassAction, IAP, Ksp, min_IAP_species;
508:            INT sp_index, plumefront;
509:            INT i,j;
510:            DOUBLE factor;
511:
512:            for (i=0;i<NSPECIES ;i++)
513:              for (j=0;j<NRATES ;j++)
514:                    s[i][j] = 0.0;        /*set stoichiometric matrix to zero*/
515:            /* stoichiometric convention is for mass lost to have negative coefficient */
516:
517:            /*KINETIC REACTIONS must be written in terms of components*/
518:
519:            /*      test if x>v(t-Tchange) for if at front or back of plume*/
520:            if (global[0] > (3.4776e-8/0.125*(t - 25.0*3.1536E7)) ) plumefront = 1;
521:            else                                                    plumefront = 0;
522:
```

The source function determines the source-sink term. It needs to return the result with the rate with units of $mol/m^3_{bulk}/s$ to be consistent with the previous functions for adsorption and transport. Kinetic rates are dealt with first. Often kinetic rate parameters for degradation have

units such as mol/m^3_{h2o}/s, and if this is the case a suitable porosity or volume fraction term will be necessary to convert the units to the required ones. The function arguments include the global coordinates of the point being evaluated and the current simulated time [492-494]. This information has been used in this example to test whether the evaluation point is in the front or rear of the plume using an assumed velocity together with the time and x position of the point. The resulting plumefront flag is used in subsequent reaction definitions to give different rates depending on position within the moving plume.

```
523:
524:            j=0;  /*reaction index (from 0 to NRATES-1 ) */
525:            /* reaction :  aerobic phenol oxidation */
526:            /* c6h6o + 7.o2 + 3.h2o -> 6.hco3- + 6.h+     */
527:            s[c6h6o][j]= -1.;     s[o2][j] =  -7.;
528:            s[hco3_m1][j]= 6.;    s[h_p1][j]= 6.;
529:            /*       Monod kinetics wrt c6h6o  & o2  */
530:            rate[j]= 1.0      *PORO_WATER          /*[m^3_h2o/m^3_bulk]*/
531:                                                  *3.9811E-10 *1000.     /* k_max = 10^-9.4
        mol/l_h20/s [mol/m^3_h2o/s] */
532:                                                  *Monod_a(C[c6h6o] *0.001,1.064E-4 )
533:                                                  *Monod_a(C[o2]     *0.001,3.125E-6 )
534:                                                  *reg_a (C[c6h6o],C[o2] )
535:                                                  *Threshold(C[c6h6o] *0.001 , 1.0E-10 , 2. )
536:                                                  *Threshold(C[o2]    *0.001 , 3.125E-7 , 2. )
537:                                                  ; /*  [mol/m3_bulk/s] */
538:
539:
540:            j=1;
541:            /* reaction : phenol denitrification */
542:            /*  5.c6h6o + 28.no3- + h2o -> 30.hco3- + 2.h+  + 14.n2  */
543:            s[c6h6o][j]= -1.;  s[no3_m1][j]=-28./5.;
544:            s[hco3_m1][j]= 30./5.;   s[h_p1][j]= 2./5.; /*s[n2][j]= 14./5.;*/
545:            /*       Monod kinetics wrt c6h6o  & no3-  */
546:            rate[j]= 1.0      *PORO_WATER          /*[m^3_h2o/m^3_bulk]*/
547:                                                  *3.9811E-11 *1000.    /*k_max = 10^-10.4
        mol/l_h20/s [mol/m^3_h2o/s] */
548:                                                  *Monod_a(C[c6h6o] *0.001, 1.064E-4 )
549:                                                  *Monod_a(C[no3_m1]*0.001, 8.064E-6 )
550:                                                  *Inhib_a(C[o2]     *0.001, 6.2E-6 )
551:                                                  *reg_a (C[c6h6o],C[no3_m1] )
552:                                                  *Threshold(C[c6h6o] *0.001 , 1.E-10 , 2. )
553:                                                  *Threshold(C[no3_m1]*0.001 , 8.064E-7 , 2. )
554:                                                  ; /*  [mol/m3_bulk/s] */
555:
556:
557:
558:            j=2;
559:            /* reaction : phenol fermentation to acetate ch3cooh and h2 (no co2) */
560:            /*  c6h6o + 5.h2o  -> 3ch3cooh + 2h2                    */
561:            s[c6h6o][j]= -1.;
562:            s[ch3cooh][j]= 3.;    s[h2][j]= 2.;
563:
564:            /*       test if x>v(t-Tchange) for if at front or back of plume*/
565:            if (plumefront) factor = 1.0;
566:            else            factor = 3.2;
567:
568:            /*       Monod kinetics wrt c6h6o  */
569:            rate[j]= factor  *PORO_WATER          /*[m^3_h2o/m^3_bulk]*/
570:                                                  *8.4e-13 * 1000.     /*k_max  [mol/l/s]  */
571:                                                  *Monod_a(C[c6h6o] *0.001, 1.1e-4 )
572:                                                  *Inhib_a(C[o2]     *0.001, 3.1e-5 )
573:                                                  *Inhib_a(C[no3_m1]*0.001, 1.6E-5 )
574:                                                  *Inhib_a(C[c6h6o]     *0.001, 6.0e-2)
575:                                                  *Threshold(C[c6h6o] *0.001 , 1.e-5 , 2. )
576:                                                  ; /*  [mol/m3_bulk/s] */
577:
578:            j=3;
579:            /* reaction : phenol fermentation to co2 and h2 (no acetate) */
580:            /*  c6h6o + 17.h2o  -> 6.hco3-  + 14h2 + 6.h+                  */
581:            s[c6h6o][j]= -1.;
582:            s[hco3_m1][j]= 6.;    s[h2][j]= 14.; s[h_p1][j]= 6.;
583:
584:            /*       test if x>v(t-Tchange) for if at front or back of plume*/
585:            if (plumefront) factor = 0.07 *3.6;
586:            else            factor = 0.3;
587:
588:            /*       Monod kinetics wrt c6h6o  */
589:            rate[j]= factor  *PORO_WATER          /*[m^3_h2o/m^3_bulk]*/
590:                                                  *8.4e-13 * 1000.     /*k_max  [mol/l/s]  */
591:                                                  *Monod_a(C[c6h6o] *0.001, 1.1e-4 )
592:                                                  *Inhib_a(C[o2]     *0.001, 3.1e-5 )
593:                                                  *Inhib_a(C[no3_m1]*0.001, 1.6E-5 )
594:                                                  *Inhib_a(C[c6h6o]     *0.001, 6.0e-2)
595:                                                  *Threshold(C[c6h6o] *0.001 , 1.e-5 , 2. )
596:                                                  ; /*  [mol/m3_bulk/s] */
597:
598:
599:            j=4;
```

```
600:          /* reaction : ironIII goethite reduction with hydrogen, h2 */
601:          /*  h2 + 2feooh + 4.h+ -> 2fe2+ +4h2o                    */
602:          s[h2][j]= -1.;   s[feooh_s][j]= -2.; s[h_p1][j]= -4.;
603:          s[fe_p2][j]= 2.;
604:
605:          /*      test if x>v(t-Tchange) for if at front or back of plume*/
606:          if (plumefront) factor = 0.3 * 4.;
607:          else            factor = 0.8 * 1.;
608:
609:          /*      Monod  kinetics wrt h2 and feooh  */
610:          rate[j]= factor  *PORO_WATER        /*[m^3_h2o/m^3_bulk]*/
611:                                          *1.8e-11 *1000.   /*k_max = 10^-12.5 mol/l_h20/s
      [mol/m^3_h2o/s] */
612:                                          *Monod_a(C[h2] *0.001, 5.e-7 )
613:                                          *Inhib_a(C[o2]    *0.001, 3.125E-5 )
614:                                          *Inhib_a(C[no3_m1]*0.001, 1.6E-5 )
615:                                          *Inhib_a(C[c6h6o] *0.001, 4.0e-2 )
616:                                          *Threshold(C[h2] *0.001 , 1.E-10 , 2. )
617:                                          ; /*  [mol/m3_bulk/s] */
618:
619:
620:          j=5;
621:          /* reaction : MnIV pyrolusite reduction with hydrogen, h2 */
622:          /*  h2 + mno2 + 2.h+ -> mn2+ + 2h2o                     */
623:          s[h2][j]= -1.;   s[mno2_s][j]= -1.; s[h_p1][j]= -2.;
624:          s[mn_p2][j]= 1.;
625:
626:          /*      test if x>v(t-Tchange) for if at front or back of plume*/
627:          if (plumefront) factor = 0.7*1.0;
628:          else            factor = 0.3*1.0;
629:
630:          /*      Monod kinetics wrt h2 and mno2  */
631:          rate[j]= factor  *PORO_WATER        /*[m^3_h2o/m^3_bulk]*/
632:                                          *1.4E-11 *1000.   /*k_max =  mol/l_h20/s
      [mol/m^3_h2o/s] */
633:                                          *Monod_a(C[h2]*0.001, 5.e-7 )
634:                                          *Inhib_a(C[o2]    *0.001, 3.125E-5 )
635:                                          *Inhib_a(C[no3_m1]*0.001, 1.6E-5 )
636:                                          *Inhib_a(C[c6h6o] *0.001, 4.0e-2 )
637:                                          *Threshold(C[h2] *0.001 , 1.E-10 , 2. )
638:                                          ; /*  [mol/m3_bulk/s] */
639:
640:          j=6;
641:          /* reaction : sulphate reduction with hydrogen, h2 */
642:          /*  4h2 + so4+2 + h+ -> hs- +4h2o                     */
643:          s[h2][j]= -1.;   s[so4_m2][j]= -0.25; s[h_p1][j]= -0.25;
644:          s[hs_m1][j]= 0.25;
645:
646:          /*      test if x>v(t-Tchange) for if at front or back of plume*/
647:          if (plumefront) factor = 0.05;
648:          else            factor = 1.0;
649:
650:          /*      Monod kinetics wrt h2 and so4-2  */
651:          rate[j]= factor  *PORO_WATER        /*[m^3_h2o/m^3_bulk]*/
652:                                          *1.55e-10 *1000.   /*k_max =  mol/l_h20/s
      [mol/m^3_h2o/s] */
653:                                          *Monod_a(C[h2] *0.001, 1.0e-6 )
654:                                          *Monod_a(C[so4_m2]*0.001, 1.6E-3 )
655:                                          *Inhib_a(C[o2]    *0.001, 3.125E-5 )
656:                                          *Inhib_a(C[no3_m1]*0.001, 1.6E-5 )
657:                                          *Inhib_a(C[c6h6o] *0.001, 4.0e-2 )
658:                                          *reg_a  (C[h2],C[so4_m2] )
659:                                          *Threshold(C[c6h6o] *0.001 , 1.E-10 , 2. )
660:                                          *Threshold(C[so4_m2]*0.001 , 1.6E-5 , 2. )
661:                                          ; /*  [mol/m3_bulk/s] */
662:
663:
664:          j=7;
665:          /* reaction : methanogenesis with hydrogen, h2 */
666:          /*  4h2 + hco3- + h+ -> ch4 + 3h2o                    */
667:          s[h2][j]= -1.;   s[hco3_m1][j]= -0.25; s[h_p1][j]= -0.25;
668:          s[ch4][j]= 0.25;
669:
670:          /*      test if x>v(t-Tchange) for if at front or back of plume*/
671:          if (plumefront) factor = 0.5;
672:          else            factor = 15.0;
673:
674:          /*      Monod kinetics wrt h2 and hco3-  */
675:          rate[j]= factor  *PORO_WATER        /*[m^3_h2o/m^3_bulk]*/
676:                                          *5.5e-11 *1000.   /*k_max  mM/s*/
677:                                          *Monod_a(C[h2] *0.001, 5.E-6 )
678:                                          /**Monod_a(C[hco3_m1]*0.001, 2e-5 )*/
679:                                          *Inhib_a(C[o2]    *0.001, 3.1e-5 )
680:                                          *Inhib_a(C[no3_m1]*0.001, 1.6E-5 )
681:                                          /**Inhib_a(C[so4_m2]    *0.001, 1.e-3 )*/
682:                                          *Inhib_a(C[c6h6o]     *0.001, 4.0e-2 )
683:                                          /**reg_a (C[h2],C[hco3_m1] )*/
684:                                          *Threshold(C[h2] *0.001 , 1.E-10 , 2. )
685:                                          /**Threshold(C[hco3_m1]*0.001 , 2.e-7 , 2. )*/
686:                                          ; /*  [mol/m3_bulk/s] */
```

```
687:
688:
```

Each of the 8 kinetic reactions are specified in a similar way. The reaction index, j, is first specified, then the stoichiometry of the reaction is defined using the stoichiometric matrix s_{ij} where i refers to species index i. Generally s_{ij} should be 1 for for species for which the rate is defined. Next the jth rate is defined using various Monod, Threshold and Inhibition terms as defined at the start of this file. Remember to convert concentrations into mol/l_h2o to be consistent with units for Monod half saturation coeffiecients.

```
689:    /* calculate sink term for each component..*/
690:    /* assumes that all rate units above are [mol/m3_bulk/s] */
691:    /*   ie aqueous rates[mol/l_h2o/s] should have been multiplied by porosity
692:            to convert [mol/m3_bulk/s] */
693:
694:    for (i=0;i<NSPECIES ;i++)   /*loop round species  */
695:      {
696:      res[i]=0.0;
697:      for (j=0;j<NRATES ;j++)
698:        res[i]=res[i] - s[i][j]*rate[j];
699:      }
```

Now the source term for each species, i, is computed by summing all the contributions from each of the j reactions. For the secondary species the result, res[i], should remain as zero. This will be the case provided all the stoichiometries were defined in terms of components only.

```
700:
701:            /*EQUILIBRIUM relationships*/
702:
703:            /*calculate Ionic Strength for activity corrections*/
704:            /*I=0.5*SUM(C_i*z_i*z_i)    [mol/l] */
705:            Ionic_Strength = 0.5*(  1.0*(   C[no3_m1]      /*monovalent*/
706:                                          + C[hs_m1]
707:                                          + C[h_p1]
708:                                          + C[hco3_m1]
709:                                          + C[oh_m1]
710:                                          + C_na_p1
711:                                          + C_cl_m1    )
712:                                    + 4.0*(   C[fe_p2]        /*divalent*/
713:                                          + C[mn_p2]
714:                                          + C[so4_m2]
715:                                          + C[co3_m2]
716:                                          + C_ca_p2
717:                                          + C_mg_p2    )
718:                                                   )*0.001;     /*conversion to mol/l*/
719:
720:            sqrt_I = sqrt(Ionic_Strength);
721:
722:            /*calculate activity coefficients for different ion charges*/
723:            /*use Davies equation  page 103 Stumm & Morgan*/
724:            /*log(gamma)= -Az^2 (I^0.5 /(1+I^0.5) - 0.2I)*/
725:            /*A = 1.82e6.(eT)^-3/2 = ~ 0.5 for water 25C*/
726:            Davies_Term = -0.5* (   sqrt_I / (1.0 + sqrt_I )
727:                            - 0.2*Ionic_Strength        );
728:
729:            /*now introduce charge z^2 to give log gamma*/
730:            GammaZ1 =  1.0  *Davies_Term;
731:            GammaZ2 =  4.0  *Davies_Term;
732:            /*GammaZ3 =  9.0  *Davies_Term;*/
733:
734:            GammaZ1 =  pow(10.0, GammaZ1 );
735:            GammaZ2 =  pow(10.0, GammaZ2 );
736:            /*GammaZ3 =  pow(10.0, GammaZ3 );*/
737:
738:            /*use unit activity coefficient*/
739:    /*      GammaZ1 = 1.0;*/
740:    /*      GammaZ2 = 1.0;*/
741:
742:            /*set activities for aqueous ions*/
743:            activity[h_p1]    = GammaZ1 *C[h_p1]    *0.001*CSCALE;
744:            activity[oh_m1]   = GammaZ1 *C[oh_m1]   *0.001*CSCALE;
745:            activity[h2co3]   =          C[h2co3]   *0.001*CSCALE;
746:            activity[hco3_m1] = GammaZ1 *C[hco3_m1] *0.001*CSCALE;
747:            activity[co3_m2]  = GammaZ2 *C[co3_m2]  *0.001*CSCALE;
748:            activity[hs_m1]   = GammaZ1 *C[hs_m1]   *0.001*CSCALE;
749:            activity[fe_p2]   = GammaZ2 *C[fe_p2]   *0.001*CSCALE;
750:            activity[mn_p2]   = GammaZ2 *C[mn_p2]   *0.001*CSCALE;
```

In preparation for providing the equilibrium mass action laws the activity coefficients, gamma, have to be calculated. The ionic strength of the aqueous solution is calculated using the major ions present in solution. Note that ions which are not computed for can also be included as constants such as Ca^{2+} which was defined earlier as C_ca_p2. The Davies equation is used to calculate gamma which depends on ionic charge. Then activity is calculated for each ion, the units of activity are converted to mol/l for ease of use in the following mass action laws.

```
751:
752:            /*Algebraic equations: mass action laws for equilibrium*/
```

```
753:
754:        /*                                log K       */
755:        /*  h2o   <-> h+ + oh-           -13.998     */
756:             res[oh_m1] =     ( (activity[oh_m1])
757:                                                  * (activity[h_p1]) )
758:                                                  / 1.00462E-14
759:                                                  - 1.0    ;
760:             res[oh_m1] = res[oh_m1] *1e-10;
761:
762:        /*                                 log K  */
763:        /* h2co3   <-> h+ + hco3-        -6.351 */
764:             res[h2co3] =   1./activity[h2co3]
765:                                                  * activity[h_p1]
766:                                                  * activity[hco3_m1]
767:                                                  / 4.4566E-07
768:                                                  - 1.0     ;
769:             res[h2co3] = res[h2co3] *1e-10;
770:
771:        /*                                 log K  */
772:        /* hco3-   <-> h+ + co3-2        -10.330 */
773:             res[co3_m2] =    (   activity[co3_m2]
774:                                                    *  activity[h_p1]
775:                                                    /   activity[hco3_m1]
776:                                                    / 4.6773E-11
777:                                                    - 1.0  )    ;
778:             res[co3_m2] = res[co3_m2] *1e-10;
779:
```

Aqueous complexation reactions are specified. The secondary species, i, has been placed first in calculating the ion activity product, IAP, although this is not essential. The mass action law format used is IAP/K-1. For the secondary species the mass storage and transport terms have been set to zero, so the overall equation to be linearised for each secondary species is {Source/Sink} = 0, That is IAP/K-1 = 0 which is the mass action law for local equilibrium.

The weighting factor applied to each result, e.g.[778], helps provide stability to the solution process.

```
780:
781:        /* mineral                          log K  */
782:        /* fes    <-> fe2+ + hs- - H+    -4.648 */
783:
784:             sp_index =  fes_s;
785:             IAP      =     activity[fe_p2]
786:                                    *  activity[hs_m1]
787:                                    / (activity[h_p1] );
788:             Ksp      =       2.2491e-5  ;
789:             min_IAP_species = MIN(C[fe_p2], C[hs_m1]);
790:
791:             res[sp_index] = Mineral_equil ( IAP,  Ksp,  C[sp_index],  min_IAP_species,1e-10);
792:
```

Here the parameters needed by the mineral equilibrium function, above, are given. The minimum function is used to test for negative concentrations in the aqueous species which will be reset by the mineral equilibrium function. Negative concentrations may occur due to the discretisation scheme for transport, or due to chemical reactions consuming too much of one species.

```
793:
794:             /*Surface Complexation equilibrium reactions*/
795:        /*   xfeoh2_p1                              log K  */
796:        /* xfeoh + h+   <-> xfeoh2_p1       6.45 */
797:             res[xfeoh2_p1] =    (   C[xfeoh2_p1]
798:                                                  /  activity[h_p1]
799:                                                  /  C[xfeoh]
800:                                                  / 1.950e7
801:                                                  - 1.0  )    ;
802:             res[xfeoh2_p1] *= 1e-10;
803:
804:        /*   xfeo_m1                             log K  */
805:        /* xfeoh    <-> xfeo_m1 +h+             -8.93 */
806:             res[xfeo_m1] =    (   C[xfeo_m1]
807:                                                   *  activity[h_p1]
808:                                                   /  C[xfeoh]
809:                                                   / 1.175e-9
810:                                                   - 1.0  )    ;
811:             res[xfeo_m1] *= 1e-10;
812:
813:
814:        /*   xfeofe_p1                             log K  */
815:        /* xfeoh  + fe+2  <-> xfeofe+   + h+            -2.98 */
816:             res[xfeofe_p1] =    (   C[xfeofe_p1]
817:                                                   *  activity[h_p1]
818:                                                   /  C[xfeoh]
819:                                                   / activity[fe_p2]
820:                                                   / 1.047e-3
821:                                                   - 1.0  )    ;
822:             res[xfeofe_p1] *= 1e-10;
823:
```

```
824:
825:        /*   xfeofeoh                          log K  */
826:        /* xfeoh  + fe+2 +h2o <-> xfeofeoh   + 2h+           -11.55 */
827:             res[xfeofeoh] =   (   C[xfeofeoh]
828:                                     *  activity[h_p1] * activity[h_p1]
829:                                           /  C[xfeoh]
830:                                           / activity[fe_p2]
831:                                           / 2.818e-12
832:                                           - 1.0  )   ;
833:             res[xfeofeoh]  *= 1e-10;
834:
835:
836:        /*   xfeohmn_p1                          log K  */
837:        /* xfeoh  + mn+2  <-> xfeomn+   + h+          -3.5 */
838:             res[xfeomn_p1] =   (   C[xfeomn_p1]
839:                                           *  activity[h_p1]
840:                                           /  C[xfeoh]
841:                                           / activity[mn_p2]
842:                                           / 3.162e-4
843:                                           - 1.0  )   ;
844:             res[xfeomn_p1] *= 1e-10;
845:
846:             res[xfeoh] = 0.0;
847:
848:          return 0;
849:     }
850:
```

The surface complexation reactions are specified in the same way as the aqueous complexation reactions, except that activity coefficients are one for the surface species.

```
851:     static INT SourceSinkPreProcess (NP_NLS_TRANSPORT_PARAM *tp, MULTIGRID *mg)
852:     {
853:              return 0;
854:     }
```

This function allows wells to be inserted in the domain. It isn't used in this example, but see course documentation for more details.

```
855:
856:     static INT Initial (NP_NLS_TRANSPORT_PARAM *tp, DOUBLE *x, INT sd, INT nc, DOUBLE *C)
857:     {
858:             /*Initial concentration in mol/m^3, i.e. mmol/l*/
859:     /* use background concs from mayer et al JCH53 2001 */
860:             C[c6h6o]       = 3.4E-10*1000;
861:             C[o2]          = 2.9E-4*1000;
862:             C[no3_m1]      = 1.7E-3*1000;
863:             C[h2]          = 1.E-10*1000;
864:             C[fe_p2]       = 8.9E-7*1000;
865:             C[mn_p2]       = 1.2E-7*1000;
866:             C[so4_m2]      = 6.7E-4*1000;
867:             C[hs_m1]       = 3.E-10*1000;
868:             C[ch4]         = 6.2E-10*1000;
869:             C[ch3cooh]     = 1.e-6*1000;
870:             C[h_p1]        = 1.26E-7*1000;      /* pH 6.9*/
871:             C[hco3_m1]     = 2.278E-3*1000;
872:             /* immobile species*/ /* C[mol/kg_solid]*/
873:             C[feooh_s] = 5.5/56.;              /*  5.5g_Fe/kg_s, Mr=56g/mol*/
874:             C[mno2_s]  = 0.048/55.;            /*  48mg_Mn/kg_s, Mr=55g/mol*/
875:             C[fes_s]      = 1.e-13;       /*  1e-13 mol_fes/kg_s*/
876:             /*aqueous equilibrium secondary species*/
877:             C[oh_m1]       = 1.00462E-14/C[h_p1]*1000.*1000.;
878:             C[h2co3]       = C[h_p1]/1000.*C[hco3_m1]/1000./4.4566E-07 *1000.;
879:             C[co3_m2]      = C[hco3_m1]/C[h_p1]*4.6773E-11*1000.;
880:
881:             /* surface complex species*/ /* C[mol/m^3_h20]*/
882:             C[xfeoh]       = 4e-3 *1000;
883:             C[xfeoh2_p1] = C[xfeoh]*C[h_p1]/1000.*1.950e7;
884:             C[xfeo_m1]     = C[xfeoh]/C[h_p1]*1000.*1.175e-9;
885:             C[xfeofe_p1] = C[xfeoh]/C[h_p1]*C[fe_p2]*1.047e-3;
886:             C[xfeofeoh]   = C[xfeoh]/C[h_p1]/C[h_p1]*C[fe_p2]*1000.*2.818e-12;
887:             C[xfeomn_p1] = C[xfeoh]/C[h_p1]*C[mn_p2]*3.162e-4;
888:
889:             /*optional conversion from mol/l to mol/m^3*/
890:             /*C[0]     = C[0]/CSCALE;*/ /*etc*/
891:             return 0;
892:     }
```

The initial values for concentrations are given here. The equilibrium species are specified using the mass action laws so that equilibrium is assured at the start of the simulation.

```
893:
894:     /* base class functions */
895:     static INT Init (NP_BASE *theNP, INT argc , char **argv)
896:     {
897:     #ifdef SOLVEFLOW
898:              NP_FLOW_TRANSPORT *ft = (NP_FLOW_TRANSPORT *) theNP;
899:              int n;
900:
901:              /* get interface class */
902:              ft->flow = (NP_WATER *) ReadArgvNumProc(theNP-
        >mg,"flow",WATER_CLASS_NAME,argc,argv);
903:              if (ft->flow == NULL) return(NP_NOT_ACTIVE);
```

```
904:    #endif
905:            return(NP_ACTIVE);
906:    }
907:
908:
909:    static INT Display (NP_BASE *theNP)
910:    {
911:            return(0);
912:    }
913:
914:    static INT Execute (NP_BASE *theNP, INT argc , char **argv)
915:    {
916:            UserWrite("Cannot execute !\n");
917:        return(0);
918:    }
919:
920:    /* constructor function */
921:    static INT Construct (NP_BASE *theNP)
922:    {
923:        NP_NLS_TRANSPORT_PARAM *np;
924:
925:            /* upcast */
926:        np = (NP_NLS_TRANSPORT_PARAM *) theNP;
927:
928:            /* base functions */
929:        theNP->Init           = Init;
930:        theNP->Display                = Display;
931:        theNP->Execute                = Execute;
932:
933:            /* problem functions */
934:        np->Adsorption          = Adsorption;
935:        np->Darcy               = Darcy;
936:        np->Dispersion          = Dispersion;
937:        np->Umatrix             = Umatrix;
938:        np->Source              = Source;
939:        np->SourceSinkPreProcess = SourceSinkPreProcess;
940:        np->Initial             = Initial;
941:
942:        return 0;
943:    }
944:
```

The Init, Display, Execute, Construct functions are standard for UG – see the Tutorial. In Init the new transport parameter object is told that it needs to know the script name for the flow solver object.

```
945:
946:
947:    /*****************************************************************************/
948:    /*****************************************************************************/
949:    /*       implement boundary conditions                                       */
950:    /*****************************************************************************/
951:    /*****************************************************************************/
952:
953:    static INT SouthBoundary (void *segdata, void *conddata, DOUBLE *in, DOUBLE
954:     *outValues, INT *bndType)
955:    {
956:    INT i;
957:    for (i=0;i<NSPECIES ;i++)
958:            {
959:            bndType[i] = TRANSPORT_FLUX;
960:            outValues[i] = 0.0;
961:            }
962:
963:            return(0);
964:    }
965:
966:
967:    static INT EastBoundary (void *segdata, void *conddata, DOUBLE *in, DOUBLE
968:     *outValues, INT *bndType)
969:    {
970:    INT i;
971:            for (i=0;i<NSPECIES ;i++)
972:                    bndType[i] = TRANSPORT_CONCENTRATION;
973:
974:            /* use background concs from mayer et al JCH53 2001 */
975:            outValues[c6h6o]      = 3.4E-10*1000;
976:            outValues[o2]         = 2.9E-4*1000;
977:            outValues[no3_m1]     = 1.7E-3*1000;
978:            outValues[h2]         = 1.E-10*1000;
979:            outValues[fe_p2]      = 8.9E-7*1000;
980:            outValues[mn_p2]      = 1.2E-7*1000;
981:            outValues[so4_m2]     = 6.7E-4*1000;
982:            outValues[hs_m1]      = 3.E-10*1000;
983:            outValues[ch4]        = 6.2E-10*1000;
984:            outValues[ch3cooh]    = 1.e-6*1000;
985:            outValues[h_p1]       = 1.26E-7*1000;      /*pH 6.9*/
986:            outValues[hco3_m1]    = 2.278E-3*1000;
987:            /* immobile species*/ /* C[mol/kg_solid]*/
988:            outValues[feooh_s] = 5.5/56.;              /* 5.5g_Fe/kg_s, Mr=56g/mol*/
```

```
989:            outValues[mno2_s]  = 0.048/55.;            /* 48mg_mn/kg_s, Mr=55g/mol*/
990:            outValues[fes_s]   = 1.0e-13;       /*  1e-13 mol_fes/kg_s*/
991:            /* surface complex species*/ /* C[mol/m^3_h20]*/
992:            outValues[xfeoh]     = 4.e-3 *1000;
993:            outValues[xfeoh2_p1] = outValues[xfeoh]*outValues[h_p1]/1000.*1.950e7;
994:            outValues[xfeo_m1]   = outValues[xfeoh]/outValues[h_p1]*1000.*1.175e-9;
995:            outValues[xfeofe_p1] = outValues[xfeoh]/outValues[h_p1]*outValues[fe_p2]*1.047e-3;
996:            outValues[xfeofeoh]  =
      outValues[xfeoh]/outValues[h_p1]/outValues[h_p1]*outValues[fe_p2]*1000.*2.818e-12;
997:            outValues[xfeomn_p1] = outValues[xfeoh]/outValues[h_p1]*outValues[mn_p2]*3.162e-4;
998:            /*secondary aqueous species*/
999:            outValues[oh_m1]      = 1.00462E-14/outValues[h_p1]*1000. *1000.;
1000:           outValues[h2co3]      = outValues[h_p1]/1000.*outValues[hco3_m1]/1000./4.4566E-07
      *1000.;
1001:           outValues[co3_m2]     = outValues[hco3_m1]/outValues[h_p1]*4.6773E-11 *1000.;
1002:
1003:           return(0);
1004:   }
1005:
1006:
1007:   static INT NorthBoundary (void *segdata, void *conddata, DOUBLE *in, DOUBLE
1008:    *outValues, INT *bndType)
1009:   {
1010:   INT i;
1011:   for (i=0;i<NSPECIES ;i++)
1012:           {
1013:           bndType[i] = TRANSPORT_FLUX;
1014:           outValues[i] = 0.0;
1015:           }
1016:
1017:           return(0);
1018:   }
1019:
1020:   /*initial west bdry representing concs observed at BH60, 350m from source */
1021:   static INT WestBoundary (void *segdata, void *conddata, DOUBLE *in, DOUBLE
1022:    *outValues, INT *bndType)
1023:   {
1024:           INT i;
1025:           DOUBLE lambda;
1026:           lambda = in[0];
1027:   #ifdef ONEDIM
1028:           lambda = 0.38;
1029:   #endif
1030:
1031:           for (i=0;i<NSPECIES ;i++)
1032:                   bndType[i] = TRANSPORT_CONCENTRATION;
1033:
1034:
1035:           if (lambda>=0.25 && lambda<0.5)
1036:                   outValues[c6h6o]      = 3.4E-7 + (48.0 - 3.4E-7)*(lambda-0.25)/(0.25);
1037:           else if (lambda>=0.5 && lambda<=0.75)
1038:                   outValues[c6h6o]      = 111.0  + (3.4E-7 - 111.0)*(lambda-0.5)/(0.25);
1039:           else
1040:                   outValues[c6h6o]      = 3.4E-10*1000 ;
1041:
1042:           if (lambda>=0.25 && lambda<0.4)
1043:                   outValues[so4_m2]     = 7.0;
1044:           else if (lambda>=0.4 && lambda<=0.75)
1045:                   outValues[so4_m2]     = 7.0   + (2.0 - 7.0)*(lambda-0.4)/(0.35);
1046:           else
1047:                   outValues[so4_m2]     = 6.7E-4*1000 ;
1048:
1049:           if (lambda>=0.25 && lambda<=0.75)
1050:                   outValues[hco3_m1]     = 1.0   + (1.6 - 1.0)*(lambda-0.25)/(0.5);
1051:           else
1052:                   outValues[hco3_m1]     = 2.28e-3*1000 ;
1053:
1054:           if (lambda>=0.25 && lambda<=0.75)
1055:                   outValues[h_p1]      = ( 8.89e-6   + (3.16e-6 - 8.89e-6)*(lambda-
      0.25)/(0.5) )*1000;
1056:           else
1057:                   outValues[h_p1]      = 1.26e-7*1000 ;
1058:
1059:           outValues[o2]         = 2.9E-4*1000;
1060:           outValues[no3_m1]     = 1.7E-3*1000;
1061:           outValues[h2]         = 1.E-8*1000;
1062:           outValues[fe_p2]      = 8.9E-7*1000;
1063:           outValues[mn_p2]      = 1.2E-7*1000;
1064:           outValues[hs_m1]      = 3.E-10*1000;
1065:           outValues[ch4]        = 6.2E-10*1000;
1066:           outValues[ch3cooh]    = 1.e-6*1000;
1067:           /* immobile species*/ /* C[mol/kg_solid]*/
1068:           outValues[feooh_s] = 5.5/56.;              /*  5.5g_Fe/kg_s, Mr=56g/mol*/
1069:           outValues[mno2_s]  = 0.048/55.;            /*  48mg_mn/kg_s, Mr=55g/mol*/
1070:           outValues[fes_s]   = 1.0e-13;          /*  1e-13 mol_fes/kg_s*/
1071:
1072:           outValues[oh_m1]      = 1.00462E-14/outValues[h_p1]*1000. *1000.;
1073:           outValues[h2co3]      = outValues[h_p1]/1000.*outValues[hco3_m1]/1000./4.4566E-07
      *1000.;
1074:           outValues[co3_m2]     = outValues[hco3_m1]/outValues[h_p1]*4.6773E-11 *1000.;
```

```
1075:
1076:           /* surface complex species*/ /* C[mol/m^3_h20]*/
1077:           outValues[xfeoh]      = 4.e-3 *1000;
1078:           outValues[xfeoh2_p1] = outValues[xfeoh]*outValues[h_p1]/1000.*1.950e7;
1079:           outValues[xfeo_m1]    = outValues[xfeoh]/outValues[h_p1]*1000.*1.175e-9;
1080:           outValues[xfeofe_p1] = outValues[xfeoh]/outValues[h_p1]*outValues[fe_p2]*1.047e-3;
1081:           outValues[xfeofeoh]  =
      outValues[xfeoh]/outValues[h_p1]/outValues[h_p1]*outValues[fe_p2]*1000.*2.818e-12;
1082:           outValues[xfeomn_p1] = outValues[xfeoh]/outValues[h_p1]*outValues[mn_p2]*3.162e-4;
1083:
1084:           return(0);
1085:   }
1086:
1087:
1088:   /*      later west bdry time>25years  representing concs observed at BH59, 130m from source
      */
1089:
1090:   static INT WestBoundary_b (void *segdata, void *conddata, DOUBLE *in, DOUBLE
1091:    *outValues, INT *bndType)
1092:   {
1093:           INT i;
1094:           DOUBLE lambda;
1095:           lambda = in[0];
1096:   #ifdef ONEDIM
1097:           lambda = 0.38;
1098:   #endif
1099:
1100:           for (i=0;i<NSPECIES ;i++)
1101:                   bndType[i] = TRANSPORT_CONCENTRATION;
1102:
1103:           if (lambda>=0.25 && lambda<0.575)
1104:                   outValues[c6h6o]      = 3.4E-7 + (32.0 - 3.4E-7)*(lambda-0.25)/(0.325);
1105:           else if (lambda>=0.575 && lambda<=0.75)
1106:                   outValues[c6h6o]      = 83.0   + (3.4E-7 - 83.0)*(lambda-0.575)/(0.175);
1107:           else
1108:                   outValues[c6h6o]      = 3.4E-10*1000 ;
1109:
1110:           if (lambda>=0.25 && lambda<0.35)
1111:                   outValues[so4_m2]      = 0.67       + (2.5 - 0.67 )*(lambda-0.25)/(0.1);
1112:           else if (lambda>=0.35 && lambda<=0.75)
1113:                   outValues[so4_m2]      =  1.5;
1114:           else
1115:                   outValues[so4_m2]      = 6.7E-4*1000 ;
1116:
1117:           if (lambda>=0.25 && lambda<=0.75)
1118:                   outValues[hco3_m1]      = 16.0   + (5.0 - 16.0)*(lambda-0.25)/(0.5);
1119:           else
1120:                   outValues[hco3_m1]      = 2.28e-3*1000 ;
1121:
1122:           if (lambda>=0.25 && lambda<=0.75)
1123:                   outValues[h_p1]      = ( 5.576e-7   + (5.576e-7 - 5.576e-7)*(lambda-
      0.25)/(0.5) )*1000;
1124:           else
1125:                   outValues[h_p1]      = 1.26e-7*1000 ;
1126:
1127:           outValues[o2]          = 2.9E-4*1000;
1128:           outValues[no3_m1]      = 1.7E-3*1000;
1129:           outValues[h2]          = 1.E-8*1000;
1130:           outValues[fe_p2]       = 8.9E-7*1000;
1131:           outValues[mn_p2]       = 1.2E-7*1000;
1132:           outValues[hs_m1]       = 3.E-10*1000;
1133:           outValues[ch4]         = 6.2E-10*1000;
1134:           outValues[ch3cooh]     = 1.e-6*1000;
1135:           /* immobile species*/ /* C[mol/kg_solid]*/
1136:           outValues[feooh_s] = 5.5/56.;              /*  5.5g_Fe/kg_s, Mr=56g/mol*/
1137:           outValues[mno2_s]  = 0.048/55.;            /*  48mg_mn/kg_s, Mr=55g/mol*/
1138:           outValues[fes_s]   = 1.0e-13;          /*  1e-13 mol_fes/kg_s*/
1139:
1140:           outValues[oh_m1]       = 1.00462E-14/outValues[h_p1]*1000. *1000.;
1141:           outValues[h2co3]       = outValues[h_p1]/1000.*outValues[hco3_m1]/1000./4.4566E-07
      *1000.;
1142:           outValues[co3_m2]      = outValues[hco3_m1]/outValues[h_p1]*4.6773E-11 *1000.;
1143:
1144:           /* surface complex species*/ /* C[mol/m^3_h20]*/
1145:           outValues[xfeoh]      = 4.e-3 *1000;
1146:           outValues[xfeoh2_p1] = outValues[xfeoh]*outValues[h_p1]/1000.*1.950e7;
1147:           outValues[xfeo_m1]    = outValues[xfeoh]/outValues[h_p1]*1000.*1.175e-9;
1148:           outValues[xfeofe_p1] = outValues[xfeoh]/outValues[h_p1]*outValues[fe_p2]*1.047e-3;
1149:           outValues[xfeofeoh]  =
      outValues[xfeoh]/outValues[h_p1]/outValues[h_p1]*outValues[fe_p2]*1000.*2.818e-12;
1150:           outValues[xfeomn_p1] = outValues[xfeoh]/outValues[h_p1]*outValues[mn_p2]*3.162e-4;
1151:
1152:           return(0);
1153:   }
1154:
```

The boundary conditions have now been given. The south and north boundaries are no flow boundaries. The east boundary (outflow) is of type constant concentration with values equal to the initial, background concentrations. The west boundary (inflow) has two versions, one

boundary condition for the first 25 years, and one for after that. Most of the species are the same as the background concentration. The first four species listed are introduced at the boundary as a line source. However, the actual line source concentration depends on the lambda parameter which varies linearly between 0 and 1 along the boundary segment, i.e. the west boundary. IN the case of 1D simulations the user can choose which flowline to look at by specifying a unique value for lamda [1097].

```
1155:   /*****************************************************************************/
1156:   /*****************************************************************************/
1157:   /*       set up boundary value problem and parameter class                   */
1158:   /*****************************************************************************/
1159:   /*****************************************************************************/
1160:
1161:   INT InitProblem_FA2DT_01 (void)
1162:   {
1163:           /* allocate new problem structure */
1164:           if
        (CreateProblem("fourash_domain_1","fa2dt_01",TRANSPORT_PROBLEMID,NULL,0,NULL,0,NULL)==NULL)
1165:    return(1);
1166:
1167:           /* allocate the boundary conditions */
1168:           if (CreateBoundaryCondition("south bnd cond",0,SouthBoundary,NULL)==NULL) return(1);
1169:           if (CreateBoundaryCondition("east bnd cond", 1,EastBoundary ,NULL)==NULL) return(1);
1170:           if (CreateBoundaryCondition("north bnd cond",2,NorthBoundary,NULL)==NULL) return(1);
1171:           if (CreateBoundaryCondition("west bnd cond", 3,WestBoundary ,NULL)==NULL) return(1);
1172:
1173:           /* make bvp, still old style */
1174:           if (CreateBVP_Problem("fa2dt_01","fourash_domain_1","fa2dt_01")==NULL) return (1);
1175:
1176:
1177:           /*for BCs after 25years*/
1178:
1179:           /* allocate new problem structure */
1180:           if
        (CreateProblem("fourash_domain_1","fa2dt_01_b",TRANSPORT_PROBLEMID,NULL,0,NULL,0,NULL)==NULL
        )
1181:    return(1);
1182:
1183:           /* allocate the boundary conditions */
1184:           if (CreateBoundaryCondition("south bnd cond",0,SouthBoundary,NULL)==NULL) return(1);
1185:           if (CreateBoundaryCondition("east bnd cond", 1,EastBoundary ,NULL)==NULL) return(1);
1186:           if (CreateBoundaryCondition("north bnd cond",2,NorthBoundary,NULL)==NULL) return(1);
1187:           if (CreateBoundaryCondition("west bnd cond", 3,WestBoundary_b ,NULL)==NULL)
      return(1);
1188:
1189:           /* make bvp, still old style */
1190:           if (CreateBVP_Problem("fa2dt_01_b","fourash_domain_1","fa2dt_01_b")==NULL) return
      (1);
1191:
1192:
1193:           /* create corresponding parameter class */
1194:       if (CreateClass(NLS_TRANSPORT_PARAM_CLASS_NAME ".FA2DT_01",sizeof(NP_FLOW_TRANSPORT),
1195:                           Construct)) return (__LINE__);
1196:
1197:           return(0);
1198:   }
```

In this problem initialisation function two boundary value problems, fa2dt_01 and fa2dt_01_b, are created from the functions above. They are the same apart from the differing west boundary functions. They are both created on the domain fourash_domain_1 which is created in the file UG/course/appl/fourash/fourash_domain2d.c. It is critical that that the flow problem uses the same domain as the transport problem.

Finally the FA2D_01 class is created for use in the script file.

This function InitProblem_FA2DT_01 must be called from the main() function in the file UG/course/appl/course.c. Course documentation has more details on this procedure. Also some step-by-step guidance and tips is available in the files:

UG/course/appl/fourash2d/notes/howto_makenewmodel.ian, and
UG/course/appl/howto_make_new_model_dir.ian.

10.7 Example of 3D Flow and Transport UG Script

The 3D model example is conceptually the same as the 2D example with the same flow field and kinetic reactions, but less species are simulated and no equilibrium reactions included.

The example script file is called:

`UG/course/appl/fourash3d/example_3d_transport.scr`

In order to run it as described above copy the files contents into:

`UG/course/appl/fourash3d/fourash_3d_transport_01.scr`

Other files of relevance specifically to the 3D example are:
Domain in 3D: UG/course/appl/fourash/fourash_domain3d.c
Mesh 3D creation script: UG/course/appl/fourash3d/fourash_3d_3.mesh
3D Flow parameters: UG/course/appl/fourash/fourash_flow3d_1.c
New 3D models help: UG/course/appl/fourash3d/create_3dmodel_comments.ian

Most items are the same as the 2D example and are not reproduced.
Parts of the 3D script which do differ from the 2D script are as follows:

```
100:    #Chemistry###################################
101:    sol= "sol";
102:
103:    MOBILE     = 11;
104:    TOTSPECIES = 11;
105:
106:    C0 = "c6h6o";              #a
107:    C1 = "o2";         #b
108:    C2 = "no3-m1";     #c
109:    C3 = "hco3-m1";            #d
110:    C4 = "h2";                 #e
111:    C5 = "mn-p2";      #f
112:    C6 = "ch4";                #g
113:    C7 = "ch3cooh";            #h
114:    C8 = "fe-p2";      #i
115:    C9 = "so4-m2";     #j
116:    C10 = "hs-m1";     #k
```

The chemical system is reduced in size to only 11 kinetic species, which are all primary components, i.e. no equilibrium reactions are simulated in this example.

```
169:    ### read stochastic file for source zone
170:    rsf_read gen/rsf_src_x6y1.out;
171:
```

The example uses a stochastic 2D field as the source plane for the phenol – this is explained more in the c code, below.

```
172:    ##########################################################
173:    # data format definition  SPARSE
174:    ##########################################################
175:    SPARSE=1;
176:
177:
178:    if (SPARSE==1)
179:    {
180:    ms :SparseFormats:Jsparse;
181:    :SparseFormats:Jsparse:Dnn =  "***00000000
182:                                   **000000000
183:                                   ***00000000
184:                                   *****0*0000
185:                                   ********0*0
186:                                   ***0**00000
187:                                   *****0*0000
188:                                   ***0000*000
189:                                   ***0*000*00
190:                                   ***0*0000*0
191:                                   ***0*0000**
192:                                           ";
193:
194:    #                              abcdefghijklmnopqrs
195:
196:    :SparseFormats:Jsparse:Tnn = " *0000000000
197:                                   0*000000000
198:                                   00*00000000
199:                                   000*0000000
200:                                   0000*000000
201:                                   00000*00000
202:                                   000000*0000
203:                                   0000000*000
204:                                   00000000*00
205:                                   000000000*0
206:                                   0000000000*
207:                                           ";
208:
```

The sparse matrix format is redefined for the smaller problem. The off-diagonal blocks are now diagonal.

```
663:    ##########################################################
664:    # create multigrid
665:    ##########################################################
666:    if (SOLVEFLOW) new mg $b FA3Dflow1 $f FlowFormat $h @HEAP;
667:    else           new mg $b fa3dt_01  $f FlowFormat $h @HEAP;
668:
```

The 3D flow and transport boundary value problems are used, as against the 2D ones. Similarly, all further occurences of these in the script are changed to the appropriate 3D names (these changes are not reproduced here).

```
669:    #ex fourash_2a.mesh;              #12 box coarse mesh
670:    ex fourash_3d_3.mesh;             #4 box coarse mesh
```

The 3D mesh creation file makes 4 (4*1*1) rectangular boxes. Note the alternative option to make 12 (4*3*1).

```
680:    ##########################################################
681:    # groundwater flow
682:    ##########################################################
683:    #flow velocity may be given or calculated
684:    if (SOLVEFLOW)
685:    {
686:    npcreate flow $c FA3DFLOW;
687:    npinit flow;

722:    ##########################################################
723:    # transport problem
724:    ##########################################################
725:
726:    npcreate trans $c FA3DT_01;
727:    if (SOLVEFLOW)  npinit   trans $flow bf;
728:    else            npinit   trans;
729:
```

The parameter classes have different names from the 2D version.

10.8 Example of 3D Transport Parameter c code

The following code is the parameter code used in the above script for 3D transport. It provides functions which are required by the boxnlsys interface.
The example parameter source code file is called:
`UG/course/appl/fourash3d/example_3d_transport.c`
In order to run it as described above copy the files contents into:
`UG/course/appl/fourash3d/fourash_3d_transport_01.c`

The following is a selection of the 3D example code which is unique from the 2d equivalent.

```
 62:    #undef SOLVEFLOW
 63:    #define HETERO_SOURCE
```

Flags for solving the flow and using a stochastic, heterogeneous source term.

```
 69:    #define c6h6o     0
 70:    #define o2        1
 71:    #define no3_m1    2
 72:    #define hco3_m1   3
 73:    #define h2        4
 74:    #define mn_p2     5
 75:    #define ch4       6
 76:    #define ch3cooh   7
 77:    #define fe_p2     8
 78:    #define so4_m2    9
 79:    #define hs_m1     10
 92:
 93:    #define NSPECIES 11
 94:    #define NMOBSPECIES 11 /* also define no of mobile species in script*/
 95:                                                    /* the script variable sets the absolute
        max
 96:                                                    no. mobile species*/
 97:    #define NNAPL     0
 98:    #define NMINERALS 0
 99:    #define NBACTERIA 0
```

The species list is modified from the 2D version.

```
197:    static INT Umatrix (NP_NLS_TRANSPORT_PARAM *tp, DOUBLE *U)
198:    {
213:
214:            int i,j;
215:            /*set up assuming all are primary components*/
216:            for (i=0; i<NSPECIES; i++)
217:                    for (j=0; j<NSPECIES; j++)
218:                            if (i==j) U[i*NSPECIES+j] = 1.0;
219:                            else      U[i*NSPECIES+j] = 0.0;
220:
221:            /* treat secondary species columns*/
262:            return 0;
263:    }
264:
```

The U matrix is simply diagonal, and does not in fact need to be used, since all the species are kinetic components.

```
267:    static INT Adsorption (NP_NLS_TRANSPORT_PARAM *tp,
268:                                                const ELEMENT *e, DOUBLE_VECTOR *corners,
269:                                                DOUBLE *local, DOUBLE *global,
270:                                                DOUBLE t, INT sd, INT nc, DOUBLE *C, DOUBLE
        *res)
271:    {
291:            INT i,j, row_i;
292:            DOUBLE U[NSPECIES*NSPECIES];
293:
294:            (tp->Umatrix)(tp,U);
295:
296:            /* multiply by U to get primary components TOTALS */
297:
298:            for (i=0;i<NSPECIES ;i++) /*loop over each component row in turn*/
299:            {
300:              row_i = i*NSPECIES;
301:              res[i] = 0.0;
302:          /*storage for mobile aqueous species*/
303:              for (j=0;j<NMOBSPECIES ;j++)
304:                    if ( U[row_i+j] != 0.0)
305:                            res[i] += PORO_WATER * CSCALE * C[j]*U[row_i+j];
327:            }
330:            return 0;
331:    }
```

Adsorption (mass storage) only needs to deal with the aqueous species since there are no other types.

```
334:    static INT Darcy (NP_NLS_TRANSPORT_PARAM *tp, const ELEMENT *e,
335:                          DOUBLE_VECTOR *corners, DOUBLE *local, DOUBLE *global,
336:                          INT subdomain, INT m, DOUBLE *jw)
337:    {
338:    #ifdef SOLVEFLOW
339:            NP_FLOW_TRANSPORT *ft = (NP_FLOW_TRANSPORT *) tp;
340:    #endif
341:            /* set to zero for immobile phases */
342:            if (m>NMOBSPECIES-1)
343:            {
344:                    jw[0]=jw[1]=jw[2]=0.;   /* for DIM 3 */
345:                    return 0;
346:            }
347:
348:    #ifdef SOLVEFLOW
349:            /*fetch flow solver results*/
350:            (ft->flow->Darcy)(ft->flow,e,corners,local,jw);
351:    #else
352:            /*water flux jw is not calculated, but given as permeability *head gradient
353:            8.4e-6*4.15e-3  = 3.528e-8 m/s   ie q=1.113m/yr  (v is higher, ~8m/yr)*/
354:            /*give darcy flux*/
355:            jw[0]= 3.4776e-8 ;
356:            jw[1]= 0.0 ;
357:            jw[2]= 0.0 ;
358:    #endif
363:            return 0;
364:    }
```

Darcy is same as 2D except that the third component is given, i.e. jw[2].

```
365:
366:    static INT Dispersion (NP_NLS_TRANSPORT_PARAM *tp, const ELEMENT *e,
367:                           DOUBLE_VECTOR *corners, DOUBLE *local, DOUBLE *global,
368:                           INT sd, INT m, DOUBLE *out)
369:    {
370:    #ifdef SOLVEFLOW
371:            NP_FLOW_TRANSPORT *ft = (NP_FLOW_TRANSPORT *) tp;
372:    #endif
373:            DOUBLE dei,aL,aTH,aTV;
374:            DOUBLE u[DIM],uabs;
375:            INT i;
376:
377:            /* set to zero for immobile phases */
378:            if (m>NMOBSPECIES-1)      /* i.e. index no 3 is immobile */
379:            {
380:                    out[0] = 0.0;   out[1] = 0.0;   out[2] = 0.0;
381:                    out[3] = 0.0;   out[4] = 0.0;   out[5] = 0.0;
382:                    out[6] = 0.0;   out[7] = 0.0;   out[8] = 0.0;
383:                    return 0;
384:            }
385:
386:            /* get data */
387:            dei = 0.0;        /* molecular diffusion in m^2/s  */
388:            aL =  1.0;        /* dispersivity longitudinal 1m, in m  */
389:            aTH = 0.01;       /* dispersivity transverse horizontal 1cm, in m  */
390:            aTV = 0.004;      /* dispersivity transverse vertical 4mm, in m  */
391:
392:    #ifdef SOLVEFLOW
393:            /* get velocity */
394:            (ft->flow->Darcy)(ft->flow,e,corners,local,u);
395:    #else
```

```
396:            /*give darcy flux*/
397:            u[0]= 3.4776e-8 ;
398:            u[1]= 0.0 ;
399:            u[2]= 0.0 ;
400:    #endif
401:
402:            uabs = sqrt(u[0]*u[0]+u[1]*u[1]+u[2]*u[2]);
403:
404:            /* fill dispersion tensor see ref. Zheng&Bennet p 46*/
405:            uabs=uabs+MYSMALL;
406:            out[0] = aL*u[0]*u[0]/uabs +aTH*u[1]*u[1]/uabs +aTV*u[2]*u[2]/uabs +dei;
407:            out[4] = aL*u[1]*u[1]/uabs +aTH*u[0]*u[0]/uabs +aTV*u[2]*u[2]/uabs +dei;
408:            out[8] = aL*u[2]*u[2]/uabs +aTV*u[0]*u[0]/uabs +aTV*u[1]*u[1]/uabs +dei;
409:
410:            out[1]=out[3]= (aL-aTH)*u[0]*u[1]/uabs;
411:            out[2]=out[6]= (aL-aTV)*u[0]*u[2]/uabs;
412:            out[5]=out[7]= (aL-aTV)*u[1]*u[2]/uabs;
418:            return 0;
419:    }
```

Dispersion tensor constructed in a three dimensional form. NB this form using 3 dispersivity values assumes roughly horizontal groundwater flow.

```
420:
421:    static INT Source (NP_NLS_TRANSPORT_PARAM *tp, const ELEMENT *e,
422:                               DOUBLE_VECTOR *corners, DOUBLE *local, DOUBLE *global,
423:                                                          DOUBLE t, INT sd, INT nc, DOUBLE *C,
        DOUBLE *res)
424:    {
447:            /*KINETIC REACTIONS must be written in terms of components*/
448:
449:            /*      test if x>v(t-Tchange) for if at front or back of plume*/
450:            if (global[0] > (3.4776e-8/0.125*(t - 25.0*3.1536E7)) ) plumefront = 1;
451:            else                                                    plumefront = 0;
452:
453:
454:            j=0;  /*reaction index (from 0 to NRATES-1 ) */
455:            /* reaction :  aerobic phenol oxidation */
456:            /* c6h6o + 7.o2 + 3.h2o -> 6.hco3- + 6.h+    */
457:            s[c6h6o][j]= -1.;     s[o2][j] =  -7.;
458:            s[hco3_m1][j]= 6.;  /* s[h_p1][j]= 6.;*/
459:            /*      Monod kinetics wrt c6h6o  & o2  */
460:            rate[j]= 1.0      *PORO_WATER          /*[m^3_h2o/m^3_bulk]*/
461:                                                  *3.9811E-10 *1000.    /* k_max = 10^-9.4
        mol/l_h20/s [mol/m^3_h2o/s] */
462:                                                  *Monod_a(C[c6h6o] *0.001,1.064E-4 )
463:                                                  *Monod_a(C[o2]    *0.001,3.125E-6 )
464:                                                  *reg_a (C[c6h6o],C[o2] )
465:                                                  *Threshold(C[c6h6o] *0.001 , 1.0E-10 , 2. )
466:                                                  *Threshold(C[o2]    *0.001 , 3.125E-7 , 2. )
467:                                                  ; /*  [mol/m3_bulk/s] */
468:

732:        return 0;
733:    }
```

The Source-sink function contains the same kinetic reactions as the 2D example, but now contains no equilibrium reactions.

The source-sink preprocess function is the same as in 2D.

```
740:    static INT Initial (NP_NLS_TRANSPORT_PARAM *tp, DOUBLE *x, INT sd, INT nc, DOUBLE *C)
741:    {
744:            C[c6h6o]      = 3.4E-10*1000;
745:            C[o2]         = 2.9E-4*1000;
746:            C[no3_m1]     = 1.7E-3*1000;
747:            C[hco3_m1]    = 2.9E-3*1000;
748:            C[h2]         = 1.E-10*1000;
749:            C[mn_p2]      = 1.2E-7*1000;
750:            C[ch4]        = 6.2E-10*1000;
751:            C[ch3cooh]    = 1.e-6*1000;
752:            C[fe_p2]      = 8.9E-7*1000;
753:            C[so4_m2]     = 6.7E-4*1000;
754:            C[hs_m1]      = 3.E-10*1000;
761:            return 0;
762:    }
```

The initial concentrations are the same as for the equivalent species in the 2D model. The exception is C[hco3_m1] which here represents TOTHCO$_3^-$ since the species H_2CO_3 and CO_3^{2-} are not included.

The Init, Display, Execute and Construct functions are identical to the 2D example.

```
849:    static INT WestBoundary (void *segdata, void *conddata, DOUBLE *in, DOUBLE
850:     *outValues, INT *bndType)
851:    {
852:            DOUBLE lambda1, lambda2;
853:            DOUBLE var;
854:            INT i, source1;
855:            lambda1 = in[0];       /* lambda1 increases with y */
856:            /*lambda2 = in[1];   */    /* lambda2 increases with z */
857:            lambda2 = 1. - in[1];  /* lambda2 decreases with z */
```

```
858:
859:
860:            for (i=0;i<NSPECIES ;i++)
861:                    bndType[i] = TRANSPORT_CONCENTRATION;
862:
863:
864:            if (lambda1<(65.0/90.0) ) source1 = 1;
865:            else source1 =0;
866:
867:
868:
869:            if ( source1 && lambda2>=0.25 && lambda2<0.5 )
870:                    outValues[c6h6o]       = 3.4E-7 + (48.0 - 3.4E-7)*(lambda2-0.25)/(0.25);
871:            else if ( ( source1 ) && (lambda2>=0.5 && lambda2<=0.75 ) )
872:                    outValues[c6h6o]       = 111.0  + (3.4E-7 - 111.0)*(lambda2-0.5)/(0.25);
873:            else
874:                    outValues[c6h6o]       = 3.4E-10*1000 ;
875:
876:    #ifdef HETERO_SOURCE
877:            if ( source1 && lambda1>(6.5/90.0) && lambda2>=0.25 && lambda2<=0.75 ) {
878:                    var = GetRSFValue(lambda1, (1.-lambda2) );
879:                    outValues[c6h6o]  = outValues[c6h6o] * exp(var);
880:            }
881:    #endif
882:
883:
884:            if ( (source1) && lambda2>=0.25 && lambda2<=0.75)
885:                    outValues[hco3_m1]       = 21.0   + (15.0 - 21.0)*(lambda2-0.25)/(0.5);
886:            else
887:                    outValues[hco3_m1]       = 2.9e-3*1000 ;
888:
889:
890:            if ( (source1) && lambda2>=0.25 && lambda2<0.4)
891:                    outValues[so4_m2]       = 7.0;
892:            else if ( (source1) && lambda2>=0.4 && lambda2<=0.75)
893:                    outValues[so4_m2]       = 7.0   + (2.0 - 7.0)*(lambda2-0.4)/(0.35);
894:            else
895:                    outValues[so4_m2]       = 6.7E-4*1000 ;
896:
897:
898:
899:            outValues[o2]          = 2.9E-4*1000;
900:            outValues[no3_m1]      = 1.7E-3*1000;
901:            outValues[h2]          = 1.E-10*1000;
902:            outValues[mn_p2]       = 1.2E-7*1000;
903:            outValues[ch4]         = 6.2E-10*1000;
904:            outValues[ch3cooh]     = 1.e-6*1000;
905:            outValues[fe_p2]       = 8.9E-7*1000;
906:            outValues[hs_m1]       = 3.E-10*1000;
907:
908:            return(0);
909:    }
```

There are now 6 boundary conditions to specify. The East boundary, like in 2D, provides the same values as the Initial function. The North and South boundaries are no flow as with 2D. Top and Bottom boundaries are introduced which are also no flow (recharge is not considered in this example, which preserves grid parallel flow lines in the homogeneous flow scenario considered). As in 2D the West boundary is given twice, once for times up to 25 years and once for later times. Since the boundaries are now planes and not lines there are two lambda parameters, varying between 0 and 1, with which to specify the boundary value in the plane.
Lambda1 is defined in the y direction and is used to set the width of the source to 65m in the 90m wide domain [864]. Note that the example is set up to model half the plume in the horizontal transverse direction. In the vertical direction the model reproduces the 2D example source[869-874]. Thus, lambda2 here is the same as lambda in the 2D example. The direction in which the lambda parameters increase depends on the set up in the domains file. The HETERO_SOURCE flag was defined above so that a spatially varying source zone can be used. The area up to 6.5m away from the plume centreline is left unchanged from the 2D model, but the rest of the source zone is selected [877]. Assuming that the random spatial field file has been read in using the script command rsf_read, the code uses the boundary plane parameters, lamba, to extract a value from the stochastic field [878]. The course documentation explains the creation of these fields. The random spatial field values are logarithmic with a mean value of zero. This random field is then multiplied by the source zone concentrations for phenol in the selected area [879]. The overall effect is a phenol source that is laterally extended from the 2D vertical plane, but with some heterogeneity superimposed away from the centreline.

```
1032:   INT InitProblem_FA3DT_01 (void)
1033:   {
1034:           /* allocate new problem structure */
```

```
1035:           if
        (CreateProblem("fourash_domain3d_1","fa3dt_01",TRANSPORT_PROBLEMID,NULL,0,NULL,0,NULL)==NULL
        ) return(1);
1036:
1037:           /* allocate the boundary conditions */
1038:           if (CreateBoundaryCondition("bottom bnd cond",0,BottomBoundary,NULL)==NULL)
        return(1);
1039:           if (CreateBoundaryCondition("west bnd cond",  1,WestBoundary ,NULL)==NULL)
        return(1);
1040:           if (CreateBoundaryCondition("south bnd cond", 2,SouthBoundary,NULL)==NULL)
        return(1);
1041:           if (CreateBoundaryCondition("top bnd cond",   3,TopBoundary ,NULL)==NULL) return(1);
1042:           if (CreateBoundaryCondition("east bnd cond",  4,EastBoundary,NULL)==NULL) return(1);
1043:           if (CreateBoundaryCondition("north bnd cond", 5,NorthBoundary ,NULL)==NULL)
        return(1);
1044:
1045:           /* make bvp, still old style */
1046:           if (CreateBVP_Problem("fa3dt_01","fourash_domain3d_1","fa3dt_01")==NULL) return(1);

1066:           /* create corresponding parameter class */
1067:       if (CreateClass(NLS_TRANSPORT_PARAM_CLASS_NAME ".FA3DT_01",sizeof(NP_FLOW_TRANSPORT),
1068:                           Construct)) return (__LINE__);
1069:
1070:           return(0);
1071:   }
```

The problem initialisation function is very similar to the 2D example. Two problems and one class are created. The problem needs to know all six boundary condition functions.

11 Appendix E. Contents of CDROM

The CDROM in the back cover of this thesis contains electronic copies of this manuscript together with data, codes, and simulation results presented in the thesis as follows:

Manuscripts:

- this PhD thesis
- Published version of chapter 2
- Various conference papers
- database of references (ENDNOTE library file)

Laboratory modelling:

- Measured data
- Simulated results
- MIN3P files

Field Modelling:

- Measured data
- Simulated results (2D and 3D)
- UG source code (including test cases and input files for field modelling)