European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2006
P. Wesseling, E. Oñate and J. Périaux (Eds)

IMPROVING MULTIGRID FOR 3-D ELECTRO-MAGNETIC DIFFUSION ON STRETCHED GRIDS

T.B. Jönsthövel*, C.W. Oosterlee* and W.A. Mulder†

* Delft University of Technology, Faculty EWI,
Mekelweg 4, 2628 CD Delft, The Netherlands

† Shell International Exploration and Production
P.O. Box 60, 2280 AB Rijswijk, The Netherlands
and
Delft University of Technology, Faculty CITG,
P.O. Box 5028, 2600 GA Delft, The Netherlands

Key words: electro-magnetics, diffusion, multigrid, controlled-source EM

Abstract. *We evaluated multigrid techniques for 3D diffusive electromagnetism. The Maxwell equations and Ohm's law were discretised on stretched grids, with stretching in all coordinate directions. We compared standard multigrid to alternative multigrid approaches with line-wise smoothing and semi-coarsening, both as a stand-alone solver and as a preconditioner. Although the number of iterations was small for some variants, none of the schemes showed grid-independent convergence rates on stretched grids.*

1 INTRODUCTION

Geophysical prospecting with electro-magnetic techniques is not only used for mineral and ore detection, but also for oil and gas exploration. With measurements performed at the earth's surface, only the low frequencies penetrate deep enough to detect the presence of hydrocarbons. At periods of the order of seconds, the light-speed waves have very small amplitudes and wavelengths larger than the size of the earth. Diffusion dominates the propagation of induction currents in the earth. As a consequence, electro-magnetic methods provide poor resolution. Still, they are useful because they can distinguish between conductive and resistive layers, for instance between non-conductive oil and conductive salty pore water. High-resolution seismic data cannot provide this information. The combination of the two types of measurements allows for better prospect evaluation.

During the last decade, marine applications of controlled-source electro-magnetic methods have become popular. Currents are injected into the sea water by a wire that is towed by a ship. Detectors are placed at the sea bottom and record the horizontal components of the electric and magnetic fields. Because sea-water is fairly conductive and the conductivity contrast with the sediments is not large, it is generally easier to detect resistive or conductive anomalies at

sea than on land. For shallow waters, the so-called air-wave may pose problems, just as with measurements on land. This air-wave is the almost static response of the highly resistive air.

Accurate and fast modelling and inversion software is required for the interpretation of electro-magnetic measurements. The problem is governed by the Maxwell equations together with Ohm's law. A frequency-domain formulation is attractive because marine data are usually provided at one or a few frequencies. We discretise the equations with the Finite Integration Technique [13, 2], which can be viewed as a finite-volume generalisation of Yee's scheme [15] for tensor-product Cartesian grids with variable grid spacings. Grid stretching is applied so that perfectly conducting boundary conditions can be used. The discretisation leads to a large, sparse linear system of equations that can be solved by, for instance, the multigrid method. Initial tests in [7, 8] showed that grid-independent convergence rates can be obtained on equidistant grids. On stretched grids, however, the convergence rates deteriorate dramatically [7, 8]. This can be partly compensated by using multigrid as a preconditioner for `bicgstab2` [11, 3]. Still, the resulting method requires careful gridding to avoid convergence problems and is not robust. The grid stretching introduces 3D anisotropies of varying nature into the discrete equations, which is a well-known problem for multigrid solvers. Line-wise, plane-wise relaxation schemes are among the several potential solutions, as is the concept of semi-coarsening in appropriate directions [10, 9]. Here, the basic relaxation method chosen is the line-wise relaxation scheme. We prefer to avoid plane relaxation because of its high computational cost. Instead, we combine the line relaxation with suitable semi-coarsening techniques [12].

The outline of this paper is as follows. Section 2 summarizes the equations, the discretisation, and the different multigrid variants evaluated. Convergence results for three test problems are presented in section 3. Section 4 lists the conclusions.

2 METHOD

2.1 Governing equations and discretisation

The propagation of electromagnetic waves in the earth can be modelled by Maxwell's equations together with Ohm's law. In the frequency domain, we have

$$i\omega\mu_0\tilde{\sigma}\mathbf{E} - \nabla \times \mu_r^{-1}\nabla \times \mathbf{E} = -i\omega\mu_0\mathbf{J}_s,$$

where the vector $\mathbf{E}(\omega, \mathbf{x})$ represents the electric field components as a function of angular frequency ω and position $\mathbf{x}$. The current source is $\mathbf{J}_s(\omega, \mathbf{x})$. Other quantities are $\tilde{\sigma}(\mathbf{x}) = \sigma - i\omega\epsilon_0\epsilon_r$, with $\sigma(\mathbf{x})$ the conductivity, $\epsilon_r(\mathbf{x})$ the relative permittivity, $\mu_r(\mathbf{x})$ the relative permeability, and ϵ_0 and μ_0 their vacuum values. We will use SI units.

The equations are discretised on a tensor-product Cartesian grid allowing for grid stretching. The grid points define block-shaped cells. The electric field components are defined as edge averages. The Finite Integration Technique [13, 2] provides a discretisation with second-order accuracy, except in the case of a discontinuous μ_r which may lead to a first-order error. Perfectly electric conducting boundary conditions are used.

2.2 Multigrid

Multigrid methods are motivated by the fact that many iterative methods, especially if they are applied to elliptic problems, have a smoothing effect on the error between an exact solution and a numerical approximation. A smooth discrete error can be well represented on a coarser grid, where its approximation is much cheaper. The design of efficient relaxation methods for the multigrid solution of *systems* of partial differential equations often requires special attention. The smoother should smooth the error for all unknowns in the equations (that are possibly of different type) of a system.

A good indication for the appropriate choice of relaxation method for a system of equations can be derived from the systems' determinant. If the main operators (or their principal parts) of the determinant lie on the main diagonal, smoothing is a straightforward matter. In that case, the differential operator that corresponds to the primary unknown in each equation is the leading operator. Therefore, a simple equation-wise decoupled relaxation method can efficiently be used. If, however, the main operators in a system are not in the desired position, the choice of an efficient smoother needs care. A first obvious choice in the case of strong off-diagonal operators in the differential system is *coupled* or *collective* smoothing: All unknowns in the system at a certain grid point are updated simultaneously.

Decoupled smoothing, however, is to be preferred for reasons of efficiency and simplicity. An elegant way to describe distributive relaxation is to introduce a *right preconditioner* in the smoothing procedure [14]. For the discrete problem $\mathbf{L_h u_h} = \mathbf{f_h}$, we introduce new variables $\mathbf{w}_h$, where $\mathbf{u}_h = \mathbf{C_h w_h}$, and consider the transformed system $\mathbf{L_h C_h w_h} = \mathbf{f_h}$, with $\mathbf{C_h}$ chosen in such a way that the resulting operator $\mathbf{L_h C_h}$ is suited for *decoupled* (non-collective) relaxation. Coupled and decoupled smoothing approaches have their advantages and disadvantages. If a system of equations consists of elliptic and of other, non-elliptic, components, decoupled relaxation allows to choose different relaxation methods for the different operators appearing. However, for general systems of equations it may not be easy to find a suitable distributive relaxation scheme. Furthermore, the proper treatment of boundary conditions in distributive relaxation may not be trivial, as typically the systems' operator is transformed by the smoother but the boundary operator is sometimes not considered. In this respect the use of coupled smoother is straightforward and often robust. A significant difference in cost between coupled and distributive relaxation, however, lies in the line-wise treatment of the unknowns, which may be necessary in the case of stretched grids. The cost of a coupled line-wise relaxation step is substantially higher than of a decoupled line-wise relaxation. The latter can be set up as a tri- (or more) diagonal matrix, whereas in the coupled version all different unknowns at the line need to be updated simultaneously. For Maxwell's equations the two variants mentioned above are basically proposed by Arnold, Falk and Winther [1] (coupled relaxation) and Hiptmair [4] (decoupled smoothing). The smoothers are constructed such that the null-spaces from the curl-curl operator are handled adequately within smoothing.

The multigrid solver described in [7, 8] and also used here employs standard 3D coarsening, combining $2 \times 2 \times 2$ fine-grid cells into a single coarse-grid cell. Transfer operators are basic

choices: The restriction operator is based on volume averaging, the prolongation operator on piecewise constant interpolation in the direction of each electric-field component and on linear and bi-linear interpolation in the plane perpendicular to that component. The smoother is of coupled relaxation type. It is the Symmetric Block Gauss-Seidel [1] applied to the six electric-field components that live on the edges connected to a single node. The nodes were processed in lexicographical order and in the reverse order. This scheme does not require an additional divergence correction. This multigrid method shows textbook convergence rates on grids with constant spacing [7, 8]. Convergence slows down with grid stretching, because the stretching introduces anisotropies of varying type in the discrete equations. The multigrid convergence remedies include line or plane relaxation and semi-coarsening.

Here, we evaluate a combination of two multigrid components for handling anisotropies: semi-coarsening combined with line-wise relaxation. The line smoother solves the simultaneous equations for all electric-field components that live on the edges connected to a single line of nodes. It is thus a (costly) coupled line-wise smoother. The directions of the lines are x-, y-, or z. The processing of the lines is done in a lexicographical Gauss-Seidel ordering or in a zebra fashion. We compared for three examples the Symmetric Line Gauss-Seidel and zebra relaxation. The line relaxation should partly reduce the adverse effects of anisotropy in the direction of the line. Semi-coarsening should then take place in the direction of the smooth error. If the anisotropy occurs in arbitrary directions, due to 3D grid stretching, simultaneous semi-coarsening alternating in all directions should be used without line relaxation [6]. Although that scheme still has $O(N^3)$ V-cycle complexity, N being the number of cells in one coordinate direction, it has rather large memory and cpu-time requirements in 3D.

Here, we look for reasonable compromises between efficiency and robustness of the smoother. Therefore, we study a more simple semi-coarsening approach. We coarsen in two of the three coordinates simultaneously. Line-wise relaxation is applied along the same two coordinate directions. The direction in which coarsening is not applied, alternates between the multigrid cycles. On the coarsest grid, the line relaxation acts as a direct solver.

We compared the performance of six methods:

1. The original scheme used in [7, 8] employs post-smoothing with Symmetric Block Gauss-Seidel, F-cycles, and `bicgstab2`. The iteration count refers to the number of multigrid cycles performed inside the `bicgstab2` algorithm. One full `bicgstab2` step requires two preconditioning steps, so two multigrid cycles.
2. Pre- and post-smoothing with Symmetric Line Gauss-Seidel in three directions, F-cycles, and `bicgstab2`.
3. Pre- and post-smoothing with Symmetric Line Gauss-Seidel in two directions, semi-coarsening in these two directions, and F-cycles. The direction that is not coarsened, changes after each full cycle and remains the same within the cycle.
4. As 2, but with Zebra smoothing.
5. As 3, but with Zebra smoothing.
6. As 3, using three multigrid cycles as a preconditioner for `bicgstab2`. A full iteration takes 2×3 cycles. Again, we use the number of multigrid cycles as iteration count. Because

`bicgstab2` may stop half way a full iteration, we will obatin multiples of three. The direction in which no coarsening is carried out, has changed through all three coordinates after three cycles.

Note that a single smoothing step with Zebra relaxation has half the cost of a Symmetric Line Gauss-Seidel sweep.

3 RESULTS

Three test problems were taken from [7, 8]. Two are artificial, unphysical test problems. The third is a more realistic, marine example. Power-law grid stretching was used for the first two problems. Away from the origin, the cell width increased by $1 + \alpha$. The marine example had hyperbolic cosine grid stretching away from the source, with a minimum spacing $h_{\min}$ and a ratio between neighbouring cell widths in a given coordinate direction of at most $1 + \alpha$.

The iterations were stopped once the norm of the residual had decreased to 10^{-8} times its value for a zero electric field. The computations were carried out on a 64-bit AMD Opteron 238 machine running Linux. The algorithm was coded in `matlab`, with the smoother and residual evaluation written in `C`. The listed computer times are therefore far from optimal but nevertheless provide some indication of the performance.

3.1 Variable conductivity

The first artificial and unphysical test problem is based on eigenfunctions on the domain $[0, 2\pi]^3\,\mathrm{m}^3$. The conductivity is $\sigma = 10 + c(x, y, z)$, with $c(x, y, z) = (x+1)(y+2)(z-\pi)^2$ for $z < \pi$ and zero elsewhere. We set $\epsilon_r = 0$ and $\mu_r = 1$. The electric field components are $E_1 = -2\cos(x)\sin(y)\sin(z)$, $E_2 = -2\sin(x)\cos(y)\sin(z)$, and $E_3 = \sin(x)\sin(y)\cos(z)\,\mathrm{V/m}$. The angular frequency is $\omega = 10^6$ Hz. The current source term follows from these expressions.

Tables 1 and 2 list convergence results for grids with $N \times N \times N$ cells and different amounts of grid stretching. Grid-independent convergence rates on the equidistant grids are only observed for the original method 1 and the new methods 2 and 4, and perhaps 6. Methods 3 and 6, with semi-coarsening one direction and Symmetric Gauss-Seidel line-relaxation in the other two directions, appear to be insensitive to the amount of stretching when considering a fixed number of cells. In terms of cpu-time, methods 3 and 6 are the only one that comes close to the original method 1 wild mild stretching. With stronger stretching on fine grids, they win.

3.2 Variable permittivity

The second problem was taken from [8]. It resembles the previous example, but now the conductivity is constant whereas $\mu_r = 1$ for $z < \pi$ and $\mu_r = 2$ for $z > \pi$. The electric field component E_3 is discontinuous across $z = \pi$.

Tables 3 and 4 list the convergence results for the six schemes. As in the previous example, methods 1, 2, 4 and possibly 6 show grid-independent convergence rates on equidistant grids, but this is lost on stretched grids. Methods 2, 3, and 6 perform reasonably well in terms of iteration count, but are much slower than the original method 1 in terms of cpu-time on mildly

		1:bi,SGS		2:bi,SLGS3		3:SC,SLGS2		4:bi,Z3		5:SC,Z2	
N	α	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)
32	0	7	$6.2\,10^0$	3	$3.7\,10^1$	4	$4.2\,10^1$	4	$3.6\,10^1$	7	$3.9\,10^1$
64		7	$3.7\,10^1$	4	$4.3\,10^2$	4	$3.5\,10^2$	4	$3.0\,10^2$	8	$3.9\,10^2$
128		6	$2.5\,10^2$	4	$5.3\,10^3$	8	$8.1\,10^3$	5	$4.8\,10^3$	12	$6.9\,10^3$
32	0.04	8	$7.8\,10^0$	4	$5.1\,10^1$	4	$4.3\,10^1$	5	$4.6\,10^1$	9	$5.6\,10^1$
64		14	$8.0\,10^1$	4	$4.3\,10^2$	5	$4.5\,10^2$	8	$6.2\,10^2$	14	$7.1\,10^2$
128		33	$1.4\,10^3$	8	$1.1\,10^4$	7	$7.2\,10^3$	26	$2.5\,10^4$	47	$2.7\,10^4$
32	0.06	10	$9.6\,10^0$	4	$5.1\,10^1$	4	$4.1\,10^1$	5	$4.6\,10^1$	10	$6.1\,10^1$
64		22	$1.2\,10^2$	5	$5.5\,10^2$	5	$4.5\,10^2$	12	$9.4\,10^2$	21	$1.1\,10^3$
128		75	$3.3\,10^3$	11	$1.5\,10^4$	7	$7.3\,10^3$	63	$6.1\,10^4$	133	$7.6\,10^4$

Table 1: Iteration counts and cpu-times in seconds for the first test problem with variable conductivity.

		1:bi,SGS		3:SC,SLGS2		6:bi,SC,SLGS2	
N	α	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)
32	0	7	$6.2\,10^0$	4	$4.2\,10^1$	6	$5.9\,10^1$
64		7	$3.7\,10^1$	4	$3.5\,10^2$	6	$5.3\,10^2$
128		6	$2.5\,10^2$	8	$8.1\,10^3$	9	$9.2\,10^3$
32	0.04	8	$7.8\,10^0$	4	$4.3\,10^1$	6	$6.3\,10^1$
64		14	$8.0\,10^1$	5	$4.5\,10^2$	6	$5.4\,10^2$
128		33	$1.4\,10^3$	7	$7.2\,10^3$	9	$9.4\,10^3$
32	0.06	10	$9.6\,10^0$	4	$4.1\,10^1$	6	$6.3\,10^1$
64		22	$1.2\,10^2$	5	$4.5\,10^2$	6	$5.4\,10^2$
128		75	$3.3\,10^3$	7	$7.3\,10^3$	9	$9.5\,10^3$
32	0.1	16	$1.5\,10^1$	4	$3.5\,10^1$	6	$6.4\,10^1$
64		41	$2.3\,10^2$	6	$5.4\,10^2$	6	$5.4\,10^2$
128		233	$9.9\,10^3$	6	$6.5\,10^3$	6	$6.5\,10^3$

Table 2: Iteration counts and cpu-times in seconds for the first test problem with variable conductivity (continued). Some data from the previous table are included.

stretched grids. Again, method 3 appears to be insensitive to the amount of stretching for a fixed number of cells. The zebra variants, though less costly than line Gauss-Seidel, do not appear to be competitive. Method 6, though expensive, is the most robust.

3.3 Marine example

The third example is a realistic marine example, described in [8]. Figure 1 shows the resistivity ($1/\sigma$, σ being the conductivity) on a logarithmic scale. A cross section at the y-coordinate of the source is displayed in Figure 2, giving an impression of the grid stretching.

Table 5 lists iteration counts and cpu-times for the various multigrid approaches. Method 5 is left out as it performed less than satisfactory. As in the previous examples, only methods 3 and 6 come anywhere near method 1 in terms of cpu-time. The iteration counts, however,

are not completely independent of the amount of stretching, although they do not deteriorate as dramatically as for the other multigrid schemes.

4 CONCLUSIONS

A multigrid method for electromagnetic diffusion with a cell-wise smoother showed good convergence rates on equidistant grids. Grid stretching, however, slowed down the convergence. Here, we have attempted to construct a more robust solver that can handle grid stretching without a significant increase of the number of iterations. We considered line relaxation with and without semi-coarsening. The number of iterations was reduced significantly in some cases. The higher cost of line relaxation did not, however, lead to a reduction of the required cpu time. Also, none of the schemes showed grid-independent convergence rates on stretched grids. The best results in terms of iteration count were obtained with Line Gauss-Seidel in two coordinate directions and semi-coarsening in the same directions. The one direction in which no coarsening took place, alternated between multigrid cycles. The method was also used as a preconditioner for `bicgstab2`, using 3 multigrid cycles. This appeared to be the most robust approach.

Further details are given in [5]. Other suggestions for improvement are listed in [8].

REFERENCES

[1] D.N. Arnold, R.S. Falk, and R. Winther. Multigrid in H(div) and H(curl). *Num. Math.*, **85**, 197–217, (2000).

[2] M. Clemens and T. Weiland. Discrete electromagnetism with the Finite Integration Technique. *Progress in Electromagnetic Research (PIER)*, **32**, 65–87, (2001).

[3] H.H. Gutknecht. Variants of BiCGStab for matrices with complex spectrum. *SIAM J. Sci. Comput.*, **14**, 1020–1033, (1993).

[4] R. Hiptmair. Multigrid for Maxwell's equations. *SIAM J. Num. Anal.*, **36**, 2040–225, (1998).

[5] T.B. Jönsthövel. Improvements on a multigrid solver for 3-D electromagnetic diffusion. *Master's Thesis, Applied Mathematics, Delft University of Technology*, (2006).

[6] W.A. Mulder. A new multigrid approach to convection problems. *J. Comput. Phys.*, **83**, 303–323, (1989).

[7] W.A. Mulder. A multigrid solver for 3-D electromagnetic diffusion. *Geophys. Prospecting*, in press, (2006).

[8] W.A. Mulder. Geophysical modelling of 3D electromagnetic diffusion with multigrid. *Computing and Visualization in Science*, to appear.

[9] C.A. Thole and U. Trottenberg. Basic smoothing procedures for the multigrid treatment of elliptic 3-D operators. *Appl. Math. Comp.*, **19**, 333–345, (1986).

[10] U. Trottenberg, C.W. Oosterlee and A. Schüller. *Multigrid*, Academic Press, London, (2000).

[11] H.A. van der Vorst, BI-CGSTAB: a fast and smoothly converging variant of bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Comput.*, **13**, 631–644, (1992).

[12] T. Washio and C.W. Oosterlee. Flexible multiple semicoarsening for three-dimensional singularly perturbed problems. *SIAM J. Sci. Comput.*, **19**, 1646–1666, (1998).

[13] T. Weiland. A discretization method for the solution of Maxwell's equations for six-components fields. *Electronics and Communications*, **31**, 116–120, (1977).

[14] G. Wittum. Multi-grid methods for Stokes and Navier-Stokes equations with transforming smoothers: algorithms and numerical results, *Num. Math.*, **54**, 543–563, (1989).

[15] K. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. Antennas and Propagation*, **16**, 302–307, (1966).

		1:bi,SGS		2:bi,SLGS3		3:SC,SLGS2		4:bi,Z3		5:SC,Z2	
N	α	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)
32	0	7	$6.2\,10^0$	4	$5.1\,10^1$	5	$5.2\,10^1$	5	$4.0\,10^1$	11	$6.7\,10^1$
64		7	$3.6\,10^1$	4	$4.3\,10^2$	9	$8.1\,10^2$	4	$3.2\,10^2$	21	$1.1\,10^3$
128		7	$3.0\,10^2$	4	$5.4\,10^3$	19	$2.0\,10^4$	4	$3.8\,10^3$	41	$2.3\,10^4$
32	0.04	10	$9.5\,10^0$	4	$5.1\,10^1$	5	$5.4\,10^1$	5	$4.6\,10^1$	12	$6.8\,10^1$
64		16	$8.4\,10^1$	4	$4.3\,10^2$	9	$8.2\,10^2$	8	$6.3\,10^2$	20	$1.0\,10^3$
128		38	$1.6\,10^3$	9	$1.2\,10^4$	15	$1.6\,10^4$	23	$2.2\,10^4$	56	$3.2\,10^4$
32	0.06	12	$1.1\,10^1$	4	$5.1\,10^1$	6	$6.3\,10^1$	6	$5.5\,10^1$	14	$8.4\,10^1$
64		25	$1.4\,10^2$	6	$6.6\,10^2$	9	$8.1\,10^2$	12	$9.6\,10^2$	27	$1.4\,10^3$
128		83	$3.5\,10^3$	16	$2.2\,10^4$	14	$1.5\,10^4$	69	$6.7\,10^4$	248	$1.4\,10^5$

Table 3: Iteration counts and cpu-times in seconds for the second test problem with variable permittivity and a discontinuous solution.

		1:bi,SGS		3:SC,SLGS2		6:bi,SC,SLGS2	
N	α	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)
32	0	7	$6.2\,10^0$	5	$5.2\,10^1$	6	$6.3\,10^1$
64		7	$3.6\,10^1$	9	$8.1\,10^2$	9	$8.1\,10^2$
128		7	$3.0\,10^2$	19	$2.0\,10^4$	15	$1.6\,10^4$
32	0.04	10	$9.5\,10^0$	5	$5.4\,10^1$	6	$6.2\,10^1$
64		16	$8.4\,10^1$	9	$8.2\,10^2$	9	$8.1\,10^2$
128		38	$1.6\,10^3$	15	$1.6\,10^4$	15	$1.6\,10^4$
32	0.06	12	$1.1\,10^1$	6	$6.3\,10^1$	6	$6.4\,10^1$
64		25	$1.4\,10^2$	9	$8.1\,10^2$	9	$8.1\,10^2$
128		83	$3.5\,10^3$	14	$1.5\,10^4$	15	$1.6\,10^4$
32	0.1	19	$1.8\,10^1$	6	$6.4\,10^1$	6	$6.4\,10^1$
64		55	$3.1\,10^2$	9	$8.2\,10^2$	9	$8.3\,10^2$
128		325	$1.4\,10^4$	13	$1.4\,10^4$	12	$1.3\,10^4$

Table 4: Iteration counts and cpu-times in seconds for the first test problem with variable permittivity (continued). Some data from the previous table are included.

		1:bi,SGS		2:bi,SLGS3		3:SC,SLGS2		4:bi,Z3		6:bi,SC,SLGS2	
$h_{\min}$	α	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)	iter	cpu (s)
50	0.035	15	$6.1\,10^2$	5	$7.1\,10^3$	14	$1.5\,10^4$	7	$6.9\,10^3$	12	$1.3\,10^4$
20	0.059	75	$3.2\,10^3$	14	$2.0\,10^4$	14	$1.5\,10^4$	31	$3.1\,10^4$	12	$1.3\,10^4$
10	0.074	163	$7.0\,10^3$	26	$3.7\,10^4$	15	$1.7\,10^4$	78	$7.7\,10^4$	15	$1.7\,10^4$
5	0.089	409	$1.8\,10^4$	44	$6.3\,10^4$	21	$2.3\,10^4$	165	$1.6\,10^5$	18	$2.0\,10^4$

Table 5: Iteration counts and cpu-times in seconds for the realistic marine example.

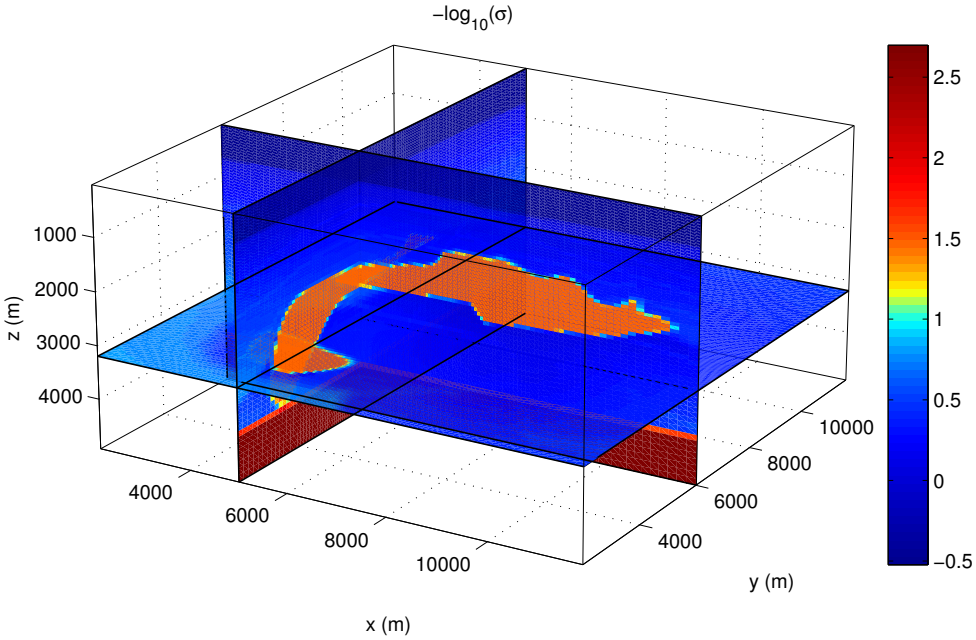

Figure 1: Model for a marine example with a resistive salt body ($h_{\min} = 50\,\mathrm{m}$). The sea bottom has a depth of about $600\,\mathrm{m}$.

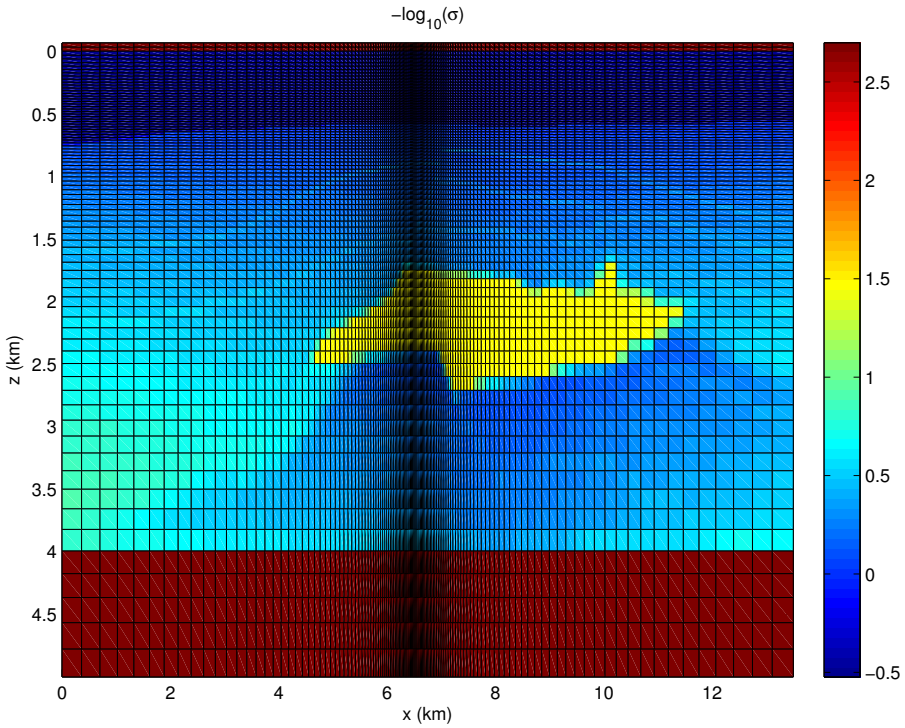

Figure 2: Cross section of a subset of the model at $y = 6500$ for $h_{\min} = 20\,\mathrm{m}$.