Chapter 9
Discussion

In the very first chapter we set the challenge for this thesis: to develop new numerical techniques, such that it is possible to do large scale computations on steady states of the global ocean circulation with sufficient accuracy.

In fact the improvement of the numerics was only the third of three goals of a larger research project. The major aim is the reconstruction of the time-mean absolute velocity field of the global ocean circulation. The way to go was to enhance the existing thermohaline ocean model THCM, which lead to three subgoals:

1. improve the physics in the model,
2. tune the model using observations,
3. speed-up the numerics.

The research on the first two subgoals was carried out by Terwisscha van Scheltinga [75], who incorporated the advanced mixing schemes of Section 2.2.2 in THCM, and developed an implicit data-assimilation algorithm, called I4DVAR. We described data-assimilation and the corresponding extra conditions on the solver in Section 2.6. We focussed on (3) the numerics. The bottleneck in the computations of THCM at the start of the project was the solution of large linear systems involving the Jacobian matrix. Hence the plan to reach goal (3) was to develop a new solver, that

3a) gives a considerable speed up of the computations,
3b) allows for parallelization,
3c) combines with data-assimilation.

In Chapter 2 we described the ingredients of THCM: the (discrete) equations, numerical continuation of steady states and the structure of the Jacobian matrix. However in a large part of the thesis we focussed on the somewhat easier saddle point problem (see Section 2.5). We did so, because iterative solvers have a hard time in solving the saddle point problem, probably for the same reason they have difficulties with the ocean matrix. In the preceding chapters of this thesis we investigated several methods to solve either the saddle point or the ocean problem and in the present chapter we will summarize the results, draw conclusions and evaluate to what extend we reached our goals.
Direct and iterative methods are hard to combine for the saddle point problem

We started off searching the new solver in a combination of techniques from iterative and direct methods, which we described in Chapter 3. Hence we studied the direct solution of saddle point matrices in Chapter 4. We proposed a new algorithm that computes an ordering for the matrix, that is feasible, gives sparse factors and it is numerically stable. One of the nice things of the algorithm is that it keeps much of structure of the saddle point problem during the factorization phase. Moreover we can prove that the size of the matrix elements encountered during factorization is bounded linearly by the size of the matrix.

We tried to transform this exact factorization into an incomplete factorization, but all attempts failed. The main problem is that the factorization is very sensitive to dropping in the divergence part of the saddle point problem. The number of iterations in the Krylov method grows rapidly even if we drop only a few small elements.

Similar problems were encountered in Chapter 5, where we combined an incomplete LU factorization with the fill-reducing orderings as used in direct methods. We introduced a new type of equivalent matrix ordering: the bare-branch ordering. The ordering can be used in an ordered incomplete LU factorization algorithm, which uses both exact and approximate elimination. The performance of the algorithm is good for the equations that involve only one type of variable, like the Poisson equation. However the results for the saddle point problem are disappointing: the factorization has either a huge amount of fill or gives bad convergence in the Krylov space.

We conclude that, notwithstanding the remark in [65] that the combination of techniques from direct and iterative methods is the way to go, it is hard to successfully combine both approaches. This holds at least for the saddle point problem, but probably for a wider class of partial differential equations that involve different types of variables and constraints.

Coupled partial differential equations need solvers that respect the structure of the equations

In Chapter 6 we showed that for different types of saddle point problems, solvers that respect and even exploit the differences between the involved variables, perform well. We proposed two new preconditioners for the saddle point matrix: one based on grad-div adding, the other on artificial compressibility. Both preconditioners give the velocity and the pressure nodes a different treatment. We compared the preconditioners to other well-known saddle point preconditioners that take benefit from the structure of the matrix. Eigenvalue analysis and numerical experiments showed that our preconditioners are good alternatives for existing methods. Nevertheless the solution of the involved grad-div added matrix \((A + \omega B^T B)\) requires some attention.

The positive results of approaches that exploit the structure on the saddle point problem, motivated to use such an approach in the design of a new solver for the Jacobian matrices that occur in THCM. In Chapter 7 we derived the new solver. It starts with a transformation (related to depth-integration) of the pressure and the continuity equation. The transformed matrix is block-lower triangular except for one upper diagonal block, which represents the presence of temperature and salinity in the hydrostatic pressure equation. Either we ignore this block,
which gives the BLOCK-GS preconditioner, or we leave it untouched, which gives the BLOCK-ILU preconditioner. The latter uses an approximation of the exact Schur complement.

Both preconditioners lead to successive solves of an equation for the pressure, a saddle point problem for the horizontal velocity and the depth-averaged pressure, an equation for the vertical velocity and finally a convection-diffusion problem for the tracers temperature and salinity. These subsystems are all quite well known and relatively easy to solve.

In Chapter 8 the BLOCK-GS/ILU solver is used in large scale ocean flow problems. The results show that this new solver considerably reduces the memory requirements and the computation time.

Based on the experience with both saddle point problems and the ocean matrices, we conclude that complex coupled partial differential equations require tailored solvers that respect and benefit from the structure of the equations.

Explicit methods can inspire the solvers for implicit methods

The BLOCK-GS solver shows a striking similarity to the barotropic-baroclinic splitting that is widely used in explicit ocean models. The same holds for one of the saddle point preconditioners that we introduced in Chapter 6. Artificial compressibility is used in explicit time step methods for incompressible flow problems, however it appears to be useful in the preconditioner as well.

The explanation for this phenomenon might be that the methods used in explicit time stepping resemble the dependencies in the solution of the problem: the most important variables are solved first. In implicit time stepping the time steps are much larger (even infinite in a steady state solution), but the dependencies are probably largely the same, which can be used in the design of a solver for the involved Jacobian matrices.

The new solver is a major step forward in bifurcation analysis of ocean flows

The main goal of this thesis was the development of a new, faster solver for the ocean systems. We already mentioned the design of the preconditioner in Chapter 7. The results of the preconditioner are above expectations. We compared the performance on four different large scale complex ocean flow problems to the performance of the old solver: MRILU directly applied to the Jacobian matrix. We observed speed-up factors of two orders of magnitude. Moreover the BLOCK-GS/ILU preconditioner uses much less memory and the scaling results are acceptable even for complex evolved flows. In a direct comparison of the two variants, i.e. the straightforward BLOCK-GS and the more complex BLOCK-ILU solver (nested iterations, approximation of the Schur-complement), we see that BLOCK-GS is the better one. It uses more outer iterations, but each one of the iterations is cheaper than an iteration with BLOCK-ILU.

There is no doubt we fulfilled condition (3a): the speed-up is considerably large. The parallelization (condition (3b)) follows naturally, as we exposed in Section 7.5.2. The parallel code is still under development. The combination with the data-assimilation code (3c) appeared
to be no problem either. In contrary, due to a stronger diagonal, the performance of the BLOCK-GS/ILU solver is even better than on steady state Jacobian matrices [75].

All together we conclude that the new solver allows us to perform computations with a much higher resolution, which makes it a major step forward in the analysis of the thermohaline ocean circulation.

**Methodology ready for reconstruction of the time-mean absolute velocity field of the global ocean circulation**

Of the three main goals, in this thesis we realized (3) a fast solver for the ocean equations. The other two, i.e. (1) the improvement of the physics and (2) the use of observations in the model, were realized by Terwisscha van Scheltinga at the IMAU [75]. Together we delivered an up-to-date ocean model THCM that is able to do the necessary large scale computations on the thermohaline circulation. The model can be combined with data-assimilation in I4DVAR using observations of the geoid an the sea surface height. Accurate data for the geoid will be provided by the GOCE mission. Unfortunately the launch of the satellite has been delayed to the end of 2007. Hence we have to wait for the data, but the methodology is ready for the reconstruction of the time-mean absolute velocity field of the global ocean circulation.