Overlapping Schwarz Domain Decomposition Methods for Implicit Ocean Modelling

A Robust Approach for Thermohaline Circulation Models

D.C.H. Gruntjes

Delft University of Technology

TU Delft

Overlapping Schwarz Domain Decomposition Methods for Implicit Ccean Modelling

A Robust Approach for Thermohaline Circulation Models

by

D.C.H. Gruntjes

studentnumber: 5831105

Instructor:	dr. J. Thies
Project Duration:	April, 2025 - July, 2025
Faculty:	Faculty of Electrical Engineering, Mathematics & Computer Science, TU Delft

Committee:	dr. J. Thies & Prof. H.M. Schuttelaars
Cover:	Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA under CC BY-NC 2.0 (Modified)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Layman's summary

The oceans cover more than 70% of our planet. Their currents play a big role in our climate by moving heat across the globe. Scientists try to understand and predict these movements using computer models. However, these models are so detailed and complex that running them requires a lot of computing power and clever mathematical techniques.

This paper looked at specific techniques to speed up such ocean simulations. The problem is that to get accurate results, you need to solve enormous systems of equations that describe how water moves, how warm or salty it is, and how it all interacts over time. Doing this directly is too slow. That is why the studied method breaks the big ocean problem into smaller chunks, which are easier and faster to solve.

In this paper, a method called the overlapping Schwarz method was tested. It works by cutting the ocean into smaller pieces that share a bit of overlap, solving each piece separately, and then it puts the different solution pieces back together. This approach can be done on many computers at once, saving time. Some more advanced versions of the Schwarz method add an extra step to ensure the whole ocean stays well-connected, even though it has been divided into pieces.

The research starts by testing this method on a simpler problem to see how changing the size of the overlaps, or the number of pieces the ocean gets divided into, affects the speed of solving. As was expected, making the overlaps bigger or adding a clever global correction helped the calculations finish in fewer steps.

Then, the Schwarz method was tried on a complete 3D ocean model that includes wind pushing the water, heat and salt changing its density. It turned out that when temperature or salt effects became stronger, solving the equations became much harder. The method still helped, but more advanced versions with global corrections are needed to keep things efficient.

In short, this paper shows that breaking the ocean into overlapping parts is a powerful way to speed up climate simulations, but we need to be careful and use extra tricks when dealing with strong temperature and salinity effects. This helps scientists run bigger, more realistic ocean models, which is vital for understanding climate change.

Summary

This paper investigates the use of overlapping Schwarz domain decomposition preconditioners to solve fully implicit three-dimensional ocean circulation models efficiently. These are essential for understanding climate-relevant ocean dynamics but pose formidable computational challenges due to their spatial and temporal dimensions combined with ill-conditioned systems typical of realistic ocean climate models.

The paper begins by describing an established ocean model which solves the primitive equations coupled with temperature and salinity convection, subject to wind stress, heat, and salt flux forcing. The model is discretized using a second-order control volume scheme on a hybrid B-C Lorenz grid. This results in a large non-linear system whose Jacobian features tightly coupled momentum, mass, and tracer blocks. Solving this system relies on Newton-Krylov iterations, whose convergence depends heavily on effective preconditioning.

The paper presents numerical experiments. It first examines a Laplace system to systematically study how the overlap size and number of subdomains affect solver convergence. Results confirm that increased overlap reduces iterations and that two-level Schwarz methods with coarse corrections outperform one-level approaches, especially as mesh refinement and subdomain counts increase.

Building on these insights, the study extends to a fully implicit ocean model that solves the primitive equations with coupled temperature and salinity dynamics under wind, heat, and salt flux forcing. Applying a one-level Schwarz preconditioner, it becomes clear that as temperature and salinity forcing intensify, the Jacobian's conditioning deteriorates sharply which leads to a steep rise in GMRES iterations. Additional experiments isolating individual tracers indicate that the temperature exerts a particularly strong influence on convergence, though buoyancy interactions complicate this attribution.

The work concludes that while one-level Schwarz methods improve performance on simpler problems, fully implicit ocean models under strong tracer forcing require advanced two-level Schwarz preconditioners with robust coarse spaces to maintain efficiency and scalability. Future work is encouraged to incorporate coarse grid corrections explicitly to better handle these challenges and enable more detailed and computationally feasible ocean simulations.

Contents

1

Introduction

The world's oceans cover more than 70% of the world, and thus, understanding the flows and structure of its circulation is of great importance when trying to understand and predict the Earth's evolving climate. Numerical models are irreplaceable tools for this understanding to come about. Yet, due to the complex nature of the physics involved, in combination with the temporal and spatial scales necessary for reliable predictions, the computational requirements for realistic models quickly become exceptionally large.

This high computational cost within realistic ocean models imposes formidable challenges. Modern ocean circulation models involve solving the primitive equations of the system under Boussinesq and hydrostatic[3] approximations that are coupled to advection-diffusion equations for temperature and salinity. Many popular models have explicit schemes for solving much of the system to avoid having to solve large and costly implicit systems. However, this comes with restrictions on the time step size due to stability concerns. Fully implicit formulations circumvent these restrictions and enable direct computation of steady-state or long-time horizon solutions. However, they often need to solve an ill-conditioned linear system, which may have degrees of freedom up to the millions.

In the literature, overlapping Schwarz domain decomposition methods are a promising preconditioning strategy to address these challenges. By decomposing the global domain into overlapping subdomains, these Schwarz methods allow localised problems to be solved in parallel while using coarse corrections to maintain the coupling necessary for global convergence. The addition of coarse grid corrections in two-level Schwarz variants further enhances their ability to propagate global information, reducing iteration counts even as the number of subdomains grows. The Schwarz preconditioners have demonstrated their scalability and robustness[4] specifically when equipped with advanced coarse spaces such as GDSW or adaptive GDSW that are designed to handle the strong heterogeneities common in ocean models.

This paper aims to investigate how overlapping Schwarz preconditioners impact the efficiency and scalability of solving fully implicit three-dimensional ocean circulation models. First, studying simplified Laplace systems to systematically examine how overlap size and the number of subdomains influence convergence behaviour in one-level and two-level Schwarz methods. The results from this analysis then allow for the isolation of the numerical properties of the preconditioners without directly analysing more complex interactive systems. After that, the full ocean

model is explored, and the impact of salinity and temperature forcing on the convergence characteristics of the Newton-Krylov solver is analysed. This is done by focussing on the effect they have on the conditioning of the system's Jacobian and the resulting consequences for the convergence of the iteration count.

Chapter 2 provides a detailed framework of the model used in this study, including its governing equations. Next, the third Chapter will discuss the theoretical background surrounding the research on ocean current modelling and the methods for solving them that will be used in the paper. The fourth Chapter will display the paper's numerical results. Lastly, Chapter 5 will discuss and interpret the results of the previous Chapter.

2

Model Description

This chapter presents the formulation, physical parametrisations, and numerical discretisation of the fully implicit ocean model used in this paper. The model is based on the bifurcation analysis framework introduced by Thies et al.(2009) [12]. The idealized ocean setup from the i-emic repository [11] is adopted, focusing on the "idealized" example case.

2.1. Governing Equations

The ocean model simulates large-scale ocean circulation within a spherical domain bounded by specific longitudes and latitudes. Vertically, the ocean is bounded above by a non-deformable ocean-atmosphere interface and below by a variable bottom topography.

The model is forced at the surface by:

- A zonal wind stress field $(\tau_\lambda, \tau_\phi)$
- A surface heat flux Q_H
- A virtual salt flux Q_S

The zonal wind stress field represents the distribution of the force exerted by the wind on the ocean's surface along the east-west(zonal) direction. Both (Q_H, Q_S) are restoring fluxes that restore the surface temperature and salinity to the prescribed function (T_S, S_S) using restoration time (τ_T, τ_S).

The heat flux Q_H is proportional to the difference in the temperature of the sea surface T and the temperature of the atmosphere T_S.

$$Q_H = -\lambda_T(T - T_S)$$

with λ_T being a constant coefficient.

The salinity flux is also formulated as a very similar restoration condition, namely

$$Q_S = -\lambda_S(S - S_S),$$

with there also being a constant coefficient λ_T within the equation.

Then, the density variations are calculated using the following.

$$\rho = \rho_0 \left(1 - \alpha_T (T - T_0) + \alpha_S (S - S_0)\right),$$

where α_T and α_S are the volumetric expansion coefficients and (ρ_0, T_0, S_0) are reference quantities.

The model solves the full three-dimensional primitive equations utilizing their standard formulations (see De Niet et al.(2007)[3]) and will thus not be repeated here. For the momentum mixing, the Laplace friction is used with constant horizontal and vertical momentum (eddy) viscosities. Tracer diffusion for temperature and salinity will employ constant horizontal and vertical diffusivities K_H and K_V.

Ocean current models often include the Boussinesq approximation. This approximation assumes that density variations are minor compared to the reference density ρ_0. This approximation allows for a simplification of mass conservation by reducing it to a divergence-free velocity field. Thus reducing numerical and mathematical complexity and allowing models to better approximate ocean circulation dynamics. However, due to the layout of this model, the density condition will not hold. Thus, to replace the effects of the Boussinesq approximation, some parametrizations, particularly convective adjustment, compensate for the small-scale processes that can not be included directly within the mesh.

2.2. Domain and Boundary Conditions

The idealized example utilises a rectangular base of dimensions $L_x \times L_y \times H$,

$$(x, y) \in [0, L_x] \times [0, L_y], \quad z \in [-H, 0].$$

$z = 0$ represents the ocean's surface while $z = -H$ represents the bottom.

No slip conditions are present on all boundaries except the top surface, which has a free slip condition. The temperature and salinity fluxes at the top surface are either homogeneous Neumann or spatially varying Neumann boundary conditions to force an overturning circulation. Other boundaries are treated as impermeable.

2.3. Numerical Discretization

The model equations are discretised in space using the second-order accurate control volume method. Build on a hybrid grid formed with an Arakawa B-grid in the zonal and meridional plane and a C-grid in the vertical one, known as a Lorenz grid (see Figure 2.1).

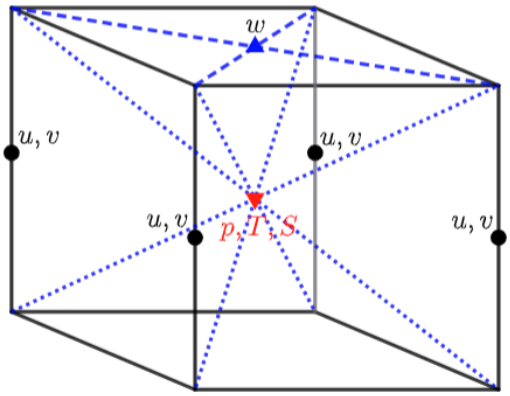


Figure 2.1: Grid cell showing positioning of the variables

The spatially discrete model can be written as follows.

$$\mathbf{M}\frac{d\mathbf{u}}{dt} = \mathbf{F}(\mathbf{u}) = \mathbf{L}(\mathbf{u}) + \mathbf{N}(\mathbf{u}, \mathbf{u}),$$

where $\mathbf{u}$ contains all of the unknowns of the system(u, v, w, p, T, S) at every cell of the grid. Giving a dimension of $d = 6NML$. The operators $\mathbf{M}$ and $\mathbf{L}$ are both linear, while $\mathbf{N}$ collects the nonlinear terms of the equation.

A steady state solution dependant on the vector of parameters $\mathbf{p}$ satisfying

$$\mathbf{F}(\mathbf{u}, \mathbf{p}) = \mathbf{0}.$$

is sought.

One of the parameters called μ is varied to find branches of steady-state solutions, and the pseudo-arclength continuation method[8] is used. Where the branches $(\mathbf{u}(s), \mu(s))$ are parametrised by an additional "arclength" variable s. This allows for the use of the following equation

$$\dot{\mathbf{u}}_0^T(\mathbf{u} - \mathbf{u}_0) + \dot{\mu}_0(\mu - \mu_0) - \Delta s = 0$$

which is appended to the system. $(\mathbf{u}_0, \mu_0)$ is a known solution or previously computed steady state and Δs is the step length. To solve this, a predictor-corrector method is applied. With a secant method for predictions and a Newton-Raphson method for correction. Each Newton step involves solving

$$\mathbf{J}(\mathbf{u}^k)\begin{pmatrix}\Delta\mathbf{u}_{k+1}\\ \Delta\mu_{k+1}\end{pmatrix} = -\begin{pmatrix}\mathbf{F}(\mathbf{u}_k, \mu_k)\\ \Delta s - \dot{\mathbf{u}}_0^T(\mathbf{u}^k - \mathbf{u}_0) + \dot{\mu}_0(\mu^k - \mu_0)\end{pmatrix},$$

where $\mathbf{J}$ is the Jacobian matrix obtained from the previous two equations.

The structure of this Jacobian matrix is the following block matrix:

$$\mathbf{J} = \begin{pmatrix} \mathbf{A}_{uv} & \mathbf{B}_1 & \mathbf{G}_{uv} & 0 \\ 0 & 0 & \mathbf{G}_w & \mathbf{B}_2 \\ \mathbf{D}_{uv} & \mathbf{D}_w & 0 & 0 \\ \mathbf{B}_3 & \mathbf{B}_4 & 0 & \mathbf{A}_{TS} \end{pmatrix}$$

The first row represents the horizontal momentum equations. $\mathbf{A}_{uv}$ represents the convection-diffusion and the Coriolis operators. $\mathbf{B}_1$ represents the influence of the vertical velocity on the horizontal momentum, and $\mathbf{G}_{uv}$ is the gradient acting on the pressure.

The second row represents the hydrostatic pressure with $\mathbf{G}_w$ being the gradient acting on the pressure and $\mathbf{B}_2$ representing the couplings of temperature and salinity such that the combination forms the density.

The third row handles the conservation of mass. $\mathbf{D}_{uv}$ denotes the divergence for the horizontal directions, and $\mathbf{D}_w$ is the divergence for the vertical direction.

Lastly, the final row represents the tracer equations for temperature and salinity. Here $\mathbf{B}_3$ and $\mathbf{B}_4$ model the influence of the horizontal and vertical velocity respectively. And $\mathbf{A}_{TS}$ is the convection and mixing of the heat and salt within the ocean.

Thus, the linear systems arising in each Newton iteration is

$$\mathbf{J}\mathbf{x} = \mathbf{b}$$

This will be a large and ill-conditioned system.

To solve this efficiently, Krylov subspace methods, specifically the Generalized Minimal Residual (GMRES) method, are used. GMRES works by iteratively building approximate solutions in successive Krylov subspaces

$$\mathcal{K}_m(\mathbf{J}, r_0) = \mathrm{span}\{r_0, \mathbf{J}r_0, \mathbf{J}^2 r_0, \ldots, \mathbf{J}^{m-1} r_0\},$$

with $r_0 = b - \mathbf{J}$ being the initial residual, and then minimizes the residual norm over the subspaces.

The convergence rate for GMRES depends mainly on the conditioning of the Jacobian matrix $\mathbf{J}$. If the matrix has eigenvalues close to each other and not close to 0 then GMRES converges well. However, due to the ill-conditioned nature of the Jacobian matrix, the convergence will be slow.

To address this problem, a preconditioner $\mathbf{P}$ is utilised. It transforms the system into

$$\mathbf{P}^{-1}\mathbf{J}\mathbf{x} = \mathbf{P}^{-1}\mathbf{b}$$

where $\mathbf{P}$ is chosen such that it makes the $\mathbf{P}^{-1}\mathbf{J}$ better conditioned. A well-designed preconditioner is chosen to approximate $\mathbf{J}$ so that it reduces the number of GMRES iterations.

2.4. Preconditioning and Parallel Implementation

To find a fitting preconditioner that accelerates the convergence of the Krylov solver and ensures good performance, a Schwarz domain decomposition preconditioner is employed.

The Schwarz method divides the global domain into overlapping subdomains. This makes multiple localized problems that can be solved in parallel. A Schwarz preconditioner can contain two levels:

- **Local Subdomain Solve**: The first level uses an additive Schwarz algorithm that solves each subdomain separately and then extends the results from this local subdomain back into the global system.
- **Global Coarse Correction**: The second level is a global coarse grid correction that accounts for the coupling of the system. This correction is crucial for the convergence behaviour of the ocean model to prevent an excessive number of Newton iterations.

Research Goals

This paper investigates the role and performance of overlapping Schwarz domain decomposition preconditioners in fully implicit three-dimensional ocean circulation models. And to achieve this the following question guides the research:

- How do overlapping Schwarz domain decomposition preconditioners affect the scalability and efficiency of solving the fully implicit three-dimensional ocean circulation model?

To address this overarching question, the study focuses on two key sub-questions:

- How do variations in overlap size and the number of subdomains affect solver convergence and iteration counts in Schwarz preconditioner methods?
- How do the salinity and temperature tracers affect the convergence within the implicit ocean model?

These questions build upon the model formulation, numerical discretization, and preconditioning strategies described in Sections 2.1 through 2.4. They will be revisited in later chapters when analysing the experimental results and drawing conclusions about the effectiveness of different domain decomposition strategies for this class of ocean models.

3

Literature review

This chapter covers research regarding implicit ocean modelling and the parallel solving of their large, sparse linear systems, focusing on domain decomposition methods. It begins with an introduction to frameworks for an ocean model, concentrating on the one used in the paper.

3.1. Implicit Ocean Models and Large Sparse Linear Systems

Modern general circulation models (GCMs) for ocean currents solve Navier-Stokes equations under Boussinesq and hydrostatic approximations and tracer transport equations for temperature and salinity. The equations are typically discretised on structured grants that advance in time and utilise a variety of numerical methods. Depending on the choice of time integration, these models will work with implicit or explicit approaches, each with trade-offs in terms of stability, computational cost and accuracy for long timescales.

Most explicit models only utilise explicit methods on certain parts of the system, like the tracer equations of salinity and temperature, while solving the pressure implicitly.

The Parallel Ocean Program (POP)[10] model, which is used in the Community Earth System Model (CESM)[7], one of the most widely used climate models, utilises explicit methods for solving horizontal tracer equations while using implicit schemes for vertical diffusion and barotropic surface solvers. Regional Ocean Modelling Systems(ROMS) established by Shchepetkin et al. (2015)[9] is another example. It employs a split explicit, implicit approach utilising an algorithm that explicitly solves the tracer equations. While separately applying implicit methods for diffusion terms and mass continuity.

Within explicit models, timesteps are constrained by the stability conditions of the methods. However, implicit formulations circumvent this limitation, allowing for larger time steps or even the discovery of steady-state solutions. However, this results in a large non-linear problem that needs to be solved. This solving is mainly done by utilising some version of Newton's method.

J.Thies et al. (2009)[12] presents such a model based upon a thermohaline circulation model (THCM). It solves this with variables representing momentum, mass, and tracer mixing equations in three dimensions. This involves solving very large sparse systems containing up to millions of degrees of freedom. The Newton methods employed in this model are Namely Newton-Krylov methods, which solve the Jacobian system iteratively. This was done by integrating with the

parallel linear solvers of Trilinos[6] to compute the steady states of the model.

The THCM model that was first utilised had a nested splitting approach with separate preconditioners, similar to how models mentioned above would split the equations to be solved explicitly or implicitly; this simplified the structure but led to high iteration counts[5] and more complex convergence patterns. This convergence problem then motivates the creation of adaptation of this model that does not split the physics into separate preconditioners but instead couples the system through a single monolithic preconditioner that spans multiple physical processes simultaneously, improving the convergence.

3.2. Domain Decomposition

Domain decomposition methods partition the global computational domain into smaller subdomains, enabling parallel local solves that allow for more efficient parallel computations on large domains. Schwarz decompositions specifically allow for iterative convergence across these subdomains by joining the local solvers and coarse-level corrections to ensure global coupling.

3.2.1. Overlapping Schwarz Method

A Schwarz domain decomposition treats a fully coupled system globally but utilises partitions of the domain into overlapping subdomains to solve systems. There are two levels with which the Schwarz preconditioner works. A one-level solves a local problem on each subdomain with overlap, and a two-level is a coarse problem that couples them globally across subdomains.

With a global system $A\mathbf{x} = b$ each subdomain i has a restriction operator R_i that makes a local matrix $A_i = R_i A R_i^T$. The first level Schwarz preconditioner solves the problem locally for the A_i in parallel. And the second level coarse correction is added via an interpolation operator R_0 that gives the two level preconditioner.

$$M^{-1} = R_0^T A_0^{-1} R_0 + \sum_i R_i^T A_i^{-1} R_i$$

with the coarse matrix $A_0 = R_0 A R_0^T$.

The overlapping nature of the subdomains ensures that interface effects are pushed further away from the actual domain while the coarse space accelerates global convergence.

3.2.2. GDSW and AGDSW Coarse Spaces

The formulation of the coarse space plays a vital role in determining the efficiency and robustness of a Schwarz preconditioner. A popular method for coarse spaces is the GDSW (Generalized-Dryja-Smith-Widlund). It functions by having basic functions that are discrete harmonic extensions of vertecies or degrees of freedom, chosen to be energy minimal. For this method, a coarse basis function that is constant on each edge and face of the subdomain and harmonic elsewhere inside the subdomains. This creates a small coarse problem and a condition number bound that is independent of the number of subdomains[4]. This allows for a more effective parallel scaling since the number of subdomains determines the number of processors that can work on the problem efficiently and simultaneously.

PDE coefficients can vary by order of magnitude within ocean models and other applications. This significant difference can cause the more classical coarse space methods to converge less quickly, leading to excessive strains on computational resources. Heinlein et al. (2018)[4] showcases the Adaptive GDSW (AGDSW) coarse spaces that address this problem by modifying

the coarse space using eigenvalue problems. The concept behind this is to identify problematic ”low energy” modes on each interface and include them globally. This is done by solving generalised eigenvalue problems on each edge or face of the subdomain decompositions and selecting the eigenvectors with small eigenvalues. These nodes are then harmonically extended into the subdomains and added to the coarse basis of the system.

This causes AGDSW to have a preconditioned system with a condition number that is independent of the differences between the coefficients. Instead, the condition number would depend on the system's user-assigned eigenvalue tolerance. The new system will “always contains the classical GDSW coarse space by construction”[4]; thus, it can only improve the system's accuracy, not worsen it. By carefully selecting the eigenmodes, AGDSW can maintain moderate dimensionality. Thus, keeping the additions to a minimum ensures optimal convergence behaviour, reducing the number of generalised minimal residual method (GMRES) iterations and pushing performance to scale only with problem size.

3.3. Summary

This chapter reviewed key advancements in parallel solving of large sparse linear systems that arise from implicit ocean models. Some explicit-implicit nested-splitting approaches were shown to result in high iteration counts and complex convergence behaviours. This motivated a shift to monolithicly preconditioned Newton-Krylov methods. Within the domain decomposition techniques, the overlapping Schwarz method provides a scalable parallel solver, which is achieved though a combination of local subdomain solves and global coarse corrections,

Special attention was given to the construction of coarse spaces. The GDSW (Generalized Dryja-Smith-Widlund) approach offers scalable performance with condition number bounds independent of subdomain count. However, classical methods may underperform in systems with highly variable coefficients, which are common in ocean modelling. The Adaptive GDSW (AGDSW) method addresses this by changing the coarse space based on an eigenvalue analyses to ensuring robust performance even under strong coefficient contrasts. AGDSW retains all features of GDSW while adaptively enhancing convergence and maintaining computational efficiency.

4

Numerical Experiments

In this chapter, some numerical experiments involving different variations of the Schwarz method to isolate and confirm the behaviour of the Schwarz preconditioner. The Laplace system is chosen for this because it provides a method for testing due to its similarity to the elliptical behaviour in the ocean model regarding the temperature and salinity parameters within the system. Furthermore, its simplistic setup allows for easy implementation and validation of results without requiring significant amounts of time to set up. These Laplace experiments directly seek to address the first research sub-question:

How do variations in overlap size and the number of subdomains affect solver convergence and iteration counts in Schwarz preconditioner methods?

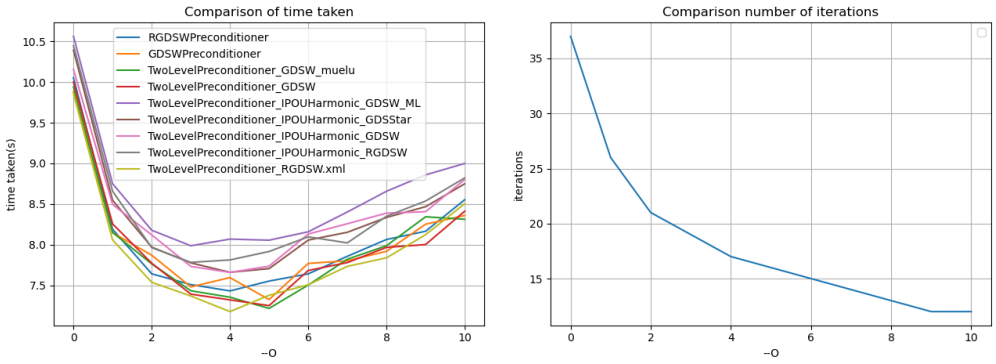
Following the experiments on the Laplace system, attention turns to the full three-dimensional implicit ocean circulation model. The aim here is to investigate how the inclusion of salinity and temperature forcing affects the convergence behaviour of the Newton-Krylov solver. By progressively increasing the strength of these forcing terms, the experiments demonstrate how tightly coupled tracer dynamics introduce considerable challenges for solver convergence. Additional tests are performed to disentangle the individual contributions of salinity and temperature, where each forcing is selectively disabled. This set of experiments is designed to address the second research question directly:

How do the salinity and temperature tracers affect the convergence within the implicit ocean model?

4.1. Laplace

4.1.1. Subdomain Overlap

These numerical experiments aim to isolate the sensitivity of solver performance to the changes in the overlap between subdomains. This was done with a varying set of Schwarz preconditioner configurations that were all implemented utilising the Trilinos library FROSch.

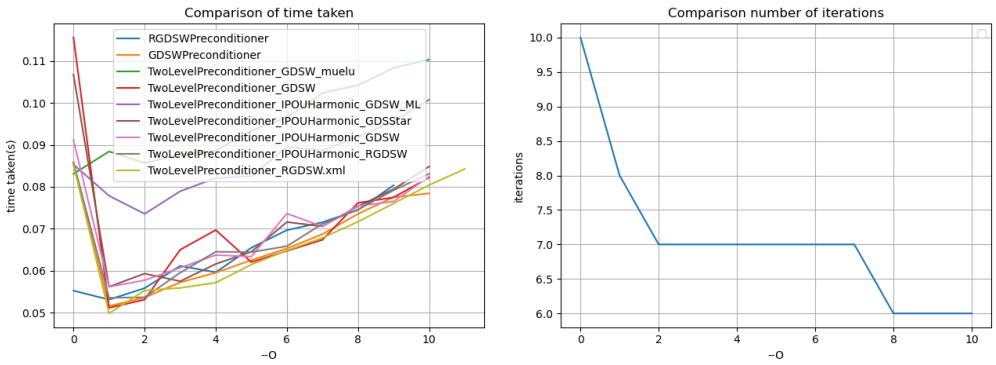


(a) Time(s) **(b)** iterations

Figure 4.1: Results for a 200×200 grid 2d Laplace equation with different Schwarz configurations run with 4 processors on Home laptop. –O signifies the size of the overlap

As shown in Figure 4.1a the different variations improve their speed as the size of the overlap $-$O increases. This increase in speed can be attributed to the need for fewer iterations, as Figure 4.1b illustrates. Here, a significant decrease in the number of iterations happens as the overlap expands from 0, resulting in a decrease in the time needed for the model.

This test was also performed with a smaller grid size to see if different grid sizes affect this pattern of behaviours.



(a) Time(s) **(b)** iterations

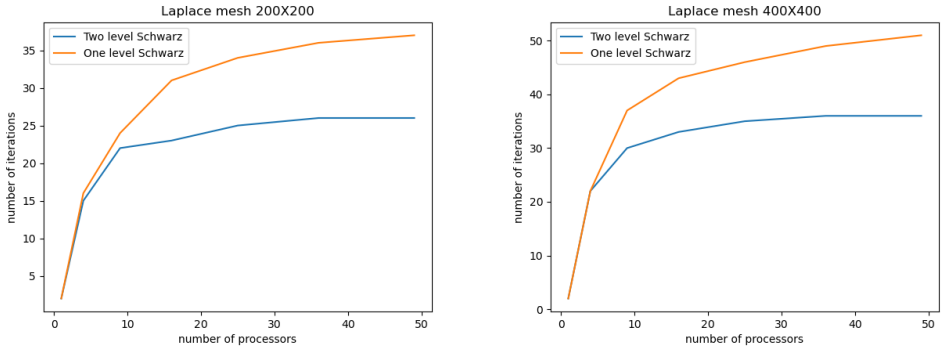
Figure 4.2: Results for a 10×10 grid 2d Laplace equation with different Schwarz configurations run with 4 processors on Home laptop

Although the results were more muddled due to the small size of this test case, the behaviours are still the same as observed with the larger grid size. This allows for a safer assumption that the behaviour of the system will not significantly change when the overlap parameters remain the same, thus allowing the rest of the numerical excrements in the paper to keep the overlap parameter constant without disrupting the reliability and robustness of the results and the conclusions drawn from them.

4.1.2. Subdomains and Mesh Refinement

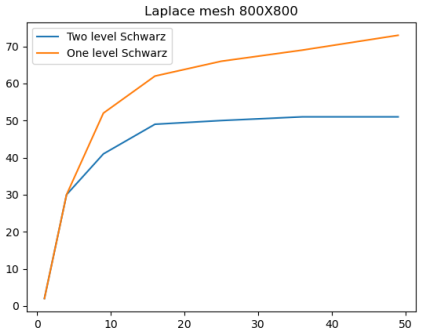
Some other vital conditions to observe are the number of subdomains and the mesh refinement. So, the following numerical experiments will aim to observe the interplay between these two parameters and how they affect one-level and two-level methods differently.

The domain is square, and the subdomains will partition this domain into more square domains to keep the behaviour consistent across different mesh refinements and simplify domain construction. The tested Schwarz methods will be narrowed down to one method for the one-level Schwarz and one for the two-level Schwarz. This is to more closely observe the differences between these two methods as the mesh and subdomain mesh refine. And because of the result from the previous experiment, we can keep the overlap constant for the different methods without interfering with the comparison of the convergence behaviour of the system.



(a) Mesh refinement of 200×200

(b) Mesh refinement of 400×400



(c) Mesh refinement of 800×800

Figure 4.3: Results for the number of iterations needed to solve the laplacian for a level-one and level-two Schwarz method with different numbers of processors ran on DelftBlue supercomputer.

From these results, it can clearly be observed that in terms of iteration count, the scaling for the number of processors is better with the level one method compared to the level two method. This is to be expected because the level two method adds to the convergence of the level one method and thus can not be worse. And the difference that develops between these two methods scales with the refinement of the mesh. Doubling from a difference of 11 at 200 to a difference of 22 at 800.

4.2. Ocean Experiments

4.2.1. Convergence Behaviour

Then, the full model was tested with the one-level FROSch preconditioner. This is the idealised model that was introduced in Chapter 2. It slowly increases the forcing terms for the wind, temperature, and salinity within the system, and at each increase of the forcing terms, a steady state solution is found.

Forcing step	Iterations	Newton steps	Forcing Term
1	29	2	0.001000
2	15	1	0.002320
3	17	1	0.004175
4	35	2	0.006780
5	45	2	0.010220
6	57	2	0.014760
7	70	2	0.020750
8	98	2	0.028660
9	121	2	0.039110
10	146	2	0.052890
11	172	2	0.071080
12	209	2	0.095100
13	251	2	0.126800
14	285	2	0.168700
15	494	3	0.218700
16	603	3	0.268700
17	713	3	0.318700
18	750	3	0.368700
19	1000	4	0.418700
20	3500	14	0.468700
21	3500	14	0.473200
22	4000	16	0.475800
23	–	–	0.476500

Table 4.1: Results of the full model executed, with 4 processors/subdomains on a $32 \times 32 \times 16$ gird, in terms of GMRES iterations, Newton steps and forcing term per continuation step

This process was prematurely terminated due to the extremely long running time. With an increase in the strength of the forcing terms of the system, it converges increasingly slowly as the number of GMRES iterations per newton step increases. Until it hits the manually set cap of 250, it exits the non-linear step, lowers the change in the forcing term and repeats the previous step, thus creating the slowing change in the value of the forcing term per forcing step. Such increases in iteration counts are not the consequence of numerical tolerances or initial conditions. Rather they reflect the underlying challenges of the system and that solving it is no simple task.

So to confirm the origin of the problematic convergence a new numerical experiment is run that fully turns off the forcing terms regarding the temperature and the salinity values. This results in Table 4.2

Forcing step	Iterations	Newton steps	Forcing Term
0	0	0	-
1	11	1	0.001000
2	0	0	0.002405
3	0	0	0.004513
4	0	0	0.007674
5	0	0	0.012420
6	0	0	0.019530
7	0	0	0.030200
8	0	0	0.046200
9	0	0	0.070210
10	0	0	0.106200
11	0	0	0.156200
12	0	0	0.206200
13	0	0	0.256200
14	0	0	0.306200
15	0	0	0.356200
16	0	0	0.406200
17	0	0	0.456200
18	0	0	0.506200
19	11	1	0.556200
20	0	0	0.606200
21	11	1	0.656200
22	0	0	0.706200
23	12	1	0.756200
24	0	0	0.806200
25	18	1	0.856200
26	0	0	0.906200
27	22	1	0.956200
28	0	0	1.000000

Table 4.2: GMRES iterations and maximum forcing term per continuation step without salinity and temperature forcing

From this, it becomes immediately apparent that the main factors in the substantial growth in number literation are the result of either or both the salinity and temperature within the system. To gain more insight into the exact nature of this result, the experiment is repeated but now with only one of the two forcing values disabled, and both cases are recorded in Table 4.3 and Table 4.4.

Forcing step	Iterations	Newton steps	Forcing Term
1	35	2	0.001000
2	19	1	0.002320
3	36	2	0.004175
4	49	2	0.006623
5	64	2	0.009854
6	80	2	0.014120
7	113	2	0.019750
8	132	2	0.027180
9	161	2	0.036990
10	195	2	0.049940
11	242	2	0.067040
12	292	2	0.089600
13	487	3	0.119400
14	588	3	0.156500
15	873	4	0.202600
16	916	4	0.252600
17	1215	5	0.302600
18	1250	5	0.352600
19	2500	10	0.377634

Table 4.3: GMRES iterations and maximum forcing term per continuation step without salinity forcing.

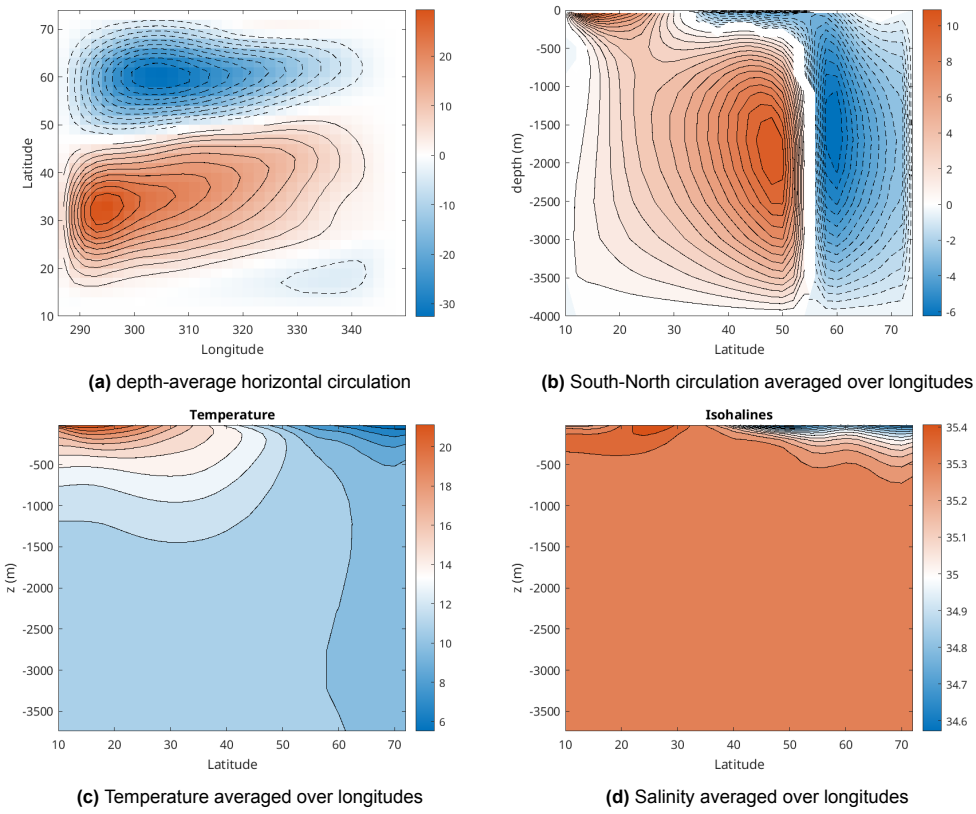
Forcing step	Iterations	Newton steps	Forcing Term
1	14	1	0.001000
2	0	0	0.002405
3	0	0	0.004513
4	21	1	0.007674
5	21	1	0.012120
6	23	1	0.018360
7	52	2	0.027120
8	65	2	0.038700
9	79	2	0.053970
10	96	2	0.074140
11	124	2	0.100800
12	169	2	0.135900
13	207	2	0.182300
14	424	3	0.232300
15	351	2	0.282300
16	398	2	0.332300
17	487	2	0.382300
18	500	2	0.432300
19	500	2	0.482300
20	500	2	0.532300
21	500	2	0.582300
22	500	2	0.632300
23	1000	4	0.682300
24	1250	5	0.732300
25	2500	10	0.782300
26	7020	30	0.794800
27	3260	14	0.801100
28	1250	5	0.809100
29	5520	24	0.811300
30	500	2	0.814300
31	500	2	0.818200
32	500	2	0.823300
33	1500	6	0.830100
34	–	-	0.833200

Table 4.4: GMRES iterations and maximum forcing term per continuation step without temperature forcing.

This implies that the both the salinity and temperature influence the convergence of the system separately but the temperature has a more outsized influence on the convergence of the system. These convergence results will be further analysed within the next Chapter.

4.2.2. Solution States

Before further analysing the solver performance metrics, confirming that the ocean model produces physically realistic and expected results is essential. For this purpose, two runs are examined, one with full forcing and one with temperature forcing fully disabled. Figures 4.4 and 4.5 show the corresponding tracer distributions and global circulation, allowing for confirmation that any solver efficiency gains do not compromise model fidelity.



(a) depth-average horizontal circulation

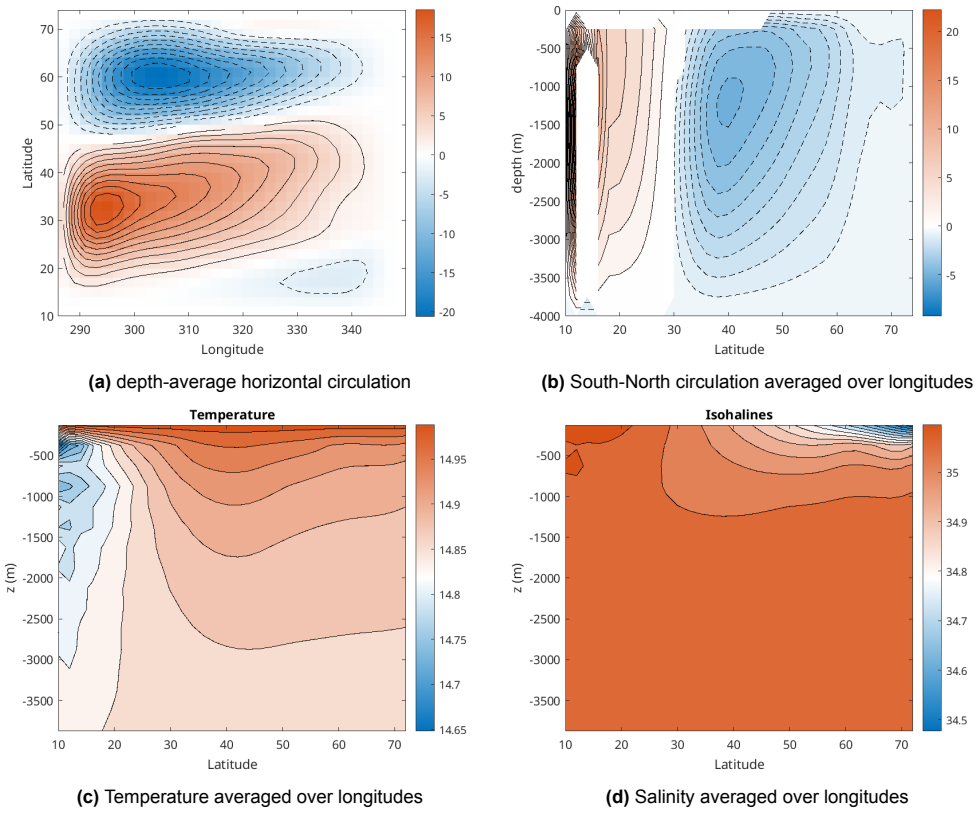
(b) South-North circulation averaged over longitudes

(c) Temperature averaged over longitudes

(d) Salinity averaged over longitudes

Figure 4.4: Solution states under normal forcing

In the fully forced run, the depth-averaged circulation in Figure 4.4a displays expected wind-driven gyres: a clockwise subtropical gyre in the south and a counterclockwise subpolar gyre in the north. The meridional overturning streamfunction within Figure 4.4b shows typical surface flow with deep sinking in the northern region. Corresponding temperature and salinity fields (Figures 4.4c and 4.4d respectively) confirm the presence of expected temperature and salinity stratification, reinforcing the physical credibility of this configuration.



(a) depth-average horizontal circulation

(b) South-North circulation averaged over longitudes

(c) Temperature averaged over longitudes

(d) Salinity averaged over longitudes

Figure 4.5: Solution states under no temperature forcing

Removing temperature forcing in Figure 4.5 maintains the wind-driven gyres of Figure 4.5a, which is expected since wind forcing remains unchanged. However, the meridional overturning circulation is notably altered, with sinking regions shifting to the southern hemisphere, indicating that salinity-driven density differences are now dominant. The temperature field in Figure 4.5c can be seen becoming very different from the normal case, reflecting the lack of temperature forcing. Meanwhile, salinity in Figure 4.5d remains structured, although the absence of temperature gradients weakens stratification.

5

Discussions

This chapter will interpret the numerical results presented in Chapter 4, Connecting them to the underpinning mathematical structures and discuss their implications and limitations.

5.1. Laplace

The initial tests on a simplified Laplace system revealed critical insights into how overlap influences Schwarz method performance. From this, it is found that there is an optimal overlap within the different methods and that they do not differ significantly, even between level 1 and level 2 methods. This allows for the exclusion of this parameter from the testing within the system. This means that this parameter can be kept constant while still allowing tests to provide valuable data on the performance of the different Schwartz methods. This test was also run for a less refined domain, and the trend of the optimal overlap remained similar for the methods regardless of domain size change. This result justifies the choice of keeping the overlap constant for the rest of the numerical test since it will not hinder the consistency of the results.

Then, the experiments that vary the mesh refinement and the number of subdomains, found in figure 4.3, demonstrate clear differences in the convergence behaviour between level one and level two Schwarz preconditioners. With the roughest mesh of 200×200, there is a significant difference in the iteration counts when reaching higher numbers of subdomains, with the trend for the level one method staying positive. In contrast, the level 2 method has fully plateaued this trend. After refining the mesh to 400×400, it is observed that the number of iterations increases for both methods, but the increase for the level 1 method grows more than that of the level 2 mesh. Furthermore, the slope towards the end of the graph for the level 1 method becomes steeper than for the level 2 method, which successfully plateaued again. Then, when looking at the most refined of 800×800, the trajectory observed with the previous refinement continues further. Both the differences in the iteration count and the slope of the figures are growing once more. Once more, it looks like the level 2 method will plateau earlier than it had done with the less refined meshes, further amplifying the differences in the scalability of the two methods.

This outcome aligns with the theoretical advantages of a level 2 method, which supplements the additive methodology of the first level with the global information propagation of a coarse grid correction. Thus mitigating the deterioration of the convergence associated with the growth in the number of subdomains.

5.2. Ocean Model

Following these experiments, the ocean model was implemented with the Schwarz level 1 preconditioner. As the model slowly increased the strength of the forcing term, the number of GMRES iterations per Newton step also increased. In Table 4.1, the sudden jump from 285 to 494 GMRES iterations between the forcing steps 14 and 15 coincides with a transition of relatively mild forcing conditions to more substantive circulations. This behaviour is consistent with the known behaviour of thermohaline circulation models discussed in Thies et al.(2009)[12], with abrupt changes in stable states emerging with stronger influences of boundary conditions. This is because the forcing terms contribute to a significantly ill-conditioned Jacobian within the system. The reason for this can be found by looking back at the Jacobian:

$$\mathbf{J} = \begin{pmatrix} \mathbf{A}_{uv} & \mathbf{B}_1 & \mathbf{G}_{uv} & 0 \\ 0 & 0 & \mathbf{G}_w & \mathbf{B}_2 \\ \mathbf{D}_{uv} & \mathbf{D}_w & 0 & 0 \\ \mathbf{B}_3 & \mathbf{B}_4 & 0 & \mathbf{A}_{TS} \end{pmatrix}$$

The $\mathbf{B}_2$, $\mathbf{B}_3$, and $\mathbf{B}_4$ represent the coupling of the Salinity and Temperature with the momentum and pressure of the system. Without the salinity and temperature forcing present, all of these blocks are practically absent from consideration when computing steady-state solutions to the system. Not only does this decrease the practical dimensionality of the system, but it also decreases the difficulty of the system. This is because $\mathbf{B}$ blocks are positioned in a manner that makes the system more ill-conditioned, generally due to the further off-diagonal placement of the block within the matrix, which makes the convergence of the solver at each Newton iteration more complex and slower.

To test if this theoretical limitation is to blame for the difficulty of the convergence of the GMRES iterations within the system, the tracer(salinity and temperature) forcing was disabled. Firstly, both the salinity and the temperature forcing were fully disabled within the Idealised model, and the resulting system only relied on the effects of the wind forcing to produce action within the model. This simplified system is almost linear in nature and has a significantly less volatile steady state than before. Thus, the system would stay in a steady state even as the general forcing term increased. This results in the model bypassing the Newton steps entirely when the steady state of the previous solution remains the same, as seen in Table 4.2.

To further diagnose the exact nature of the traces' influence, the system was running with each of the tracers disabled individually. First, disabling only the salinity forcing keeps the ill-conditioned nature of the matrix. However, the system appears more ill-conditioned when compared to the original run with full forcing due to the higher number of iterations at the same level of forcing. The system now has problems with the GMRES convergence earlier along the forcing term's size, as seen in Table 4.3. The implications are that the system's salinity contributes positively to the convergence behaviour of the GMRES iterations while the temperature contributes negatively. Still, it is not clear if this positive effect is the only effect the salinity has, and if this positive effect remains for higher forcing values is also unclear.

Table 4.4 showcases the system's behaviour if the temperature forcing is entirely removed. Here, it can be observed that the system moves more in alignment with the original forcing than when it is without the salinity. However, convergence speed is still lost at a higher value for the forcing term. This makes it clear that the system's salinity forcing contributed positively to the convergence when the forcing was low enough, but once it reached a high enough level, it started converging slower, and the effect of the temperature on the iterations took over.

It is essential to note that as mentioned in Chapter 2.1, the ocean model used here has an additional factor placed within the salinity and temperature forcing. Namely, because the system does not include the Boussinesq approximation, it needs a way to compensate for the influence of the buoyancy in the model. This was done by including some compensating factors that interplay with the salinity and temperature of the system. And the experiments entirely blocked the forcing of the salinity and/or temperature on the system. Thus, these compensations don't act on the system either and produce strange results, as seen in Figure A.2. The problematic convergence behaviour can't be entirely attributed to the influence of salinity and temperature, and this buoyancy compensation factor needs to be included.

To combat the significant increases in the convergent methods of the salinity temperature and buoyancy forcing, a more powerful and expansive method should be employed; there is a method of doing so, namely the coarse grid correction. The coarse grid correction would take the level one preconditioner currently used for the system and expand it with a coarse correction. As discussed in chapter 3 this method has a strictly better convergence in terms of GMRES iterations than the level one method. Because the number of iterations is the main contributor to the time, it can significantly accelerate the system in finding a steady-state solution. Furthermore, as seen within the Laplace equations, the scalability of this two-level method is better, meaning that it can also be run on a more refined grid with a large number of processors much more efficiently. Thus, it not only increases the level of accuracy and speed of the solver but also allows for more precise numerical solutions. However this was not directly tested within the system and thus the effects of the

6

Conclusion

This paper explored applying overlapping Schwarz domain composition preconditioner methods for solving large implicit ocean models.

Utilizing a series of controlled experiments on a Laplace system, the theoretical convergence properties of both level one and level 2 Schwarz methods were confirmed. The experiments demonstrated how increased overlap reduces iteration counts and that coarse grid corrections can significantly improve scalability concerning both mesh refinement and the number of subdomains. Thus illustrating how these factors influence the performance of Schwarz preconditioner methods.

Then, the application of the one-level Schwarz preconditioner was extended to a fully implicit three-dimensional ocean circulation model. While the level one Schwarz preconditioning had good computational performance in simpler setups. The limitations of this preconditioning became clear as the system got more complex under strong thermal and salinity forcing. The numerical experiments showcased how the tracer-driven effects substantially worsened the Jacobian's conditioning, significantly increasing the number of GMRES iterations needed for convergence. Thus, the comparative tests between the temperature and salinity forcing, which were disabled individually, indicate that the temperature tracers are the main drivers for the lack of convergence in the model. Still, due to the inclusion of buoyancy within this convective part of the system, it is not possible to conclude this is solely due to the influence of temperature or is caused mainly by buoyancy.

Thus, to answer how the overlapping Schwarz domain decomposition preconditioners affect the efficiency and scalability of the ocean model? The experiments of the Laplace system strongly indicate that adopting a more advanced precondition, which incorporates coarse grid correction into the Schwarz preconditioner, would increase the solver system's efficiency. Such changes would improve scalability for both larger processor counts and finer meshes. However, whether or not this would clear up the ill-conditioning of the model's Laplacian enough to allow for convergence to the full forcing factor wasn't tested and thus could not be concluded.

For further work, this paper emphasizes the vital role of carefully designed preconditioners in attaining efficient and scalable solutions for implicit ocean circulation models. It also advises future research on Schwarz preconditioners to focus on implementing coarse spaces to improve the solver performance of ocean circulation models.

References

[1] Delft High Performance Computing Centre (DHPC). *Delftblue Hardware*. https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2. 2024.

[2] Intel Corporation. *Intel® Core™ i7-13650HX Processor (24 MB Cache, up to 4.90 GHz)*. https://www.intel.com/content/www/us/en/products/sku/232101/intel-core-i713650hx-processor-24m-cache-up-to-4-90-ghz/specifications.html. Accessed June 16, 2025. 2023.

[3] Arie de Niet et al. "A tailored solver for bifurcation analysis of ocean-climate models". In: *Journal of Computational Physics* 227.1 (2007), pp. 654–679. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2007.08.006. URL: https://www.sciencedirect.com/science/article/pii/S0021999107003518.

[4] Alexander Heinlein et al. "Adaptive GDSW Coarse Spaces for Overlapping Schwarz Methods in Three Dimensions". In: *SIAM Journal on Scientific Computing* 41.5 (2019), A3045–A3072. DOI: 10.1137/18M1220613. eprint: https://doi.org/10.1137/18M1220613. URL: https://doi.org/10.1137/18M1220613.

[5] Alexander Heinlein et al. "FROSch: A fast and robust overlapping Schwarz domain decomposition preconditioner based on Xpetra in Trilinos". In: *Domain Decomposition Methods in Science and Engineering XXV*. Ed. by Ronald D. Haynes et al. Springer International Publishing, 2020, pp. 176–184. DOI: 10.1007/978-3-030-56750-7_18. URL: https://doi.org/10.1007/978-3-030-56750-7_18.

[6] Michael A. Heroux et al. "An overview of the Trilinos project". In: *ACM Trans. Math. Softw.* 31.3 (Sept. 2005), pp. 397–423. ISSN: 0098-3500. DOI: 10.1145/1089014.1089021. URL: https://doi.org/10.1145/1089014.1089021.

[7] James W. Hurrell et al. "The Community Earth System Model: A Framework for Collaborative Research". In: *Bulletin of the American Meteorological Society* 94.9 (2013). Provides an overview of CESM's coupled components, design, and capabilities, pp. 1339–1360. DOI: 10.1175/BAMS-D-12-00121.1.

[8] H. B. Keller. "Numerical solution of bifurcation and nonlinear eigenvalue problems". In: *Applications of Bifurcation Theory*. Ed. by P. H. Rabinowitz. New York: Academic Press, 1977, pp. 359–384.

[9] Alexander F. Shchepetkin and James C. McWilliams. "The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model". In: *Ocean Modelling* 9.4 (2005), pp. 347–404. ISSN: 1463-5003. DOI: https://doi.org/10.1016/j.ocemod.2004.08.002. URL: https://www.sciencedirect.com/science/article/pii/S1463500304000484.

[10] Richard D. Smith and Penny Gent. *Reference Manual for the Parallel Ocean Program (POP)*. Los Alamos Unclassified Report LA-UR-02-2484. Approved for public release; distribution is unlimited. Los Alamos National Laboratory, 2002.

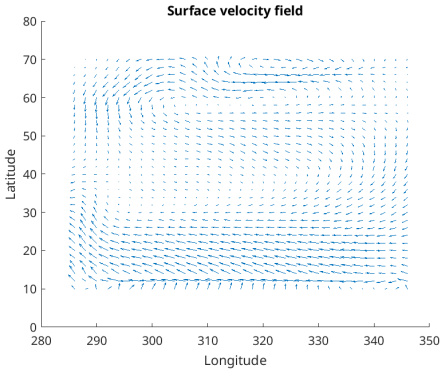
[11] J. Thies. *i-emic: Interactive Earth system Model of Intermediate Complexity (ocean-frosch branch)*. https://github.com/jthies/i-emic/tree/ocean-frosch. Accessed: 2025-06-19. 2019.

[12] Jonas Thies, Fred Wubs, and Henk A. Dijkstra. “Bifurcation analysis of 3D ocean flows using a parallel fully-implicit ocean model”. In: *Ocean Modelling* 30.4 (2009), pp. 287–297. ISSN: 1463-5003. DOI: https://doi.org/10.1016/j.ocemod.2009.07.005. URL: https://www.sciencedirect.com/science/article/pii/S1463500309001553.

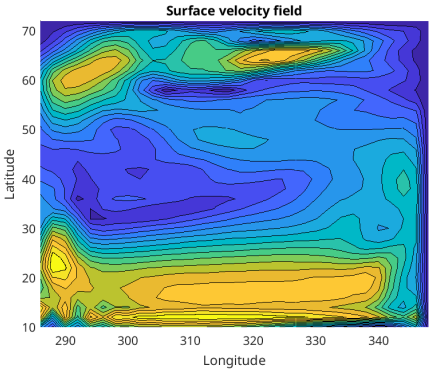
A

additional figures, notation and preliminaries

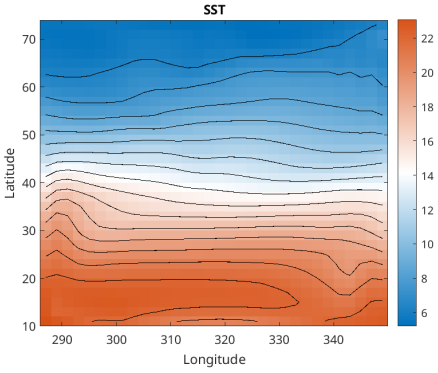
A.1. Figures



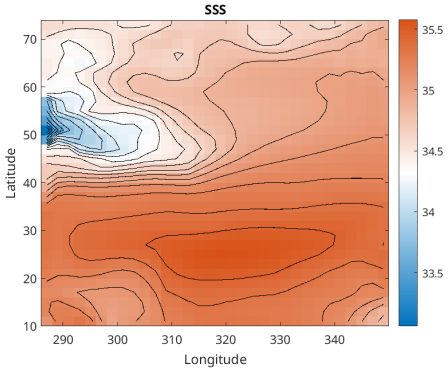
(a) surface velocity field with no salinity forcing



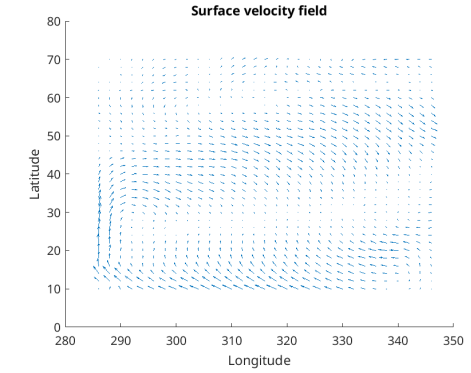
(b) surface velocity field with no salinity forcing



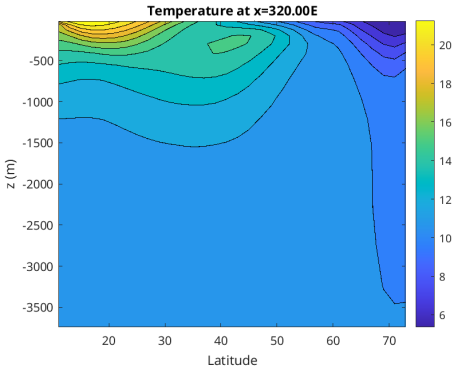
(c) sea surface temperature with normal forcing



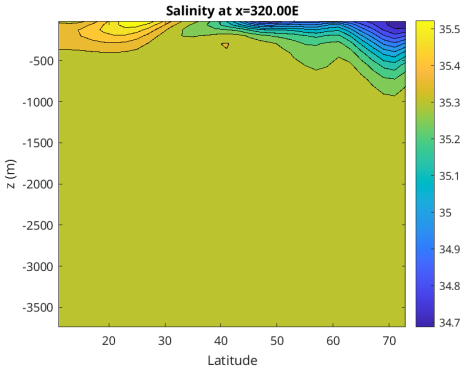
(d) sea surface salinity with normal forcing



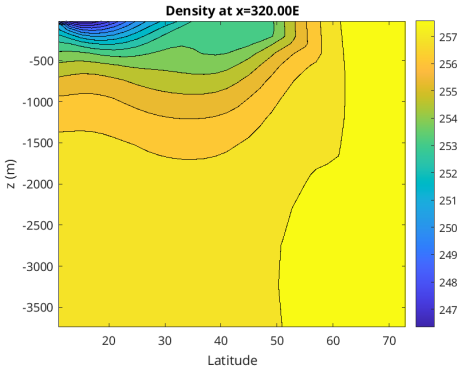
(a) surface velocity field with normal forcing



(b) vertical sections of temperature with normal forcing



(c) vertical sections of salinity with normal forcing



(d) vertical sections of density with normal forcing

A.2. Notation and preliminaries

This par of the appendix gives an overview of terminology and symbols utilised within this paper. followed by a description of the software infrastructure and libraries that were used to preform the numerical experiments.

1. Notation

- u Zonal velocity or movement along latitudinal lines
- v Meridional velocity or movement along longitudinal lines
- w Vertical velocity
- p Pressure perturbation from hydrostatic equilibrium.
- T Temperature field
- S Salinity field
- ρ Seawater density (diagnosed from T and S).
- ν, κ_T, κ_S Viscosity and diffusivity tensors.
- M Mass matrix
- L linear operator
- N non-linear operator

- A Global Jacobian matrix.
- A_i Local subdomain matrix.
- A_0 Coarse matrix in two-level Schwarz method.
- R_i, R_0 Restriction and prolongation operators for Schwarz preconditioning.
- τ_T, τ_S Surface restoring times for temperature and salinity.
- Δs: Arc-length step size in pseudo-arclength continuation.
- μ: Bifurcation continuation parameter (e.g., surface forcing).

2. Software and Libraries

i-emic. This paper works with an ocean branch of the i-emic repository. here i-emic serves as the core solver driver, originally incorporating direct LU-based kernels for subdomain solves. By integrating FROSch preconditioners into i-emic the aim is to evaluate how well the domain decomposition approach can preform in terms of runtime and memory usage while preserving the accuracy.

Belos. Belos is a package within Trilinos for Krylov subspace solvers (e.g. GMRES). Used for solving linearized systems in Newton–Krylov iterations.

Trilinos. The used solver is based on solver infrastructure from the Trilinos[6] framework, using in particular the Amesos2, Belos, and FROSch packages. Trilinos provides a flexible parallel enabled environment for the assembly and solution of large-scale linear systems. In this paper the FROSch package is used as a preconditioner within Belos Krylov solvers, with coarse space corrections managed by Amesos2.

FROSch. The FROSch (Fast and Robust Overlapping Schwarz) package is the main package within Trilinos that will be tested in this paper . It provides overlapping Schwarz preconditioners customized for unstructured grids. In this ocean model tests FROSch is configured to partition the global mesh efficiently.

Xpetra. Xpetra is a linear algebra layer in Trilinos that give a united interface to multiple matrix and vector libraries. It allows for code portability and flexibility by allowing a switch in back-ends within the code without requiring changes in the high-level code.

Amesos2 Amesos2 A Trilinos package for sparse direct solvers. Used here to solve local subdomain problems and coarse systems in the overlapping Schwarz preconditioner.

3. Computational devices

The numerical experiments were executed on 2 separate computing systems

- Home laptop uses a 13th Gen Intel(R) Core(TM) i7-13650HX[2] CPU for calculations. When possible the laptop is run using only performance cores for best comparable results.
- Delftblue supercomputer[1] a high-performance computing facility at TU Delft. Utilised when looking at higher numbers of subdomains.

B

Source code

The full source code for this ocean model and other programs run to get numerical results is available on github. [1]

[1] https://github.com/DaGr1/i-emic-DG/tree/final_Branch