A high-resolution finite difference method for weather and climate models

Magnus Ulimoen
Abstract

**A high-resolution finite difference method for weather and climate models**

*Magnus Ulimoen*

Weather and climate modelling has been one of the major actors in scientific computing. The need for higher resolution for the current models has revealed problems, some of which can be solved by using the SBP-SAT method. Ground effects in the atmosphere, caused by e.g. trees, buildings or mountains may require the discrete grid to be adapted to fit the atmosphere. The work presented here combines grids with different resolution in the vertical direction to solve a simplified model of the atmosphere. The results suggest making minor adjustments to parts of the grid in conjunction with the grid adaptions creates an efficient solver for the heterogeneous atmosphere.
Contents

Abstract iii

1 Introduction 1
   1.1 Aim of the Thesis ........................................ 1
   1.2 Climate and weather models ............................. 1
   1.3 Collaboration with SMHI ................................. 3
   1.4 Difficulties with current models ....................... 3

2 Fluid Dynamics 4
   2.1 Macroscopic view ........................................ 4
   2.2 Conserved quantities .................................... 4
   2.3 Navier-Stokes equations ............................... 4
   2.4 The Euler Equations .. ................................. 5
      2.4.1 Non-Conservative formulation ..................... 5
      2.4.2 Conservative formulation ........................ 5
      2.4.3 An analytical solution ........................... 6
   2.5 Model Atmosphere ....................................... 6
      2.5.1 Equation of state ............................... 7

3 Partial Differential Equations 8
   3.1 Partial Differential Equation ......................... 8
   3.2 IBVP .................................................. 8
   3.3 Consistency ............................................ 9
   3.4 Stability of PDE ...................................... 9
   3.5 Existence .............................................. 10
   3.6 Convergence ........................................... 10

4 SBP-SAT 12
   4.1 The Energy Method ...................................... 12
      4.1.1 Constant Coefficient Problem .................... 13
   4.2 Definition of SBP ..................................... 13
   4.3 SAT ..................................................... 14
   4.4 Discrete Constant Coefficient Problem ............... 14
   4.5 Upwind operators .................................... 15
   4.6 Higher dimensions ................................... 16
   4.7 SBP-SAT of Euler’s equations ......................... 16
   4.8 Multi-block ............................................ 17
   4.9 Multi-block conforming meshes ....................... 18
   4.10 Multi-block non-conforming Meshes ................. 18
   4.11 Shifting grid points ................................ 20
5 Results and Discussion

5.1 Implementation and Replication .................................. 21
5.2 Convergence .......................................................... 21
5.3 Stability ...................................................................... 22
5.4 Efficiency ................................................................. 26

6 Conclusions ..................................................................... 32

Bibliography ....................................................................... 33

List of Figures ..................................................................... 36

List of Tables ....................................................................... 36

Appendix A Curvilinear Mapping ............................................. 37
   A.1 Reformulating to Computational Space ......................... 37
   A.2 Conservative treatment ............................................. 38
   A.3 Derivation of Conservative form ................................. 38
1 Introduction

1.1 Aim of the Thesis

The thesis aims to explore the use of new methods to improve on grid adaptations with finite differences in climate and weather simulation applications. The intention is to find an accurate and stable extension to the Summation-By-Parts Simultaneous Approximation Term (SBP-SAT)\(^*\) method which allows coarser resolution across grid interfaces, thereby decreasing computational cost.

In order to solve a numerical atmospheric model efficiently, it may be necessary to distribute the points according to the underlying physical model. The atmospheric column might have stronger currents and higher frequencies due to ground effects, that may need a higher resolved grid than the upper atmosphere. By using grid adaptations, the difference in frequencies of the different areas may be modelled accurately and more efficiently.

1.2 Climate and weather models

Climate models have been used for more than half a century, with their complexity and accuracy increasing over time. Simple models [15] limited themselves to radiative convective equilibrium in the atmosphere. Later models [16] describe general circulation models involving interactions with the ocean and land surface. More fully coupled models including biogeochemical cycles and dynamic vegetation [5] were developed from the late 90s.

Numerical weather prediction (NWP) has been using the same model components as climate research to describe the atmosphere and the interactions with the surface. The scale of the climate and NWP models determines which resolution one can realistically model an area. A model covering the earth will henceforth be known as GCM (global climate model). Regional climate models (RCM), allows for more localised information. [28] shows modern RCM modelling mesoscale\(^{1}\) phenomena.

The numerical climate models provide a tool to assess future climate and predict human induced changes to the atmosphere and land surface [30]. Accurately predicting climate change allows fundamental decisions to be made regarding climate mitigation, adaption and policy making. A more accurate model will allow decisions to be made with a higher confidence.

\(^*\)See section 4 for a complete description of the method
\(^{1}\)10 to 1000 km in horizontal extent
The main equation for atmospheric modelling is the Navier-Stokes system of partial differential equations (NS). The NS equations show many interesting phenomena including shocks, turbulence, diffusion, and transport. This results in difficulties in modelling, and much research has been directed towards finding effective ways to capture all of these phenomena. In atmospheric modelling some extra terms are added to NS, describing radiative gains and losses, and humidity content giving clouds and rain.

The global and regional scale of weather and climate forecasting requires highly effective models, as the degrees of freedom (DOF) will soon reach 60 billion*. To ensure the best utilisation of available computing power, it is vital to minimise the accumulated error (accuracy), be efficient with low communication between computing nodes, and allow a distribution of grid points matching the physics (adaptive). This permits the meteorologist and climatologist to focus on the results of the model.

The following example shows the importance of higher order methods. Figure 1 shows the convergence for two different methods, with the DOF plotted against the total error for two different order methods. If one wants a maximum error that is three magnitudes less (e.g. from $10^{-3}$ to $10^{-6}$), the higher order method requires 8 times fewer DOF than the lower order method from break-even. As the computational power necessary scales linearly with the DOF†, the higher order method will become more efficient than a lower order method as DOF increases.

![Figure 1: Example of two methods with different convergence order, and how the errors evolve with the degrees of freedom](image)

---

*Grid resolution of 10km×10km with 120 levels vertically, covering the globe, with 10 variables per grid point

†As is the case for finite differences, as shown in a later section
1.3 Collaboration with SMHI

Part of this project has been conducted in collaboration with the Rossby Centre at the Swedish Meteorological and Hydrological Institute (SMHI). They are currently experimenting using SBP methods in ESSENSE, a three dimensional Navier-Stokes solver for atmospheric modelling. ESSENSE sets boundary conditions based on the flow situations described in [25]. Some previous work can be found in [13]. The aim for ESSENSE is to be able to simulate weather and climate in large volumes over long timescales. Some of the work presented here has been developed and tested using this program.

1.4 Difficulties with current models

The Rossby Centre at SMHI has developed a RCM (RCA4), which has produced climate scenarios with a horizontal resolution of 50 km for Europe [11] and Africa [23]. Higher resolution (12.5 km) studies has also been conducted [9]. The attempt at increasing the horizontal resolution has focused efforts to the non-hydrostatic NWP system HARMONIE, which is based on the AROME model [29]. HARMONIE has been applied to climate [12], [14].

HARMONIE bases itself on a two-dimensional horizontal Fourier decomposition [26] using periodisation windows and Davies relaxation to allow non-periodic fields [3]. The vertical component is discretised with either finite elements or finite differences. RCA4 uses a finite difference scheme. Both methods import horizontal boundary data using the Källberg-Davies method [4], [10].

The dynamical schemes of both method renders them at most second order accurate, giving them poor resolving power. The inherent periodicity requirement of Fourier schemes is also problematic for accuracy reasons. The use of semi-Lagrangian schemes does not conserve the physical quantities. [7] reports a 35% loss of mass on a 48 hour simulation, and [26] reports a lack of conservation of angular momentum.

Both HARMONIE and RCA4 rely on fields that must be reconstructed for every iteration, and a Fourier decomposition. The reconstruction and Fourier decomposition requires heavy inter-process communication, which is a major bottleneck in communication with respect High Performance Computing (HPC). This reconstruction can be avoided using finite differences.
2 Fluid Dynamics

The field of fluid dynamics is vast, and encompasses far more than just atmospheric modelling. This section introduces the physics behind the fluid model, and the simplified atmospheric model that is used in this paper.

2.1 Macroscopic view

From a microscopic view, a fluid consists of individual molecules with velocities, positions and energies. As a fluid consists of a very large number of molecules, it is prohibitively expensive to compute the trajectories of all of them*. If we work on a scale where the characteristic length is far larger than the mean free path, the fluid can instead be described by continuous microscopic fields, such as pressure, density, velocity and energy.

2.2 Conserved quantities

The flow of a fluid is described by a set of conservation laws. The conserved quantities in an ordinary fluid is mass, momentum, energy and entropy. Fluids may also contain other quantities, such as liquid phases, charges, or chemical species. These are not covered here.

The fields are modelled as continuous, and the conserved quantities of such fields are described by [34],

- Conservation of mass — Mass of a material volume is constant
- Newton’s second law — The material volume changes momentum based on the sum of surface forces
- Conservation of energy — Energy is only gained or lost through forces working on the material volume, and heat transfer
- Increase of entropy — Entropy rate is given by the material entropy and entropy flow through surface of the material

2.3 Navier-Stokes equations

Combining the conservation laws, see for example [17], yields the Navier-Stokes system of equations. In two dimensions this takes the form:

\[
\begin{align*}
    u_t = (Au)_x + (Bu)_y + \left( \frac{\partial}{\partial x} \right)^T \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix}.
\end{align*}
\] (2.1)

*There are around \(2.6 \cdot 10^{25}\) molecules in a cubic meter of air
Au and Bu are known as the fluxes, and the matrix $C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$ is the viscosity matrix. Stability and uniqueness of NS is still an open problem [8], but parabolicity requires that $C + C^T$ is positive semi-definite.

2.4 The Euler Equations

Setting $C = 0$ in eq. (2.1) gives the Euler equations. This equation is a useful first approximation of the Navier-Stokes equations for modelling purposes, as it is a hyperbolic non-linear PDE, with some known solutions. This gives a good benchmark for testing numerical approximations to the Navier-Stokes equation, as they will capture shocks, vorticity and transport. If one reintroduces $C$, it can be shown [19] that it leads to a loss of energy, and can serve to stabilise the equations.

2.4.1 Non-Conservative formulation

The primitive formulation of the Euler equations is given by [27]

$$\partial_t \hat{Q} + M \partial_x \hat{Q} + N \partial_y \hat{Q} = 0, \quad \text{(2.2)}$$

with

$$\hat{Q} = \begin{bmatrix} \rho \\ u \\ v \\ p \end{bmatrix}, \quad M = \begin{bmatrix} u & \rho & 0 & 0 \\ 0 & u & 0 & \rho^{-1} \\ 0 & 0 & u & 0 \\ \gamma p & 0 & 0 & u \end{bmatrix}, \quad N = \begin{bmatrix} v & 0 & \rho & 0 \\ 0 & v & 0 & 0 \\ 0 & 0 & v & \rho^{-1} \\ 0 & 0 & \gamma p & v \end{bmatrix}.$$  \text{(2.3)}$$

$u, v$ is the velocity in $x, y$ direction, $\rho$ is the density of the fluid, $p$ is the pressure, and $\gamma$ is the ratio of heat capacities.

2.4.2 Conservative formulation

In order to preserve shocks, it may be necessary to preserve the momentum density during all computational steps. Using the non-conservative form may give non-linear discretisation errors that may not preserve shocks, or may yield incorrect shock-speeds. The conservative formulation is given by [27]

$$\partial_t Q + \partial_x E + \partial_y F = 0, \quad \text{(2.4)}$$
where
\[
Q = \begin{bmatrix}
\rho \\
\rho u \\
\rho v \\
e
\end{bmatrix}, \quad E = \begin{bmatrix}
\rho u \\
\rho u^2 + p \\
\rho uv \\
ue(e + p)
\end{bmatrix}, \quad F = \begin{bmatrix}
\rho v \\
\rho v^2 + p \\
\rho uv \\
v(e + p)
\end{bmatrix}.
\] (2.5)

\(e\) is the energy/temperature. To relate between pressure and temperature, one must have an equation of state.

### 2.4.3 An analytical solution

One can construct an analytical solution of the Euler equations by perturbing a steady-state field with a Gaussian wave. The form of the analytical solution is taken from [20], with a vortex travelling left to right,

\[
\begin{align*}
\frac{\epsilon y}{2\pi \sqrt{p_{\infty} r_z^2}} \exp\left(\frac{f(x, y, t)}{2}\right) & \quad (2.6a) \\
\frac{\epsilon ((x - x_0) - t)}{2\pi \sqrt{p_{\infty} r_z^2}} \exp\left(\frac{f(x, y, t)}{2}\right) & \quad (2.6b) \\
\left(1 - \frac{\epsilon^2 (y - 1) M^2}{8\pi^2 p_{\infty} r_z^2} \exp(f(x, y, t))\right)^\frac{1}{\gamma - 1} & \quad (2.6c) \\
p = p_{\infty}\rho^\gamma & \quad (2.6d) \\
f = \frac{1 - ((x - x_0) - t)^2 + y^2}{r_z^2} & \quad (2.6e)
\end{align*}
\]

The parameters \(p_{\infty} = \frac{1}{\gamma M^2}\) is the pressure far away from the vortex, \(r_z\) is the radius of the vortex, \(t\) is the time the vortex has travelled, \(x_0, y_0\) gives the centre of the vortex at \(t = 0\), \(\epsilon\) the circulation of the vortex, and \(M\) is the Mach number.

### 2.5 Model Atmosphere

In the numerical experiments later in this paper, the atmosphere will be modelled by the Euler equations. As the Euler equations are hyperbolic, they might be harder to solve than the parabolic NS. A stable treatment of the Euler equations will transfer to a stable treatment of NS. The model developed here is extendable to NS by inserting the viscous terms and associated boundary conditions [25], [31], [32].
2.5.1 Equation of state

The equation of state relates the different properties of the fluid to the pressure. For the atmosphere, it is sufficient to deal with the ideal gas equation:

\[ p = (\gamma - 1) \left( e - \frac{1}{2} \rho (u^2 + v^2) \right), \quad (2.7) \]

where \( \gamma = 1.4 \) is the ratio of heat capacities for the atmosphere. The speed of sound for the ideal gas is given by

\[ c^2 = \gamma p/\rho. \quad (2.8) \]
3 Partial Differential Equations

This section will give the definitions used to solve the mathematical formulation of the problem, and the terminology used for the numerical approximation.

3.1 Partial Differential Equation

A Partial Differential Equation (PDE) is an equation involving multiple derivatives in space and time. A simple example of a PDE is given by

\[ u_t = -u_x. \]  

This is a transport equation, as the solution is satisfied by a function \( u = f(x-t) \), which moves the solution along \( x \). Another PDE is

\[ u_t = \nabla \cdot D \nabla u, \]

where \( D \) is the diffusion coefficients. This is known as the diffusion equation, and tends to smooth a solution over time.

3.2 IBVP

The Initial Boundary Value Problem (IBVP) of interest here can be formulated:

\[ u_t + H \left( x, t, \frac{\partial}{\partial x_i} \right) u = F(x, t), \quad x \in \Omega, \quad t \in (t_0, T] \]  

\[ u(t) = f(x), \quad x \in \Omega, \quad t = t_0 \]  

\[ Lu = g(t), \quad x \in \Gamma, \quad t \in (t_0, T] \]

where \( u \) is a function of time and space. \( H \) is a (non-linear) given functional. \( F \) is a given source-term, \( L \) a boundary operator, \( g \) boundary data, \( \Omega \) the domain, \( \Gamma \) the edge of the domain, \( t \) the time, \( x \) the spatial location in the domain.

Most of these equations can not be solved analytically, and so the focus has been on solving a subset of these problems numerically, to allow for the widest class of solutions. The following sections will detail what the numerical approximation of the IBVP must guarantee to be correct.
3.3 Consistency

A consistent formulation of a PDE has an error that decreases as the resolution increases,

$$\lim_{\Delta x, t\to 0} (v_t + Hv) = u_t + Hu, \quad (3.4)$$

where \( v \) is the discrete solution, and \( u \) the continuous solution. A common approach to ensure consistency is by Taylor expansion of the PDE in time and space. If a discrete PDE and a continuous PDE are not convergent, they do not describe the same system.

3.4 Stability of PDE

The stability determines whether the solution to an equation behaves in such a way that it is feasible to solve it. The solution of a stable PDE, will not "explode" in a singularity, but has a controlled growth in energy. The energy of a PDE is not a general quantity, but represents an abstract property. Section 4.1 gives the energy for a type of IBVP.

In [24] the following three definitions for the stability of a PDE are given:

**Definition 3.1.** A PDE (such as eq. (3.3)) is strongly well-posed if a unique solution exists, and the estimate

$$\|u\|_{\Omega}^2 + \int_0^t \|u\|_1^2 \, dt \leq K_c e^{\eta_c t} \left( \|f\|_\Omega^2 + \int_0^t \left( \|F\|_\Omega^2 + \|g\|_\Gamma^2 \right) \, d\tau \right)$$

holds. \( K_c \) and \( \eta_c \) does not depend on \( F, f, \) or \( g \) (data). The norms must be some suitable norms.

This definition limits the solution to not grow faster than an exponential, and it is bound by the boundary data, source terms and initial conditions.

**Example.** The stability criteria in definition 3.1, can be shown to be fulfilled by the Euler equations by integrating eq. (2.4) in the domain \( \Omega \),

$$\frac{d}{dt} \int_\Omega Q \, dV + \int_{\gamma} (E \, dx - F \, dy) = 0. \quad (3.5)$$

The conservation laws of the Euler equations gives a bound based on the influx from the boundaries, as long as the boundaries conditions are correctly set.

The next definition defines stability for a discrete solution.

**Definition 3.2.** The discrete approximation of eq. (3.3) is said to be strongly stable if, for a sufficiently small \( \Delta x \), there is an unique solution that satisfies

$$\|v\|_{\Omega}^2 + \int_0^t \|v\|_1^2 \, d\tau \leq K_d e^{\eta_d} \left( \|f\|_{\Omega}^{h_2} + \int_0^t \left( \|F\|_{\Omega}^{h_2} + \|g\|_{\Gamma}^{h_2} \right) \, d\tau \right).$$
$K_d$ and $\eta_d$ does not depend on data, and the norms are suitable discrete norms.

It is also useful to have an estimate of stability from the continuous case to the discrete case,

**Definition 3.3.** We call the discrete solution of eq. (3.3) strictly stable if the growth rate in definition 3.1 and definition 3.2 satisfy

$$\eta_d \leq \eta_c + O(\Delta x).$$

This ensures the growth of the discrete solution will not be faster than for the continuous solution as the resolution increases.

### 3.5 Existence

Existence of a solution is dependent on the boundary conditions. Over-specifying the boundary conditions with e.g the Källberg-Davies method, will destroy the existence of a solution. Under-specifying the boundary conditions allows for multiple solutions, and will not bound the integrals in definitions 3.1 and 3.2, therefore being unstable. The boundary conditions must be set to the correct number and type for the PDE in question, both for the continuous case and the discrete case.

### 3.6 Convergence

A convergent PDE discretisation has the property

$$\lim_{\Delta x, t \to 0} v(x, t) = u(x, t).$$  \hspace{1cm} (3.6)

An increase in mesh resolution will give a discrete solution that converges to the continuous solution. The relation between consistency, stability and convergence is given by the Lax-Richtmyer Theorem [6]:

**Theorem 3.4.** Given a well-posed linear initial value problem, a consistent stable finite difference approximation are sufficient conditions for the approximation to be convergent.

In most cases it is possible to measure convergence numerically. The existence of an analytical solution may be used to measure the convergence. Testing whether the numerical method is convergent to the design order of accuracy is a good way to ensure correctness. The convergence rate is given by

$$q = \log_{10} \left( \frac{\| u - v^{m_0} \|_h}{\| u - v^{m_1} \|_h} \right) / \log_{10} \left( \frac{m_0}{m_1} \right)^{1/d}$$  \hspace{1cm} (3.7)
Where $v^{m_i}$ is the solution with $m_i$ unknowns, $u$ is the exact solution, and $d$ is the dimensionality of the domain, which in this paper is two. The error norm is here given by the discrete $H$ norm,

$$\|v\|_h = \sqrt{v^T H v}$$

(3.8)
4 SBP-SAT

The Summation-By-Parts Simultaneous Approximation Terms (SBP-SAT) method is a special kind of finite difference method with stronger emphasis on boundary treatments, which allows strong stability criteria to be fulfilled for linearised problems. The method mimics integration by parts for a continuous PDE. A review of the method can be found in [33].

The stability, accuracy and conservative properties of the SBP-SAT will be demonstrated. The analysis will focus on the semi-discrete problem, leaving time continuous. This is to focus on the difficulties of spatial discretisation. For integration in time, explicit time schemes provides a simple and stable procedure.

4.1 The Energy Method

The energy of a PDE is a conserved quantity if there is no influx from boundaries. This is can for some PDE be expressed by an inner product, such as the inner product for a continuous solution:

**Definition 4.1.** An inner product between two vectors is given by the $L_2$-norm:

$$(u, v) = \int u^* v \, dx,$$

where the integral is taken in the domain $\Omega$, and $u^*$ is the complex conjugate of $u$. The norm is defined by

$$\|u\|^2 = (u, u).$$

An analogous inner product for the discrete solution can be defined:

**Definition 4.2.** The discrete inner product between two vectors is given by

$$(u, v)_H = u^T Hv.$$

The discrete norm is given by

$$\|u\|^2_H = u^T Hu,$$

where $H$ is a positive definite matrix.
4.1.1 Constant Coefficient Problem

Equation (3.3) takes a simple form for a constant coefficient problem, where \( H = A \frac{\partial}{\partial x} \), and \( A \) a constant coefficient matrix:

\[
\begin{align*}
  u_t &= (Au)_x, \\
  A^- (u_t - g_l) &= 0, \\
  A^+ (u_r - g_r) &= 0, \\
  u(t = 0) &= f.
\end{align*}
\]

\( Au \) is the flux, with \( A^+ u \) and \( A^- u \) the positive and negative fluxes respectively. This can be eg. Maxwell’s equations or the linearised Euler equations. These boundary terms are known as the characteristic boundary conditions. The splitting of \( A \) is given by the following definition:

**Definition 4.3.** Let \( A^+ \) be a positive semi-definite matrix, and \( A^- \) be a negative semi-definite matrix. If \( A = A^+ + A^- \), \( A \pm \) are the characteristics of equation eq. (4.1).

Equation (4.1) can be diagonalised with a change of variables,

\[
\begin{align*}
  \hat{u}_t &= \Lambda \hat{u}_x \\
  \Lambda^- (\hat{u}_t - \hat{g}_l) &= 0 \\
  \Lambda^+ (\hat{u}_r - \hat{g}_r) &= 0 \\
  \hat{u}(t = 0) &= \hat{f}.
\end{align*}
\]

Where \( \Lambda = S^{-1} AS \) is diagonal matrix with the eigenvalues of \( A \) on the diagonal, \( \hat{u} = S^{-1} u \). From here on, the hat on the variables will be omitted and the PDE will be assumed to be symmetrised. To calculate solution, the symmetrisation is not required. To obtain a valid energy estimate from this equation, the sum of the inner products \((u, u_t) + (u_t, u)\) can be computed:

\[
\begin{align*}
  \frac{d}{dt} \|u\|^2 &= (u, u_t) + (u_t, u) \\
  &= u_r \Lambda^- u_r - u_l \Lambda_+ u_l + u_r \Lambda_+ g_r - u_l \Lambda_- g_l.
\end{align*}
\]

With zero data \((g = 0)\), this gives an energy loss through the boundaries. Uniqueness of the solution is not shown here, but exists for the constant coefficient problem. Definition 3.1 is then satisfied, and the problem is well posed.

4.2 Definition of SBP

The discrete solution will now be a vector of length \( n \), and the matrices applied to this will be of size \( n \times n \). To work with the boundaries, the
The following definition from [21] gives the SBP property:

**Definition 4.4.** A difference operator $D_1$ is an SBP operator if it approximates $\frac{\partial}{\partial x}$ and is on the form

$$D_1 = H^{-1} \left( Q + \frac{1}{2} B \right),$$

with the matrices satisfying

$$Q + Q^T = 0, \quad B = -e_0 e_0^T + e_n e_n^T, \quad H = H^T > 0.$$

$H$ must be symmetric and positive definite.

Remark: $H$ is a suitable norm for the discrete inner product.

### 4.3 SAT

Strongly enforcing boundary terms may lead to instability of the approximation. As a remedy one can use weak enforcing of the boundary data with SAT. This weak enforcing yields a provably stable method. The form of the SAT terms is determined by the energy estimate.

### 4.4 Discrete Constant Coefficient Problem

An SBP-SAT formulation of eq. (4.1);

$$v_t = A \otimes D_1 v + \tau_1 A_- \otimes H^{-1} e_1 (e_1^T v - g_l) + \tau_r A_+ \otimes H^{-1} e_n (e_n^T v - g_r),$$

where $\otimes$ is the Kronecker product. An energy estimate for this problem can be calculated the inner product $(v, v_t)$, and adding the transpose, here with data set to zero;

$$\frac{d}{dt} \|v\|^2_H = v^T A \otimes (Q + Q^T) v + v_r A_- v_r - v_l A_+ v_l + (1 + 2 \tau_r) v_r A_+ v_r - (1 - 2 \tau_l) v_l A_- v_l$$

(4.6)
where $Q + Q^T = 0$ follows from the SBP property. Equation (4.6) mimics eq. (4.3), with a parameter $\tau$ to be chosen such that we get a bounded energy estimate. Choosing

$$\tau_r \leq -1/2, \quad \tau_l \geq 1/2,$$

produces an energy estimate which satisfies definition 3.2. Increasing $\tau$ to unity magnitude introduces some damping at the boundaries and yields dual-consistency [2]. This shows that SBP-SAT is consistent, stable and thus convergent.

### 4.5 Upwind operators

Upwind operators are a dual pair of SBP operators which operate on the positive and negative fluxes of a PDE. This splitting provides artificial damping, which may be vital for non-linear PDEs. In [20], the following definitions are given for the upwind operators:

**Definition 4.5.** The difference operators $D_{\pm} = H^{-1} \left( Q_{\pm} + \frac{B}{2} \right)$ approximating $\frac{\partial}{\partial x}$, are said to be diagonal-norm upwind SPB operators if the diagonal matrix $H$ defines a discrete norm, $Q_{\pm} + Q_{\pm}^T = 0$ and $\frac{Q_{\pm} + Q_{\pm}^T}{2} = S$ is negative semi-definite.

Using these operators on the positive and negative fluxes of eq. (4.1), treating the boundary terms (BT) as the SAT terms above, gives the following discretisation:

$$u_t = A_+ \otimes D_+ u + A_- \otimes D_- u + BT$$

$$= A \otimes \left( \frac{D_+ + D_-}{2} \right) u + R \otimes \left( \frac{D_+ - D_-}{2} \right) u + BT$$

$$= A \otimes D_1 u + R \otimes H^{-1} Su + BT.$$  

$D_1$ is still a consistent approximation of $\frac{\partial}{\partial x}$. $R$ is a positive definite matrix given by

$$2A_{\pm} = A \pm R,$$

where $R$ can be selected from e.g. Steger-Warming flux splitting.

$$2A_{\pm} = S^{-1} (\lambda \pm |\lambda|) S = A \pm R,$$

where $\lambda$ are the eigenvalues of $A$. The resulting energy estimate will then be

$$\frac{d}{dt} \|v\|_H^2 = BT + v^T R \otimes S v = BT + AD.$$  

15
As $R \otimes S$ is negative semi-definite, this gives a loss of energy for the solution, which dampens the spurious frequencies often associated with non-linear PDEs. Equation (4.8c) is equivalent to the regular SBP-SAT formulation in eq. (4.5), with the addition of artificial damping.

### 4.6 Higher dimensions

Extending the SBP-SAT formulation to more spacial dimensions requires some additional notation, here shown for two dimensions.

$$
H_x = H \otimes I_m, \quad H_y = I_n \otimes H, \quad \bar{H} = I_k \otimes H \otimes H
$$

$$
D_x = D_1 \otimes I_m, \quad D_y = I_n \otimes D_1
$$

$$
e_W = e_1 \otimes I_m, \quad e_E = I_k \otimes e_n \otimes I_m
$$

$$
e_S = I_k \otimes I_n \otimes e_1, \quad e_N = I_k \otimes I_n \otimes e_m
$$

Remark: Operators in different directions does not need to be equal.

The boundary edges are given in fig. 2. More complicated geometries can be mapped to the geometry in fig. 2 by using grid-mapping. This grid-mapping is detailed in appendix A.

### 4.7 SBP-SAT of Euler’s equations

The SBP-SAT formulation of eq. (2.4) takes the form

$$
v_t + I_k \otimes D_x E + I_k \otimes D_y F = BT. \quad (4.12)
$$

Using the flux form $E = Au, \ F = Bu$ of this equation and linearising the coefficient matrix,

$$
v_t + A \otimes D_x v + B \otimes D_y v = BT. \quad (4.13)
$$
It is now possible to flux split $A$ and $B$ into a positive and negative flux. The boundary terms are given by the characteristics, resulting in the following SAT:

$$
\begin{align*}
BT &= \tau_W A_+ \otimes H_x^{-1} e_W (e_W^T v + g_W) + \tau_E A_- \otimes H_x^{-1} e_E (e_E^T v - g_E) \\
&\quad + \tau_S B_+ \otimes H_y^{-1} e_S (e_S^T v - g_S) + \tau_N B_- \otimes H_y^{-1} e_N (e_N^T v - g_N).
\end{align*}
$$

The discrete energy estimate results in

$$
\frac{d}{dt} \|v\|^2_H = v^T (A + 2\tau_W A_+) \otimes e_1 e_1^T \otimes Hv + v^T (-A + 2\tau_E A_-) \otimes e_n e_n^T \otimes Hv \\
+ v^T (B + 2\tau_S B_+) \otimes H \otimes e_1 e_1^T v + v^T (-B + 2\tau_N B_-) \otimes H \otimes e_m e_m^T v.
$$

Linear stability is ensured if the penalty parameters are set to:

$$
\tau_W \leq -\frac{1}{2}, \quad \tau_S \leq -\frac{1}{2}, \quad \tau_E \geq \frac{1}{2}, \quad \tau_N \geq \frac{1}{2}.
$$

It is customary to set these to have a magnitude of unity, as shown in section 4.4. The artificial damping from the upwind operators is trivial to add,

$$
AD = R_A \otimes (H_x^{-1} S_x)v + R_B \otimes (H_y^{-1} S_y)v.
$$

Giving an energy loss which may be necessary for non-linear problems.

### 4.8 Multi-block

A domain may be irregular or involve areas with different physics or boundary conditions. An appropriate way to divide such a domain should be sought, and is provided by dividing the domains in blocks. The SBP-SAT method can easily handle interface conditions between such blocks, and provide a provably stable method for combined domains.

A characterisation of the interfaces between two blocks is given by the following definition.

**Definition 4.6.** Two meshes are conforming if they share all the boundary points along an edge. Otherwise the meshes are non-conforming. The meshes must have the same diagonal norm along the shared edge.
4.9 Multi-block conforming meshes

By placing two grids side-by-side, as in fig. 3, the PDE can be solved simultaneously on both domains. As the meshes are conforming, the domains can be coupled by using equation eq. (4.15), with \( v \) the solution in one domain, and \( w \) the solution in the other domain.

\[
\begin{align*}
v_t + I_k \otimes D_x E(v) + I_k \otimes D_y F(v) &= BT + \tau_E A_- \otimes H_x^{-1} e_E \left( e_E^T v - e_W^T w \right), \\
w_t + I_k \otimes D_x E(w) + I_k \otimes D_y F(w) &= BT + \tau_W A_+ \otimes H_x^{-1} e_W \left( e_W^T w - e_E^T v \right).
\end{align*}
\]

(4.18a) (4.18b)

The interface conditions are weakly imposed by \( v_E = w_W \). \( BT \) are the boundary terms from the three independent edges, with \( \tau \) as in eq. (4.16). The energy estimate follows as usual, but will now include the term

\[
IT = \begin{bmatrix} v_W \\ w_E \end{bmatrix}^T \begin{pmatrix} -A + 2\tau A_+ & -\tau A_+ - \tau A_- \\ -\tau A_+ - \tau A_- & A + 2\tau A_- \end{pmatrix} \otimes H^y \begin{bmatrix} v_W \\ w_E \end{bmatrix}
\]

(4.19)

To have an energy estimate, this product needs to be negative semi-definite. By specifying \( \tau \) as in eq. (4.16), this is ensured.

4.10 Multi-block non-conforming Meshes

To resolve differences in physics between areas, it may be necessary to have different resolution on two neighbouring meshes. This can give non-conforming interfaces, as seen in fig. 4.

The interface conditions will now weakly impose \( v_N = w_S \) by using the interpolation matrices \( I_{F2C} \) and \( I_{C2F} \). These matrices are defined in [18] by:

**Definition 4.7.** Let the row vectors \( \mathbf{x}^k_f \) and \( \mathbf{x}^k_c \) be the projections of the polynomials \( x^k \) onto equidistant one-dimensional (2-D) grids corresponding to a fine and coarse grid, respectively. We say that \( I_{C2F} \) and \( I_{F2C} \) are \( p \)th-order accurate interpolation operators if \( e^k_c \equiv I_{F2C} \mathbf{x}^k_f - \mathbf{x}^k_c \) and \( e^k_f \equiv I_{C2F} \mathbf{x}^k_c - \mathbf{x}^k_f \) vanish for \( k = 0 \ldots p-1 \) in the interior and for \( k = 0 \ldots (p-1)/2 \) at the boundaries.
Using these matrices results in the SAT terms

\[
\begin{align*}
SAT_{I_N} &= \tau_N B_- H_y^{-1} e_N \left( e_N^T v - I_{C2F} w_S \right), \\
SAT_{I_S} &= \tau_S B_+ H_y^{-1} e_S \left( e_S^T w - I_{F2C} v_N \right).
\end{align*}
\]

(4.20a)

(4.20b)

The resulting energy estimate gives

\[
IT = \begin{bmatrix} v_N \\ w_S \end{bmatrix}^T X \begin{bmatrix} v_N \\ w_S \end{bmatrix},
\]

(4.21a)

with

\[
X = \left( -B + 2 \tau_N B_+ \right) \otimes H_y^w \begin{bmatrix} Y \\ Y^T \end{bmatrix} \left( B + 2 \tau_S B_- \right) \otimes H_y^w \begin{bmatrix} Y \\ Y^T \end{bmatrix},
\]

(4.21b)

and

\[
Y = -\tau_N B_- \otimes H_y^w I_{C2F} - \tau_S B_+ \otimes I_{F2C}^T H_y^w.
\]

(4.21c)

In order to ensure the interface terms bounds the energy estimate, the following relations must hold:

\[
H_v I_{C2F} = I_{F2C}^T H_w, \\
H_w (I_C - I_{F2C} I_{C2F}) \geq 0, \quad H_v (I_F - I_{C2F} I_{F2C}) \geq 0.
\]

(4.22)

(4.23)

The \( \tau \) parameters must be set according to eq. (4.16). Conservation is not a necessary requirement, but will increase the accuracy, and is guaranteed with

\[
H_w I_{C2F} I_{F2C} = H_w, \quad H_v I_{F2C} I_{C2F} = H_v.
\]

(4.24)
4.11 Shifting grid points

In [21], better convergence and accuracy is obtained by moving some points in the difference stencil closer to the boundary, obtaining a new difference operator. This effect does not persist when using grid-mapping* from a regular physical space to this modified computational space†. The idea of shifting the sampling of the boundary stencil is also examined in [22], where two different SBP operators are used on the same mesh. This paper will heavily use the idea of the $h/2$ (“staggered” in [22]) operators to examine better interface couplings. The $h/2$ operator is illustrated in fig. 5, as contrasted with the traditional configuration in fig. 3. These operators tend to stiffen the problem, and requires a decrease in CFL.

---

*see appendix A
†tested preliminary of this work
5 Results and Discussion

This section demonstrates the suitability of upwind operators with interpolation between two meshes for scenarios where grid adaptations are necessary. The convergence and stability of an Euler vortex travelling along the vortex is examined, and the efficiency when modelling a heterogeneous physical situation is determined.

5.1 Implementation and Replication

The implementation is available at https://gitlab.com/mulimoen/EulerSolver. This repository contains all the necessary test-cases and details needed to replicate the results of this paper. The time discretisation is an explicit Runge-Kutta 6 method [1]. With a time-step determined from

\[ \Delta t = \text{CFL} \sum_g \min(h_x, h_y) = \text{CFL} \sum_g \min \left( \frac{1}{n_x - 1}, \frac{1}{n_y - 1} \right). \]  (5.1)

CFL will be fixed at 0.2, as the operators do not differ significantly in stiffness for the problems in this paper.

5.2 Convergence

This section focuses on the convergence properties of different upwind operators. The convergence rate is determined from the difference of the discrete solution to the analytical solution in eq. (2.6).

The wish is to split a physical domain with two heterogeneous vortices, see fig. 12a, into the coarse and fine resolved mesh as in fig. 12b. To ensure splitting the domain does not interfere with the accuracy of the solution, especially along the interface, a setup is constructed with a vortex travelling along the interface.

The domain is split in three different ways, the first (fig. 6a) does not have a split, the second (fig. 6b) has two conforming meshes, and the third (fig. 6c) has two non-conforming meshes with an interpolating interface. The parameters for the vortex is given in table 1, and the vortex is integrated to \( T = 2 \).

The setup with the single domain works as a benchmark for accuracy. The resulting errors and convergence rates can be found in table 2. The convergence order is consistent with the internal schemes of the different order operators.
Table 1: Parameters for the vortex used to determine the convergence in section 5.2

<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>Mach</th>
<th>$r_0$</th>
<th>$x_0$</th>
<th>$y_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
<td>−1.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Introducing an interface in the middle of the vortex gives the results in table 3. The introduction of the interface lowers the convergence rate to the boundary order for the regular upwind operators. Using the $h/2$ operator helps regain an order in the convergence rate for the 9th order operator.

Adding interpolation to the setup yields table 4. The convergence rate for the regular operators will be lower about an order lower than for the conforming meshes. An increase in convergence order with the $h/2$ operator can again be seen, now increasing the convergence rate by almost two. Using the $h/2$ operator in both directions gives a different interpolation scheme, but results in the same accuracy. This is consistent with earlier testing for periodic problems with higher order interpolation, where the order of interpolation was not seen to increase the accuracy of the solution, as long as the interpolation order was at least the boundary order of the difference operator.

The convergence results for the interpolating interface in table 4 is also given in fig. 7, including the time to run the simulation. The 4th order $h/2$ operator gives a minor gain in efficiency over the regular 4th order operator. Using a regular 9th order operator will be more efficient in time over the 4th order operators. This is further shown for the 9th order $h/2$ operator, which gives a significant advantage over the other three operators, which is also seen in the convergence rate in the table.

### 5.3 Stability

This section ensures that the introduction of the interpolating interface will be stable. The domain is split into a coarse and a fine domain, with a vortex travelling along this interface, see fig. 9a. The parameters for the vortex are given by table 5, and the vortex is integrated up to $T = 1000$, with the boundaries at the left and the right side set by periodic SAT.

Figure 8 shows how the traditional operators solve the Euler vortex. The traditional operators do not dampen the solution enough, allowing errors to propagate in the domain. This leads to a break-up of the vortex, even with such a weak vortex (almost linear) as here. By using the upwind operator, see fig. 9, the vortex can be solved for longer timescales. The additional artificial
(a) Initial conditions covered with a single mesh
(b) Initial conditions covered with two conforming meshes
(c) Initial conditions covered with two non-conforming meshes
(d) Solution for two non-conforming meshes using the 9th order upwind operator in $x$ and the 9th order upwind $h/2$ operator in $y$

Figure 6: Initial conditions and solution for the different scenarios used to determine the convergence properties
Table 2: Convergence results for single mesh  

(a) Regular upwind operators in $x, y$

<table>
<thead>
<tr>
<th>$N_x \times N_y$</th>
<th>$\log | e(4th) |_H$</th>
<th>$q(4th)$</th>
<th>$\log | e(9th) |_H$</th>
<th>$q(9th)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$100 \times 100$</td>
<td>-2.16</td>
<td>-3.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$140 \times 140$</td>
<td>-2.72</td>
<td>3.82</td>
<td>-4.97</td>
<td>8.04</td>
</tr>
<tr>
<td>$200 \times 200$</td>
<td>-3.35</td>
<td>4.11</td>
<td>-6.31</td>
<td>8.62</td>
</tr>
<tr>
<td>$284 \times 284$</td>
<td>-4.07</td>
<td>4.70</td>
<td>-7.65</td>
<td>8.84</td>
</tr>
<tr>
<td>$400 \times 400$</td>
<td>-4.80</td>
<td>4.89</td>
<td>-8.98</td>
<td>8.93</td>
</tr>
</tbody>
</table>

(b) Regular upwind operator in $x$, upwind $h/2$ operator in $y$

<table>
<thead>
<tr>
<th>$N_x \times N_y$</th>
<th>$\log | e(4th) |_H$</th>
<th>$q(4th)$</th>
<th>$\log | e(9th) |_H$</th>
<th>$q(9th)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$100 \times 100$</td>
<td>-2.27</td>
<td>-3.78</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$140 \times 140$</td>
<td>-2.82</td>
<td>3.74</td>
<td>-4.96</td>
<td>8.06</td>
</tr>
<tr>
<td>$200 \times 200$</td>
<td>-3.48</td>
<td>4.27</td>
<td>-6.30</td>
<td>8.64</td>
</tr>
<tr>
<td>$284 \times 284$</td>
<td>-4.20</td>
<td>4.75</td>
<td>-7.65</td>
<td>8.86</td>
</tr>
<tr>
<td>$400 \times 400$</td>
<td>-4.93</td>
<td>4.87</td>
<td>-8.98</td>
<td>8.94</td>
</tr>
</tbody>
</table>

Table 3: Convergence results for two conforming meshes

(a) Regular upwind operators in $x, y$

<table>
<thead>
<tr>
<th>$N_x \times N_y + N_x \times N_y$</th>
<th>$\sum | e(4th) |_H$</th>
<th>$q(4th)$</th>
<th>$\sum | e(9th) |_H$</th>
<th>$q(9th)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$100 \times 50 + 100 \times 50$</td>
<td>-1.88</td>
<td>-2.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$140 \times 70 + 140 \times 70$</td>
<td>-2.42</td>
<td>3.70</td>
<td>-3.66</td>
<td>5.20</td>
</tr>
<tr>
<td>$200 \times 100 + 200 \times 100$</td>
<td>-3.07</td>
<td>4.14</td>
<td>-4.32</td>
<td>4.20</td>
</tr>
<tr>
<td>$284 \times 142 + 284 \times 142$</td>
<td>-3.74</td>
<td>4.42</td>
<td>-5.04</td>
<td>4.77</td>
</tr>
<tr>
<td>$400 \times 200 + 400 \times 200$</td>
<td>-4.34</td>
<td>4.07</td>
<td>-5.80</td>
<td>5.08</td>
</tr>
</tbody>
</table>

(b) Regular upwind operator in $x$, upwind $h/2$ operator in $y$

<table>
<thead>
<tr>
<th>$N_x \times N_y + N_x \times N_y$</th>
<th>$\sum | e(4th) |_H$</th>
<th>$q(4th)$</th>
<th>$\sum | e(9th) |_H$</th>
<th>$q(9th)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$100 \times 50 + 100 \times 50$</td>
<td>-1.99</td>
<td>-3.67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$140 \times 70 + 140 \times 70$</td>
<td>-2.52</td>
<td>3.65</td>
<td>-4.70</td>
<td>7.07</td>
</tr>
<tr>
<td>$200 \times 100 + 200 \times 100$</td>
<td>-3.19</td>
<td>4.34</td>
<td>-5.74</td>
<td>6.66</td>
</tr>
<tr>
<td>$284 \times 142 + 284 \times 142$</td>
<td>-3.92</td>
<td>4.77</td>
<td>-6.64</td>
<td>5.92</td>
</tr>
<tr>
<td>$400 \times 200 + 400 \times 200$</td>
<td>-4.64</td>
<td>4.81</td>
<td>-7.47</td>
<td>5.60</td>
</tr>
</tbody>
</table>
Table 4: Convergence results for two non-conforming meshes

(a) Regular upwind operators in $x$, $y$

<table>
<thead>
<tr>
<th>Mesh Size</th>
<th>$\sum | e(4th) |_H$</th>
<th>$q(4th)$</th>
<th>$\sum | e(9th) |_H$</th>
<th>$q(9th)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$99 \times 50 + 50 \times 25$</td>
<td>-1.11</td>
<td>-1.53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$139 \times 70 + 70 \times 35$</td>
<td>-1.55</td>
<td>3.03</td>
<td>-1.99</td>
<td>3.10</td>
</tr>
<tr>
<td>$199 \times 100 + 100 \times 50$</td>
<td>-2.11</td>
<td>3.59</td>
<td>-3.08</td>
<td>7.06</td>
</tr>
<tr>
<td>$283 \times 142 + 142 \times 71$</td>
<td>-2.67</td>
<td>3.65</td>
<td>-3.64</td>
<td>3.67</td>
</tr>
<tr>
<td>$399 \times 200 + 200 \times 100$</td>
<td>-3.30</td>
<td>4.25</td>
<td>-4.28</td>
<td>4.29</td>
</tr>
</tbody>
</table>

(b) Regular upwind operator in $x$, upwind $h/2$ operator in $y$

<table>
<thead>
<tr>
<th>Mesh Size</th>
<th>$\sum | e(4th) |_H$</th>
<th>$q(4th)$</th>
<th>$\sum | e(9th) |_H$</th>
<th>$q(9th)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$99 \times 50 + 50 \times 25$</td>
<td>-1.17</td>
<td>3.24</td>
<td>-2.24</td>
<td>4.13</td>
</tr>
<tr>
<td>$139 \times 70 + 70 \times 35$</td>
<td>-1.65</td>
<td>3.59</td>
<td>-2.85</td>
<td>6.21</td>
</tr>
<tr>
<td>$199 \times 100 + 100 \times 50$</td>
<td>-2.20</td>
<td>3.74</td>
<td>-3.90</td>
<td>6.87</td>
</tr>
<tr>
<td>$283 \times 142 + 142 \times 71$</td>
<td>-2.77</td>
<td>4.51</td>
<td>-4.94</td>
<td>6.24</td>
</tr>
<tr>
<td>$399 \times 200 + 200 \times 100$</td>
<td>-3.45</td>
<td>4.51</td>
<td>-5.87</td>
<td>6.24</td>
</tr>
</tbody>
</table>

Figure 7: Convergence plot of the vortex travelling along the interface between two different resolution meshes, given by table 4. The error is plotted against both DOF and time. The operators are the regular upwind operators in $x$, $y$ when not annotated with $h/2$, and a regular upwind operator in $x$ and upwind $h/2$ operator in $y$ for the other.
Table 5: Parameters for the vortex used to determine the stability in section 5.3

<table>
<thead>
<tr>
<th>ε</th>
<th>Mach</th>
<th>ra</th>
<th>x₀</th>
<th>y₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.5</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

dissipation from the upwind behaviour helps stabilise, and is necessary for the Euler vortex.

Figure 10 shows the error for different combinations of operators. It is clear that the traditional operators have a swift increase in error before the solution breaks up. The 9th order regular upwind operator forms a baseline to compare the $h/2$ operators. Using the $h/2$ operator in just $y$ is more efficient than using the operator in just $x$. As the vortex travels mostly in the interface which requires a higher accuracy for this difference operator, this is not surprising. Combining the two $h/2$ operators do however lead to a large effect on the accuracy.

5.4 Efficiency

To measure the performance of the SBP-SAT model, a setup with the parameters given in table 1, with $T = 2$ is examined. Using a single mesh, the average time for each run is recorded. The resulting time against DOF is plotted in fig. 11. The time complexity is expected to be linear in DOF, and increasing as the square of DOF as the time-step has to decrease, for a combined time complexity $O(DOF^{1.5})$. This is in accordance with the complexity measured experimentally.

For three dimensional domains, the expected growth in time complexity will be $t \propto DOF \cdot \sqrt{DOF}$. As a reduced time-step is comparatively cheap, the spatial accuracy can come at the expense of the time-step and still be worth it. With the SBP-SAT method having low communication overhead in a multi-mesh setting, the method is highly scalable.

The atmospheric model that is now considered is illustrated by fig. 12. A strong small vortex will simulate the ground effects, with a larger weaker vortex simulating the upper atmosphere. The vortices are given by table 6, and the solution is integrated to $T = 2$. Comparing the two ways to split the domain, one by covering the domain with a single mesh (a traditional approach) and another by dividing the domain in two, with the top half being covered by a coarser resolution mesh, results in table 7. This is plotted in figs. 13a and 13b for error against both DOF and time. The accuracy improves for both when using the double grid with interpolation.
Figure 8: Long time integration using traditional 8th order operators
Figure 9: Long time integration using a regular 9th order upwind operator in $x$ and an $h/2$ 9th order operator in $y$.
Figure 10: Error when the vortex in fig. 8 and fig. 9 is integrated over many periods. The 8th order operator is a traditional non-upwind operator and breaks up after 12 periods. The 4th and 9th order operators are the upwind operators. The size of the meshes are $50 \times 25$ and $100 \times 50$, with the higher resolution grid on the bottom.

Figure 11: Time plotted against degrees of freedom for computations on a single mesh. The time can be fitted by $T \propto \text{DOF}^{1.53}$. 

29
Figure 12: Initial conditions for the model atmosphere for the two different grid adaptations

Figure 13: Total error for the heterogeneous atmosphere used as efficiency test. The figures show the error against DOF and time. The operators used are the 9th order upwind $h/2$ operators.
Table 6: Parameters for the two vortices in section 5.4 used for measuring the efficiency of the double mesh model.

<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>Mach</th>
<th>$r_s$</th>
<th>$x_0$</th>
<th>$y_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
<td>-1.0</td>
<td>5.0</td>
</tr>
<tr>
<td>6.0</td>
<td>0.5</td>
<td>1.0</td>
<td>-1.0</td>
<td>15.0</td>
</tr>
</tbody>
</table>

Table 7: Convergence results for the heterogeneous physical atmosphere

(a) Domain covered with a single mesh

| $N_x \times N_y$ | $\log ||e||_H$ | $q$ | Time [s] |
|------------------|---------------|-----|----------|
| 39 $\times$ 78   | -1.87         |     | 6.1      |
| 55 $\times$ 110  | -2.62         | 5.03| 13.9     |
| 78 $\times$ 156  | -3.11         | 3.19| 33.1     |
| 112 $\times$ 224 | -4.15         | 6.61| 87.0     |
| 157 $\times$ 314 | -5.36         | 8.28| 203.8    |

(b) Domain covered with two meshes

| $N_x \times N_y + N_x \times N_y$ | $\log \sum ||e||_H$ | $q$ | Time [s] |
|-----------------------------------|---------------------|-----|----------|
| 48 $\times$ 50 + 25 $\times$ 25   | -2.11               |     | 6.0      |
| 68 $\times$ 70 + 35 $\times$ 35   | -2.65               | 3.66| 13.9     |
| 90 $\times$ 100 + 50 $\times$ 50  | -3.59               | 6.01| 33.5     |
| 140 $\times$ 142 + 71 $\times$ 71| -4.83               | 8.10| 92.4     |
| 198 $\times$ 200 + 100 $\times$ 100| -6.09              | 8.43| 228.0    |
6 Conclusions

Using the SBP-SAT method on a single grid yields high convergence orders for single meshes. Adding an interface between two meshes will come at a penalty. Some of this penalty can be avoided by making a minor change to the mesh, moving a single point closer to the boundary to form the \( h/2 \) mesh and corresponding difference operator.

Adding an interpolation between the two meshes adds another interpolation error to the total error. The work here suggests this is mostly due to low accuracy of the difference operators close to the boundary, and using the \( h/2 \) operator will help increase the accuracy of the solution. The accuracy for a heterogeneous system can be improved by using grid adaptations. This is shown with two different strength vortices on the same domain and comparing the traditional approach with a single mesh to an adaptive grid approach. This is shown to be advantageous when comparing error with the time required to run the integration.

Even though the method is provably linearly stable, the traditional operators alone are not stable enough to solve the non-linear Euler equations. Using the upwind operators provides the artificial damping that is necessary to compute a solution for long time-integration. Combining this with the \( h/2 \) approach gives a stable and high-accuracy solution.

This thesis has focused on applying the SBP-SAT method to problems encountered in atmospheric modelling. Adding grid adaptation to the SBP-SAT model requires some minor changes to the mesh, which ensures a good convergence even for areas close to the interface between the meshes. The final result is a stable, convergent, conservative and efficient model that can be applied to the atmosphere.
References


List of Figures

1. Example of two methods with different convergence order . . 2
2. Two dimensional grid . . . . . . . . . . . . . . . . . . . . . . 16
3. Example of conforming grid combination . . . . . . . . . . 18
4. Example of non-conforming grid combination . . . . . . . . 19
5. $h/2$ grid in the $x$ direction . . . . . . . . . . . . . . . . 20
6. Initial conditions and solution for the different scenarios used to determine the convergence properties . . . . . . . . . . 23
7. Convergence plot of vortex travelling on interface of non-conforming meshes . . . . . . . . . . . . . . . . . . . . . . . . . 25
8. Stability for traditional operators . . . . . . . . . . . . . . 27
9. Stability for upwind operators . . . . . . . . . . . . . . . . 28
10. Long time error for vortex along interpolation interface . . 29
11. Time against DOF for single mesh . . . . . . . . . . . . . . 29
12. Initial conditions for the model atmosphere for the two different grid adaptations . . . . . . . . . . . . . . . . . . . . . . 30
13. Total error for the heterogeneous atmosphere used as test for efficiency . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
14. Curvilinear coordinate system . . . . . . . . . . . . . . . . 37

List of Tables

1. Convergence vortex parameters . . . . . . . . . . . . . . . . 22
2. Convergence results for single mesh . . . . . . . . . . . . . . 24
3. Convergence results for two conforming meshes . . . . . . . 24
4. Convergence results for two non-conforming meshes . . . . . 25
5. Stability vortex parameters . . . . . . . . . . . . . . . . . . 26
6. Efficiency vortex parameters . . . . . . . . . . . . . . . . . 31
7. Convergence results for the heterogeneous physical atmosphere 31
A Curvilinear Mapping

A difference operator can be created for any arbitrary point, using any neighbouring points. It is however expensive to compute and store a unique difference operator for every point in a grid. If all points are spaced at regular intervals, one operator can be used for all the points, with some adjustments for points close to the boundary. This will be problematic for grids such as fig. 14, where the stencils are hard to determine. The difference operators working on a evenly spaced grid such as fig. 2, can however be mapped to a grid such as fig. 14 by using curvilinear mapping, as detailed in this section.

The notation used here differs from the rest of the paper. The computational space is given by fig. 2, with \( x \) replaced by \( \xi \) and \( y \) replaces by \( \eta \). The physical space such as in fig. 14 will now have positions denoted by \( x \) and \( y \).

A.1 Reformulating to Computational Space

The original PDE which we wish to solve is given by eq. (2.4), repeated here for completeness,

\[
    u_t + F_x + G_y = 0,
\]

where \( F \) and \( G \) might be nonlinear functions. This can be written in the non-conservative form

\[
    u_t + \xi_x F_\xi + \xi_y G_\xi + \eta_x F_\eta + \eta_y G_\eta = 0, \tag{A.1}
\]

or the conservative form

\[
    \det\{J\} u_t + \hat{F}_\xi + \hat{G}_\eta = 0, \tag{A.2}
\]

with the new fluxes \( \hat{F} \) and \( \hat{G} \) given by

\[
    \hat{F} = \det\{J\} (\xi_x F + \xi_y G), \tag{A.3a}
\]

\[
    \hat{G} = \det\{J\} (\eta_x F + \eta_y G). \tag{A.3b}
\]
The matrix $J$ is given by the metric Jacobian

$$J = \begin{pmatrix} x_{\xi} & y_{\xi} \\ x_{\eta} & y_{\eta} \end{pmatrix}$$  \hspace{1cm} (A.4)

### A.2 Conservative treatment

Computing the derivative of a constant function should give zero. Using the conservative form gives the partial derivatives which cancels out:

$$u_x = \frac{1}{\det\{J\}} \left( (\det\{J\} \xi_x u)_{\xi} + (\det\{J\} \eta_x u)_{\eta} \right)$$  \hspace{1cm} (A.5a)

$$= \frac{1}{\det\{J\}} (y_{\xi} - y_{\eta}) u = 0.$$  \hspace{1cm} (A.5b)

For the discrete case, the order of the difference operators matters,

$$u_x = D_{\xi}(\xi_x u) + D_{\eta}(\eta_x u) = D_{\xi}((D_{\xi}x) u) + D_{\eta}((D_{\xi}x) u) \neq 0.$$  \hspace{1cm} (A.6)

As we do not have any more constraints on the difference operators, they will not cancel out, and the solution will contain discretisation errors. Using the conservative form gives (with metric derivatives denoted with a hat);

$$v_x = \frac{1}{\det\{J\}} \left( D_{\eta}(\hat{D}_{\xi}x)u - D_{\xi}(\hat{D}_{\eta}x)u \right)$$  \hspace{1cm} (A.7a)

$$= \frac{1}{\det\{J\}} \left( D_{\eta}\hat{D}_{\xi}x - D_{\xi}\hat{D}_{\eta}x \right) u.$$  \hspace{1cm} (A.7b)

As the operators commute, choosing the metric difference operators to be the same as the the regular difference operators gives the discrete analogy to eq. (A.5).

### A.3 Derivation of Conservative form

The metric Jacobians gives by the chain rule the relation between the metrics,

$$\begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} = \begin{pmatrix} \xi_x & \eta_x \\ \xi_y & \eta_y \end{pmatrix} \begin{pmatrix} \partial_{\xi} \\ \partial_{\eta} \end{pmatrix},$$  \hspace{1cm} (A.8a)

$$\begin{pmatrix} \partial_{\xi} \\ \partial_{\eta} \end{pmatrix} = \begin{pmatrix} x_{\xi} & y_{\xi} \\ x_{\eta} & y_{\eta} \end{pmatrix} \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix}.$$  \hspace{1cm} (A.8b)

From this it is clear that the relation

$$\begin{pmatrix} \xi_x & \eta_x \\ \xi_y & \eta_y \end{pmatrix} = \begin{pmatrix} x_{\xi} & y_{\xi} \\ x_{\eta} & y_{\eta} \end{pmatrix}^{-1} = \frac{1}{\det\{J\}} \begin{pmatrix} y_{\eta} & -y_{\xi} \\ -x_{\eta} & x_{\xi} \end{pmatrix}.$$  \hspace{1cm} (A.9)
holds between the metrics. The geometrical conservation laws* are given by:

\[\begin{align*}
(J_x)_{\xi} + (J_{\eta})_{\eta} &= (y_{\eta})_{\xi} + (-y_{\xi})_{\eta} = 0, \\
(J_y)_{\xi} + (J_{\eta})_{\eta} &= (-x_{\eta})_{\xi} + (x_{\xi})_{\eta} = 0.
\end{align*}\] (A.10a)

Using commutation of the difference operators, we also have

\[\begin{align*}
(J_x)_{\xi} &= (\xi_x)_x = 0
\end{align*}\] (A.11)

And equivalent for \(\eta\). These properties ensure we can include them in the flux

\[\begin{align*}
\det\{J\} \xi_x(F)_{\xi} + \det\{J\} \eta_x(F)_{\eta} &= (\det\{J\} \xi_x F)_{\xi} + (\det\{J\} \eta_x F)_{\eta}, \quad (A.12)
\end{align*}\]

It is now trivial to combine these properties, using the chain rule on the original PDE,

\[\begin{align*}
u_t + \xi_x F_{\xi} + \eta_x F_{\eta} + \xi_y G_{\xi} + \eta_y G_{\eta} &= 0, \\
u_t + (\xi_x F)_{\xi} + (\eta_x F)_{\eta} + (\xi_y G)_{\xi} + (\eta_y G)_{\eta} &= 0, \\
\det\{J\} u_t + (\det\{J\} \xi_x F)_{\xi} + (\det\{J\} \eta_x F)_{\eta} + (\det\{J\} \xi_y G)_{\xi} + (\det\{J\} \eta_y G)_{\eta} &= 0, \\
\det\{J\} u_t + \hat{F}_{\xi} + \hat{G}_{\eta} &= 0. \quad (A.13d)
\end{align*}\]

Which is equivalent to eq. (A.2).

---

*Necessary in three dimensions