

UC Santa Cruz

UC Santa Cruz Electronic Theses and Dissertations

Title

SORh: Hyperbolic Relaxation Methods For Elliptic Problems In Computational Fluid Dynamics

Permalink

<https://escholarship.org/uc/item/5gc906gb>

ISBN

9798190914900

Author

Guite, Jonathan

Publication Date

2026-08-20

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

SANTA CRUZ

**SORH: HYPERBOLIC RELAXATION METHODS FOR ELLIPTIC
PROBLEMS IN COMPUTATIONAL FLUID DYNAMICS**

A thesis submitted in partial satisfaction
of the requirements for the degree of

MASTER OF SCIENCE

in

SCIENTIFIC COMPUTATION & APPLIED MATHEMATICS

by

Jonathan Guité

August 2026

The Thesis of Jonathan Guité
is approved:

Professor Dongwook Lee, Chair

Professor Nic Brummell

Professor Daniele Venturi

Donald Smith
Interim Vice Provost and
Dean of Graduate Studies

Contents

Acknowledgments	x
1 Introduction	1
2 One-Dimensional Formulation, Methods, and Analysis	6
2.1 One-Dimensional Boundary Value Problem	6
2.2 Parabolic Pseudo-Time Model	8
2.3 Spatial and Temporal Discretization	8
2.4 Parabolic Iterative Methods: Jacobi, Gauss–Seidel, and SOR	9
2.5 Hyperbolic Pseudo-Time Model	10
2.6 SORh: 1D Hyperbolic SOR Formulation	11
2.7 Accuracy, Consistency, and Stability of SORh	12
2.8 Connection to AOR and Classical SOR	13
2.9 Spectral Analysis of the Iterative Methods	15
2.9.1 Comparison of Spectral Behavior	16
2.10 Numerical Diagnostics and Transition to Results	18
3 Results of One-Dimensional Experiments	20
3.1 Experimental Setup	20
3.2 Linear Performance Comparison	21
3.3 Nonlinear Performance Comparison	23
3.3.1 Solution Accuracy & Summary	25

3.4	Behavior of the Relaxation Parameter	25
4	Two-Dimensional Formulation, Methods, and Analysis	28
4.1	Governing Elliptic Problems	29
4.1.1	Poisson's Equation	29
4.1.2	Nonlinear Diffusion Problem	30
4.2	Two-Dimensional Pseudo-Time Setting	31
4.2.1	Parabolic vs. Hyperbolic Pseudo-time Models	31
4.3	Two-Dimensional Hyperbolic Models	33
4.3.1	Coupled Advection Model	34
4.3.2	Directional Transport Models	34
4.4	Common Discrete Framework	36
4.4.1	Discrete Elliptic Operators and Residuals	37
4.5	Discrete Methods: SORp and SORh	39
4.5.1	Parabolic SOR (SORp)	39
4.5.2	Hyperbolic SOR (SORh)	40
4.6	Connection to AOR and Classical SOR	42
4.6.1	AOR and Classical SOR	42
4.6.2	Optimal Relaxation Parameters	44
4.6.3	Pseudo-Time-Step Scaling	45
4.7	Directional SORh	46
4.8	Diagnostics for Two-Dimensional Experiments	47
5	Results of Two-Dimensional Experiments	49
5.1	Two-Blob Poisson Problem	49
5.1.1	Performance Results	50
5.1.2	Solution Accuracy	51
5.2	Nonlinear Diffusion Problem	53
5.2.1	Optimized Performance and Grid Scaling	53

5.2.2	Coefficient Contrast	54
5.2.3	Supporting Parameter Studies	56
5.3	FLASH Experiments	57
5.3.1	Zel’dovich Pancake	57
6	Conclusion	62
6.1	Results, Interpretations, & Limitations	63
6.2	Alternative Methods and Future Work	65
6.3	The End	67
A	Supplemental Analysis	68
A.1	One-Dimensional Coefficient Comparison of SORh, AOR, and Classical SOR	68
A.2	Two-Dimensional SORh–AOR Equivalence	70
A.3	Pseudo-Time Iteration Algorithm	71
	Source Code Availability	73

List of Figures

2.1	Spectral radius $\rho(G)$ as a function of the relaxation parameter ω for the parabolic and hyperbolic SOR methods, with $N_{\text{int}} = 64$	17
3.1	Iteration count and runtime for the linear problem as the number of interior unknowns is refined. SORp and SORh use experimentally optimized relaxation parameters.	21
3.2	Iteration count and runtime for the nonlinear problem using the same convergence tolerance and grid sequence as in the linear comparison. . .	23
3.3	Predicted and experimentally measured optimal relaxation parameters for SORh and SORp on the linear and nonlinear problem.	26
3.4	Sensitivity of nonlinear convergence to the relaxation parameter at fixed grid resolution, $N_{\text{int}} = 128$. Vertical markers indicate the experimental optimum and highest convergent relaxation parameter.	26
5.1	Two-blob Poisson test problem. Panel (a) shows contours of the prescribed source term $f(x, y)$, while panel (b) shows contours of the corresponding exact solution $\phi_{\text{exact}}(x, y)$. The exact solution is used to impose the Dirichlet boundary conditions and evaluate the numerical error. . .	50

5.2	Performance of SORp and SORh for the two-blob problem using the selected relaxation parameters. Panel (a) shows the iteration counts, and panel (b) shows the mean runtimes over repeated trials. At $\text{CFL} = 1$, the methods require the same number of iterations, although SORp is slightly faster because its discrete update is less expensive. At $\text{CFL} = 0.8$, SORh also requires more iterations as the grid is refined. Runtime error bars are omitted because the measured variation between repetitions is negligible on the report scale.	52
5.3	Spatial accuracy for the two-blob problem, represented by the SORp results at $\text{CFL} = 1$. Panel (a) shows the final pseudo-time RMS solution error e_s , and the RMS truncation error, obtained by applying the discrete operator to the exact solution. Panel (b) shows the corresponding local observed convergence orders. The coarse grids exhibit pre-asymptotic behavior, while both error measures approach second-order convergence under refinement. The other method and CFL configurations produce effectively indistinguishable accuracy results.	52
5.4	Optimized fixed-timestep performance for the two-dimensional nonlinear problem. Panel (a) shows sweep counts and descriptive power-law fits over $N = 16, 32, 64$; panel (b) shows the corresponding SORh sweep advantage.	53
5.5	Effect of reducing coefficient contrast at $N = 32$. Panel (a) shows optimized sweep counts, panel (b) their ratio, and panel (c) the selected relaxation parameters normalized by the linear predictions. The horizontal reference in panels (b) and (c) is one.	55
5.6	Pre-caustic profiles at $z \simeq 10.419$. MG and wavefront SORh with $m_{\max} = 200$ include the same hydrodynamic and particle discretization errors. .	60

List of Tables

2.1	Spectral-radius comparison for the one-dimensional SOR formulations at $N_{\text{int}} = 64$ and $\text{CFL} = 0.8$. Here, ω_{opt} is obtained from numerically calculated ρ_{min} and ω_{pred} are calculated using Eq. 2.37.	17
3.1	Coefficients and parameters used in the 1D experiments.	21
3.2	Representative final diagnostics for the optimized SOR methods at $N_{\text{int}} = 32$ and $\varepsilon_u = 10^{-7}$	25
5.1	Numerical configuration and parameters used for the two-blob performance and accuracy studies.	51
5.2	Final-state diagnostic of the weakest local upstream-correction factor in the optimized nonlinear runs.	54
5.3	Shared Pancake setup and solver-specific controls. The 16×16 cell block layout (64 blocks) is used for the primary comparison.	58
5.4	Primary MG–SORh timing comparison. Times are medians of three repetitions in seconds; ratios use MG at the same rank count.	59
5.5	Four-rank wavefront budget study. Time ratios between SORh and MG for both total wall-clock time and Poisson-solver time are shown. The last two columns are the largest relative L_2 difference from matched MG among gas density, normal velocity, particle density, pressure, and derived temperature.	60

Abstract

SORh: Hyperbolic Relaxation Methods for Elliptic Problems in Computational Fluid Dynamics

Jonathan Guité

This thesis investigates iterative methods for solving elliptic partial differential equations (PDEs) in computational fluid dynamics (CFD), with particular emphasis on self-gravity modeled by Poisson’s equation. Available solution strategies—including direct factorizations, FFT-based methods, multigrid, and classical relaxation schemes—offer different balances among computational cost, scalability, implementation complexity, and adaptability to changing boundaries and mesh structures. In this work, we examine how hyperbolic pseudo-time dynamics can be used to balance these considerations while providing a simple, local route to an elliptic steady state.

To explore this question, we develop SORh, a relaxation framework derived from a hyperbolic pseudo-time reformulation of elliptic equations. Its linear discretization belongs to the established accelerated over-relaxation (AOR) family and reduces exactly to classical successive over-relaxation (SOR) at unit CFL. The derivation interprets AOR’s two parameters as directional correction propagation and residual injection. One- and two-dimensional formulations are analyzed for consistency, stability, spectral behavior, and pseudo-time scaling. Numerical experiments recover classical-SOR behavior for constant-coefficient linear problems and show substantially fewer iterations for selected nonlinear problems characterized by weak local diffusion and strong coefficient variation. Directional and parallel formulations culminate in an optimized FLASH implementation with performance comparable to its production multigrid solver on the tested self-gravity problem. A companion application of SORh to MHD divergence cleaning is reported separately.

Acknowledgments

The software used in this work was developed in part by the DOE NNSA- and DOE Office of Science-supported Flash Center for Computational Science at the University of Chicago and the University of Rochester.

This work was supported in part by the Department of Energy (DOE) under the supplemental award to the Accelerator R&D and Production grant DE-SC0023152.

Prof. Dongwook Lee — thesis advisor

Prof. Nic Brummell and Prof. Daniele Venturi — committee members

Chapter 1

Introduction

This thesis explores iterative methods for solving one- and two-dimensional elliptic PDEs. Elliptic PDEs are a very important class of equation, as they can be used to model many physical phenomena, including a variety of equilibrium or steady-state systems. One of the canonical examples of an elliptic PDE is Poisson's equation, characterized simply as $\Delta\phi = f$. Poisson's equation is used for many applications in physics and engineering, such as fluid dynamics and electrostatics. This thesis focuses on iterative numerical schemes that solve Poisson's equation and similar elliptic systems.

Elliptic PDEs have many useful analytical properties, and their behavior under many different configurations, in a continuous and discrete sense, are well-characterized. Under standard regularity assumptions, the Poisson equation with Dirichlet boundary conditions defines a well-posed boundary value problem (BVP) [8]. This implies the problem has a unique solution, which is especially important for iterative solvers, because it ensures that two consistent but distinct solvers applied to the same consistent discretization cannot converge to different solutions. Elliptic equations are also known to possess strong regularity properties [8], which often ensure sufficient smoothness of the solution to analyze the order of accuracy of finite-difference discretizations using Taylor-series expansions [19].

For a finite-difference scheme applied to an elliptic problem, spectral properties

of the elliptic operator and its discrete matrix can help us understand the behavior of the numerical scheme. After finite differencing, a standard elliptic problem gives a linear system $A\mathbf{u} = \mathbf{f}$, where A often has a real, positive spectrum, and for common problems such as Poisson with Dirichlet BCs it is symmetric positive definite [4, 27]. This allows us to understand the method mode-by-mode. For pseudo-time relaxation methods, we can use this alongside an error evolution equation to understand the propagation of errors throughout the spatial domain and through pseudo-time. For example, a pseudo-time method of the form

$$u_t = -Au + f, \tag{1.1}$$

can be described by the error equation

$$e_t = -Ae. \tag{1.2}$$

Expanding Eq. 1.2 according to the spectrum of A then reveals how different spatial error modes evolve and decay over time. In general, methods of the form (1.1) damp high-frequency error modes rapidly, while low-frequency errors decay slowly [4]. This numerical smoothing property is characteristic of relaxation methods like Jacobi and Gauss–Seidel applied to discrete elliptic systems.

In an elliptic BVP, any local disturbance or change in boundary solution can affect the solution everywhere. For a finite-difference elliptic matrix, this corresponds to a globally coupled solution operator. Although these matrices are typically sparse, direct factorizations may produce substantially larger coupling structures, which increases memory use and computational costs. More generally, direct matrix inversion and other direct or implicit methods can be impractical due to high implementation complexity and computational costs. For instance, introducing more than one spatial dimension can cause even a moderate level of grid refinement to produce very large discrete elliptic systems. This is because in a grid with N points in each of d dimensions, the total number of unknowns scales like N^d . Furthermore, modular AMR environments such

as FLASH can introduce new complexities. For example, FFT-based solvers can be highly efficient on certain structured grids with compatible boundary conditions, but they may under-perform on irregular domains, nonstandard boundary conditions, or adaptive mesh structures. These limitations motivate the use of methods with a more local and adaptable update structure. Because these methods rely on local update formulas rather than global solvers, they can be adapted to a much broader range of configurations, and can be parallelized much more naturally, although ordered Gauss–Seidel and SOR sweeps do introduce their own parallel dependencies [10, 7].

Since a perturbation in an elliptic problem affects the whole domain, it is reasonable to expect that an iterative solver have some way of quickly propagating information across the domain. For local methods, though, the speed of information travel is highly limited. For example, iterative methods based on parabolic PDEs like the Jacobi method [27], under the appropriate discretizations, have a strict time-stepping requirement of $\Delta t = O(\Delta x^2)$ [19] and are governed by diffusion-based dynamics, which can result in extremely slow convergence to the steady-state solution.

One improvement to the Jacobi method is the Gauss–Seidel method, which under specific ordering requirements, uses recently updated upstream values as soon as they are available. This allows for faster propagation of information throughout the domain. Successive over-relaxation (SOR) then introduces a relaxation parameter ω , allowing for a larger correction toward the Gauss–Seidel update, further accelerating convergence. Classical SOR’s rate of acceleration is limited by the spectral radius of the discrete system, giving the known stability condition $0 < \alpha < 2$, where α is the relaxation parameter [27, 12].

The modification studied in this thesis is to consider a hyperbolic pseudo-time model, rather than the usual parabolic model. Hyperbolic PDEs do not act diffusively, but rather exhibit advection-based or wave-like behavior, such as that of the wave equation. The question, then, is whether these dynamics can be leveraged by a hyperbolic pseudo-time method to allow for more efficient propagation of information toward an

elliptic steady-state solution.

To address this, we embed the elliptic problem into a hyperbolic pseudo-time evolution equation as a controlled extension of classical SOR. This hyperbolic-SOR model, which we will present in one and two spatial dimensions, will be referred to as SORh. Similar to classical SOR, SORh will use a relaxation parameter to control the rate of propagation. Though “SOR” implies *over*-relaxation, the choice of under-relaxation or over-relaxation varies based on the stability requirements and convergence behavior of the given method. Understanding and optimizing the precise relaxation factor for the presented methods is a major point of discussion in this thesis.

Hyperbolic reformulations of diffusion provide an important precedent for this approach. Nishikawa replaces the second-order diffusion equation with an equivalent first-order hyperbolic relaxation system in which the solution gradients are evolved as auxiliary unknowns; the resulting schemes permit an explicit $O(h)$ time step rather than the usual parabolic $O(h^2)$ restriction [21]. SORh shares the broader objective of reaching a diffusive steady state through hyperbolic pseudo-time dynamics, but differs in its construction. It retains a scalar unknown and uses a mixed space–pseudo-time transport model whose ordered discretization produces an SOR-like correction sweep. Moreover, the nominal $O(h)$ pseudo-time step in SORh does not by itself imply an $O(h)$ elliptic correction, because the residual term carries an additional mesh-dependent scaling, as shown in Chapter 2.

At the discrete linear level, SORh belongs to the accelerated overrelaxation (AOR) family introduced by Hadjidimos as a two-parameter extension of SOR [11, 13, 12]. In fact, its parameters are related to the AOR parameters by a simple linear mapping. AOR can be seen as a generalization of classical SOR for specific parameter choices, and SORh exhibits the same reduction to classical SOR with the correct mapping from the AOR parameters. Thus, the two-parameter linear iteration is not itself new. The contribution developed here is its independent derivation from the hyperbolic model, the resulting interpretation of the parameters as correction propagation and

residual injection, and the extension and numerical characterization of this structure in nonlinear, directional, and parallel settings.

Numerical experiments show similar SORh and SORp behavior for the tested constant-coefficient linear problems, but substantially fewer SORh iterations for nonlinear problems with weak and strongly nonuniform diffusion. Directional and parallel studies culminate in a FLASH implementation that performs comparably to its production multigrid solver in the tested configuration, with only slightly reduced accuracy. A companion application uses SORh for projection-based divergence cleaning in a fifth-order finite-volume MHD scheme [17, 18]. Together, these results show practical behavior and applications of the hyperbolic formulation.

The remainder of the thesis is organized as follows. Chapter 2 introduces a one-dimensional boundary value problem (BVP) which is the basis of one-dimensional methods and experiments. We will briefly discuss classical iterative relaxation methods and introduce the one-dimensional formulation of SORh. Here we will establish fundamental results relevant to SORh and related relaxation methods. In Chapter 3, we will implement these methods in numerical experiments to evaluate and compare performance, accuracy, and stability of these methods. Chapter 4 will then extend the one-dimensional models and schemes into two dimensions. Due to the increased complexity, some alternative two-dimensional hyperbolic formulations will be discussed to justify the main presented form of SORh. Chapter 5 will then apply these methods to test problems based on a Poisson’s equation and a nonlinear diffusion problem. Chapter 5 will also include preliminary results from an implementation of SORh in the FLASH code. The method will be applied to the Zel’dovich Pancake problem [28, 1], a cosmological test problem with known analytical solutions. This will allow us to evaluate the performance and accuracy of SORh in a more realistic and physically relevant astrophysical simulation environment. Finally, Chapter 6 will present a unified interpretation of the methods and results studied in this thesis, including directions for future work.

Chapter 2

One-Dimensional Formulation, Methods, and Analysis

This chapter introduces the one-dimensional model problems used as the foundation for the numerical methods studied in this thesis. We will begin with a simple boundary value problem whose steady-state solution is easily characterized, reformulate it as a pseudo-time evolution problem, and then compare several iterative methods for reaching the steady state. This model and the later mixed-derivative reformulation are adapted from the finite-difference problems collected by Chattot [5]. The one-dimensional setting provides a controlled environment for developing the target hyperbolic pseudo-time method. Later chapters will extend these ideas to two-dimensional models, where we will consider more complex configurations and perform more in-depth analysis.

2.1 One-Dimensional Boundary Value Problem

The starting point is the nonlinear boundary value problem

$$\frac{d}{dx} \left(\nu(u) \frac{du}{dx} \right) = 0, \quad \nu(u) = \frac{u}{1+u}, \quad u \geq 0, \quad 0 < x < 1, \quad (2.1)$$

with boundary conditions

$$u(0) = 0, \quad u(1) = 1. \quad (2.2)$$

For this particular one-dimensional problem, the exact solution can be expressed implicitly as

$$u - \ln(1 + u) = (1 - \ln 2)x. \quad (2.3)$$

A useful linear comparison problem is obtained by replacing the nonlinear diffusion coefficient by a positive constant,

$$\nu(u) = \tilde{\nu} > 0. \quad (2.4)$$

The original boundary value problem then reduces to

$$\tilde{\nu} \frac{d^2 u}{dx^2} = 0, \quad u(0) = 0, \quad u(1) = 1, \quad (2.5)$$

whose steady-state solution is

$$u_{\text{ss}}(x) = x. \quad (2.6)$$

The linear and nonlinear version of this problem will help us understand the behavior of iterative methods that approach the steady state of an associated time-dependent problem. The linear case in particular is useful for spectral analysis of the corresponding finite-difference methods, which will be shown in later sections.

2.2 Parabolic Pseudo-Time Model

Instead of solving (2.1) directly, the boundary value problem may be viewed as the steady state of the parabolic time-evolution problem

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(\nu(u) \frac{\partial u}{\partial x} \right), & 0 < x < 1, \quad t \geq 0, \\ u(x, 0) = 0, & 0 < x < 1, \\ u(0, t) = 0, \quad u(1, t) = 1, & t \geq 0. \end{cases} \quad (2.7)$$

At steady state, the time derivative vanishes, and (2.7) reduces to the original nonlinear boundary value problem (2.1)–(2.2). For the linear case $\nu(u) = \tilde{\nu}$, the parabolic equation becomes the heat equation

$$\frac{\partial u}{\partial t} = \tilde{\nu} \frac{\partial^2 u}{\partial x^2}. \quad (2.8)$$

2.3 Spatial and Temporal Discretization

The one-dimensional domain is discretized using a uniform grid. For a positive integer N , define

$$\Delta x = \frac{1}{N}, \quad x_j = j\Delta x, \quad 0 \leq j \leq N. \quad (2.9)$$

The boundary points are $x_0 = 0$ and $x_N = 1$, while $x_1, x_2, \dots, x_{N-1}$ are the interior unknowns. Thus, there are $N - 1$ interior cells and $N + 1$ total grid cells. Pseudo-time is discretized as

$$t^n = n\Delta t, \quad n = 0, 1, 2, \dots \quad (2.10)$$

For the nonlinear diffusion coefficient, interface values are defined by averaging neighboring solution values before evaluating ν :

$$\nu_{j+\frac{1}{2}} = \nu \left(\frac{u_{j+1}^n + u_j^n}{2} \right), \quad \nu_{j-\frac{1}{2}} = \nu \left(\frac{u_j^n + u_{j-1}^n}{2} \right). \quad (2.11)$$

For the linear problem, these reduce simply to

$$\nu_{j+\frac{1}{2}} = \nu_{j-\frac{1}{2}} = \tilde{\nu}. \quad (2.12)$$

2.4 Parabolic Iterative Methods: Jacobi, Gauss–Seidel, and SOR

Using the interface coefficients defined in Section 2.3, the discrete residual for the non-linear problem is

$$\mathcal{R}_{j,N}^n = \frac{\nu_{j+\frac{1}{2}} (u_{j+1}^n - u_j^n) - \nu_{j-\frac{1}{2}} (u_j^n - u_{j-1}^n)}{\Delta x^2}. \quad (2.13)$$

For the linear problem, where $\nu(u) = \tilde{\nu}$, this becomes

$$\mathcal{R}_{j,L}^n = \frac{\tilde{\nu}}{\Delta x^2} (u_{j+1}^n - 2u_j^n + u_{j-1}^n). \quad (2.14)$$

In what follows, $\mathcal{R}_j$ denotes the residual corresponding to the selected problem.

The Jacobi method evaluates the residual entirely from the previous iterate:

$$u_j^{n+1} = u_j^n + \Delta t_P \mathcal{R}_j^n. \quad (2.15)$$

For Gauss–Seidel, let $\mathcal{R}_j^{GS}$ denote the same residual evaluated using the most recently available values during the sweep. For a left-to-right ordering, u_{j-1}^{n+1} replaces u_{j-1}^n , while u_j^n and u_{j+1}^n remain unchanged. The (unrelaxed) Gauss–Seidel update is therefore

$$u_j^{GS} = u_j^n + \Delta t_P \mathcal{R}_j^{GS}. \quad (2.16)$$

Parabolic successive over-relaxation (SORp) then relaxes toward the Gauss–Seidel value:

$$u_j^{n+1} = u_j^n + \omega_P (u_j^{GS} - u_j^n) = u_j^n + \omega_P \Delta t_P \mathcal{R}_j^{GS}. \quad (2.17)$$

The choice $\omega_P = 1$ recovers Gauss–Seidel, while $0 < \omega_P < 1$ and $\omega_P > 1$ correspond to under-relaxation and over-relaxation, respectively. Jacobi and Gauss–Seidel are retained as unrelaxed performance baselines in Chapter 3. These Jacobi, Gauss–Seidel, and SOR iterations follow the standard stationary matrix-splitting framework [27, 12].

2.5 Hyperbolic Pseudo-Time Model

The main modification studied in this report is to replace the parabolic pseudo-time equation with a hyperbolic pseudo-time formulation:

$$\frac{\partial^2 u}{\partial t \partial x} = \frac{\partial}{\partial x} \left(\nu(u) \frac{\partial u}{\partial x} \right), \quad 0 \leq x \leq 1, \quad t \geq 0. \quad (2.18)$$

The mixed space–time model is adapted from Chattot’s hyperbolic reformulation of the corresponding elliptic model problem [5]. The same boundary conditions and diffusion coefficients are used as in the parabolic problem. At steady state, the time derivative vanishes, and (2.18) again reduces to the original nonlinear boundary value problem defined in Eq. 2.1. For the linear case $\nu(u) = \tilde{\nu}$, the modified equation becomes

$$u_{tx} = \tilde{\nu} u_{xx}, \quad (2.19)$$

which reduces to the corresponding linear steady state solution.

Therefore, the parabolic and hyperbolic pseudo-time formulations are designed to converge to the same steady-state solution, but they approach that solution through very different transient dynamics. The parabolic formulation behaves diffusively: information propagates through smoothing, and the explicit time step is restricted by $\Delta t = O(\Delta x^2)$. The hyperbolic formulation introduces a mixed time-space derivative and changes the dynamics of the continuous pseudo-time evolution from diffusion-based to advection-based. However, their discrete pseudo-time iterations may not exhibit the same pseudo-time behavior of the underlying continuous models.

2.6 SORh: 1D Hyperbolic SOR Formulation

A basic discretization of (2.18) is

$$\frac{u_j^{n+1} - u_{j-1}^{n+1} - u_j^n + u_{j-1}^n}{\Delta t \Delta x} = \frac{\nu_{j+\frac{1}{2}} \left(\frac{u_{j+1}^n - u_j^n}{\Delta x} \right) - \nu_{j-\frac{1}{2}} \left(\frac{u_j^n - u_{j-1}^n}{\Delta x} \right)}{\Delta x}. \quad (2.20)$$

For the constant-coefficient linear case, this becomes

$$\frac{u_j^{n+1} - u_{j-1}^{n+1} - u_j^n + u_{j-1}^n}{\Delta t \Delta x} = \tilde{\nu} \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}. \quad (2.21)$$

Multiplying by $\Delta t \Delta x$ and rearranging gives

$$u_j^{n+1} = u_j^n + (u_{j-1}^{n+1} - u_{j-1}^n) + r_H (u_{j+1}^n - 2u_j^n + u_{j-1}^n), \quad r_H = \frac{\tilde{\nu} \Delta t}{\Delta x}. \quad (2.22)$$

Because (2.22) contains the updated value u_{j-1}^{n+1} , the update has a naturally sequential, GS-type structure, even when the right-hand side is evaluated explicitly in pseudo-time. Similar to SORp, an SOR correction step is introduced, allowing control over the rate of relaxation toward the hyperbolic update. The hyperbolic update obtained from (2.22) is denoted by u_j^H , and the relaxation step is

$$u_j^{n+1} = u_j^n + \omega_H (u_j^H - u_j^n). \quad (2.23)$$

where ω is the relaxation parameter. This leads to the full discrete SORh update

$$u_j^{n+1} = u_j^n + \omega_H \left[(u_{j-1}^{n+1} - u_{j-1}^n) + r_H (u_{j+1}^n - 2u_j^n + u_{j-1}^n) \right], \quad r_H = \frac{\tilde{\nu} \Delta t}{\Delta x}. \quad (2.24)$$

As in SORp, $\omega_H = 1$ recovers the underlying hyperbolic update, while $0 < \omega_H < 1$ corresponds to under-relaxation and $\omega_H > 1$ to over-relaxation. In subsequent sections, we will discuss the behavior of SORh compared to the classical methods, as well as the role and stability limits of ω_H , in analytical and experimental contexts.

2.7 Accuracy, Consistency, and Stability of SORh

For the linear constant-coefficient problem, the truncation error of (2.21) is defined by substituting the exact solution $U_j^n = u(x_j, t^n)$:

$$\text{TE}_H = \frac{U_j^{n+1} - U_{j-1}^{n+1} - U_j^n + U_{j-1}^n}{\Delta t \Delta x} - \tilde{\nu} \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2}. \quad (2.25)$$

A Taylor expansion about (x_j, t^n) gives the formal accuracy of the method. The first term approximates u_{tx} , while the second term approximates $\tilde{\nu} u_{xx}$. Because the mixed derivative is approximated backward in space and forward in time, (2.21) is first-order accurate in both space and time as a transient approximation. At steady state, however, the time-dependent contribution vanishes and the remaining centered second difference has $O(\Delta x^2)$ spatial truncation error for smooth solutions [19].

The stability behavior is more subtle than in the parabolic case. The explicit parabolic discretization has the classical restriction $\Delta t = O(\Delta x^2)$, whereas the hyperbolic formulation introduces the parameter $r_H = \tilde{\nu} \Delta t / \Delta x$. Substituting a Fourier mode of the form $u_j^n = e^{an\Delta t} e^{I\xi j \Delta x}$ with $I = \sqrt{-1}$ into the discrete update equation and requiring the amplification factor to satisfy $|e^{an\Delta t}| \leq 1$ yields the CFL-type stability condition [19]

$$0 \leq \frac{\tilde{\nu} \Delta t}{\Delta x} \leq 1 \quad \Longleftrightarrow \quad 0 \leq \Delta t \leq \frac{\Delta x}{\tilde{\nu}}. \quad (2.26)$$

Thus, unlike the quadratic time-step restriction associated with explicit parabolic methods, the SORh formulation admits a linear CFL condition that scales directly with the spatial resolution. On the surface, this suggests an accelerated rate of relaxation and increased efficiency for SORh. However, on closer inspection, SORh does *not* actually apply a larger correction to the elliptic residual. In SORp, the pseudo-time step multiplies the discrete elliptic residual directly, so the residual contribution to the point update scales as

$$\Delta t_P R_j = O(\Delta x^2) R_j. \quad (2.27)$$

In SORh, there is an extra factor of Δx in the left-hand side of Eq. (2.20), so solving that equation for the point value introduces an additional factor of Δx into the residual factor r_H . Because $\Delta t_H = O(\Delta x)$, the resulting residual contribution also satisfies

$$\Delta t_H \Delta x R_j = O(\Delta x^2) R_j. \quad (2.28)$$

The parabolic and hyperbolic pseudo-time steps therefore cannot be compared directly as measures of solver progress. Their differing scalings are required by their respective continuous pseudo-time models, while the resulting scalar corrections have the same elliptic $O(\Delta x^2)$ scale. Solver efficiency must instead be assessed through the contraction of the discrete error, the number of sweeps, and the computational cost per sweep.

This differs from first-order hyperbolic diffusion formulations such as Nishikawa's [21], where introducing the solution gradients as independent variables produces a genuine $O(\Delta x)$ hyperbolic CFL scale. This distinction is discussed further in Section 4.6.3.

2.8 Connection to AOR and Classical SOR

The discrete linear SORh iteration belongs to the accelerated overrelaxation (AOR) family introduced by Hadjidimos as a two-parameter generalization of SOR [11]. For a linear system $Au = b$ with the splitting $A = D - L - U$, the AOR iteration can be written in correction form, where the correction $\delta^{(n+1)} = u^{(n+1)} - u^{(n)}$ satisfies

$$(D - r_A L) \delta^{(n+1)} = \omega_A (b - Au^{(n)}). \quad (2.29)$$

See Appendix A.1 for a derivation of (2.29). This form clarifies the roles of r_A and ω_A : r_A controls the lower-triangular, Gauss–Seidel-like propagation of already-computed corrections, and ω_A scales the newly introduced residual correction. Classical SOR is recovered when $r_A = \omega_A$. To see this, define the unnormalized residual stencil for the

three-point operator

$$\mathcal{S}_j^{(n)} := u_{j-1}^{(n)} - 2u_j^{(n)} + u_{j+1}^{(n)}. \quad (2.30)$$

The linear SORh update in Section 2.6 can then be written in correction form as

$$\delta_j^{(n+1)} = \omega_H \delta_{j-1}^{(n+1)} + \omega_H r_H \mathcal{S}_j^{(n)}, \quad r_H = \frac{\tilde{\nu} \Delta t_H}{\Delta x}. \quad (2.31)$$

For the same residual stencil $\mathcal{S}_j$, Eq. (2.29) reduces to the same form as Eq. (2.31) with coefficients $r_A/2$ and $\omega_A/2$ instead of ω_H and $\omega_H r_H$, respectively. Coefficient matching therefore yields

$$\boxed{r_A = 2\omega_H, \quad \omega_A = 2r_H \omega_H}. \quad (2.32)$$

Again, see Appendix A.1 for a fuller comparison. AOR supplies an established algebraic framework for the generalized-SOR interpretation of SORh. The distinction for SORh is that it is independently derived from the hyperbolic pseudo-time model introduced in Section 2.5. Additionally, SORh gives the two weights a direct pseudo-time interpretation: for fixed ω_H , the hyperbolic CFL factor r_H changes the amount of residual injection without changing the strength with which an upstream correction is propagated through the sweep. At unit hyperbolic CFL, $r_H = 1$, the two AOR parameters coincide. SORh then reduces exactly to classical SOR under the parameter mapping

$$\alpha = 2\omega_H, \quad (2.33)$$

where α is the classical SOR relaxation parameter. This factor of two comes from the diagonal $2/\Delta x^2$ of the one-dimensional discrete Poisson operator: classical SOR applies $\alpha/2$ to the unnormalized three-point residual, whereas unit-CFL SORh applies ω_H .

For M interior unknowns with Dirichlet boundary conditions, the Jacobi spectral

radius and classical optimal SOR parameter [9, 27] are

$$\rho_J = \cos\left(\frac{\pi}{M+1}\right), \quad \alpha_{\text{opt}} = \frac{2}{1 + \sqrt{1 - \rho_J^2}} = \frac{2}{1 + \sin\left(\frac{\pi}{M+1}\right)}. \quad (2.34)$$

Combining this result with Eq. (2.33) gives the optimum

$$\omega_{H,\text{opt}} = \frac{1}{1 + \sin\left(\frac{\pi}{M+1}\right)}. \quad (2.35)$$

This formula is exact for the constant-coefficient linear problem at $r_H = 1$. When $r_H < 1$, the mapping Eq. (2.33) is not exact, so the iteration is no longer classical SOR and the classical optimum is only an analytical reference. The numerical results in Chapter 3 test how well that reference continues to predict the optimal ω_H .

For comparison, the SORp correction has equal propagation and residual coefficients. Its AOR parameters therefore satisfy

$$r_A = \omega_A = 2r_P\omega_P = \text{CFL}_P\omega_P, \quad (2.36)$$

so SORp remains classical SOR, with effective parameter $\alpha = \text{CFL}_P\omega_P$, for any chosen parabolic CFL number. In the notation used by the numerical experiments, $M = N - 1$, so the parameter estimates are implemented as

$$\omega_{H,\text{opt}} = \frac{1}{1 + \sin(\pi/N)}, \quad \omega_{P,\text{opt}} = \frac{2}{\text{CFL}_P [1 + \sin(\pi/N)]}. \quad (2.37)$$

2.9 Spectral Analysis of the Iterative Methods

The convergence of the iterative methods considered in this work is governed by the spectral properties of the associated iteration matrices. Writing the update in the form

$$U^{(k+1)} = GU^{(k)} + c, \quad (2.38)$$

the error satisfies

$$e^{(k+1)} = Ge^{(k)}, \quad (2.39)$$

so convergence requires $\rho(G) < 1$, where $\rho(G)$ denotes the spectral radius [27].

2.9.1 Comparison of Spectral Behavior

The hyperbolic and parabolic SOR methods derived in the previous sections both admit iteration matrices of the form

$$G = \omega L^{-1} R, \quad (2.40)$$

where the specific forms of L_H , R_H (SORh) and L_P , R_P (SORp) are

$$\begin{cases} L_H = \text{tridiag}(-\omega, 1, 0), & R_H = \text{tridiag}(r_H - 1, \frac{1}{\omega} - 2r_H, r_H). \\ L_P = \text{tridiag}(-\omega r_P, 1, 0), & R_P = \text{tridiag}(0, \frac{1}{\omega} - 2r_P, r_P). \end{cases} \quad (2.41)$$

The entries of L_H and L_P depend differently on the relaxation and CFL parameters. In SORh, the lower-triangular dependence represents the explicit transmission of the previously computed upstream correction. In SORp, the corresponding dependence arises through the Gauss–Seidel evaluation of the parabolic residual. These algebraic differences do not provide evidence of intrinsically stronger coupling in either formulation. This claim is supported by the mappings to classical SOR discussed in the previous section,

$$\alpha = 2\omega_H, \quad \alpha = \omega_P \text{CFL}_P. \quad (2.42)$$

with SORh specifically requiring $\text{CFL}_H = 1$. The two methods are closely related stationary iterations for the constant-coefficient linear problem. When $\text{CFL}_H < 1$, SORh no longer has the classical balance between residual injection and upstream correction propagation, but the classical optimal parameter remains a useful analytical reference. To compare the convergence properties of the two methods, we examine the spectral radius $\rho(G)$ as a function of the relaxation parameter ω . Figure 2.1 shows $\rho(G)$ for both

formulations, computed for a one-dimensional problem with $N_{\text{int}} = 64$, i.e. 64 interior grid cells, and $\text{CFL} = 0.8$.

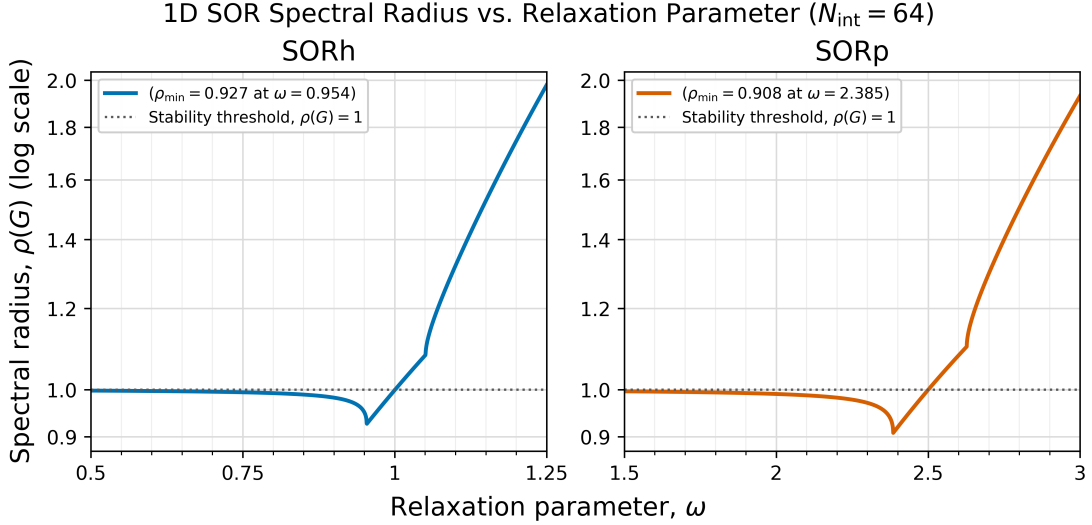


Figure 2.1: Spectral radius $\rho(G)$ as a function of the relaxation parameter ω for the parabolic and hyperbolic SOR methods, with $N_{\text{int}} = 64$.

Table 2.1: Spectral-radius comparison for the one-dimensional SOR formulations at $N_{\text{int}} = 64$ and $\text{CFL} = 0.8$. Here, ω_{opt} is obtained from numerically calculated ρ_{min} and ω_{pred} are calculated using Eq. 2.37.

Method	ρ_{min}	ω_{opt}	ω_{pred}
SORh	0.9263	0.9539	0.9539
SORp	0.9078	2.3848	2.3848

The resulting figure supports the analysis. Indeed, the shape of the $\rho(G)$ vs. ω curve is very similar between SORh and SORp. The only major difference is the range of ω for which the stability condition $\rho(G) < 1$ is satisfied. SORp remains stable over a wide range extending to $\omega_P = 2.5$. This is larger than the classical SOR restriction $\omega < 2.0$ because of the dependence of ω_P on the CFL number. In particular, for $\text{CFL} = 0.8$, $\omega_{P,\text{max}} = \frac{2.0}{0.8} = 2.5$. The hyperbolic method is stable only up to $\omega_H = 1.0$. These stability limits are consistent with the optimal omega formulas and classical-SOR scaling factors derived in Section 2.8. Also, the values ω_{pred} and ω_{opt} agree to machine precision, further supporting the accuracy of the optimal- ω predictions in Section 2.8, at least for

the specific resolution $N_{\text{int}} = 64$.

The parabolic method attains a slightly smaller minimum spectral radius ($\rho_{P,\min} \approx 0.908$ vs $\rho_{H,\min} \approx 0.926$), suggesting slightly faster asymptotic convergence per iteration. Interestingly, both formulations exhibit strong sensitivity to the relaxation parameter near their respective optimal values of ω . In each case, the spectral radius decreases sharply as ω approaches the optimum and increases rapidly for $\omega > \omega_{\text{opt}}$. The steep change in convergence rate near the optimum is consistent with the classical parameter sensitivity of SOR [27, 12].

2.10 Numerical Diagnostics and Transition to Results

The numerical studies use a common set of root-mean-square (RMS) diagnostics. For a grid function v defined on the set of interior grid points $\mathcal{I}_h$, with $N_I = |\mathcal{I}_h|$, define

$$\|v\|_{\text{RMS},\mathcal{I}_h} := \frac{1}{\sqrt{N_I}} \|v\|_{2,\mathcal{I}_h} = \left(\frac{1}{N_I} \sum_{i \in \mathcal{I}_h} |v_i|^2 \right)^{1/2}. \quad (2.43)$$

Three diagnostic names are used throughout the thesis:

1. The update norm

$$e_u^{(n+1)} := \left\| u^{(n+1)} - u^{(n)} \right\|_{\text{RMS},\mathcal{I}_h}, \quad (2.44)$$

which measures the change between successive pseudo-time iterates. It is a convenient stopping and stagnation measure, but a small e_u does not by itself establish that the discrete equation has been satisfied.

2. The residual norm

$$e_r^{(n+1)} := \left\| \mathcal{R}_h \left(u^{(n+1)} \right) \right\|_{\text{RMS},\mathcal{I}_h} \quad (2.45)$$

which directly measures how well the discrete equation is satisfied. It is generally always available and is the preferred general-purpose convergence diagnostic.

3. An exact-solution error or diagnostic, denoted e_s . This quantity tests accuracy

against known solution information and is generally used for validation of numerical accuracy and convergence. For the one-dimensional linear problem, the exact solution is $u(x) = x$, giving

$$e_{s,L}^{(n)} := \left\| u^{(n)} - x \right\|_{\text{RMS}, \mathcal{I}_h}. \quad (2.46)$$

For the nonlinear problem, the exact solution is represented implicitly by the relation derived in Section 2.1. Its corresponding diagnostic is

$$e_{s,NL}^{(n)} := \left\| u^{(n)} - \ln(1 + u^{(n)}) - (1 - \ln 2)x \right\|_{\text{RMS}, \mathcal{I}_h}. \quad (2.47)$$

Unlike (2.46), this is the error in an implicit exact-solution relation rather than a direct pointwise difference from an explicitly evaluated u_{exact} . It nevertheless vanishes when the numerical solution satisfies that relation.

A one-dimensional run is declared converged when $e_u^{(n+1)} \leq \varepsilon_u$, where ε_u is the prescribed update tolerance. A non-finite or excessively large update terminates the run as unstable, and reaching the iteration limit is treated as non-convergence. The exact-solution diagnostic e_s is not used to terminate the reported performance runs.

Within this framework, Chapter 3 tests the stability and parameter dependence predicted by the preceding analysis, compares SORh with SORp, and relates the measured iteration counts and errors to the spectral results. The numerical results in Chapter 3 will determine which distinctions are consequential in computation.

Chapter 3

Results of One-Dimensional Experiments

In this chapter, we show the results of experiments using the one-dimensional iterative methods described in Chapter 2. We will show the convergence behavior and computational performance of SORh compared to SORp and other classical parabolic relaxation methods, as well as how these results differ between the linear and nonlinear problems introduced in Section 2.1.

3.1 Experimental Setup

Experiments were conducted on the linear and nonlinear problems using four methods for comparison: Parabolic Jacobi, Parabolic Gauss-Seidel (GS), Parabolic SOR (SORp), and Hyperbolic SOR (SORh). We first define simulation parameters in Table 3.1 which will be used for all one-dimensional simulations unless otherwise noted.

For the linear problem, Δt remains fixed since $\nu = \nu_L$ is constant. For the nonlinear problem, the initial diffusion coefficient is $\nu_N = 0.5$, while the face-centered diffusion coefficients ν_N^+ and ν_N^- evolve throughout the iteration. Whenever a larger

Quantity	Value
CFL number	$C_a = 0.8$
Linear diffusion coefficient	$\nu_L = 0.5$
Initial nonlinear diffusion coefficient	$\nu_N = 0.5$
Parabolic pseudo-time step	$\Delta t_p = \text{CFL } \Delta x^2 / (2\nu_{\max})$
Hyperbolic pseudo-time step	$\Delta t_h = \text{CFL } \Delta x / \nu_{\max}$
Convergence tolerance	$\varepsilon_u = 10^{-7}$, with $e_u^{(n)} < \varepsilon_u$

Table 3.1: Coefficients and parameters used in the 1D experiments.

local diffusion coefficient is encountered, $\nu_{\max}$ is updated according to

$$\nu_{\max} \leftarrow \max(\nu_{\max}, |\nu_N^+|, |\nu_N^-|), \quad (3.1)$$

after which Δt is recomputed using the appropriate stability condition. However, for the particular nonlinear diffusion coefficient used in these experiments, the face values satisfy $|\nu_N^+|, |\nu_N^-| \leq 0.5$ throughout the iteration. Consequently, $\nu_{\max}$ remains equal to its initial value, $\nu_N = 0.5$, and Δt remains fixed.

3.2 Linear Performance Comparison

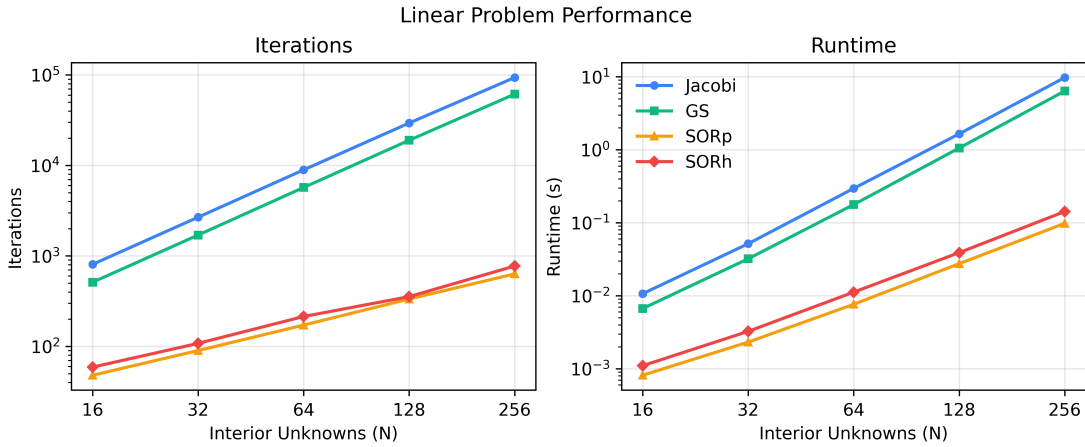


Figure 3.1: Iteration count and runtime for the linear problem as the number of interior unknowns is refined. SORp and SORh use experimentally optimized relaxation parameters.

First we compare the performance of all four methods on the linear problem. Results are shown in Figure 3.1. Jacobi, GS, and SORp perform as expected. Jacobi is the slowest, taking about 10^5 relaxation steps to complete, and GS is only slightly faster. SORp is much more efficient, taking about 10^3 iterations and improving on Jacobi and GS by roughly two orders of magnitude.

The results for SORh show that though it is much faster than Jacobi and GS, it is still slightly outperformed by SORp. Due to the simplicity of the linear problem, however, this is not unexpected. The optimal relaxation parameters for both methods are derived from the same underlying spectral structure. More precisely, with the linear steady-state solution $u(x) = x$, constant $\nu(u)$, and no further nonlinear coupling, the separation of SOR correction propagation and residual injection that is characteristic of SORh has very little effect on the speed of error correction. The strength of residual injection is constant throughout the iteration for both SORh and SORp.

Two crucial differences can help explain the slight performance advantage for SORp. First, we can look at differences in per-cell update cost. SORh’s per-cell update formula was found to cost about 20% more on average than SORp. The results show that SORh takes about 42% longer on average in total runtime than SORp, but with only 19–22% more iterations than SORp. Thus, since there are no other significant differences in the computational cost per iteration step, some of the performance discrepancy can be attributed to this discrete update cost difference. The second explanation is the differing spectral radii from Table 2.1,

$$\rho_{P,\min} \approx 0.908, \quad \rho_{H,\min} \approx 0.926, \quad (3.2)$$

suggesting a slightly faster rate of convergence for SORp. This is because, for SORh, $\text{CFL}_H < 1$ causes residual contribution to be reduced relative to upstream correction propagation. This an imbalance that cannot be restored by changing ω_H . SORp retains this balance even for $\text{CFL}_P < 1$ because it is still equivalent to classical SOR.

3.3 Nonlinear Performance Comparison

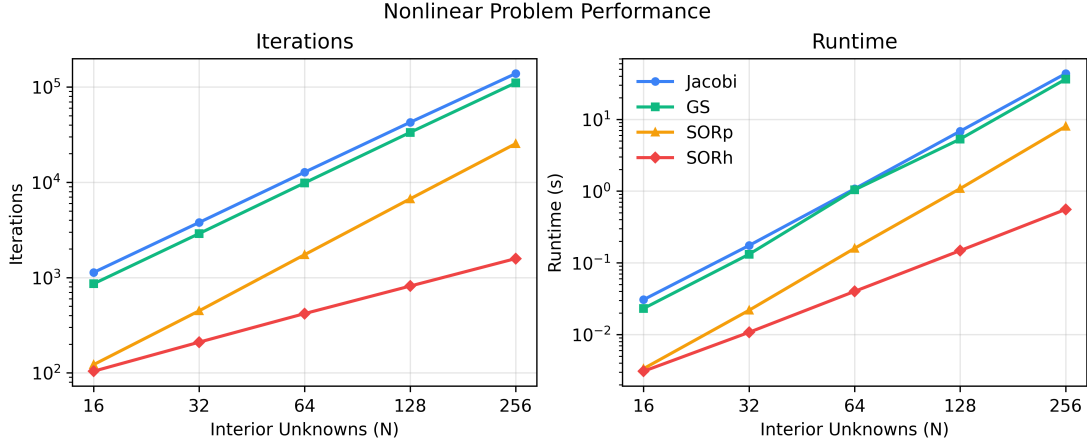


Figure 3.2: Iteration count and runtime for the nonlinear problem using the same convergence tolerance and grid sequence as in the linear comparison.

Unlike the linear problem, the nonlinear problem produces a substantial performance separation between SORp and SORh. Over the tested grid sequence, the SORh iteration count grows by approximately a factor of two with each refinement, while the SORp iteration count grows by approximately a factor of four. SORh therefore retains growth close to the $O(N)$ behavior expected from an optimally relaxed one-dimensional iteration [9, 27], whereas SORp exhibits growth closer to $O(N^2)$. The SORp iteration count remains lower than those of Jacobi and Gauss–Seidel at each individual resolution, but its empirical growth rate is steeper over the tested range.

This difference cannot be explained by a poor choice of relaxation parameter. The values of ω used in the nonlinear performance comparison were optimized independently for the nonlinear problem. Nevertheless, no single value of ω_P produces the same uniform acceleration observed for the constant-coefficient problem. The linear spectral analysis assumes a fixed iteration matrix, while the nonlinear coefficient $\nu(u)$ varies spatially and changes as the solution evolves. Consequently, the local error-propagation operator also changes during the iteration, and the classical linear optimal- ω analysis no longer applies directly. The particular nonlinear problem considered here becomes

degenerate at the left boundary because

$$\nu(u) = \frac{u}{1+u} \quad (3.3)$$

vanishes when $u = 0$. The parabolic time step is selected using the maximum diffusion coefficient, but the strength of the local SORp correction depends on the local face coefficients. If

$$\delta_j^{n+1} = u_j^{n+1} - u_j^n \quad (3.4)$$

denotes the correction generated during a sweep, the SORp correction can be written exactly as

$$\delta_j^{n+1} = \omega_P \Delta t_P R_j^{\text{GS}}. \quad (3.5)$$

However, to expose the propagation of an upstream correction, suppose temporarily that the nonlinear face coefficient is frozen at its old-iterate value. Then

$$\delta_j^{n+1} \approx \omega_P \left[\Delta t_P R_j^n + \frac{\Delta t_P \nu_{j-\frac{1}{2}}^n}{\Delta x^2} \delta_{j-1}^{n+1} \right], \quad (3.6)$$

with additional terms arising from the change in $\nu_{j-\frac{1}{2}}$ during the nonlinear Gauss–Seidel sweep. Thus, both residual relaxation and upstream correction propagation depend on the local diffusion coefficient. By contrast, the implemented SORh correction can be written exactly as

$$\delta_j^{n+1} = \omega_H \left[\delta_{j-1}^{n+1} + \Delta t_H \Delta x R_j^n \right]. \quad (3.7)$$

The residual contribution continues to depend on the nonlinear diffusion coefficient through $\mathcal{R}_j^n$, but the upstream correction-propagation term is not multiplied by the local value of ν . SORh can therefore continue transmitting previously generated corrections through regions of weak diffusion. This structural difference is largely inactive for the constant-coefficient linear problem, where the two methods reduce to closely related SOR iterations, but becomes important when the coefficient is strongly nonuniform.

Grid refinement further resolves the region near $u = 0$, where increasingly small face coefficients occur. This provides a plausible explanation for why the performance separation grows with resolution: the locally coefficient-weighted propagation in SORp becomes progressively less effective, while the explicit correction-propagation term in SORh remains intact. Although the SORh point update is somewhat more expensive, its substantially smaller iteration count more than compensates for that cost on the nonlinear problem.

3.3.1 Solution Accuracy & Summary

Table 3.2: Representative final diagnostics for the optimized SOR methods at $N_{\text{int}} = 32$ and $\varepsilon_u = 10^{-7}$.

Problem	Final update, e_u		Exact-solution diagnostic, e_s	
	SORp	SORh	SORp	SORh
Linear	9.40×10^{-8}	9.07×10^{-8}	1.93×10^{-7}	2.12×10^{-7}
Nonlinear	9.96×10^{-8}	8.70×10^{-8}	1.230×10^{-4}	1.223×10^{-4}

Table 3.2 shows accuracy diagnostics for SORh and SORp on the linear and nonlinear problems. Their linear exact-solution errors are both approximately 2×10^{-7} , while their nonlinear diagnostics differ by only 0.57%. This agreement between methods supports the conclusion that they both approach consistent steady-state solutions. This is strong evidence that the observed performance difference must concern the path by which corrections propagate through the domain and do not involve an accuracy trade-off. Overall, the results are consistent with the interpretation of SORh as an AOR or generalized classical SOR iteration. The separation of residual relaxation from upstream correction propagation provides additional robustness for this nonlinear problem.

3.4 Behavior of the Relaxation Parameter

Figures 3.3 and 3.4 provide two complementary views of the relaxation-parameter behavior on the nonlinear problem. Figure 3.3 shows that the linear spectral prediction

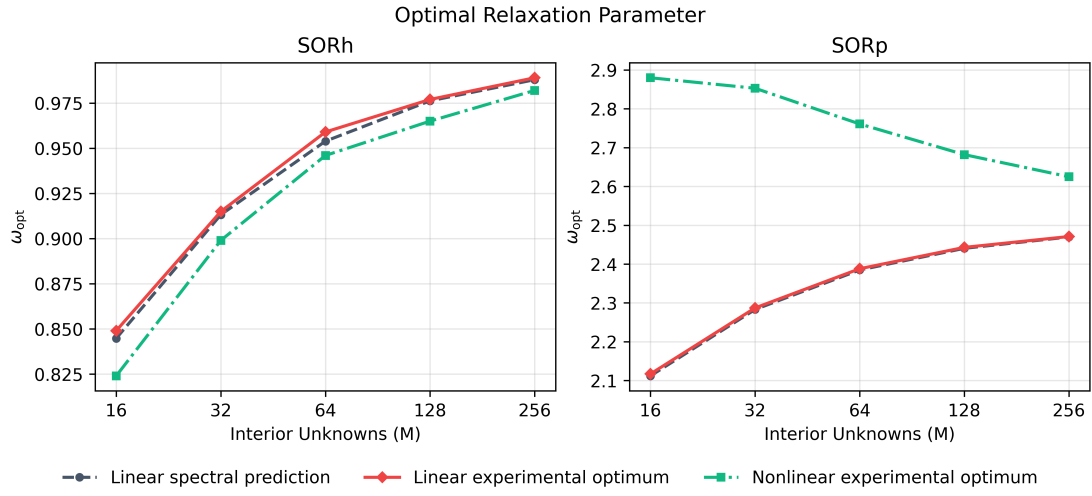


Figure 3.3: Predicted and experimentally measured optimal relaxation parameters for SORh and SORp on the linear and nonlinear problem.

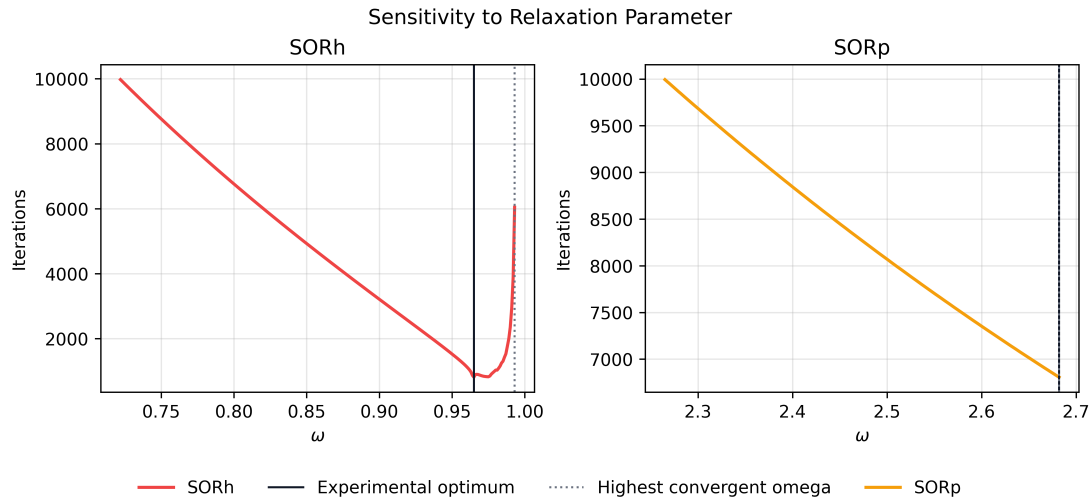


Figure 3.4: Sensitivity of nonlinear convergence to the relaxation parameter at fixed grid resolution, $N_{\text{int}} = 128$. Vertical markers indicate the experimental optimum and highest convergent relaxation parameter.

remains a useful approximation for the optimal SORh parameter, though there are small differences because the nonlinear coefficient changes throughout the iteration. In contrast, the nonlinear SORp optimum departs substantially from the linear prediction, substantially overshooting its maximum predicted stable parameter, $\omega_{P,\max} = 2.5$.

Figure 3.4 clarifies the nature of this departure. SORh exhibits a well-defined minimum iteration count within its stable parameter range. Its performance improves as ω_H approaches this minimum and then deteriorates before instability occurs. SORp, however, continues to improve as ω_P is increased, and its lowest observed iteration count occurs at the edge of the stable range. Its usable optimum is therefore constrained by instability rather than determined by a clear interior minimum.

Both results are consistent with the influence of the degenerate diffusion coefficient. In regions where $\nu(u)$ is small, the coefficient-weighted SORp correction becomes weak, so increasing ω_P partially compensates for the loss of local correction strength. The same global relaxation parameter also acts in regions where $\nu(u)$ is larger, however, and these regions ultimately limit stability. SORp therefore cannot simultaneously provide sufficient relaxation in the weak-diffusion region and remain far from instability elsewhere. By contrast, the upstream correction-propagation term in SORh is not multiplied by the local diffusion coefficient. Its effective relaxation behavior consequently remains closer to the constant-coefficient prediction and produces a stable, well-resolved optimum. Together with the performance results, these figures show that SORh is both more predictable and more effectively tunable on the nonlinear problem.

Chapter 4

Two-Dimensional Formulation, Methods, and Analysis

In Chapter 2, we introduced one-dimensional relaxation methods to solve simple elliptic boundary value problems. In this chapter, we will extend these methods to two spatial dimensions, which will introduce new pseudo-time dynamics and allow us to apply the core ideas behind the one-dimensional methods to more complex and physically relevant problems.

As we will see, extending SORh from a one-dimensional to two-dimensional formulation is not straightforward, as it is for SORp. For this reason, instead of immediately describing the form of 2D SORh used in numerical experiments, we will first take a close look at analytical properties, discretization strategies, and other modifications to hyperbolic-based methods that were considered. This exploration is by no means exhaustive, but it allows us to gain an understanding of the properties of hyperbolic pseudo-time methods and their behavior under a variety of conditions. Most importantly, it will provide a basis to show why SORh formulation used in numerical experiments was ultimately chosen.

4.1 Governing Elliptic Problems

We start by introducing the two primary elliptic problems that we are interested in solving. The first is Poisson’s equation and the application to self-gravity. The second is a nonlinear diffusion problem, based on the one-dimensional “nonlinear problem” from Chapter 2, now extended to the unit square.

4.1.1 Poisson’s Equation

Poisson’s equation can be expressed simply as

$$\Delta\phi = f, \quad f = 4\pi G\rho. \quad (4.1)$$

where ϕ is the gravitational potential, ρ is the mass density, and G is the gravitational constant. The Laplacian of the potential ϕ is given as

$$\Delta\phi \equiv \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} \equiv \phi_{xx} + \phi_{yy}. \quad (4.2)$$

Poisson’s equation is a powerful model with a wide range of applications in physics, such as heat conduction, electrostatics, and fluid dynamics. Our particular application of interest is self-gravitation, which describes the gravitational force exerted by a system onto itself. The effect of gravitational forces depends on the total mass of the system and its density distribution. This is often applied to celestial bodies, such as the inward gravitational force felt by a planet, star, or a system of celestial objects.

The gravitational potential ϕ depends only on the forcing term f on the right-hand side of (4.1), which involves the density distribution ρ . However, hydrodynamical systems such as the Euler equations usually require the gravitational acceleration $\mathbf{g}$, which is computed from ϕ using the relation

$$\mathbf{g} = -\nabla\phi. \quad (4.3)$$

Hence, in order to determine $\mathbf{g}$, a numerical solver must first solve Poisson's equation for ϕ , then solve for $\mathbf{g}$ using (4.3). There are a variety of strategies used by hydrodynamical solvers to compute $\mathbf{g}$, and introducing it as a source term must be done carefully so that overall performance and solver accuracy is not hindered [15]. In FLASH experiments involving the Euler equations (Chapter 5), this process is used to couple the Poisson solver to the hydrodynamical solver. However, the main focus of numerical experiments will still be on the numerical Poisson solvers discussed later in this chapter.

4.1.2 Nonlinear Diffusion Problem

In addition to the linear Poisson problem, we consider a nonlinear variable-coefficient diffusion equation. To distinguish its scalar unknown from the gravitational potential ϕ , the nonlinear solution is denoted by u . The governing equation is

$$\nabla \cdot (v(u) \nabla u) = 0, \quad (x, y) \in \Omega = (0, 1)^2, \quad (4.4)$$

where

$$v(u) = \frac{u}{1 + u}. \quad (4.5)$$

Dirichlet boundary conditions are prescribed on the entire boundary according to

$$u(x, y) = x, \quad (x, y) \in \partial\Omega. \quad (4.6)$$

For the expected solution range $0 \leq u \leq 1$, the diffusion coefficient satisfies

$$0 \leq v(u) \leq \frac{1}{2}. \quad (4.7)$$

The coefficient depends on the unknown solution and vanishes at $u = 0$. Consequently, Eq. (4.4) is nonlinear and is not uniformly elliptic over the full solution range. It is therefore referred to here as a *degenerate nonlinear elliptic diffusion problem*. The degeneracy produces regions of weak local diffusion and provides a useful test of how the

different relaxation methods propagate corrections when the elliptic coupling becomes strongly nonuniform.

4.2 Two-Dimensional Pseudo-Time Setting

Similar to the 1D methods, we consider pseudo-time methods of the form

$$\frac{\partial}{\partial t} (L(\phi)) = \Delta\phi - f \quad (4.8)$$

where the spatial operator $L(\cdot)$ varies depending on the specific numerical method used. The choice of L plays an important role in pseudo-time dynamics and will be our main focus in building specific parabolic and hyperbolic pseudo-time models. Most of the proceeding analysis in this chapter will be based on Eq. (4.8), which uses the linear Poisson residual and pseudo-time structure. The resulting discrete method then marches forward in time until effectively

$$\frac{\partial}{\partial t} (L(\phi)) = 0, \quad (4.9)$$

up to some numerical precision, so that the original elliptic steady-state is satisfied. Details of iteration logic and stopping conditions will be discussed in later sections.

4.2.1 Parabolic vs. Hyperbolic Pseudo-time Models

We consider two classes of pseudo-time models derived from Eq. (4.8). Both are constructed so that their steady states satisfy the linear Poisson model ($\Delta\phi = f$), but they approach this steady state through fundamentally different pseudo-time dynamics.

Parabolic pseudo-time dynamics. The first model is the standard parabolic relaxation

$$\phi_t = \Delta\phi - f. \quad (4.10)$$

At a steady state, $\phi_t = 0$, and Eq. (4.10) clearly reduces to Poisson's equation. This makes the parabolic formulation a natural and widely used pseudo-time model for elliptic problems. To explain the dynamics of (4.10), we can denote ϕ^s as the steady elliptic solution and describe the pseudo-time error as

$$e(\mathbf{x}, t) = \phi(\mathbf{x}, t) - \phi^s(\mathbf{x}) \quad \Rightarrow \quad e_t = \Delta e \quad (4.11)$$

Thus, the pseudo-time error evolves according to the diffusion equation. A Fourier-mode analysis shows that high-frequency error is damped rapidly, whereas smooth, long-wavelength error decays much more slowly. This smoothing behavior is characteristic of stationary relaxation methods and is the central motivation for multigrid coarse-grid correction [4].

Hyperbolic pseudo-time dynamics. The one-dimensional hyperbolic model introduced previously, written for Poisson's equation, is

$$\phi_{tx} = \phi_{xx} - f. \quad (4.12)$$

More generally, it belongs to the class

$$\frac{\partial}{\partial t} (L(\phi)) = R(\phi) - f, \quad (4.13)$$

where L is a spatial differential operator applied to the pseudo-time derivative and R is the target elliptic operator. A steady state again satisfies

$$R(\phi) - f = 0,$$

provided that $\partial_t L(\phi) \rightarrow 0$. In one dimension, choosing

$$L(\phi) = \phi_x, \quad R(\phi) = \phi_{xx},$$

recovers Eq. (4.12). Integrating once in space gives

$$\phi_t = \phi_x - H(x) + C(t), \quad H'(x) = f(x), \quad (4.14)$$

where the spatially uniform integration term $C(t)$ is determined by the boundary treatment or normalization. For the corresponding homogeneous error equation, this reduces to the transport form

$$e_t = e_x, \quad (4.15)$$

up to the analogous boundary-determined integration term. Unlike the parabolic model, the hyperbolic formulation does not primarily eliminate error through diffusion. Instead, it transports error and correction information in a preferred direction. On a bounded domain with appropriate inflow boundary data, errors can be carried toward and out of the domain. In the discrete method, an upwind stencil and a compatible sweep ordering reproduce this directional propagation: values updated earlier in a sweep transmit their corrections to downstream points.

These dynamics exist regardless of dimensionality, but they gain an extra degree of freedom when a second spatial dimension is introduced. The parabolic Laplacian has no preferred error propagation direction, but a hyperbolic transport model requires the selection of characteristic directions, upwind neighbors, and compatible sweep orderings. Information can theoretically be propagated in any direction in the 2D plane. Accounting for this directional structure without changing the desired elliptic steady state is the primary difficulty in extending SORh to two dimensions, and it is the focus of the following section.

4.3 Two-Dimensional Hyperbolic Models

Starting from the general hyperbolic model in Eq. (4.15), this section explores two strategies for extending its directional structure to two dimensions before introducing

the formulation used by the discrete SORh methods.

4.3.1 Coupled Advection Model

A natural first approach is to use the two-dimensional advection equation

$$u_t = au_x + bu_y \quad (4.16)$$

as a starting point. Differentiating separately with respect to x and y gives the coupled system

$$\begin{aligned} u_{tx} &= au_{xx} + bu_{xy}, \\ u_{ty} &= au_{xy} + bu_{yy}. \end{aligned} \quad (4.17)$$

This preserves the familiar advection-like structure of the one-dimensional model and shows how mixed derivatives arise when a scalar transport equation is differentiated in multiple spatial directions. This model was ultimately not selected because the resulting gradient-transport equations do not relax the gradient components toward derivatives of a common Poisson solution, such as in Nishikawa's first-order hyperbolic-system approach [21]. Their directional advection structure does not combine naturally into the Laplacian $u_{xx} + u_{yy}$. In addition, evolving two coupled derivative equations would complicate both the formulation and its numerical implementation.

4.3.2 Directional Transport Models

The one-dimensional spatial operator ∂_x can instead be generalized to a directional derivative. Let $\mathbf{n} = (n_1, n_2)$ be a constant unit vector and define

$$\partial_{\mathbf{n}} := \mathbf{n} \cdot \nabla = n_1 \partial_x + n_2 \partial_y. \quad (4.18)$$

Applying this operator twice gives

$$(\partial_{\mathbf{n}})^2 = n_1^2 \partial_{xx} + 2n_1 n_2 \partial_{xy} + n_2^2 \partial_{yy}, \quad (4.19)$$

and the corresponding single-direction model would be

$$\partial_t \partial_{\mathbf{n}} \phi = (\partial_{\mathbf{n}})^2 \phi - f(x, y). \quad (4.20)$$

The operator on the right of Eq. (4.20) contains only the second derivative along $\mathbf{n}$; equivalently, it is a rank-one directional diffusion operator. No single choice of $\mathbf{n}$ can reproduce the full two-dimensional Laplacian without reducing the dimensionality of the model. A more flexible construction can instead use two constant directional derivatives, $\partial_{\mathbf{n}}$ and $\partial_{\mathbf{m}}$:

$$\partial_t (\partial_{\mathbf{n}} \phi + \partial_{\mathbf{m}} \phi) = (\partial_{\mathbf{n}})^2 \phi + (\partial_{\mathbf{m}})^2 \phi - f. \quad (4.21)$$

For the right-hand side to equal $\Delta \phi - f$, the two direction vectors must form an orthonormal basis of $\mathbb{R}^2$ [6]. Under this condition,

$$(\partial_{\mathbf{n}})^2 + (\partial_{\mathbf{m}})^2 = \Delta.$$

The left-hand side can be expressed using the single net direction

$$\mathbf{s} := \mathbf{n} + \mathbf{m}, \quad \partial_{\mathbf{s}} = \partial_{\mathbf{n}} + \partial_{\mathbf{m}},$$

so Eq. (4.21) becomes

$$\partial_t \partial_{\mathbf{s}} \phi = \Delta \phi - f. \quad (4.22)$$

For an orthonormal pair, $\|\mathbf{s}\| = \sqrt{2}$; this constant normalization can be absorbed into the pseudo-time scale. This form makes explicit that the orthonormal pair preserves the isotropic Laplacian on the right while selecting one net pseudo-time transport direction on the left. Using the canonical basis $\mathbf{n} = (1, 0)$ and $\mathbf{m} = (0, 1)$ gives $\mathbf{s} = (1, 1)$ and reduces Eq. (4.22) to

$$\phi_{tx} + \phi_{ty} = \Delta \phi - f. \quad (4.23)$$

This is the model selected as the starting point for the discrete hyperbolic methods in

the following sections. The canonical basis is biased toward the positive (x, y) grid-diagonal direction, but $\mathbf{s} = \mathbf{n} + \mathbf{m}$ could be constructed in a way that rotates the constant basis and changes the directional bias. This freedom is intrinsic to the transport construction and provides an additional degree of freedom. The bias could change, however, due to implementation decisions like sweep orderings, and other methods can produce related directional effects. Numerical experiments suggest that bases chosen along an intermediate direction (far from any major axes) result in the most stable and well-performing methods, as they balance the contributions from both spatial directions.

This continuous model determines the direction in which correction information is transported. After a discretization and compatible sweep are chosen, the directional bias of $\mathbf{s}$ dictates the propagation direction of upstream SOR corrections. Sections 4.5–4.7 will define the resulting discrete pseudo-time update structures.

4.4 Common Discrete Framework

Let

$$\Omega = [x_a, x_b] \times [y_a, y_b], \quad (4.24)$$

and introduce the grid

$$\begin{cases} x_i = x_a + i\Delta x, & i = 0, \dots, N_x + 1, \\ y_j = y_a + j\Delta y, & j = 0, \dots, N_y + 1. \end{cases} \quad (4.25)$$

where

$$\Delta x = \frac{x_b - x_a}{N_x + 1}, \quad \Delta y = \frac{y_b - y_a}{N_y + 1}.$$

The target unknown $\phi_{i,j}^{n+1}$ approximates $\phi(x_i, y_j, t^{n+1})$, where n is the pseudo-time iteration index. The set of interior unknowns is

$$\mathcal{I}_h = \{1, \dots, N_x\} \times \{1, \dots, N_y\}, \quad (4.26)$$

and therefore contains $N_I = N_x N_y$ grid points. Boundary values are prescribed on the fixed outer layers, and the stored array includes g boundary or ghost layers on each side, giving total array dimensions $(N_x + 2g) \times (N_y + 2g)$. The stencil determines the required value of g . For a 5-point centered Laplacian and, for hyperbolic methods, first-order one-sided differences on the left-hand side, $g = 1$ is sufficient. Each complete solver sweep advances the solution from $\phi_{i,j}^n$ to $\phi_{i,j}^{n+1}$, with

$$t^{n+1} = t^n + \Delta t^n. \quad (4.27)$$

The interior solution is initialized with as $\phi^0 = \mathbf{0}$. All test configurations in Chapter 5 will use Dirichlet boundary conditions, which are applied at initialization and maintained throughout the iteration.

4.4.1 Discrete Elliptic Operators and Residuals

Poisson's Equation

Unless otherwise specified, all methods solving the linear Poisson equation will use the five-point centered Laplacian

$$(\Delta\phi)_{i,j} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{\Delta x^2} + \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{\Delta y^2}, \quad (4.28)$$

and the pointwise discrete residual

$$R_{i,j}^k = (\Delta\phi^k)_{i,j} - f_{i,j}. \quad (4.29)$$

This choice of Laplacian discretization is reasonable because it is the natural extension of the one-dimensional three-point Laplacian and is computationally cheap. Of course, larger or higher-order discretizations are possible, but they require extra care when applied to the hyperbolic pseudo-time framework.

Nonlinear Diffusion Problem

The nonlinear operator is discretized in conservative flux form. The diffusion coefficient at a cell face is evaluated using the arithmetic average of the neighboring solution values:

$$v_{i+\frac{1}{2},j} = v\left(\frac{u_{i+1,j} + u_{i,j}}{2}\right), \quad v_{i,j+\frac{1}{2}} = v\left(\frac{u_{i,j+1} + u_{i,j}}{2}\right), \quad (4.30)$$

with analogous definitions at the left and bottom faces. The pointwise discrete nonlinear residual is then

$$\begin{aligned} R_{i,j}^{\text{NL}}(u) = & \frac{v_{i+\frac{1}{2},j}(u_{i+1,j} - u_{i,j}) - v_{i-\frac{1}{2},j}(u_{i,j} - u_{i-1,j})}{h_x^2} \\ & + \frac{v_{i,j+\frac{1}{2}}(u_{i,j+1} - u_{i,j}) - v_{i,j-\frac{1}{2}}(u_{i,j} - u_{i,j-1})}{h_y^2}. \end{aligned} \quad (4.31)$$

This has the same five-point stencil topology as the centered Poisson operator, but its coefficients vary spatially and change as the solution evolves. In the nonlinear experiments, $R_{i,j}^{\text{NL}}$ replaces the linear Poisson residual in both the SORh and SORp updates. Convergence therefore requires

$$R_{i,j}^{\text{NL}}(u) = 0 \quad (4.32)$$

at every interior grid point. To isolate the influence of degeneracy, some experiments use the regularized coefficient family

$$v_\epsilon(u) = \frac{1}{1+2\epsilon} \left(\epsilon + \frac{u}{1+u} \right), \quad \epsilon \geq 0. \quad (4.33)$$

This normalization preserves

$$v_\epsilon(1) = \frac{1}{2},$$

while varying the coefficient ratio according to

$$\frac{v_\epsilon(0)}{v_\epsilon(1)} = \frac{\epsilon}{\epsilon + \frac{1}{2}}. \quad (4.34)$$

The original degenerate problem is recovered at $\epsilon = 0$, whereas increasing ϵ produces a more uniformly elliptic problem without changing the maximum coefficient. A constant-coefficient case, $v(u) \equiv 1/2$, is also used as a linear control. The centered Poisson stencil and conservative flux construction are standard finite-difference discretizations [19].

4.5 Discrete Methods: SORp and SORh

This section develops discrete approximations of the two-dimensional pseudo-time models introduced in Sections 4.3 and 4.4. All methods are formulated on a common Cartesian grid and use the same discrete elliptic operator, boundary treatment, and convergence diagnostics. We first derive the standard parabolic scheme (SORp), then the canonical unsplit hyperbolic scheme (SORh). Modified formulations such as directional SORh are then presented as secondary alternatives to the main SORh formulation.

4.5.1 Parabolic SOR (SORp)

We first consider a parabolic pseudo-time relaxation, denoted SORp. Given the parabolic pseudo-time model

$$\phi_t = \Delta\phi - f, \quad (4.35)$$

A forward-Euler pseudo-time step combined with successive over-relaxation gives the pointwise update

$$\phi_{i,j}^{k+1} = \phi_{i,j}^k + \omega \Delta t [R(\phi)_{i,j}^{\text{GS}} - f_{i,j}], \quad (4.36)$$

where $R(\phi)^{\text{GS}}$ uses the most recently available neighboring values. For the Poisson problem, $R(\phi) = \Delta\phi$; the nonlinear problem uses the corresponding variable-coefficient discrete operator. The size of the pseudo-time step is determined by a standard von Neumann analysis [19] with the Poisson residual, yielding the stability condition

$$\Delta t_P = \frac{\text{CFL}_P}{2(h_x^{-2} + h_y^{-2})}, \quad 0 < \text{CFL}_P \lesssim 1, \quad (4.37)$$

Hence, the same $O(h^2)$ restriction applies both to one-dimensional and two-dimensional SORp. For the full update sweep, we use a standard lexicographic ordering

$$i = 1, \dots, N_x, \quad j = 1, \dots, N_y, \quad (4.38)$$

resulting in cells $(i-1, j)$ and $(i, j-1)$ containing the updated next-iterate values. Red-black ordering, in which the two checkerboard color classes are updated successively, is a valid alternative but is not used in numerical experiments here.

4.5.2 Hyperbolic SOR (SORh)

For the coupled pseudo-time model

$$\phi_{tx} + \phi_{ty} = \Delta\phi - f, \quad (4.39)$$

a forward-time, backward in space approximation of the mixed derivatives gives

$$\frac{(D_x^- \phi^{k+1})_{i,j} - (D_x^- \phi^k)_{i,j}}{\Delta t} + \frac{(D_y^- \phi^{k+1})_{i,j} - (D_y^- \phi^k)_{i,j}}{\Delta t} = R(\phi)_{i,j}^k. \quad (4.40)$$

where

$$D_x^- = \frac{\phi_{i,j}^k - \phi_{i-1,j}^k}{\Delta x}, \quad D_y^- = \frac{\phi_{i,j}^k - \phi_{i,j-1}^k}{\Delta y}. \quad (4.41)$$

During a forward lexicographic sweep, solving this relation for a provisional value $\phi_{i,j}^*$ gives

$$\phi_{i,j}^* = \phi_{i,j}^k + c_1 \left[h_y \left(\phi_{i-1,j}^{k+1} - \phi_{i-1,j}^k \right) + h_x \left(\phi_{i,j-1}^{k+1} - \phi_{i,j-1}^k \right) + c_2 \left(R(\phi)_{i,j}^k - f_{i,j} \right) \right]. \quad (4.42)$$

where

$$c_1 = \frac{1}{h_x + h_y}, \quad c_2 = \Delta t h_x h_y, \quad h_x = \Delta x, \quad h_y = \Delta y. \quad (4.43)$$

The relaxed update is

$$\phi_{i,j}^{k+1} = \phi_{i,j}^k + \omega \left(\phi_{i,j}^* - \phi_{i,j}^k \right). \quad (4.44)$$

The pseudo-time scale restriction follows from the standard CFL condition for first-order upwind transport discretizations. For example, the two-dimensional donor-cell method for constant characteristic speeds is monotone when

$$\Delta t_H \left(\frac{|a_x|}{h_x} + \frac{|a_y|}{h_y} \right) \leq 1, \quad (4.45)$$

which implies $\Delta t_H = O(h)$ under uniform refinement [20]. This result motivates the pseudo-time step definition

$$\Delta t_H = \text{CFL}_H \min(h_x, h_y), \quad (4.46)$$

The validity of this choice is justified experimentally in the numerical results. To reiterate, $\phi_{i,j}^*$ is the provisional point update, and $\phi_{i,j}^{k+1}$ is the relaxed update. This method is considered *point-implicit* because already-updated neighbors enter the local formula, and *unsplit* because the two directional contributions are retained in one update. Because there is a one-sided GS-like dependency, we must consider the order in which individual cells are updated during a single pseudo-time step. For our experiments, we use the same lexicographic ordering as SORp,

$$i = 1, \dots, N_x, \quad j = 1, \dots, N_y, \quad (4.47)$$

with i as the outer loop and j as the inner loop. This ordering ensures that cell (i, j) , uses the updated values from both $(i-1, j)$ and $(i, j-1)$. For problems with certain symmetries, the inner and outer loops could be swapped, but care must be taken to ensure the iteration is unaffected.

Red-black ordering *is* a valid alternative for SORh, but it must be treated carefully due to the explicit correction terms. In particular, the first half of the update

(the “red” sweep) should not contain any correction terms, while the second half of the update must contain all four correction terms. This directly corresponds to the upstream correction dependencies in red–black SORp, since in both cases, all four neighboring values are updated in the first half and used in the second half of the sweep. This ordering is not used in any experiments, so we will not pursue it further.

As written, the update in Eq. (4.42) contains two distinct components. The increments $h_y(\phi_{i-1,j}^{k+1} - \phi_{i-1,j}^k)$ and $h_x(\phi_{i,j-1}^{k+1} - \phi_{i,j-1}^k)$ are nonzero only after the upstream values have been updated within the current sweep. They are correction signals propagating downstream, helping the convergence rate accelerate. The second component of Eq. (4.42) is the local residual corresponding to the discrete elliptic steady state. This distinction is important in understanding the dynamics and performance of the numerical method.

4.6 Connection to AOR and Classical SOR

The continuous SORh model is related to hyperbolic relaxation approaches for diffusion, including Nishikawa’s first-order-system formulation [21], but its linear discrete update has a more specific algebraic classification. It belongs to the accelerated overrelaxation (AOR) family introduced by Hadjidimos [11]. This section serves a similar purpose as Section 2.8, but now for the two-dimensional formulations. We will identify the two-dimensional SORh–AOR parameter map, recover classical SOR as the balanced special case, and use that reduction to obtain a reference value for the optimal relaxation parameter.

4.6.1 AOR and Classical SOR

Let

$$\delta_{i,j}^{k+1} := \phi_{i,j}^{k+1} - \phi_{i,j}^k, \quad R_{i,j}^k := (\Delta_h \phi^k)_{i,j} - f_{i,j}, \quad (4.48)$$

where $R_{i,j}^k$ is evaluated from the centered five-point stencil. On a uniform square grid, the canonical SORh sweep can be written as

$$\delta_{i,j}^{k+1} = \frac{\omega_H}{2} \left[\delta_{i-1,j}^{k+1} + \delta_{i,j-1}^{k+1} + \text{CFL}_H h^2 R_{i,j}^k \right]. \quad (4.49)$$

The two upstream terms propagate corrections already computed during the lexicographic sweep; the residual term generates a new local correction. To compare Eq. (4.49) with AOR, write the discrete Poisson system as $A\phi = b$, with $A = -\Delta_h$, $b = -f$, and $A = D - L - U$. Thus, $b - A\phi^k = R^k$. The AOR correction equation is

$$(D - r_A L)\delta^{k+1} = \omega_A (b - A\phi^k), \quad (4.50)$$

where r_A weights lower-triangular propagation and ω_A weights the residual correction. This correction identity is established for the one-dimensional system in Appendix A.1 and is independent of spatial dimension. Since $D = (4/h^2)I$ for the five-point operator, Eq. (4.50) gives the local relation

$$\delta_{i,j}^{k+1} = \frac{r_A}{4} \left(\delta_{i-1,j}^{k+1} + \delta_{i,j-1}^{k+1} \right) + \frac{\omega_A}{4} h^2 R_{i,j}^k. \quad (4.51)$$

Coefficient matching therefore yields the exact map

$$\boxed{r_A = 2\omega_H, \quad \omega_A = 2\omega_H \text{CFL}_H.} \quad (4.52)$$

This map makes the two roles in the hyperbolic update explicit. The parameter ω_H sets the strength of directional correction propagation and also enters the common scale of the update, while CFL_H changes residual injection relative to that propagation. AOR supplies the established two-parameter algebraic framework; the transport derivation supplies the directional and pseudo-time interpretation of its parameters.

Classical SOR is the AOR subfamily $r_A = \omega_A = \alpha$. Thus, when $\text{CFL}_H = 1$, Eq. (4.52)

reduces to

$$\boxed{\alpha = 2\omega_H}, \quad (4.53)$$

and SORh is exactly classical lexicographic SOR. The same mapping holds in one and two dimensions: the factor $1/(2h)$ introduced by the two-dimensional transport discretization compensates for the change from the three-point diagonal $2/h^2$ to the five-point diagonal $4/h^2$. The coefficient cancellation and corresponding iteration-matrix identity are given in Appendix A.2. For comparison, the square-grid SORp update is

$$\delta_{i,j}^{k+1} = \frac{\omega_P \text{CFL}_P}{4} \left[\delta_{i-1,j}^{k+1} + \delta_{i,j-1}^{k+1} + h^2 R_{i,j}^k \right]. \quad (4.54)$$

Its AOR parameters are equal for every CFL_P ,

$$r_A = \omega_A = \alpha = \omega_P \text{CFL}_P. \quad (4.55)$$

Hence SORp remains on the classical-SOR slice, whereas SORh moves through the broader AOR family when $\text{CFL}_H \neq 1$.

4.6.2 Optimal Relaxation Parameters

At $\text{CFL}_H = 1$, the classical SOR optimum applies directly. On an $M \times M$ square grid with homogeneous Dirichlet data, the five-point Jacobi eigenvalues and spectral radius are

$$\mu_{p,q} = \frac{1}{2} \left[\cos\left(\frac{p\pi}{M+1}\right) + \cos\left(\frac{q\pi}{M+1}\right) \right], \quad \rho_J = \mu_{1,1} = \cos\left(\frac{\pi}{M+1}\right). \quad (4.56)$$

The spectral radius is therefore the same as for the corresponding one-dimensional grid. The classical optimum [9, 27] is

$$\alpha_{\text{opt}} = \frac{2}{1 + \sqrt{1 - \rho_J^2}} = \frac{2}{1 + \sin\left(\frac{\pi}{M+1}\right)}. \quad (4.57)$$

Combining Eqs. (4.53) and (4.57) gives

$$\boxed{\omega_{H,\text{opt}}^{2D} = \frac{1}{1 + \sin\left(\frac{\pi}{M+1}\right)} = \frac{1}{1 + \sin(\pi/N)}, \quad N = M + 1.} \quad (4.58)$$

Thus, the one- and two-dimensional SORh formulas agree for two separate reasons: the discrete normalization preserves $\alpha = 2\omega_H$, and the square-grid Jacobi spectral radii agree. Similarly, Eq. (4.55) gives

$$\omega_{P,\text{opt}}^{2D} = \frac{2}{\text{CFL}_P \left[1 + \sin\left(\frac{\pi}{M+1}\right) \right]}. \quad (4.59)$$

These expressions are exact for the linear, constant-coefficient problem on a uniform square grid with the centered five-point operator and compatible lexicographic ordering. For $\text{CFL}_H \neq 1$, SORh is AOR rather than classical SOR, so Eq. (4.58) is only a reference value. Optimization over both AOR parameters is a distinct problem studied by Avdelas and Hadjidimos [2]. Applying that analysis to the nonlinear, directional, and alternative-stencil SORh variants considered here is outside the scope of this thesis. Numerical parameter studies are carried out and examined in Chapter 5.

4.6.3 Pseudo-Time-Step Scaling

As for the one-dimensional formulation, SORh does not actually apply an $O(h)$ correction directly to the elliptic residual, despite the different scaling factors $\Delta t_H = O(h)$ and $\Delta t_P = O(h^2)$. Solving the discrete mixed-derivative equation for $\phi_{i,j}$ introduces another factor of h , so the SORh residual contribution is $O(\Delta t_H h) = O(h^2)$, matching the parabolic correction scale. As explained in Section 2.7, the $O(h)$ stability scale is genuinely hyperbolic at the pseudo-time-model level, as in first-order hyperbolic diffusion formulations, while the scalar sweep-to-sweep update still has the elliptic scaling required by Eqs. (4.52) and (4.53).

4.7 Directional SORh

An interesting variation of SORh follows from the hyperbolic directional transport model introduced in Section 4.3.2. First, we introduce a prescribed direction

$$\mathbf{b} = (b_x, b_y), \quad b_x \propto \cos \theta, \quad b_y \propto \sin \theta.$$

The signs of b_x and b_y determine the sweep orientation, while their magnitudes determine the relative propagation of corrections in the coordinate directions. On a uniform square grid with spacing h , the directional correction equation may be written as

$$\delta_{i,j}^{(k+1)} = \omega_H \left[p_x \delta_{x,\text{up}}^{(k+1)} + p_y \delta_{y,\text{up}}^{(k+1)} + \frac{\text{CFL}_H h^2}{|b_x| + |b_y|} R_{i,j}^{(k)} \right],$$

where

$$p_x = \frac{|b_x|}{|b_x| + |b_y|}, \quad p_y = \frac{|b_y|}{|b_x| + |b_y|}, \quad p_x + p_y = 1.$$

Here, $\delta_{x,\text{up}}$ and $\delta_{y,\text{up}}$ denote the corrections at the two upstream neighbors selected by the signs of b_x and b_y . Thus, changing the quadrant of θ selects a mirrored lexicographic ordering, whereas varying θ within a quadrant continuously changes the relative x - and y -propagation weights. The default SORh update is then recovered for a grid-diagonal direction. With the normalization used here, the four diagonal directions satisfy

$$\theta \in \left\{ \frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4} \right\}, \quad |b_x| = |b_y| = 1.$$

Consequently, $p_x = p_y = 1/2$, and the update reduces to

$$\delta_{i,j}^{(k+1)} = \frac{\omega_H}{2} \left[\delta_{x,\text{up}}^{(k+1)} + \delta_{y,\text{up}}^{(k+1)} + \text{CFL}_H h^2 r_{i,j}^{(k)} \right].$$

Each diagonal direction therefore represents the default SORh method under a corresponding mirrored sweep ordering. When $\text{CFL}_H = 1$, this expression is identical to

classical SOR,

$$\delta_{i,j}^{(k+1)} = \frac{\alpha}{4} \left[\delta_{x,\text{up}}^{(k+1)} + \delta_{y,\text{up}}^{(k+1)} + h^2 r_{i,j}^{(k)} \right],$$

under the usual parameter correspondence

$$\alpha = 2\omega_H.$$

For $\text{CFL}_H < 1$ and a non-diagonal direction $p_x \neq p_y$, directional SORh is a distinct generalization of the default SORh method, which is itself a special case of classical SOR. For the isotropic five-point Poisson operator, standard SOR cannot reproduce the off-diagonal update, since it assigns equal coefficients to the two upstream corrections. Changing the sweep ordering changes which neighbors are upstream but does not change their relative weights. The directional formulation therefore continuously deforms the triangular splitting used by the iteration while leaving the underlying discrete residual unchanged. In this sense, default SORh is the balanced-propagation member of a broader directional family, while non-diagonal directions provide a genuine extension beyond classical SOR and default SORh.

While the directional-generalization idea is interesting, it has not been proven to meaningfully improve performance against SORh in any useful metric. There are cases where under certain conditions, such as a strong directional bias in the elliptic steady solution, certain grid-diagonal sweep orderings can be somewhat more efficient than others. These cases, however, can be achieved with careful reordering of sweep directions and update coefficients, and generally do not require specific off-diagonal directions for optimal performance. This idea may be explored further in future work.

4.8 Diagnostics for Two-Dimensional Experiments

The two-dimensional studies use the RMS norm defined in (2.43), now over the interior two-dimensional grid $\mathcal{I}_h$. Let $w = \phi$ for the Poisson problem and $w = u$ for the nonlinear

diffusion problem. The update norm is

$$e_u^{(k+1)} := \left\| w^{(k+1)} - w^{(k)} \right\|_{\text{RMS}, \mathcal{I}_h} . \quad (4.60)$$

The problem-specific equation residuals are

$$e_{r,P}^{(k+1)} := \left\| \Delta_h \phi^{(k+1)} - f_h \right\|_{\text{RMS}, \mathcal{I}_h} , \quad (4.61)$$

$$e_{r,NL}^{(k+1)} := \left\| \mathcal{R}_h^{NL} \left(u^{(k+1)} \right) \right\|_{\text{RMS}, \mathcal{I}_h} , \quad (4.62)$$

where $\mathcal{R}_h^{NL}$ is the discrete nonlinear diffusion residual defined above. Because an exact solution is available for the Poisson configuration, its solution error is also reported:

$$e_{s,P}^{(k+1)} := \left\| \phi^{(k+1)} - \phi_{\text{exact},h} \right\|_{\text{RMS}, \mathcal{I}_h} . \quad (4.63)$$

For either problem, convergence is declared when the corresponding equation residual satisfies $e_{r,P}^{(k+1)} \leq \varepsilon_r$ or $e_{r,NL}^{(k+1)} \leq \varepsilon_r$. If $e_u^{(k+1)} \leq \varepsilon_u$ before the residual tolerance is reached, the run is terminated as stagnated rather than converged. Non-finite diagnostics, diagnostics above the prescribed upper bound, and the iteration limit also terminate a run without convergence. The experiment-specific tolerances and iteration limits are stated with the corresponding results.

The Poisson results in Chapter 5 use $e_{s,P}$ to distinguish iterative convergence from solution accuracy; no exact solution is assumed for the nonlinear configuration, so that study instead uses $e_{r,NL}$ together with agreement between the converged numerical solutions. The Chapter 5 results therefore test whether the stability, relaxation-parameter, and sweep-order behavior developed in this chapter persists for the Poisson and nonlinear two-dimensional configurations.

Chapter 5

Results of Two-Dimensional Experiments

In this chapter, we show the results of experiments using the two-dimensional iterative methods described in Chapter 4.

5.1 Two-Blob Poisson Problem

The Poisson solvers are evaluated using a two-dimensional adaptation of the smooth two-component test problem introduced by Hanawa [14]. Define

$$A(x, y) = \frac{(x - 5R)^2 + y^2}{3R^2}, \quad B(x, y) = \frac{(x + 5R)^2 + y^2}{3R^2}.$$

Restricting Hanawa's analytical potential to the plane $z = 0$ gives

$$\phi_{\text{ex}}(x, y) = -4\pi G\rho_0 R^2 \left[(1 + A)^{-1/2} + (1 + B)^{-1/2} \right].$$

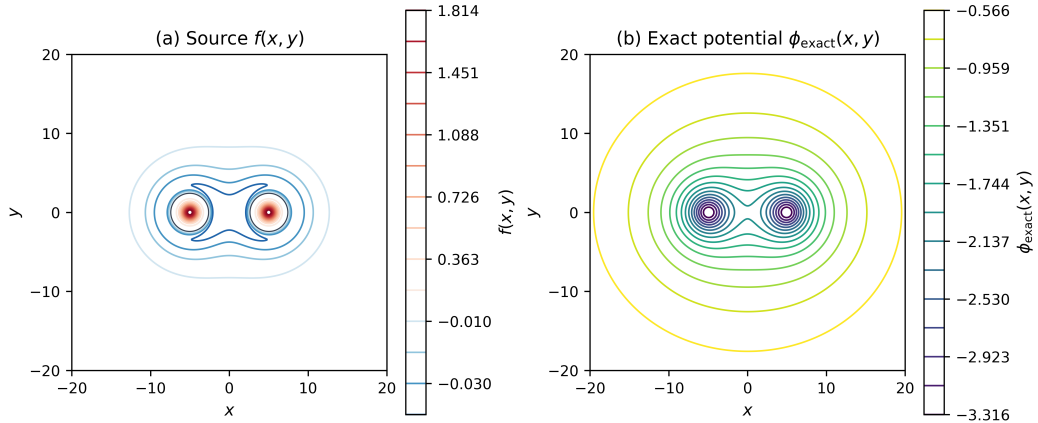


Figure 5.1: Two-blob Poisson test problem. Panel (a) shows contours of the prescribed source term $f(x, y)$, while panel (b) shows contours of the corresponding exact solution $\phi_{\text{exact}}(x, y)$. The exact solution is used to impose the Dirichlet boundary conditions and evaluate the numerical error.

Applying the two-dimensional Laplacian to this potential, rather than setting $z = 0$ in the original three-dimensional density, yields the compatible planar source

$$\rho_{2D}(x, y) = \frac{\rho_0}{3} \left[(2 - A)(1 + A)^{-5/2} + (2 - B)(1 + B)^{-5/2} \right],$$

so that $\nabla^2 \phi_{\text{ex}} = 4\pi G \rho_{2D}$ exactly. The two source peaks are centered at $(\pm 5R, 0)$. All calculations use $R = 1$, $\rho_0 = 3$, and $4\pi G = 1$ on $\Omega = [-20, 20]^2$, with ϕ_{ex} imposed as the Dirichlet boundary condition and used to measure the numerical error.

Note that while this problem is constructed to resemble a realistic Poisson problem with source $f(x, y)$ and gravitational potential $\phi_{\text{exact}}(x, y)$ (shown in Figure 5.1), the sampled minimum of the density distribution $\rho_{2D}(x, y)$ is approximately -0.05 , so the problem does not have a strictly physical mass density.

5.1.1 Performance Results

Figure 5.2 shows combined performance results for SORp and SORh. At $\text{CFL} = 1$, both methods require essentially the same number of iterations at every resolution. The measured iteration counts increase from 33 to 522 as N increases from 16 to 256,

Table 5.1: Numerical configuration and parameters used for the two-blob performance and accuracy studies.

Parameter	Setting
Spatial grids	$N_x = N_y = N$, with $N \in \{16, 32, 64, 128, 256\}$
Methods and ordering	SORp and SORh, both using lexicographic ordering
Relaxation parameters	Selected separately for each combination of N , method, and CFL number by minimizing the converged iteration count.
Performance convergence	$e_r^{k+1} \leq \varepsilon_r = 10^{-5}$, $e_u^{k+1} \leq \varepsilon_u = 10^{-10}$
Accuracy convergence	$e_r^{k+1} \leq \varepsilon_r = 10^{-10}$, $e_u^{k+1} \leq \varepsilon_u = 10^{-13}$
Maximum iteration limit	20,000 for performance runs, 100,000 for accuracy runs
Stability guard	Iteration halts if $e_r^{k+1} > 10^8$
Accuracy validation	The estimated algebraic error $\ r\ _{\text{RMS}}/\lambda_{\min}(L_h)$ was required to be no more than 10^{-3} of the measured RMS solution error

with a fitted dependence $k \propto N^{0.995}$. This agreement reproduces the known $O(N)$, or equivalently $O(h^{-1})$, asymptotic convergence rate of optimal classical SOR [27].

SORp still has a small runtime advantage because its discrete update is less expensive. At CFL = 0.8, there is a clearer separation—SORh requires moderately more iterations than SORp as the grid is refined, and the runtime difference grows accordingly. This result is consistent with the performance results for the one-dimensional linear problem in Section 3.2, and its explanation of the performance discrepancy can be applied to this problem as well. The small variation between repeated timing measurements does not affect these trends.

5.1.2 Solution Accuracy

In Figure 5.3, the representative solution and truncation errors both approach second-order convergence on the finer grids, as expected for the centered spatial discretization. The irregular observed orders on the coarsest grids indicate that these resolutions are still outside the asymptotic regime, particularly for resolving the localized features of the two-blob solution. The accuracy results are effectively independent of the iterative method and CFL number: once the residual tolerance is satisfied, SORp and SORh converge to the same discrete steady-state solution.

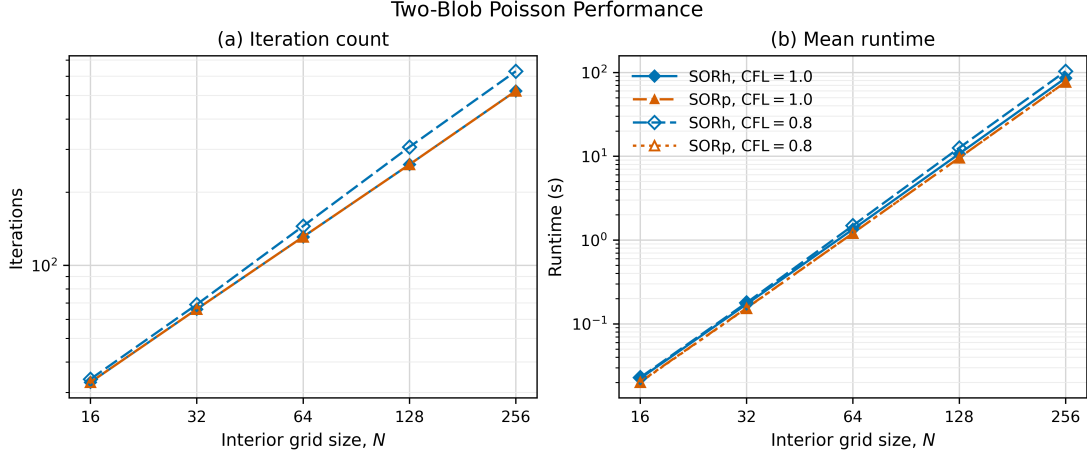


Figure 5.2: Performance of SORp and SORh for the two-blob problem using the selected relaxation parameters. Panel (a) shows the iteration counts, and panel (b) shows the mean runtimes over repeated trials. At CFL = 1, the methods require the same number of iterations, although SORp is slightly faster because its discrete update is less expensive. At CFL = 0.8, SORh also requires more iterations as the grid is refined. Runtime error bars are omitted because the measured variation between repetitions is negligible on the report scale.

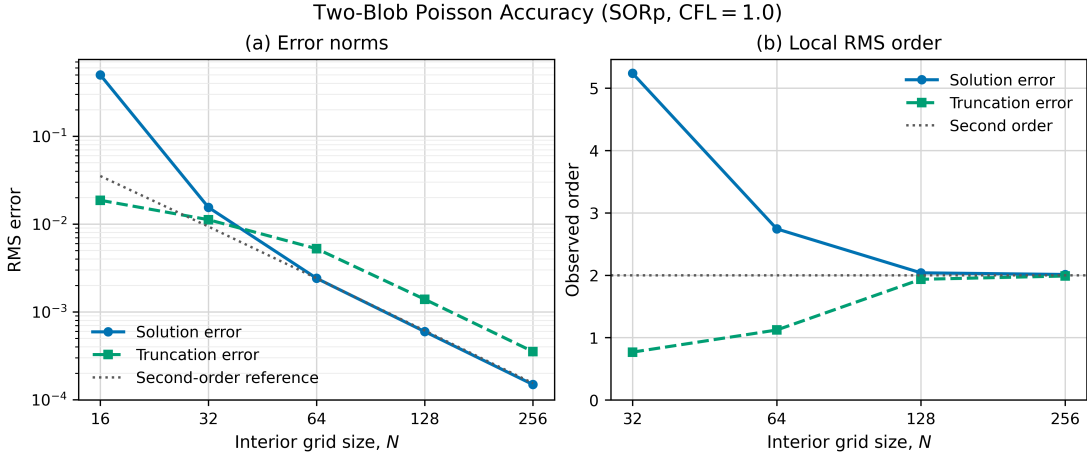


Figure 5.3: Spatial accuracy for the two-blob problem, represented by the SORp results at CFL = 1. Panel (a) shows the final pseudo-time RMS solution error e_s , and the RMS truncation error, obtained by applying the discrete operator to the exact solution. Panel (b) shows the corresponding local observed convergence orders. The coarse grids exhibit pre-asymptotic behavior, while both error measures approach second-order convergence under refinement. The other method and CFL configurations produce effectively indistinguishable accuracy results.

5.2 Nonlinear Diffusion Problem

The nonlinear experiments use the problem defined in Section 4.1.2, with the original coefficient $v(u) = u/(1 + u)$, a zero initial solution, and $\text{CFL} = 0.8$. SORh and SORp use fixed pseudo-time steps, identical stopping tolerances, and relaxation parameters optimized separately for each method and grid.

5.2.1 Optimized Performance and Grid Scaling

Figure 5.4 compares the optimized methods. As N increases from 16 to 64, the SORh sweep count increases from 139 to 662, whereas the SORp count increases from 221 to 3502. The corresponding SORh advantage grows from 1.59 to 5.29 in sweep count and from 1.58 to 5.19 in elapsed time. Fits over these three grids give empirical sweep exponents of 1.13 for SORh and 1.99 for SORp. These are descriptive finite-range fits, not established asymptotic rates. The final solutions differ by only 7.5×10^{-7} to 1.0×10^{-6} in the normalized solution norm, so the performance difference is not caused by convergence to different discrete solutions.

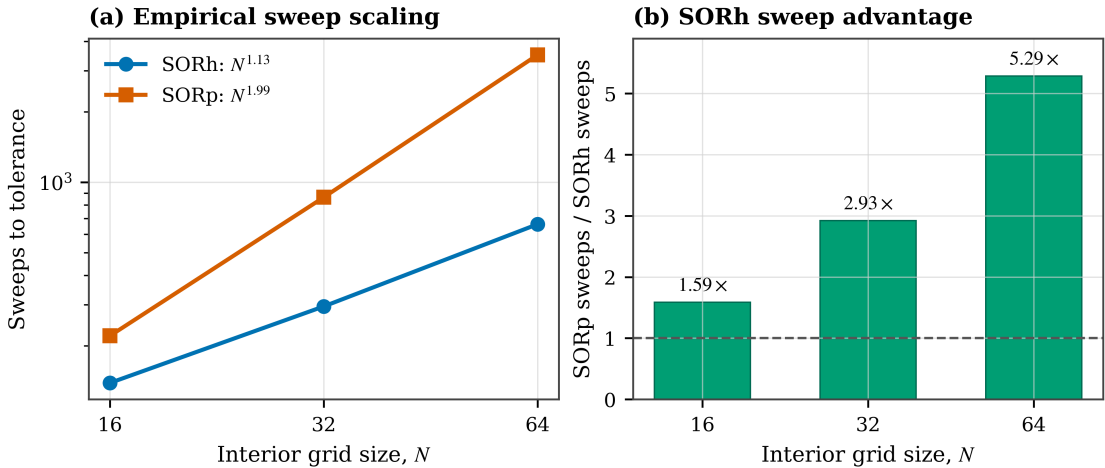


Figure 5.4: Optimized fixed-timestep performance for the two-dimensional nonlinear problem. Panel (a) shows sweep counts and descriptive power-law fits over $N = 16, 32, 64$; panel (b) shows the corresponding SORh sweep advantage.

This pattern agrees with the one-dimensional nonlinear results in Section 3.3. There,

the difference was attributed to the local diffusion coefficient multiplying SORp’s transmission of upstream corrections, while the corresponding SORh propagation term is coefficient-independent. A direct diagnostic is available from the two-dimensional baseline data. On a square grid, the coefficient multiplying each upstream SORh correction is $c_H = \omega_H/2$. The weakest corresponding SORp coefficient at the final state is

$$c_{P,\min} = \omega_P \frac{\Delta t_P}{h^2} v_{\min} = 0.4 \omega_P v_{\min}, \quad (5.1)$$

because $\Delta t_P/h^2 = 0.4$ in these experiments. Table 5.2 shows that refinement resolves smaller face coefficients near $u = 0$. Consequently, the weakest SORp propagation factor decreases while the SORh factor remains nearly constant.

Table 5.2: Final-state diagnostic of the weakest local upstream-correction factor in the optimized nonlinear runs.

N	$v_{\min}$	c_H	$c_{P,\min}$	$c_H/c_{P,\min}$
16	0.0519	0.416	0.0567	7.3
32	0.0301	0.448	0.0323	13.9
64	0.0169	0.468	0.0177	26.5

This diagnostic is not a nonlinear spectral proof, but it directly connects the growing performance separation to increasingly weak local SORp correction propagation.

5.2.2 Coefficient Contrast

The regularized coefficient family from Section 4.4.2 varies the ellipticity ratio $v_{\min}/v_{\max}$ while preserving $v(1) = 0.5$. Figure 5.5 shows the locally optimized results. SORh remains approximately 2.9–3.0 times faster in sweep count from the degenerate case through an ellipticity ratio of 0.167. At a ratio of 0.667, its advantage falls to 1.59. In the constant-coefficient control, SORp requires 109 sweeps and SORh requires 123, restoring the modest linear SORp advantage.

Strict degeneracy is therefore not required for the SORh advantage. The more precise association is with weak local diffusion and strong spatial coefficient contrast: the

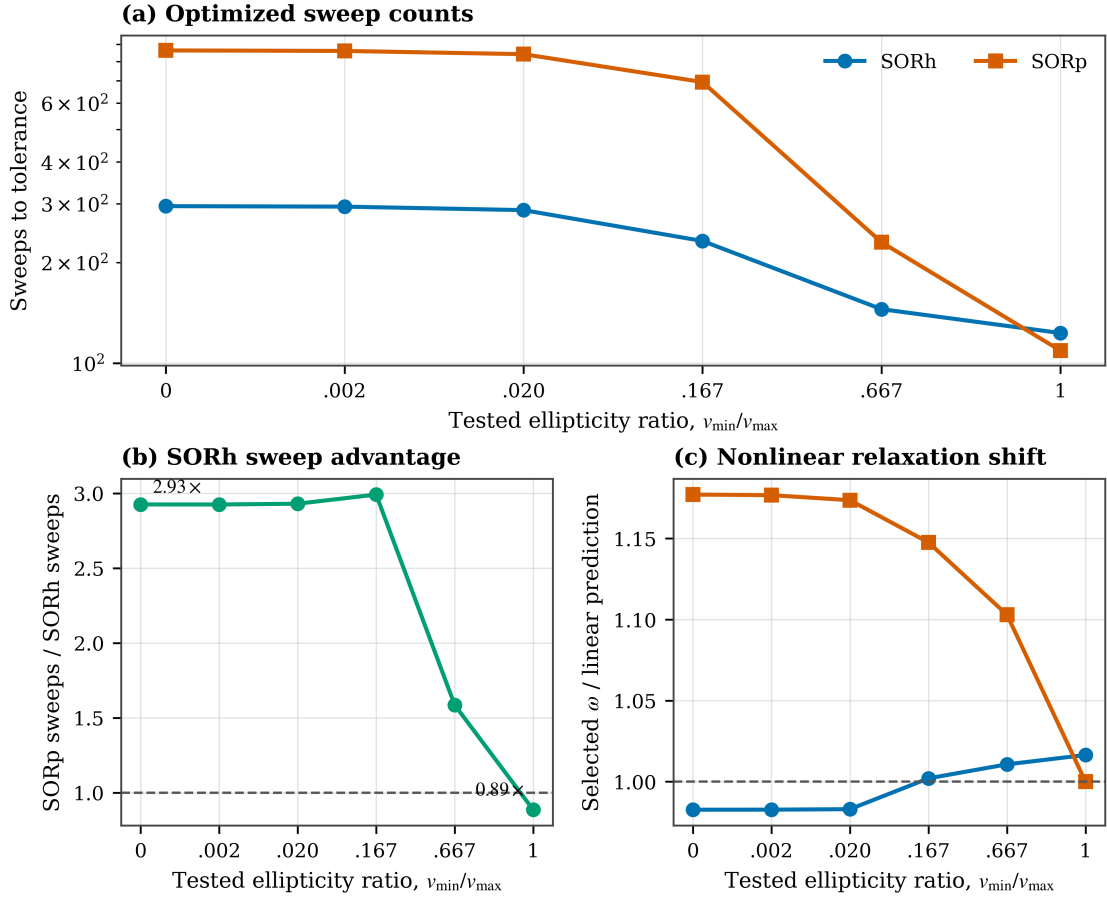


Figure 5.5: Effect of reducing coefficient contrast at $N = 32$. Panel (a) shows optimized sweep counts, panel (b) their ratio, and panel (c) the selected relaxation parameters normalized by the linear predictions. The horizontal reference in panels (b) and (c) is one.

advantage persists under mild regularization, diminishes as the coefficient becomes more uniform, and reverses only at the constant-coefficient endpoint. Panel (c) provides complementary evidence. The optimized SORp parameter moves toward its linear prediction as coefficient contrast is removed, whereas the SORh parameter changes comparatively little. These trends support, but do not by themselves prove, the correction-propagation interpretation.

5.2.3 Supporting Parameter Studies

The relaxation-parameter searches reproduce the findings from Figures 3.3 and 3.4. Across $N = 16, 32, 64$, the SORh optima remain within 1.7% of the linear predictions and have clear interior minima. The SORp optima exceed their predictions by approximately 31%, 18%, and 10%; at $N = 32$ and 64, they lie at the edge of the first stable branch. Thus, the linear formula remains a useful SORh reference but does not predict the nonlinear SORp optimum.

Control experiments do not remove the performance separation. With locally reoptimized parameters, adaptive timesteps change the SORh count by less than 1% and increase the SORp count by 6–8%. Freezing all SORp face coefficients at the start of each sweep changes its $N = 32$ count from 863 to 849, versus 295 for SORh. Neither the fixed timestep nor within-sweep coefficient evaluation therefore explains the difference. An experiment using the initial guess $u^0 = x$ rather than $u^0 = 0$ at $N = 32$ and with newly optimized relaxation parameters alters the nonlinear trajectories but does not reduce the performance separation.

Varying ω_H and CFL_H separately also shows that their product does not determine convergence. At $N = 32$ and fixed $\omega_H = 0.895$, raising CFL_H from 0.5 to 1.0 reduces the count from 467 to 238. Holding $\omega_H \text{CFL}_H = 0.716$ instead fails for $\text{CFL}_H = 0.5$ – 0.7 and requires 606 and 856 sweeps at 0.9 and 1.0.

These results associate SORh’s advantage with this separation of correction propagation from residual injection, specifically through weak-diffusion regions. For

consistently ordered linear systems, optimal-AOR analysis shows that the best two-parameter pair can either lie away from the classical-SOR slice or coincide with optimal SOR, depending on the relevant spectral case [2]. Whether the SORh parameter separation provides a comparable advantage for problems beyond the weak-diffusion configurations tested here remains an open question.

5.3 FLASH Experiments

We next test whether these findings carry into a multiphysics application by implementing SORh in FLASH [10, 7] and applying it to the Zel’dovich Pancake problem. SORh is compared with FLASH’s multigrid Poisson solver, based on the Huang–Greengard algorithm and Ricker’s finite-volume oct-tree extension [16, 24]. The comparison examines runtime, parallel scaling, accuracy, iteration limits, and SORh iteration ordering variations.

5.3.1 Zel’dovich Pancake

The Zel’dovich pancake is a coupled hydrodynamics, collisionless-particle, and self-gravity problem that tests a Poisson solver inside a complete cosmological time integration. A single sinusoidal density perturbation is imposed normal to a plane in an expanding, periodic domain [28, 1]. If q and x denote the Lagrangian and Eulerian coordinates normal to the plane, the pre-caustic solution may be written

$$x = q - \frac{1 + z_c}{1 + z} \frac{\sin(kq)}{k}, \quad \rho_b(x, z) = \bar{\rho}_b \left[1 - \frac{1 + z_c}{1 + z} \cos(kq) \right]^{-1}, \quad (5.2)$$

where $k = 2\pi/\lambda$ and z_c is the caustic redshift. At $z > z_c$ the mapping is single valued, so the density and normal velocity provide useful analytical checks. At $z = z_c$ the collisionless sheet first folds. Thereafter the dark-matter particles form a multistream region with repeated caustics, while the gas forms two shocks that propagate away from a small central unshocked region. A global analytical solution is no longer available

Table 5.3: Shared Pancake setup and solver-specific controls. The 16×16 cell block layout (64 blocks) is used for the primary comparison.

Item	Setting
Physical problem	$[0, \lambda]^2$, periodic, $\lambda = 3.0857 \times 10^{25}$ cm; $z : 50 \rightarrow 0$, $z_c = 5$; $\Omega_m = 1$, $\Omega_b = 0.15$, $\Omega_\Lambda = 0$, $H_0 = 1.62038 \times 10^{-18}$ s $^{-1}$; $T_{\text{fid}} = 550$ K at $z_{\text{fid}} = 200$.
Discretization	Uniform 128×128 mesh; 128^2 particles; 16×16 cells per block; third-order PPM reconstruction, HLLC Riemann solver [26], $\gamma = 5/3$, hydrodynamic CFL 0.8; periodic gravity.
Parallel environment	1 and 4 MPI ranks, one thread per rank; MPICH 4.2.2; Apple M1 Pro (8 cores, 16 GB), Darwin 25.5. Repeated runs were serialized and configuration order was rotated between repetitions.
MG	Huang–Greengard multigrid; relative residual tolerance 10^{-6} .
SORh wavefront	Pseudo-time CFL 0.9; relative residual tolerance 10^{-6} ; $\omega_H = [1 + \sin(\pi/128)]^{-1} = 0.97605$; primary cap $m_{\text{max}} = 200$; residual checked every 10 sweeps.
SORh budget study	$m_{\text{max}} = 100, 200, 300, 400, 500$; check intervals 10, 10, 15, 20, 25, respectively.
Measurement	Three repetitions per configuration; reported times are medians. Wall time is external elapsed time; Poisson time is the sum of the maximum-per-rank solver timers over all gravity calls.

after this transition, so post-caustic comparisons are treated primarily as cross-solver consistency tests.

FLASH initializes the gas and particles from the same perturbation. During each time step, particle mass is deposited by cloud-in-cell interpolation, the periodic Poisson equation is solved for the total matter distribution, and the resulting gravitational field is coupled to the hydrodynamic and particle updates. The essential numerical and physical settings are summarized in Table 5.3. The fixed uniform grid isolates the solver and parallel ordering effects; adaptive refinement was disabled.

Sweep ordering

The iteration of SORh is highly dependent on the order of the update sweep, due to its directional dependencies, so our experiments use two distinct ordering modes. In *block mode*, blocks may sweep concurrently, using the newest values inside each block but values from the preceding exchange across upstream block boundaries. This exposes substantial concurrency, but changes the effective global iteration matrix and makes the stable relaxation parameter depend on block size. The full-trajectory tests required

Table 5.4: Primary MG–SORh timing comparison. Times are medians of three repetitions in seconds; ratios use MG at the same rank count.

Ranks	Method	Wall	Poisson	Poisson/Wall	Wall/MG	Poisson/MG
1	MG	22.81	7.133	31.3%	1.000	1.000
1	SORh, $m_{\max} = 200$	23.61	8.653	36.6%	1.035	1.213
4	MG	12.40	6.287	50.7%	1.000	1.000
4	SORh, $m_{\max} = 200$	11.47	5.780	50.4%	0.925	0.919

$\omega_H = 0.87$, 0.93 , and 0.96 for blocks containing 8^2 , 16^2 , and 32^2 cells, respectively.

In *wavefront mode*, an update is released only after its west and south predecessors in the global sweep have advanced. Points on the same diagonal front may still execute concurrently, but the method preserves the global ordered-sweep iteration and permits the grid-optimal $\omega_H = 0.97605$ for every block layout tested. This allows for better stability and faster convergence when compared to block mode. MPI communication changes when a front crosses a rank boundary, but the mathematical ordering does not. The primary performance comparisons use the wavefront mode.

Performance and iteration-budget tradeoff

The primary comparison uses the matched 16×16 block layout. Both methods made 417 gravity solves over 208 evolution steps. Table 5.4 shows that wavefront SORh with $m_{\max} = 200$ was within 3.5% of MG in total one-rank time and was 7.5% faster with four ranks. Its Poisson phase was 21.3% slower with one rank but 8.1% faster with four ranks. SORh is thus not the faster serial elliptic algorithm, but with carefully controlled $m_{\max}$, its timing results are still comparable with MG. The results also suggest that wavefront SORh exhibits stronger intranode parallel scaling. From one to four ranks, MG obtained wall-time speedup 1.84 (46.0% parallel efficiency) and Poisson speedup 1.13 (28.4%). The corresponding SORh values were 2.06 (51.5%) and 1.50 (37.4%). These are small-scale intranode measurements on one workstation, but they still support the expected benefit of diagonal-front concurrency.

The iteration cap is a deliberate accuracy–cost control, not merely a failure limit.

Table 5.5: Four-rank wavefront budget study. Time ratios between SORh and MG for both total wall-clock time and Poisson-solver time are shown. The last two columns are the largest relative L_2 difference from matched MG among gas density, normal velocity, particle density, pressure, and derived temperature.

$m_{\max}$	Converged solves	SORh/MG (Wall)	SORh/MG (Poisson)	Pre-caustic	Final
100	21/419	0.729	0.494	16.765%	17.207%
200	34/417	0.925	0.919	0.440%	0.228%
300	48/417	1.133	1.337	0.071%	0.224%
400	214/417	1.345	1.754	0.006%	0.004%
500	372/417	1.415	1.895	0.001%	0.002%

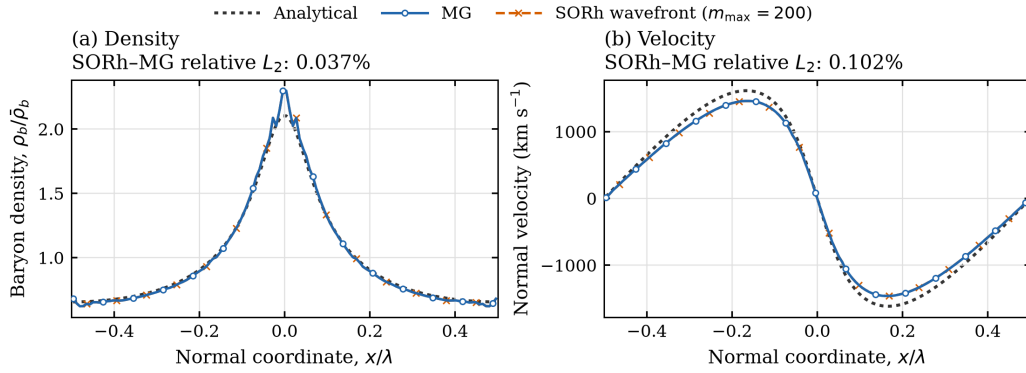


Figure 5.6: Pre-caustic profiles at $z \simeq 10.419$. MG and wavefront SORh with $m_{\max} = 200$ include the same hydrodynamic and particle discretization errors.

At $m_{\max} = 200$, only 34 of 417 solves reached the 10^{-6} residual criterion; the remaining 383 returned at the cap, with an average of 189.4 sweeps per call. Table 5.5 therefore reports both cost and application-level departure from the converged MG reference. A cap of 100 was too aggressive, producing order-10% profile differences. Increasing the cap to 400–500 nearly reproduced MG, but removed the time-to-solution advantage. The 200-sweep case is the best observed compromise: it retained comparable profiles while providing the most competitive four-rank elapsed time.

Accuracy and solution consistency

The pre-caustic checkpoint at $z \simeq 10.419$ provides the strongest accuracy test. Figure 5.6 compares the analytical, MG, and primary SORh profiles. The direct relative

L_2 differences between the solvers were 0.0367% for baryon density and 0.1018% for normal velocity. Against the analytical solution, the corresponding MG errors were 3.164% and 9.578%, and the SORh errors were 3.146% and 9.669%. The small decrease in the SORh density norm is simply due to error calculation, and is far below the common discretization error. The consistency of SORh and MG errors suggests that the capped Poisson solve is not the leading error source at this checkpoint, even though most gravity calls reached the iteration cap. This also coincides with Figure 5.6, in which the MG and SORh pre-caustic density and velocity profiles are nearly indistinguishable.

At the final checkpoint, the solvers differed by at most 0.228% across five fields and placed the gas shocks and outer particle caustics in the same cells. Since no complete post-caustic solution is available and the shock asymptotes are unsuitable here, this is a test of consistency rather than absolute accuracy. Absolute thermal metrics were excluded because all solvers shared a reference mismatch; the particle-density reference omits cloud-in-cell filtering.

At $m_{\max} = 500$, every wavefront layout and rank count agreed within 1.7×10^{-13} in relative L_2 . At $m_{\max} = 300$, the maximum MG difference was 7.73% for block mode and 0.224% for wavefront, which also reduced block-mode wall time by 47.0% and 38.8% at one and four ranks. Block mode’s greater parallel efficiency was outweighed by its smaller stable ω_H and altered iteration.

A tightly converged SORh solve therefore remains slower on this linear Poisson operator. In the coupled evolution, however, a controlled wavefront inexact solve closely matches MG and obtains comparable or lower four-rank time to solution. This result is scoped to the tested resolution, mesh, and rank counts, but suggests that in particular configurations, SORh could be a viable alternative to MG. A stronger comparison requires experiments with larger and varied grid sizes, high-rank experiments, and other parameter studies.

Chapter 6

Conclusion

SORh, nominally standing for “Successive Over-Relaxation hyperbolic” is a discrete, local-stencil, unsplit, point-implicit pseudo-time relaxation method constructed from a hyperbolic reformulation of Poisson’s equation. The initial premise of this research considered how a hyperbolic approach could be used to solve an elliptic problem. One of the clearest outcomes, however, is that the nature of the hyperbolic pseudo-time model changes character after discretization. For the linear, constant-coefficient problem, the SORh update belongs exactly to the established accelerated over-relaxation (AOR) family [11]. The connection is summarized by

$$r_A = 2\omega_H, \quad \omega_A = 2\omega_H \text{CFL}_H, \quad \text{CFL}_H = 1 \implies r_A = \omega_A = \alpha = 2\omega_H. \quad (6.1)$$

Thus, at unit CFL, SORh reduces all the way to classical SOR. This explains why its linear convergence behavior so closely resembles stationary relaxation even though the continuous pseudo-time model has genuinely hyperbolic dynamics.

This algebraic connection does not make the hyperbolic derivation incidental. In fact, neither AOR nor the classical-SOR reduction was used to construct SORh; both were recognized only after the discretization had been developed and tested. The hyperbolic model therefore provides an independent route to the AOR iteration and,

more importantly, suggests a physical meaning for its two parameters. In the SORh interpretation, r_A controls the propagation of upstream corrections, while ω_A controls the injection of the local residual. Direct coefficient matching establishes the exact linear equivalence, and the spectral analysis connects that equivalence to the observed parameter and convergence behavior. The contribution is not the invention of a new two-parameter stationary method, but the derivation, interpretation, and extension of that structure from a hyperbolic model.

The results become less predictable for nonlinear diffusion. As the solution changes, so do the diffusion coefficients and the local correction weights, and the iteration can no longer be represented by one fixed AOR matrix. Here the hyperbolic interpretation is useful, because it provides a natural way to distinguish the transport of an existing correction from the creation of a new correction by the residual. This distinction motivated the central explanation tested in the nonlinear experiments.

6.1 Results, Interpretations, & Limitations

For the one- and two-dimensional linear Poisson problems, SORh and parabolic SOR (SORp) performed similarly. At unit CFL their sweep counts coincided, as the classical-SOR reduction predicts. Below unit CFL, SORh moves away from the classical-SOR slice of AOR, and its modest performance difference from SORp is consistent with both this parameter change and the additional work in its update stencil.

However, for the nonlinear problems, SORh required substantially fewer sweeps than SORp, and this separation grew over the tested grid refinements. The one-dimensional results first revealed the trend, while the two-dimensional coefficient-contrast study provided the strongest evidence for its proposed cause. The advantage was associated most closely with weak local diffusion and strong coefficient contrast, rather than with strict elliptic degeneracy alone. It persisted after degeneracy was removed and diminished as the coefficient became more uniform; only at the constant-coefficient

endpoint did SORp become faster. As refinement resolved smaller face coefficients, SORp’s weakest local propagation factor continued to decrease, while the corresponding upstream factor in SORh remained nearly constant. This is direct evidence for the propagation mechanism, although it is not a general nonlinear spectral proof.

Several controlled tests reinforced this interpretation. Reoptimizing SORp for a different initial guess, freezing its face coefficients within each sweep, and changing the timestep strategy affected individual sweep counts but did not remove the performance separation. Across the controlled problems, the converged solutions and accuracy diagnostics also agreed within the stated tolerances. The reported differences therefore reflect how efficiently the methods reached the same discrete solution, rather than a difference in the solution being computed.

The remaining experiments broadened the picture without changing this central result. Directional SORh showed that the transport direction of the continuous model can become an additional degree of freedom in the discrete method, although the cases tested here did not reveal a practical advantage. In FLASH, an optimized wavefront implementation of SORh achieved runtime comparable to the Huang–Greengard multigrid solver on the tested hydrodynamic problems, with slightly reduced solver accuracy. An iteration cap that occasionally stopped SORh before the requested elliptic tolerance had only a small effect on the hydrodynamic solution. Within that environment, SORh was therefore a viable alternative; the result should not be read as a general claim that SORh matches multigrid performance or scalability.

These results also draw a fairly clear boundary around the present study. The analysis does not yet provide a general nonlinear convergence theory, the directional method remains exploratory, and the FLASH experiments cover a limited collection of problems and do not test true adaptive-mesh refinement or comparative parallel scaling. The numerical conclusions are consequently specific to the grids, parameter ranges, models, and implementations studied here. These limitations narrow the claims, but they also point directly toward the questions that are now worth asking.

6.2 Alternative Methods and Future Work

Not every route explored during this research led to a method worth retaining. First- and second-order directional splittings, including a symmetric Strang-type composition [25], alternated one-dimensional SORh updates in the two coordinate directions. Their steady solutions contained directional artifacts and did not reproduce the intended discrete Poisson solution. Classical alternating-direction implicit methods instead alternate one-dimensional implicit solves [23]. They provided a useful comparison, but the exploratory hyperbolic ADI variants tested here had poor stability. Wider and higher-order Laplacians, centered and higher-order one-sided mixed space–pseudo-time approximations, and additional regularization terms were also considered, but none improved the central SORh formulation enough to justify a larger role in this study. For example, we considered a weighted-hybrid approach which used additional parameters to rebalance residual forcing and coordinate-wise correction propagation, providing further generalization beyond directional SORh. In general, for the square uniform linear Poisson problem, the optimal cases for these variations coincided with optimal classical SOR (like the default SORh cases we studied here). This is because they merely provided alternate parameterizations and orderings of an AOR-like iteration, whose attainable spectral optimum for the uniform linear case is classical optimal SOR. These variations could provide better optimal control in nonlinear or nonuniform cases, but this requires further numerical experiments.

Looking back, the connection between the successful SORh discretization and its classical-SOR reduction was not initially known. The five-point Laplacian and the first-order upstream approximation of the mixed derivative were chosen independently. Nevertheless, at unit CFL their upstream old-iterate terms cancel, leaving exactly over-relaxed Gauss–Seidel. This suggests a possible reason that some alternative formulations lost favorable propagation or stability: they no longer retained the matrix splitting hidden in the original scheme. That explanation remains a hypothesis, since the present

experiments do not separate loss of the SOR reduction from inconsistency, splitting error, or other stability effects.

A natural theoretical extension would therefore be to characterize when a discretization of the mixed space–pseudo-time operator and a chosen elliptic stencil induce a prescribed AOR splitting, and when that splitting reduces further to classical SOR. Such a framework would have to treat the two spatial discretizations, grid geometry, timestep scaling, boundary treatment, and sweep ordering as parts of one design problem. Related literature offers useful pieces of this idea but do not supply the SORh sweep-equivalence criterion. Nishikawa constructs diffusion schemes by discretizing an auxiliary-variable first-order hyperbolic system, showing that first-, second-, and third-order hyperbolic finite-volume schemes can generate diffusion schemes of the corresponding order on unstructured grids [21, 22]. Compatible-discrete-operator methods similarly begin from the principle that continuous structure should survive the joint design of the discrete operators [3]. Neither construction is algebraically equivalent to SORh, but both offer useful guidance for a future stencil-compatibility theory.

Other directions follow naturally from the numerical results. Optimization over both SORh/AOR parameters should be studied beyond the cases where the best pair lies near classical SOR [2], together with a nonlinear convergence analysis of the observed coefficient-contrast effect. A direct comparison with Nishikawa-style hyperbolic diffusion could separate the effects of evolving auxiliary gradients from those of the scalar SORh construction. Directional, symmetric, multicolor, block, and adaptive-parameter variants may become more useful for anisotropic solutions, rectangular or nonuniform grids, or three-dimensional problems. SORh or one of the discussed variations could be introduced as a smoother or similar component of a multigrid framework. Finally, broader FLASH studies should include true adaptive meshes, additional physical test problems, and parallel scaling over a wider range of processor counts.

6.3 The End

SORh is neither a universal replacement for existing elliptic solvers nor simply a new name for AOR. Its most revealing feature is the bridge between the two viewpoints: a method born from hyperbolic transport that becomes established stationary relaxation in the linear limit, yet retains a distinct and useful structure for the nonlinear problems studied here. The analysis explains where that structure comes from, the experiments show when it matters, and the FLASH implementation demonstrates that it can survive the transition from a model problem to a production scientific code. Together, these results provide a clear identity for SORh and a foundation for what should come next.

Appendix A

Supplemental Analysis

A.1 One-Dimensional Coefficient Comparison of SORh, AOR, and Classical SOR

Let $Au = b$ with the unnormalized splitting $A = D - L - U$. Hadjidimos instead writes the AOR method using the diagonally normalized matrices $D^{-1}L$ and $D^{-1}U$ [11]. Multiplying that normalized equation by D gives the equivalent affine iteration used here,

$$u^{(n+1)} = (D - r_AL)^{-1} \left(B_A u^{(n)} + \omega_A b \right), \quad B_A = (1 - \omega_A)D + (\omega_A - r_A)L + \omega_A U. \quad (\text{A.1})$$

Thus, the factor $D - r_AL$ is simply the unnormalized counterpart of $I - r_AD^{-1}L$. To expose the two kinds of correction in AOR, define $\delta^{(n+1)} = u^{(n+1)} - u^{(n)}$. Multiplying Eq. (A.1) by $D - r_AL$ and subtracting $(D - r_AL)u^{(n)}$ gives

$$\begin{aligned} (D - r_AL)\delta^{(n+1)} &= [B_A - (D - r_AL)]u^{(n)} + \omega_A b \\ &= \omega_A \left[-(D - L - U)u^{(n)} + b \right] \\ &= \omega_A \left(b - Au^{(n)} \right). \end{aligned} \quad (\text{A.2})$$

The correction form is therefore algebraically equivalent to the original AOR matrix update. For the one-dimensional negative Laplacian, $D = (2/h^2)I$, while the subdiagonal of L is $1/h^2$. Taking the j th row of Eq. (A.2), write the unnormalized residual as $\mathcal{S}_j^{(n)} := h^2(b - Au^{(n)})_j$. For the homogeneous problem, $\mathcal{S}_j^{(n)} = u_{j-1}^{(n)} - 2u_j^{(n)} + u_{j+1}^{(n)}$, and the local correction is

$$\delta_j^{(n+1)} = \frac{r_A}{2}\delta_{j-1}^{(n+1)} + \frac{\omega_A}{2}\mathcal{S}_j^{(n)}. \quad (\text{A.3})$$

Substituting $\delta_j^{(n+1)} = u_j^{(n+1)} - u_j^{(n)}$ and collecting the old and new iterates yields the usual AOR point update,

$$u_j^{(n+1)} - \frac{r_A}{2}u_{j-1}^{(n+1)} = (1 - \omega_A)u_j^{(n)} + \frac{\omega_A - r_A}{2}u_{j-1}^{(n)} + \frac{\omega_A}{2}u_{j+1}^{(n)}. \quad (\text{A.4})$$

The linear SORh correction derived in Section 2.8 is

$$\delta_j^{(n+1)} = \omega_H\delta_{j-1}^{(n+1)} + \omega_H r_H \mathcal{S}_j^{(n)}. \quad (\text{A.5})$$

Comparison with Eq. (A.3) gives

$$r_A = 2\omega_H, \quad \omega_A = 2r_H\omega_H. \quad (\text{A.6})$$

Substituting this map into Eq. (A.4) recovers the SORh point update of Section 2.6. The same map follows directly from the AOR matrix or point update; the correction comparison is used because $r_A/2$ multiplies the propagated upstream correction, while $\omega_A/2$ multiplies the injected residual. At $r_H = 1$, the map gives $r_A = \omega_A = \alpha = 2\omega_H$, the classical-SOR subfamily of AOR, and Eq. (A.4) becomes

$$u_j^{(n+1)} = u_j^{(n)} + \frac{\alpha}{2} \left(u_{j-1}^{(n+1)} - 2u_j^{(n)} + u_{j+1}^{(n)} \right). \quad (\text{A.7})$$

For comparison, the parabolic correction $\delta_j^{(n+1)} = r_P\omega_P\delta_{j-1}^{(n+1)} + r_P\omega_P\mathcal{S}_j^{(n)}$ has equal AOR parameters $r_A = \omega_A = 2r_P\omega_P = \text{CFL}_P\omega_P$, which establishes the SORp compari-

son used in Section 2.8.

A.2 Two-Dimensional SORh–AOR Equivalence

This appendix records why the one-dimensional parameter map remains valid for the square-grid five-point discretization. For $A = D - L - U$, the AOR iteration matrix is

$$G_{\text{AOR}} = (D - r_A L)^{-1} [(1 - \omega_A)D + (\omega_A - r_A)L + \omega_A U]. \quad (\text{A.8})$$

Equivalently, its correction satisfies

$$(D - r_A L)\delta^{k+1} = \omega_A(b - A\phi^k). \quad (\text{A.9})$$

With lexicographic ordering, L contains the west and south couplings, so Eq. (A.9) for the five-point operator is

$$\delta_{i,j}^{k+1} = \frac{r_A}{4} \left(\delta_{i-1,j}^{k+1} + \delta_{i,j-1}^{k+1} \right) + \frac{\omega_A}{4} h^2 R_{i,j}^k. \quad (\text{A.10})$$

The square-grid SORh correction is

$$\delta_{i,j}^{k+1} = \frac{\omega_H}{2} \left(\delta_{i-1,j}^{k+1} + \delta_{i,j-1}^{k+1} + \text{CFL}_H h^2 R_{i,j}^k \right). \quad (\text{A.11})$$

Thus the local equations, and global stationary iterations are identical under

$$r_A = 2\omega_H, \quad \omega_A = 2\omega_H \text{CFL}_H. \quad (\text{A.12})$$

The unchanged factor of two is a consequence of two compensating changes. In two dimensions the elliptic diagonal doubles from $2/h^2$ to $4/h^2$, but the canonical transport discretization also divides the two upstream contributions by $2h$. Each upstream SORh correction therefore has coefficient $\omega_H/2$, which matches the AOR coefficient $r_A/4$ under

the same map $r_A = 2\omega_H$ found in one dimension. At $\text{CFL}_H = 1$, substituting $\delta_{i-1,j}^{k+1} = \phi_{i-1,j}^{k+1} - \phi_{i-1,j}^k$ and the corresponding south correction into Eq. (A.11) cancels the old west and south values in the five-point residual. The result is

$$\phi_{i,j}^{k+1} = \phi_{i,j}^k + \frac{\omega_H}{2} (\phi_{i-1,j}^{k+1} + \phi_{i,j-1}^{k+1} + \phi_{i+1,j}^k + \phi_{i,j+1}^k - 4\phi_{i,j}^k - h^2 f_{i,j}). \quad (\text{A.13})$$

Classical SOR has the same bracket multiplied by $\alpha/4$, so $\alpha = 2\omega_H$. In iteration-matrix terms, Eq. (A.8) then has $r_A = \omega_A = \alpha$ and becomes the classical SOR matrix

$$G_{\text{SOR}} = (D - \alpha L)^{-1} [(1 - \alpha)D + \alpha U]. \quad (\text{A.14})$$

For $\text{CFL}_H \neq 1$, the classical-SOR reduction does not occur. Other grids, stencils, transport normalizations, or sweep orderings require a new coefficient comparison.

A.3 Pseudo-Time Iteration Algorithm

For reference, Algorithm 1 shows the general structure of a full experimental solver run for the two-dimensional SORh and SORp methods. Specific definitions of parameters used in numerical experiments are defined in Chapter 4 and 5. For SORh, upstream values from $\phi_{i,j}^{k+1}$ are used by neighboring downstream values during the sweep, but the previous iterate $\phi_{i,j}^k$ is frozen throughout the sweep. The update could separate the computation of provisional update from the relaxation step, but these steps are combined to simplify the implementation.

Algorithm 1 Pseudo-time iteration (SORh)

Require: Grid, source f_h , boundary data, initial field ϕ^0 , Δt , relaxation parameter ω , tolerances ε_u , ε_r , and maximum iteration count $k_{\max}$

Ensure: Approximate solution ϕ

- 1: Initialize $\phi \leftarrow \phi^0$
 - 2: Impose the boundary data on ϕ^0
 - 3: **for** $k = 0, \dots, k_{\max} - 1$ **do**
 - 4: **for all** $(i, j) \in \mathcal{I}_h$ in the prescribed sweep order **do**
 - 5: Compute the relaxed value $\phi_{i,j}^{k+1}$
 - 6: **end for**
 - 7: Reimpose or verify the boundary data
 - 8: Evaluate e_u^{k+1} and e_r^{k+1}
 - 9: **if** $e_r^{k+1} \leq \varepsilon_r$ and $e_u^{k+1} \leq \varepsilon_u$ **then**
 - 10: **return** ϕ^{k+1} ▷ Convergence criteria met
 - 11: **end if**
 - 12: **end for**
 - 13: **return** $\phi^{k_{\max}}$ ▷ Maximum iteration count reached
-

Source Code Availability

The source code, experiment scripts, and plotting tools used in this thesis are available at <https://github.com/slug-cfd/sorh-poisson>.

Bibliography

- [1] Wenbo Y. Anninos and Michael L. Norman. “Nonlinear Hydrodynamics of Cosmological Sheets. I: Numerical Techniques and Tests”. In: *The Astrophysical Journal* 429 (1994), pp. 434–464. DOI: 10.1086/174335.
- [2] G. Avdelas and Apostolos Hadjidimos. “Optimum Accelerated Overrelaxation Method in a Special Case”. In: *Mathematics of Computation* 36.153 (1981), pp. 183–187. DOI: 10.1090/S0025-5718-1981-0595050-5.
- [3] Jérôme Bonelle and Alexandre Ern. “Analysis of Compatible Discrete Operator Schemes for Elliptic Problems on Polyhedral Meshes”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 48.2 (2014), pp. 553–581. DOI: 10.1051/m2an/2013104.
- [4] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A Multigrid Tutorial*. 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2000. DOI: 10.1137/1.9780898719505.
- [5] Jean-Jacques Chattot. *Computational Aerodynamics and Fluid Dynamics: An Introduction*. 1st ed. Scientific Computation. Berlin, Heidelberg: Springer, 2002. DOI: 10.1007/978-3-662-05064-4.
- [6] Rustum Choksi. *Partial Differential Equations: A First Course*. Vol. 54. Pure and Applied Undergraduate Texts. Providence, Rhode Island: American Mathematical Society, 2022. ISBN: 978-1-4704-6491-2.

- [7] Anshu Dubey et al. “Extensible Component-Based Architecture for FLASH, a Massively Parallel, Multiphysics Simulation Code”. In: *Parallel Computing* 35.10–11 (2009), pp. 512–522. DOI: 10.1016/j.parco.2009.08.001.
- [8] Lawrence C. Evans. *Partial Differential Equations*. 2nd ed. Vol. 19. Graduate Studies in Mathematics. Providence, Rhode Island: American Mathematical Society, 2010. DOI: 10.1090/gsm/019.
- [9] Stanley P. Frankel. “Convergence Rates of Iterative Treatments of Partial Differential Equations”. In: *Mathematical Tables and Other Aids to Computation* 4.30 (1950), pp. 65–75. DOI: 10.1090/S0025-5718-1950-0046149-3.
- [10] Bruce Fryxell et al. “FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes”. In: *The Astrophysical Journal Supplement Series* 131.1 (2000), pp. 273–334. DOI: 10.1086/317361.
- [11] Apostolos Hadjidimos. “Accelerated Overrelaxation Method”. In: *Mathematics of Computation* 32.141 (1978), pp. 149–157. DOI: 10.1090/S0025-5718-1978-0483340-6.
- [12] Apostolos Hadjidimos. “Successive Overrelaxation (SOR) and Related Methods”. In: *Journal of Computational and Applied Mathematics* 123.1–2 (2000), pp. 177–199. DOI: 10.1016/S0377-0427(00)00403-9.
- [13] Apostolos Hadjidimos and A. Yeyios. “The Principle of Extrapolation in Connection with the Accelerated Overrelaxation Method”. In: *Linear Algebra and its Applications* 30 (1980), pp. 115–128. DOI: 10.1016/0024-3795(80)90187-1.
- [14] Tomoyuki Hanawa. “4th-Order Accurate Discrete Form of the Poisson Equation for the Adaptive Mesh Refinement”. In: *Journal of Physics: Conference Series* 3252.1 (2026), p. 012005. DOI: 10.1088/1742-6596/3252/1/012005.
- [15] Tomoyuki Hanawa and Patrick Dean Mullen. “Toward Higher-Order Accuracy in Self-Gravitating Hydrodynamics”. In: *The Astrophysical Journal Supplement Series* 277.2 (2025), p. 53. DOI: 10.3847/1538-4365/adb97b.

- [16] Jingfang Huang and Leslie Greengard. “A Fast Direct Solver for Elliptic Partial Differential Equations on Adaptively Refined Meshes”. In: *SIAM Journal on Scientific Computing* 21.4 (2000), pp. 1551–1566. DOI: 10.1137/S1064827598346235.
- [17] Dongwook Lee. *Fifth-Order Accurate Multidimensional Finite Volume MHD Scheme Based on Divergence-Cleaning*. Abstract presented at ASTRONUM 2026. Conference abstract describing the SORh projection method. 2026. URL: <https://irfu.cea.fr/ASTRONUM2026/ABSTRACT-BOOK-2026.pdf>.
- [18] Dongwook Lee and Jonathan Guité. *A hyperbolic-variant successive over-relaxation Poisson solver for divergence cleaning in a cell-centered fifth-order finite volume MHD scheme*. Submitted to the Journal of Computational Physics.
- [19] Randall J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. Philadelphia: Society for Industrial and Applied Mathematics, 2007. DOI: 10.1137/1.9780898717839.
- [20] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002. DOI: 10.1017/CB09780511791253.
- [21] Hiroaki Nishikawa. “A First-Order System Approach for Diffusion Equation. I: Second-Order Residual-Distribution Schemes”. In: *Journal of Computational Physics* 227.1 (2007), pp. 315–352. DOI: 10.1016/j.jcp.2007.07.029.
- [22] Hiroaki Nishikawa. “First-, Second-, and Third-Order Finite-Volume Schemes for Diffusion”. In: *Journal of Computational Physics* 256 (2014), pp. 791–805. DOI: 10.1016/j.jcp.2013.09.024.
- [23] Donald W. Peaceman and Henry H. Rachford Jr. “The Numerical Solution of Parabolic and Elliptic Differential Equations”. In: *Journal of the Society for Industrial and Applied Mathematics* 3.1 (1955), pp. 28–41. DOI: 10.1137/0103003.

- [24] Paul M. Ricker. “A Direct Multigrid Poisson Solver for Oct-Tree Adaptive Meshes”. In: *The Astrophysical Journal Supplement Series* 176.1 (2008), pp. 293–300. DOI: 10.1086/526425.
- [25] Gilbert Strang. “On the Construction and Comparison of Difference Schemes”. In: *SIAM Journal on Numerical Analysis* 5.3 (1968), pp. 506–517. DOI: 10.1137/0705041.
- [26] Eleuterio F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. 3rd ed. Berlin, Heidelberg: Springer, 2009. DOI: 10.1007/b79761.
- [27] Richard S. Varga. *Matrix Iterative Analysis*. 2nd ed. Vol. 27. Springer Series in Computational Mathematics. Berlin, Heidelberg: Springer, 2000. DOI: 10.1007/978-3-642-05156-2.
- [28] Yakov B. Zel’dovich. “Gravitational Instability: An Approximate Theory for Large Density Perturbations”. In: *Astronomy and Astrophysics* 5 (1970), pp. 84–89.