

**High-Order Finite Difference Methods for Wave
Equations**

by

Michelle Lynn Ghrist

M.S. in Applied Mathematics, University of Colorado,
Boulder, 1997

B.S. in Mathematics, University of Toledo, 1994

B.S. in Physics, University of Toledo, 1994

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Applied Mathematics
2000

This thesis entitled:
High-Order Finite Difference Methods for Wave Equations
written by Michelle Lynn Ghrist
has been approved for the Department of Applied Mathematics

Bengt Fornberg

Steve McCormick

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Ghrist, Michelle Lynn (Ph. D., Applied Mathematics)
High-Order Finite Difference Methods for Wave Equations
Thesis directed by Professor Bengt Fornberg

We have investigated the very high computational efficiency of high-order finite difference methods, especially as they incorporate features such as implicitness (also known as compactness in the literature) and grid staggering. While remaining relatively compact, these methods can approach the superior accuracy and effectiveness of spectral methods while still allowing some boundary flexibility. In the past, grid staggering has been observed to be beneficial in some cases (e.g. the Yee scheme for computational electrodynamics), but that idea has been shown here to combine favorably with both implicitness and high orders of accuracy.

In addition, we have explored the new idea of grid staggering for time integrators. In the important application of solving linear wave equations (e.g. acoustic or elastic waves equations, or Maxwell's equations for electromagnetic fields), nearly an order of magnitude gain can usually be achieved in accuracy (for the same computational cost in both operation count and in memory) compared to classical ODE solvers such as Adams or Runge-Kutta methods. In addition, our new staggered methods have superior stability properties to the classical methods in the context of solving wave equations. We investigate the accuracy and stability of these methods analytically, experimentally, and through the use of a novel root portrait technique. We also consider several theoretical questions concerning these staggered time integrators.

Acknowledgements

I wish to express my great appreciation and gratitude to certain individuals and organizations who helped make this research possible.

Support for this research was provided by NASA-Lewis Research Center and the University of Colorado-Boulder through a National Physical Science Consortium Fellowship (NASA grant NPSC-OCG1035B) and by the National Science Foundation (NSF traineeship grant DMS-9256335). I am especially grateful to Director Nan Snow of NPSC for her unwavering support and to Ms. Bonnie McBride of NASA-Lewis for her influence and kindness. Additional funding was provided by a Dorothy Stribic fellowship.

I wish to thank my advisor, Professor Bengt Fornberg, for his leadership, enthusiasm, support, and practicality. His sense of humor and willingness to explain things as many times as needed were indispensable. I am also grateful for the indispensable help that post-doc Dr. Toby Driscoll provided through the years. His computer expertise and ability to explain complex ideas in simple language was invaluable. My fellowship mentor, Dr. Anne Dougherty, deserves my great appreciation for her leadership, excellent teaching, patience, and willingness to offer advice. This would have been a much more difficult journey without her assistance. I also thank Professor Ben Herbst of South Africa for his guidance on a proof and thank the rest of my committee members for their support and help: Professor Ellen Zweibel, Professor Patrick Weidman, Professor Steve McCormick, and Professor Bob Easton.

Lastly, I offer my extreme gratitude to my husband Rich, without whom this work would not have been possible, and to my son Aaron, for enduring my work.

Contents

Chapter	
1	Introduction 1
1.1	Introduction 1
1.2	Glossary of abbreviations and terms used in this paper 4
2	Spatial Finite Difference Approximations for Wave Equations 6
2.1	Introduction 6
2.2	Illustrations of grid staggering 9
2.3	Algorithm for finite difference weights and examples 9
2.4	Tables of some weights, including formulae for weights and for limits of infinite order 16
2.5	Equivalence between implicit and explicit formulae in the case of limiting order 23
2.6	Operation counts 27
2.7	Comparison of accuracies and cost-effectiveness 28
2.8	Test problem 33
2.9	Conclusions 35
3	Staggered Time Integrators for Wave Equations 41
3.1	Introduction 41
3.2	Illustrations of grid staggering for linear wave equations 43
3.3	Preliminaries 47

3.3.1	Definitions	47
3.3.2	Maximum Imaginary Stability Boundary	49
3.4	Staggered Adams–Bashforth and backwards differentiation methods	50
3.5	Staggered free parameter multistep methods	56
3.5.1	Fourth order free parameter method: $\frac{7}{2}$ -step, one parameter .	57
3.5.2	Fourth order free parameter method: 4-step, two parameter .	59
3.6	Theoretical Considerations	60
3.6.1	Imaginary Stability Boundary of Adams–Bashforth methods	61
3.6.2	Staggered analogue of Dahlquist’s First Stability Barrier . . .	61
3.7	Staggered Predictor-Corrector Methods	61
3.8	Staggered Runge–Kutta methods	62
3.8.1	Advancing u and v separately	63
3.8.2	Advancing u and v together	67
3.9	Root portraits	69
3.10	Numerical experiments	72
3.11	Conclusions	82
4	Conclusions	84
	Bibliography	86
	Appendix	
A	Explanation of the Padé algorithm given in Section 2.3	90
B	Derivation of the limit expressions in Table 2.8	92
C	Conversions between Kopal’s coefficients and those in Chapter 2	94
D	Staggered differentiation matrix	96

E	Derivation of equation (2.7)	99
E.1	Derivation of the expression for d_k	99
E.2	Verification of the expression for d_k : 5-diagonal case	101
F	Defining the local truncation error of time integrators	102
F.1	Discussion	102
F.2	Examples	104
F.2.1	Adams-type methods	104
F.2.2	Backwards differentiation methods	104
F.2.3	Free Parameter Methods	105
F.3	Conclusions	105
G	Nonstaggered free parameter methods	106
H	Proof of results found in Section 3.6.1 concerning the stability ordinates of AB, ABS, and AM methods	109
H.1	Proof of Theorem H.1: nonstaggered AB methods	112
H.2	Proof of Theorem H.2: ABS methods	114
H.3	Proof of Theorem H.3: AM methods	117
I	Proof of the staggered analogue of Dahlquist's first barrier	119
I.1	Case 1: m a half-integer	121
I.2	Case 2: m an integer	124

Figures

Figure

2.1	Schematic illustration of composite grid concept	8
2.2	(a)-(b) Illustrations of spatial staggering for some linear wave equations: one-dimensional acoustic wave equation and two-dimensional elastic wave equation	11
2.2	(c) Illustration of spatial staggering for three-dimensional Maxwell's equations	12
2.3	Illustration of notation used in the Padé weight algorithm	14
2.4	Notation used to index entries in nonstaggered and staggered spatial stencils	15
2.5	Equivalent weights comparison	26
2.6	Fourier multiplication factors for different methods	32
2.7	Deviation of Fourier multiplication factors from the ideal straight line, compared by operation count	34
2.8	Regular grid solutions at $t = 100$ using different spatial approximations	36
2.9	Same as Figure 2.8, but using staggered approximations for the spatial derivative	37
2.10	Solutions for staggered grid at $t = 2000$ with grid sizes selected to make computations equally time-consuming in 2-D	38
2.11	Solutions at $t = 2000$ for the staggered explicit $n = 1$ (Yee-type) scheme for increasingly fine grids	39

3.1	Representative samples of various spatial/time grid layouts for the one-dimensional wave equation	45
3.2	Representative sample spatial grid layouts for the two-dimensional elastic wave equation	46
3.3	Representative sample of a spatial-staggered, time-staggered grid for the two-dimensional elastic wave equation	46
3.4	Four representations of ABS3	51
3.5	Trade-off between accuracy and stability for fourth order staggered free parameter method	58
3.6	Stability domain of method (3.17) for $\beta_1 = 0.126$	58
3.7	Stability domains of the fourth order staggered free parameter scheme (3.18)	60
3.8	Stability domains for staggered Runge–Kutta methods	68
3.9	Example of a root portrait	71
3.10	Root portraits for classical and staggered methods of different orders	73
3.11	Sample run of ABS3 using the physically relevant root and $N = 375$ function evaluations	76
G.1	Trade-off between accuracy and stability for fourth order nonstaggered free parameter method (G.1)	108
G.2	Stability domain of method (G.1) for $t = -0.41$	108

Tables

Table

2.1	Examples of finite difference approximations	10
2.2	Weights for explicit, regular grid FD formulae	17
2.3	Weights for implicit 3-diagonal, regular grid FD formulae	18
2.4	Weights for implicit 5-diagonal, regular grid FD formulae	19
2.5	Weights for explicit, staggered grid FD formulae	19
2.6	Weights for implicit 3-diagonal, staggered grid FD formulae	20
2.7	Weights for implicit 5-diagonal, staggered grid FD formulae	21
2.8	Limits of weights as $n \rightarrow \infty$	22
2.9	Operation count to calculate f' at one grid point	29
2.10	Coefficients of leading error terms for different first derivative approx- imations	29
2.11	Approximate ratio of leading error coefficients : staggered to nonstag- gered	30
3.1	Staggered backwards differentiation time integrators	52
3.2	Staggered Adams-Bashforth time integrators	53
3.3	Nonstaggered Adams-Bashforth time integrators	54
3.4	Error given by running the second order leapfrog method using the root portrait technique	77

3.5	Error given by running third order methods using the root portrait technique	78
3.6	(a) Error given by running fourth order nonstaggered methods using the root portrait technique	79
3.6	(b) Error given by running fourth order staggered methods using the root portrait technique	80
3.7	Error given by running seventh order methods using the root portrait technique	81
3.8	Error given by running eighth order methods using the root portrait technique	81
3.9	Comparison of fourth order time integrators: nonstaggered vs. staggered	83

Chapter 1

Introduction

This thesis documents two research projects which have been carried out at the University of Colorado at Boulder in collaboration with my supervisor, Professor Bengt Fornberg, and NSF (later NSF-VIGRE) post-doc Dr. Tobin A. Driscoll.

The two projects are described in Chapters 2 and 3:

2. Spatial finite difference approximations for wave equations
3. Staggered time integrators for wave equations.

Although it is difficult to identify individual contributions in collaborative projects, an attempt will be made in the introduction of each chapter to outline my contributions to each project.

1.1 Introduction

In many situations, finding an analytic solution to a partial differential equation or a system of such equations is unrealistic or even impossible; numerical methods that utilize computer algorithms are then used to find approximate solutions. The focus of our research has been to produce new schemes for numerically solving linear wave-type equations (such as Maxwell's equations, acoustic wave equations, and elastic wave equations) that are more accurate, less computationally intensive, and/or easier to implement than currently used industry standards such as the second order (in both space and time) finite-difference Yee scheme. Computational cost becomes especially important as the number of equations in the system

increases.

Finite difference methods utilize linear combinations of function values on a discrete grid to approximate derivative values. These methods have been known for many years and several newer methods have been found since then (e.g. finite elements, spectral, and pseudospectral methods); thus, one might question why we are revisiting what to some might seem to be an outdated field. In the context of composite grid methods, however, finite difference methods play a key role. For example, in one model, block pseudospectral methods [1,2] are utilized on boundaries and interfaces as well as in other regions of high activity, while finite difference methods are used for the background grid which overlaps with these pseudospectral strips. Simple, relatively low-cost finite difference methods are used on the majority of the grid, producing a novel composite method that combines high accuracy with low computational cost that is still applicable to complex geometries. The specific focus of this research was to explore and improve the finite difference methods and time stepping methods to be used for this background grid.

The first goal of this research project was to investigate the very high computational efficiency of high-order finite difference methods, especially as they incorporate features such as implicitness and grid staggering. Chapter 2 presents our analysis of finite difference approximations for the first derivative in terms of accuracy and computational cost, considering both explicit and implicit (also known as compact in the literature) schemes on regular and staggered grids. While both implicit stencils and staggered grids have been considered before, this is the first study to combine these concepts and to thoroughly analyze them in terms of computational cost. Though the schemes remain relatively compact, they can approach the superior accuracy and effectiveness of spectral methods while still allowing some boundary flexibility. Grid staggering has been observed in the past to be beneficial in some cases (e.g. the Yee scheme for computational electromagnetics), but our work has shown that staggering combines quite favorably with both implicitness and

high orders of accuracy, features that are lacking in the Yee scheme and in most other previous applications of staggered grids.

The idea to also use grid staggering for time integrators, the topic of Chapter 3, is entirely new. We have explored variations of the Adams-Bashforth, backwards differentiation, and Runge-Kutta families of time integrators to solve systems of linear wave equations on uniform, time-staggered grids. In the important application of solving linear wave equations (e.g. acoustic or elastic waves, or Maxwell's equations for electromagnetics), nearly an order of magnitude gain can be achieved in accuracy (for the same computational cost, in both operation count and memory) compared to classical nonstaggered time integrators. In addition, the stability properties of these new staggered methods are superior to the classical methods. We investigate the accuracy and stability of these new classes of methods analytically, experimentally, and through the use of a novel root portrait technique. In addition, several key theoretical results concerning staggered time integrators are given, including a generalization of Dahlquist's First Stability Barrier for staggered methods.

Our results have already impacted the field of wave computations. For example, several research groups (e.g. at Uppsala University and Brown University) currently employ our staggered time integrators for major industrial electromagnetics calculations. In addition, Weidlinger Associates (in Los Altos, CA) has implemented our methods to simulate effects relevant to medical ultrasound, with one long-term goal being to replace X-rays with ultrasound in areas such as mammography. Another potential application for our schemes is forward seismic modeling for hydrocarbon exploration.

1.2 Glossary of abbreviations and terms used in this paper

Because we use a number of acronyms which may be unfamiliar to the reader, a glossary of these abbreviations is included here.

AB p	Adams–Bashforth method of order p
ABS p	Staggered Adams–Bashforth method of order p
BD p	Backwards differentiation method of order p
BDS p	Staggered backwards differentiation method of order p
error constant	Coefficient C that gives an estimate of local truncation

error to be expected from a method; the local truncation error is given by $Ck^{p+1}f^{(p+1)}(\xi)$, where p is the order of the method.

(To obtain an adequate global error estimate, normalize the error constant by $\sigma(1)$ for multistep methods. To obtain a valid comparison, multiply the error constant by s^p for an s -stage method.)

explicit	For spatial FD approximations, a method where one unknown derivative value is given by a linear combination of known function values. For FD time integrators, a method where the unknown (future) function value is given in terms of known function and derivative values.
----------	---

FD	Finite difference
FDTD	Finite-difference time-domain (see [27])
FPS p	Staggered free parameter method of order p

implicit	In spatial FD approximations, a method where linear combinations of unknown derivative values are given in terms of linear combinations of known function values. For time integrators, a FD approximation where the unknown (future) function value includes a term involving an unknown (future) derivative value.
ISB	Imaginary Stability Boundary - the largest (real) value of S_I such that the interval $[-iS_I, iS_I]$ is contained in the stability domain of a time-stepping scheme. (For an s -stage RK method, normalize the ISB by s .)
ODE	Ordinary differential equation
Padé	Approximation of a given function by a rational function (by matching as many derivatives as possible at some point)
PDE	Partial differential equation
PS	Pseudospectral
RK p	Runge–Kutta method of order p
RKSp	Staggered Runge–Kutta method of order p
Stability domain	For a given time integrator, the set of λk values in the complex plane that give stable solutions for the problem $y' = \lambda y$ (k is the step size).
Toeplitz	A Toeplitz matrix is constant along all diagonals, i.e. there exist numbers $\alpha_{-d+1}, \alpha_{-d+2}, \dots, \alpha_0, \dots, \alpha_{d-1}$ such that $a_{k,l} = \alpha_{k-l}$ for $k, l = 1, 2, \dots, d$.

Chapter 2

Spatial Finite Difference Approximations for Wave Equations

The simplest finite difference approximations for spatial derivatives are centered and explicit and are applied to ‘regular’ equispaced grids. Well-established generalizations include the use of implicit (compact) approximations and staggered grids. In this chapter, we find that the combination of these two concepts, together with high formal order of accuracy, is very effective for approximating the first derivatives in space that occur in many wave-type PDEs.

2.1 Introduction

Linear wave equations, especially in two or more dimensions, are often formulated as first order systems. First order formulations tend to better reflect the physics of the problem and often allow for easier implementation of boundary conditions. We are not aware of any linear wave equations which cannot be rewritten as first order systems. Thus, in this chapter, we will consider only approximations of first derivatives.

The primary requirements for numerical approximations of spatial first derivatives are

- (1) high accuracy,
- (2) low operation count,

and the overall resulting method should feature

(3) compatibility with curved interfaces and non-reflecting far field boundary conditions.

Several numerical approaches excel in one or sometimes two of these respects; for example, finite element methods have difficulty achieving high accuracy without a large operation count, while spectral and pseudospectral methods are often incompatible with curved interfaces.

We consider the use of composite methods, where high-order interface techniques are used along material discontinuities and boundaries [6, 7] and finite difference methods (for equispaced grids) are used on the remaining areas of the computational domain. This combination approach meets all three requirements and can achieve near-spectral accuracy requiring only about 4-5 points per wavelength and about 6-8 arithmetic operations for each spatial derivative at each grid point, with full spectral accuracy maintained at interfaces. Figure 2.1 illustrates schematically how a composite grid for this approach can be structured in the case of coupling of different media in a 2-D calculation for Maxwell's equations.

This chapter focuses on the problem of obtaining high accuracy economically on the “background” grid by the use of finite difference methods. This background grid overlaps at its (usually jagged) edges with a strip following the interface on which we run a block pseudospectral method, for example. Results on computations with this composite setup are reported in [7]. Note that because of this intended usage, we do not discuss here the implementation of traditional boundary conditions.

The main topics of the remaining Sections 2-9 are as follows:

2. Illustrations of grid staggering
3. Simple symbolic algebra code for calculating weights of finite difference stencils
4. Tables of weights; formulae for weights, including limits of infinite order
5. Observation that in the limit of increasing order, implicit and explicit formulae become equivalent in terms of how a derivative value depends on function values

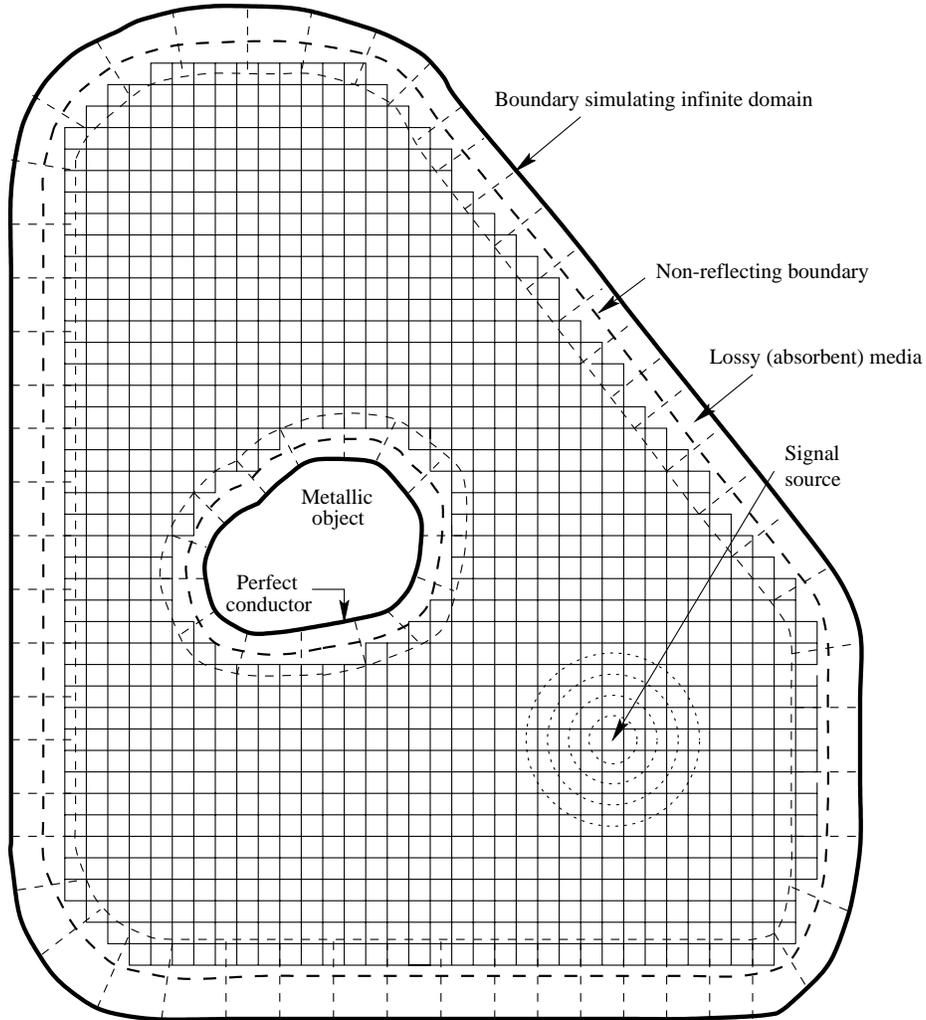


Figure 2.1: Schematic illustration of composite grid concept. A block pseudospectral (BPS) method would be run in the curved strip domains (dashed grid) and a high-order finite difference (FD) method would be run in the overlapping equispaced background grid (solid grid). Implementation of strips with media interfaces is described in [7].

For this case, we imagine solving Maxwell's equations in a case where a metallic object (coated with an absorbing medium) is illuminated with a radar source. Around the outer boundary is wrapped a strip-like domain which, along its middle, features a perfectly non-reflecting boundary between the main dielectric (e.g. vacuum) and a strong signal absorber. The grid densities (especially in the strips around the object and the boundary) would in general be considerably higher than shown.

- 6. Operation counts
- 7. Comparisons of accuracy and cost-effectiveness
- 8. Test example
- 9. Summary

The author made significant contributions to sections 4 and 5. Some contributions were made to sections 6, 7, and 8.

2.2 Illustrations of grid staggering

Most linear wave equations of general interest (in any number of space dimensions) have only a few of all possible spatial derivatives present; these appear in such a way that spatial grid staggering becomes straightforward. We observe that the structure of the governing equations turns out to be precisely such that a unique and internally conflict-free staggering arrangement is possible, but we are unaware of any discussion of this issue in the literature. Figures 2.2 a-c illustrate spatial staggering in three representative cases: 1-D acoustic, 2-D elastic, and 3-D Maxwell's equations. In each case, we contrast two grid layouts, regular vs. staggered, both featuring the same density of grid data. The ease of creating staggered layouts for all major linear wave equations makes the analysis of this chapter widely applicable.

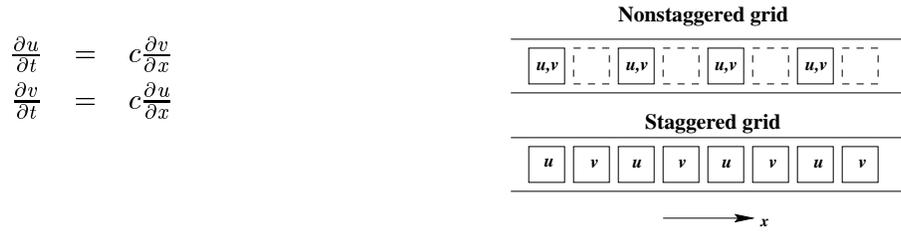
One can also stagger in time, as is done, for example, in the Yee scheme for time-domain computational electrodynamics [22, 27, 29]. Chapter 3 discusses higher-order time staggering.

2.3 Algorithm for finite difference weights and examples

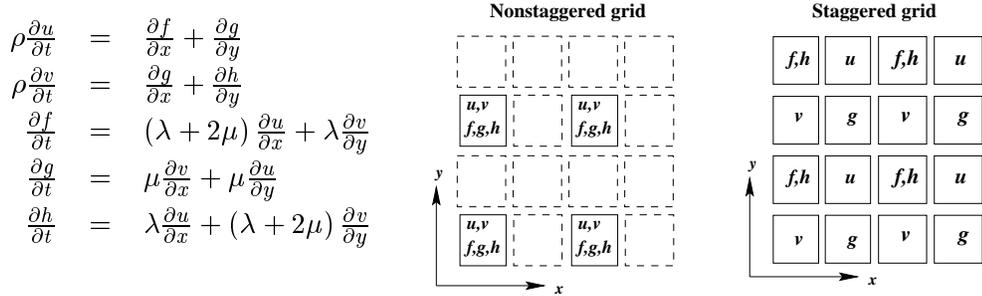
Table 2.1 gives examples of the simplest first derivative approximations for each of the four stencil types considered here: explicit and implicit approximations on regular and staggered grids. Explicit approximations relate one derivative value

Table 2.1: Examples of finite difference approximations (NS = nonstaggered, S=staggered)

Approximation and grid type	Lowest order stencil of given type	Leading error term		Parameters for weights			Reference Table	Stencil picture (See Figure 2.3)
		error term	code	m	s	d		
Explicit NS	$f'(x)$	$\frac{1}{h}f(x-h) + \frac{1}{2}f(x+h)$	$\frac{1}{6}h^2f^{(3)}(x)$	1	1	0	2.2 (n=1)	
Implicit NS	$\frac{1}{6}f'(x-h) + \frac{2}{3}f'(x) + \frac{1}{6}f'(x+h)$	$-\frac{1}{2}f(x-h) + \frac{1}{2}f(x+h)$	$-\frac{1}{180}h^4f^{(5)}(x)$	1	0	2	2.3 (n=1)	
Explicit S	$f'(x)$	$-f(x-\frac{h}{2}) + f(x+\frac{h}{2})$	$\frac{1}{24}h^2f^{(3)}(x)$	1	$\frac{1}{2}$	0	2.5 (n=1)	
Implicit S	$\frac{1}{24}f'(x-h) + \frac{11}{12}f'(x) + \frac{1}{24}f'(x+h)$	$-f(x-\frac{h}{2}) + f(x+\frac{h}{2})$	$-\frac{17}{3150}h^4f^{(5)}(x)$	1	$-\frac{1}{2}$	2	2.6 (n=1)	



where $u, v =$ velocity, pressure



where $u, v =$ velocities in x - and y - directions

$f, g, h =$ x -compression, shear, and y -compression

$\rho, \lambda, \mu =$ density and elastic constants

Figure 2.2: Illustrations of spatial staggering for some linear wave equations: (a) One-dimensional acoustic wave equation and (b) Two-dimensional elastic wave equation.

$$\begin{aligned}
\frac{\partial E_x}{\partial t} &= \frac{1}{\epsilon} \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right) \\
\frac{\partial E_y}{\partial t} &= \frac{1}{\epsilon} \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right) \\
\frac{\partial E_z}{\partial t} &= \frac{1}{\epsilon} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) \\
\frac{\partial H_x}{\partial t} &= \frac{1}{\mu} \left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \right) \\
\frac{\partial H_y}{\partial t} &= \frac{1}{\mu} \left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \right) \\
\frac{\partial H_z}{\partial t} &= \frac{1}{\mu} \left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right)
\end{aligned}$$

where E_x, E_y, E_z = components of the electric field

H_x, H_y, H_z = components of the magnetic field

μ, ϵ = permeability, permittivity

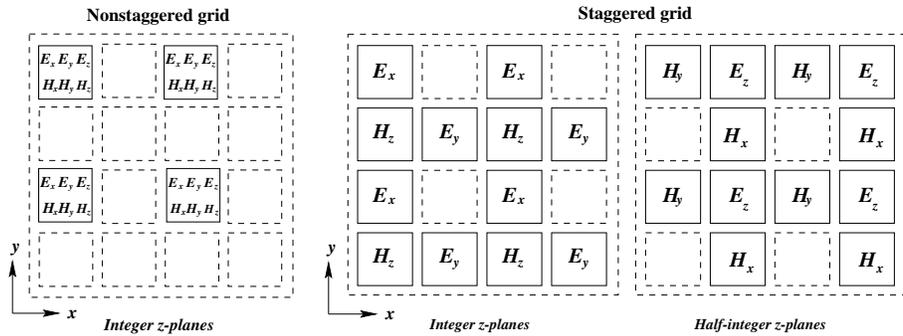


Figure 2.2: (c) Illustration of spatial staggering for three-dimensional Maxwell's equations

to a linear combination of function values, while implicit expressions relate a linear combination of derivative values to a linear combination of function values, thus requiring one to solve a linear system to find the desired derivative values.

As discussed in [11], the weights in any of the stencils we discuss in this paper can be calculated by the 2-line Mathematica algorithm

```
t=Padé[x^s*Log[x]^m,{x,1,n,d}];
{CoefficientList[Denominator[t],x],CoefficientList[Numerator[t],x]/h^m}
```

or in Maple by

```
t:=pade (x^s*ln(x)^m,x=1,[n,d]):
coeff (expand (denom(t)),x,i)      $i=0..d;
coeff (expand (numer(t)),x,i)/h^m  $i=0..n;
```

In both cases, a Padé package must be pre-loaded; this is done with the commands `<<Calculus'Padé'` or with `(numapprox)`: respectively. In these lines of code, `m` denotes which order of derivative we wish to approximate (this will be one in all cases considered in this paper but can, in general, be any nonnegative integer; the case `m=0` will generate interpolation formulae). The remaining three input parameters `s`, `d`, and `n` describe the shape of the stencil, as illustrated in Figure 2.3 (`s` may be any real number; `d` and `n` must be non-negative integers).

For the first derivative (i.e. `m = 1`) and with the stencil shown in Figure 2.3, the Mathematica output becomes

$$\left\{ \left\{ \frac{9}{80}, \frac{31}{40}, \frac{9}{80} \right\}, \left\{ -\frac{17}{240h}, -\frac{63}{80h}, \frac{63}{80h}, \frac{17}{240h} \right\} \right\}. \quad (2.1)$$

(cf. the case `n = 2` listed in Table 2.6).

In Appendix A we explain why the above Padé algorithm works.

Hereafter in this chapter, we consider only first derivative approximations, and we use `m`, `n`, and `k` to denote stencil entries as illustrated in Figure 2.4. Note

that $\mathbf{d} = 2m + 1$, while $\mathbf{n} = 2n + 1$ for nonstaggered grids and $\mathbf{n} = 2n$ for staggered grids.

In addition, note that we consider only centered FD approximations. In general, centered schemes have better accuracy properties than noncentered schemes; one usually only considers noncentered schemes when considering boundaries.

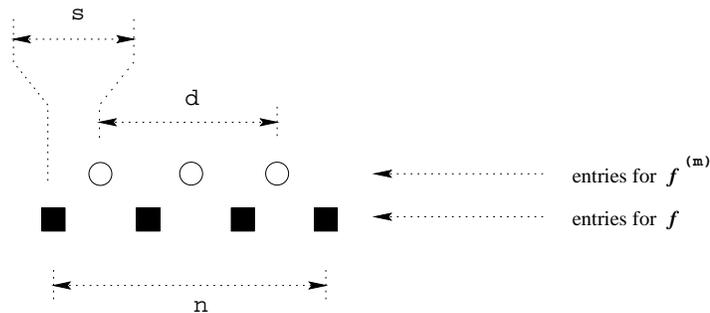


Figure 2.3: Schematic illustration of the notation used in the Padé weight algorithm for a staggered case; here, $\mathbf{s} = \frac{1}{2}$, $\mathbf{d} = 2$, and $\mathbf{n} = 3$. (All distances \mathbf{s} , \mathbf{d} , \mathbf{n} are in units of the unit step length h .)

The notation in this and subsequent illustrations of stencils follows the convention (as was adopted for example in [10]):

unknown / known

○ / ● derivative entry

□ / ■ function entry

The symbols are unfilled or filled depending on whether the corresponding derivative or function value would be unknown (i.e. to be solved for) or known in the anticipated application of the stencil.

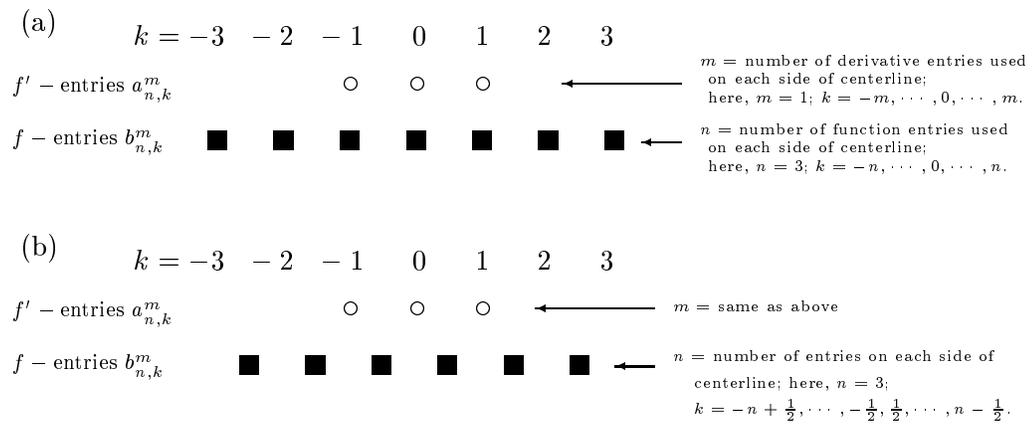


Figure 2.4: Notation used to index entries in nonstaggered and staggered spatial stencils: (a) Notation for nonstaggered grid, (b) Notation for staggered grid.

2.4 Tables of some weights, including formulae for weights and for limits of infinite order

Tables 2.2-2.7 provide numerical values and closed-form expressions (in terms of n and k) for the coefficient weights of finite difference schemes for the cases of main interest. From these follow the quoted limits for k fixed and $n \rightarrow \infty$. These closed-form expressions and limits were found primarily by the author with some assistance from Bengt Fornberg. We note the use of Wallis' product in this limiting procedure:

$$\lim_{k \rightarrow \infty} k \left[\frac{(2k-1)!!}{(2k)!!} \right]^2 = \frac{1}{2} \prod_{j=1}^{\infty} \left(1 - \frac{1}{(2j)^2} \right) = \frac{1}{\pi} \quad (2.2)$$

(See [12], for example.)

The limiting weights can also be derived from a more general integral formulation, which is given in Table 2.8. The integrals given in this table can be evaluated explicitly. The regular grid expressions in Table 2.8 were derived by Bengt Fornberg while the staggered grid expressions were derived by the author. Appendix B gives a brief argument leading to the limit expressions in Table 2.8.

The literature on both regular and staggered grid explicit finite difference schemes is extensive. Some schemes have been designed with the goal of enhancing the accuracy for certain frequencies rather than maximizing the formal order of accuracy [16, 20, 24]. Implicit (also known as compact in the literature) regular schemes have been derived and studied on numerous occasions, e.g. [1, 4, 14, 15, 23, 25]. Kopal [21] presents tables which allow easy calculation of weights in numerous schemes which include cases that combine staggering with implicitness (also known in the literature as compactness); however, his coefficients are presented in terms of difference operators whereas we give explicit closed-form expressions for the coefficients and also consider limiting cases (in Tables 2.2-2.7). (In Appendix C, we demonstrate how to convert from Kopal's coefficients to our coefficients.) However,

Table 2.2: Weights for explicit, regular grid FD formulae (order of accuracy = $2n$).

		weights $b_{n,k}^0$ for f										
n	$k =$	-5	-4	-3	-2	-1	0	1	2	3	4	5
1						$-\frac{1}{2}$	0	$\frac{1}{2}$				
2					$\frac{1}{12}$	$-\frac{2}{3}$	0	$\frac{2}{3}$	$-\frac{1}{12}$			
3				$-\frac{1}{60}$	$\frac{3}{20}$	$-\frac{3}{4}$	0	$\frac{3}{4}$	$-\frac{3}{20}$	$\frac{1}{60}$		
4			$\frac{1}{280}$	$-\frac{4}{105}$	$\frac{1}{5}$	$-\frac{4}{5}$	0	$\frac{4}{5}$	$-\frac{1}{5}$	$\frac{4}{105}$	$-\frac{1}{280}$	
5		$-\frac{1}{1260}$	$\frac{5}{504}$	$-\frac{5}{84}$	$\frac{5}{21}$	$-\frac{5}{6}$	0	$\frac{5}{6}$	$-\frac{5}{21}$	$\frac{5}{84}$	$-\frac{5}{504}$	$\frac{1}{1260}$

For general n, k :

$$a_{n,0}^0 = 1 \text{ for all } n \quad b_{n,k}^0 = \begin{cases} 0 & k = 0 \\ \frac{(-1)^{k+1}}{k} \frac{(n!)^2}{(n+k)!(n-k)!} & k \neq 0 \end{cases}$$

Limit as $n \rightarrow \infty$:

$$a_{\infty,0}^0 = 1 \quad b_{\infty,k}^0 = \begin{cases} 0 & k = 0 \\ \frac{(-1)^{k+1}}{k} & k \neq 0 \end{cases}$$

Table 2.3: Weights for implicit 3-diagonal, regular grid FD formulae (order of accuracy = $2n + 2$).

		weights $a_{n,k}^1$					weights $b_{n,k}^1$								
		for f'					for f								
n	$k =$	-1	0	1	-5	-4	-3	-2	-1	0	1	2	3	4	5
1		$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$					$-\frac{1}{2}$	0	$\frac{1}{2}$				
2		$\frac{1}{5}$	$\frac{3}{5}$	$\frac{1}{5}$				$-\frac{1}{60}$	$-\frac{7}{15}$	0	$\frac{7}{15}$	$\frac{1}{60}$			
3		$\frac{3}{14}$	$\frac{4}{7}$	$\frac{3}{14}$			$\frac{1}{840}$	$-\frac{1}{35}$	$-\frac{25}{56}$	0	$\frac{25}{56}$	$\frac{1}{35}$	$-\frac{1}{840}$		
4		$\frac{2}{9}$	$\frac{5}{9}$	$\frac{2}{9}$		$-\frac{1}{7560}$	$\frac{1}{378}$	$-\frac{1}{27}$	$-\frac{13}{30}$	0	$\frac{13}{30}$	$\frac{1}{27}$	$-\frac{1}{378}$	$\frac{1}{7560}$	
5		$\frac{5}{22}$	$\frac{6}{11}$	$\frac{5}{22}$	$\frac{1}{55440}$	$-\frac{1}{2772}$	$\frac{5}{12332}$	$-\frac{10}{231}$	$-\frac{14}{33}$	0	$\frac{14}{33}$	$\frac{10}{231}$	$-\frac{5}{12332}$	$\frac{1}{2772}$	$-\frac{1}{55440}$

For general n, k :

$$a_{n,k}^1 = \begin{cases} \frac{n+1}{2n+1} & k = 0 \\ \frac{n}{2(2n+1)} & k = \pm 1 \end{cases} \quad b_{n,k}^1 = \begin{cases} 0 & k = 0 \\ (-1)^{k+1} \frac{(3n+1)(n+2)}{4(2n+1)(n+1)} & k = \pm 1 \\ \frac{(-1)^k}{(k+1)(k)(k-1)} \frac{(n+1)}{(2n+1)} \frac{(n!)^2}{(n-k)!(n+k)!} & |k| \geq 2 \end{cases}$$

Limit as $n \rightarrow \infty$:

$$a_{\infty,k}^1 = \begin{cases} \frac{1}{2} & k = 0 \\ \frac{1}{4} & k = \pm 1 \end{cases} \quad b_{\infty,k}^1 = \begin{cases} 0 & k = 0 \\ \frac{3}{8} \text{sign}(k) & k = \pm 1 \\ \frac{1}{2} \frac{(-1)^k}{(k-1)(k)(k+1)} & |k| \geq 2 \end{cases}$$

Table 2.4: Weights for implicit 5-diagonal, regular grid FD formulae (order of accuracy = $2n + 4$).

		weights $a_{n,k}^2$					weights $b_{n,k}^2$								
		for f'					for f								
n	$k =$	-2	-1	0	1	2	-4	-3	-2	-1	0	1	2	3	4
1		$-\frac{1}{180}$	$\frac{17}{90}$	$\frac{19}{30}$	$\frac{17}{90}$	$-\frac{1}{180}$				$-\frac{1}{2}$	0	$\frac{1}{2}$			
2		$\frac{1}{70}$	$\frac{8}{35}$	$\frac{18}{35}$	$\frac{8}{35}$	$\frac{1}{70}$			$-\frac{5}{84}$	$-\frac{8}{21}$	0	$\frac{8}{21}$	$\frac{5}{84}$		
3		$\frac{1}{42}$	$\frac{5}{21}$	$\frac{10}{21}$	$\frac{5}{21}$	$\frac{1}{42}$		$-\frac{1}{1260}$	$-\frac{101}{1260}$	$-\frac{85}{252}$	0	$\frac{85}{252}$	$\frac{101}{1260}$	$\frac{1}{1260}$	
4		$\frac{1}{33}$	$\frac{8}{33}$	$\frac{5}{11}$	$\frac{8}{33}$	$\frac{1}{33}$	$\frac{1}{27720}$	$-\frac{2}{1155}$	$-\frac{91}{990}$	$-\frac{14}{45}$	0	$\frac{14}{45}$	$\frac{91}{990}$	$\frac{2}{1155}$	$-\frac{1}{27720}$

For general n, k :

$$a_{n,k}^2 = \begin{cases} \frac{3}{2} \frac{(n+1)(n+2)}{(2n+1)(2n+3)} & k = 0 \\ \frac{n(n+2)}{(2n+1)(2n+3)} & k = \pm 1 \\ \frac{n(n-1)}{4(2n+1)(2n+3)} & k = \pm 2 \end{cases} \quad b_{n,k}^2 = \begin{cases} 0 & k = 0 \\ \frac{(-1)^{k+1}(n+2)(n+3)(5n+2)}{6(n+1)(2n+1)(2n+3)} & k = \pm 1 \\ \frac{(-1)^k(n+3)(25n^3+23n^2-22n-8)}{48(n+1)(n+2)(2n+1)(2n+3)} & k = \pm 2 \\ \frac{(-1)^k}{(k-2)(k-1)(k)(k+1)(k+2)} \frac{6(n+1)(n+2)}{(2n+1)(2n+3)} \frac{(n!)^2}{(n-k)!(n+k)!} & |k| > 2 \end{cases}$$

Limit as $n \rightarrow \infty$:

$$a_{\infty,k}^2 = \begin{cases} \frac{3}{8} & k = 0 \\ \frac{1}{4} & k = \pm 1 \\ \frac{1}{16} & k = \pm 2 \end{cases} \quad b_{\infty,k}^2 = \begin{cases} 0 & k = 0 \\ \frac{5}{24} \text{sign}(k) & k = \pm 1 \\ \frac{25}{192} \text{sign}(k) & k = \pm 2 \\ \frac{3}{2} \frac{(-1)^{k+1}}{(k-2)(k-1)(k)(k+1)(k+2)} & |k| > 2 \end{cases}$$
Table 2.5: Weights for explicit, staggered grid FD formulae (order of accuracy = $2n$).

		weights $b_{n,k}^0$ for f									
n	$k =$	$-\frac{9}{2}$	$-\frac{7}{2}$	$-\frac{5}{2}$	$-\frac{3}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{5}{2}$	$\frac{7}{2}$	$\frac{9}{2}$
1						-1	1				
2					$\frac{1}{24}$	$-\frac{9}{8}$	$\frac{9}{8}$	$-\frac{1}{24}$			
3				$-\frac{3}{640}$	$\frac{25}{384}$	$-\frac{75}{64}$	$\frac{75}{64}$	$-\frac{25}{384}$	$\frac{3}{640}$		
4			$\frac{5}{7168}$	$-\frac{49}{5120}$	$\frac{245}{3072}$	$-\frac{1225}{1024}$	$\frac{1225}{1024}$	$-\frac{245}{3072}$	$\frac{49}{5120}$	$-\frac{5}{7168}$	
5		$-\frac{35}{294912}$	$\frac{405}{229376}$	$-\frac{567}{40960}$	$\frac{735}{8192}$	$-\frac{19845}{16384}$	$\frac{19845}{16384}$	$-\frac{735}{8192}$	$\frac{567}{40960}$	$-\frac{405}{229376}$	$\frac{35}{294912}$

For general n, k :

$$a_{n,0}^0 = 1 \text{ for all } n \quad b_{n,k}^0 = \frac{(-1)^{k-1/2}}{2k^2} \frac{[(2n-1)!]^2}{(2n-1-2k)!(2n-1+2k)!}$$

Limit as $n \rightarrow \infty$:

$$a_{\infty,0}^0 = 1 \quad b_{\infty,k}^0 = \frac{(-1)^{k-1/2}}{\pi k^2}$$

Table 2.6: Weights for implicit 3-diagonal, staggered grid FD formulae (order of accuracy = $2n + 2$).

weights $a_{n,k}^1$ for f'				weights $b_{n,k}^1$ for f									
n	$k = -1$	0	1	$-\frac{9}{2}$	$-\frac{7}{2}$	$-\frac{5}{2}$	$-\frac{3}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{5}{2}$	$\frac{7}{2}$	$\frac{9}{2}$
1	$\frac{1}{24}$	$\frac{11}{12}$	$\frac{1}{24}$					-1	1				
2	$\frac{9}{80}$	$\frac{31}{40}$	$\frac{9}{80}$				$-\frac{17}{240}$	$-\frac{63}{80}$	$\frac{63}{80}$	$\frac{17}{240}$			
3	$\frac{25}{168}$	$\frac{59}{84}$	$\frac{25}{168}$			$\frac{61}{40320}$	$-\frac{925}{8064}$	$-\frac{2675}{4032}$	$\frac{2675}{4032}$	$\frac{925}{8064}$	$-\frac{61}{40320}$		
4	$\frac{49}{288}$	$\frac{95}{144}$	$\frac{49}{288}$	$-\frac{43}{430080}$	$\frac{343}{110592}$	$-\frac{78841}{552960}$	$-\frac{64925}{110592}$	$\frac{64925}{110592}$	$\frac{78841}{552960}$	$-\frac{343}{110592}$	$\frac{43}{430080}$		
5	$\frac{81}{440}$	$\frac{139}{220}$	$\frac{81}{440}$	$\frac{221}{22708224}$	$-\frac{15957}{63078400}$	$\frac{70821}{15769600}$	$-\frac{364119}{2252800}$	$-\frac{96579}{180224}$	$\frac{96579}{180224}$	$\frac{364119}{2252800}$	$-\frac{70821}{15769600}$	$\frac{15957}{63078400}$	$-\frac{221}{22708224}$

For general n, k :

$$a_{n,k}^1 = \begin{cases} \frac{4n^2 + 8n - 1}{4n(2n+1)} & k = 0 \\ \frac{(2n-1)^2}{8n(2n+1)} & k = \pm 1 \end{cases} \quad b_{n,k}^1 = \frac{(-1)^{k+1/2} [(2n-1)!!]^2}{8n(2n+1)(2n-1-2k)!(2n-1+2k)!} \frac{(3k^2-1)(4n^2-1)+8n(k^2-1)}{[(k-1)(k)(k+1)]^2}$$

Limit as $n \rightarrow \infty$:

$$a_{\infty,k}^1 = \begin{cases} \frac{1}{2} & k = 0 \\ \frac{1}{4} & k = \pm 1 \end{cases} \quad b_{\infty,k}^1 = \frac{(-1)^{k+1/2}}{2\pi} \frac{3k^2-1}{[(k-1)(k)(k+1)]^2}$$

Table 2.7: Weights for implicit 5-diagonal, staggered grid FD formulae (order of accuracy = $2n + 4$).

n	weights $a_{n,k}^2$ for f'			weights $b_{n,k}^2$ for f					
	$k = 0$	± 1	± 2	$-\frac{5}{2}$	$-\frac{3}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{5}{2}$
1	$\frac{863}{960}$	$\frac{77}{1440}$	$-\frac{17}{5760}$			-1	1		
2	$\frac{3667}{5440}$	$\frac{3057}{19040}$	$\frac{183}{76160}$		$-\frac{367}{2856}$	$-\frac{585}{952}$	$\frac{585}{952}$	$\frac{367}{2856}$	
3	$\frac{288529}{491904}$	$\frac{48425}{245952}$	$\frac{1075}{109312}$	$-\frac{69049}{14757120}$	$-\frac{505175}{2951424}$	$-\frac{683425}{1475712}$	$\frac{683425}{1475712}$	$\frac{505175}{2951424}$	$\frac{69049}{14757120}$
4	$\frac{1461701}{2724480}$	$\frac{879403}{4086720}$	$\frac{54145}{3269376}$	$-\frac{19618669}{1961625600}$	$-\frac{124703971}{653875200}$	$-\frac{2698675}{7133184}$	$\frac{2698675}{7133184}$	$\frac{124703971}{653875200}$	$\frac{19618669}{1961625600}$

For general n, k :

$$a_{n,k}^2 = \begin{cases} \frac{576n^6 + 1920n^5 + 1488n^4 - 4288n^3 - 3156n^2 + 952n - 81}{8(2n)(2n+1)(2n+2)(2n+3)(12n^2 - 16n + 1)} & k = 0 \\ \frac{(2n-1)^2(48n^4 + 128n^3 - 120n^2 - 160n + 27)}{4(2n)(2n+1)(2n+2)(2n+3)(12n^2 - 16n + 1)} & k = \pm 1 \\ \frac{(2n-1)^2(2n-3)^2(12n^2 + 8n - 3)}{16(2n)(2n+1)(2n+2)(2n+3)(12n^2 - 16n + 1)} & k = \pm 2 \end{cases}$$

$$b_{n,k}^2 = \frac{(-1)^{k-1/2} [(2n-1)!!]^2}{4(2n-1-2k)!!(2n-1+2k)!!} \frac{P(n,k)}{(2n)(2n+1)(2n+2)(2n+3)(12n^2 - 16n + 1)[(k-2)(k-1)(k)(k+1)(k+2)]^2}$$

$$\text{where } P(n, k) = (576n^6 - 81)(5k^4 - 15k^2 + 4) + 384n^5(15k^4 - 59k^2 + 20) - 48n^4(101k^4 - 175k^2 - 124) - 64n^3(169k^4 - 653k^2 + 268) - 12n^2(35k^4 - 745k^2 + 1052) + 8n(293k^4 - 1129k^2 + 476)$$

Limit as $n \rightarrow \infty$:

$$a_{\infty,k}^2 = \begin{cases} \frac{3}{8} & k = 0 \\ \frac{1}{4} & k = \pm 1 \\ \frac{1}{16} & k = \pm 2 \end{cases} \quad b_{\infty,k}^2 = \frac{(-1)^{k-1/2}}{2\pi} \frac{3(4-15k^2+5k^4)}{[(k-2)(k-1)(k)(k+1)(k+2)]^2}$$

Table 2.8: Limits of weights as $n \rightarrow \infty$ for schemes with $2m + 1$ diagonals, $m = 0, 1, 2, \dots$. (See Appendix B for a brief derivation of the integral forms.)

Limit	Integral form	Explicit form
[Regular grid (k integer)]		
$a_{\infty,k}^m$	$= \frac{1}{\pi} \int_0^{\pi} \cos(kx) \left[\cos\left(\frac{x}{2}\right) \right]^{2m} dx$	$= \begin{cases} \frac{1}{2^{2m}} \frac{(2m)!}{(m-k)!(m+k)!} & k \leq m \\ 0 & k > m \end{cases}$
$b_{\infty,k}^m$	$= \frac{1}{\pi} \int_0^{\pi} x \sin(kx) \left[\cos\left(\frac{x}{2}\right) \right]^{2m} dx$	$= \begin{cases} 0 & k = 0 \\ \frac{(\text{sign } k)(2m)! \sum_{j=1- k }^{ k } \frac{1}{j+m}}{2^{2m} (m-k)!(m+k)!} & 0 < k \leq m \\ \frac{(-1)^{k+m+1} (2m)!}{2^{2m} \prod_{j=-m}^m (k+j)} & k > m \end{cases}$
[Staggered grid (k half – integer)]		
$a_{\infty,k}^m$	$= \frac{1}{\pi} \int_0^{\pi} \cos(kx) \left[\cos\left(\frac{x}{2}\right) \right]^{2m} dx$	$= \begin{cases} \frac{1}{2^{2m}} \frac{(2m)!}{\Gamma(m-k+1)\Gamma(m+k+1)} & k \leq m \\ 0 & k > m \end{cases}$
$b_{\infty,k}^m$	$= \frac{1}{\pi} \int_0^{\pi} x \sin(kx) \left[\cos\left(\frac{x}{2}\right) \right]^{2m} dx$	$= \frac{(\text{sign } k)(2m)! \sum_{j=1- k }^{ k } \frac{1}{j+m}}{2^{2m} \Gamma(m-k+1)\Gamma(m+k+1)}$

we are not aware of any references which analyze such combined schemes.

2.5 Equivalence between implicit and explicit formulae in the case of limiting order

An explicit finite difference (FD) stencil directly expresses how the approximation of a derivative is influenced by changes in function values at different locations. For regular grids, as the order of accuracy increases (i.e. $n \rightarrow \infty$), the weights approach

$$b_{\infty,k}^0 = \frac{(-1)^k}{k+1} \quad (k \neq 0) \quad (2.3)$$

(cf. Table 2.2, also [10], pp. 20-22). The derivative approximation therefore depends significantly on function values quite far away (in contrast to the exact derivative being a strictly local property of a function).

For the tridiagonal implicit stencil, the decay of the weights is much faster:

$$b_{\infty,k}^1 = \frac{1}{2} \frac{(-1)^k}{(k-1)k(k+1)} \quad (|k| \geq 2) \quad (2.4)$$

(cf. Table 2.3). Superficially, it might appear that these approximations remain more ‘local.’ However, to actually obtain derivative approximations, we need to solve a tridiagonal system. The equivalent explicit scheme (obtained through multiplication by the inverse of this infinite tridiagonal matrix) is as globally coupled as the original explicit scheme. The same will also hold true if we have 5 or more diagonals; in fact, in the limit of increasing order, these schemes all become identical, as demonstrated later in this section through numerical simulations.

In the staggered case, all implicit schemes have similarly the equivalent

ficients

$$a_{n,k}^1 = \begin{cases} \frac{n+1}{2n+1}, & k = 0 \\ \frac{n}{2(2n+1)}, & k = \pm 1. \end{cases} \quad (2.8)$$

After substituting this into (2.7), we simplify to find that, for the nonstaggered $m = 1$ case,

$$d_k = \sqrt{2n+1} \left\{ -1 + \frac{1}{n} (\sqrt{2n+1} - 1) \right\}^{|k|}. \quad (2.9)$$

In the staggered grid tridiagonal case ($m = 1$), the Toeplitz matrix has coefficients

$$a_{n,k}^1 = \begin{cases} \frac{4n^2+8n-1}{4n(2n+1)}, & k = 0 \\ \frac{(2n-1)^2}{8n(2n+1)}, & k = \pm 1. \end{cases} \quad (2.10)$$

After substituting these into (2.7), we simplify to find that, in the staggered $m = 1$ case,

$$d_k = \sqrt{\frac{2n(2n+1)}{6n-1}} \left\{ \frac{-(n^2 + 2n - \frac{1}{4}) + \sqrt{n(n + \frac{1}{2})(6n-1)}}{(n - \frac{1}{2})^2} \right\}^{|k|}. \quad (2.11)$$

We note that in both (2.9) and (2.11), the quantities inside the braces are bounded between -1 and 1 . Thus, the inverse matrix is well-defined because elements decay away from the main diagonal. We note that for $m > 1$, it is much more difficult to evaluate the integral in (2.7), so we do not give explicit expressions for d_k for $m > 1$.

The author has numerically computed equivalent weights using (2.6), the inverse matrix given by (2.7), and the weights $a_{n,k}^m$ and $b_{n,k}^m$ given in Tables 2.3, 2.4, 2.6, and 2.7. Figure 2.5 illustrates graphically how the equivalent weights compare for explicit, 3-diagonal, and 5-diagonal schemes, both nonstaggered and staggered. These

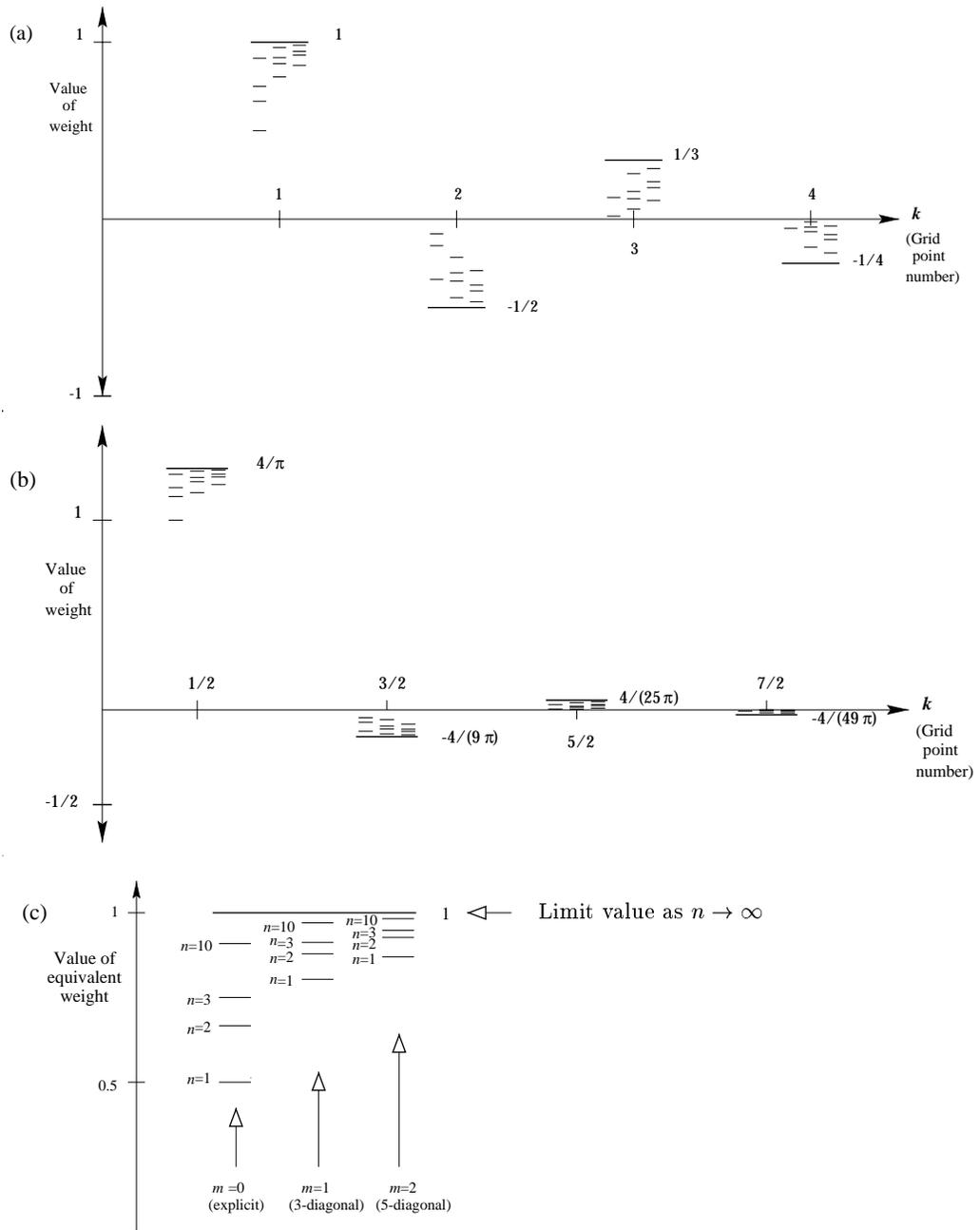


Figure 2.5: (a) Equivalent weights for the nonstaggered grid case. (b) Equivalent weights for the staggered grid case. (c) Legend (for the $k = 1$ case of Figure 2.5(a)). Note that for explicit methods, weights are 0 for $n < k$.

figures demonstrate that all nonstaggered implicit schemes have the same limiting equivalent weights, as $n \rightarrow \infty$, as the explicit nonstaggered scheme, and that all staggered implicit schemes have the same limiting equivalent weights, as $n \rightarrow \infty$, as the explicit staggered scheme. Thus, there is little advantage in using implicit schemes from the standpoint of localization; grid staggering is seen to have more effect in this respect. We note that these limits (as $n \rightarrow \infty$) for both regular and staggered grids, if implemented on periodic data, become identical to the respective periodic pseudospectral methods [9, 10].

2.6 Operation counts

Regarding the number of arithmetic operations required to obtain the derivative at a grid point, we note:

- There is no need to make any distinction between regular and staggered grids; their operation counts are identical (when expressed in n and k).
- For the implicit cases, the LU factorization of the finite difference coefficient matrix can be stored. The entries in these matrices do not depend on the system size; i.e., one copy suffices even if the domain geometry is such that we have to solve systems of different sizes.

One example of operation count suffices to illustrate the general counting process. Consider the 3-diagonal regular grid case with $n = 1$, which has weights

$$\begin{bmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix} f' = \begin{bmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} f. \quad (2.12)$$

Table 2.9: Operation count to calculate f' at one grid point when using different schemes. (Note: there is no difference between regular and staggered grids.)

	$n = 1$	2	3	4	\dots	General n
Explicit	2	5	8	11	\dots	$3n - 1$
3 - diagonal	6	9	12	15	\dots	$3n + 3$
5 - diagonal	10	13	16	19	\dots	$3n + 7$
\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
k - diagonal	$2k$	$2k + 3$	$2k + 6$	$2k + 9$	\dots	$2k + 3n - 3$

Table 2.10: Coefficients of leading error terms for different first derivative approximations (Expressions for general p found by the author). n is the row number in Tables 2.2 - 2.7.

Order of accuracy	$p = 2$	4	6	8	p	Order p in terms of n
Error form	$h^2 f^{(3)}(x)$	$h^4 f^{(5)}(x)$	$h^6 f^{(7)}(x)$	$h^8 f^{(9)}(x)$	$h^p f^{(p+1)}(x)$	
<i>Regular</i>						
Explicit	$\frac{1}{6}$	$-\frac{1}{30}$	$\frac{1}{140}$	$-\frac{1}{630}$	$\frac{(-1)^{p/2+1} [(\frac{p}{2})!]^2}{(p+1)!}$	$2n$
3 - diag.		$-\frac{1}{280}$	$\frac{1}{2100}$	$-\frac{1}{17640}$	$\frac{(-1)^{p/2-1} [(\frac{p}{2}-1)!]^2 (\frac{p}{2})}{(p+1)!(p-1)!}$	$2n + 2$
5 - diag.			$\frac{1}{1512}$	$-\frac{1}{44100}$	$\frac{(-1)^{p/2-1} 6 [(\frac{p}{2}-2)!]^2 (\frac{p}{2}-1) (\frac{p}{2})}{(p+1)!(p-3)!(p-1)!}$	$2n + 4$
<i>Staggered</i>						
Explicit	$\frac{1}{24}$	$-\frac{3}{640}$	$\frac{5}{7168}$	$-\frac{35}{294912}$	$\frac{(-1)^{p/2+1} (p)!}{2^{2p} [(\frac{p}{2})!]^2 (p+1)}$	$2n$
3 - diag.		$-\frac{17}{5760}$	$\frac{61}{358400}$	$-\frac{215}{14450688}$	$\frac{(-1)^{p/2-1} (p-3)(3p^2-9p+1)}{2^{2p-2} (\frac{p}{2}-1)! (\frac{p}{2})! (p-1)^2 (p+1)}$	$2n + 2$
5 - diag.			$\frac{367}{967680}$	$-\frac{69049}{6141542400}$	$\frac{(-1)^{p/2-1} (p-5)! R(p)}{2^{2p-4} (\frac{p}{2})! (\frac{p}{2}+1)! (p-3)^2 (p-1)^2 (p+1)}$	$2n + 4$
					where $R(p) = \frac{(45p^6 - 900p^5 + 6897p^4 - 25304p^3 + 44631p^2 - 31396p + 2187)}{(3p^2 - 32p + 81)}$	

A frequently used alternative error comparison approach (e.g. [8, 23]) for wave equation analysis consists of inspecting how the different derivative approximations treat a pure Fourier mode $e^{i\omega x}$ on a grid over $[-1, 1]$ (for example), with grid spacing h . The modes that can be represented on the grid will satisfy $-\pi < \omega h \leq \pi$; higher modes will appear equivalent to a lower one on the grid due to aliasing. The exact derivative of $e^{i\omega x}$ is

$$\frac{d}{dx} (e^{i\omega x}) = i\omega e^{i\omega x} = \omega (ie^{i\omega x}) \quad (2.13)$$

We wish to compare how well various FD approximations do in approximating this exact factor ω .

Applying to $e^{i\omega x}$ the explicit, regular grid, second order FD approximation for the first derivative gives

$$\frac{1}{h} \left[\frac{1}{2} e^{i\omega(x+h)} - \frac{1}{2} e^{i\omega(x-h)} \right] = \frac{\sin(\omega h)}{h} (ie^{i\omega x}) \quad (2.14)$$

These factors (ω and $\frac{\sin \omega h}{h}$) are seen as the diagonal straight line and the bottom curve respectively in the top left subplot of Figure 2.6. The $\frac{\sin(\omega h)}{h}$ - curve is seen to be approximately correct only for a small fraction of the Fourier modes the grid can represent; using it is extremely wasteful in terms of computational efficiency. As the order of accuracy p increases, the coverage over $\omega = [0, \pi]$ clearly improves. The other five subplots in Figure 2.6 show how coverage is gained both by adding diagonals (using increasingly more implicit approximations) and by using staggering. The fact that the curves for staggered approximations are not forced to be zero at $\omega = \pi$ (but instead have zero slope there) allows them to provide better coverage across the spectrum.

A major advantage of this spectral comparison method (as opposed to looking at error coefficients) is that we can directly compare methods of different orders of accuracy. To better see the differences between the methods, we show in Figure 2.7 how the different curves in Figure 2.6 deviate from the exact result. In this

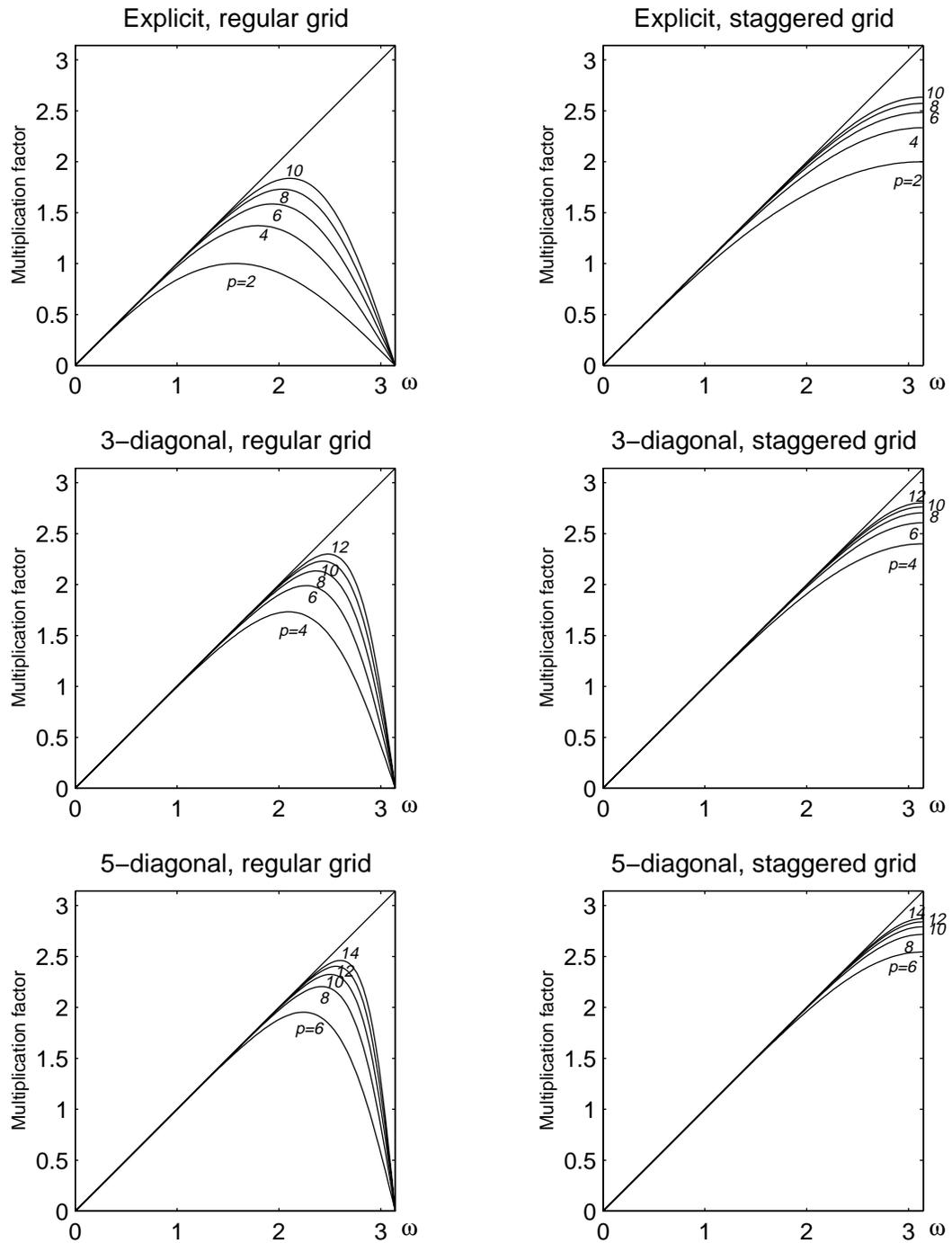


Figure 2.6: Fourier multiplication factors for different methods, displayed against ω . The curves are labeled according to the order of accuracy p of the methods.

figure, the curves are not labeled according to their accuracy, but according to the computational cost per grid point, as displayed in Table 2.9. We note that for either type of grid, there is a significant improvement in going from explicit to 3-diagonal schemes, but to proceed further to 5-diagonal does not improve efficiency much, if at all. Staggering is again seen to clearly be advantageous in all cases. The three schemes that are highlighted as particularly effective in Figure 2.7 are all staggered, and have the stencils

$$\begin{array}{ccc}
 \begin{array}{c} \circ \\ \blacksquare \blacksquare \\ \text{Order 2} \end{array} &
 \begin{array}{c} \circ \ \circ \ \circ \\ \blacksquare \blacksquare \blacksquare \blacksquare \\ \text{Order 6} \end{array} &
 \begin{array}{c} \circ \ \circ \ \circ \ \circ \ \circ \\ \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare \\ \text{Order 10} \end{array}
 \end{array}$$

where “ \blacksquare ” denotes a known function value, and “ \circ ” represents an unknown derivative value entry.

2.8 Test problem

The work in this section was done by Bengt Fornberg. As a simple test problem, we consider

$$u_t + u_x = 0, \tag{2.15}$$

periodic over $[-1, 1]$, with initial condition

$$u(x, 0) = \begin{cases} [1 + \cos(\frac{\pi x}{0.15})]^2, & |x| \leq 0.15 \\ 0, & |x| > 0.15 \end{cases} \tag{2.16}$$

This equation is discretized in space and its solution advanced analytically in time (thus the plots in Figures 2.8–2.11 show only spatial errors). Figure 2.8 shows the numerical and exact solutions for a regular grid using different methods at time $t = 100$, after the pulse has traversed the domain 50 times. The number shown in the bottom left of each subplot tells the number of spatial grid points used;

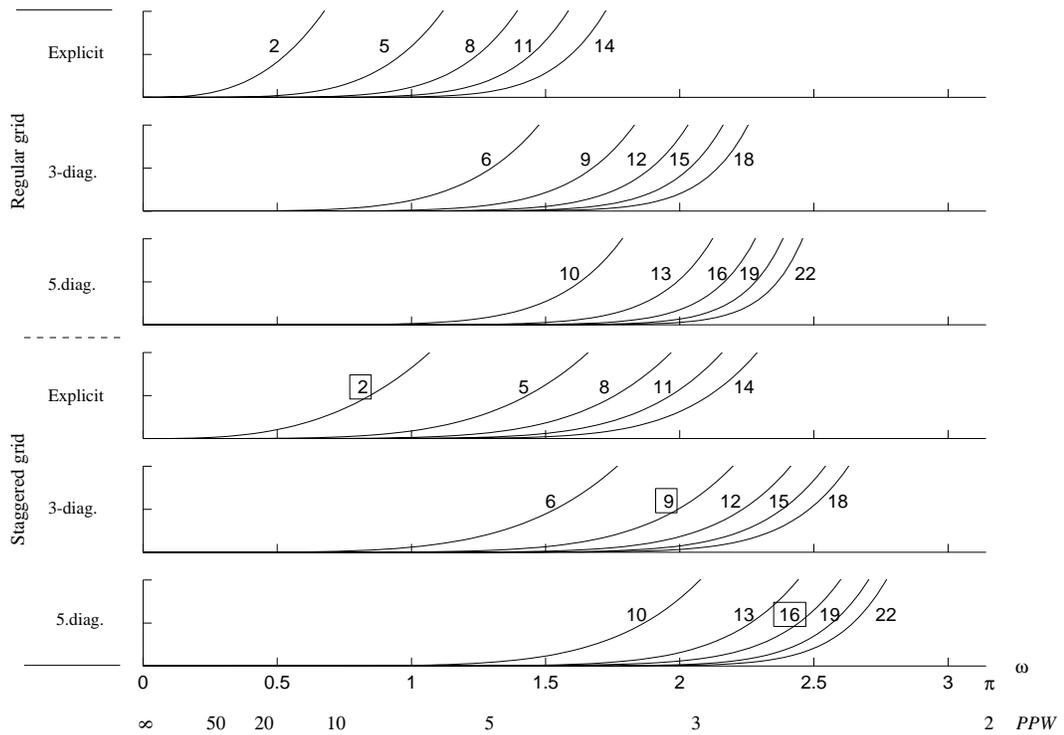


Figure 2.7: Deviation of the Fourier multiplication factors from the ideal straight line for cases shown in Figure 2.6. The horizontal axis is labeled in both frequency ω and in points per wavelength (PPW). The vertical axis in each subplot extends to 0.05. All curves are labeled according to the operation count per grid point, as given in Table 2.9. The boxed numbers mark schemes that are particularly advantageous in terms of operation count in their respective accuracy ranges.

this number was selected so that, based on the operation counts in Table 2.9, all cases would be equally costly if run in 1-D. Figure 2.9 shows the equivalent data for staggered grids. As expected from our analysis, staggering is advantageous in all the cases, but less so for the most implicit scheme.

Higher dimensions and longer time integration more strongly favor higher order methods over lower order ones. Figure 2.10 shows the same test run to time $t = 2000$ using grid sizes (in each spatial direction) which would provide equal cost in 2-D.

In Figure 2.11, we refine the explicit $n = 1$ scheme successively. The additional number within each subplot shows the relative computer time required (with "1" corresponding to the cost of each of the cases in Figure 2.10). It is clear that to achieve acceptable (say, about 1%) accuracy requires exorbitant computational costs in both time and memory. The bottom left subplot in Figure 2.11 shows comparable accuracy to the bottom right one in Figure 2.9 — at about 4,000 times larger cost (in 3-D, this factor increases to about 260,000). The staggered second order scheme in this comparison corresponds to the spatial discretization of the Yee-scheme, which was pioneering work when first proposed for time-dependent computational electromagnetics (Maxwell's equations) in 1966 [29]. It has since enjoyed a long-lasting popularity (e.g. [22, 27]) in spite of its low order of accuracy.

2.9 Conclusions

Combining

- high orders of accuracy (i.e. wider stencils),
- implicitness, and
- staggering

leads to a class of computationally very cost-effective finite difference schemes. As their orders of accuracy increase, these schemes approach in accuracy the well-known

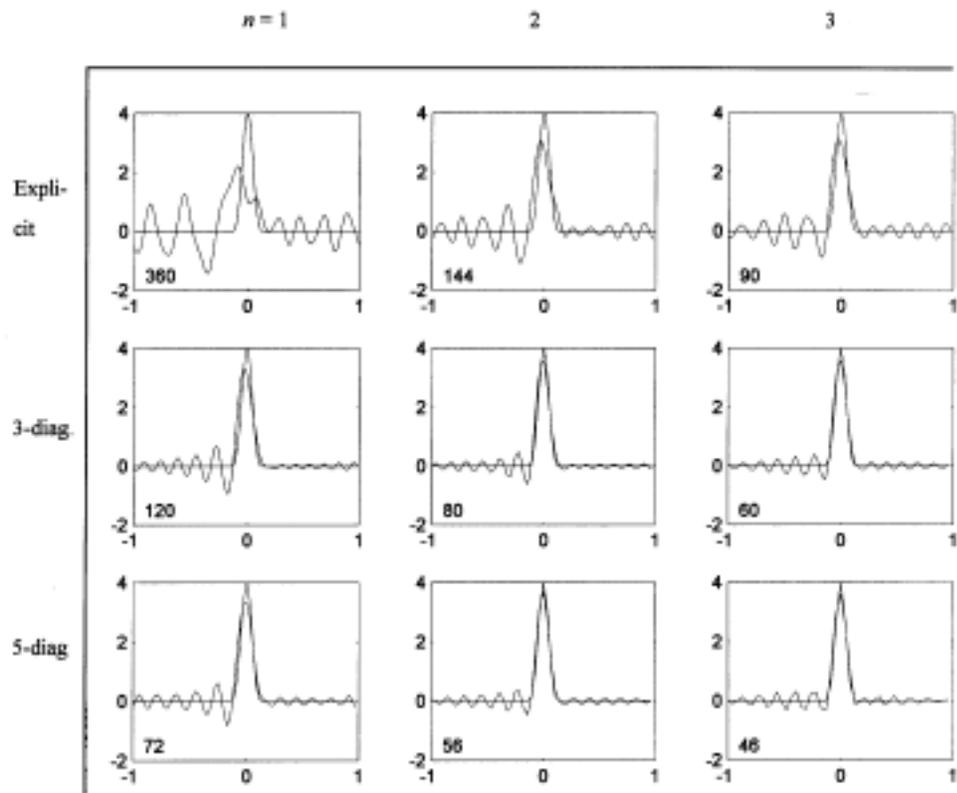


Figure 2.8: Regular grid solutions at $t = 100$ using different spatial approximations. The grid sizes (shown in the bottom left corner of each subplot) were selected to make each case equally costly in computer time (assuming 1-D).

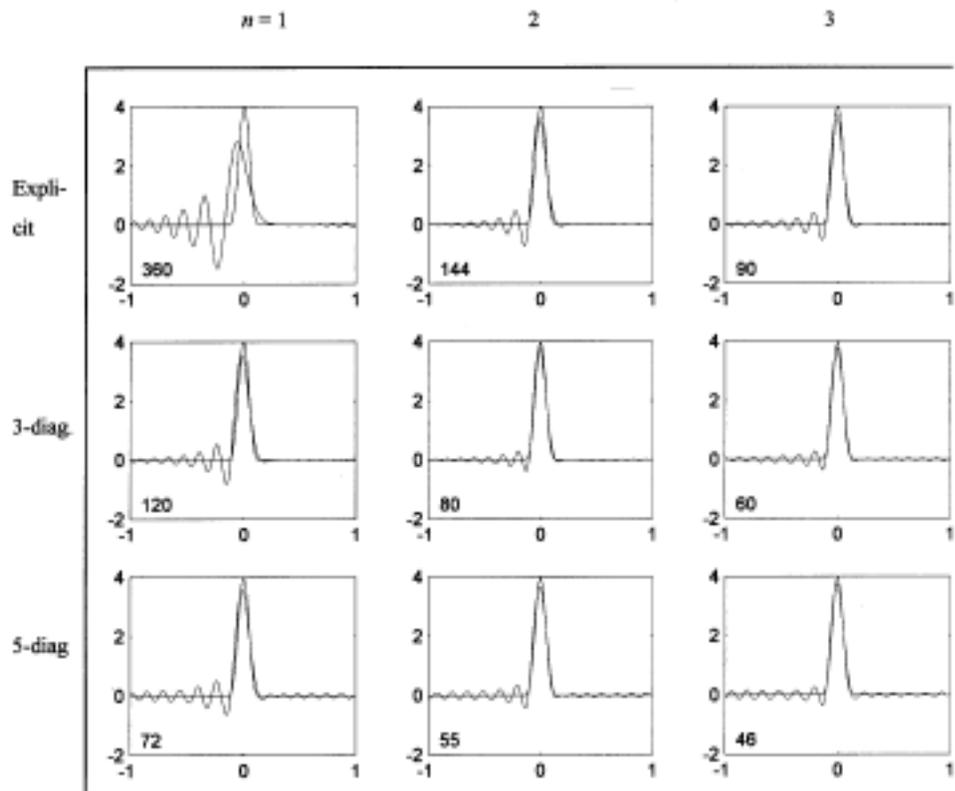


Figure 2.9: Same as Figure 2.8, but using staggered approximations for the spatial derivative. The reductions in the amplitudes of the dispersive wave trains (compared to the regular grid cases in Figure 2.8) are particularly noticeable if one compares the waves trains as they leave the left edge of the domains and (because of periodicity) reappear in the right half of the subplots.

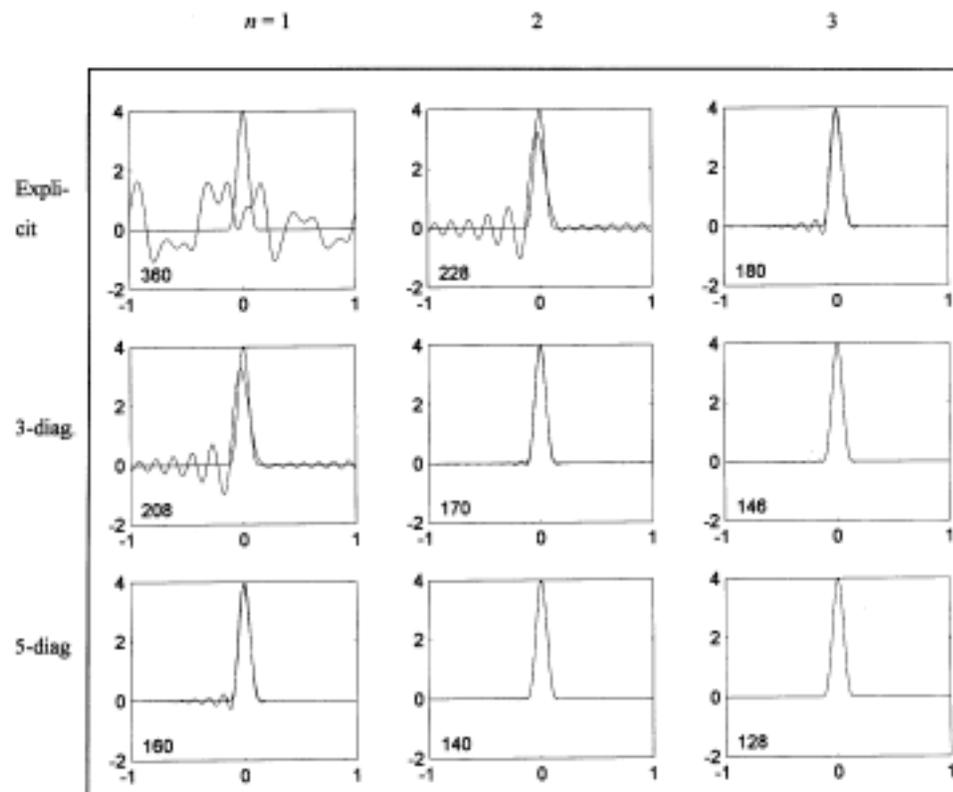


Figure 2.10: Solutions for staggered grid at $t = 2000$ with grid sizes selected to make computations equally time-consuming in 2-D. (The memory requirements scale with the square of the numbers given in the bottom left corners.)

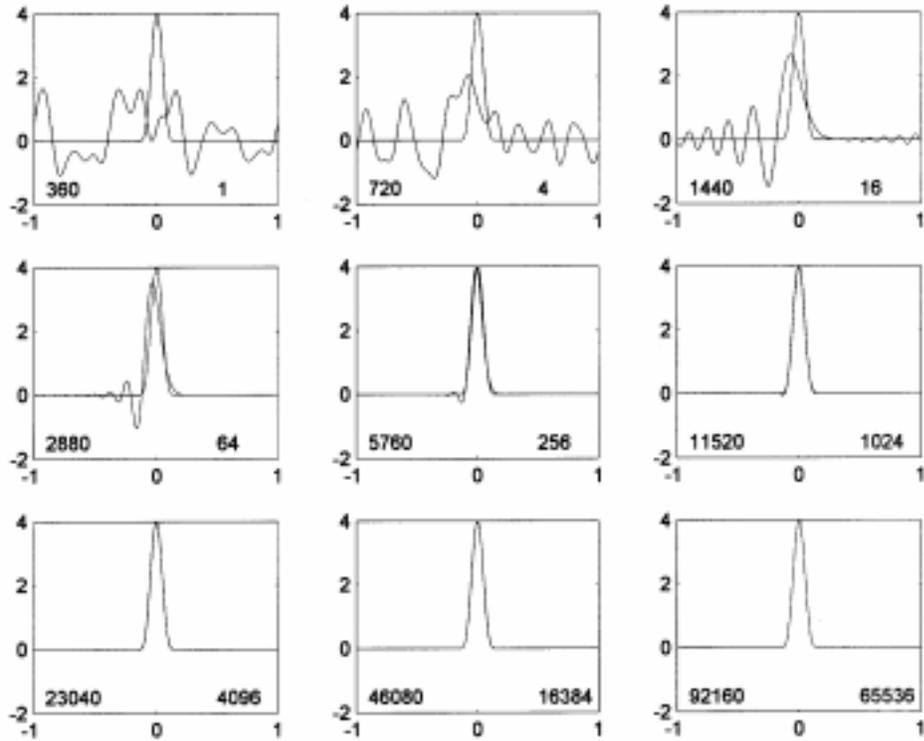


Figure 2.11: Solutions at $t = 2000$ for the staggered explicit $n = 1$ scheme for increasingly fine grids. The numbers in the bottom left corners denote (as in Figures 2.8-2.10) the number of grid points across the period. The numbers in the bottom right corners give the relative cost in both computer time and memory, if implemented in 2-D, compared to the unit (1) cost of all the cases in Figure 2.10. The top left subplot of Figures 2.10 and 2.11 are identical; we see here how costly it is to achieve high accuracy by refining the grid using this scheme.

spectral accuracy of periodic, explicit pseudospectral schemes. The schemes discussed in this chapter are defined on equispaced Cartesian grids. When combined with the idea of overlapping subdomains, the relatively narrow stencil widths make the schemes well-suited for computations in media with curvilinear material interfaces. The schemes can be applied to most linear wave-type PDEs of broad interest. In the particular application of time-domain computational electromagnetics (also known as FDTD), the classical Yee scheme uses only the last of the three highlighted concepts. We find that major improvements in accuracy and efficiency can be achieved by also incorporating the other two ideas of implicitness and staggering.

Chapter 3

Staggered Time Integrators for Wave Equations

In this chapter, we consider variations of the Adams–Bashforth, backwards differentiation, and Runge–Kutta families of time integrators to solve systems of linear wave equations on uniform, time-staggered grids. These methods are found to have smaller local truncation errors and to allow larger stable time steps than traditional nonstaggered versions of equivalent orders. We investigate the accuracy and stability of these methods analytically, experimentally, and through the use of a novel root portrait technique. In addition, we address several theoretical questions regarding staggered time integrators.

3.1 Introduction

When wave equations are posed as first order systems and discretized in space to yield a system of ordinary differential equations (ODEs), the linearization of the resulting system has a purely imaginary spectrum. This corresponds to the fact that only propagation takes place. Many classical methods for ODEs have stability regions that include an interval of the form $[-iS_I, iS_I]$ on the imaginary axis. We call the largest such value of S_I the **imaginary stability boundary (ISB)** of the ODE integrator. In the context of a semidiscrete wave equation, two features are desired for an ODE integrator:

- (1) small local truncation error and

- (2) large imaginary stability boundary (ISB).

These two properties are typically in opposition to one another.

In Chapter 2 and [9] it has been shown that the use of staggered or interlaced grids in space can increase the accuracy of finite difference and pseudospectral differentiation methods when used for linear wave equations. Similarly, the unknown variables of linear wave equations (and systems of such equations) can be staggered in time to yield benefits in both accuracy and stability. In this chapter we introduce novel families of multistep and multistage staggered ODE integrators. We find that for multistep methods of the same order of accuracy, staggering in time usually improves accuracy by a factor of about 9 and increases the ISB by a factor of 2.4–7.4, with the factor growing as order increases. We also present a fourth order multistage method which, compared to classical fourth order Runge–Kutta, has an error constant smaller by a factor of 16 and an ISB larger by a factor of about two.

Typically the computational cost of using an implicit method is justified only in the presence of stiffness (not an issue for linear wave equations) or when there is a relatively small number of equations in the system. We envision our methods being used to solve systems with a very large number of equations, possibly in the millions (as is the case when 2-D or 3-D wave equations are solved with a method of lines approach). For such situations, it is impractical to generate (and store) an LU decomposition. We thus consider only explicit methods in this analysis.

Although we focus our discussion on linear wave equations, linearity is not a requirement in any of our proposed schemes. Additionally, our time integrators are designed to solve first order systems. Although systems of wave equations can often be rewritten as second order systems, first order formulations are generally preferred in the literature (e.g. Maxwell’s equations), partly due to easier implementation of boundary conditions. (It is known that staggered grids are better for approximating odd-order derivatives and nonstaggered grids are better for approximating even-order derivatives [9].)

The rest of this chapter is organized as follows:

2. Illustrations of grid staggering in time for linear wave equations
3. Preliminaries
4. Staggered Adams-Bashforth and backwards differentiation methods
5. Staggered free parameter methods
6. Theoretical considerations
7. Staggered predictor-corrector methods
8. Staggered Runge–Kutta methods
9. Root portraits
10. Numerical experiments
11. Conclusions

The author made significant contributions to sections 4, 5, 6, 7, 8, and 10. Some contributions were made to section 2.

Because we use a number of acronyms which may be unfamiliar to the reader, a glossary of these abbreviations is included in Section 1.2.

3.2 Illustrations of grid staggering for linear wave equations

(Note: this section is an extension of Section 2.2. The time staggering grids in this section were created partly by the author.)

Staggered grid techniques apply to linear hyperbolic equations which have been written as first order systems. The variables in the system are staggered in such a way that the locations of values and their derivatives are interlaced.

We give three examples; other linear wave equations can be treated similarly. Figure 3.1 gives four different ways to lay out the grid of unknowns u and v

for the one-dimensional acoustic wave equation

$$\begin{aligned}\frac{\partial u}{\partial t} &= c \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial t} &= c \frac{\partial u}{\partial x}.\end{aligned}\tag{3.1}$$

One can choose to utilize time staggering, space staggering, both, or neither. In every case the space-time density of data is exactly the same. Note that if one wants to incorporate staggering in time, the variables u and v must exist on interlaced time intervals (e.g., u exists on integer time levels, while v exists on half-integer time levels).

Figure 3.2 shows nonstaggered and staggered space grids for the two-dimensional elastic wave equation

$$\begin{aligned}\rho \frac{\partial u}{\partial t} &= \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} \\ \rho \frac{\partial v}{\partial t} &= \frac{\partial g}{\partial x} + \frac{\partial h}{\partial y} \\ \frac{\partial f}{\partial t} &= (\lambda + 2\mu) \frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y} \\ \frac{\partial g}{\partial t} &= \mu \frac{\partial v}{\partial x} + \mu \frac{\partial u}{\partial y} \\ \frac{\partial h}{\partial t} &= \lambda \frac{\partial u}{\partial x} + (\lambda + 2\mu) \frac{\partial v}{\partial y}.\end{aligned}\tag{3.2}$$

The spatial staggering layout given in Figure 3.2 is uniquely determined. For example, the first equation requires that u be represented halfway between values of f in the x -direction and halfway between values of g in the y -direction. If $\frac{\partial g}{\partial x}$ also appeared in the first equation, it would not be possible to stagger spatially. We observe that for a large number of linear wave equations (e.g. Maxwell's equations in any number of dimensions), the structure of the governing equations turns out to be precisely such that a unique and internally conflict-free staggering arrangement is possible, but we are unaware of any discussion of this in the literature. If one also wants to incorporate time staggering for this equation (with or without spatial

staggering), we must again split the variables into two groups that exist on interlaced time intervals (e.g. u and v on integer time levels and f, g , and h on half-integer time levels). An illustration of this arrangement is given in Figure 3.3.

As a final example, to stagger the 3-D Maxwell's equations given in Figure 2.2(c) in time, one would, for example, represent the electric field components E_x , E_y , and E_z on integer time-levels and the magnetic field components H_x , H_y , and H_z on interlacing half-integer time-levels. This is possible whether or not one staggers in space.

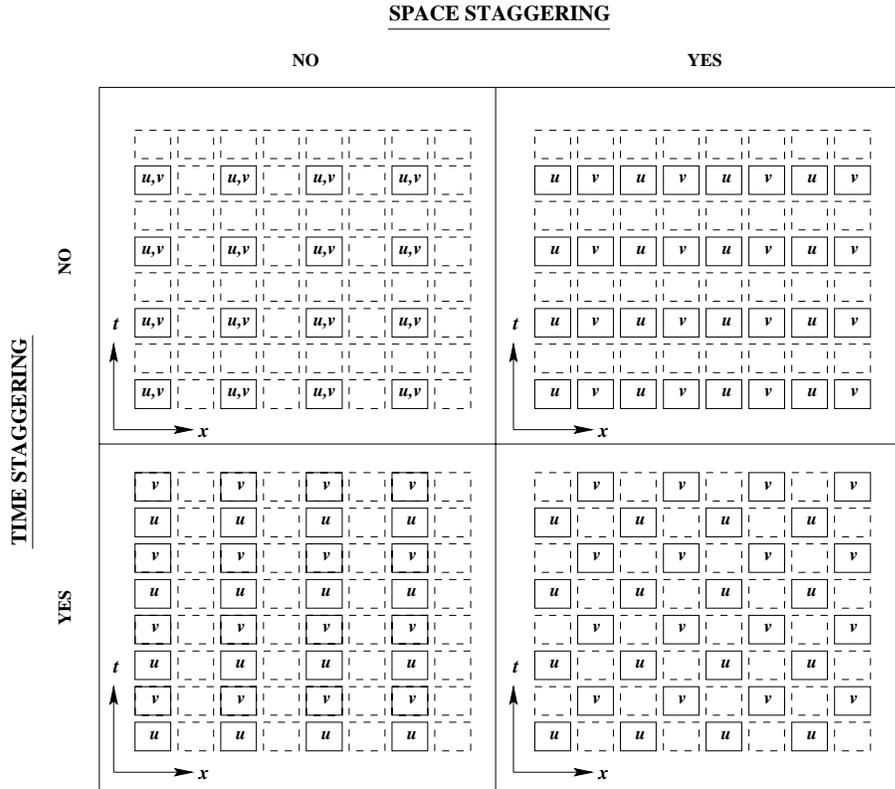


Figure 3.1: Representative samples of various spatial/time grid layouts for the one-dimensional wave equation (3.1)

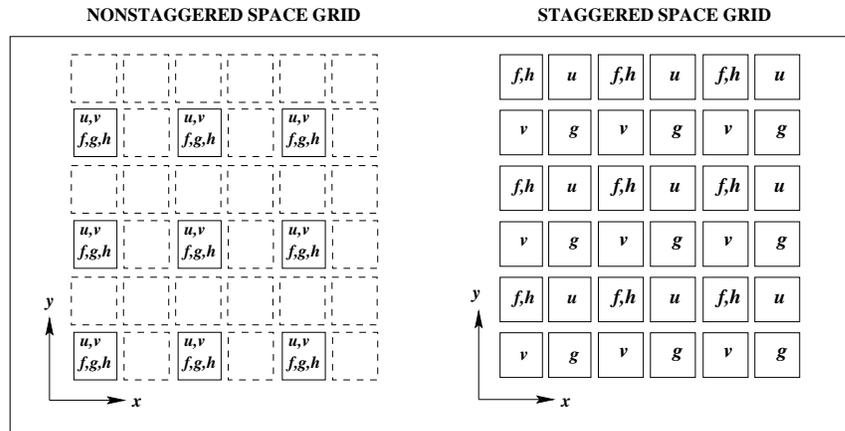


Figure 3.2: Representative sample spatial grid layouts for the two-dimensional elastic wave equation (3.2)

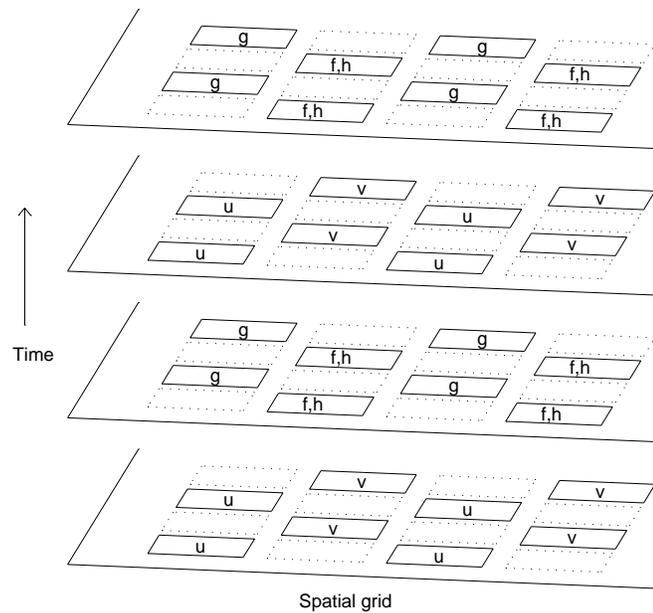


Figure 3.3: Representative sample of a spatial-staggered, time-staggered grid for the two-dimensional elastic wave equation (3.2)

3.3 Preliminaries

3.3.1 Definitions

An m -step linear multistep method for solving the ODE

$$\frac{dy}{dt} = f(t, y(t)) \quad (3.3)$$

is a difference equation of the form

$$\alpha_m y_{n+m} + \alpha_{m-1} y_{n+m-1} + \dots + \alpha_0 y_n = k(\beta_m f_{n+m} + \dots + \beta_0 f_n) \quad (3.4)$$

where k is the step size, α_i and β_i are real parameters, $t_i = t_0 + ik$, $y_i = y(t_i)$, and $f_i = f(t_i, y_i)$. The coefficients α_i and β_i can be generated by using a two-line Mathematica or Maple algorithm based on Padé expansions (see [11] or Section 2.4). Another way of representing the above general multistep method is through the use of generating polynomials

$$\begin{aligned} \rho(z) &= \alpha_m z^m + \alpha_{m-1} z^{m-1} + \dots + \alpha_0 \\ \sigma(z) &= \beta_m z^m + \beta_{m-1} z^{m-1} + \dots + \beta_0. \end{aligned} \quad (3.5)$$

We consider only explicit methods, in which case $\beta_m = 0$. The local truncation error of a multistep method of order p is usually defined as

$$L(y, t, k) = \rho(Z)y(t) - k\sigma(Z)y'(t) = C_{p+1}k^{p+1}y^{(p+1)}(t) + O(k^{p+2}), \quad (3.6)$$

where Z is the forward shift operator; this results from a simple Taylor expansion.

The constant C_{p+1} is given by

$$C_{p+1} = \frac{1}{(p+1)!} \left(\sum_{i=0}^m \alpha_i i^{p+1} - (p+1) \sum_{i=0}^m \beta_i i^p \right). \quad (3.7)$$

However, as discussed in Appendix F, this constant does not accurately reflect the global error to be expected when using a method. The proper **error constant** is given by

$$C = \frac{C_{p+1}}{\sigma(1)}. \quad (3.8)$$

It is this coefficient C that we use when comparing the accuracy of methods of the same order.

Similarly, an explicit s -stage Runge–Kutta method can be represented as

$$y_{n+1} = y_n + \sum_{j=1}^s b_j d_j \quad (3.9)$$

where

$$\begin{aligned} d_1 &= kf(t_n, y_n) \\ d_2 &= kf(t_n + c_2 k, y_n + a_{21} d_1) \\ d_3 &= kf(t_n + c_3 k, y_n + a_{31} d_1 + a_{32} d_2) \\ &\vdots \\ d_s &= kf\left(t_n + c_s k, y_n + \sum_{i=1}^{s-1} a_{si} d_i\right) \end{aligned} \quad (3.10)$$

The (linear) error constant for such a method can be found by considering the linear problem

$$y' = f(t, y) = \lambda y \quad (3.11)$$

and Taylor expanding $(y_{n+1} - e^{\lambda k} y_n)$ about $k = 0$ to find C :

$$k \left(\sum_{j=1}^s b_j d_j \right) + (1 - e^{\lambda k}) y_n = C(\lambda k)^{p+1} + O((\lambda k)^{p+2}). \quad (3.12)$$

For multistage methods, it is appropriate to normalize the stability domain by dividing by the number of stages s , and to normalize the error constant by a factor s^p , where p is the order of the method. This ensures that we are comparing all time-stepping methods on the basis of equal work.

Stability domains tell which values of $k\lambda$ produce stable solutions in solving the linear problem $y' = \lambda y$ for a given time integrator. For linear multistep methods, the boundary of the stability domain $\xi(\theta)$ is found by solving for ξ in

$$\rho\left(e^{i\theta}\right) - \xi\sigma\left(e^{i\theta}\right) = 0. \quad (3.13)$$

3.3.2 Maximum Imaginary Stability Boundary

Jeltsch and Nevanlinna [18] have shown that the normalized ISB for a large class of schemes, including multistep and RK methods, cannot exceed 1 in the classical (nonstaggered) case. This limit is achieved by the classical leapfrog scheme

$$y_{n+1} = y_{n-1} + 2kf(t_n, y_n). \quad (3.14)$$

This method has a stability domain $[-i, i]$ on the imaginary axis.

Leapfrog can also be used as a time-staggered method, namely

$$y_{n+1} = y_n + kf\left(t_n + \frac{k}{2}, y_{n+\frac{1}{2}}\right). \quad (3.15)$$

In this context the stability domain is $[-2i, 2i]$; the extra factor of two simply reflects the fact that the time levels are $\{n, n + \frac{1}{2}, n + 1\}$ rather than $\{n - 1, n, n + 1\}$. For staggered multistep and RK methods that we will consider, this implies a maximum normalized ISB of 2.

3.4 Staggered Adams–Bashforth and backwards differentiation methods

(This section is the author’s work, with the comparisons to Störmer methods done by Toby Driscoll.)

To utilize the methods in Sections 3.4 - 3.6, we require only that u and $\frac{\partial u}{\partial t}$ are used on interlaced time levels. However, as noted in Section 3.2, many (if not all) systems of linear wave equations can be rewritten in the form $u_t = f(t, v(t)), v_t = g(t, u(t))$ (where u and v may be vectors). In this case, by having u on one time level and v on the other interlaced time level, one is effectively able to double the ISB. (Section 3.3.2 demonstrates this for the leapfrog method.) We envision our methods being used for such systems of wave equations.

We first consider staggered versions of the Adams–Bashforth and backwards differentiation time integrators, denoted ABS and BDS respectively. To illustrate our notation, we show in Figure 3.4 four different ways of representing the third order ABS method (ABS3): a representative stencil, the stencil coefficients, the polynomials $\rho(z)$ and $\sigma(z)$, and the explicit Taylor formula. Note that all coefficients listed in this paper can be found via Padé expansions (see [11] or Section 2.3).

In Table 3.1 we give for stable BDS methods the shape and coefficients of the stencil, the error constant, a picture of the stability domain, and the ISB. (Note that by stable, we mean zero-stable, i.e. that all roots of $\rho(z)$ are located within the unit disk, with roots on the unit circle being simple.) Tables 3.2 and 3.3 give the same information for useful ABS and AB methods up to order 8.

time ↑	$\begin{bmatrix} 1 \\ 25/24 \\ -1 \\ -1/12 \\ 1/24 \end{bmatrix}$	$\rho(z) = (z^3 - z^2)$ $\sigma(z) = \frac{1}{24} (25z^2 - 2z + 1) z^{1/2}$ $y(t+k) = y(t) + \frac{k}{24} [25y'(t + \frac{k}{2}) - 2y'(t - \frac{k}{2}) + y'(t - \frac{3k}{2})] + O(k^4)$
<div style="display: flex; flex-direction: column; align-items: center; gap: 5px;"> <div style="border: 1px solid black; width: 10px; height: 10px; margin-bottom: 5px;"></div> <div style="width: 10px; height: 10px; background-color: black; border-radius: 50%; margin-bottom: 5px;"></div> <div style="width: 10px; height: 10px; background-color: black; border-radius: 50%; margin-bottom: 5px;"></div> <div style="width: 10px; height: 10px; background-color: black; border-radius: 50%; margin-bottom: 5px;"></div> <div style="width: 10px; height: 10px; background-color: black; border-radius: 50%; margin-bottom: 5px;"></div> </div>		

Figure 3.4: Four representations of ABS3: stencil shape, coefficients, generating polynomials $\rho(z)$ and $\sigma(z)$, and explicit Taylor formula. Here, \square represents an unknown function value, while \blacksquare / \bullet stands for a known function/derivative value.

Table 3.1: Staggered backwards differentiation time integrators. The normalized local truncation error for BDS_p is $Ck^{p+1}f^{(p+1)}(\eta)$, where C is the error constant. Only stable methods are shown.

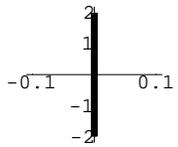
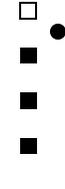
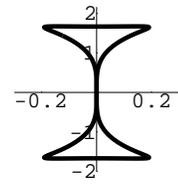
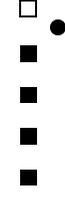
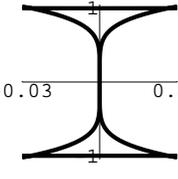
Name	Stencil	Coefficients	Error Constant	Stability Domain	ISB
BDS2 (leapfrog)		$\left[\begin{array}{c c} 1 & \\ \hline -1 & 1 \end{array} \right]$	$\frac{1}{24}$		2
BDS3		$\left[\begin{array}{c c} 1 & \frac{24}{23} \\ \hline -\frac{21}{23} & \\ -\frac{3}{23} & \\ \frac{1}{23} & \end{array} \right]$	$\frac{1}{24}$		$\frac{5}{3} \simeq 1.667$
BDS4		$\left[\begin{array}{c c} 1 & \frac{12}{11} \\ \hline -\frac{17}{22} & \\ -\frac{9}{22} & \\ \frac{5}{22} & \\ -\frac{1}{22} & \end{array} \right]$	$\frac{71}{1920}$		1

Table 3.2: Staggered Adams-Bashforth time integrators. The normalized local truncation error for ABS_p is $Ck^{p+1}f^{(p+1)}(\eta)$, where C is the error constant. Only methods with nonzero ISBs are shown (for orders $p < 10$).

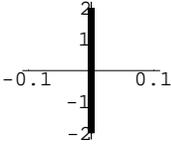
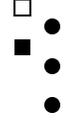
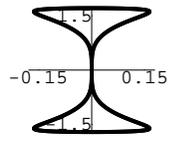
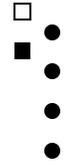
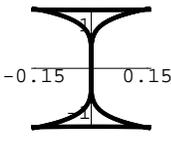
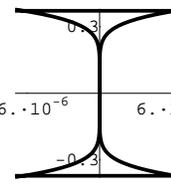
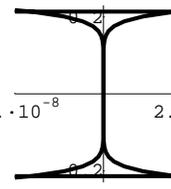
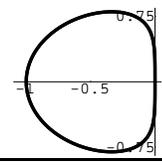
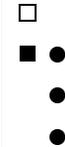
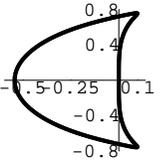
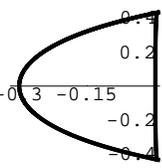
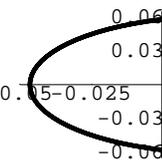
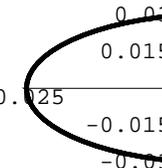
Name	Stencil Shape	Coefficients	Error Constant	Stability Domain	ISB
ABS2 (leapfrog)		$\left[\begin{array}{c c} 1 & \\ \hline -1 & 1 \end{array} \right]$	$\frac{1}{24}$		2
ABS3		$\left[\begin{array}{c c} 1 & 25/24 \\ \hline -1 & -1/12 \\ & 1/24 \end{array} \right]$	$\frac{1}{24}$		$\frac{12}{7} \approx 1.714$
ABS4		$\left[\begin{array}{c c} 1 & 13/12 \\ \hline -1 & -5/24 \\ & 1/6 \\ & -1/24 \end{array} \right]$	$\frac{223}{5760}$		$\frac{4}{3} \approx 1.333$
ABS7		$\left[\begin{array}{c c} 1 & 1152511/967680 \\ \hline -1 & -7969/10752 \\ & 134881/107520 \\ & -294659/241920 \\ & 76921/107520 \\ & -12629/53760 \\ & 32119/967680 \end{array} \right]$	$\frac{1111}{35840}$		$\frac{30240}{81469} \approx 0.371$
ABS8		$\left[\begin{array}{c c} 1 & 295627/241920 \\ \hline -1 & -103021/107520 \\ & 102437/53760 \\ & -2228531/967680 \\ & 24197/13440 \\ & -95251/107520 \\ & 121049/483840 \\ & -1111/35840 \end{array} \right]$	$\frac{13528301}{464486400}$		$\frac{4320}{20209} \approx 0.214$

Table 3.3: Nonstaggered Adams-Bashforth time integrators. The normalized local truncation error for AB_p is $Ck^{p+1}f^{(p+1)}(\eta)$, where C is the error constant. Other than AB_2 , only methods with nonzero ISBs are shown (for orders $p < 10$).

Name	Stencil Shape	Coefficients	Error Constant	Stability Domain	ISB
AB2		$\begin{bmatrix} 1 & & \\ -1 & & 3/2 \\ & & -1/2 \end{bmatrix}$	$\frac{5}{12}$		0
AB3		$\begin{bmatrix} 1 & & \\ -1 & & 23/12 \\ & & -4/3 \\ & & 5/12 \end{bmatrix}$	$\frac{3}{8}$		$\frac{12}{5\sqrt{11}} \approx 0.724$
AB4		$\begin{bmatrix} 1 & & \\ -1 & & 55/24 \\ & & -59/24 \\ & & 37/24 \\ & & -3/8 \end{bmatrix}$	$\frac{251}{720}$		$\frac{52}{15\sqrt{65}} \approx 0.430$
AB7		$\begin{bmatrix} 1 & & \\ -1 & & 198721/60480 \\ & & -18637/2520 \\ & & 235183/20160 \\ & & -10754/945 \\ & & 135713/20160 \\ & & -5603/2520 \\ & & 19087/60480 \end{bmatrix}$	$\frac{5257}{17280}$		≈ 0.058
AB8		$\begin{bmatrix} 1 & & \\ -1 & & 16083/4480 \\ & & -1152169/120960 \\ & & 242653/13440 \\ & & -296053/13440 \\ & & 2102243/120960 \\ & & -115747/13440 \\ & & 32863/13440 \\ & & -5257/17280 \end{bmatrix}$	$\frac{1070017}{3628800}$		≈ 0.029

The ABS and BDS methods of order 2 are both equivalent to the leapfrog method. ABS and BDS methods are both explicit (whereas nonstaggered BD methods are implicit). AB and ABS methods are always stable; BDS methods are stable for orders up through 4 (while nonstaggered BD methods are stable for orders up through 6). Note that either $\rho(z)$ or $\sigma(z)$ has a $z^{1/2}$ factor because of the staggering setup.

Stability domains for staggered methods are symmetric with respect to both coordinate axes; one can see this by noting that there is symmetry across the x -axis (true of all stability domains because $\xi(\theta) = \xi(-\theta)$) as well as symmetry about the origin (which comes from the structure of staggered methods: ξ is an odd function of $r = e^{i\theta}$). Because of these symmetries and because stability domains must approach $\xi = 0$ vertically (near the origin, $\xi(\theta) \approx i\theta$), staggered methods have no real axis coverage. Thus, these methods are only appropriate for propagation problems. (However, through exponential time-stepping [27], the schemes can also be applied to problems such as attenuation in Maxwell's equations for lossy media.)

The exact ISBs in Tables 3.1, 3.2, and 3.3 were found by solving for θ in the equation

$$\text{Real} \left[\frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})} \right] = 0. \quad (3.16)$$

For staggered methods, the ISB occurs at $\theta = \frac{\pi}{2}$ because of stability domain symmetries. For AB3, the ISB occurs at $\theta = \arccos\left(\frac{1}{10}\right)$. For AB4, the ISB occurs at $\theta = \arccos\left(-\frac{4}{9}\right)$. The ISB can be found from these values by substituting θ into $\frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})}$.

As will be discussed in section 3.6.1, AB methods have a non-zero ISB only for methods of order 3, 4, 7, 8, 11, 12, etc.; ABS methods additionally include order 2. Note that the error constants for the staggered methods are approximately nine to ten times smaller than those of the nonstaggered methods of equivalent orders. In

addition, staggering increases the ISB by a factor of 2.4-7.4, with the factor growing as order increases.

We can also compare the staggered methods to Störmer methods [13] in those cases for which the problem can be reformulated as a second order system, $u_{tt} = F(t, u)$, $v_{tt} = G(t, v)$. With compatible definitions we find ISBs of around 2, 1.73, 1.41, 1.11, 0.84, 0.62, and 0.46 for orders 2–8. The associated error constants are approximately 0.083, 0.083, 0.079, 0.075, 0.071, 0.068, and 0.066. Thus the ABS methods compare favorably for orders of accuracy four and less, and unfavorably thereafter. However, formulating wave equations using two time derivatives sometimes creates difficulties in implementing boundary conditions.

To implement one of the time-staggered methods, one needs to obtain starting values for several time levels after the initial condition. For nonstaggered multistep methods, this is usually accomplished with a Runge–Kutta method. For staggered time integrators, one should obtain as many (half-integer) levels of u and v as needed using a nonstaggered Runge–Kutta method and then select out those needed to interlace u and v appropriately.

3.5 Staggered free parameter multistep methods

(This section is the author’s work.)

We have developed multistep methods that allow free parameters due to suboptimization of order. These free parameters may be used to decrease the error constant, increase the ISB, or often both. In this section, we discuss staggered free parameter schemes and offer two examples of such methods as illustrations of opportunities available in this area. Appendix G discusses nonstaggered free parameter schemes.

3.5.1 Fourth order free parameter method: $\frac{7}{2}$ -step, one parameter

We consider methods with stencils of the following form:

$$\begin{array}{c} \square \\ \bullet \\ \blacksquare \\ \blacksquare \\ \bullet \\ \bullet \end{array} \left[\begin{array}{c} 1 \\ \alpha_2 \\ \alpha_1 \\ \beta_3 \\ \beta_2 \\ \beta_1 \\ \beta_0 \end{array} \right]$$

While it is possible to find a fifth order method of this form, it is not stable (see Section 3.6.2). We instead search for fourth order methods containing one free parameter. This gives a linear system of equations which has the following solution with parameter β_1 .

$$\left[\begin{array}{c} 1 \\ 24\beta_1 - 5 \\ -24\beta_1 + 4 \end{array} \middle| \begin{array}{c} \beta_1 + \frac{11}{12} \\ 22\beta_1 - \frac{31}{8} \\ \beta_1 \\ -\frac{1}{24} \end{array} \right] \quad (3.17)$$

This method is stable for $\beta_1 \in (\frac{1}{8}, \frac{5}{24}]$. We note that for $\beta_1 = \frac{1}{6}$, we recover ABS4.

This method's error constant is $C = \frac{97-136\beta_1}{5760(8\beta_1-1)}$. As $\beta_1 \rightarrow \frac{1}{8}$, the ISB of the method approaches ≈ 1.714 but the error constant becomes unbounded. We thus observe a trade-off between accuracy and stability for this method. This is illustrated in Figure 3.5.1. If one is willing to sacrifice some accuracy to gain in stability, this method would provide the means to do so.

As an example, we give the stability domain of the method that has $\beta_1 = 0.126$ in Figure 3.5.1. This method has an ISB of ≈ 1.708 and an error constant of ≈ 2.48 .

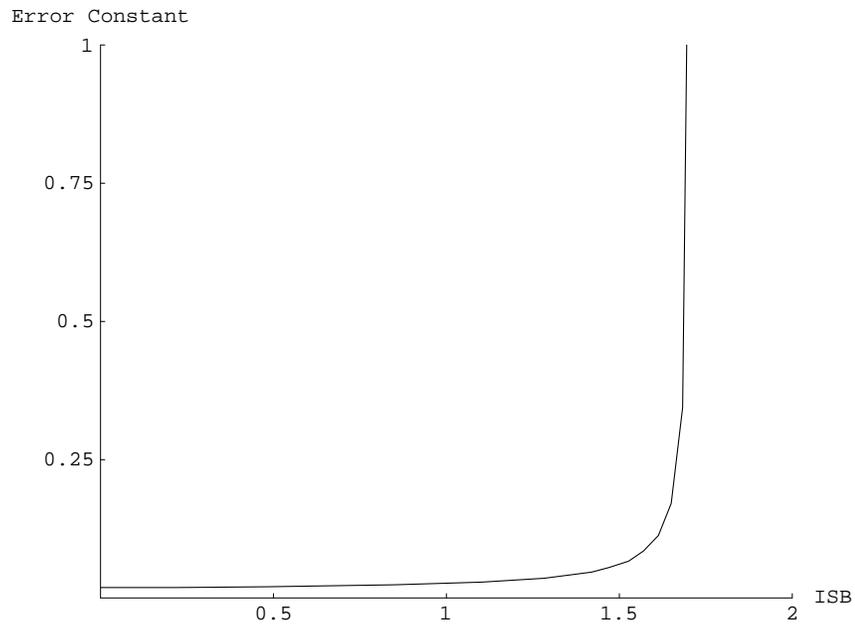


Figure 3.5: Trade-off between accuracy (error constant) and stability (ISB) for method (3.17)

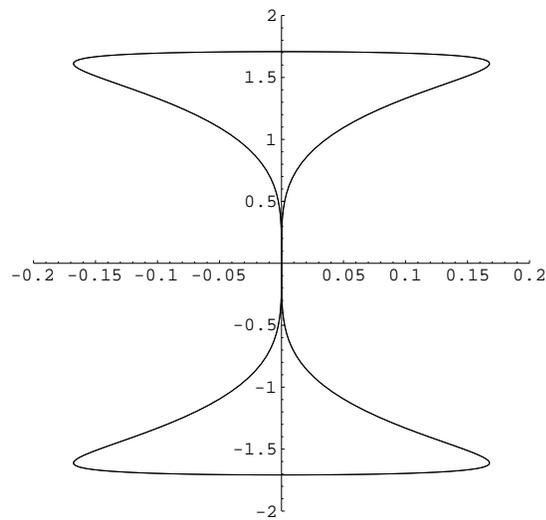


Figure 3.6: Stability domain of method (3.17) for $\beta_1 = 0.126$

3.5.2 Fourth order free parameter method: 4-step, two parameter

We consider methods with stencils of the following form:

$$\begin{array}{c} \square \\ \bullet \\ \blacksquare \\ \bullet \\ \blacksquare \\ \bullet \\ \blacksquare \\ \bullet \\ \blacksquare \end{array} \left[\begin{array}{c} 1 \\ \alpha_3 \\ \beta_3 \\ \alpha_2 \\ \beta_2 \\ \alpha_1 \\ \beta_1 \\ \alpha_0 \end{array} \right]$$

We search for fourth order methods containing two free parameters. The linear system of equations to be solved gives the following solution with parameters s and t .

$$\left[\begin{array}{c} 1 \\ -\frac{17}{22} + \frac{577}{528}s - \frac{1}{24}t \\ \frac{-9}{22} - \frac{201}{176}s + \frac{9}{8}t \\ \frac{5}{22} + \frac{9}{176}s - \frac{9}{8}t \\ \frac{1}{22} - \frac{1}{528}s + \frac{1}{24}t \end{array} \left| \begin{array}{c} \frac{12}{11} + \frac{1}{22}s \\ s \\ t \end{array} \right. \right] \quad (3.18)$$

Note that for $s = 0$ and $t = 0$, we recover BDS4. The set of (s, t) -values for which this scheme is stable is roughly a triangle in the (s, t) -plane with vertices at approximately $(-1.98, 1.0016)$, $(-0.3, -0.8)$, and $(1.5, 1.16)$.

The error constant of this scheme is

$$C = \frac{(1704 - 127s - 198t)}{1920(24 + 23s + 22t)}. \quad (3.19)$$

By choosing various values of the parameters for which the method is stable, we can change the error constant and the ISB of the method. As the ISB approaches the theoretical limit of 2 (s and t approach the upper left-hand corner of the triangle of stability), the error constant becomes unbounded. We thus again observe a trade-off between accuracy and stability.

We give three examples of interest:

- $s = -1, t = 1.045$. ISB ≈ 1.8822 , error constant $C \approx 0.03526$. While the error constant is comparable to that of ABS4, there is a dramatic improvement in the ISB. We show the stability domain of this method in Figure 3.7(a).
- $s = 0.74, t = 1.121$. ISB ≈ 1.337 , $C \approx 0.0110$. This method improves on the accuracy of ABS4 by about a factor of 3 while maintaining about the same ISB.
- $s = -1.95, t = 1.00155$, ISB ≈ 1.995 , $C \approx 32.34$. This method has an ISB very close to 2, the theoretical limit. See Figure 3.7(b) for the stability domain of this method.

Note that these free parameter methods require no more function evaluations than ABS4; all multistep methods require only one function evaluation per time step. As we have not explored the properties of these free parameter schemes in any great detail, we exclude them from further analysis.

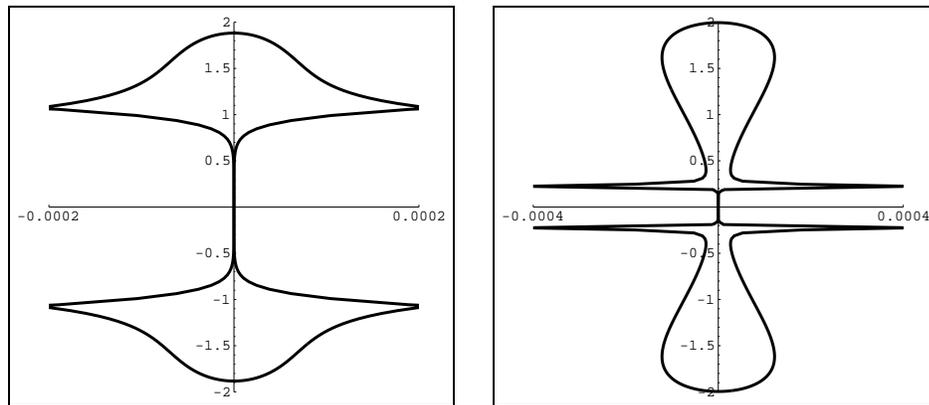


Figure 3.7: Stability domains of the fourth order staggered free parameter scheme (3.18): (a) ISB ≈ 1.8822 , $C \approx 0.0353$ (b) ISB ≈ 1.995 , $C \approx 32.34$

3.6 Theoretical Considerations

(Proof of the first result was done by Bengt Fornberg and by the author, while proof of the second result was done by the author.)

3.6.1 Imaginary Stability Boundary of Adams–Bashforth methods

Adams–Bashforth methods have non-zero ISBs only for orders 3, 4, 7, 8, 11, 12, etc. Staggered AB methods additionally include order 2 (leapfrog). Nonstaggered Adams–Moulton methods (implicit versions of Adams–Bashforth methods) have non-zero ISBs only for orders 1, 2, 5, 6, 9, 10, etc. Proofs of these results are given in Appendix H.

3.6.2 Staggered analogue of Dahlquist’s First Stability Barrier

Dahlquist’s First Stability Barrier for multistep methods states that the order p of an explicit stable m -step method must satisfy $p \leq m$ [5]. The analogue of this theorem for staggered multistep methods is:

The order p of an explicit stable m -step staggered method satisfies

$$p \leq \begin{cases} m & , \quad m \text{ an even integer} \\ m + \frac{1}{2} & , \quad m \text{ a half-integer} \\ m + 1 & , \quad m \text{ an odd integer} \end{cases} \quad (3.20)$$

Our proof of this theorem follows those of Jeltsch and Nevanlinna [19] and Dahlquist [5] and is given in Appendix I.

3.7 Staggered Predictor-Corrector Methods

We have investigated staggered predictor-corrector methods and found that these methods do not hold as much promise as staggered multistep and Runge–Kutta methods. We discuss our results here.

Our goal was to find a staggered predictor-corrector method with a large ISB. From Section 3.3.2, we know that because predictor-corrector methods require two function evaluations per time step, the largest possible ISB for staggered predictor-corrector methods is 4. We considered various combinations of ABS and

AMS methods as well as other methods. The maximum possible ISB that we found was ≈ 1 , which was for ABS4 (predictor)/AMS4 (corrector). After normalizing this for comparison to multistep methods, we find that it is far inferior to ABS4 and BDS4. We believe that these schemes do not hold promise because implicit staggered multistep methods are too implicit to be of practical use (i.e. because of staggering, y_{n+1} depends implicitly on f_{n+2} rather than f_{n+1}). We thus did not further consider staggered predictor-corrector methods.

3.8 Staggered Runge–Kutta methods

(The first subsection was researched by Toby Driscoll while the second subsection was researched by the author.)

Multistage methods can also be put into a staggering framework. We rewrite the ordinary differential equation in the form

$$\begin{aligned} u' &= f(t, v(t)) \\ v' &= g(t, u(t)). \end{aligned} \tag{3.21}$$

The splitting into u and v (each could be a vector) allows quantities to be given at offset time levels, as suggested in section 3.2. The splitting into f and g reflects that values of u' (or v') are given at time levels staggered with respect to u (or v).

3.8.1 Advancing u and v separately

One form for a staggered Runge–Kutta (RKS) method is

$$\begin{aligned}
 d_1 &= kf(t_{n+1/2}, v_{n+1/2}) \\
 d_2 &= kg(t_n + c_2k, u_n + a_{21}d_1) \\
 d_3 &= kf(t_{n+1/2} + c_3k, v_{n+1/2} + a_{32}d_2) \\
 d_4 &= kg(t_n + c_4k, u_n + a_{41}d_1 + a_{43}d_3) \\
 &\vdots \\
 d_s &= kf(t_{n+1/2} + c_s k, v_{n+1/2} + a_{s2}d_2 + \cdots + a_{s,s-1}d_{s-1}) \\
 u_{n+1} &= u_n + b_1d_1 + b_3d_3 + \cdots + b_sd_s,
 \end{aligned} \tag{3.22}$$

if s is odd. (If s is even, the first stage should be an evaluation of g at time t_n , and the stages used to advance from u_n to u_{n+1} are the even-numbered ones.) The same formula can then be used to advance v , once references to f and g are switched and time levels are shifted forward by $\frac{1}{2}$. Observe that advancing both u and v by one step requires s evaluations each of f and g . The form of the governing equations in (3.21) suggests that an evaluation of both f and g should count as one stage, so (3.22) is an s -stage method.

The coefficients in such a formula can be derived by straightforward, if laborious, Taylor expansion of both the exact difference

$$u(t = k(n + 1)) - u(t = kn)$$

and the RKS approximation

$$b_1d_1 + \cdots + b_sd_s.$$

The expansions must be made so that v (and consequently f) is evaluated only at

$t_{n+1/2}$, and u (hence g) is evaluated at t_n . If more than a few stages are desired, a symbolic computational package is useful both for generating these expansions and for solving the system of nonlinear equations that results from equating their coefficients.

Stability analysis follows the usual pattern. The model problem is linear:

$$\begin{bmatrix} u \\ v \end{bmatrix}' = \begin{bmatrix} 0 & \lambda \\ \lambda & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}.$$

(Using different scalars in the off-diagonal entries of the matrix does not change anything essential because the eigenvalues depend only on the product of those entries.) In applying the RKS method to the model problem, one finds that

$$\begin{aligned} u_{n+1} &= \beta(k\lambda) v_{n+1/2} + \alpha(k\lambda) u_n \\ v_{n+3/2} &= \beta(k\lambda) u_{n+1} + \alpha(k\lambda) v_{n+1/2}, \end{aligned}$$

where β and α are polynomials in $k\lambda$. We can thus write

$$\begin{bmatrix} u_{n+1} \\ v_{n+3/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\beta & 1 \end{bmatrix}^{-1} \begin{bmatrix} \alpha & \beta \\ 0 & \alpha \end{bmatrix} \begin{bmatrix} u_n \\ v_{n+1/2} \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \alpha\beta & \alpha + \beta^2 \end{bmatrix} \begin{bmatrix} u_n \\ v_{n+1/2} \end{bmatrix} = Q(k\lambda) \begin{bmatrix} u_n \\ v_{n+1/2} \end{bmatrix}.$$

The stability region consists of all values of $k\lambda$ for which both eigenvalues of $Q(k\lambda)$ are inside the unit circle, or simple and on the unit circle. After a short calculation, one finds that $[1 \ w]^T$ is an eigenvector if and only if

$$w^2 = \beta(k\lambda)w + \alpha(k\lambda), \tag{3.23}$$

and the corresponding eigenvalue is w^2 . (This is the same equation that arises when using the ansatz $u_n = w^{2n}$, $v_{n+1/2} = w^{2n+1}$.) The two roots w of (3.23) thus determine the stability region.

As was mentioned in section 3.3.1, we normalize the stability region by the number of stages in order to make a fair comparison to one-stage methods. An error constant can also be defined by looking at the first error term in the approximate solution of the linear model problem. This too should be normalized by a factor s^p for a p^{th} order method.

We recognize leapfrog as a 1-stage RKS method of order 2. Computation of the expansions for 2-stage and 3-stage methods reveals that neither has enough additional free parameters to improve upon the order of leapfrog. While 4-stage, third order methods do exist, they do not seem to improve on their nonstaggered counterparts.

The first interesting higher order method is the five-stage RKS method. Here there are 13 constants to be determined in the formula. To achieve fourth order accuracy, 21 conditions (most of which are nonlinear) must be satisfied. Remarkably, there is a family of solutions parameterized by b_5 . With $\gamma = (6b_5)^{-1/2}$, the tableau for the general solution is

$$\begin{array}{c|ccc}
 & 0 & & \\
 \frac{1}{4}(2 - \gamma) & \frac{1}{4}(2 - \gamma) & & \\
 -\frac{1}{2}\gamma & & -\frac{1}{2}\gamma & \\
 \frac{1}{4}(2 + \gamma) & \frac{1}{4}(2 + \gamma) & & 0 \\
 \frac{1}{2}\gamma & & 0 & \frac{1}{2}\gamma \\
 \hline
 & 1 - 2b_5 & b_5 & b_5
 \end{array} \tag{3.24}$$

(Entries which are blank are zero for structural reasons.) The stability region and error constant are independent of the choice of the free parameter. The most appealing member of the family, which we call RKS4, is given with $b_5 = 1/24$ and hence

$\gamma = 2$:

$$\begin{aligned}
 d_1 &= kf(t_{n+1/2}, v_{n+1/2}) \\
 d_2 &= kg(t_n, u_n) \\
 d_3 &= kf(t_{n+1/2} - k, v_{n+1/2} - d_2) \\
 d_4 &= kg(t_n + k, u_n + d_1) \\
 d_5 &= kf(t_{n+1/2} + k, v_{n+1/2} + d_4) \\
 u_{n+1} &= u_n + \frac{11}{12}d_1 + \frac{1}{24}d_3 + \frac{1}{24}d_5.
 \end{aligned} \tag{3.25}$$

Observe that while 5 stages are required, the stage d_1 is actually equivalent to the future stage d_2 for the advance of v from time level $n + \frac{1}{2}$ to $n + \frac{3}{2}$. Hence only four evaluations each of f and g are needed to advance both u and v one time step, and we consider this to be a four-stage method for purposes of normalization of the ISB and of the error constant.

RKS4 has a simple interpretation. Given the original data u_n and $v_{n+1/2}$, leapfrog is used repeatedly to estimate $v_{n-1/2}$, u_{n+1} , and $v_{n+3/2}$ in succession. The three estimates of v values are then combined according to a finite difference stencil (AMS4) to relate u_n to the new value of u_{n+1} . The method is fourth order due to a symmetry which produces cancellation in the leapfrog errors.

The stability region of RKS4 is a segment of the imaginary axis, and the normalized ISB of this method is about 1.425 (see Figure 3.8(a)). Hence for equivalent amounts of work per step, time steps about twice as large as those of standard RK4 are possible. The error constant is $1/1920$, compared to $1/120$ for RK4. (After normalization for comparison to one-stage methods, these constants become $2/15$ and $32/15$ respectively.) As with multistep methods, the problem may in many cases be written as a second order system in time. The three-stage, fourth order Nyström method presented in [13] has an equivalent normalized ISB of about 0.86, far less than that of RKS4.

Notice that stages 4 and 5 are independent of stages 2 and 3. The storage requirements can therefore be kept low. In the following procedure, time dependence and subscripts on u and v are omitted for clarity, and z_1 is assumed to start with the value $kg(u)$, obtained from the previous advance of v .

$$\begin{aligned}
 z_1 &\leftarrow v - z_1 \\
 z_1 &\leftarrow kf(z_1) \\
 z_2 &\leftarrow kf(v) \\
 z_3 &\leftarrow u + z_2 \\
 z_3 &\leftarrow kg(z_3) \\
 z_3 &\leftarrow v + z_3 \\
 z_3 &\leftarrow kf(z_3) \\
 u &\leftarrow u + z_1/24 + 11z_2/12 + z_3/24.
 \end{aligned}$$

At the end of this procedure, z_2 holds the value that serves as stage 2 of the next advance of v . Only three temporary variables are needed. Each needs to have as many components as the larger of u and v . In the common situation where u and v each hold half of the variables of the system, the additional storage is equivalent to $3/2$ of the total number of unknowns. In standard fourth order Runge–Kutta, the best temporary storage is twice the number of unknowns.

3.8.2 Advancing u and v together

The RKS method suggested above evaluates a number of stages to advance u , then a new set of independent stages to advance v (although RKS4 can reuse one stage). An alternative is to use a joint set of stages to advance u and v simultaneously. While still using the same number $2s$ of individual f and g evaluations per time step as the other s -stage staggered methods, a potential advantage here is that the number

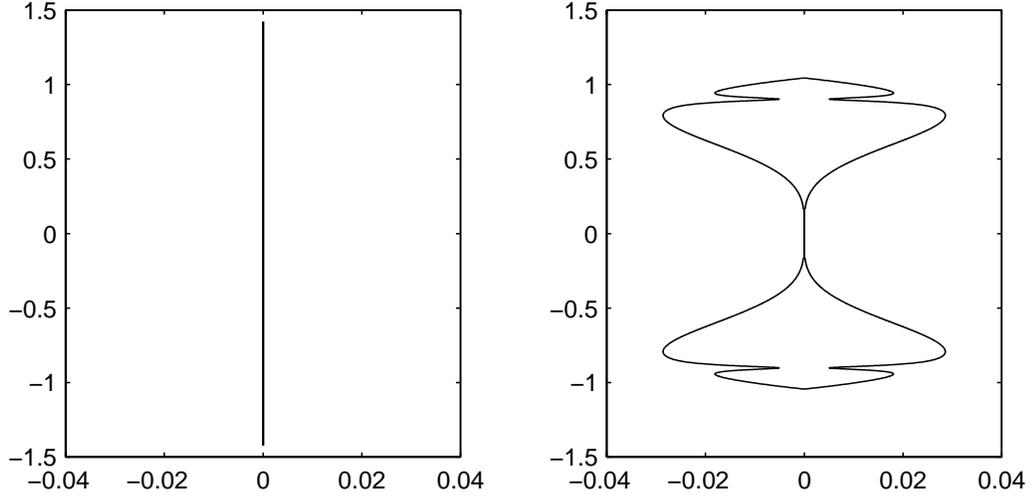


Figure 3.8: Stability domains for staggered Runge–Kutta methods: (a) order 4 (eqn. (3.25)) (b) order 3 (eqn. (3.26))

of free constants grows more quickly with s .

We have explored third order methods that use 6 (half-)stages to advance both u and v . There are 16 (mostly) nonlinear equations in 20 variables for this case. We have found three families of solutions to these equations, with the most interesting case having just one free parameter. This third order, three-stage method is given here:

$$\begin{aligned}
 d_1 &= kf(t_{n+1/2}, v_{n+1/2}) \\
 d_2 &= kg(t_n, u_n) \\
 d_3 &= kf(t_{n+1/2} - \frac{1}{2}k, v_{n+1/2} - \frac{1}{2}d_2) \\
 d_4 &= kg(t_n + \frac{13}{12}k, u_n + \frac{13}{12}d_1) \\
 d_5 &= kf(t_{n+1/2} + \frac{1}{2}k, v_{n+1/2} + \frac{7}{26}d_2 + \frac{3}{13}d_4) \\
 d_6 &= kg(t_n + \frac{13}{12}k, u_n + (\frac{91}{72} - 2\gamma)d_1 + (-\frac{13}{72} + \gamma)d_3 + \gamma d_5) \\
 u_{n+1} &= u_n + \frac{2}{3}d_1 + \frac{1}{6}d_3 + \frac{1}{6}d_5 \\
 v_{n+3/2} &= v_{n+1/2} + \frac{1}{13}d_2 + \frac{6}{13}d_4 + \frac{6}{13}d_6.
 \end{aligned} \tag{3.26}$$

Here γ is a constant that affects the accuracy and stability of the method. The ISB is approximately optimized if $\gamma = 104/181$. For this choice, the normalized ISB is ≈ 1.044 and the normalized error constant is $27\gamma/24 \approx 0.6464$. The stability region is displayed in Figure 3.8(b). For comparison, classical RK3 has a normalized ISB of $1/\sqrt{3} \approx 0.577$ and normalized error constant of $9/8 = 1.125$.

We have by no means exhausted the possibilities for either type of staggering in Runge–Kutta methods; our intent has been to demonstrate that such methods do exist and can improve on their nonstaggered counterparts.

3.9 Root portraits

(The work in this section was done primarily by Toby Driscoll after the concept was introduced by Bengt Fornberg.)

Since our goal is to perform time-stepping for wave equations, it is illuminating to compare methods based on their performance on the one-dimensional scalar wave equation,

$$\begin{aligned} u_t &= v_x \\ v_t &= u_x. \end{aligned} \tag{3.27}$$

We think of the spatial domain as unbounded and spatial derivatives as exact. For a Fourier mode whose spatial dependence is $e^{i\omega x}$, the wave equation becomes the ODE system

$$\begin{bmatrix} u \\ v \end{bmatrix}_t = \begin{bmatrix} 0 & i\omega \\ i\omega & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}. \tag{3.28}$$

These equations allow leftgoing and rightgoing modes. When a mode is advanced in time by an amount k , the solution is multiplied by a factor $e^{\pm ik\omega}$, with the sign determining only direction of travel.

For classical multistep methods, the analysis reduces to the situation familiar from linear stability. The numerical solution is capable of travel in either direction, and to advance a mode by time k the solution is multiplied by a factor $z(ik\omega)$. For linear multistep methods, z is a root of the characteristic polynomial equation

$$\rho(z) - ik\omega\sigma(z) = 0. \quad (3.29)$$

When $k\omega = 0$, a stable method has exactly one root at $z = 1$. As $ik\omega$ travels along the imaginary axis, this root approximates the exact factor $e^{ik\omega}$ (or its conjugate) but eventually becomes noticeably different. The other roots of the characteristic polynomial are physically irrelevant (as long as they are inside the unit circle). When $k\omega$ is larger than the ISB of the method, some root is outside the unit disk and the method becomes unstable.

To visualize this process, we draw a “root portrait” that traces the physically relevant root as $k\omega$ takes on all stable values. An example for AB3 is shown in Figure 3.9.

A point $ik\omega$ on the imaginary axis should ideally map to $e^{\pm ik\omega}$ on the unit circle, as the tick marks outside the unit circle suggest. The physically relevant root, as determined by the characteristic polynomial, is perfect at the origin and a good approximation nearby, but eventually the path of the root diverges from the circle. When $ik\omega$ encounters the boundary of the stability region, one of the parasitic roots not shown is just crossing the unit circle on its way to creating time instability.

A similar analysis can be made for classical Runge–Kutta methods. Here the characteristic polynomial is linear in z , but there is a polynomial dependence on $ik\omega$. (For orders p less than five, this polynomial is just the p^{th} order Taylor polynomial for $e^{ik\omega}$.) Also, the stability region must be normalized by the number s of stages in the method, and the s^{th} root of z must be taken in accordance. Because

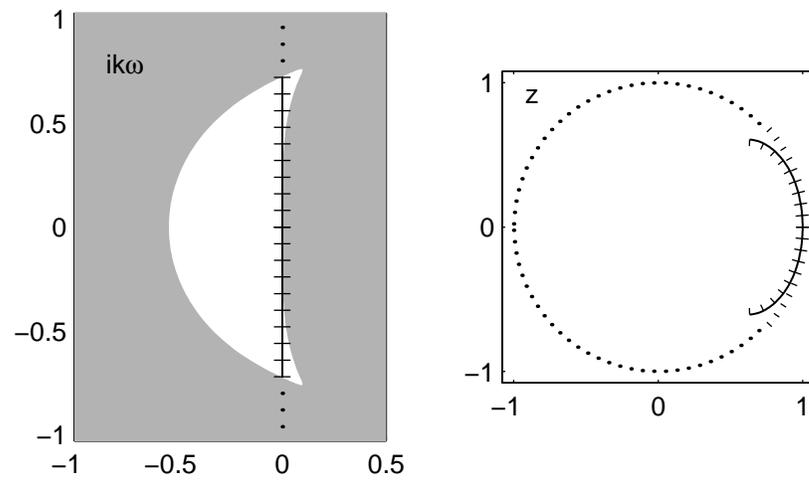


Figure 3.9: Example of a “root portrait.” The portion of the imaginary axis which lies inside the stability region of AB3 is mapped to the physically relevant roots inside the unit circle. Ideally, the evenly spaced tick marks along the unit circle should line up with the tick marks along the root path, but this is true only near the origin.

there is only one root, the physically relevant root also determines stability.

For staggered multistep schemes, the characteristic equation is again (3.29). This is now a polynomial in $z^{1/2}$, and the roots are easily found. Again only one root per direction of travel is physically relevant. For staggered Runge–Kutta methods, the stability analysis in section 5 applies; in fact, $z^{1/2}$ is just the variable w in the characteristic equation (3.23), and λ is purely imaginary in that formula.

Figure 3.10 displays the root portraits for classical and staggered methods of orders 2, 3, 4, and 7. As the order of a method increases, inner and outer ticks match up more accurately near $z = 1$. The stability restriction is made clear by where the tick marks on the unit circle end. The AB2 and RK2 methods are stable but have zero ISB. The ABS2, BDS2, and RKS2 methods are all equivalent to leapfrog, which has the maximum possible ISB of 2. In every case, staggered schemes are seen to have stability and accuracy properties superior to their nonstaggered counterparts.

Another way to view the root portraits is in terms of numerical dissipation and dispersion. Because we have eliminated the spatial discretization errors, root portraits clearly show the errors solely due to the time stepping schemes. The amount of numerical dissipation in a scheme is shown by how close the path of the root portrait stays to the unit circle, whereas the amount of dispersion is shown by how well the inner ticks on the root portrait path match up to the outer ticks on the unit circle. For example, ABS2/BDS2/RKS2 (leapfrog) and RKS4 have no dissipation because all roots stay on the unit circle but have significant dispersion near the edge of the stability domain because the inner and outer ticks do not match well there.

3.10 Numerical experiments

(The work in this section was done by the author.)

The root portrait data can be used to experimentally compare AB, ABS, BDS, RK, and RKS time integrators for wave propagation. As discussed in the previous section, when solving equation (3.27), one can model the effect that a

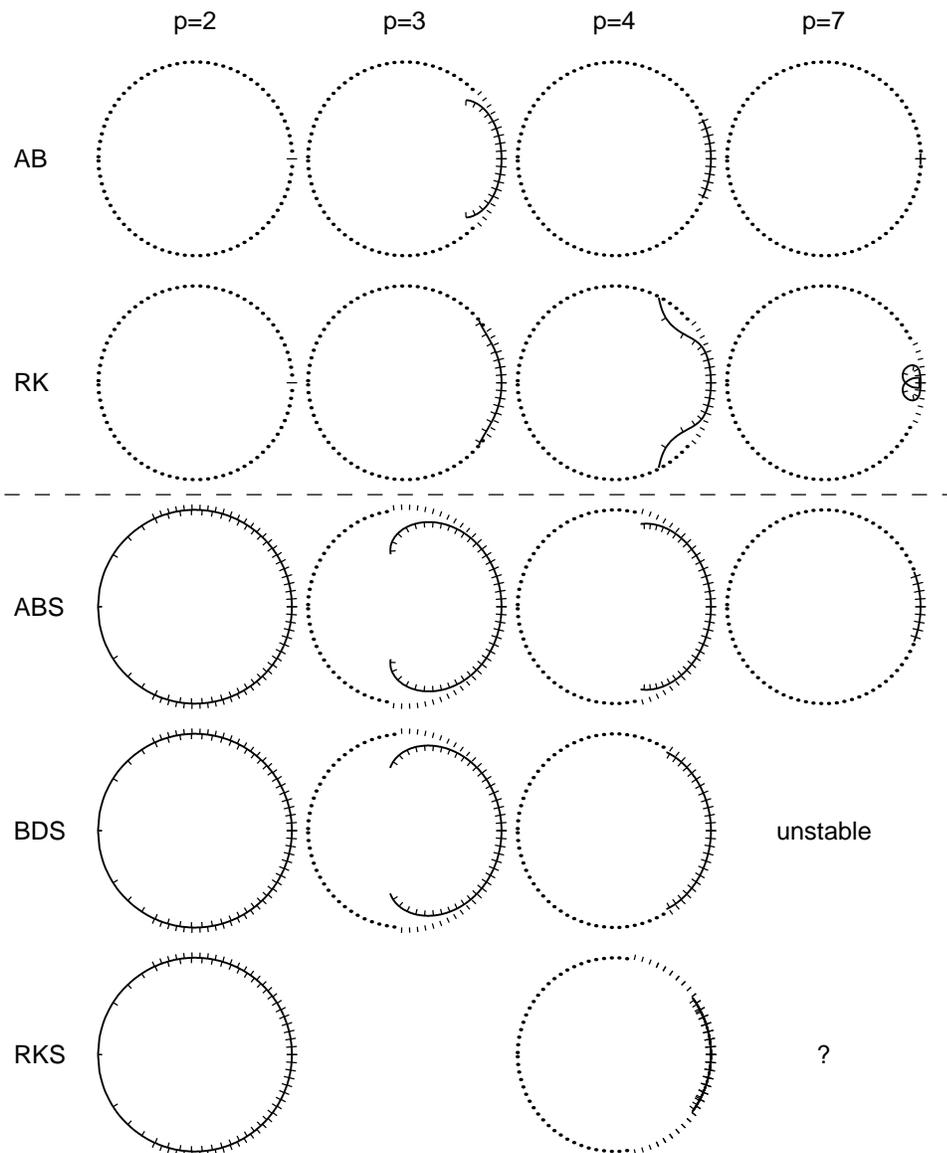


Figure 3.10: Root portraits for classical and staggered methods of different orders. Stability of the methods for the wave equation is reflected by the length of the arc made by the tick marks on the unit circle. Accuracy is judged by the matching of inner and outer ticks along the root paths (solid lines). In the case of RKS4, the root path doubles back on the unit circle in the wrong direction; those tick marks are omitted for clarity. The existence of an RKS7 formula is unknown, and we do not yet have a useful RKS3 method for which this analysis is appropriate.

particular numerical time integrator has on a particular Fourier mode $e^{i\omega x}$ by solving equation (3.29) for the physical root $z(ik\omega)$. We choose the physical root so that the solution moves strictly to the right. Then the solution at the n^{th} time step is given by

$$\begin{aligned} u(t = kn) &= z^n e^{i\omega x} \\ v(t = kn) &= -z^n e^{i\omega x}. \end{aligned} \tag{3.30}$$

We use the initial condition

$$u(x, 0) = \begin{cases} (1 + \cos(\frac{x}{0.15}))^2, & |x| < 0.15 \\ 0 & , |x| \geq 0.15 \end{cases} \tag{3.31}$$

and advance the solution to final time $T = 6\pi$, so that the exact final solution is the same as the initial condition. We define N to be the number of function evaluations used to advance the solution from $T = 0$ to $T = 6\pi$, i.e. the number of time steps taken multiplied by the number of stages of the time integrating method. This provides a legitimate comparison between one-step methods like AB, ABS, and BDS and multistage methods like RK and RKS.

The stability restriction for this problem is $k\omega_{max} < ISB$, where $k = \frac{6\pi}{N}$ and $\omega_{max} = \frac{M}{2}$. Thus we have

$$N > \frac{3\pi M}{ISB}. \tag{3.32}$$

We used $M = 64$ for the experiments in this section.

Figure 3.11 shows a sample run of this method for third order ABS using $N = 375$: the initial condition (and exact solution at time $T = 6\pi$), the numerical solution at $T = 6\pi$, and the error in the numerical solution. In our comparison tables given later, we show only the error in the numerical solution. In order to address the

numerical dissipation of the schemes, we have included the relative loss of energy in the discrete L^2 -norm in the upper-right corner of the error plots.

Table 3.4 shows the error in running the second order leapfrog (ABS2, BDS2, RKS2) method for $N = 500$ and $N = 1000$. Table 3.5 compares the errors obtained by running AB3, ABS3, and BDS3 for $N = 500$ and $N = 1000$, while Table 3.6 shows the errors resulting from running AB4, RK4, ABS4, BDS4, and RKS4 for $N = 800$ and $N = 1600$. Finally, Table 3.7 compares the errors in AB7 and ABS7 for $N = 2000$ and $N = 4000$, while Table 3.8 compares the errors in AB8 and ABS8 for $N = 3000$ and $N = 6000$.

In all cases, staggered methods are superior to nonstaggered methods in terms of accuracy and stability. While the relative accuracy of nonstaggered versus staggered methods does not change with order, the improvement in stability from using staggered methods continues to improve as order increases. It is also interesting to note that the RKS4 method is inferior to ABS4 and BDS4 in accuracy but marginally better in stability, whereas it improves on RK4 in both respects.

Notice that there is a different character to the error than is customary. Typically, error trains are one-sided due to spatial discretization error. However, as noted in the previous section, we have eliminated spatial discretization errors through the use of the root portrait technique. Thus, the errors shown in these pictures are solely time discretization errors. These error trains are almost symmetric rather than one-sided since the schemes are almost dispersion-free. The amount of dissipation is on the order of machine precision for leapfrog and RKS4 and is reasonably small for the other schemes.

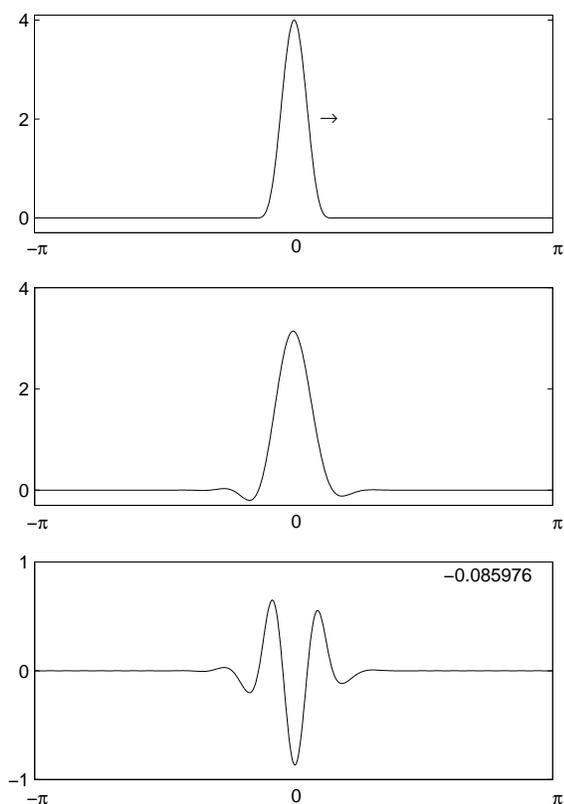


Figure 3.11: Sample run of ABS3 using the physically relevant root and $N = 375$ function evaluations. (a) Initial condition (and exact final solution). Note that we pick the physical root so that the hump moves to the right. (b) Numerical solution at final time $T = 6\pi$. (c) Error in the numerical solution. The relative loss $\left(\frac{\|u_{final}\|_2}{\|u_{initial}\|_2} - 1\right)$ is shown in the upper right-hand corner of the error plot.

Table 3.4: Error given by running the second order leapfrog method using the root portrait technique. Leapfrog is the only classical second order multistep method that has a non-zero ISB. Note that vertical scales differ by $\frac{10}{3}$. The relative loss $\left(\frac{\|u_{final}\|_2}{\|u_{initial}\|_2} - 1\right)$ is shown in the upper right-hand corner of the error plots. N is the number of function evaluations used to reach the final time $T = 6\pi$.

Method	Error at $N = 500$	Error at $N = 1000$
Leapfrog (ABS2) (BDS2) (RKS2)		

Table 3.5: Error given by running third order methods using the root portrait technique. Observe that the vertical scales are the same in all cases and that AB3 is not stable until $N > 834$. The relative loss $\left(\frac{\|u_{final}\|_2}{\|u_{initial}\|_2} - 1\right)$ is shown in the upper right-hand corner of the error plots. N is the number of function evaluations used to reach the final time $T = 6\pi$.

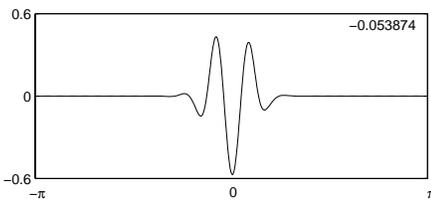
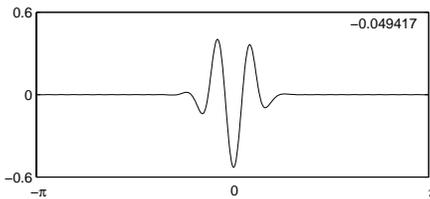
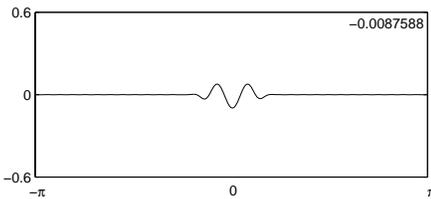
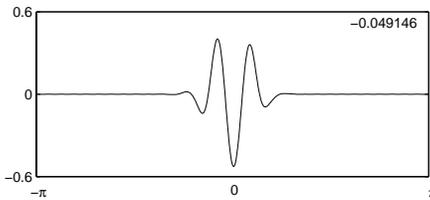
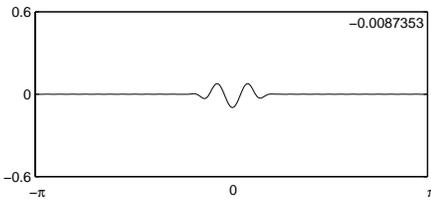
Method	Error at $N = 500$	Error at $N = 1000$
AB3	Unstable	
ABS3		
BDS3		

Table 3.6: (a) Error given by running fourth order nonstaggered methods using the root portrait technique. Note that the vertical scales are the same in all cases and that AB4 is not stable until $N > 1403$ and RK4 is not stable until $N > 854$. N is the number of function evaluations used to reach the final time $T = 6\pi$. The relative loss $\left(\frac{\|u_{final}\|_2}{\|u_{initial}\|_2} - 1\right)$ is shown in the upper right-hand corner of the error plots.

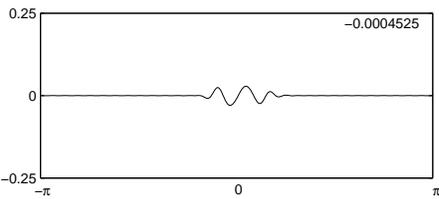
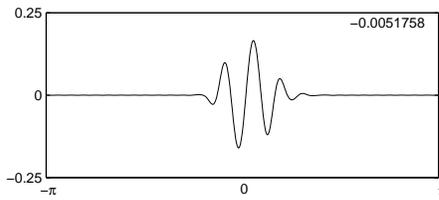
Method	Error at $N = 800$	Error at $N = 1600$
AB4	Unstable	
RK4	Unstable	

Table 3.6: (b) Error given by running fourth order staggered methods using the root portrait technique. Note that the vertical scales are the same in all cases. N is the number of function evaluations used to reach the final time $T = 6\pi$. The relative loss $\left(\frac{\|u_{final}\|_2}{\|u_{initial}\|_2} - 1\right)$ is shown in the upper right-hand corner of the error plots.

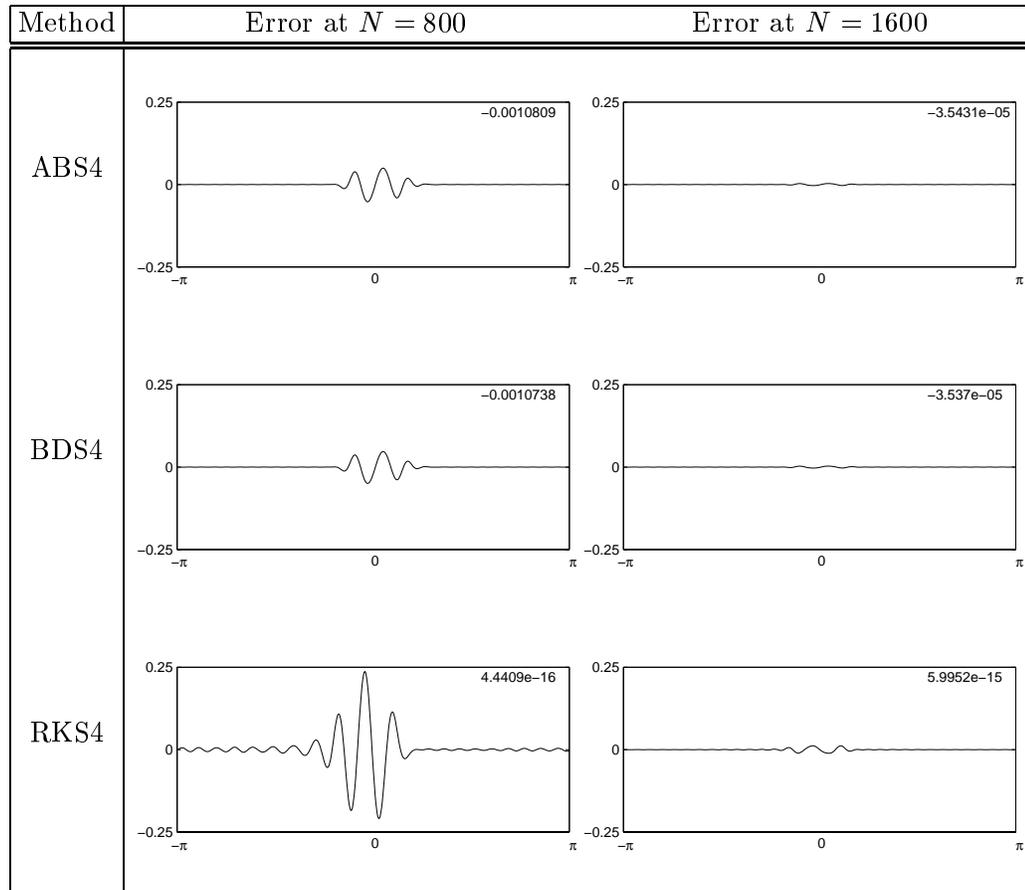


Table 3.7: Error given by running seventh order methods using the root portrait technique. AB7 is not stable until $N > 10384$. Note that vertical scales differ by 100. The relative loss $\left(\frac{\|u_{final}\|_2}{\|u_{initial}\|_2} - 1\right)$ is shown in the upper right-hand corner of the error plots. N is the number of function evaluations used to reach the final time $T = 6\pi$.

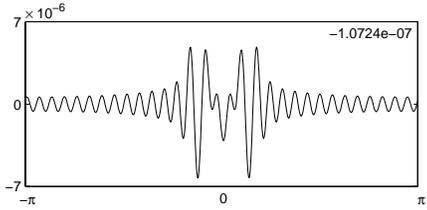
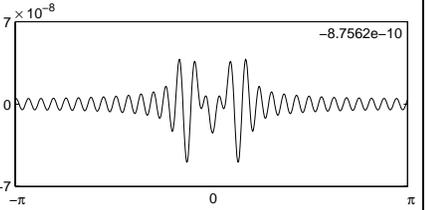
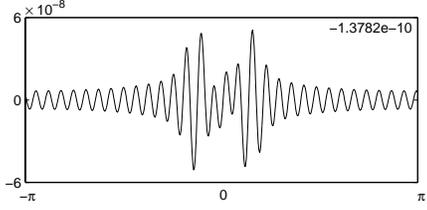
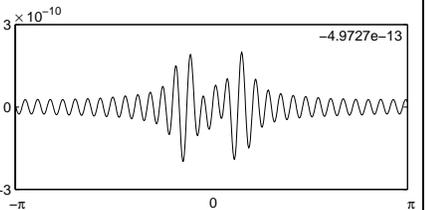
Method	Error at $N = 2000$	Error at $N = 4000$
AB7	Unstable	Unstable
ABS7		

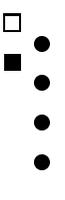
Table 3.8: Error given by running eighth order methods using the root portrait technique. AB8 is not stable until $N > 20455$. Note that vertical scales differ by 2000. The relative loss $\left(\frac{\|u_{final}\|_2}{\|u_{initial}\|_2} - 1\right)$ is shown in the upper right-hand corner of the error plots.

Method	Error at $N = 3000$	Error at $N = 6000$
AB8	Unstable	Unstable
ABS8		

3.11 Conclusions

We have introduced staggered time integrators for solving systems of wave equations. We find that the staggered versions of Adams-Bashforth and backwards differentiation methods have significantly smaller local truncation errors and greater ISBs than their nonstaggered counterparts. In addition, staggered schemes are no more difficult to implement than nonstaggered schemes. We have also considered free parameter multistep methods that allow for additional improvement in the ISB. Staggered Runge-Kutta methods also show promise for treating hyperbolic systems. We have introduced a low-storage fourth order method that has twice the ISB and a much smaller error constant than the classical fourth order Runge-Kutta method and a third order method with 80% larger ISB and 43% small error constant than RK3. Table 3.9 summarizes our results concerning staggered fourth order methods and compares them to some explicit nonstaggered fourth order methods. Experimental results verify the feasibility of these new methods. In addition, we have presented several theoretical considerations concerning staggered time integrators.

Table 3.9: Comparison of fourth order time integrators: nonstaggered vs. staggered. The normalized local truncation error is $Ck^5 f^{(5)}(\eta)$, where C is the normalized error constant.

Nonstaggered				Staggered			
Name	Stencil	Normalized ISB	Normalized error constant	Name	Stencil	Normalized ISB	Normalized error constant
(BD4 implicit)				BDS4		1.000	≈ 0.0370
AB4		≈ 0.430	≈ 0.3486	ABS4		≈ 1.333	≈ 0.0387
RK4		≈ 0.707	≈ 2.1333	RKS4		≈ 1.425	≈ 0.1333

Chapter 4

Conclusions

In this dissertation, we have investigated high-order finite difference methods and staggered time integrators for linear wave equations. A composite method combines a highly accurate (e.g. block pseudospectral) method on boundaries and near interfaces with a low computational cost method used on a background grid on the rest of the computational domain. We envision the methods discussed in this paper being used for such a background grid.

We find that combining

- high orders of accuracy,
- implicitness, and
- staggering

leads to a class of computationally very cost-effective finite difference schemes for equispaced Cartesian grids. As their orders of accuracy increase, these schemes approach the well-known spectral accuracy of periodic, explicit pseudospectral schemes. When we combine this with the idea of overlapping subdomains (as in a composite method), the relatively narrow stencil widths make the schemes well-suited for computations in media with curvilinear material interfaces. The schemes can be applied to most linear wave-type PDEs of broad interest. In the particular application of time-domain computational electromagnetics (also known as FDTD), the classical

Yee scheme uses only the last of the three highlighted concepts. We find that major improvements in accuracy and efficiency can be achieved by also incorporating implicitness and high orders of accuracy.

We have also introduced staggered time integrators for solving systems of linear wave equations. We find that the staggered versions of Adams-Bashforth and backwards differentiation methods have significantly smaller local truncation errors and greater imaginary stability boundaries than their nonstaggered counterparts. In addition, staggered schemes are no more difficult to implement than nonstaggered schemes. We have also considered free parameter multistep methods that allow for additional improvement in the imaginary stability boundary. Staggered Runge-Kutta methods also show promise for treating hyperbolic systems. We have introduced a low-storage fourth order method that has twice the imaginary stability boundary and a much smaller error constant than the classical fourth order Runge-Kutta method and a third order method with 80% larger imaginary stability boundary and 43% smaller error constant than RK3. In addition, we have presented several theoretical considerations concerning staggered time integrators, including a generalization of Dahlquist's First Stability Barrier to staggered schemes.

Bibliography

- [1] Y. ADAM, Highly accurate compact implicit methods and boundary conditions, J. Comput. Phys., 24 (1977), pp. 10-22.
- [2] K. ATKINSON, An Introduction to Numerical Analysis, John Wiley, New York, 1989.
- [3] R. BURDEN and J. FAIRES, Numerical Analysis, 4th ed., PWS-KENT, Boston, 1989.
- [4] L. COLLATZ, The Numerical Treatment of Differential Equations, Springer-Verlag, Berlin, 1960.
- [5] G. DAHLQUIST, Convergence and stability in the numerical integration of ordinary differential equations, Math. Scand., 4 (1956), pp. 33-53.
- [6] T. A. DRISCOLL and B. FORNBERG, A block pseudospectral method for Maxwell's equations: I. One-dimensional, discontinuous-coefficients case, J. Comput. Phys., 140 (1998), pp. 1-19.
- [7] T. A. DRISCOLL and B. FORNBERG, Block pseudospectral methods for Maxwell's equations: II. Two-dimensional, discontinuous-coefficients case, to appear in SIAM J. Sci. Comput.
- [8] B. FORNBERG, On a Fourier method for the integration of hyperbolic equations, SIAM J. Numer. Anal., 12 (1975), pp. 509-528.
- [9] B. FORNBERG, High-order finite differences and the pseudospectral method on staggered grids, SIAM J. Num. Anal., 27 (1990), pp. 904-918.
- [10] B. FORNBERG, A Practical Guide to Pseudospectral Methods, Cambridge University Press, Cambridge, UK, 1996.
- [11] B. FORNBERG, Calculation of weights in finite difference formulas, SIAM Review, 40, (1998), pp. 685-691.
- [12] I.S. GRADSHTEYN and I.M. RYZHIK, (Alan Jeffrey, ed.) Table of Integrals, Series, and Products, 5th ed., Academic Press, Inc., San Diego, CA, 1994.

- [13] E. HAIRER, S.P. NØRSETT, and G. WANNER, Solving Ordinary Differential Equations I, Springer-Verlag, Berlin, 1991.
- [14] R. S. HIRSCH, Higher-order accurate difference solutions of fluid mechanics problems by a compact differencing technique, J. Comput. Phys., 19 (1975), pp. 90-109.
- [15] R. S. HIRSCH, High-order approximations in fluid mechanics, VKI Lecture Series 1983-04, Von Karman Inst. for Fluid Dyn., Brussels, 1983.
- [16] O. HOLBERG, Computational aspects of the choice of operator and sampling interval for numerical differentiation in large-scale simulation of wave phenomena, Geophys. Prospecting, 35 (1987), pp. 629-655.
- [17] A. ISERLES, A First Course in the Numerical Analysis of Differential Equations, Cambridge University Press, Cambridge, 1996.
- [18] R. JELTSCH and O. NEVANLINNA, Stability of explicit time discretizations for solving initial value problems, Numer. Math., 37 (1981), pp. 61-91.
- [19] R. JELTSCH and O. NEVANLINNA, Dahlquist's first barrier for multistage multistep formulas, BIT, 24 (1984), pp. 538-555.
- [20] M. KINDELAN, A. KAMEL, AND P. SGUAZZERO, On the construction and efficiency of staggered numerical differentiators for the wave equation, Geophysics, 55 (1990), pp. 107-110.
- [21] Z. KOPAL, Numerical Analysis (2nd Ed.), Chapman and Hall, London, 1961.
- [22] K. S. KUNZ AND R. J. LUBBERS, The Finite Difference Time Domain for Electromagnetics, CRC Press, Boca Raton, FL, 1993.
- [23] S.K. LELE, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys., 103 (1992), pp. 16-42.
- [24] R. MITTET, O. HOLBERG, B. ARNSTEN, AND L. AMUNDSEN, Fast finite difference modeling of the 3-D elastic wave equation, Society of Exploration Geophysics Expanded Abstracts, 1 (1988), pp. 1308-1311.
- [25] S. A. ORSZAG AND M. ISRAELI, Numerical simulation of viscous incompressible flows, Ann. Rev. Fluid Mech., 6 (1974), pp. 281-318.
- [26] J. STOER and R. BULIRSCH, Introduction to Numerical Analysis, Springer-Verlag, New York, 1980.
- [27] A. TAFLOVE, Computational Electrodynamics: The Finite-Difference Time-Domain Method, Artech House, Boston, 1995.
- [28] R. VICHNEVETSKY AND J. B. BOWLES, Fourier Analysis of Numerical Approximations of Hyperbolic Equations, Studies in Applied Mathematics 5, SIAM, Philadelphia, PA, 1982.

- [29] K.S. YEE, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, IEEE Trans. Antennas and Propagation, 14 (1966), pp. 302-307.

APPENDICES

- A. Explanation of the Padé algorithm given in Section 2.3
- B. Derivation of the limit expressions in Table 2.8
- C. Conversions between Kopal's FD coefficients and those given in Chapter 2
- D. Staggered differentiation matrix
- E. Derivation of equation (2.7)
- F. Defining the local truncation error
- G. Nonstaggered free parameter methods
- H. Proof of the result found in Section 3.6.1 concerning ISBs of AB, ABS, and AM methods
- I. Proof of the staggered analogue of Dahlquist's first barrier

The author made significant contributions to Appendices C, D, E, F, H, and I. Some contributions were made to Appendix G. Appendices A and B were formulated by Bengt Fornberg.

Appendix A

Explanation of the Padé algorithm given in Section 2.3

In this appendix, we explain why the Padé algorithm works in the special case given in Section 2.3. We follow the argument in [11], which can easily be generalized to other values of m , s , d and n .

We search for coefficients b_i and c_i so that

$$\begin{aligned} & b_0 f'(x-h) + b_1 f'(x) + b_2 f'(x+h) \\ & \approx c_0 f\left(x - \frac{3}{2}h\right) + c_1 f\left(x - \frac{1}{2}h\right) + c_2 f\left(x + \frac{1}{2}h\right) + c_3 f\left(x + \frac{3}{2}h\right) \end{aligned} \quad (\text{A.1})$$

becomes exact for as high degree polynomials $f(x)$ as possible. Substituting $f(x) = e^{i\omega x}$ into (A.1) gives

$$\begin{aligned} & i\omega \left[b_0 e^{-i\omega h} + b_1 + b_2 e^{i\omega h} \right] e^{i\omega x} \\ & \approx \left[c_0 e^{-\frac{3}{2}i\omega h} + c_1 e^{-\frac{1}{2}i\omega h} + c_2 e^{\frac{1}{2}i\omega h} + c_3 e^{\frac{3}{2}i\omega h} \right] e^{i\omega x} \end{aligned} \quad (\text{A.2})$$

with the new goal being to make the relation as accurate as possible if locally expanded around $\omega = 0$ (cf. [28], pp. 24-26). After cancelling the factor $e^{i\omega x}$ and substituting $e^{i\omega h} = \xi$, we get

$$\xi^{\frac{1}{2}} \frac{\ln \xi}{h} \approx \frac{c_0 + c_1 \xi + c_2 \xi^2 + c_3 \xi^3}{b_0 + b_1 \xi + b_2 \xi^2} \quad (\text{A.3})$$

This needs to be as accurate as possible (meaning as high order as possible) around

$\xi = 1$. Padé expansion of the left-hand side around $\xi = 1$ to order [3,2] produces the desired coefficients:

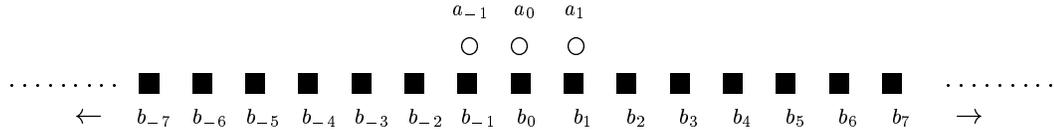
$$\begin{aligned} \xi^{\frac{1}{2}} \frac{\ln \xi}{h} &\approx \frac{1}{h} \frac{(\xi - 1) + (\xi - 1)^2 + \frac{17}{240}(\xi - 1)^3}{1 + (\xi - 1) + \frac{9}{80}(\xi - 1)^2} \\ &= \frac{\left(-\frac{17}{240} - \frac{63}{80}\xi + \frac{63}{80}\xi^2 + \frac{17}{240}\xi^3\right) \frac{1}{h}}{\left(\frac{9}{80} + \frac{31}{40}\xi + \frac{9}{80}\xi^2\right)} \end{aligned} \quad (\text{A.4})$$

The notation above in the description of the weights algorithm was chosen to agree with [11]. It is noted there that the explicit case ($\mathbf{d} = 0$) can be handled even more easily by substituting a Taylor expansion for the Padé expansion used here (since the denominator in (A.3) is then one). Note that this argument can easily be generalized to other values of \mathbf{m} , \mathbf{s} , \mathbf{n} , and \mathbf{d} .

Appendix B

Derivation of the limit expressions in Table 2.8

We offer a brief argument that leads to the limit expressions in Table 2.8. One case is sufficient to illustrate the general argument. We consider the following tridiagonal (3-diagonal) regular grid case:



As their widths n increase, the finite-sized stencils are exact for polynomials of increasingly higher orders. The infinite-width stencil will then also be exact for all trigonometric modes $\sin(\omega x)$, which have derivative $\omega \cos(\omega x)$. When the infinite-width stencil is applied to a particular mode at $x = 0$ with step size $h = 1$, this gives

$$\begin{aligned} & \omega [a_{-1} \cos(-\omega) + a_0 \cos(0\omega) + a_1 \cos(\omega)] \\ & = 2 [b_1 \sin(\omega) + b_2 \sin(2\omega) + b_3 \sin(3\omega) + \dots] \end{aligned} \tag{B.1}$$

which holds for all $\omega \in \mathbf{R}$. (Note that we have used that $b_{-j} = -b_j$.) We next make the very reasonable assumptions that

- (1) the desired limit represents the most accurate derivative approximation available of the desired form, and

- (2) the derivative, being a local property of a function, is best approximated when the coefficients b_k decay to zero as fast as possible (subject to (B.1) holding true).

The left-hand side of (B.1) is a product of the step function ω (which jumps at $\omega = \pm\pi$ when periodically extended) and a periodic function $a_0 + 2a_1 \cos \omega$ (since by symmetry $a_{-1} = a_1$). The decay rate of the Fourier coefficients of this product is maximal if the LHS is as smooth as possible at $\omega = \pm\pi$, i.e. when $(a_0 + 2a_1 \cos \omega)$ has a zero there of as high degree as possible. This occurs when $a_0 = 2a_1$, which when combined with the restriction that $\sum_j a_j = 1$, gives $a_0 = \frac{1}{2}$ and $a_1 = \frac{1}{4}$. With these values,

$$a_0 + 2a_1 \cos \omega = \left(\cos \frac{\omega}{2} \right)^2. \quad (\text{B.2})$$

By integrating equations (B.2) and (B.1) against $\cos(\omega k)$ and $\sin(\omega k)$ respectively (and substituting (B.2) into (B.1)), we thus (in the tridiagonal case $m = 1$) obtain the infinite-order coefficients from (B.1) as

$$\begin{aligned} a_k &= \frac{1}{\pi} \int_0^\pi \cos(\omega k) \left(\cos \frac{\omega}{2} \right)^2 d\omega \\ b_k &= \frac{1}{\pi} \int_0^\pi \omega \sin(\omega k) \left(\cos \frac{\omega}{2} \right)^2 d\omega \end{aligned} \quad (\text{B.3})$$

The case of general m follows completely analogously and leads to the expressions listed in Table 2.8.

Appendix C

Conversions between Kopal's coefficients and those in Chapter 2

In [21], Kopal gives tables of coefficients in terms of the difference operators δ and Δ where

$$\begin{aligned}\delta(y_n) &= y_{n+1/2} - y_{n-1/2} \\ \Delta(y_n) &= y_n - y_{n-1}.\end{aligned}\tag{C.1}$$

In this appendix, we help make clear the connection between the coefficients given in Chapter 2 and those listed in Appendix II of Kopal.

One can use Kopal's Tables 2.1 and 2.2 to find the FD coefficients for nonstaggered approximations of first and second derivatives, respectively. Tables 2.4 and 2.5 of Kopal can be used to find the coefficients of staggered approximations of first and second derivatives, respectively.

For explicit schemes in Kopal, one should consider $\frac{N_j^{(0)}}{D_0^{(j)}}$. For 3-diagonal schemes, one wants $\frac{N_j^{(1)}}{D_1^{(j)}}$. For 5-diagonal schemes, one should consider $\frac{N_j^{(2)}}{D_2^{(j)}}$. We offer one example, the sixth order staggered 3-diagonal approximation to the first

derivative. Using Table 2.4 of Kopal and $j = 1$, we find that

$$\begin{aligned}
D_1^{(1)} f'_k &\approx \delta N_1^{(1)} f_k \\
\left[1 + \frac{9}{5} \left(\frac{\delta}{4} \right)^2 \right] f'_k &\approx \delta \left[1 + \frac{17}{240} \delta^2 \right] f_k \\
f'_k + \frac{9}{80} (f'_{k-1} - 2f'_k + f'_{k+1}) &\approx \delta \left[f_k + \frac{17}{240} (f_{k-1} - 2f_k + f_{k+1}) \right] \\
\frac{9}{80} f'_{k-1} + \frac{31}{40} f'_k + \frac{9}{80} f'_{k+1} &\approx \left[-f_{k-1/2} + f_{k+1/2} \right. \\
&\quad \left. + \frac{17}{240} (-f_{k-3/2} + 3f_{k-1/2} - 3f_{k+1/2} + f_{k+3/2}) \right] \\
\frac{9}{80} f'_{k-1} + \frac{31}{40} f'_k + \frac{9}{80} f'_{k+1} &\approx -\frac{17}{240} f_{k-3/2} - \frac{63}{80} f_{k-1/2} + \frac{63}{80} f_{k+1/2} + \frac{17}{240} f_{k+3/2}
\end{aligned} \tag{C.2}$$

Note that this matches the $n = 2$ case in Table 2.7. Other coefficients can be derived in the same manner.

Although not discussed here, one can use other combinations of N 's and D 's from Kopal's tables than those discussed above to find coefficients for noncentered schemes.

Appendix D

Staggered differentiation matrix

A differentiation matrix is a matrix which, when multiplied by a vector of function values at discrete gridpoints, gives (pseudospectral) derivative values at a set of discrete equispaced locations.

An example is given in [10]. If one assumes that the data is periodic with period 2, it is possible to find a matrix \bar{D} such that

$$\begin{bmatrix} v'(x_1) \\ \vdots \\ v'(x_j) \\ \vdots \\ v'(x_N) \end{bmatrix} = \begin{bmatrix} & & & & \\ & & & & \\ & & \bar{D} & & \\ & & & & \\ & & & & \end{bmatrix} \begin{bmatrix} v(x_1) \\ \vdots \\ v(x_j) \\ \vdots \\ v(x_N) \end{bmatrix} \quad (\text{D.1})$$

This (periodic) nonstaggered pseudospectral differentiation matrix \bar{D} is cyclic and is given by

$$\bar{D}_{i,j} = \begin{cases} \frac{\pi(-1)^{i-j}}{2 \sin[\pi(i-j)/N]} & i \neq j \\ 0 & i = j \end{cases} \quad (\text{D.2})$$

Similarly, one can find the staggered pseudospectral differentiation matrix

D (assuming the data has period 2) that satisfies

$$\begin{bmatrix} v'(x_{1/2}) \\ \vdots \\ v'(x_{j-1/2}) \\ \vdots \\ v'(x_{N-1/2}) \end{bmatrix} = \begin{bmatrix} & & & & \\ & & & & \\ & & D & & \\ & & & & \\ & & & & \end{bmatrix} \begin{bmatrix} v(x_1) \\ \vdots \\ v(x_j) \\ \vdots \\ v(x_N) \end{bmatrix} \quad (\text{D.3})$$

This matrix D is cyclic and given by

$$D_{i,j} = \frac{\pi(-1)^{i-j-1} \cot \left[\left(i - j - \frac{1}{2} \right) \frac{\pi}{N} \right]}{2N \sin \left[\left(i - j - \frac{1}{2} \right) \frac{\pi}{N} \right]}. \quad (\text{D.4})$$

If one instead desired the differentiation matrix for

$\left[v'(x_{3/2}) \dots v'(x_{j+1/2}) \dots v'(x_{N+1/2}) \right]$, this would be given by

$$D_{i,j} = \frac{\pi(-1)^{i-j} \cot \left[\left(i - j + \frac{1}{2} \right) \frac{\pi}{N} \right]}{2N \sin \left[\left(i - j + \frac{1}{2} \right) \frac{\pi}{N} \right]}. \quad (\text{D.5})$$

One can prove this by following the derivation for \bar{D} given in [10], as follows.

We begin with the limiting (infinite width) staggered FD stencil from equation (2.5) (with $h = 2/N$)

$$b_{\infty,k} = \frac{N(-1)^{(k-\frac{1}{2})}}{\pi k^2} \quad (\text{D.6})$$

Assuming we have data with period 2, we can add together period-wide sections of the stencil to create an equivalent stencil that covers only one period of the data.

These weights are given by

$$\begin{aligned}
d_{\infty,k} &= \frac{N(-1)^{k-1/2}}{2\pi} \sum_{j=-\infty}^{\infty} \frac{(-1)^j}{\pi(k+jN)^2} \\
&= \frac{(-1)^{k-1/2}}{2\pi N} \sum_{j=-\infty}^{\infty} \frac{(-1)^j}{(j+k/N)^2} \\
&= \frac{\pi(-1)^{k-1/2}}{2N} \frac{\cot(k\pi/N)}{\sin(k\pi/N)}
\end{aligned} \tag{D.7}$$

We know that this differentiation matrix is cyclic with (i, j) th element $D_{i,j}$ given by $d_{\infty, i-j-1/2}$ for the first case given above and given by $d_{\infty, i-j+1/2}$ for the second case given above.

We thus find that for the case given in (D.3),

$$D_{i,j} = \frac{\pi(-1)^{i-j-1}}{2N} \frac{\cot\left[\left(i-j-\frac{1}{2}\right)\frac{\pi}{N}\right]}{\sin\left[\left(i-j-1/2\right)\frac{\pi}{N}\right]}. \tag{D.8}$$

The second case simply changes the $-\frac{1}{2}$ to $\frac{1}{2}$. For data with period other than 2, one must scale appropriately.

Appendix E

Derivation of equation (2.7)

This derivation was done by the author with the help of Professor Ben Herbst.

E.1 Derivation of the expression for d_k

Our goal is to solve $Ax = b$, where A is an infinite banded symmetric Toeplitz matrix and b is a vector containing the coefficients $b_{n,k}^m$, where m and n are fixed. Equivalently, we wish to solve $a \circ x = b$, where

$$a = \left[\cdots 0 \ a_{-m} \ a_{-m+1} \ \cdots \ a_{-1} \ a_0 \ a_1 \ \cdots \ a_{m-1} \ a_m \ 0 \ \cdots \right],$$

$$x = \{x_j\}_{j=-\infty}^{\infty},$$

and

$$b = e_p \text{ (the vector consisting of all zeros except for a one in the } p^{\text{th}} \text{ position).}$$

We take the discrete Fourier transforms of a, x , and b and use the fact that the original matrix is banded and symmetric to obtain

$$\begin{aligned}
\hat{a}(\theta) &= \sum_{j=-\infty}^{\infty} a_j e^{ij\theta} \\
&= \sum_{j=-m}^m a_j e^{ij\theta} \\
&= a_0 + 2 \sum_{j=1}^m a_j \cos(jx), \\
\hat{x}(\theta) &= \sum_{j=-\infty}^{\infty} x_j e^{ij\theta}, \\
&\text{and} \\
\hat{b}(\theta) &= \sum_{j=-\infty}^{\infty} b_j e^{ij\theta} \\
&= e^{ip\theta}.
\end{aligned} \tag{E.1}$$

Since $a \circ x = b$, from the Fourier convolution theorem we have $\hat{a}(\theta)\hat{x}(\theta) = \hat{b}(\theta)$, so that $\hat{x}(\theta) = \hat{b}(\theta)/\hat{a}(\theta)$. Taking the inverse Fourier transform to solve for x gives

$$\begin{aligned}
x_j &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{x}(\theta) e^{-ij\theta} d\theta \\
&= \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\hat{b}(\theta)}{\hat{A}(\theta)} e^{-ij\theta} d\theta \\
&= \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{e^{i(p-j)\theta}}{a_0 + 2 \sum_{q=1}^m a_q \cos(qx)} d\theta \\
&= \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\cos((p-j)x) + i \sin((p-j)x)}{a_0 + 2 \sum_{q=1}^m a_q \cos(qx)} d\theta
\end{aligned} \tag{E.2}$$

We then recognize that because we only want to measure position relative to the center position, the relevant index variable is $k \equiv p - j$. In addition, because the denominator of the integrand is even, our integrand is the sum of an even function and an odd function; after integration, only the first part remains. We thus conclude that the inverse matrix of a symmetric banded Toeplitz matrix is also symmetric and Toeplitz with entries d_k along diagonal k :

$$d_k = \frac{1}{\pi} \int_0^{\pi} \frac{\cos(kx)}{a_{n,0}^m + 2 \sum_{j=1}^m a_{n,j}^m \cos(jx)} dx. \tag{E.3}$$

E.2 Verification of the expression for d_k : 5-diagonal case

Although we verify the d_k formula only for the 5-diagonal case ($m = 2$), one can easily generalize the argument given for any case.

A is a Toeplitz matrix of bandwidth 5 and A^{-1} is the Toeplitz matrix with entries d_k along diagonal k . Because $A^{-1}A = I$, we claim that

$$d_{k-2}a_{-2} + d_{k-1}a_{-1} + d_k a_0 + d_{k+1}a_1 + d_{k+2}a_2 = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}. \quad (\text{E.4})$$

where $a_i = a_{n,i}^2$. We substitute d_k from (E.3), and the left-hand side of (E.4) becomes

$$\begin{aligned} & \frac{1}{\pi} \int_0^\pi \frac{a_{-2} \cos(k-2)x + a_{-1} \cos(k-1)x + a_0 \cos kx + a_1 \cos(k+1)x + a_2 \cos(k+2)x}{(a_0 + 2a_1 \cos x + 2a_2 \cos 2x)} dx \\ &= \frac{1}{\pi} \int_0^\pi \frac{a_0 \cos kx + 2a_1 \cos kx \cos x + 2a_2 \cos kx \cos 2x}{a_0 + 2a_{n,1} \cos x + 2a_2 \cos 2x} dx \\ &= \frac{1}{\pi} \int_0^\pi \cos kx dx \\ &= \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}. \end{aligned} \quad (\text{E.5})$$

where we have used $a_{-i} = a_i$ and trigonometric identities.

Appendix F

Defining the local truncation error of time integrators

(The work in the section was done by Toby Driscoll and the author.)

Traditionally, local truncation error for a linear multistep numerical time integrator has been defined in terms of Taylor expansions. Although this definition gives the correct order of accuracy for a scheme, often it does not give the correct coefficient multiplying the error term. The proper definition of local truncation error is defined in terms of Padé expansions. We give examples of schemes for which the definitions coincide and schemes for which they differ.

F.1 Discussion

In general, there are three different ways to define the local truncation (discretization) error of a p^{th} order multistep scheme:

- (1) Taylor expansion of the stencil about $k = 0$ [2, 3, 26] :

$$\rho(Z)y(t) - k\sigma(Z)y'(t) = C_1 k^{p+1} y^{(p+1)}(\xi) + O(k^{p+2})$$

- (2) Taylor expansion about $z = 1$ [17] (although see the footnote on p.76 referring to definition 3) :

$$\rho(z) - \sigma(z) \ln z = C_2 (z - 1)^{p+1} + O((z - 1)^{p+2})$$

(3) Padé expansions about $z = 1$ [13] :

$$\frac{\rho(z)}{\sigma(z)} - \ln(z) = C_3(z-1)^{p+1} + O((z-1)^{p+2})$$

where Z is the forward shift operator.

Definitions 1 and 2 are equivalent (because the differentiation operator $D = \frac{1}{k} \ln Z$), but definition 3 leads to different leading error coefficients for many methods because

$$C_1 = C_2 = \sigma(1)C_3$$

For standard Adams-Bashforth and Adams-Moulton schemes, $\sigma(1) = 1$, and there is no difference between C_1 and C_3 . However, for backwards differentiation and other schemes, C_1 and C_3 differ. Experimental results giving the global error strongly indicate that definition 3 is the appropriate definition. One can also come to this conclusion by examining a special case. Consider a multistep method that depends only on previous function values and not on derivative values. For this case, $\sigma(z) = 0$ for all z , and the differential equation plays no role, so the solution cannot be convergent. Definitions 1 and 2 could nevertheless indicate a high degree of accuracy if the difference scheme is an extrapolation method. On the other hand, definition 3 would correctly reflect an infinite error (since $\sigma(1) = 0$ for this case).

More generally, $\rho(1) = 0$ and $\rho'(1) = \sigma(1)$ for any consistent method. If $\sigma(1)$ is small, then 1 is nearly a double root of the difference equation and the method is nearly unstable.

We thus conclude that the correct definition of local truncation error should be based on Padé expansions (definition 3) and not on Taylor expansion (definitions 1 and 2).

F.2 Examples

F.2.1 Adams-type methods

All Adams-Bashforth and Adams-Moulton methods are consistent and have $\rho(z) = z^m - z^{m-1}$. Then, $\sigma(z = 1) = \rho'(z = 1) = m - (m - 1) = 1$. Since $\sigma(1) = 1$, Taylor expansions and Padé expansions are equivalent for these methods.

One example is the fourth order Adams-Bashforth scheme, which has

$$\rho(z) = z^4 - z^3, \quad \sigma(z) = \frac{55}{24}z^3 - \frac{59}{24}z^2 + \frac{37}{24}z - \frac{3}{8}. \quad (\text{F.1})$$

The error constant is $C = \frac{251}{720}$.

The third order Adams-Moulton scheme has

$$\rho(z) = z^2 - z, \quad \sigma(z) = \frac{5}{12}z^2 + \frac{2}{3}z - \frac{1}{12}. \quad (\text{F.2})$$

The error constant is $C = -\frac{1}{24}$.

F.2.2 Backwards differentiation methods

The fourth order backwards differentiation scheme has

$$\rho(z) = z^4 - \frac{48}{25}z^3 + \frac{36}{25}z^2 - \frac{16}{25}z + \frac{3}{25}, \quad \sigma(z) = \frac{12}{25}z^4. \quad (\text{F.3})$$

Using Taylor definitions, the error constant is $-\frac{12}{125}$ whereas the Padé definition gives an error constant of $C = -\frac{1}{5}$.

The third order staggered backwards differentiation scheme has

$$\rho(z) = z^3 - \frac{21}{23}z^2 - \frac{3}{23}z + \frac{1}{23}, \quad \sigma(z) = \frac{24}{23}z^{\frac{5}{2}}. \quad (\text{F.4})$$

The error constant using Taylor definitions is $\frac{1}{23}$, while the correct error constant is $C = \frac{1}{24}$.

F.2.3 Free Parameter Methods

As discussed in Section 3.5, we have developed multistep methods that allow for free parameters due to suboptimization of order. A nonstaggered example of such a method (which is discussed in Section G) is a fourth order scheme with

$$\begin{aligned} \rho(z) &= z^4 + (8 - 24s)z^3 + (24s - 9)z^2 \\ \sigma(z) &= \left(\frac{17}{3} - 9s\right)z^3 + \left(\frac{14}{3} - 19s\right)z^2 + \left(\frac{1}{3} + 5s\right)z - s \end{aligned} \quad (\text{F.5})$$

where s is a free parameter. In order to have a stable method, we must have $\frac{1}{3} \leq s < \frac{5}{12}$. Taylor expansions give the error constant as $\frac{(10+57s)}{90}$, while Padé expansions give $C = \frac{(10+57s)}{180(5-12s)}$, where $\sigma(1) = \rho'(1) = 10 - 24s$. Note that for z near 1, as $s \rightarrow \frac{5}{12}$, our method becomes an extrapolation method, with the Padé error constant correctly reflecting this.

F.3 Conclusions

Local truncation error for a linear multistep method can be defined in terms of Taylor expansions or Padé expansions. Both give the same order of accuracy but often give different error constants. For Adams-type methods, there is no difference between the definitions, but for most other schemes, the definitions differ. In order to obtain a realistic estimate of the global error to be expected from a scheme, one should use a definition of error constant based on Padé expansions.

Appendix G

Nonstaggered free parameter methods

(The work in this appendix was done by the author.)

Free parameter methods are multistep schemes containing “free” parameters due to suboptimization of order. The free parameters may be used to decrease the error constant, increase the ISB, or often both. In addition to studying staggered free parameter methods (discussed in Section 3.5), we have also examined some nonstaggered free parameter methods and discuss one such method here as an illustration of opportunities in this area. We consider methods with stencils of the following form:

$$\begin{array}{c} \square \\ \blacksquare \bullet \\ \blacksquare \bullet \\ \bullet \\ \bullet \end{array} \left[\begin{array}{c|c} 1 & \\ \alpha_3 & \beta_3 \\ \alpha_2 & \beta_2 \\ & \beta_1 \\ & \beta_0 \end{array} \right]$$

While it is possible to find a fifth order method of this form, it is not stable (see Section 3.6.2). We instead search for fourth order methods containing one free parameter. The linear system of equations to be solved has the following solution

with parameter t .

$$\left[\begin{array}{c|c} 1 & \\ \hline 8 + 24t & \frac{17}{3} + 9t \\ -9 - 24t & \frac{14}{3} + 9t \\ & -\frac{1}{3} - 5t \\ & t \end{array} \right] \quad (\text{G.1})$$

This method is stable for $t \in \left(-\frac{5}{12}, -\frac{1}{3}\right]$. We note that for $t = -\frac{3}{8}$, we recover AB4.

This method's error constant is $C = \frac{10-57t}{180(5+12t)}$. As $t \rightarrow -\frac{5}{12}$, the ISB of the method approaches ≈ 0.727 but the error constant becomes unbounded. We thus observe a trade-off between accuracy and stability for this method. This is illustrated in Figure G.1. If one is willing to sacrifice some accuracy to gain in stability, this method would provide the means to do so.

As an example, we give the stability domain of the method that has $t = -0.41$ in Figure G.2. This method has an ISB of ≈ 0.680 , and an error constant of ≈ 2.317 . The stability domain extends to ≈ -0.5092 on the negative real axis.

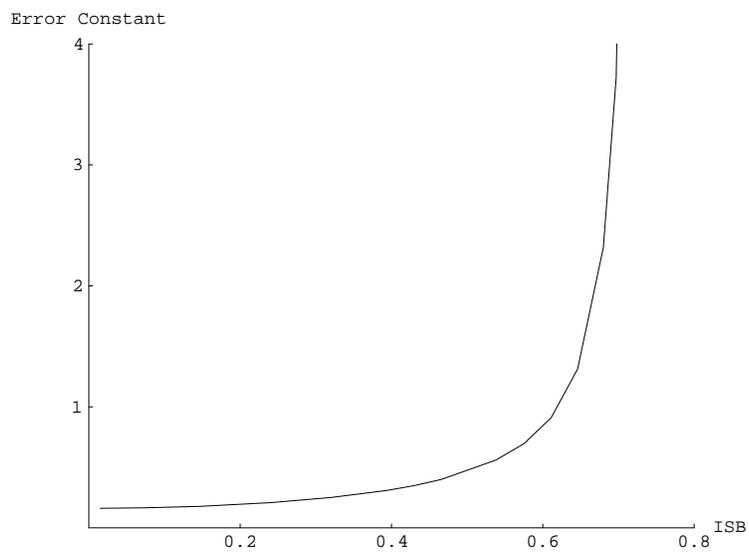


Figure G.1: Trade-off between accuracy (error constant) and stability (ISB) for method (G.1)

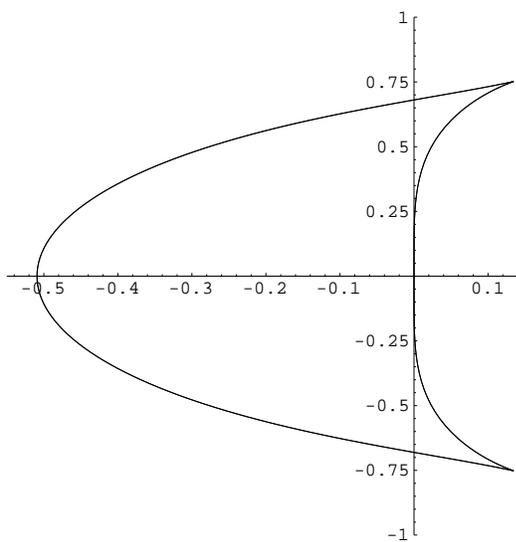


Figure G.2: Stability domain of method (G.1) for $t = -0.41$

Appendix H

Proof of results found in Section 3.6.1 concerning the stability ordinates of AB, ABS, and AM methods

We wish to establish that

Theorem H.1 AB methods have nonzero ISBs only for orders $p = 3, 4, 7, 8, \dots$

Theorem H.2 ABS methods have nonzero ISBs only for orders $p = 2, 3, 4, 7, 8, 11, 12, \dots$

Theorem H.3 AM methods have nonzero ISBs only for orders $p = 1, 2, 5, 6, 9, 10, \dots$

Theorem H.1 was proven by Bengt Fornberg, and the author extended his proof to Theorems H.2 and H.3.

Lemma H.4 (Bengt Fornberg) Given $f(x)$, let $p_n(x)$ be the polynomial of degree n that interpolates $f(x)$ at $x = 0, -k, -2k, \dots, -nk$. Then,

$$\begin{aligned} f(x) - p_n(x) &= x(x+k)\dots(x+nk) \left[\frac{f^{(n+1)}(0)}{(n+1)!} + \frac{f^{(n+2)}(0)}{(n+2)!} \left(x - \frac{n(n+1)}{2}k \right) + \dots \right] \quad (\text{H.1}) \end{aligned}$$

Proof:

- (1) Pick $g(x)$ such that

$$f(x) - p_n(x) = [x(x+k)\dots(x+nk)]g(x). \quad (\text{H.2})$$

By Lagrange's interpolation formula, $p_n(x)$ depends linearly on $f(x)$. Then, $(f(x) - p_n(x))$ must also depend linearly on $f(x)$ and thus by equation (H.2), $g(x)$ must also depend linearly on $f(x)$.

- (2) If $f(x) = x^m$ for $m = 0, \dots, n$, then $g(x) \equiv 0$ because $f(x) = p_n(x)$ for all polynomials of degree less than or equal to n .
- (3) If $f(x) = x^{n+1}$, then $g(x) \equiv 1$.

One can see this because $\frac{x^{n+1} - p_n(x)}{x(x+k)\dots(x+nk)} = g(x)$ must be a constant since the numerator and denominator have the same degree and the same roots. The leading coefficients of the numerator and denominator are both 1, so this constant must be 1.

- (4) If $f(x) = x^{n+2}$, then $g(x) = x - \frac{n(n+1)}{2}k$.

Consider $\frac{x^{n+2} - p_n(x)}{x(x+k)\dots(x+nk)} \equiv g(x)$. This is a linear function with leading coefficient 1 (by the same reasoning as (3)). Thus, $g(x) = x + \beta$. Multiplying, we find

$$x^{n+2} - p_n(x) = (x + \beta)[x(x+k)\dots(x+nk)]. \quad (\text{H.3})$$

Since the left-hand side is missing the x^{n+1} term, the sum of its roots must equal 0. Using this on the right-hand side gives

$$-0 - k - 2k - \dots - nk - \beta = 0. \quad (\text{H.4})$$

Thus, $\beta = -\frac{n(n+1)}{2}k$, giving the desired result.

(5) Let $f(x) = \sum_{j=0}^{\infty} a_j x^j$. We first note that $a_j = \frac{f^{(j)}(0)}{j!}$ by Taylor expansion.

Then, we have

$$\begin{aligned} f(x) - p_n(x) &= \sum_{j=0}^{\infty} a_j x^j - p_n(x) \\ &= [(x)(x+k)\dots(x+nk)]g(x) \end{aligned} \quad (\text{H.5})$$

Thus, by items (3) and (4), we find

$$\begin{aligned} g(x) &= a_{n+1} + a_{n+2} \left(x - \frac{n(n+1)}{2}k \right) + \dots \\ &= \frac{f^{(n+1)}(0)}{(n+1)!} + \frac{f^{(n+2)}(0)}{(n+2)!} \left(x - \frac{n(n+1)}{2}k \right) + \dots \end{aligned} \quad (\text{H.6})$$

Thus, Lemma H.4 is proven.

We proceed with proving Theorems H.1, H.2, and H.3.

When solving the linear problem $\frac{dy}{dt} = \lambda y$ (with $\xi = k\lambda$), the edge of a stability domain is described by the root ξ of the equation

$$\rho(r) - \xi\sigma(r) = 0$$

when r travels around the unit circle ($r = e^{i\theta}$). When considering whether a stability domain can have imaginary axis coverage or not, we wish to describe the behavior of the stability domain boundary near $\xi = 0$.

For an exact method, we would get

$$\xi(\theta) = i\theta$$

because

$$\xi = \frac{\rho(r)}{\sigma(r)} = \ln r = \ln(e^{i\theta}) = i\theta. \quad (\text{H.7})$$

A numerical scheme of order p will instead lead to

$$\xi(\theta) = i\theta + c_p(i\theta)^{p+1} + d_p(i\theta)^{p+2} + \dots \quad (\text{H.8})$$

The sign of the first **real** term of this expansion will dictate whether the stability domain boundary near the origin swings to the right or to the left of the imaginary axis.

H.1 Proof of Theorem H.1: nonstaggered AB methods

For AB methods, we find that $c_p > 0$ and $d_p < 0$ (to be shown below). The pattern for which methods have nonzero ISBs then follows from the powers of the imaginary unit in (H.8). For example, for $p = 3$, then the first real term in the above expansion is given by $c_3(i\theta)^4 = c_3\theta^4$. Because this term is positive, the boundary of the stability domain swings to the right of the imaginary axis and thus we have a nonzero ISB for this method. For $p = 6$, the first real term in the expansion is $d_6(i\theta)^8 = d_6(\theta)^8$. Because this term is negative, the stability domain boundary swings to the left of the imaginary axis and the ISB of this method is zero.

To find the values of c_p and d_p in the case of nonstaggered AB p methods, we note that these schemes, when applied to $y' = \lambda y$ (with $\lambda = \xi/k$), take the form

$$y(t = k) - y(t = 0) = \frac{\xi}{k} \int_0^k (p_n(t)) dt \quad (\text{H.9})$$

where $p_n(t)$ is the interpolating polynomial of y over $t = [-(p-1)k, 0]$.

Now, by Lemma H.4,

$$y(t) - p_n(t) = t(t+k) \dots (t+nk) \left[\frac{y^{(n+1)}(0)}{(n+1)!} + \frac{y^{(n+2)}(0)}{(n+2)!} \left(t - \frac{n(n+1)}{2}k \right) + \dots \right] \quad (\text{H.10})$$

so that

$$\begin{aligned} y(k) - y(0) &= \frac{\xi}{k} \left[\int_0^k y(t) dt + \int_0^k (p_n(t) - y(t)) dt \right] \\ &= -\frac{\xi}{k} \int_0^k t(t+k) \dots (t+nk) \left[\frac{y^{(n+1)}(0)}{(n+1)!} + \frac{y^{(n+2)}(0)}{(n+2)!} \left(t - \frac{n(n+1)}{2}k \right) + \dots \right] dt \\ &\quad + \frac{\xi}{k} \int_0^k y(t) dt. \end{aligned} \quad (\text{H.11})$$

When we substitute $y(t) = e^{i\theta t/k}$ into (H.11) (noting that $y^{(n+1)}(0) = \left(\frac{i\theta}{k}\right)^{n+1}$ and $y^{(n+2)}(0) = \left(\frac{i\theta}{k}\right)^{n+2}$), we find

$$e^{i\theta} - 1 = \frac{\xi}{i\theta} (e^{i\theta} - 1) - \xi \frac{a_n k^{n+2}}{(n+1)!} \left(\frac{i\theta}{k}\right)^{n+1} - \xi \frac{k^{b_n n+3}}{(n+2)!} \left(\frac{i\theta}{k}\right)^{n+2} - \dots \quad (\text{H.12})$$

where

$$\begin{aligned} a_n &= \frac{1}{k^{n+2}} \int_0^k t(t+k) \dots (t+nk) dt \\ &= \int_0^1 s(s+1) \dots (s+n) ds \\ b_n &= \frac{1}{k^{n+3}} \int_0^k t(t+k) \dots (t+nk) \left(t - \frac{n(n+1)}{2}k \right) dt \\ &= \int_0^1 s(s+1) \dots (s+n) \left(s - \frac{n(n+1)}{2} \right) ds \end{aligned} \quad (\text{H.13})$$

We then solve for $\xi(\theta)$ in (H.12) and do an asymptotic expansion about $i\theta = 0$ to find

$$\xi(\theta) \approx i\theta + \frac{a_n}{(n+1)!} (i\theta)^{n+2} + \frac{1}{2(n+2)!} (2b_n - a_n(n+2)) (i\theta)^{n+3} + \dots \quad (\text{H.14})$$

By noting that the order $p = n + 1$, we thus find our coefficients c_p and d_p from equation H.8 to be

$$c_p = \int_0^1 \binom{x+p-1}{p} dx, \quad d_p = - \int_0^1 \binom{x+p-1}{p} \frac{p^2+1-2x}{2(p+1)} dx \quad (\text{H.15})$$

Because both integrands are nonnegative for $p \geq 1$, we thus have that $c_p > 0$ and $d_p < 0$. This establishes our result that nonstaggered Adams-Bashforth methods have nonzero stability ordinate only for orders $p = 3, 4, 7, 8, 11, 12, \dots$ and thus Theorem H.1 is proven.

H.2 Proof of Theorem H.2: ABS methods

For the staggered case, we proceed similarly to the above case, noting that we are now using the polynomial approximation of $y(t)$ on $t \in [nk, 0]$ to step from $t = -\frac{k}{2}$ to $t = \frac{k}{2}$ instead of from $t = 0$ to $t = k$. Thus, equation (H.11) becomes

$$y\left(\frac{k}{2}\right) - y\left(-\frac{k}{2}\right) = \frac{\xi}{k} \left[\int_{-k/2}^{k/2} y(t) dt + \int_{-k/2}^{k/2} [p_n(t) - y(t)] dt \right] \quad (\text{H.16})$$

After using Lemma H.4 and substituting $y(t) = e^{i\theta t/k}$, we have

$$\begin{aligned} & e^{i\theta/2} - e^{-i\theta/2} \\ &= \frac{\xi}{k} \left\{ - \int_{-k/2}^{k/2} t(t+k) \dots (t+nk) \left[\frac{y^{(n+1)}(0)}{(n+1)!} + y^{(n+2)}(0) \left(t - \frac{n(n+1)}{2}k \right) + \dots \right] dt \right. \\ & \quad \left. + \int_{-k/2}^{k/2} e^{i\theta t/k} dt \right\} \end{aligned} \quad (\text{H.17})$$

After solving for ξ and doing an asymptotic expansion about $i\theta = 0$, we find

$$\xi \approx i\theta + \frac{c_n}{(n+1)!}(i\theta)^{n+1} + \frac{d_n}{(n+2)!}(i\theta)^{n+3} \quad (\text{H.18})$$

where

$$\begin{aligned} c_n &= \frac{1}{k^{n+2}} \int_{-k/2}^{k/2} t(t+k) \dots (t+nk) dt \\ d_n &= \frac{1}{k^{n+3}} \int_{-k/2}^{k/2} t(t+k) \dots (t+nk) \left(t - \frac{n(n+1)}{2}k \right) dt \end{aligned} \quad (\text{H.19})$$

After substituting $p = n + 1$ and transforming our integrals, we find our coefficients c_p and d_p from equation (H.8) to be

$$c_p = \int_{-1/2}^{1/2} \binom{x+p-1}{p} dx, \quad d_p = - \int_{-1/2}^{1/2} \binom{x+p-1}{p} \frac{p(p-1) - 2x}{2(p+1)} dx \quad (\text{H.20})$$

Although the integrands for both c_p and d_p are no longer of constant sign, we can again establish that $c_p > 0$ and $d_p < 0$ (for $p > 2$) by induction, as shown below.

Lemma H.5

$$\int_{-1/2}^{1/2} \prod_{s=0}^{p-1} (s+x) dx > 0 \quad \forall p \geq 2 \quad (\text{H.21})$$

Proof by induction:

- ($p = 2$):

$$\int_{-1/2}^{1/2} x(1+x) dx = \frac{1}{12} \quad (\text{H.22})$$

- Assume (H.21) is true for $p = k$, so that

$$\int_{-1/2}^{1/2} \prod_{s=0}^{k-1} (s+x) dx > 0 \quad (\text{H.23})$$

- Show that equation (H.21) is true for $p = k + 1$.

$$\begin{aligned} \int_{-1/2}^{1/2} \prod_{s=0}^k (s+x) dx &= \int_{-1/2}^{1/2} (k+x) \prod_{s=0}^{k-1} (s+x) dx \\ &= k \int_{-1/2}^{1/2} \prod_{s=0}^{k-1} (s+x) dx + \int_{-1/2}^{1/2} x^2 \prod_{s=1}^{k-1} (s+x) dx \end{aligned} \quad (\text{H.24})$$

The first term is positive by (H.23), and the second term is positive because the integrand is positive on $[-\frac{1}{2}, \frac{1}{2}]$, so we have that $c_p > 0$ for $p \geq 2$ by induction.

Lemma H.6

$$\int_{-1/2}^{1/2} (p^2 - p - 2x) \prod_{s=0}^{p-1} (s+x) dx > 0 \quad \forall p \geq 3 \quad (\text{H.25})$$

Proof by induction:

- For $p = 3$, we have

$$\int_{-1/2}^{1/2} (9 - 3 - 2x) \prod_{s=0}^{3-1} (s+x) dx = \frac{137}{120}. \quad (\text{H.26})$$

- Assume that equation (H.25) is true for $p = k$ so that

$$\int_{-1/2}^{1/2} (k^2 - k - 2x) \prod_{s=0}^{k-1} (s+x) dx > 0 \quad (\text{H.27})$$

- Show that (H.25) is true for $p = k + 1$.

$$\begin{aligned}
& \int_{-1/2}^{1/2} [(k+1)^2 - (k+1) - 2x] \prod_{s=0}^k (s+x) dx \\
&= k \int_{-1/2}^{1/2} (k^2 - k - 2x) \prod_{s=0}^{k-1} (s+x) dx \\
&\quad + \int_{-1/2}^{1/2} (2k^2 + xk(k+1) - 2x^2) \prod_{s=0}^{k-1} (s+x) dx \\
&= k \int_{-1/2}^{1/2} (k^2 - k - 2x) \prod_{s=0}^{k-1} (s+x) dx + 2k^2 \int_{-1/2}^{1/2} \prod_{s=0}^{k-1} (s+x) dx \\
&\quad + \int_{-1/2}^{1/2} x^2 (k(k+1) - 2x) \prod_{s=1}^{k-1} (s+x) dx \tag{H.28}
\end{aligned}$$

The first term is positive by (H.27). The second is positive by Lemma H.5, and the third term is positive because the integrand is positive on $[-1/2, 1/2]$ for $k \geq 3$.

Thus, $d_p < 0$ for $p \geq 3$ by induction so that ABS methods have nonzero ISBs only for orders $p = 2, 3, 4, 7, 8, 11, 12, \dots$ (Leapfrog, $p = 2$, is a special case.)

H.3 Proof of Theorem H.3: AM methods

For this case, we wish to step y from $t = -k$ to $t = 0$ using interpolating data at $t = 0, -k, \dots, -(n+1)k$. (Note that this is equivalent to stepping y from $t = 0$ to $t = k$ using interpolating data at $t = k, 0 - k, \dots, -nk$, which is how we usually interpret Adams-Moulton methods.) Following the same procedure as done in the previous two cases (and noting that $p = n + 2$), we find that

$$\xi \approx i\theta + \frac{a_p}{p!} (i\theta)^p + \frac{b_p}{2(p+1)!} (i\theta)^{p+1} + \dots \tag{H.29}$$

where

$$\begin{aligned}
 a_p &= \int_0^1 \prod_{s=-1}^{p-2} (s+x) dx \\
 b_p &= \int_0^1 (2x - (p-1)^2) \prod_{s=-1}^{p-2} (s+x) dx
 \end{aligned} \tag{H.30}$$

Because the integrands are of constant sign on $[0, 1]$, $a_p < 0$ and $b_p > 0$.

Examining the sign of the first real term in (H.29) allows us to conclude that Adams-Moulton methods have nonzero ISBs only for orders $p = 1, 2, 5, 6, 9, 10, \dots$ (Backwards Euler ($p = 1$) and AM2 ($p = 2$) are special cases.)

Appendix I

Proof of the staggered analogue of Dahlquist's first barrier

(The work in this appendix was done by the author.)

Theorem I.1 The order p of an explicit stable m -step staggered method satisfies

$$p \leq \begin{cases} m & , \quad m \text{ an even integer} \\ m + \frac{1}{2} & , \quad m \text{ a half-integer} \\ m + 1 & , \quad m \text{ an odd integer} \end{cases} \quad (\text{I.1})$$

Our proof of this theorem follows those done by Jeltsch and Nevanlinna [19] and Dahlquist [5] for nonstaggered methods.

Lemma I.2 The asymptotic expansion of

$$\frac{z}{\sqrt{1-z^2} \log \frac{1+z}{1-z}} = \sum_{j=0}^{\infty} d_j z^j \quad (\text{I.2})$$

satisfies $d_{2j+1} = 0$ and $d_{2j} > 0$.

Proof:

One can see that $d_0 = \frac{1}{2}$ by considering the limit of the left-hand side as $z \rightarrow 0$. Also, $d_{2j+1} = 0$ because we have an even function. We divide both sides of equation (I.2) by z and then transform $z \rightarrow \frac{1}{w}$. We are then considering the

expansion

$$\frac{w}{\sqrt{w^2-1} \log\left(\frac{1+w}{w-1}\right)} = \frac{w}{2} + \sum_{j=0}^{\infty} \delta_{2j+1} w^{-(2j+1)} \quad (\text{I.3})$$

where $\delta_{2j+1} = d_{2j+2}$. By Cauchy's Integral Formula

$$\begin{aligned} \delta_{2j+1} &= \frac{1}{2\pi i} \oint_C w^{2j} \frac{w}{\sqrt{w^2-1} \log\left(\frac{1+w}{w-1}\right)} dw \\ &= \frac{1}{2\pi i} \oint_C \frac{w^{2j+1} \sqrt{\frac{w+1}{w-1}}}{(w+1) \log\left(\frac{w+1}{w-1}\right)} dw \end{aligned} \quad (\text{I.4})$$

where C is an arbitrary curve enclosing $(-1, 1)$ on the real axis.

By taking our branch cut on $(-1, 1)$ of the real axis, we thus find that

$$\begin{aligned} \delta_{2j+1} &= \frac{1}{2\pi i} \left[\int_{-1}^1 \frac{x^{2j+1} \sqrt{\frac{1+x}{1-x}}(i)}{(x+1) \left(i\pi + \log\left(\frac{1+x}{1-x}\right)\right)} dx - \int_{-1}^1 \frac{x^{2j+1} \sqrt{\frac{1+x}{1-x}}(-i)}{(x+1) \left(-i\pi + \log\left(\frac{1+x}{1-x}\right)\right)} dx \right] \\ &= \frac{1}{\pi} \int_{-1}^1 \frac{x^{2j+1} \log\left(\frac{1+x}{1-x}\right)}{\sqrt{1-x^2} \left(\pi^2 + \log^2\left(\frac{1+x}{1-x}\right)\right)} dx \end{aligned} \quad (\text{I.5})$$

Because the integrand is non-negative on $(-1, 1)$, we thus find that $\delta_{2p+1} > 0$ and thus that $d_{2p} > 0$ in (I.2), thus proving Lemma I.2.

Lemma I.3 In the asymptotic expansion

$$\frac{z \sqrt{\frac{1+z}{1-z}}}{\log\left(\frac{1+z}{1-z}\right)} = \sum_{j=0}^{\infty} \gamma_j z^j \quad (\text{I.6})$$

we have $\gamma_j > 0$.

Proof:

We note that

$$\frac{z\sqrt{\frac{1+z}{1-z}}}{\log\left(\frac{1+z}{1-z}\right)} = (1+z)\frac{z}{\sqrt{1-z^2}\log\left(\frac{1+z}{1-z}\right)} = (1+z)\sum_{j=0}^{\infty} d_{2j}z^{2j} \quad (\text{I.7})$$

Then, by Lemma I.2, $\gamma_{2j} = \gamma_{2j+1} = d_{2j} > 0$, thus proving Lemma I.3.

Lemma I.4 (Germund Dahlquist) If $\rho(z) = \sum_{j=0}^m a_j z^j$ for a stable multistep method, then all coefficients a_j have the same sign.

Proof of this lemma can be found in [5].

We proceed with the proof of Theorem I.1.

I.1 Case 1: m a half-integer

We first consider the case of m a half-integer. For this case, we can represent the generating polynomials as

$$\begin{aligned} \rho(\zeta) &= \zeta^{1/2} \left[\alpha_0 + \alpha_1 \zeta + \dots + \alpha_{m-1/2} \zeta^{m-1/2} \right] \\ \sigma(\zeta) &= \left[\beta_0 + \beta_1 \zeta + \dots + \beta_{m-1/2} \zeta^{m-1/2} \right] \end{aligned} \quad (\text{I.8})$$

We make the Greek-Roman transformation $\zeta = \frac{1+z}{1-z}$ and define the functions

$$\begin{aligned}
r(z) &\equiv \left(\frac{1-z}{2}\right)^{m-1/2} \rho\left(\frac{1+z}{1-z}\right) \\
&= \frac{1}{2^{m-1/2}} \sqrt{\frac{1+z}{1-z}} \left[\alpha_0(1-z)^{m-1/2} + \alpha_1(1-z)^{m-3/2}(1+z) + \right. \\
&\quad \left. \dots + \alpha_{m-1/2}(1+z)^{m-1/2} \right] \\
&\equiv \sqrt{\frac{1+z}{1-z}} \sum_{i=1}^{m-1/2} a_i z^i
\end{aligned} \tag{I.9}$$

and

$$\begin{aligned}
s(z) &\equiv \left(\frac{1-z}{2}\right)^{m-1/2} \sigma\left(\frac{1+z}{1-z}\right) \\
&= \frac{1}{2^{m-1/2}} \left[\beta_0(1-z)^{m-1/2} + \beta_1(1-z)^{m-3/2}(1+z) + \dots + \beta_{m-1/2}(1+z)^{m-1/2} \right] \\
&\equiv \sum_{i=0}^{m-1/2} b_i z^i.
\end{aligned} \tag{I.10}$$

We note that because $\rho(\zeta = 1) = 0$ (consistency), we have $r(z = 0) = 0$. Thus, $a_0 = 0$.

Because we have a stable method, all roots ζ of $\rho(\zeta)$ must be inside the unit disk, with roots on the unit circle simple. So, all roots z of $r(z)$ must lie in the closed left-hand plane, with roots on the imaginary axis simple. Then, by Lemma I.4, all coefficients a_i that are nonzero must have the same sign. Since

$$a_1 = r'(z = 0) = 2^{3/2-m} > 0, \tag{I.11}$$

we have that

$$a_j \geq 0 \quad \forall j \geq 1. \tag{I.12}$$

The condition for a multistep method to be of order p can be written as

$$\frac{\rho(\zeta)}{\log \zeta} - \sigma(\zeta) = c_{p+1}(\zeta - 1)^p + O[(\zeta - 1)^{p+1}] \quad (\text{I.13})$$

(for some $c_{p+1} \neq 0$). This can be rewritten in terms of z , $r(z)$, and $s(z)$ as

$$\frac{r(z)}{z\sqrt{\frac{1+z}{1-z}}} \left[\frac{z\sqrt{\frac{1+z}{1-z}}}{\log\left(\frac{1+z}{1-z}\right)} \right] - s(z) = 2^{p-m+1/2} c_{p+1} z^p + O(z^{p+1}). \quad (\text{I.14})$$

We define

$$\frac{z\sqrt{\frac{1+z}{1-z}}}{\log\left(\frac{1+z}{1-z}\right)} \equiv \sum_{j=0}^{\infty} \gamma_j z^j \quad (\text{I.15})$$

where we know that $\gamma_j > 0 \quad \forall j$ by Lemma I.3.

Thus, the order condition becomes

$$\left(\sum_{i=1}^{m-1/2} a_i z^{i-1} \right) \left(\sum_{j=0}^{\infty} \gamma_j z^j \right) - \sum_{i=0}^{m-1/2} b_i z^i = 2^{p-m+1/2} c_{p+1} z^p + O(z^{p+1}). \quad (\text{I.16})$$

The first term in the expansion of the product of series in (I.1) that cannot possibly be cancelled by a term in the series for $s(z)$ is the $z^{m+1/2}$ term. We let $p = m + 1/2$ and consider the coefficients of the $z^{m+1/2}$ term on both sides of the equation to find

$$\sum_{j=1}^{m-1/2} a_j \gamma_{m+3/2-j} = 2c_{m+3/2}. \quad (\text{I.17})$$

Then, because $\gamma_j > 0$ and $a_j \geq 0$ for all $j > 0$, we have

$$c_{m+3/2} = \frac{1}{2} \sum_{j=1}^{m-1/2} a_j \gamma_{m+3/2-j} \geq \frac{1}{2} a_1 \gamma_{m+1/2} > 0. \quad (\text{I.18})$$

Since $c_{m+3/2} \neq 0$, we find that p cannot equal (or be larger than) $m + \frac{3}{2}$. Thus, the order p of an m -step method, where m is a half-integer, must satisfy $p \leq m + \frac{1}{2}$. We note that ABS p methods have $p = m + \frac{1}{2}$ and thus achieve equality for this case.

I.2 Case 2: m an integer

When m is an integer, we can represent our generating polynomials as

$$\begin{aligned}\rho(\zeta) &= [\alpha_0 + \alpha_1\zeta + \dots + \alpha_m\zeta^m] \\ \sigma(\zeta) &= \zeta^{1/2} [\beta_0 + \beta_1\zeta + \dots + \beta_{m-1}\zeta^{m-1}].\end{aligned}\tag{I.19}$$

After making the transformation $\zeta = \frac{1+z}{1-z}$, we define the functions

$$\begin{aligned}r(z) &\equiv \left(\frac{1-z}{2}\right)^m \rho\left(\frac{1+z}{1-z}\right) \\ &= \frac{1}{2^m} [\alpha_0(1-z)^m + \alpha_1(1-z)^{m-1}(1+z) + \dots + \alpha_m(1+z)^m] \\ &\equiv \sum_{i=1}^m a_i z^i\end{aligned}\tag{I.20}$$

and

$$\begin{aligned}s(z) &\equiv \left(\frac{1-z}{2}\right)^m \sigma\left(\frac{1+z}{1-z}\right) \\ &= \frac{1}{2^m} \sqrt{1-z^2} [\beta_0(1-z)^m + \beta_1(1-z)^{m-1}(1+z) + \dots + \beta_{m-1}(1+z)^{m-1}] \\ &\equiv \sqrt{1-z^2} \sum_{i=0}^{m-1} b_i z^i.\end{aligned}\tag{I.21}$$

Again, we have $a_0 = 0$ for consistency and $a_j \geq 0 \quad \forall j \geq 1$ for stability (noting that $a_1 = 2^{1-m} > 0$) by Lemma I.4.

The order condition (I.13) becomes

$$\frac{r(z)}{z} \left[\frac{z}{\sqrt{1-z^2}} \log \left(\frac{1+z}{1-z} \right) \right] - \frac{s(z)}{\sqrt{1-z^2}} = 2^{p-m} c_{p+1} z^p + O(z^{p+1}) \quad (\text{I.22})$$

which, using (I.2), can be rewritten as

$$\left(\sum_{i=1}^m a_i z^{i-1} \right) \left(\sum_{j=0}^{\infty} d_j z^j \right) - \sum_{i=0}^{m-1} b_i z^i = 2^{p-m} c_{p+1} z^p + O(z^{p+1}). \quad (\text{I.23})$$

We know from Lemma I.2 that $d_{2j} > 0$ and $d_{2j+1} = 0$. Then there are two cases:

(1) m even

If m is even, then $d_m > 0$. The first term in the product of the two series in (I.23) that cannot be cancelled by a term in the series for $s(z)$ is the z^m term. We let $p = m$ and consider the coefficients of the z^m terms, giving

$$c_{m+1} = \sum_{i=1}^m a_i d_{m-i+1} \geq a_1 d_m > 0 \quad (\text{I.24})$$

(using the fact that $a_i \geq 0$). Thus, this term cannot equal 0 and we find that such a method cannot have order $m+1$ (or higher). Thus, we have that $p \leq m$ for m even.

(2) m odd

If m is odd, then $d_m = 0$ but $d_{m+1} > 0$. The first term in the product of the two series in (I.23) that cannot be cancelled by a term in the series for $s(z)$ is the z^{m+1} term. We then let $p = m+1$ and consider the coefficients of the z^{m+1} terms, giving

$$c_{m+2} = \frac{1}{2} \sum_{i=1}^m a_i d_{m-i+2} \geq \frac{1}{2} a_1 d_{m+1} > 0 \quad (\text{I.25})$$

(using the fact that $a_i > 0$. Thus, this term cannot equal 0 and we find that such a method cannot have order $m + 2$ (or higher). Thus, we must have that $p \leq m + 1$ for m odd.

This completes the proof of Theorem I.1.

We note that BDS p methods have $p = m$ and thus achieve equality for m even. For m odd, we list a few examples of methods that achieve equality.

- $m = 1$: leapfrog

$$y_{n+1} = y_n + kf_{n+1/2} \tag{I.26}$$

The stability domain of this second order scheme extends from $-2i$ to $2i$ on the imaginary axis.

- $m = 3$:

$$\begin{aligned} & y_{n+1} + (-27 + 24t)y_n - (-27 + 24t)y_{n-1} - y_{n-2} \\ &= k(tf_{n+1/2} + (-24 + 22t)f_{n-1/2} + tf_{n-3/2}) \end{aligned} \tag{I.27}$$

This fourth order scheme with parameter t is stable for $t \in (1, 7/6)$ with stability domains consisting of only the origin.