

EULERIAN-LAGRANGIAN LOCALIZED ADJOINT
METHOD AND SMOOTHED AGGREGATIONS
ALGEBRAIC MULTIGRID

by

Caroline Heberton

B.A., Grinnell College, 1978

A thesis submitted to the
University of Colorado at Denver
in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Applied Mathematics

2000

This thesis for the Doctor of Philosophy

degree by

Caroline Heberton

has been approved

by

Thomas Russell

Leopoldo Franca

Andrew Knyazev

Jan Mandel

Thomas Manteuffel

Date

Heberton, Caroline (Ph.D., Applied Mathematics)

Eulerian-Lagrangian Localized Adjoint Method and Smoothed Aggregations Algebraic Multigrid

Thesis directed by Professor Thomas Russell

ABSTRACT

A three-dimensional implementation of an Eulerian-Lagrangian Localized Adjoint Method (ELLAM) for the solution of an advection-diffusion equation is described. A system of integral equations is derived from a second order partial differential equation, and the numerical treatment of each term in the integral equation is discussed. Computational results for a variety of test problems are presented. Analysis of a one-dimensional problem is developed. Stability is demonstrated in the purely diffusive and purely advective limits, under assumptions on boundary conditions, and in the case of an infinite spatial domain with no diffusion. The effect of repeated application of the operator on a concentration peak is investigated.

A theoretical result for the convergence of a variant of a smoothed aggregations algebraic multigrid (AMG) method for the solution of a discretized second-order scalar problem with jumps in coefficients is presented. An algorithm is developed with components shown to satisfy abstract convergence criteria. The algorithm uses a coarsening strategy which respects boundaries of

high-coefficient subdomains, and introduces a tentative prolongator and prolongator smoother designed to treat problems with discontinuous coefficients.

This abstract accurately represents the content of the candidate's thesis. I recommend its publication.

Signed _____
Thomas Russell

ACKNOWLEDGMENTS

Portions of Chapter One of this thesis are being published as a part of U.S. Geological Survey Water Resources Report 00-xxxx by C. I. Heberton, T. F. Russell, L. F. Konikow, and G. Z. Hornberger. Funding was also provided by the USGS. Research was also supported in part by National Science Foundation Grant No. DMS-9706866 and Army Research Office Grant No. 37119-GS-AAS. Chapter Two represents joint work with Petr Vaněk and Marian Brezina.

CONTENTS

<u>Figures</u>	ix
<u>Tables</u>	xvi
<u>Chapter</u>	
1. Eulerian-Lagrangian Localized Adjoint Method	1
1.1 Introduction	1
1.2 Three-Dimensional ELLAM	4
1.2.1 Governing Equation for Solute Transport	5
1.3 Formulation of ELLAM equations	8
1.3.1 Cell Integral Equations	9
1.3.2 Outflow Boundary Equations	9
1.3.3 Assumptions	11
1.4 Numerical Methods	11
1.4.1 Mass Tracking	12
1.4.2 Numerical Integration	14
1.4.3 Dispersion	14
1.4.4 Storage at New Time Level	14
1.4.5 Mass Storage at Old Time Level	17
1.4.6 Approximate Test Functions	18
1.4.7 Source Integral	21
1.4.8 Sink Integral	22

1.4.9	Inflow Boundary Integral	22
1.4.10	Outflow Integrals	23
1.4.11	Decay	25
1.4.12	Assumptions	25
1.5	Test Problems	26
1.5.1	One-Dimensional Flow	26
1.5.2	Uniform, Three-Dimensional Flow	29
1.5.3	Two-Dimensional Radial Flow	33
1.5.4	Initial Condition in Uniform Flow	40
1.5.5	Constant Source in Nonuniform Flow	42
1.6	One-Dimensional ELLAM	51
1.6.1	Limiting Case of No Advection	54
1.6.2	Limiting Case of No Diffusion	57
1.6.3	Infinite Spatial Domain	58
1.6.4	Further Considerations for Finite Spatial Domain	63
1.6.5	Numerical Dispersion and Oscillations	65
1.7	Conclusion	77
1.8	Further Research	77
2.	Algebraic Multigrid for Problems with Discontinuous Coefficients	78
2.1	Introduction	78
2.2	Abstract Convergence Theory	82
2.3	Model Problem	86
2.4	Algorithm	87
2.4.1	Stages of Processing	90

2.5	Smoother properties	92
2.6	Tentative Prolongator	99
2.6.1	Assumptions on aggregates.	99
2.6.2	Kinds of aggregates	100
2.7	Single subdomain with high-coefficient	102
2.8	Multiple subdomains with high-coefficients	106
2.9	Computational Experiments	112
2.10	Conclusion	114
	<u>References</u>	116

FIGURES

Figure

1.1	Cell 7 receives advected mass from inflow, source in cell 1, and storage in cell 1.	10
1.2	Outflow boundary receives advected mass from inflow, source in cell 1, and storage in all cells.	11
1.3	Preimage of cell may be irregularly shaped and not easily delimited by backtracking. Instead, the known mass distribution at time t^n is tracked forward along streamlines of the advective flow to time t^{n+1}	14
1.4	Approximate test functions in one direction on a uniform grid, with NS=2.	19
1.5	Approximate test functions in one direction on a uniform grid, with NS=4.	20
1.6	Approximate test functions in one direction on a nonuniform grid, with NS=2.	20

1.7	Plots of concentration as a function of cell node for one-dimensional flow with a constant velocity field and low dispersion. Shown are the analytical solution (lowest graph), ELLAM results using CELDIS = 1 (121 time steps), NSC = 32, NSR = NSL = 2, NT = 128, and ELLAM results using CELDIS = 10.1 (12 time steps), NSC = 4, NSR = NSL = 2, NT = 128 (upper graph). Results for CELDIS = 1 are virtually identical to the analytical.	29
1.8	Plots of concentration as a function of cell node for one-dimensional flow with a constant velocity field and high dispersion. Shown are the analytical solution (lowest graph), ELLAM results using CELDIS = 1 (121 time steps), NSC = 32, NSR = NSL = 2, NT = 128, and ELLAM results using CELDIS = 10.1 (12 time steps), NSC = 4, NSR = NSL = 2, NT = 128 (upper graph). Results using CELDIS = 1 are virtually identical to the analytical.	30
1.9	Concentration vs. cell node plot with CELDIS = 61 (two time steps), NSC = 4, NSR = NSL = 2, NT = 128.	32
1.10	Plots of concentration as a function of cell node for decay constant $\lambda = 0.01 \text{ sec}^{-1}$. Shown are the analytical solution (lower graph) and ELLAM results using CELDIS = 1 (121 time steps), NSC = 32, NSR = NSL = 2, NT = 128.	32

1.11	Concentration contours of analytical solution in the horizontal plane containing the solute source (layer 1) for three-dimensional solute transport in a uniform steady flow problem.	35
1.12	Concentration contours in the horizontal plane containing the solute source (layer 1) for three-dimensional solute transport in a uniform steady flow problem with CELDIS = 7 (two time steps), NSC = NSR = NSL = 4, NT = 16.	35
1.13	Concentration contours in the horizontal plane containing the solute source (layer 1) for three-dimensional solute transport in a uniform steady flow problem with CELDIS = 1 (14 time steps), NSC = NSR = NSL = 4, NT = 4.	36
1.14	Concentration contours in the horizontal plane containing the solute source (layer 1) for three-dimensional solute transport in a uniform steady flow problem with CELDIS = 0.1 (134 time steps), NSC = NSL = 4, NSR = 8, NT = 16.	36
1.15	Contour plot of analytical solution for two dimensional radial flow problem.	37
1.16	Contour plot of concentration for run with two time steps using CELDIS = 75, NSC = NSR = 4, NSL = 2, NT = 16.	38
1.17	Contour plot of concentration for run with 29 time steps using CELDIS = 5, NSC = NSR = 4, NSL = 2, NT = 4.	40
1.18	Contour plot of concentration for run with 563 time steps using CELDIS = 0.25, NSC = NSR = 4, NSL = 2, NT = 4. Concentration maximum is 1.019.	40

1.19	Contour plot of concentration for run with 563 time steps using CELDIS = 5, NSC = NSR = 8, NSL = 2, NT = 4. Concentra- tion maximum is 1.0056.	41
1.20	Contour plot showing log of concentrations in analytical soluion of Dirac problem at $t = 90$. Concentration maximum is 25195. .	44
1.21	Contour plot showing log of concentrations for spike initial con- dition, and CELDIS = 5, NSC = NSR = NSL = 4, NT = 2. Concentration maximum is 15160.	44
1.22	Contour plot showing log of concentrations in analytical soluion of Dirac problem at $t = 130$. Concentration maximum is 14539.	45
1.23	Contour plot showing log of concentrations for dispersed initial condition, and CELDIS = 5, NSC = NSR = NSL = 4, NT = 2. Concentration maximum is 16909.	45
1.24	Contour plot showing log of concentrations of analytical solution to Dirac problem at $t = 130$. Concentration maximum is 8645. .	46
1.25	Contour plot showing log of concentrations for dispersed initial condition, and CELDIS = 5, NSC = NSR = NSL = 4, NT = 2 at $t = 130$. Concentration maximum is 8167.	46
1.26	Contour plot showing concentrations of a two-dimensional finite element solution of the Burnett and Frind problem. Contours shown are 0.1 to 0.9.	47

1.27	Contour plot showing concentrations of ELLAM solution to two-dimensional Burnett and Frind problem using CELDIS = 30, NSC = NSR = NSL = 4, NT = 32. Contours shown are 0.1 to 0.9.	49
1.28	Contour plot showing concentrations of a three-dimensional finite element solution of the Burnett and Frind problem. Contours shown are 0.1 to 0.9.	50
1.29	Contour plot showing concentrations of ELLAM solution to low dispersion Burnett and Frind problem using CELDIS = 30, NSC = NSR = NSL = 4, NT = 32. Contours shown are 0.1 to 0.9.	51
1.30	Contour plot showing concentrations of a three-dimensional finite element solution of the Burnett and Frind problem with high vertical transverse dispersivity. Contours shown are 0.1 to 0.9.	51
1.31	Contour plot showing concentrations of ELLAM solution to high dispersion Burnett and Frind problem using CELDIS = 21, NSC = NSR = NSL = 4, NT = 32. Contours shown are 0.1 to 0.9.	52
1.32	Mass at old time level is integrated exactly using cell centers and pre-images of cell boundaries as integration points, and applying the trapezoidal integration rule.	54
1.33	Initial and advected peaks with $Cr = 10$	68
1.34	Initial and advected peaks with $Cr = \frac{1}{2}$	69
1.35	Initial and advected peaks with $Cr = \frac{1}{4}$	69
1.36	Initial and advected peaks with $Cr = \frac{1}{32}$	70

1.37	Initial and advected peaks with $Cr = 10\frac{1}{32}$	70
1.38	Eigenvectors of $10 \times 10 S(A^{-1}B)^{\frac{1}{Cr}}$ with $Cr = \frac{1}{8}$	72
1.39	Initial condition and results using $10 \times 10 (S(A^{-1}B)^{\frac{1}{Cr}})^T$ with $T = 1, 10, 100, 1000$ and $Cr = \frac{1}{8}$	73
1.40	Initial condition and results using $40 \times 40 (S(A^{-1}B)^{\frac{1}{Cr}})^T$ with $T = 1, 10, 100, 1000, 10000$ and $Cr = \frac{1}{8}$	75
1.41	Initial condition and results using $200 \times 200 (S(A^{-1}B)^{\frac{1}{Cr}})^T$ with $T = 1, 10, 100, 1000, 10000$ and $Cr = \frac{1}{8}$	75
1.42	Initial condition with 2 nodes on a front, and results using 40×40 $(S(A^{-1}B)^{\frac{1}{Cr}})^T$ with $T = 1, 100$ and $Cr = \frac{1}{8}$	76
1.43	Initial condition with 3 nodes on a front, and results using 40×40 $(S(A^{-1}B)^{\frac{1}{Cr}})^T$ with $T = 1, 100$ and $Cr = \frac{1}{8}$	76
1.44	Initial condition with 4 nodes on a front, and results using 40×40 $(S(A^{-1}B)^{\frac{1}{Cr}})^T$ with $T = 1, 100$ and $Cr = \frac{1}{8}$	77
1.45	Initial condition with 5 nodes on a front, and results using 40×40 $(S(A^{-1}B)^{\frac{1}{Cr}})^T$ with $T = 1, 100$ and $Cr = \frac{1}{8}$	77
1.46	Superposition of results showing initial condition and advected peak using $Cr = \frac{1}{2}$. Grids are 5×5 , and 4 refinements, each by a factor of three. Initial condition is a spike on 5×5 grid. Advected peaks show increasing height with decreasing Δx . . .	79
1.47	Superposition of results showing initial condition and advected peak using $Cr = \frac{4}{9}$. Grids are 5×5 , and 4 refinements, each by a factor of three. Initial condition is a spike on 5×5 grid. Advected peaks show increasing height with decreasing Δx . . .	79

2.1 Configuration with two high-coefficient subregions touching at a single node.	118
--	-----

TABLES

Table

1.1	Parameters used in ELLAM simulation of solute transport in a one-dimensional, steady-state flow system.	28
1.2	Parameters used in ELLAM simulation of transport from a continuous point source in a three-dimensional, uniform, steady-state flow system	33
1.3	Parameters used in ELLAM simulation of	37
1.4	Parameters used in ELLAM simulation of three-dimensional transport from a point source with flow in the x-direction and flow at 45 degrees to x- and y-axes.	43
1.5	Parameters used for ELLAM simulation of transport in a vertical plane from a continuous point source in a nonuniform, steady-state, two-dimensional flow system (described by Burnett and Frind, 1987).	48
1.6	Sections represent Courant numbers, $Cr = 1/2, 1/3, 1/64, 1/500, 1/2000, 99/200$, respectively.	66
1.7	Eigenvalue and coefficient of the respective eigenvector in eigenvector decomposition of spike initial condition at node seven on 10×10 grid.	72

2.1	Checkerboard pattern, coefficients $10^\sigma, 10^{-\sigma}$	116
2.2	Two cubes touching at a node. Coefficient= 10^σ in dark subregions (see Figure 2.1).	117
2.3	Element coefficients random in $(10^{-\sigma}, 10^\sigma)$	118

1. ELLAM

1.1 Introduction

Solute transport in flowing ground-water has been intensively studied in recent years in an effort to predict long-term effects of contamination, and to design strategies for remediation. Convective and diffusive processes both move contaminant through saturated subsurface media. Therefore, advection-diffusion transport equations, important in many areas of applied science and engineering, are of particular concern to groundwater hydrologists active in the assessment of aquifer contamination.

Solution of an advection-diffusion transport equation may pose difficult challenges to a numerical method. While diffusion equations tend to be tractable using standard finite difference or finite element methods with a level of discretization cost-effective for application to large systems over extensive time periods, admitting an advective component to the equation may require prohibitively fine discretizations. Consider a one-dimensional prototypical solute transport equation,

$$\frac{\partial c}{\partial t} + v \frac{\partial c}{\partial x} - D \frac{\partial^2 c}{\partial x^2} = 0, \quad (1.1)$$

where c is concentration, v is velocity, and D is diffusion. Define the dimensionless Peclet number by

$$Pe = \frac{vL}{D},$$

where L is a characteristic length of the system, and the grid Peclet number by

$$Pe_g = \frac{v\Delta x}{D},$$

where Δx is the mesh size of the grid. For mesh density such that Δx satisfies

$$Pe_g \leq 2$$

standard Eulerian numerical methods, such as centered finite differences or Galerkin finite elements, perform well. These methods produce nonphysical oscillations in concentration with larger grid Peclet number. Such oscillations are often avoided by using an upstream finite difference or finite element method, which effectively increases D enough to make $Pe_g \leq 2$. The resulting concentration profiles, however, are damped relative to their physical counterparts, and also involve a grid orientation effect due to the introduction of a large numerical dispersion term which is not rotationally invariant [83]. A solution with neither spurious oscillations nor numerical dispersion indeed requires that $Pe_g \leq 2$, meaning

$$n_x = \frac{L}{\Delta x} \leq \frac{Pe}{2}, \quad (1.2)$$

$$\Delta x \leq \frac{2L}{Pe}, \quad (1.3)$$

where n_x is the number of grid cells in the length L . Enforcement of (1.2) in three dimensions is often impractical, since Peclet numbers in the hundreds are common in field problems [].

The need for the scale (1.3) is actually an artifact of the numerical methods and not the result of a physical process. A moving front solution of (1.1) with a step function as the initial concentration profile, has the form

$$c(x, t) = erf\left(\frac{x - vt}{\sqrt{4Dt}}\right),$$

where erf is the error function. The width of this front is proportional to \sqrt{D} , hence to $\frac{1}{\sqrt{Pe}}$. Eulerian methods need a number of cells proportional to \sqrt{Pe} on a front to satisfy Δx proportional to $\frac{1}{Pe}$. Thus the number of cells on a moving front must increase as a problem becomes more advection dominated. A method requiring Δx proportional to $\frac{1}{\sqrt{Pe}}$ would require a constant number of cells on a front, regardless of its width. This would be practical in three dimensions, even with large Pe .

Eulerian methods also tend to suffer from large time truncation errors when applied to advection dominated problems. These errors will be proportional to a power of Δt multiplied by a higher time derivative of the analytic

solution c , depending on the particular time stepping procedure. Errors will be large when a steep front passes, unless time steps are very small. Time stepping in a way which follows the flow, would permit larger time steps without degradation of accuracy.

Subsurface contaminant transport problems involve steep, but not discontinuous fronts because of the combined effects of advection and dispersion. Problems may be advection- or dispersion-dominated, often varying in different parts of the space-time computational domain. Methods are sought which are effective with these combined effects.

Eulerian and characteristic methods are both being advanced to accurately and efficiently solve advection-diffusion transport equations. Eulerian methods are characterized by the use of a fixed spatial grid. Petrov-Galerkin techniques [77, 16, 98, 22], optimal test functions [15, 25, 28, 10, 8], total variation diminishing scheme [29], stabilized finite elements [46], and the streamline diffusion finite element method [39, 63, 20, 64, 62, 51, 67, 66, 69, 68, 74, 102, 103], are alternatives to classical difference or element methods on fixed grids. Each incorporates some strategy to minimize or to tune space or time truncation error in order to avoid oscillations, while introducing as little numerical dispersion as possible. They tend to need restricted size of the time steps for good accuracy.

Characteristic methods treat separately the advective and diffusive components of the transport equation, tracking along characteristics of the flow, then solving a diffusion equation on a fixed grid. These methods include the method of characteristics [48, 78, 11, 59], modified method of characteristics [65, 42], characteristic Galerkin method [32, 93], transport-diffusion method [79], characteristic-mixed finite-element method [2, 101, 3], operator splitting method of [40, 100, 30], the Lagrangian-Galerkin method [76], and Eulerian-Lagrangian methods as discussed below. These methods offer reduced time truncation error compared to Eulerian methods, and can use larger time steps. Primary drawbacks are difficulties in mass conservation and in rigorous formulation of boundary flux.

The Eulerian-Lagrangian localized adjoint method (ELLAM) maintains mass conservation and provides a framework for treatment of general boundary conditions. ELLAM was introduced by Celia et al. [26], Russell [81], and Herrera et al. [58] for constant-coefficient one-dimensional equations, then extended to one-dimensional variable-coefficient cases by Russell and Trujillo [82], Wang [94], and Wang, Ewing, and Russell [96]. Celia and Ferrand [24] and Healy and Russell [55] extended ELLAM to a one-dimensional finite-volume setting. One-dimensional nonlinear advection-diffusion equations have been treated by Ewing [33] and Dahle, Ewing, and Russell [31]. Additional work has been done on ELLAM schemes for one-dimensional transport equations involving reaction [34, 43, 35, 27, 36, 37, 85, 41, 95, 38].

Implementation of ELLAM schemes in multiple spatial dimensions involves additional concerns relative to the one-dimensional case [82]. Various two-dimensional ELLAM schemes have been developed [94, 43, 35, 97, 12, 13, 56], including one with optimal order error estimates proved by Wang. Celia [23] explores a three-dimensional ELLAM in a framework within which all characteristic methods can be viewed.

In this thesis, a three-dimensional ELLAM implementation is described, and numerical results are provided. A forward tracking approach is taken to advection, and dispersion is treated by an implicit formulation in time. The method's apparent robustness is discussed in a one-dimensional context.

1.2 Three-Dimensional ELLAM

A three-dimensional finite-volume Eulerian-Lagrangian localized adjoint method (FVELLAM, [55]) code has been developed to simulate three-dimensional solute transport in flowing ground water for a single dissolved chemical constituent and represents the processes of advective transport, hydrodynamic dispersion (including both mechanical dispersion and diffusion), mixing (or dilution) from fluid sources, and simple chemical reactions (including linear sorption and decay). This code has been implemented as an alternative algorithm within the U.S. Geological Survey (MOC3D) transport model. It is integrated into the MOC3D model, which is itself integrated with

MODFLOW-96, the USGS three-dimensional, finite-difference, ground-water flow model [52], [54], [53]. The code is written in FORTRAN-77.

ELLAM [26] solves an integral form of the solute-transport equation. The code uses an implicit method for dispersion calculations, which allows for large time steps without stability constraints. The Eulerian-Lagrangian approach involves tracking mass through time, then solving a dispersion equation on a fixed-in-space grid. This is particularly advantageous for advection-dominated problems, as are typical of many field problems involving ground-water contamination, since Eulerian-Lagrangian approaches tend to generate less numerical dispersion than standard Eulerian finite-difference or finite-element methods.

1.2.1 Governing Equation for Solute Transport

The solute transport equation is the same as presented in Konikow and others ([71], equation 6):

$$\begin{aligned}
& \frac{\partial(\varepsilon C)}{\partial t} + \frac{\partial(\varrho_b \bar{C})}{\partial t} + \frac{\partial}{\partial x_i}(\varepsilon C V_i) \\
& - \frac{\partial}{\partial x_i} \left(\varepsilon D_{ij} \frac{\partial C}{\partial x_j} \right) - \sum C' W \\
& + \lambda(\varepsilon C + \varrho_b \bar{C}) \\
& = 0
\end{aligned} \tag{1.4}$$

(summation over repeated indices is understood), where C is volumetric concentration (mass of solute per unit volume of fluid, ML^{-3}), ϱ_b is the bulk density of the aquifer material (mass of unit solids per unit volume of aquifer, ML^{-3}), \bar{C} is the mass concentration of solute sorbed on or contained within the solid aquifer material (mass of solute per mass of aquifer material, MM^{-1}), ε is the effective porosity (dimensionless), \mathbf{V} is a vector of interstitial fluid velocity components (LT^{-1}), \mathbf{D} is a second-rank tensor of dispersion coefficients (L^2T^{-1}), W is a volumetric fluid sink ($W < 0$) or fluid source ($W > 0$) rate per unit volume of aquifer (T^{-1}), C' is the volumetric concentration in the sink/source fluid (ML^{-3}), λ is the reactive decay rate (T^{-1}), t is time (T), and x_i are the Cartesian coordinates (L). Porosity is the ratio of pore volume to

the bulk volume of the porous medium. Interstitial velocity is the velocity that fluid moves through the pores of the subsurface medium to achieve a given flow rate across the faces of a volume of the saturated medium.

The terms controlling sorption are combined into a single parameter, the retardation factor (R_f), assumed to be constant in time, since we consider a linear phase-equilibrium relationship in which \bar{C} is proportional to C . The retardation factor is defined as:

$$R_f = 1 + \frac{\rho_b \bar{C}}{\varepsilon C}.$$

An integral form of the solute-transport equation, which is a statement of conservation of mass over the domain of integration, is the governing equation for this finite-volume ELLAM approach. Equation (1.4) is integrated against a space-time test function to provide the formulation, including treatment of (cell or subdomain) boundary conditions and solute decay.

The test function effectively specifies the domain of integration for the transport equation by the portion of the space-time domain where its value is nonzero. On a subdomain of integration, the test function can be seen as an integration weight at each point. Varying the weight along streamlines of the flow is a convenient mechanism to provide solute (growth or) decay.

Divide (1.4) by R_f , multiply by a test function $u(\mathbf{x}, t)$ and integrate over time and space. Assuming R_f is constant in time, we have:

$$\int_{\Omega} \int_0^T \left(u \frac{\partial(\varepsilon C)}{\partial t} + \frac{u}{R_f} \nabla \cdot (\varepsilon C \mathbf{V} - \varepsilon \mathbf{D} \nabla C) - \frac{u}{R_f} \sum C' W + u \lambda \varepsilon C \right) dt d\mathbf{x} = 0, \quad (1.5)$$

where Ω is the entire spatial transport subdomain, and T is the end of the simulation time period starting at time zero. Integrate equation (1.5) is integrated by parts using

$$u \frac{\partial(\varepsilon C)}{\partial t} = \frac{\partial(u \varepsilon C)}{\partial t} - \frac{\partial u}{\partial t} \varepsilon C$$

and

$$\frac{u}{R_f} \nabla \cdot (\varepsilon C \mathbf{V} - \varepsilon \mathbf{D} \nabla C) = \frac{1}{R_f} \nabla \cdot (u \varepsilon C \mathbf{V} - u \varepsilon \mathbf{D} \nabla C) - \frac{1}{R_f} \nabla u \cdot (\varepsilon C \mathbf{V} - \varepsilon \mathbf{D} \nabla C)$$

to yield the global equation,

$$\begin{aligned} \int_{\Omega} \int_0^T \left(\frac{\partial(u\varepsilon C)}{\partial t} + \frac{1}{R_f} \nabla \cdot (u\varepsilon C \mathbf{V} - u\varepsilon \mathbf{D} \nabla C) + \frac{1}{R_f} \nabla u \cdot \varepsilon \mathbf{D} \nabla C \right. \\ \left. - \frac{u}{R_f} \sum C' W - \varepsilon C \left(\frac{\partial u}{\partial t} + \frac{\mathbf{V}}{R_f} \cdot \nabla u - \lambda u \right) \right) dt d\mathbf{x} = 0. \end{aligned} \quad (1.6)$$

The Eulerian-Lagrangian aspects of the method derive from the requirement that the test function satisfy the adjoint equation, $\frac{\partial u}{\partial t} + \frac{\mathbf{V}}{R_f} \cdot \nabla u - \lambda u = 0$. Thus, for the time step from t^n to t^{n+1} , we choose u of the form $u(\mathbf{x}, t) = f(\mathbf{x}, t) e^{-\lambda(t^{n+1}-t)}$, where $\frac{\partial f}{\partial t} + \frac{\mathbf{V}}{R_f} \cdot \nabla f = 0$, so that f is constant along characteristics of the retarded interstitial velocity field. Note that with $u = e^{-\lambda(t^{n+1}-t)}$ (that is, $f = 1$) in the following, we arrive at a statement of global conservation of mass over the time step:

$$\begin{aligned} \int_{\Omega} (u\varepsilon C)^{n+1} d\mathbf{x} - \int_{\Omega} (u\varepsilon C)^n d\mathbf{x} \\ + \int_{\Omega} \int_{t^n}^{t^{n+1}} \frac{1}{R_f} \nabla \cdot (u\varepsilon C \mathbf{V} - u\varepsilon \mathbf{D} \nabla C) + \frac{1}{R_f} \nabla u \cdot \varepsilon \mathbf{D} \nabla C \\ - \frac{u}{R_f} \sum C' W dt d\mathbf{x} \\ = 0. \end{aligned}$$

To obtain a system of equations, each representing mass conservation on a cell, we use $u = \sum_l u_l$ where l is the index for the finite-difference cells Ω_l in the transport subdomain, and local space-time test functions are defined with $f_l(\mathbf{x}, t^{n+1}) = 1$ on Ω_l and $f_l(\mathbf{x}, t^{n+1}) = 0$ elsewhere:

$$u_l = \begin{cases} e^{-\lambda(t^{n+1}-t)} & \text{on characteristics from any } (\mathbf{x}, t) \in \Omega \times (t^n, t^{n+1}) \\ & \text{into } \Omega_l \text{ at time level } n+1 \\ 0 & \text{otherwise.} \end{cases} \quad (1.7)$$

We thus arrive at a system of local ELLAM equations,

$$\begin{aligned}
& \int_{\Omega_l} (\varepsilon C)^{n+1} d\mathbf{x} - e^{-\lambda \Delta t} \int_{\Omega_l^*} (\varepsilon C)^n d\mathbf{x} \\
+ & \iint_{supp u_l \cap \Gamma^{n+1}} \frac{e^{-\lambda(t^{n+1}-t)}}{R_f} (\varepsilon C \mathbf{V} - \varepsilon \mathbf{D} \nabla C) \cdot \mathbf{n} dt ds \\
& - \int_{t^n}^{t^{n+1}} \int_{\partial supp u_l} \frac{e^{-\lambda(t^{n+1}-t)}}{R_f} (\varepsilon \mathbf{D} \nabla C) \cdot \mathbf{n} dt ds \\
& - \iint_{supp u_l \cap supp W} \frac{e^{-\lambda(t^{n+1}-t)}}{R_f} \sum C' W dt d\mathbf{x} \\
& = 0
\end{aligned} \tag{1.8}$$

where $\partial \cdot$ signifies the spatial boundary of the argument; $supp \cdot$ denotes the support of a function, that is, the closure of the part of its domain where a function assumes a nonzero value; \mathbf{n} is the unit outward normal vector on the specified boundary; Ω_l^* is the pre-image at time t^n under advection of cell Ω_l at time t^{n+1} ; $\Gamma^{n+1} \equiv \partial \Omega \times (t^n, t^{n+1})$ is the space-time boundary at time step $n+1$; t denotes time, and $d\mathbf{x}$ and ds signify differential volume and area, respectively.

Note that equations (1.8) appear as space-time integrals of diffusion equations. ELLAM can be viewed as a method of characteristics, tracking mass along streamlines of the flow to accumulate data to the right-hand side of the system of equations.

1.3 Formulation of ELLAM equations

ELLAM approaches the hyperbolic-parabolic nature of the solute transport equation by combining a method of characteristics technique for advection, with a backward Euler in time and centered differences in space solution to a diffusion equation. The details of these approximations depend on the evaluations of the integrals in (1.8).

ELLAM requires knowledge of fluid velocity everywhere within the domain of the transport problem. Consequently, a flow equation is solved on a potentially larger domain, and output is used to solve the solute transport equation on a subdomain. The flow equation is an elliptic or parabolic pressure (groundwater flow) equation, which is solved using the USGS finite-difference code, MODFLOW.

This ELLAM method approximates total solute flux across the transport subdomain boundary by advective flux. It should be noted that this approximation is not demanded by ELLAM methods, in general, but is a feature of this particular implementation. This approximation means that boundary face concentrations are not coupled to cell center concentrations through the numerical derivative (concentration gradient). All mass moving into and out of the transport subdomain can be tracked using the advective algorithm. Mass tracked across outflow boundaries provides data for a system of outflow boundary equations decoupled from the cell equations. User-input inflow concentrations, together with the outflow solutions, then appear on the right-hand side of the system of cell equations representing local statements of mass conservation. These cell integral equations are solved for C^{n+1} , concentration at the new $n + 1$ time level at each cell center.

1.3.1 Cell Integral Equations

Take the conservation of mass equations for each cell (1.8), approximate the total boundary flux with advective flux, the dispersion time integral with a backward Euler formulation, and then rearrange terms. The system of equations to be solved is then,

$$\begin{aligned} \int_{\Omega_l} (\varepsilon C)^{n+1} d\mathbf{x} - \Delta t \int_{\partial\Omega_l} \frac{1}{R_f} (\varepsilon \mathbf{D} \nabla C)^{n+1} \cdot \mathbf{n} ds &= e^{-\lambda \Delta t} \int_{\Omega_l^*} (\varepsilon C)^n d\mathbf{x} \\ &- \iint_{supp u_l \cap \Gamma^{n+1}} e^{-\lambda(t^{n+1}-t)} \varepsilon C_{inflow} \frac{\mathbf{V}}{R_f} \cdot \mathbf{n} dt ds \\ &+ \iint_{supp u_l \cap supp W} e^{-\lambda(t^{n+1}-t)} \sum C' \frac{W}{R_f} dt d\mathbf{x}, \end{aligned} \quad (1.9)$$

where Ω_l^* means the pre-image in the spatial domain at t^n of Ω_l at t^{n+1} , $\Gamma^{n+1} \equiv \partial\Omega \times (t^n, t^{n+1})$ is the space-time boundary at time step $n + 1$, \mathbf{n} is the unit outward normal vector, and $supp f \equiv \overline{\{x | f(x) \neq 0\}}$. Note that the right-hand side of (1.9) consists of advective mass contributions from storage, inflow boundary, and sources. (The term "storage" will refer to solute mass already in the transport system at the end of the last time step.) Figure 1.1 illustrates the possibility of all of these mass contributions being nonzero.

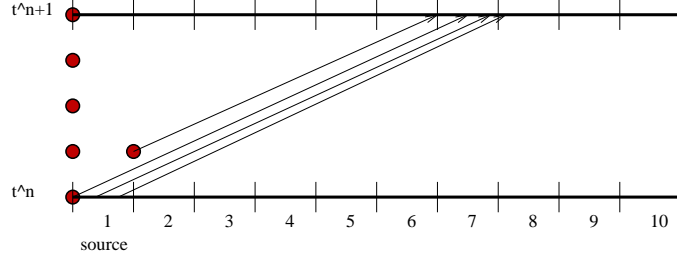


Figure 1.1. Cell 7 receives advected mass from inflow, source in cell 1, and storage in cell 1.

1.3.2 Outflow Boundary Equations

The term in (1.7) expressing mass crossing the transport subdomain boundary during a time step is:

$$\int_{\Omega} \frac{1}{R_f} \int_{t^n}^{t^{n+1}} \nabla \cdot (u \varepsilon C \mathbf{V} - u \varepsilon \mathbf{D} \nabla C) dt d\mathbf{x} = \int_{t^n}^{t^{n+1}} \int_{\partial\Omega} \frac{1}{R_f} (u \varepsilon C \mathbf{V} - u \varepsilon \mathbf{D} \nabla C) \cdot \mathbf{n} ds dt.$$

Considering just the outflow portion of the boundary, this becomes,

$$\int_{t^n}^{t^{n+1}} \int_{(\partial\Omega)_{outflow}} \frac{1}{R_f} (u \varepsilon C \mathbf{V} - u \varepsilon \mathbf{D} \nabla C) \cdot \mathbf{n} ds dt \approx \int_{(\partial\Omega)_{outflow}} \int_{t^n}^{t^{n+1}} (u \varepsilon C_{outflow} \frac{\mathbf{V}}{R_f}) \cdot \mathbf{n} dt ds, \quad (1.10)$$

where total flux across the boundary is now approximated by advective flux.

We index the outflow boundary faces with ll and define the following test functions:

$$u_{ll} = \begin{cases} e^{-\lambda(t^{n+1}-t)} & \text{on characteristics from } \Omega \text{ at time level } n \\ & \text{into boundary area } (\partial\Omega)_{ll} \text{ at any time during time step} \\ 0 & \text{otherwise.} \end{cases} \quad (1.11)$$

The mass across outflow boundary face ll is the mass stored at the previous time level which flows across the face, together with any inflow and source mass that both enters the transport subdomain, and leaves through cell face ll during the time step, as illustrated in figure 1.2, is given by,

$$\int_{(\partial\Omega)_{ll}} \int_{t^n}^{t^{n+1}} e^{-\lambda(t^{n+1}-t)} \varepsilon C_{outflow} \frac{\mathbf{V}}{R_f} \cdot \mathbf{n} dt ds = \quad (1.12)$$

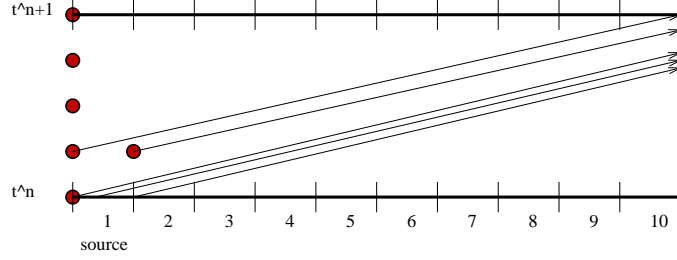


Figure 1.2. Outflow boundary receives advected mass from inflow, source in cell 1, and storage in all cells.

$$\begin{aligned}
& e^{-\lambda\Delta t} \int_{(\partial\Omega)_U^*} (\varepsilon C)^n d\mathbf{x} \\
& - \iint_{\text{supp } u_{il} \cap \partial\Omega_{inflow} \times (t^n, t^{n+1})} e^{-\lambda(t^{n+1}-t)} \varepsilon C_{inflow} \frac{\mathbf{V}}{R_f} \cdot \mathbf{n} dt ds \\
& + \iint_{\text{supp } u_{il} \cap \text{supp } W \times (t^n, t^{n+1})} e^{-\lambda(t^{n+1}-t)} \sum C' \frac{W}{R_f} dt d\mathbf{x},
\end{aligned}$$

where $(\partial\Omega)_U$ is a boundary face, and $(\partial\Omega)_U^*$ is the preimage at time t^n of $(\partial\Omega)_U \times [t^n, t^{n+1}]$. This system of equations will be solved for $C_{outflow}$.

1.3.3 Assumptions

As described by Konikow and others [71], a number of assumptions have been made in the development of the governing equations for a coupled system modeling flow and transport. For the derivation of the transport equation, we assume:

- Chemical reactions do not affect the fluid or aquifer properties.
- The dispersivity coefficients are constant over a flow time step, and the aquifer isotropic with respect to longitudinal dispersivity.

1.4 Numerical Methods

The groundwater velocity field needed for solution of the transport equation is obtained using the USGS MODFLOW code. This implicit finite-difference code yields a head (pressure) distribution which is calculated for a given time step or steady-state flow condition. The specific discharge, or flow

rate per unit area, across each face of every finite-difference cell within the transport subgrid is calculated and used to find the interstitial velocity at any point using an interpolated value of the specific discharge, divided by porosity.

1.4.1 Mass Tracking

For each cell in the fixed finite-difference grid, the integrals on the right-hand side of equation (1.9) represent solute mass advected into the cell during the time step from storage, the transport subdomain boundary, or a fluid source, respectively. These integrals are formed by accumulating mass tracked forward along characteristic curves of the velocity field, determined from the MODFLOW solution as described above. Velocity at any point is determined by linearly interpolating between cell faces in the direction of the component of interest, and piecewise-constant interpolation in the other two directions.

A system of three ordinary differential equations is solved to find the characteristic curves $[x = x(t), y = y(t); z = z(t)]$ along which the fluid is advected:

$$\frac{dx}{dt} = \frac{V_x}{R_f} \tag{1.13}$$

$$\frac{dy}{dt} = \frac{V_y}{R_f} \tag{1.14}$$

$$\frac{dz}{dt} = \frac{V_z}{R_f} \tag{1.15}$$

This is accomplished by introducing a set of moving points that can be traced within the stationary coordinates of a finite-difference grid. Each point corresponds to one characteristic curve, and values of x , y , and z are obtained as functions of t for each characteristic [47]. Each point moves through the flow field by the flow velocity acting along its trajectory.

The ELLAM equations, (1.9) and (1.12) suggest that mass is tracked backwards along characteristics to the pre-image of each cell or boundary face. It is not possible to exactly locate all of the mass at the previous time level by backtracking a finite number of points, however. In order to achieve mass balance, the known mass distribution at the old time level is tracked forward

to the new time level. (See figure 1.3.) This approach ensures that mass will not be lost or gained in advection from one time level to the next, but doesn't guarantee an accurate mass distribution at the new time.

ELLAM tracks points which are the centers of volumes of fluid. Thus mass in a fluid volume is tracked under advection during a time step, distributed among destination cells, and accumulated to the right-hand side storage, inflow, or source integral for each cell.

1.4.2 Numerical Integration

We will now discuss the numerical treatment of (1.9) and (1.12). The j, i, k subscripts for a cell Ω_t will denote the spatial finite-difference grid indexing, with this index order indicating an x, y, z sequencing. A right-handed coordinate system, with the vertical index increasing from top to bottom is used.

The equations are first divided through by porosity, which is represented by a piecewise constant function in space and time. Each individual term will now be discussed.

1.4.3 Dispersion

Time integration is accomplished using a one point in time backward Euler rule. Spatially, a one point integration rule with a seven point stencil is used:

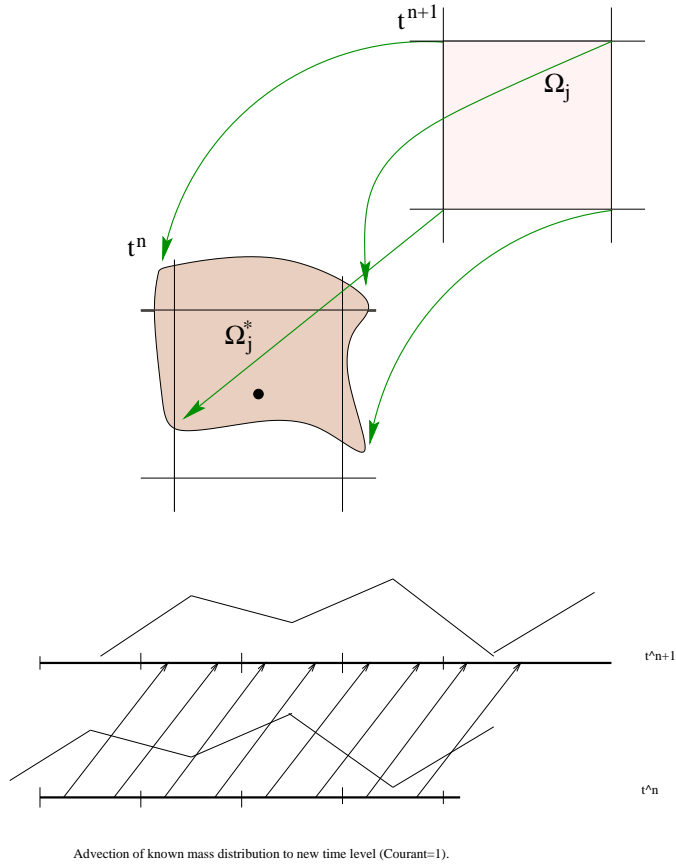
$$\Delta t \int_{\partial\Omega_t} \frac{1}{\varepsilon R_f} (\varepsilon \mathbf{D} \nabla C)^{n+1} \cdot \mathbf{n} \, ds = \tag{1.16}$$

$$\frac{\Delta t}{(R_f)_k (\varepsilon b)_{j,i,k}^{n+1}} \left\{ \left[\left(\varepsilon b D_{1m} \frac{\partial C}{\partial x_m} \right)_{j+\frac{1}{2},i,k}^{n+1} - \left(\varepsilon b D_{1m} \frac{\partial C}{\partial x_m} \right)_{j-\frac{1}{2},i,k}^{n+1} \right] \Delta y_i b_{j,i,k}^{n+1} \right.$$

$$+ \left[\left(\varepsilon b D_{2m} \frac{\partial C}{\partial x_m} \right)_{j,i+\frac{1}{2},k}^{n+1} - \left(\varepsilon b D_{2m} \frac{\partial C}{\partial x_m} \right)_{j,i-\frac{1}{2},k}^{n+1} \right] \Delta x_j b_{j,i,k}^{n+1}$$

$$\left. + \left[\left(\varepsilon b D_{3m} \frac{\partial C}{\partial x_m} \right)_{j,i,k+\frac{1}{2}}^{n+1} - \left(\varepsilon b D_{3m} \frac{\partial C}{\partial x_m} \right)_{j,i,k-\frac{1}{2}}^{n+1} \right] \Delta x_j \Delta y_i \right\}$$

where $m = 1, 2, 3$ is the summation index for the dispersion term. Finite-difference approximations to the space derivatives in the dispersion integral



Advection of known mass distribution to new time level (Courant=1).

Figure 1.3. Preimage of cell may be irregularly shaped and not easily delimited by backtracking. Instead, the known mass distribution at time t^n is tracked forward along streamlines of the advective flow to time t^{n+1} .

are calculated using centered differences as in MOC3D, generalized for varying grid dimensions. (See [71], page 64, for the form of the $\varepsilon b D_m \frac{\partial C}{\partial x_m}$ expansions.)

1.4.4 Storage at New Time Level

The quantity mass/porosity in a cell at the new time level t^{n+1} is expressed using the trapezoidal rule for integration, formulated over each cell octant. Concentrations at octant corners are weighted averages of neighboring node concentrations, determined by trilinear interpolation.

For each octant,

$$\begin{aligned} \frac{mass}{porosity} &= \frac{1}{64} \Delta x_j \Delta y_i b_{j,i,k}^{n+1} \sum_{corner=1}^8 C_{corner} \\ &= \frac{1}{64} \Delta x_j \Delta y_i b_{j,i,k}^{n+1} \sum_{corner=1}^8 \sum_{nbr=1}^8 (weight)_{nbr} C_{nbr}, \end{aligned} \quad (1.17)$$

where for an interior octant, nbr is one of the eight neighboring grid nodes between which concentration varies trilinearly. In case of a boundary octant, a boundary face value is needed for calculation, and is taken to be the following:

- Inflow - user input;
- No flow - same as associated interior node;
- Outflow - calculated using cell parameters, boundary flow rate, and mass tracked across boundary during transport time step.

Coefficients calculated by $(1/8) * octant\ volume * nodal\ weight$ for all nodes neighboring a cell comprise the storage matrix entries for the equation for each cell. Boundary terms are put on the right-hand side of the equation because all boundary face concentrations are determined before the solution of the interior equations.

It should be noted that linear interpolation is approximate in the case where adjacent cells in the same layer of the the transport subdomain have varying thicknesses, as is allowed by MODFLOW. Extreme variations could adversely affect accuracy of the solution.

For an interior cell with all neighbors active and of the same thickness, using b values at time $n + 1$:

$$\begin{aligned} \int_{\Omega_i} C^{n+1} d\mathbf{x} \\ = \left(\left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) + \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) - 4 \right) \end{aligned}$$

$$\begin{aligned}
& \times \left(\left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) + \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) - 4 \right) \\
& \times \left(\left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k-1}} \right) + \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k+1}} \right) - 4 \right) C_{j,i,k} \\
& - \left(\left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) \left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k-1}} \right) \right) C_{j-1,i-1,k-1} \\
& + \left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) \left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k+1}} \right) C_{j-1,i-1,k+1} \\
& + \left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k+1}} \right) C_{j-1,i+1,k+1} \\
& + \left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k-1}} \right) C_{j-1,i+1,k-1} \\
& + \left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k-1}} \right) C_{j+1,i-1,k-1} \\
& + \left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k+1}} \right) C_{j+1,i-1,k+1} \\
& + \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k-1}} \right) C_{j+1,i+1,k-1} \\
& + \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k+1}} \right) C_{j+1,i+1,k+1} \\
& + \left(\left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) + \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) - 4 \right) \\
& \times \left(\left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k-1}} \right) \right) C_{j,i-1,k-1} \\
& + \left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k+1}} \right) C_{j,i-1,k+1} \\
& + \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k-1}} \right) C_{j,i+1,k-1} \\
& + \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k+1}} \right) C_{j,i+1,k+1} \\
& + \left(\left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) + \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) - 4 \right) \\
& \times \left(\left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k+1}} \right) \right) C_{j-1,i,k+1} \\
& + \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k-1}} \right) C_{j+1,i,k-1}
\end{aligned}$$

$$\begin{aligned}
& + \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k+1}} \right) C_{j+1,i,k+1} \\
& + \left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k-1}} \right) C_{j-1,i,k-1} \\
& + \left(\left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k-1}} \right) + \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k+1}} \right) - 4 \right) \\
& \times \left(\left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) C_{j+1,i-1,k} \right. \\
& + \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) C_{j+1,i+1,k} \\
& + \left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) \left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) C_{j-1,i-1,k} \\
& + \left. \left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) C_{j-1,i+1,k} \right) \\
& - \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k-1}} \right) \left(\left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) \right. \\
& \times \left. \left(\left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) + \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) - 4 \right) \right. \\
& + \left. \left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) \left(\left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) - 4 \right) \right. \\
& + \left. \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) \left(\left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) - 4 \right) \right. \\
& - \left. 4 \left(\left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) - 4 \right) C_{j,i,k-1} \right. \\
& - \left. \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k+1}} \right) \left(\left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) \right. \right. \\
& \times \left. \left. \left(\left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) + \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) - 4 \right) \right. \right. \\
& + \left. \left. \left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) \left(\left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) - 4 \right) \right. \right. \\
& + \left. \left. \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) \left(\left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) - 4 \right) \right. \right. \\
& - \left. \left. 4 \left(\left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) - 4 \right) C_{j,i,k+1} \right. \right. \\
& - \left. \left. \left(\left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k-1}} \right) + \left(\frac{b_{j,i,k}}{b_{j,i,k} + b_{j,i,k+1}} \right) - 4 \right) \right. \right.
\end{aligned}$$

$$\begin{aligned}
& \times \left(\left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) + \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) - 4 \right) \left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) C_{j,i-1,k} \\
& + \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) C_{j,i+1,k} \left(\left(\frac{\Delta y_i}{\Delta y_{i-1} + \Delta y_i} \right) + \left(\frac{\Delta y_i}{\Delta y_{i+1} + \Delta y_i} \right) - 4 \right) \\
& \times \left(\left(\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j} \right) C_{j-1,i-1,k} + \left(\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j} \right) C_{j+1,i,k} \right).
\end{aligned}$$

1.4.5 Mass Storage at Old Time Level

The total mass advected into each cell during a transport time step that was already stored within the system at the old time level is needed for the right-hand side of the ELLAM equation. Numerically, this is accomplished by tracking mass forward from the old time level, t^n , along characteristics. Each cell is divided into subcells determined by parameters NSC, NSR, and NSL, specifying the number of subcells in the column, row, and layer direction, respectively. The center of each subcell is tracked through the time step under advection. Depending on the exact location of this point in the destination cell at the new time, all of the mass in the subcell may or may not also be found in that destination cell. In order to mitigate the effects of unwarranted mass lumping, subcell mass is distributed among cells neighboring the destination cell using the “approximate test functions”, w_l , described below. The value of w_l at the subcell center destination point is the fraction of subcell mass to be distributed to cell Ω_l .

This yields the formulation,

$$e^{-\lambda \Delta t} \int_{\Omega_i^*} C^n d\mathbf{x} = e^{-\lambda \Delta t} \sum_{j,i,k} \sum_{\substack{p= \\ \text{subcell} \\ \text{center}}} \frac{\Delta x_j \Delta y_i b_{j,i,k}}{(NSC)(NSR)(NSL)} w_l(p^f) C(p) \quad (1.18)$$

where summation runs through all subcells of each cell in the transport subdomain, and p^f is the image of p under forward tracking to the new time level.

1.4.6 Approximate Test Functions

For each active cell, an approximate test function is defined for the purpose of distributing advected mass from the old time level among neighboring cells at the new time level. The designation “approximate test function”

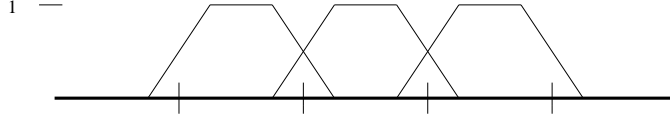


Figure 1.4. Approximate test functions in one direction on a uniform grid, with NS=2.

is given since the graph of this function looks like a characteristic (indicator) function with slant sides extending into adjacent cells, while the test functions described in the derivation of the governing equation are exactly characteristic functions at time t^{n+1} . An approximate test function is determined by NSC, NSR, and NSL, the proximity of the transport subdomain boundary, and the active status of neighboring cells. Mass is not split across the transport boundary or into inactive cells.

An approximate test function associated with interior cell Ω_j is the product of a one-dimensional approximate test function in each direction:

$$w_{jik}(\hat{x}, \hat{y}, \hat{z}) = f(\hat{x})g(\hat{y})h(\hat{z}).$$

On a uniform grid, we define reference coordinates $\hat{x}, \hat{y}, \hat{z}$ with respect to a cell Ω_l with node indices j, i, k by

$$\hat{x} = \frac{x - x_j}{\Delta x_j},$$

and similarly for \hat{y} and \hat{z} . For an interior cell on a uniform grid with all surrounding cells active,

$$f(\hat{x}) = \begin{cases} 0 & \hat{x} \leq -\frac{1}{2} - \frac{1}{2NSC} \\ NSC\hat{x} + \frac{1}{2}(NSC + 1) & -\frac{1}{2} - \frac{1}{2NSC} < \hat{x} < -\frac{1}{2} + \frac{1}{2NSC} \\ 1 & -\frac{1}{2} + \frac{1}{2NSC} \leq \hat{x} \leq \frac{1}{2} - \frac{1}{2NSC} \\ -NSC\hat{x} + \frac{1}{2}(NSC + 1) & \frac{1}{2} - \frac{1}{2NSC} < \hat{x} < \frac{1}{2} + \frac{1}{2NSC} \\ 0 & \frac{1}{2} + \frac{1}{2NSC} \leq \hat{x} \end{cases}$$

and similarly for g and h. These functions are shown graphically in figures 1.4 and 1.5.

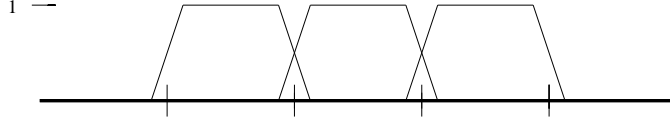


Figure 1.5. Approximate test functions in one direction on a uniform grid, with NS=4.

On a nonuniform grid, test functions are scaled by grid ratios. In one direction on a nonuniform grid, with neighboring cells active, approximate test functions corresponding to f defined above are illustrated graphically in figure 1.6.

For a point in Ω_l with reference coordinates $\hat{x}, \hat{y}, \hat{z} \in [-\frac{1}{2}, \frac{1}{2}]$, f is defined generally, on a potentially nonuniform grid by,

$$f(\hat{x}) = \begin{cases} 2NSC(1 - (\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j}))\hat{x} + (\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j}) + NSC(1 - (\frac{\Delta x_j}{\Delta x_{j-1} + \Delta x_j})) & -\frac{1}{2} \leq \hat{x} < -\frac{1}{2} + \frac{1}{2NSC} \\ 1 & -\frac{1}{2} + \frac{1}{2NSC} \leq \hat{x} \leq \frac{1}{2} - \frac{1}{2NSC} \\ 2NSC((\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j}) - 1)\hat{x} + (\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j}) + NSC(1 - (\frac{\Delta x_j}{\Delta x_{j+1} + \Delta x_j})) & \frac{1}{2} - \frac{1}{2NSC} < \hat{x} < \frac{1}{2} \end{cases} \quad (1.19)$$

and similarly for g and h . There are seven more test functions which may be nonzero at any given point in cell j, i, k . If $\hat{x}, \hat{y}, \hat{z} > 0$ their values given by,

$$\begin{aligned} w_{j+1,i,j} &= (1 - f(\hat{x})) g(\hat{y}) h(\hat{z}), \\ w_{j,i+1,k} &= f(\hat{x}) (1 - g(\hat{y})) h(\hat{z}), \\ w_{j,i,k+1} &= f(\hat{x}) g(\hat{y}) (1 - h(\hat{z})), \\ w_{j+1,i+1,k} &= (1 - f(\hat{x})) (1 - g(\hat{y})) h(\hat{z}), \\ w_{j,i+1,k+1} &= f(\hat{x}) (1 - g(\hat{y})) (1 - h(\hat{z})), \\ w_{j+1,i,k+1} &= (1 - f(\hat{x})) g(\hat{y}) (1 - h(\hat{z})), \\ w_{j+1,i+1,k+1} &= (1 - f(\hat{x})) (1 - g(\hat{y})) (1 - h(\hat{z})). \end{aligned}$$

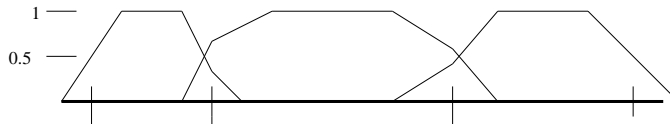


Figure 1.6. Approximate test functions in one direction on a nonuniform grid, with NS=2.

For $\hat{x}, \hat{y}, \hat{z}$ in any other octant of the cell, there are also eight potentially nonzero test functions evaluated similarly. Thus equations of the form (1.19) for f, g, h , defined explicitly for reference coordinates in the interval $[-\frac{1}{2}, \frac{1}{2}]$, are sufficient to evaluate all test functions at any point, and serve as a global definition.

The sum of the values of the test functions at any point in the transport domain is one, thus conserving mass for the integral equation.

The test function component in the direction normal to the boundary extends from the center of a boundary cell to the boundary face with the value of one. Thus there is no splitting of mass across the boundary.

There is no test function associated with an inactive cell. In a cell adjacent to an inactive cell, the value of the test function that would normally be assigned to the inactive cell is distributed proportionally to other test functions that are nonzero at that point. All test functions are zero in inactive cells.

Extreme variation in cell thickness among neighboring cells in a layer may adversely affect model results. In this case, linear interpolation of concentration or approximate distribution of advected mass may be inaccurate.

1.4.7 Source Integral

The last term in (1.9) pertains to sources and sinks within the transport subdomain. ELLAM assumes a source or sink acts uniformly over the entire cell surrounding a source or sink node.

To treat a source, a single time step is discretized into a number of sub-time steps determined by parameter NT, and the composite trapezoidal integration rule is applied in time. This time discretization evens the spatial distribution of incoming mass: Mass is tracked to varying locations within the transport subgrid depending on when in the transport time step the mass enters the system. At each sub-time step, inflow mass is spatially discretized, tracked, and accumulated just like mass already in the system at the start of the transport time step, but for the shorter interval.

Mass from sources is accumulated to the right-hand side of the local conservation of mass equation for cell Ω_l with the following integration. The

domain of integration is all sources which intersect the space-time test function for cell Ω_l , so that all source mass that flows into Ω_l during a time step is integrated. Multiple sources within a source cell are summed.

$$\begin{aligned} & \iint_{\text{supp } u_l \cap \text{supp } W} e^{-\lambda(t^{n+1}-t)} \frac{1}{\varepsilon} \sum C' \frac{W}{R_f} dt d\mathbf{x} \\ & \approx \frac{1}{\varepsilon_{j,i,k}} \sum_{\substack{\text{all} \\ \text{sources}}} \sum_{\substack{p= \\ \text{source} \\ \text{subcell} \\ \text{center}}} \sum_{m=0}^{NT} e^{-\lambda(t^{n+1}-t_m)} \frac{\Delta T_m}{NSC NSR NSL} w_l(p^f) \sum_{\substack{s= \\ \text{source} \\ \text{in} \\ \text{cell}}} C'_s \frac{Q_s}{R_f} \quad (1.20) \end{aligned}$$

where summation runs through all sources in all subcells of every source cell in the transport subdomain; p^f is the image of p under forward tracking to the new time level; Q_s is the flow rate of source s in the source cell; $\Delta T = \frac{\Delta t}{NT}$, or $\frac{\Delta t}{2NT}$ if $m = 0$ or NT ; and $t_m = t^n + \frac{m}{NT} \Delta t$ represents the time during time step at which discretized source mass enters the system.

1.4.8 Sink Integral

Analytically, the domain of integration is the support of the space-time test function for a cell Ω_l , intersected with any sink cells. To approximate, this term is formulated only if Ω_l is a sink, and sink concentration is assumed to be the average nodal concentration for the transport time step, with the exception of a sink due to evapotranspiration, where sink concentration is taken to be zero. Integration rules are a one point in space and a one point backward Euler in time. The averaging of concentration results in this integral approximation contributing to both the left- and right-hand sides of the equation for sink cell Ω_l with coordinates j, i, k . This gives

$$\begin{aligned}
& \iint_{\text{supp } u_i \cap \text{supp } W} e^{-\lambda(t^{n+1}-t)\frac{1}{\varepsilon}} \sum C' \frac{W}{R_f} dt d\mathbf{x} \\
&= \Delta t \int_{\Omega_{\text{sink}}} \frac{1}{\varepsilon} (C_{\text{ave}} \sum \frac{W}{R_f} + C_{\text{et}} \frac{W_{\text{et}}}{R_f}) d\mathbf{x} \\
&\approx \frac{\Delta t}{\varepsilon_{j,i,k}} \frac{(C^{n+1} + C^n)}{2} \left(C_{\text{et}} \frac{Q_{\text{et}}}{R_f} + \sum_{\substack{s= \\ \text{sink} \\ \text{in} \\ \text{cell}}} \frac{Q_s}{R_f} \right),
\end{aligned} \tag{1.21}$$

where Q is sink flow rate; et refers to evapotranspirative flux; and multiple sinks in a sink cell are summed.

1.4.9 Inflow Boundary Integral

For an inflow boundary integral as for a source, a single time step is discretized into a number of sub-time steps determined by parameter NT. The composite trapezoidal rule is applied in time. At each sub-time step, inflow mass is spatially discretized, tracked, and accumulated just like mass already in the system at the start of the transport time step, but for the shorter interval. The only difference in the treatment of the inflow boundary from the treatment of the source is that only the two-dimensional boundary face is discretized, while for a source, the entire cell is discretized. For a cell Ω_l , the integration is performed over the intersection of the space-time test function for that cell and the transport subdomain boundary; that is, all mass entering through the boundary and advected to Ω_l during the time step is accumulated to the right-hand side of local equation l . Mass advected into Ω_l during a time step from the inflow portion of the transport subdomain boundary is give by,

$$\begin{aligned}
& \iint_{\text{supp } u_l \cap \Gamma^{n+1}} e^{-\lambda(t^{n+1}-t)} C_{\text{inflow}} \frac{\mathbf{V}}{R_f} \cdot \mathbf{n} dt ds \\
&\approx \sum_{\substack{\text{all} \\ \text{inflow} \\ \text{faces}}} \sum_{\substack{p= \\ \text{face} \\ \text{subarea} \\ \text{center}}} \sum_{m=1}^{NT} e^{-\lambda(t^{n+1}-t_m)} \frac{\Delta T_m}{\text{ref area}} w_l(p^f) C_{\text{inflow}} Q
\end{aligned} \tag{1.22}$$

where $ref\ area = NSC\ NSR, NSC\ NSL, \dots$, depending on plane of face; p^f is the image of p under forward tracking to the new time level; Q is inflow rate; $t_m = t^n + \frac{m}{NT}\Delta t$ represents the time during time step at which discretized source mass enters the system; and $\Delta T = \frac{\Delta t}{NT}$, or $\frac{\Delta t}{2NT}$ if $m = 1$ or NT . Summation runs over each p on the transport subdomain inflow boundary, with the approximate test function w_l used to select mass advected to Ω_l .

1.4.10 Outflow Integrals

Concentration is calculated at each outflow boundary face using cell parameters, velocity information from MODFLOW, and the amount of mass tracked across the cell boundary determined by ELLAM.

On the left hand side of the system of boundary equations (1.24) is an integral approximated using a one point in space, one point backward Euler in time formulation. This time approximation eliminates the exponential factor, and yields,

$$\begin{aligned} & \int_{(\partial\Omega)_{ll}} \int_{t^n}^{t^{n+1}} e^{-\lambda(t^{n+1}-t)} C_{outflow} \frac{\mathbf{v}}{R_f} \cdot \mathbf{n} dt ds \\ & \approx \Delta t \int_{(\partial\Omega)_{ll}} C_{outflow} \frac{\mathbf{v}}{R_f} \cdot \mathbf{n} ds \\ & \approx \Delta t Q_{ll} C_{outflow} \end{aligned} \tag{1.23}$$

where ll is the index for boundary faces; and Q_{ll} is the outflow rate across boundary face ll , determined using the outflow velocity calculated from MODFLOW output and cell parameters. The concentration on face ll is the unknown in the boundary equation.

The right-hand side boundary integrals are constructed from the mass contributions tracked across the boundary from interior cells, sources, and inflow boundaries during the transport time step. All mass associated with a tracked point that reaches the outflow boundary at any time during the time step is considered to leave the transport subdomain. Test functions are evaluated to distribute mass among neighboring boundary outflow faces.

Numerically, the right-hand side of (1.12) is accomplished using a trapezoidal in time, midpoint on subcell in space approximation:

$$\begin{aligned}
& \iint_{supp u_{i_l} \cap \Gamma^{n+1}} e^{-\lambda(t^{n+1}-t)} C \frac{\mathbf{V}}{R_f} \cdot \mathbf{n} dt ds \\
& \approx e^{-\lambda \Delta t} \sum_{j,i,k} \sum_{\substack{p= \\ \text{subcell} \\ \text{center}}} \frac{\Delta x_j \Delta y_i b_{j,i,k}}{(NSC)(NSR)(NSL)} w_{i_l}(p^f) C(p) \\
& + \sum_{\substack{\text{all} \\ \text{inflow} \\ \text{faces}}} \sum_{\substack{p= \\ \text{face} \\ \text{subarea} \\ \text{center}}} \sum_{m=0}^{NT} e^{-\lambda(t^{n+1}-t_m)} \frac{\Delta T}{ref\ area} w_{i_l}(p^f) C_{inflow} Q \\
& + \frac{1}{\varepsilon_{j,i,k}} \sum_{\substack{\text{all} \\ \text{sources}}} \sum_{\substack{p= \\ \text{source} \\ \text{subcell} \\ \text{center}}} \sum_{m=0}^{NT} e^{-\lambda(t^{n+1}-t_m)} \frac{\Delta T_m}{NSC\ NSR\ NSL} w_{i_l}(p^f) \sum_{\substack{s= \\ \text{source} \\ \text{in} \\ \text{cell}}} C'_s \frac{Q_s}{R_f}
\end{aligned} \tag{1.24}$$

where the approximations to the integrals of stored, inflow, and source mass are as in (1.18), (1.22), and (1.20), except that the approximate test function for cell Ω_l has been replaced by w_{i_l} , the approximate test function for boundary face ll . Boundary face test functions are defined analogously to those associated with cells, but only using factors in the two directions parallel to the boundary face. No mass is distributed to a neighboring boundary face which is not part of the outflow boundary.

We thus have a system of equations represented by a diagonal matrix, to be solved for $C_{outflow}$.

1.4.11 Decay

When simulating linear decay, all mass in the system at the beginning of each transport time step is decayed by a factor $e^{-\lambda t}$, where λ is the decay rate. Inflow and source mass are decayed in the same way, where the time interval is now not the entire time step, but the portion of it during which new mass is in the transport subdomain. This decay algorithm has no numerical stability restrictions associated with it. If the half-life is on the order of or smaller than the transport time step, however, some accuracy will be lost.

When a solute subject to decay enters the aquifer through a fluid source, it is assumed that the fluid source contains the solute in the concentration specified by C' . The ELLAM simulator allows decay to occur only within the ground-water system, and not within the source reservoir. In other words, for a given stress period, C' remains constant in time.

1.4.12 Assumptions

Implicit in the above numerical treatment of the terms of the integral equations are the following assumptions:

- Concentration at an outflow boundary face at the new time level is well approximated by the mass crossing the face during the time step divided by the fluid volume across the face.
- Mass in or out of the transport subdomain during a time step via a source or sink cell is well approximated by the average nodal concentration during the time step times the fluid volume through the source or sink.
- Cell thicknesses are smoothly varying within a vertical layer.
- Transport subgrid boundaries are assumed to be far enough from the plume that any errors in the treatment of the boundaries will not have a significant effect on the solution. The boundary condition is that the normal component of the concentration gradient on the boundary is zero, so there is no dispersive flux across the transport subdomain boundary.

1.5 Test Problems

The three-dimensional ELLAM code was tested by running the same suite of test cases as was applied to the USGS three-dimensional method of characteristics code, MOC3D. These tests were first used by Konikow and others [71] to evaluate the MOC3D Version 1 implementation incorporating an explicit formulation of the dispersion equation, then by Kipp and others [70] for MOC3D Version 2, an implicit method. These benchmark problems are used to evaluate features of a solute transport code with relevance to field

applications.

1.5.1 One-Dimensional Flow

The first test case is a simple one-dimensional system involving solute transport in a finite-length aquifer with flow of constant velocity. Boundary conditions are third-type, although ELLAM approximates total flux boundary conditions using advective flux. The numerical results are compared to a solution by Wexler [99]. Parameters for the model are specified in table 1.5.1.

A low dispersion and a high dispersion case are presented here. In both cases, ELLAM results for CELDIS = $\frac{1}{2}$ (241 time steps), NSC = 4, NSR = NSL = 2, NT = 128 are essentially identical to the analytical results, and so are not plotted. Instead, the results for substantially fewer time increments are shown. For the low dispersion case, $D_{xx} = 0.01$, or $\alpha_L = 0.1\text{cm}$. The analytic solution and two ELLAM solutions are graphed in figure 1.7. For CELDIS = 1, which requires 121 time steps, there is a very close match between the analytical and numerical solutions, using NSC = 32, NSR = NSL = 2, NT = 128. Setting NSC = 4 results in a slightly low concentration at the first grid node, but quite accurate values elsewhere. The NSC = 4 solution is computed in about a quarter of the time used for the finer, NSC = 32, discretization on a Data General Unix workstation [57] (4 min 20 sec for NSC = 32, and 1 min 0 sec for NSC = 4). For CELDIS = 10.1 (12 time increments, simulation time, 15 sec [57]), NSC = 4, NSR = NSL = 2, and NT = 128, concentrations at early times and short distances are somewhat low, but the results are good elsewhere. Thus, ELLAM compares favorably with explicit MOC3D which needed 2401 time steps to satisfy stability criteria, and with implicit MOC3D which required 241 time increments. In the high dispersion case, $D_{xx} = 0.1$, or $\alpha_L = 1.0\text{cm}$. The analytical solution, and concentrations for CELDIS = 1 (121 time steps) and CELDIS = 10.1 (12 time steps) are plotted in figure 1.8. The CELDIS = 1 results are very close to the analytical solution, and without the oscillations produced by MOC3D at short distances. The solution using fewer time steps has concentrations which are somewhat low near the inflow boundary, and high near the outflow boundary.

Parameter	Value
$T_{xx} = T_{yy}$	0.01 cm ² /s
ε	0.1
α_L	0.1 cm
$\alpha_{TH} = \alpha_{TV}$	0.1 cm
PERLEN (length of stress period)	120 sec
V_x	0.1 cm/s
$V_y = V_z$	0.0 cm/s
Initial concentration (C^0)	0.0
Source concentration (C')	1.0
Number of rows	1
Number of columns	122
Number of layers	1
DELR (Δx)	0.1 cm
DELC (Δy)	0.1 cm
Thickness (b)	1.0 cm

Table 1.1. Parameters used in ELLAM simulation of solute transport in a one-dimensional, steady-state flow system.

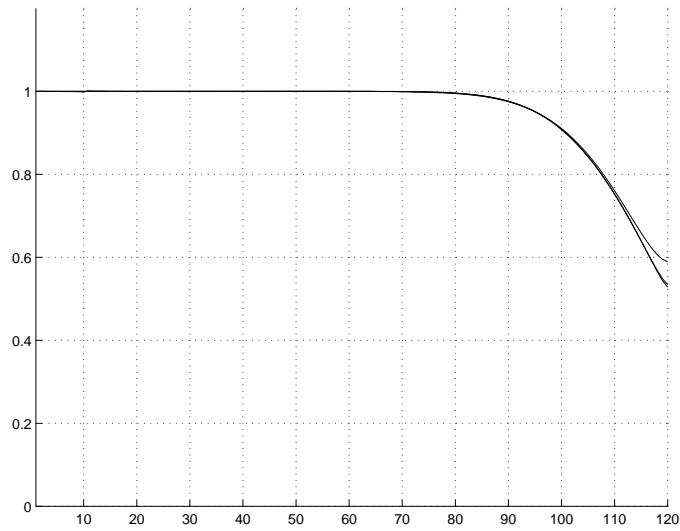


Figure 1.7. Plots of concentration as a function of cell node for one-dimensional flow with a constant velocity field and low dispersion. Shown are the analytical solution (lowest graph), ELLAM results using $CELDIS = 1$ (121 time steps), $NSC = 32$, $NSR = NSL = 2$, $NT = 128$, and ELLAM results using $CELDIS = 10.1$ (12 time steps), $NSC = 4$, $NSR = NSL = 2$, $NT = 128$ (upper graph). Results for $CELDIS = 1$ are virtually identical to the analytical.

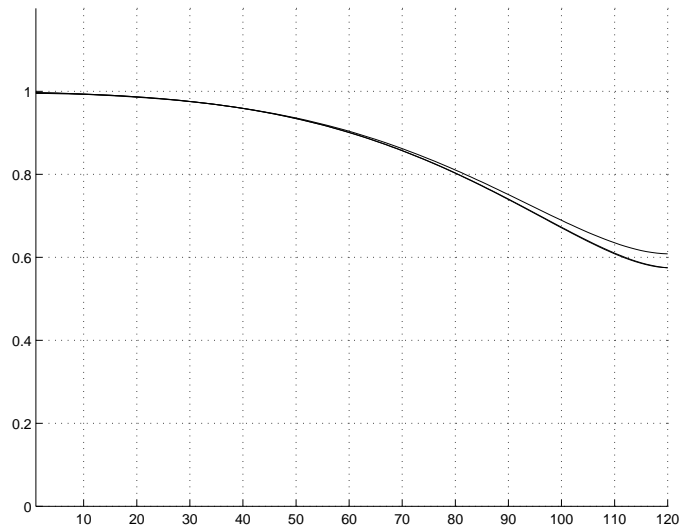


Figure 1.8. Plots of concentration as a function of cell node for one-dimensional flow with a constant velocity field and high dispersion. Shown are the analytical solution (lowest graph), ELLAM results using $\text{CELDIS} = 1$ (121 time steps), $\text{NSC} = 32$, $\text{NSR} = \text{NSL} = 2$, $\text{NT} = 128$, and ELLAM results using $\text{CELDIS} = 10.1$ (12 time steps), $\text{NSC} = 4$, $\text{NSR} = \text{NSL} = 2$, $\text{NT} = 128$ (upper graph). Results using $\text{CELDIS} = 1$ are virtually identical to the analytical.

Various other tests were performed, including runs with a retardation factor of other than one. Concentration profiles in space at various times were also evaluated. Results comparable to the above were obtained. Even in the extreme case shown in figure 1.9 of $CELDIS = 61$ (two time steps), $NSC = 4$, $NSR = NSL = 2$, $NT = 128$, qualitatively good results were calculated, except near the outflow face. This demonstrates the apparent robustness of the method.

The low dispersion, no sorption problem was also used to test the way the method handles nonzero decay. Figure 1.10 exhibits results for $\lambda = 0.01 \text{ sec}^{-1}$, $CELDIS = 1$, $NSC = 32$, $NSR = NSL = 2$, $NT = 128$ which are in excellent agreement with the analytical solution. As in the case of no decay, $NSC = 4$ (not shown) yields a slightly low concentration near the inflow boundary.

In all cases described above, the mass balance error was less than 0.001 percent, in contrast to method-of-characteristics code, which is not mass conservative. Explicit and implicit MOC3D, for example, yielded mass balance errors of up to a few percent in some cases reported above.

1.5.2 Uniform, Three-Dimensional Flow

Next, ELLAM results were compared with the analytical solution developed by Wexler [99] for three-dimensional solute transport from a continuous point source in a steady, uniform flow field in a homogeneous aquifer of infinite extent. The problem and analytical solution are described in detail by Konikow and others in [71], and specific parameter values for the test case are shown in figure 1.5.2.

Since the flow velocity is aligned with the grid and dispersive flux cross-terms are zero, this problem provides a test of the accuracy of dispersive flux calculations in three directions [71]. It also offers a test of the source mass algorithm, and its use in representing the effects of a specified flux boundary condition.

Analytical results are shown in figure 1.11, and ELLAM results are plotted in figures 1.12, 1.13, and 1.14 for the xy plane passing through the

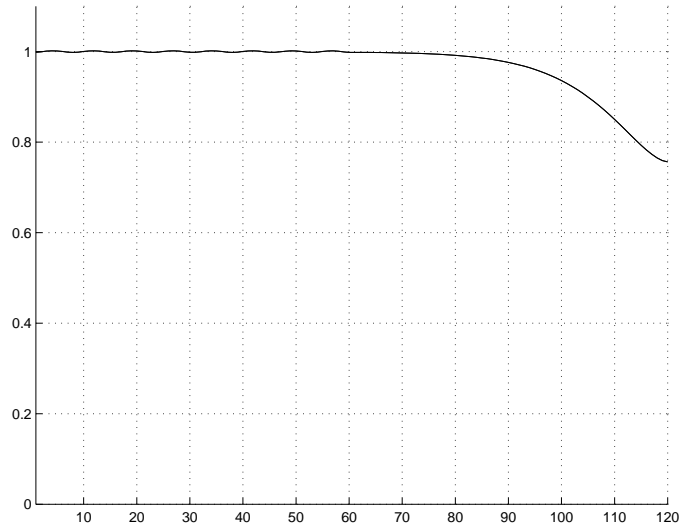


Figure 1.9. Concentration vs. cell node plot with CELDIS = 61 (two time steps), NSC = 4, NSR = NSL = 2, NT = 128.

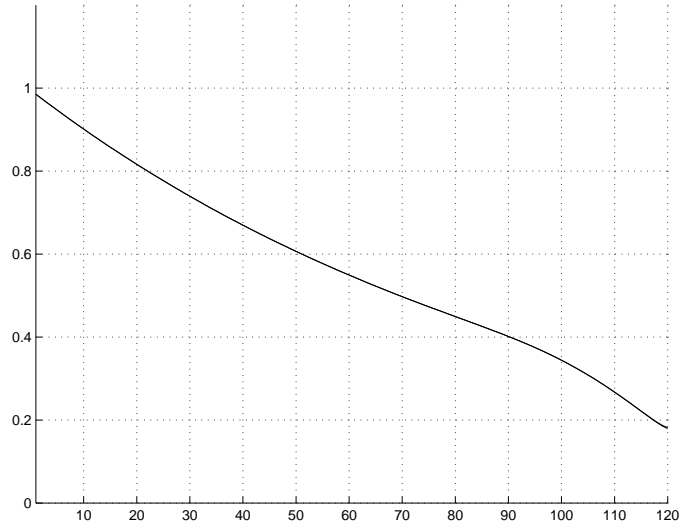


Figure 1.10. Plots of concentration as a function of cell node for decay constant $\lambda = 0.01 \text{ sec}^{-1}$. Shown are the analytical solution (lower graph) and ELLAM results using CELDIS = 1 (121 time steps), NSC = 32, NSR = NSL = 2, NT = 128.

Parameter	Value
$T_{xx} = T_{yy}$	0.0125 m ² /day
ε	0.25
α_L	0.6 m
α_{TH}	0.03 m
α_{TV}	0.006 m
PERLEN (length of stress period)	400 days
V_x	0.1 m/day
$V_y = V_z$	0.0 m/day
Initial concentration (C_0)	0.0
Source concentration (C')	2.5106 g/m ³
Q (at well)	1.010-6 m ³ /d
Source location	row 8, column 1, layer 11
Number of rows	30
Number of columns	12
Number of layers	40
DELR (Δx)	0.5 m
DELC (Δy)	3 m
Thickness (b)	0.05 m

Table 1.2. Parameters used in ELLAM simulation of transport from a continuous point source in a three-dimensional, uniform, steady-state flow system

point source. For 1.12, CELDIS = 7 (two time steps), NSC = NSR = NSL = 4, NT = 16; for 1.13, CELDIS = 1 (14 time steps), NSC = NSR = NSL = 4, NT = 4; and for 1.14, CELDIS = 0.1 (134 time steps), NSC = NSL = 4, NSR = 8, NT = 16. MOC3D reports, [71] and [70] discuss the greater spreading in numerical models than the analytical solution, and note that this is partially explained by the application of the source over an entire grid cell in the models, while the analytical solution portrays a point source.

Figure 1.12 can be interpreted as demonstrating that more than two time steps are needed to adequately resolve dispersion. As seen in figure 1.13, ELLAM results with 14 time steps accurately characterize the dispersive flux, without the spreading upstream that is produced by MOC3D. With 134 time increments, ELLAM produces yet less upstream spreading; so little, however, that the solution manifests the numerical oscillations typical where concentration gradient is too steep relative to grid mesh density.

Graphs in vertical planes parallel and perpendicular to the flow direction are comparable to the above, in terms of the proximity of the ELLAM results to the analytical.

1.5.3 Two-Dimensional Radial Flow

The ELLAM solution was compared to the analytical solution give by Hsieh [61] to a radial flow and dispersion problem with a finite-radius well in an infinite aquifer of two-dimensions. There is flow from a single injection well, with velocities inversely related to distance from the well.

The parameters for the problem are given in table 1.5.3. More details about the problem are presented in [71] and [70]. Due to symmetry, the problem can be modelled on one quadrant of the radial flow field.

The analytical solution is plotted in 1.15, along with four ELLAM solutions in figures 1.16- 1.19. Each ELLAM solution captures the salient qualitative features of the analytical solution, with the exception of the high concentration contour of 1.16, which has an articulated rather than a smooth shape.

The two time step run graphed in figure 1.16 uses CELDIS = 75,

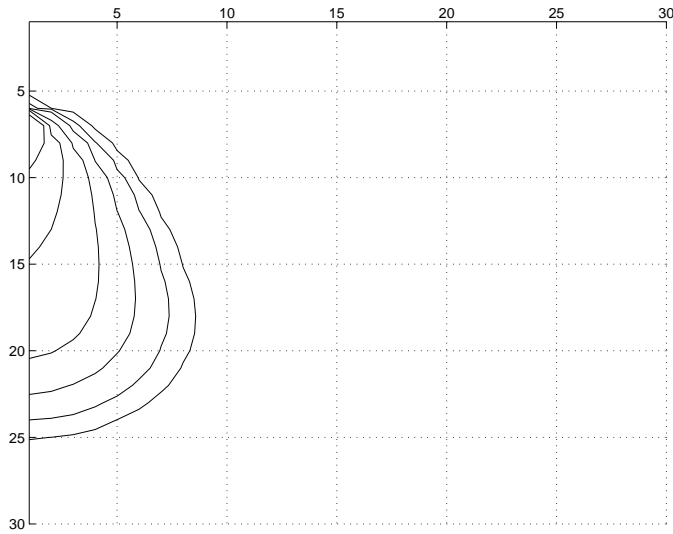


Figure 1.11. Concentration contours of analytical solution in the horizontal plane containing the solute source (layer 1) for three-dimensional solute transport in a uniform steady flow problem.

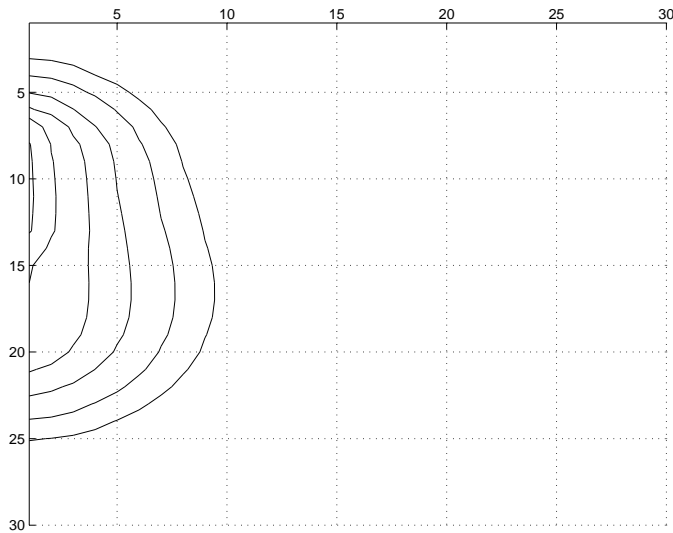


Figure 1.12. Concentration contours in the horizontal plane containing the solute source (layer 1) for three-dimensional solute transport in a uniform steady flow problem with $CELDIS = 7$ (two time steps), $NSC = NSR = NSL = 4$, $NT = 16$.



Figure 1.13. Concentration contours in the horizontal plane containing the solute source (layer 1) for three-dimensional solute transport in a uniform steady flow problem with $CELDIS = 1$ (14 time steps), $NSC = NSR = NSL = 4$, $NT = 4$.

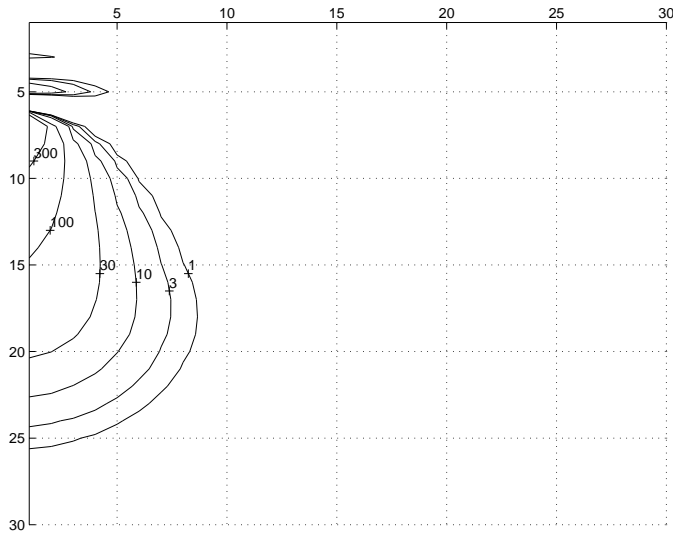


Figure 1.14. Concentration contours in the horizontal plane containing the solute source (layer 1) for three-dimensional solute transport in a uniform steady flow problem with $CELDIS = 0.1$ (134 time steps), $NSC = NSL = 4$, $NSR = 8$, $NT = 16$.

Parameter	Value
$T_{xx} = T_{yy}$	3.6 m ² /hour
ε	0.2
α_L	10.0 m
$\alpha_{TH} = \alpha_{TV}$	10.0 m
PERLEN (length of stress period)	1000 hours
Q (at well)	56.25 m ³ /hour
Source concentration (C^0)	1.0
Number of rows	30
Number of columns	30
Number of layers	1
DELR (Δx) = DELC (Δy)	10.0 m
Thickness (b)	10.0 m

Table 1.3. Parameters used in ELLAM simulation of two-dimensional, steady-state, radial flow case.

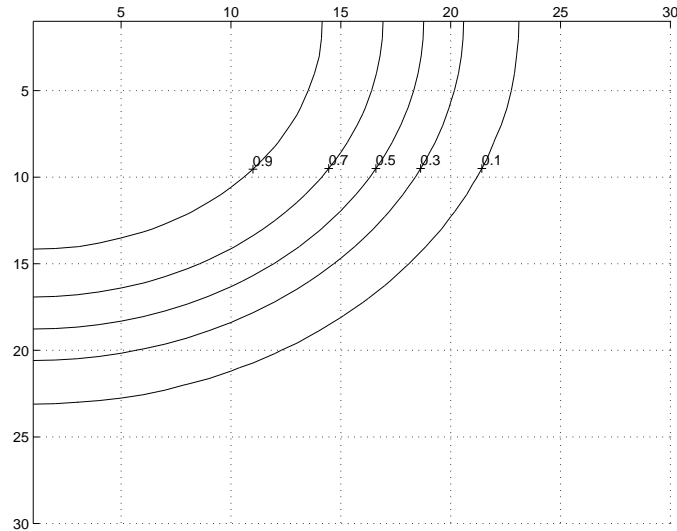


Figure 1.15. Contour plot of analytical solution for two dimensional radial flow problem.

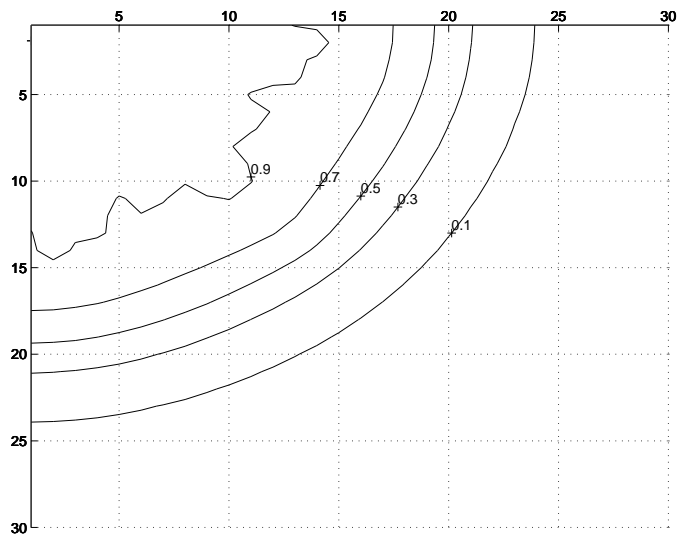


Figure 1.16. Contour plot of concentration for run with two time steps using $\text{CELDIS} = 75$, $\text{NSC} = \text{NSR} = 4$, $\text{NSL} = 2$, $\text{NT} = 16$.

$NSC = NSR = 4$, $NSL = 2$, $NT = 16$. The high concentration contour has non-physical oscillations, which cannot be entirely eliminated even with higher NS and NT values. The 29 time step run shown in figure 1.17 uses CELDIS = 5, $NSC = NSR = 4$, $NSL = 2$, $NT = 4$. Although the high concentration contour is somewhat flat, it is a close approximation to the analytical solution.

The advisability of using higher NS values when modeling with many time steps is illustrated in figures 1.18 and 1.19, where 563 time steps are used. Parameter values $NSC = NSR = 4$, $NSL = 2$, $NT = 4$ produce a rather flat set of contours, whereas $NSC = NSR = 8$, $NSL = 2$, $NT = 4$ yield a much better match to the analytical solution.

Both versions of MOC3D produce close to analytical results for this problem. For comparison, explicit MOC3D needed 596 time steps and 12 min 30 s of cpu time; implicit MOC3D used 282 time steps and 7 min 25 s of cpu time; and ELLAM runs discussed above needed 2 min 18 s, 33 min 10 s, and 96 min 20 s, respectively, all on a UNIX workstation [57].

1.5.4 Initial Condition in Uniform Flow

A problem of three-dimensional solute transport from an instantaneous point source in a flow field with uniform velocity was also considered. Wexler [99] presents an analytical solution for a continuous point source in a homogeneous aquifer of infinite extent, which was modified to yield quantitative analytical results for the instantaneous problem.

This initial condition is represented numerically by a nonzero concentration in a single finite-difference cell. For MOC3D, this signifies a constant concentration distributed evenly across a cell. For ELLAM, with its assumption of concentrations varying piecewise trilinearly between cell nodes, this signifies initial mass in 27 cells. Furthermore, ELLAM is known to be ill-equipped to handle advection of fronts discretized with fewer than four nodes (see section 1.6.5), so that ELLAM is expected to have difficulty propagating even this dispersed point source.

However, a nonzero initial concentration in one cell, with a uniform

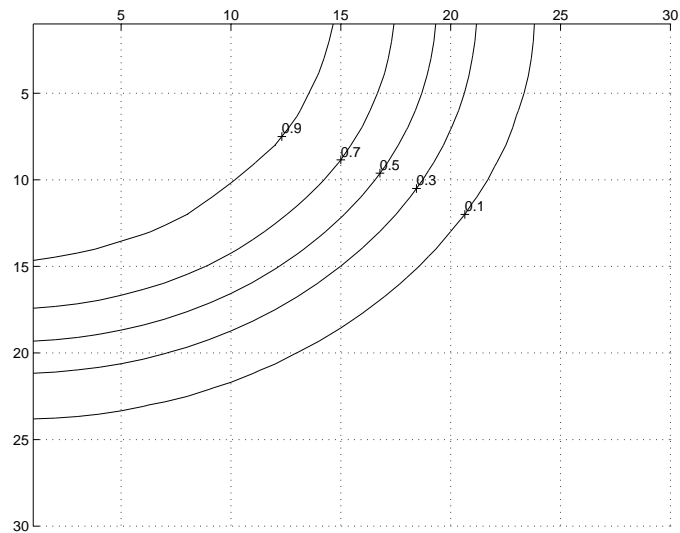


Figure 1.17. Contour plot of concentration for run with 29 time steps using $CELDIS = 5$, $NSC = NSR = 4$, $NSL = 2$, $NT = 4$.

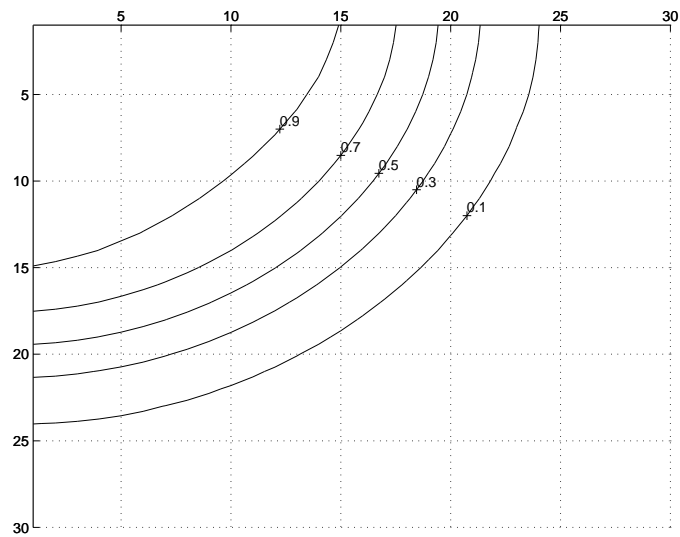


Figure 1.18. Contour plot of concentration for run with 563 time steps using $CELDIS = 0.25$, $NSC = NSR = 4$, $NSL = 2$, $NT = 4$. Concentration maximum is 1.019.

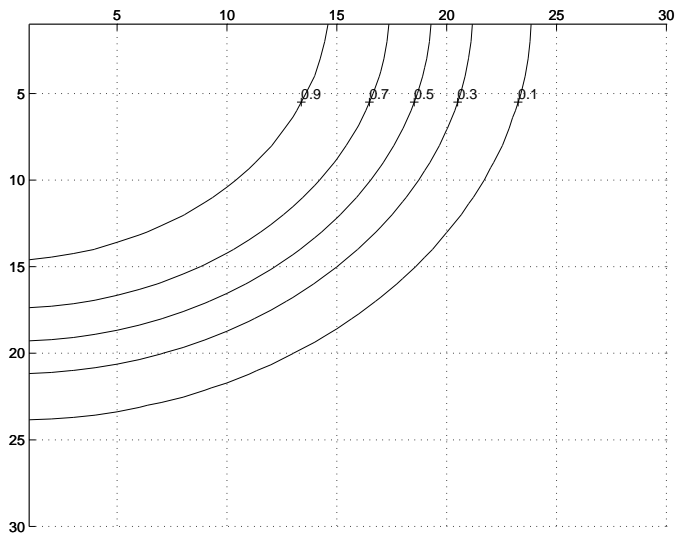


Figure 1.19. Contour plot of concentration for run with 563 time steps using $CELDIS = 5$, $NSC = NSR = 8$, $NSL = 2$, $NT = 4$. Concentration maximum is 1.0056.

velocity field, has been tested, and results are compared to the analytical solution of a problem with an instantaneous point source. Also, to better test the ELLAM code, this analytical solution has been allowed to develop over time, and the smoothed concentration distribution is used as an initial condition for ELLAM. The uniform flow problem with this second initial condition, is also solved with two different velocity fields to test for grid orientation effects on the dispersion calculations. Parameters for the problem are given in table 1.5.4.

For the case of flow in the x direction only, $V_x = 1.0275$, and $V_y = V_z = 0$. For flow at 45 degrees to the grid, $V_x = V_y = 1.0275$, and $V_z = 0$. So the distance the center of the plume moves in the x direction is the same for both cases, for equal simulation times, but the magnitude of the velocity is greater for the flow at 45 degrees, introducing more dispersion. For all runs reported below, CELDIS = 5 (six time steps), NSC = NSR = NSL = 4, NT = 2. All graphs are concentration contours in the plane of the initial source of solute, and of movement. In figure 1.20 the analytical solution using an initial point source is plotted, and in figure 1.21, the ELLAM solution. The numerical results show some spreading relative to the analytical, in both the transverse and longitudinal directions. Increasing the number of time steps does not completely eliminate the spreading, and causes some loss of peak concentration, even with increased NS values. In contrast to both versions of MOC3D, ELLAM results exhibit the symmetry of the analytical solution. In figure 1.22 the analytical solution using an initial dispersed source is plotted, and in figure 1.23, the ELLAM solution. These solutions resemble each other more closely, although the ELLAM solution is still somewhat dispersed.

The analogous results with a dispersed initial condition are plotted for flow at 45 degrees to the grid. In figure 1.24 is the analytical solution, and in figure 1.25 the ELLAM solution is plotted. The characteristic symmetry of the analytic solution is captured by ELLAM, but there is longitudinal spreading, and some distortion of the shape of the plume. This narrowing is characteristic of a grid-orientation effect caused primarily by off-diagonal terms of the dispersion tensor.

Parameter	Value
$T_{xx} = T_{yy}$	10.0 m ² /day
ε	0.1
α_L	1.0 m
$\alpha_{TH} = \alpha_{TV}$	0.1 m
PERLEN (length of stress period)	90 days
V_x	1.0 m/day
$V_y = V_z$	0.0 m/day*
Initial concentration at source	1 x 10 ⁶
Source location	column 11,
in transport	row 36,
subgrid	layer 4
Number of rows	72
Number of columns	72
Number of layers	24
DELR (Δx)	3.33 m
DELC (Δy)	3.33 m
Thickness (b)	10.0 m

* For flow at 45 degrees to x - and y - axes, $V_y = 1.0275\text{m/day}$.

Table 1.4. Parameters used in ELLAM simulation of three-dimensional transport from a point source with flow in the x-direction and flow at 45 degrees to x- and y-axes.

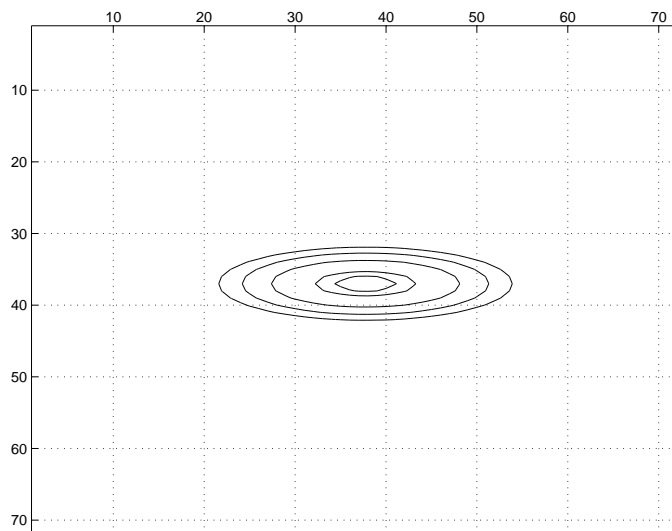


Figure 1.20. Contour plot showing log of concentrations in analytical solution of Dirac problem at $t = 90$. Concentration maximum is 25195.

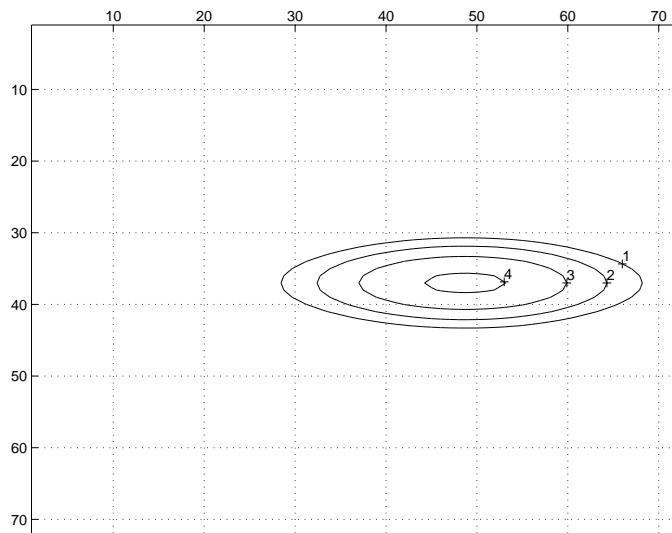


Figure 1.21. Contour plot showing log of concentrations for spike initial condition, and $CELDIS = 5$, $NSC = NSR = NSL = 4$, $NT = 2$. Concentration maximum is 15160.

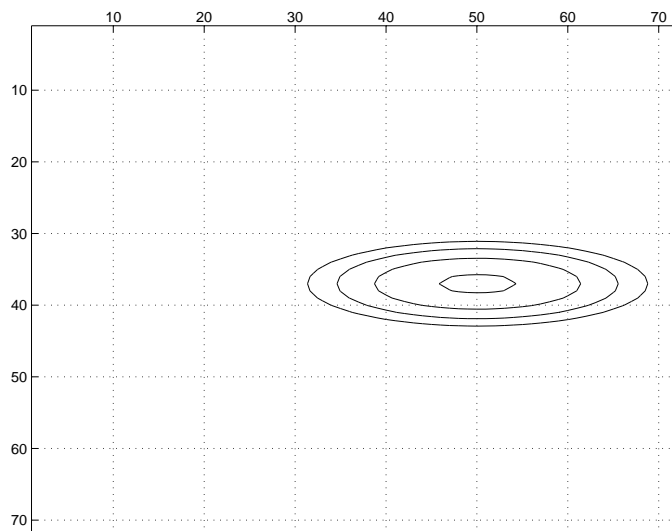


Figure 1.22. Contour plot showing log of concentrations in analytical solution of Dirac problem at $t = 130$. Concentration maximum is 14539.

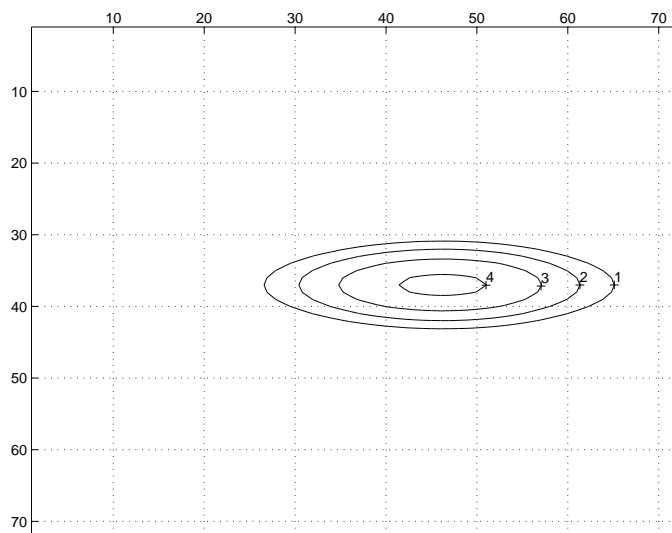


Figure 1.23. Contour plot showing log of concentrations for dispersed initial condition, and $CELDIS = 5$, $NSC = NSR = NSL = 4$, $NT = 2$. Concentration maximum is 16909.

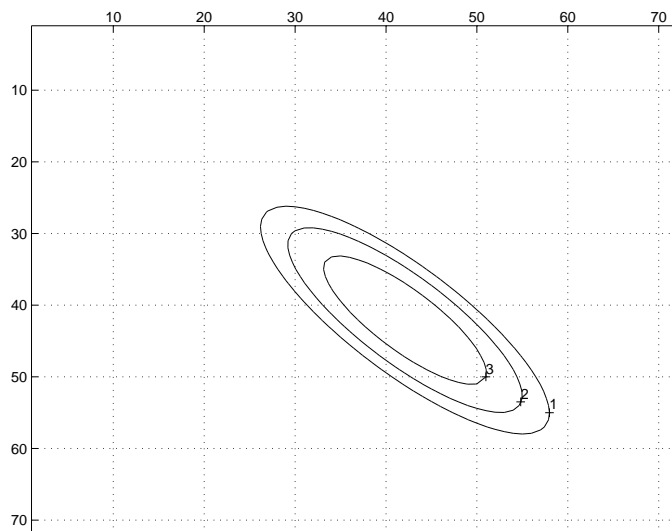


Figure 1.24. Contour plot showing log of concentrations of analytical solution to Dirac problem at $t = 130$. Concentration maximum is 8645.

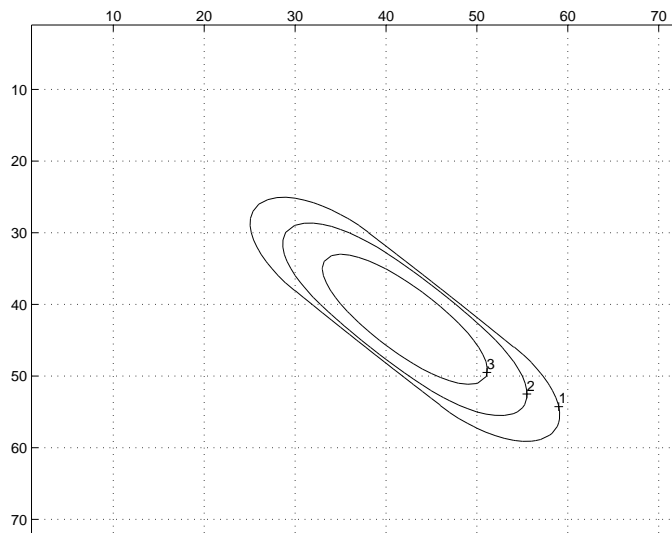


Figure 1.25. Contour plot showing log of concentrations for dispersed initial condition, and $CELDIS = 5$, $NSC = NSR = NSL = 4$, $NT = 2$ at $t = 130$. Concentration maximum is 8167.

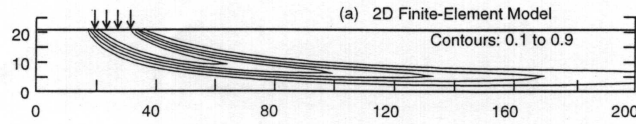


Figure 1.26. Contour plot showing concentrations of a two-dimensional finite element solution of the Burnett and Frind problem. Contours shown are 0.1 to 0.9.

1.5.5 Constant Source in Nonuniform Flow

A problem having a constant source of solute over a finite area at the surface of a homogeneous aquifer with boundary conditions producing nonuniform flow was introduced by Burnett and Frind [21]. They used an alternating-direction Galerkin finite element method to solve the flow and transport equations in two and three dimensions. This Burnett and Frind problem has been used as a test case to evaluate both versions of MOC3D, with the finite element results as the benchmark. It is a complex problem, including a variable dispersion coefficient. Konikow and others [71] present a detailed description of the problem geometry including a discussion of the differences between the finite element grid and the cell-centered finite-difference grid. Parameters used for the ELLAM simulation are given in table 1.5.5.

The ELLAM simulator was used on the two dimensional case, using $CELDIS = 30$ (seven time steps), $NSC = NSR = NSL = 4$, $NT = 32$. The plume calculated by Burnett and Frind is shown in figure 1.26, and the ELLAM results in figure 1.27. The contours are very similar, although the high concentration contour from ELLAM doesn't extend as far downgradient as that of Burnett and Frind, while the lowest concentration curve extends further. Nevertheless, the ELLAM solution provides a closer match to the Burnett and Frind contours than do MOC3D results using 381, 1901, or 4218 time steps [70]. Increasing the number of time increments yields a solution with a greater downgradient extent, but still short of the Burnett and Frind results.

Fourteen rows were added to the two-dimensional grid in order to expand it to three dimensions. Transport results for a vertical plane at the

Parameter	Value
K	1.0 m/day
ε	0.35
α_L	3.0 m
α_{TH}	0.10 m
α_{TV}	0.01 m
PERLEN (length of stress period)	12,000 days
Q (at well)	56.25 m ³ /hour
Source concentration(C')	1.0
Number of rows	1
Number of columns ¹	141
Number of layers ¹	91
DELR (Δx)	1.425 m
DELC (Δy)	1.0 m
Thickness (b)	0.2222-0.2333 m

¹ One row and layer were allocated to defining boundary conditions, so concentrations calculated in only 140 columns and 90 layers were used for comparison.

Table 1.5. Parameters used for ELLAM simulation of transport in a vertical plane from a continuous point source in a nonuniform, steady-state, two-dimensional flow system (described by Burnett and Frind, 1987).

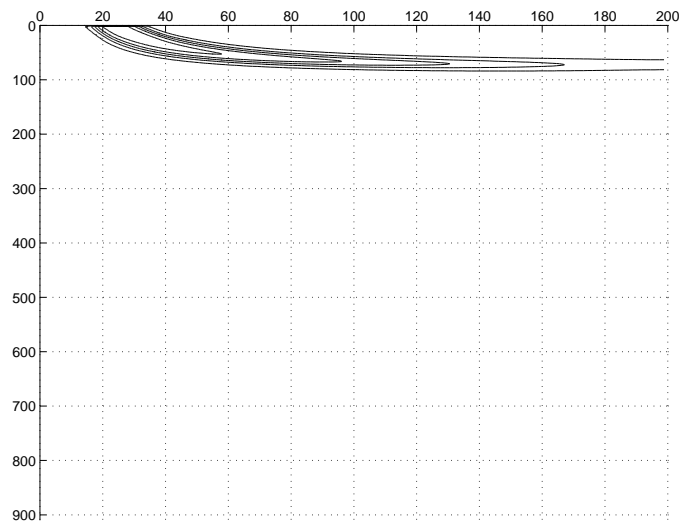


Figure 1.27. Contour plot showing concentrations of ELLAM solution to two-dimensional Burnett and Frind problem using $CELDIS = 30$, $NSC = NSR = NSL = 4$, $NT = 32$. Contours shown are 0.1 to 0.9.

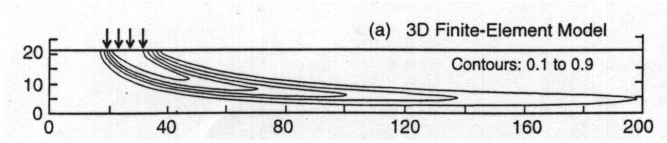


Figure 1.28. Contour plot showing concentrations of a three-dimensional finite element solution of the Burnett and Frind problem. Contours shown are 0.1 to 0.9.

middle of the plume are plotted for runs with lower and higher dispersion. Figure 1.28 shows Burnett and Frind results for the low dispersion case where $\alpha_{TV} = 0.01m$ and $\alpha_{TH} = 0.1m$, while ELLAM are given in figure 1.29. The ELLAM plume extends slightly further downstream with low concentrations of solute than the Burnett and Frind. Again, the ELLAM solution is a closer match to the Burnett and Frind than any MOC3D solution.

With vertical transverse dispersivity increased by a factor of ten so that $\alpha_{TV} = \alpha_{TH} = 0.1m$, the results of simulation were again compared. The ELLAM solution using CELDIS = 30 had noticeably low concentrations near the source. Plotted in figures 1.30 and 1.31 are the Burnett and Frind solution for higher dispersion, and an ELLAM solution using CELDIS = 21 (10 time steps), respectively. The contours are in close agreement.

1.6 One-Dimensional ELLAM

Consider a one-dimensional problem with $\varepsilon = R_f = 1$, and $D \geq 0$, and assume, for simplicity, a uniform grid and $v = constant > 0$. Also, forego the use of "approximate test functions," that is, let

$$w_{jik}(\hat{x}, \hat{y}, \hat{z}) = f(\hat{x})g(\hat{y})h(\hat{z})$$

where

$$f(x) = \begin{cases} 0 & x \leq -\frac{1}{2} \\ 1 & -\frac{1}{2} < x \leq \frac{1}{2} \\ 0 & \frac{1}{2} < x \end{cases} ,$$

and $g \equiv h \equiv 1$.

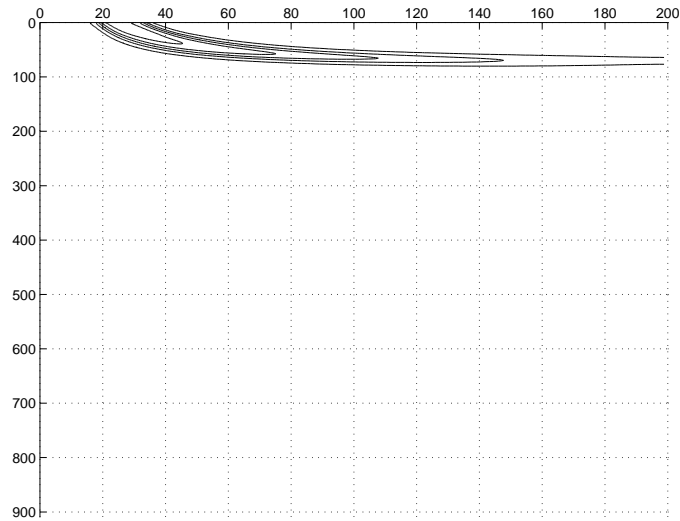


Figure 1.29. Contour plot showing concentrations of ELLAM solution to low dispersion Burnett and Frind problem using $CELDIS = 30$, $NSC = NSR = NSL = 4$, $NT = 32$. Contours shown are 0.1 to 0.9.

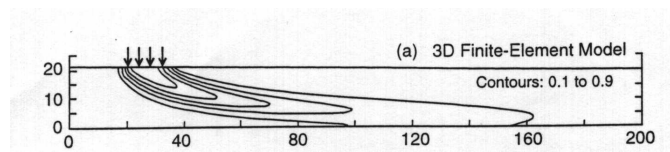


Figure 1.30. Contour plot showing concentrations of a three-dimensional finite element solution of the Burnett and Frind problem with high vertical transverse dispersivity. Contours shown are 0.1 to 0.9.

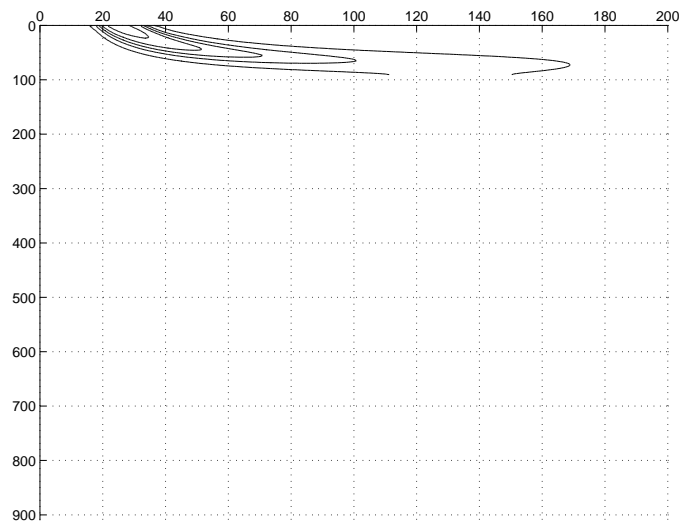


Figure 1.31. Contour plot showing concentrations of ELLAM solution to high dispersion Burnett and Frind problem using $CELDIS = 21$, $NSC = NSR = NSL = 4$, $NT = 32$. Contours shown are 0.1 to 0.9.

Neglecting sources and inflow, which do not affect stability, the system of ELLAM cell equations can be expressed

$$(A + M)\mathbf{c}^{n+1} = B\mathbf{c}^n \quad (1.25)$$

where A is the coefficient matrix for integration of concentration at the new time level; M is the matrix of diffusion coefficients obtained from a backward Euler in time and centered differences in space approximation to the diffusion term of the continuous equation; B is the coefficient matrix for concentration from the old time level, incorporating an integration to determine mass in a cell, followed by a shift operation to represent advection; and \mathbf{c} is the vector of nodal concentration values at the time level denoted by the superscript. Due to the assumption of a uniform grid, dependence of the numerical integrals on cell length has been divided out.

Using the trapezoidal rule to integrate the unknown piecewise linear concentration function, and using zero Dirichlet boundary conditions for simplicity, the A matrix is tridiagonal with

$$A = \frac{1}{8}[1, 6, 1] \quad (1.26)$$

denoting the sub-, main, and super-diagonal entries, respectively.

The matrix M of centered difference coefficients at the new time level is

$$M = \Delta t k[-1, 2, -1], \quad (1.27)$$

where for diffusion coefficient $D > 0$, $k = \frac{D\Delta t}{(\Delta x)^2}$. Note that $A + M$ is symmetric positive definite for all $k \geq 0$, so $(A + M)^{-1}$ exists.

The matrix B depends on Courant number, defined by $Cr = \frac{v\Delta x}{\Delta t}$. The matrix considered here is for a constant Courant number, $0 \leq Cr \leq \frac{1}{2}$. To construct a B which apportions the correct mass to each cell at t^{n+1} , integrate exactly at the old time level using integration points at cell centers and pre-images at the old time level of cell boundaries at the new time, as shown in figure 1.32. Denoting by x^* the position at the old time level of a point x at

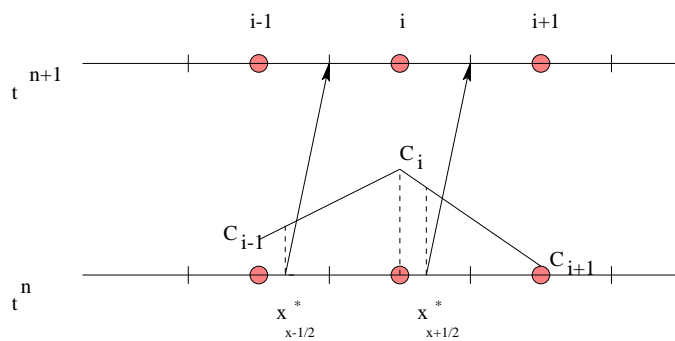


Figure 1.32. Mass at old time level is integrated exactly using cell centers and pre-images of cell boundaries as integration points, and applying the trapezoidal integration rule.

t^{n+1} , and interpolating concentration values between nodes, the trapezoid on the left has area,

$$\left(\frac{\Delta x}{2} + Cr\Delta x\right) \left(\frac{C(x_{i-\frac{1}{2}}^*) + C_i}{2}\right),$$

and the area on the right is,

$$\left(\frac{\Delta x}{2} - Cr\Delta x\right) \left(\frac{C_i + C(x_{i+\frac{1}{2}}^*)}{2}\right).$$

Since

$$C(x_{i-\frac{1}{2}}^*) = \left(\frac{1}{2} + Cr\right) C_{i-1} + \left(\frac{1}{2} - Cr\right) C_i$$

and

$$C(x_{i+\frac{1}{2}}^*) = \left(\frac{1}{2} + Cr\right) C_i + \left(\frac{1}{2} - Cr\right) C_{i+1},$$

the mass in cell i at the new time level is

$$\begin{aligned} \Delta x & \left[\left(\frac{1}{2} + Cr\right) \left(\frac{1}{4} + \frac{Cr}{2}\right) C_{i-1} \right. \\ & + \left(\left(\frac{1}{2} + Cr\right) \left(\frac{3}{4} - \frac{Cr}{2}\right) + \left(\frac{1}{2} - Cr\right) \left(\frac{3}{4} + \frac{Cr}{2}\right) \right) C_i \\ & \left. + \left(\left(\frac{1}{2} - Cr\right) \left(\frac{1}{4} - \frac{Cr}{2}\right) C_{i+1} \right] \right]. \end{aligned}$$

Thus, disregarding boundary conditions, the B matrix is tridiagonal with

$$B = \frac{1}{8}[1, 6, 1] + \frac{Cr}{2}[1, 0, -1] + \frac{Cr^2}{2}[1, -2, 1], \quad (1.28)$$

denoting the sub-, main, and super-diagonal entries, respectively. In the case of $Cr = \frac{1}{2}$, B is singular. The more general case of arbitrary, still constant, Courant number is considered analogously: the Courant number is decomposed into the sum the nearest integer, and a fractional part with $|fractional\ part| \leq \frac{1}{2}$. In (1.28), Cr in the second two terms is replaced by the fractional part, and the entire matrix sum is diagonal-shifted by minus the integer nearest to the Courant number. For simplicity, only the case $0 \leq Cr \leq \frac{1}{2}$ is treated here.

The integration points and rule used in the above construction differ from those in the three-dimensional ELLAM implementation. Application of the three-dimensional code to a one-dimensional problem on a uniform grid

with Courant number a multiple of $\frac{1}{2^n}$, $n = 1, 2, 3, \dots$ likewise tracks and integrates advected mass exactly, and yields a solution identical to that produced by $A^{-1}B$, up to treatment of boundary conditions.

Convergence of the method (1.25) will follow from its consistency and stability. The exact solution, $\mathbf{c}(\cdot, t)$ satisfies

$$(A + M)\mathbf{c}(\cdot, t^{n+1}) = B\mathbf{c}(\cdot, t^n) + \Delta t\mathbf{e}_c, \quad (1.29)$$

where \mathbf{e}_c denotes the vector of truncation (consistency) errors at each node. Letting \mathbf{e}^n denote the error $\mathbf{c}^n - \mathbf{c}(\cdot, t^n)$, and subtracting (1.29) from (1.25), we have the error propagation

$$(A + M)\mathbf{e}^{n+1} = B\mathbf{e}^n + \Delta t\mathbf{e}_c.$$

Because $A + M - \frac{1}{2}I$ is nonnegative definite, this implies

$$\mathbf{e}^{n+1} = (A + M)^{-1}B\mathbf{e}^n + \Delta t\mathbf{e}_c,$$

with a different constant incorporated into \mathbf{e}_c . This leads to

$$\mathbf{e}^{n+1} = \sum_{j=0}^n ((A + M)^{-1}B)^j \Delta t\mathbf{e}_c + ((A + M)^{-1}B)^{n+1}\mathbf{e}^0,$$

where \mathbf{e}^0 denotes any initial error. For stability we then require,

$$\|((A + M)^{-1}B)^n\| \leq K$$

in some norm, for $0 \leq n\Delta t \leq T$, where the constant K is independent of Δt , and T is simulation time. Then $\|\mathbf{e}^{n+1}\| \leq O(K)\|\mathbf{e}_c\|$, and consistency yields convergence as $\Delta x, \Delta t \rightarrow 0$.

Stability depends on the properties of the matrix $(A + M)^{-1}B$ which operates explicitly upon a error vector at the old time level to yield error at the new time.

1.6.1 Limiting Case of No Advection

In the limiting case of nonzero diffusion, but no advection,

$$A + M = \left[\frac{1}{8} - k, \frac{6}{8} + 2k, \frac{1}{8} - k\right]. \quad (1.30)$$

The right-hand side matrix B becomes

$$B = \frac{1}{8}[1, 6, 1], \quad (1.31)$$

since with no advection, $Cr = 0$.

Lemma 1.1 In the limiting case of no advection, the method given by equation (1.25), using (1.30), and (1.31) is consistent, with error of order,

$$\Delta t c_e = O(\Delta x^2 \Delta t + \Delta t^2).$$

Proof: We use standard finite difference arguments.

First note that integrating the Taylor expansion of a function f gives,

$$\begin{aligned} & \int_{x_{i-1/2}}^{x_{i+1/2}} f(x) dx \\ &= \int_{x_{i-1/2}}^{x_{i+1/2}} f(x_i) + f'(x_i)(x - x_i) + \frac{1}{2}f''(x_i)(x - x_i)^2 + O((x - x_i)^3) dx \\ &= \Delta x f(x_i) + 0 + \frac{1}{2}f''(x_i) \int_{x_{i-1/2}}^{x_{i+1/2}} (x - x_i)^2 dx + 0 \\ &= \Delta x f(x_i) + \frac{\Delta x^3}{8} f''(x_i) + O(\Delta x^3), \end{aligned}$$

where it is useful to separate the $\frac{\Delta x^3}{8}$ and $O(\Delta x^3)$ terms. Then, we have

$$\begin{aligned} & \int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} c_t(x, t) dx dt \\ &= \Delta x \int_{t^n}^{t^{n+1}} c_t(x_i, t) + \frac{\Delta x^2}{8} \frac{\partial^2}{\partial x^2} c_t(x_i, t) + O(\Delta x^2) dt \\ &= \Delta x \int_{t^n}^{t^{n+1}} c_t(x_i, t) + \frac{\Delta x^2}{8} \left(\frac{c_t(x_{i-1}, t) - 2c_t(x_i, t) + c_t(x_{i+1}, t)}{\Delta x^2} + O(\Delta x^2) \right) + O(\Delta x^2) dt \\ &= \Delta x \left(c_i^{n+1} - c_i^n + \frac{1}{8} [c_{i-1}^{n+1} - c_{i-1}^n - 2(c_i^{n+1} - c_i^n) + c_{i+1}^{n+1} - c_{i+1}^n] \right) \\ & \quad + O(\Delta t \Delta x^4) + O(\Delta t \Delta x^2) \\ &= \Delta x \left(\frac{1}{8}(c_{i-1}^{n+1} - c_{i-1}^n) + \frac{6}{8}(c_i^{n+1} - c_i^n) + \frac{1}{8}(c_{i+1}^{n+1} - c_{i+1}^n) + O(\Delta t \Delta x^2) \right). \end{aligned}$$

Also,

$$\begin{aligned} c_{xx} &= c_{xx}(x, t^{n+1}) - \int_t^{t^{n+1}} c_{xxt}(x, s) ds \\ &= c_{xx}(x_i, t^{n+1}) + c_{xxx}(x_i, t^{n+1})(x - x_i) + O(\Delta x^2) - \int_t^{t^{n+1}} c_{xxt}(x, s) ds \\ &= \frac{c_{i-1}^{n+1} - 2c_i^{n+1} + c_{i+1}^{n+1}}{(\Delta x)^2} + c_{xxx}(x_i, t^{n+1})(x - x_i) + O(\Delta x^2) \\ & \quad - \int_t^{t^{n+1}} c_{xxt}(x, s) ds. \end{aligned}$$

Integrating then gives

$$\begin{aligned} & \int_{x_{i-1/2}}^{x_{i+1/2}} c_{xx}(x, t) dx \\ &= \Delta x \left(\frac{c_{i-1}^{n+1} - 2c_i^{n+1} + c_{i+1}^{n+1}}{(\Delta x)^2} \right) + O(\Delta x^3) \\ & \quad - \int_{x_{i-1/2}}^{x_{i+1/2}} \int_t^{t^{n+1}} c_{xxt}(x, s) ds dx; \end{aligned}$$

and

$$\begin{aligned} & \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{t^n}^{t^{n+1}} -Dc_{xx}(x, t) dt dx \\ &= \Delta x \frac{D\Delta t}{\Delta x^2} (-c_{i-1}^{n+1} + 2c_i^{n+1} - c_{i+1}^{n+1}) + \int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_t^{t^{n+1}} c_{xxt}(x, s) ds dx dt \\ &= \Delta x \frac{D\Delta t}{\Delta x^2} (-c_{i-1}^{n+1} + 2c_i^{n+1} - c_{i+1}^{n+1}) + O(\Delta x^3 \Delta t) + O(\Delta x \Delta t^2). \end{aligned}$$

Combining terms gives the estimate,

$$\begin{aligned} & \int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} c_t(x, t) dx dt - \int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} Dc_{xx}(x, t) dx dt \\ &= \Delta x \left(\frac{1}{8}(c_{i-1}^{n+1} - c_{i-1}^n) + \frac{6}{8}(c_i^{n+1} - c_i^n) + \frac{1}{8}(c_{i+1}^{n+1} - c_{i+1}^n) + O(\Delta t \Delta x^2) \right) \\ & \quad \Delta x \frac{D\Delta t}{\Delta x^2} (-c_{i-1}^{n+1} + 2c_i^{n+1} - c_{i+1}^{n+1}) + O(\Delta x^3 \Delta t) + O(\Delta x \Delta t^2). \end{aligned}$$

Dividing by Δx to reflect (1.25) completes the proof. ■

Lemma 1.2 In the case of no advection, the method given by (1.25) is unconditionally stable in the l^2 norm, independent of the meshsize.

Proof: Write (1.30) as

$$A + M = I + \left(k - \frac{1}{8}\right)[-1, 2, -1]$$

and (1.31) as

$$B = I - \frac{1}{8}[-1, 2, -1]$$

where I is the identity matrix. The matrix $(A + M)^{-1}B$ has eigenvalues

$$\frac{1 - \frac{1}{8}\lambda}{1 + \left(k - \frac{1}{8}\right)\lambda}$$

where λ is an eigenvalue of $[-1, 2, -1]$. Since $0 \leq \lambda \leq 4$, $(A + M)^{-1}B$ has nonnegative eigenvalues less than or equal to one for any $k \geq 0$. Because $(A + M)^{-1}B$ is normal, this gives $\|(A + M)^{-1}B\|_2 \leq 1$. Therefore, the method is stable. \blacksquare

Convergence follows directly from consistency and stability.

Theorem 1.3 In the case of no advection, the method given by (1.25) is convergent, independent of the meshsize.

It is interesting to note that for $k \geq \frac{1}{8}$, $A^{-1}B$ satisfies a discrete maximum principle. For $0 < k < \frac{1}{8}$, however, the operator need not satisfy a maximum principle; consider, for example, 3×3 $A^{-1}B$ with $k = \frac{1}{16}$. Here,

$$\begin{aligned} A^{-1}B &= \frac{4}{679} \begin{pmatrix} \frac{2721}{14} & -14 & 1 \\ -14 & 196 & -14 \\ 1 & -14 & 195 \end{pmatrix} \begin{pmatrix} \frac{6}{8} & \frac{1}{8} & 0 \\ \frac{1}{8} & \frac{6}{8} & \frac{1}{8} \\ 0 & \frac{1}{8} & \frac{6}{8} \end{pmatrix} \\ &= \frac{4}{679} \begin{pmatrix} \frac{16130}{112} & \frac{1559}{112} & -1 \\ \frac{91}{4} & \frac{574}{4} & \frac{56}{4} \\ -1 & \frac{56}{4} & \frac{578}{4} \end{pmatrix} \end{aligned}$$

The absolute row sum of the second row is $\frac{686}{679}$. Since this is greater than 1, the matrix does not satisfy a maximum principle.

1.6.2 Limiting Case of No Diffusion

Consistency analysis for the purely hyperbolic problem has been done by Stynes and Russell [84]. Consistency error of $O((\Delta x)^2 + \Delta t)$ for the interior of the domain, and $O(\Delta x + \Delta t)$ near an inflow boundary is established.

Theorem 1.4 In the case of no diffusion, the method given by (1.25) is stable in the l^2 norm, for any fixed meshsize.

Proof: Since for no diffusion, $k = 0$,

$$\begin{aligned} (A + M)^{-1}B &= A^{-1}B \\ &= I + A^{-1} \left(\frac{Cr}{2}[1, 0, -1] + \frac{Cr^2}{2}[1, -2, 1] \right) \\ &= I + \frac{Cr}{2}A^{-1} ([1, 0, -1] + Cr[1, -2, 1]). \end{aligned}$$

Then, since $0 \leq Cr \leq \frac{1}{2}$,

$$\begin{aligned}
\|(A + M)^{-1}B\|_2 &= \|A^{-1}B\|_2 \\
&\leq 1 + \frac{Cr}{2}\|A^{-1}\|_2 (\|[1, 0, -1]\|_2 + Cr\|[1, -2, 1]\|_2) \\
&\leq 1 + \frac{Cr}{2}\|A^{-1}\|_2(2 + 4Cr) \\
&\leq 1 + 2Cr\|A^{-1}\|_2 \\
&\leq 1 + 4\frac{v\Delta t}{\Delta x} \\
&= 1 + O(\Delta t) \text{ for fixed } \Delta x.
\end{aligned}$$

We have

$$\begin{aligned}
\|(A^{-1}B)^{\frac{T}{\Delta t}}\|_2 &\leq \|A^{-1}B\|_2^{\frac{T}{\Delta t}} \\
&\leq (1 + O(\Delta t))^{\frac{T}{\Delta t}} \\
&\leq e^{\frac{4v}{\Delta x}T},
\end{aligned}$$

where T is the total simulation time. Thus the method is stable for fixed meshsize. ■

Convergence again follows from consistency and stability.

Theorem 1.5 In the case of no diffusion, the method given by (1.25) is convergent, for fixed meshsize.

1.6.3 Infinite Spatial Domain

An infinite spatial domain is used in order to eliminate boundary effects. Here, both the left-hand side (1.26) and right-hand side (1.28) operators are Laurent and $\Delta x > 0$. For an introduction to the theory of Laurent matrices, see [14], which serves as a reference for the following discussion. We will show that in the case of an infinite spatial domain, that $A^{-1}B$ is a Laurent matrix with norm of one, and note properties of its spectrum.

Given a sequence $\{z_{-n}\}_{n=-\infty}^{\infty}$ of complex numbers, a doubly-infinite

matrix with z_{-n} along the n th diagonal is called a Laurent matrix:

$$\left(\begin{array}{cccc|cccc} \dots & \dots & \dots & & \dots & \dots & \dots & \dots \\ \dots & z_0 & z_{-1} & & z_{-2} & z_{-3} & z_{-4} & \dots \\ \dots & z_1 & z_0 & & z_{-1} & z_{-2} & z_{-3} & \dots \\ \hline \dots & z_2 & z_1 & & z_0 & z_{-1} & z_{-2} & \dots \\ \dots & z_3 & z_2 & & z_1 & z_0 & z_{-1} & \dots \\ \dots & z_4 & z_3 & & z_2 & z_1 & z_0 & \dots \\ \dots & \dots & \dots & & \dots & \dots & \dots & \dots \end{array} \right) \quad (1.32)$$

The following theorem is well-known:

Theorem 1.6 (Böttcher and Silberman [14], Theorem 1.1) The Laurent matrix (1.32) generates a bounded linear operator on $l^2(\mathbf{Z})$ if and only if there is a function $z \in L^\infty(\mathbf{T})$ such that $\{z_n\}_{n=-\infty}^\infty$ is the sequence of Fourier coefficients of z :

$$z_n = \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} z(e^{i\theta}) e^{-in\theta} d\theta \quad (n \in \mathbf{Z}),$$

where \mathbf{Z} means the integers, and \mathbf{T} is the complex unit circle.

Should a Laurent matrix indeed represent a bounded linear operator, denote the matrix and the operator by $\mathcal{L}(z)$. Let $L^\infty := L^\infty(\mathbf{T})$ and $L^2 := L^2(\mathbf{T})$.

Laurent matrices represent multiplication operators on L^2 with respect to the orthonormal basis

$$\left\{ \frac{1}{\sqrt{2\pi}} e^{in\theta} \right\}_{n \in \mathbf{Z}}.$$

Define

$$\mathcal{M}(z) : L^2 \rightarrow L^2, \quad f \mapsto zf.$$

This multiplication operator is bounded if and only if $z \in L_\infty$, in which case,

$$\|\mathcal{M}(z)\| = \|z\|_\infty. \quad (1.33)$$

Consider the operator which maps a function to the sequence of its Fourier coefficients,

$$\Phi : L^2 \rightarrow l^2(\mathbf{Z}), \quad f \mapsto \{f_n\}_{n \in \mathbf{Z}}.$$

The operator Φ is bijective and

$$\|\Phi f\|_2 = \|f\|_2 \tag{1.34}$$

for $f \in L^2$. Then

$$\mathcal{L}(z) = \Phi \mathcal{M}(z) \Phi^{-1}, \tag{1.35}$$

and properties of \mathcal{L} can be established by determining the analogous property of \mathcal{M} .

Theorem 1.7 For the case of no diffusion and an infinite spatial domain,

$$A^{-1}B = \mathcal{L}(a^{-1}b), \tag{1.36}$$

with

$$a^{-1}b = \frac{4}{3 + \cos(\theta)} \left(\left(Cr - \frac{1}{2} \right)^2 \cos(\theta) + Cr e^{i\theta} + \frac{3}{4} - Cr^2 \right) \in L^\infty, \tag{1.37}$$

and $\|\mathcal{L}(a^{-1}b)\| = 1$ for any Courant number $Cr \in [0, \frac{1}{2}]$.

Proof: The continuous function $a \in L^\infty(\mathbf{T})$ with Fourier coefficients composing the diagonals of (1.26) is given by

$$\begin{aligned} a(\theta) &= \frac{1}{8\sqrt{2\pi}} (e^{-i\theta} + 6 + e^{i\theta}) \\ &= \frac{1}{4\sqrt{2\pi}} (3 + \cos(\theta)). \end{aligned}$$

The continuous function $b \in L^\infty(\mathbf{T})$ with Fourier coefficients composing the diagonals of (1.28) is given by

$$\begin{aligned} b(\theta) &= \frac{1}{\sqrt{2\pi}} \left[\left(\frac{1}{8} - \frac{Cr}{2} + \frac{Cr^2}{2} \right) e^{-i\theta} + \left(\frac{6}{8} - Cr^2 \right) + \left(\frac{1}{8} + \frac{Cr}{2} + \frac{Cr^2}{2} \right) e^{i\theta} \right] \\ &= \frac{1}{\sqrt{2\pi}} \left[\left(Cr - \frac{1}{2} \right)^2 \cos(\theta) + Cr e^{i\theta} + \frac{3}{4} - Cr^2 \right]. \end{aligned} \tag{1.38}$$

We thus have $\mathcal{L}(a)$ and $\mathcal{L}(b)$. Furthermore, since zero is not in the range of a , the inverse of $\mathcal{L}(a)$ is $\mathcal{L}(a^{-1})$, (see [14], Theorem 1.2), where

$$a^{-1} = \frac{4\sqrt{2\pi}}{3 + \cos(\theta)}. \quad (1.39)$$

From (1.35), it follows that

$$\mathcal{L}(z_1)\mathcal{L}(z_2) = \mathcal{L}(z_1z_2) \quad \text{for all } z_1, z_2 \in L^\infty. \quad (1.40)$$

So $\mathcal{L}(a^{-1})\mathcal{L}(b) = \mathcal{L}(a^{-1}b)$. Multiplying continuous functions (1.39) and (1.38), we get the bounded linear Laurent operator on an infinite domain,

$$A^{-1}B = \mathcal{L}(a^{-1}b),$$

with $a^{-1}b$ given by (1.37).

From (1.35), (1.34), and (1.33), it follows that

$$\|\mathcal{L}(z)\| = \|z\|_\infty \quad \text{for all } z \in L^\infty. \quad (1.41)$$

Thus, to determine the norm of $\mathcal{L}(a^{-1}b)$, we find the maximum modulus of the continuous function $a^{-1}b$. For

$$\begin{aligned} p &= (4Cr^2 - 1)^2, \\ q &= 2(4Cr^2 + 1)(3 - 4Cr^2), \\ r &= 16Cr^4 - 8Cr^2 + 9, \end{aligned}$$

we have

$$|a^{-1}b|^2 = \frac{p \cos^2(\theta) + q \cos(\theta) + r}{(3 + \cos(\theta))^2}.$$

Then

$$\begin{aligned} \frac{\partial |a^{-1}b|^2}{\partial \theta} &= \frac{\sin(\theta)}{(3 + \cos(\theta))^3} [2r - 3q + (q + 6p) \cos(\theta)] \\ &= \frac{\sin(\theta)}{(3 + \cos(\theta))^3} (64Cr^2(2Cr^2 - 1)) (\cos(\theta) - 1) \\ &= 0 \end{aligned}$$

for θ such that $\sin(\theta) = 0$ or $\cos(\theta) = 1$. Thus the critical points for $|a^{-1}b|^2$ are 0 and π when $\theta \in [0, 2\pi)$. By (1.37), for $\theta = 0$,

$$a^{-1}b = 1,$$

and for $\theta = \pi$,

$$a^{-1}b = 1 - 4Cr^2 \in [0, 1].$$

Thus the maximum modulus occurs at $\theta = 0$ and is equal to 1. Then (1.41) gives $\|\mathcal{L}(a^{-1}b)\| = 1$ for any Courant number $Cr \in [0, \frac{1}{2}]$, where the operator norm is that induced by the l^2 norm. ■

For an element z in a Banach algebra, define the *spectrum* of z as the set

$$\text{sp}z := \{\lambda \in \mathbf{C} : z - \lambda e \text{ is not invertible in the algebra}\},$$

where e is the algebra identity element.

For $z \in L^\infty$, further define the *essential range* of z as the set

$$\mathcal{R}(z) := \{\lambda \in \mathbf{C} : \text{meas}(t \in \mathbf{T} : |z(t) - \lambda| < \varepsilon) > 0 \text{ for every } \varepsilon > 0\},$$

where meas means the Lebesgue measure. The essential range of z is the spectrum of z as an element of the Banach algebra L^∞ , and equivalently, the spectrum of the multiplication operator $M(z)$ on L^2 as an element of the Banach algebra of bounded linear operators on L^2 . It then follows from (1.35) that $\text{sp}\mathcal{L}(z) = \mathcal{R}(z)$, whenever $z \in L^\infty$.

We can now consider features of the spectrum of $\mathcal{L}(a^{-1}b)$ as an element of the Banach algebra of bounded linear operators on the complex Banach space l^2 . We note that this definition of spectrum coincides with the usual definition for the spectrum of a linear operator defined on a domain in a complex, normed space (see Kreyszig [72] pg 397).

Since $|a^{-1}b| = 1$, the spectrum of $\mathcal{L}(a^{-1}b)$ is contained in the closed, complex unit disk. Since $a^{-1}b$ is not (globally or locally) a complex constant, $\mathcal{L}(a^{-1}b)$ has no eigenvalues. The entire spectrum is seen to be continuous. Also, $0 \in \text{sp}(a^{-1}b)$ only for a Courant number of $\frac{1}{2}$:

Lemma 1.8 For $a^{-1}b$ given by (1.37) and $Cr \in [0, \frac{1}{2}]$, $0 \in \text{sp}(a^{-1}b)$ if and only if $Cr = \frac{1}{2}$.

Proof: We have $a^{-1}b = 0$ if and only if

$$\begin{aligned} 0 &= \cos(\theta) \left(Cr^2 - Cr + \frac{1}{4} \right) + Cr e^{i\theta} - Cr^2 + \frac{3}{4} \\ &= (\cos(\theta) - 1)Cr^2 + (e^{i\theta} - \cos(\theta))Cr + \frac{1}{4} \cos(\theta) + \frac{3}{4}. \end{aligned} \tag{1.42}$$

For $\theta \neq 0$, using the quadratic theorem,

$$\begin{aligned} Cr &= \frac{\cos(\theta) - e^{i\theta} \pm \sqrt{e^{2i\theta} - 2 \cos(\theta) e^{i\theta} + \cos^2(\theta) - (\cos(\theta) - 1)(\cos(\theta) + 3)}}{2(\cos(\theta) - 1)} \\ &= \frac{\cos(\theta) - e^{i\theta} \pm 2 \sin\left(\frac{\theta}{2}\right)}{-4 \sin^2\left(\frac{\theta}{2}\right)}. \end{aligned}$$

Since Courant number is real, this is satisfied if and only if,

$$i \sin(\theta) = 0,$$

so $\sin(\theta)$ must be identically zero. Since $\theta \in [0, 2\pi)$, this means $\theta = 0$ or $\theta = \pi$. If $\theta = 0$, substitution into (1.42) yields $0 = 1$, hence no solutions. For $\theta = \pi$, we have $Cr = \frac{1}{2}$. ■

1.6.4 Stability Independent of Meshsize

In Section 1.6.2, stability of the method was demonstrated for a fixed meshsize on a finite spatial domain with zero Dirichlet boundary conditions. A bound of $\|A^{-1}B\| \leq 1$ using the l^2 , A , or other norm equivalent to $\|\cdot\|_2$ independent of matrix size would guarantee stability independent of Δx . Shown in table 1.6.4 are MATLAB calculations, for various Courant numbers and problem sizes, of the Euclidean and A norms of $A^{-1}B$. Here, we consider both the tridiagonal $A^{-1}B$, and $A^{-1}B$ constructed with $A_{1,1} = A_{n,n} = \frac{7}{8}$ to reflect Neumann boundary conditions. In all examples, $\|A^{-1}B\|_A$ is bounded by one for the tridiagonal case corresponding to the Dirichlet boundary condition.

Numerical study of the eigenvalues of $A^{-1}B$, with A as in (1.26) and B given by (1.28), suggests the moduli of the eigenvalues of this matrix are bounded by one for any order of the matrix. If this is indeed the case, and the inequality is strict, then the existence of a matrix norm with a value less than or equal to one is assured:

n	$\ A^{-1}B\ $ tridiagonal	$\ A^{-1}B\ $ $A_{1,1} = A_{n,n} = \frac{7}{8}$	$\ A^{-1}B\ _A$ tridiagonal	$\ A^{-1}B\ _A$ $A_{1,1} = A_{n,n} = \frac{7}{8}$
5	0.99798782758721	1.02731655221192	0.99367278911431	1.04572629392131
10	0.99984912202045	1.02646524138853	0.99945791347173	1.04220454113547
50	0.99999971570515	1.02646524324671	0.99999884309114	1.04216137329937
100	0.99999998183770	1.02646524324671	0.99999992482548	1.04216137329937
200	0.99999999885218	1.02646524324671	0.99999999520802	1.04216137329937
300	0.99999999977242	1.02646524324671	0.99999999904715	1.04216137329937
5	0.99911499209546	0.99811477727258	0.99578523861235	0.99365010768407
10	0.99998979236483	0.99994432525461	0.99962924605506	0.99948444126033
50	1.00002196618037	1.00001117880143	0.99999920055886	0.99999912922811
100	1.00002196618549	1.00001117886019	0.99999994803407	0.99999994570318
200	1.00002196618549	1.00001117886019	0.99999999668714	0.99999999661270
300	1.00002196618549	1.00001117886019	0.99999999934125	0.99999999933138
5	1.00154201321822	1.00009454693413	0.99998843721148	0.99996021692627
10	1.00154201699366	1.00009495089198	0.99999896202120	0.99999809688996
50	1.00154201699366	1.00009495089972	0.99999999774353	0.99999999745662
100	1.00154201699366	1.00009495089972	0.99999999985328	0.99999999984424
200	1.00154201699366	1.00009495089972	0.99999999999065	0.99999999999036
300	1.00154201699366	1.00009495089972	0.99999999999814	0.99999999999810
5	1.00023639568756	1.00003687264714	0.99999981047593	0.99999933999554
10	1.00023639647505	1.00003687289972	0.99999998298590	0.99999996868398
50	1.00023639647505	1.00003687289972	0.99999999996301	0.99999999995829
100	1.00023639647505	1.00003687289972	0.99999999999760	0.99999999999745
200	1.00023639647505	1.00003687289972	0.99999999999985	0.99999999999984
300	1.00023639647505	1.00003687289973	0.99999999999997	0.99999999999997
5	1.00006026622643	1.00001031695580	0.99999998815467	0.9999995869556
10	1.00006026665609	1.00001031868571	0.9999999893661	0.9999999804191
50	1.00006026665609	1.00001031868571	0.9999999999769	0.9999999999739
100	1.00006026665608	1.00001031868571	0.9999999999985	0.9999999999984
200	1.00006026665608	1.00001031868571	0.9999999999999	0.9999999999999
300	1.00006026665609	1.00001031868571	1.00000000000000	1.00000000000000
5	0.99799236275199	1.02487950575327	0.99368383249979	1.04147443273890
10	0.99984936308447	1.02391532781825	0.99945833836339	1.03829256968434
50	0.99999971662918	1.02391531969142	0.99999884356986	1.03823646002098
100	0.99999998194544	1.02391531969142	0.99999992485559	1.03823646002098
200	0.99999999886528	1.02391531969141	0.99999999520993	1.03823646002098
300	0.99999999977629	1.02391531969141	0.99999999904753	1.03823646002098

Table 1.6. Sections represent Courant numbers, $Cr = 1/2, 1/3, 1/64, 1/500, 1/2000, 99/200$, respectively.

Lemma 1.9 (Horn and Johnson [60], Lemma 5.6.10) Let A be an $n \times n$ complex matrix and $\varepsilon > 0$ be given. There is a matrix norm $\|\cdot\|$ such that $\varrho(A) \leq \|A\| \leq \varrho(A) + \varepsilon$.

The equivalence of the unknown norm to the l^2 norm may be related to the order of the matrix, however.

Various attempts to prove the bound on the moduli of the eigenvalues have failed: Bounds on the numerical radius have been studied (see Gustafson and Duggirala [49]). The generalized eigenvalue problem, $A\mathbf{x} = \lambda B\mathbf{x}$, has been solved to express an eigenvalue in terms of a component of its (scaled) eigenvector. The inverse matrix of (1.26) has been expressed explicitly (see [73]), reformulated in terms of exponentials, and applied to B of (1.28). In no case have eigenvalue bounds of one been forthcoming.

Sharp bounds on

$$\|A^{-1}B\|_2 = \varrho((A^{-1}B)^T A^{-1}B)$$

and

$$\|A^{-1}B\|_A = \varrho(A^{\frac{1}{2}}B^T A^{-1}BA^{-\frac{1}{2}}) = \varrho((BA^{-1})^T A^{-1}B),$$

where ϱ is the spectral radius, have not been analytically established.

It is easy to see that the matrix equation $A\mathbf{c}^{n+1} = B\mathbf{c}^n$, with A given in (1.26), and B defined by (1.28), results from a forward difference in time, centered difference in space discretization of the equation,

$$C_t + \frac{(\Delta x)^2}{8}C_{xxt} - Cr\frac{\Delta x}{\Delta t}C_x - \frac{Cr^2(\Delta x)^2}{2\Delta t}C_{xx} = 0,$$

where the coefficients are taken to be constant. With the opposite sign on the space-time derivative, this would be an equation of the Sobolev, or pseudo-parabolic, type. Disregarding the C_x term, this continuous operator has eigenvalues greater than 1.

1.6.5 Numerical Dispersion and Oscillations

It is well-known (cf. [83]) that ELLAM methods, in general, require sufficient mesh-density for about four grid nodes on a front for accurate modeling of advected concentration. (Compare with Eulerian methods where number

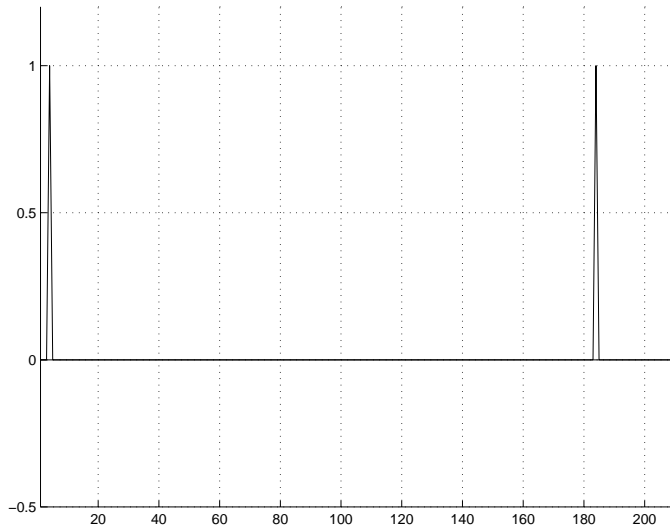


Figure 1.33. Initial and advected peaks with $Cr = 10$.

of nodes on a front must increase as the front sharpens, as discussed in Section 1.1.) Since an arbitrary concentration distribution is a linear combination of chapeau functions, the behavior of an initial condition of nonzero concentration at one node only under repeated application of the no-diffusion operator was studied. Qualitative features of the solution are independent of the problem size. The following represent results for the one-dimensional problem currently under discussion.

In each of figures 1.33, 1.34, 1.35, 1.36, and 1.37, an initial condition of one at node four is plotted. Also plotted is the solution after time $T = 180$ under pure advection, with $v = 1$ and Courant number as indicated. With time discretization to yield an integer Courant number, the peak is advected without distortion as shown in figure 1.33. If $Cr = \frac{1}{2}$, symmetric numerical dispersion is evident, as shown in figure 1.34. An nonsymmetric, oscillatory graph is produced for $Cr \in (-\frac{1}{2}, \frac{1}{2})$. As $Cr \rightarrow 0$, the maximum decreases and the oscillations become more extensive. This behavior is shown in figures 1.35 and 1.36. The train of oscillations extends primarily to the right of the maximum for $Cr < 0$ and to the left for positive Courant number. Figure 1.37 illustrates the advisability of using large time steps: fewer applications of the

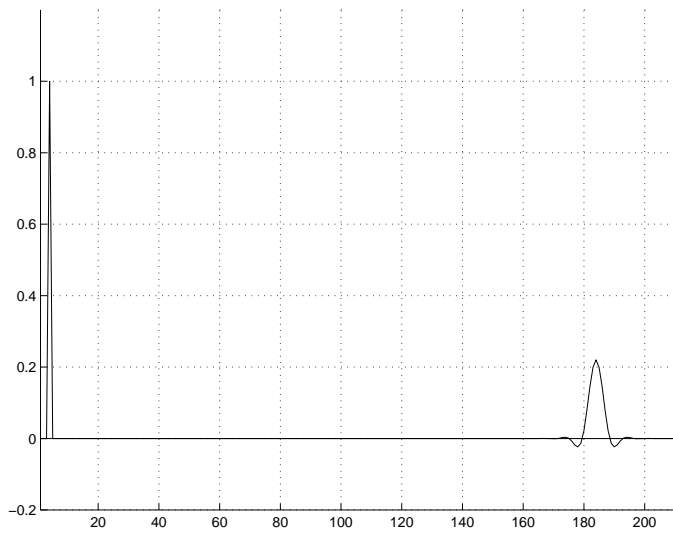


Figure 1.34. Initial and advected peaks with $Cr = \frac{1}{2}$.

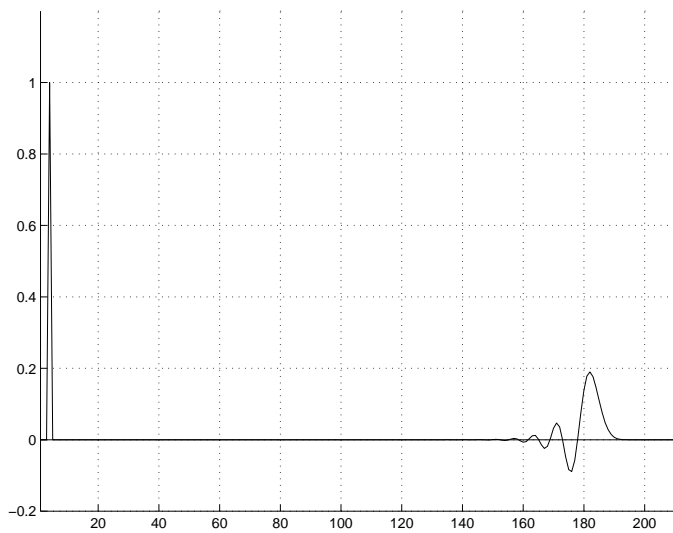


Figure 1.35. Initial and advected peaks with $Cr = \frac{1}{4}$.

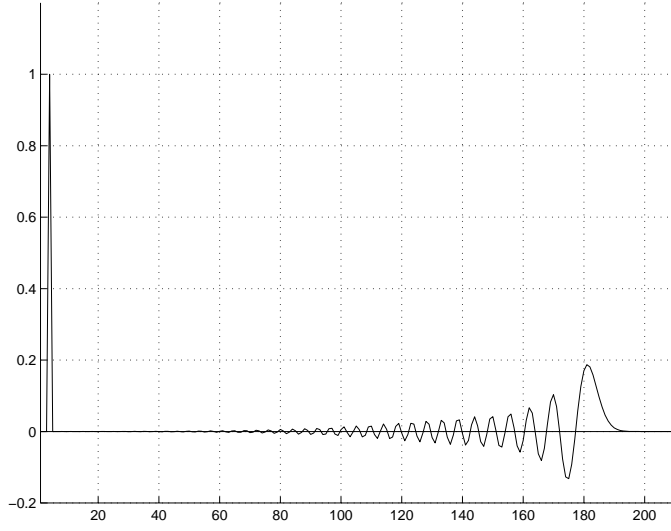


Figure 1.36. Initial and advected peaks with $Cr = \frac{1}{32}$

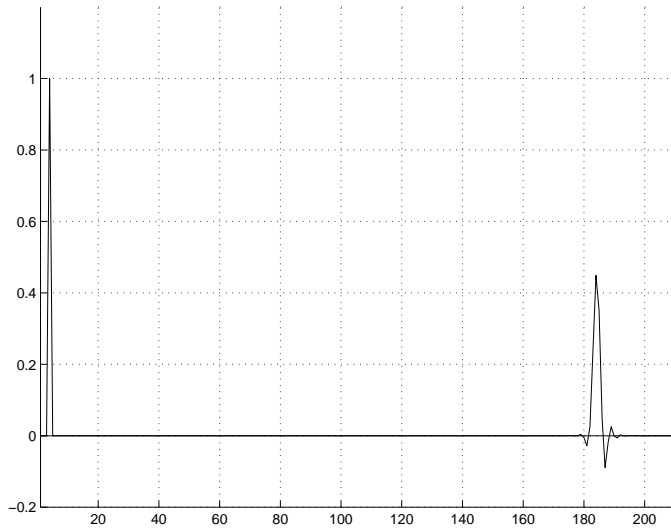


Figure 1.37. Initial and advected peaks with $Cr = 10\frac{1}{32}$.

operator with noninteger Courant number results better preservation of the initial contour.

This behaviour can be considered in light of the eigensystem of the related operator $S(A^{-1}B)^{\frac{1}{Cr}}$ in cases where $Cr = \frac{1}{integer}$. Here, S is the diagonal-shift by -1 operator,

$$S = [1, 0, 0].$$

Disregarding boundary effects, application of $(S(A^{-1}B)^{\frac{1}{Cr}})^T$ results in a concentration distribution with the same form as the advected solution, but stationary. The shifted matrix acts repeatedly on a linear combination of its eigenvectors, and the solution evolves accordingly. Ideally, if the initial condition were advected without distortion, this matrix would be the identity. Consider the 10×10 problem with an initial condition of one at node seven, $v = 1$, and $Cr = \frac{1}{8}$. All eigenvectors of $S(A^{-1}B)^{\frac{1}{Cr}}$ are plotted in 1.38. The placement of entries with largest modulus near the beginning of any eigenvector is a feature that characterizes matrices $(S(A^{-1}B)^{\frac{1}{Cr}})$ of various sizes. (This is also seen for $((A^{-1}B)^{\frac{1}{Cr}}S)$, which first shifts, then advects; and also $(S(i)(A^{-1}B)^{\frac{i}{Cr}})$ and $((A^{-1}B)^{\frac{i}{Cr}}S(i))$, with advection and shifting of i cells.) The eigenvector corresponding to the maximum eigenvalue is the smoothest vector, with its minimum at node three. Eigenvalues, and coefficients of their respective eigenvectors to produce the initial condition are shown in table 1.6.5. (The matrix is singular, due to the shift.) The initial condition holds large contributions from eigenvectors with small eigenvalues, so diminution and change of shape of the solution is to be expected under repeated matrix application. Indeed, this is observed. The initial condition is plotted along with the solution after different elapsed times (number of matrix applications). Shown in figure 1.39 are graphs for $T = 1$, $T = 10$, $T = 100$, and $T = 1000$, corresponding to the curves with highest, second highest, next, and smallest peaks, respectively. The eigenvector of the maximum eigenvalue is seen to emerge as predominate.

As the size of the problem increases, the system of eigenvectors becomes increasingly ill-conditioned. The entries with largest modulus remain near the beginning of any eigenvector. It is not possible to approximate the

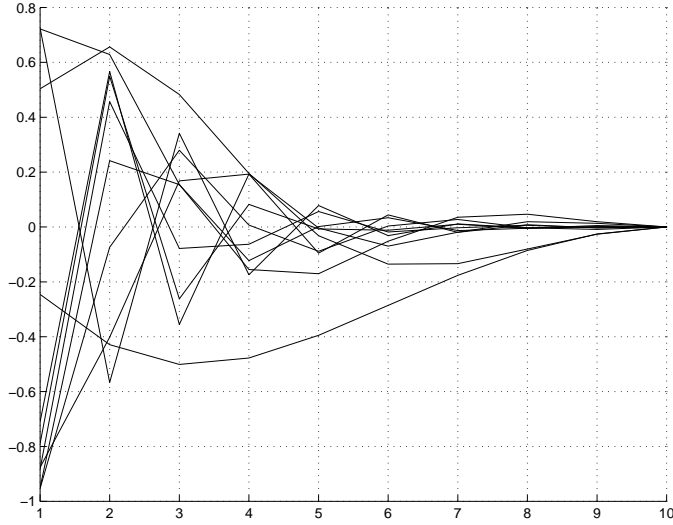


Figure 1.38. Eigenvectors of $10 \times 10 S(A^{-1}B)^{\frac{1}{Cr}}$ with $Cr = \frac{1}{8}$.

eigenvalues	coefficients
0.02512380295106	11.08455231527136
0.14348830209936	17.99738031282700
0.34537563664597	-6.21949567802237
0.56442393631698	-5.06404951515980
0.74559736660795	6.99680498309164
0.87006283785412	-2.97173348456930
0.99708068200960	-0.76361709637109
0.94405857162778	0.71286631865321
0.98197582016563	-1.50653675213281
0	0

Table 1.7. Eigenvalue and coefficient of the respective eigenvector in eigenvector decomposition of spike initial condition at node seven on 10×10 grid.

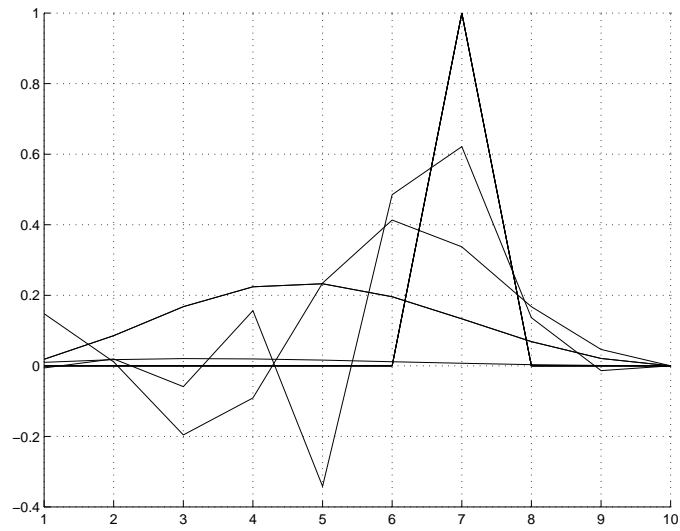


Figure 1.39. Initial condition and results using $10 \times 10 (S(A^{-1}B)\frac{1}{\sigma_r})^T$ with $T = 1, 10, 100, 1000$ and $Cr = \frac{1}{8}$.

initial condition very well without using many of the eigenvectors. The largest eigenvalues approach one (from below) in modulus. The eigenvector corresponding to the maximum eigenvalue remains the smoothest eigenvector.

A system with 40 unknowns is still small enough to find reasonable approximations to the coefficients for an eigenvector expansion of the initial condition, in spite of ill-conditioning of the matrix of eigenvectors. The initial condition and the results of 1, 10, 100, 1000, and 100000 applications of $S(A^{-1}B)^{\frac{1}{Cr}}$ are shown in figure 1.40. Again, the emergence of the dominant eigenvector can be traced. Analogous results are plotted in figure 1.41 for a system of order 200. The matrix of eigenvectors is too ill-conditioned for MATLAB to find coefficients for an eigenvector expansion of the initial condition, but the eigenvalues are numerically clearly distinct, so presumably a full set of eigenvectors exists. Figure 1.41 would then be understood as showing the emergence of eigenvector components of eigenvalues close to one. The single, smooth mode visible for $T = 10000$ in the smaller systems still is not isolated in the 200×200 system. The case of $Cr = \frac{1}{2}$ is anomalous: The nonzero entries of the eigenvector are periodically distributed evenly across the domain.

Next shown are graphs of the initial condition, and the concentration profile after $T = 1$, and $T = 100$ for $Cr = \frac{1}{8}$ for the 40×40 system with two (figure 1.42), three (figure 1.43), four (figure 1.44), and five (figure 1.45) nodes across a front or peak. Superposition of the oscillatory contours from the spikes composing the smoothed initial condition yields an advected concentration profile much closer to the original for the single spike initial condition. The peak maximum initially may increase slightly with respect to the initial condition. Still, with enough time steps, the peak tends to smooth, oscillate, and be displaced upstream, as would be expected considering the underlying eigensystem. Thus, even with well-discretized fronts, using few (large) time steps may be advisable with respect to the advection algorithm.

Note that the forgoing discussion applies exclusively to the consideration of finer time discretization, without accompanying refinement of the spatial grid. In figures 1.46 and 1.47 are shown superimposed plots illustrating

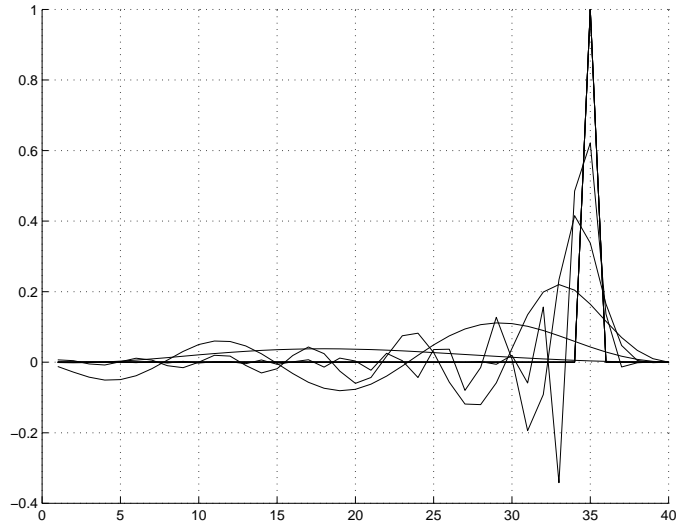


Figure 1.40. Initial condition and results using $40 \times 40 (S(A^{-1}B)^{\frac{1}{\sigma_r}})^T$ with $T = 1, 10, 100, 1000, 10000$ and $Cr = \frac{1}{8}$.

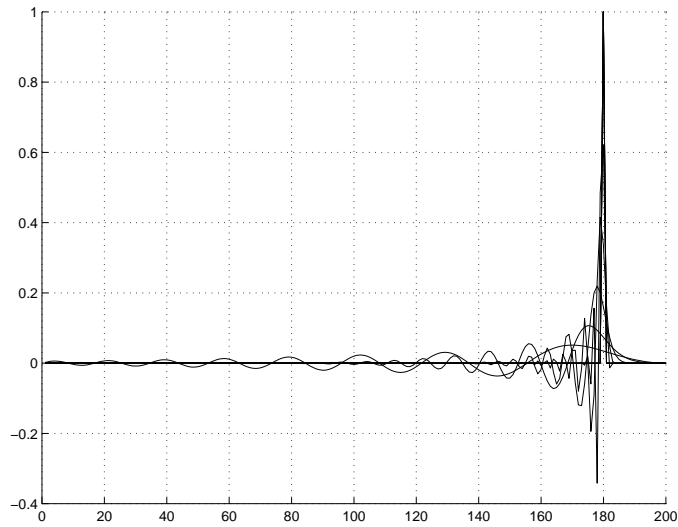


Figure 1.41. Initial condition and results using $200 \times 200 (S(A^{-1}B)^{\frac{1}{\sigma_r}})^T$ with $T = 1, 10, 100, 1000, 10000$ and $Cr = \frac{1}{8}$.

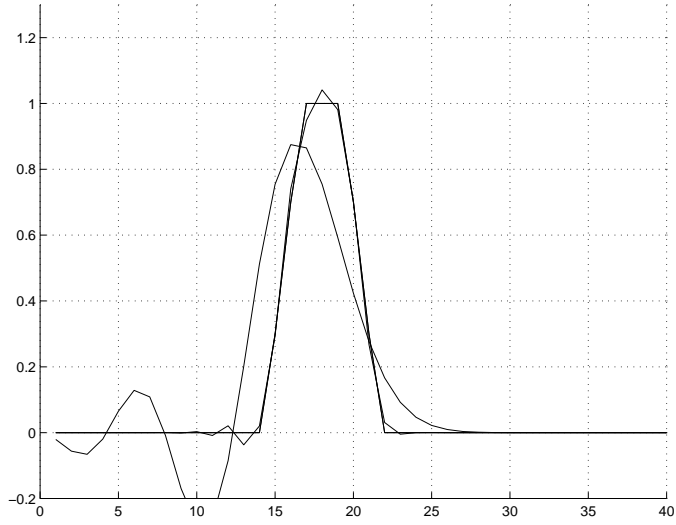


Figure 1.42. Initial condition with 2 nodes on a front, and results using $40 \times 40 (S(A^{-1}B)^{\frac{1}{Cr}})^T$ with $T = 1, 100$ and $Cr = \frac{1}{8}$.

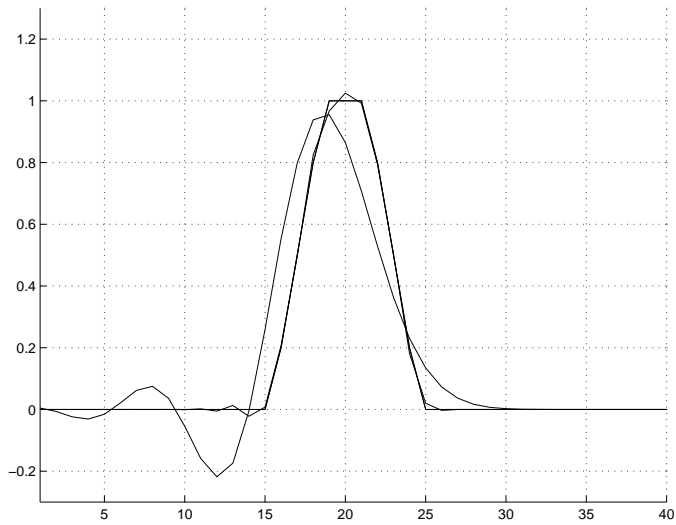


Figure 1.43. Initial condition with 3 nodes on a front, and results using $40 \times 40 (S(A^{-1}B)^{\frac{1}{Cr}})^T$ with $T = 1, 100$ and $Cr = \frac{1}{8}$.

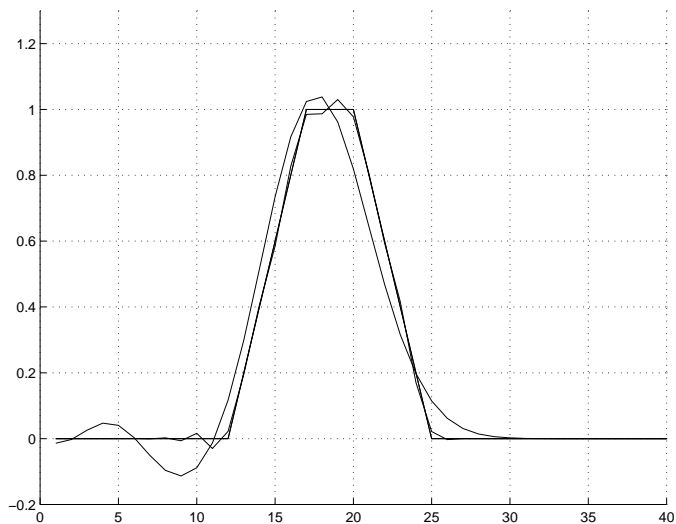


Figure 1.44. Initial condition with 4 nodes on a front, and results using $40 \times 40 (S(A^{-1}B)^{\frac{1}{Cr}})^T$ with $T = 1, 100$ and $Cr = \frac{1}{8}$.

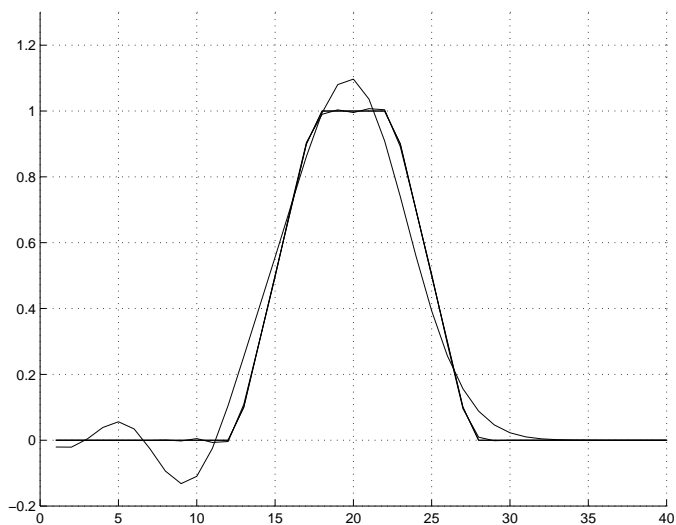


Figure 1.45. Initial condition with 5 nodes on a front, and results using $40 \times 40 (S(A^{-1}B)^{\frac{1}{Cr}})^T$ with $T = 1, 100$ and $Cr = \frac{1}{8}$.

the convergence of the method as both Δx and Δt approach zero. A spike initial condition on a 5×5 grid is advected two cells under a uniform velocity field. The grid is refined four times, each time by a factor of three. Simulation time and Courant number are constant for each figure, meaning a finer time discretization with each spatial refinement. Increasing height of peak height corresponds to progressively more refined grids of 5×5 , 15×15 , 45×45 , 135×135 , and 405×405 .

1.7 Conclusion

The ELLAM algorithm for solution of the advection-diffusion equation can simulate the transient, three-dimensional transport of a solute subject to decay and retardation. The accuracy of the ELLAM numerical results were tested and evaluated by comparison to analytical and numerical solutions to a set of test problems. These tests indicate that ELLAM can accurately simulate three-dimensional transport and dispersion of a solute in flowing ground water. The method appears robust, with demonstrated stability in a variety of test situations. It compares favorably to the method of characteristics codes used as benchmarks, in some cases. To avoid non-physical oscillations and loss of peak concentrations, care must be taken to use a grid with sufficient mesh density to adequately resolve sharp fronts. ELLAM is globally and locally mass conservative, and can provide good solutions using large time steps.

1.8 Further Research

With regard to an analysis of the one-dimensional ELLAM method (1.25), a proof of stability independent of the problem size remains to be accomplished. Incorporation of various boundary conditions into a stability proof is another direction for further effort. Generalization of convergence results to higher dimensional problems remains to be done. With respect to an ELLAM implementation, the possibility of specifying different spatial discretization for mass tracking (NS values) in different parts of the computational grid, might result in greater efficiency of computation. Incorporation of a backtracking scheme to identify the preimage at the old time level of a finite

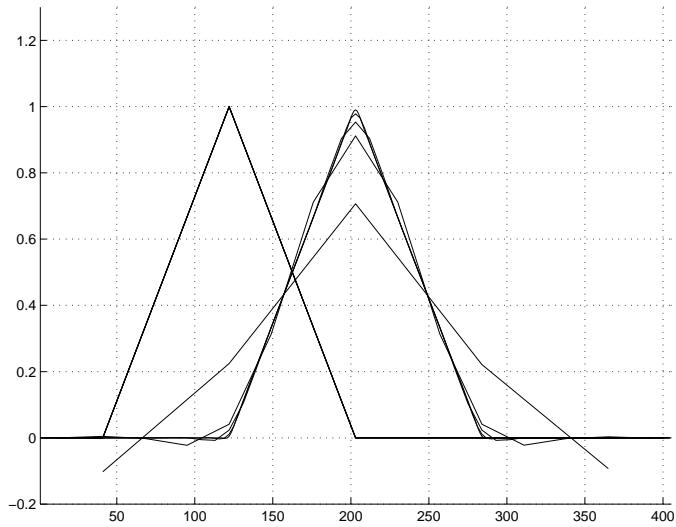


Figure 1.46. Superposition of results showing initial condition and advected peak using $Cr = \frac{1}{2}$. Grids are 5×5 , and 4 refinements, each by a factor of three. Initial condition is a spike on 5×5 grid. Advected peaks show increasing height with decreasing Δx .

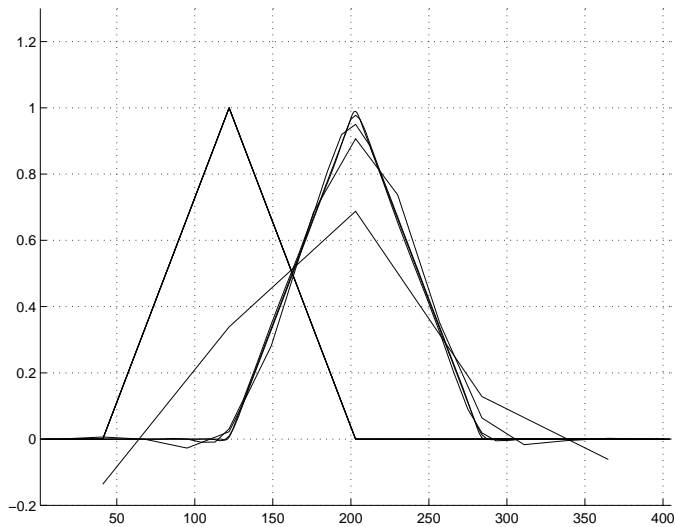


Figure 1.47. Superposition of results showing initial condition and advected peak using $Cr = \frac{4}{9}$. Grids are 5×5 , and 4 refinements, each by a factor of three. Initial condition is a spike on 5×5 grid. Advected peaks show increasing height with decreasing Δx .

difference cell at the new time level, could lead to improvements in accuracy and efficiency.

2. Smoothed Aggregations Algebraic Multigrid

2.1 Introduction

In engineering practice, problems are often encountered in which the material properties change by orders of magnitude from one part of the model to the next. Such changes usually adversely affect the conditioning of the linear algebraic systems obtained from discretization of these problems, and pose a challenge to the methods used for solving these systems. Similar effects may result from discretizations using locally-refined meshes. Adding to the difficulty, the problems are often discretized using unstructured meshes. This renders classical multilevel methods impractical. Algebraic multilevel schemes have been designed specifically to deal with the problems discretized over unstructured grids [18, 80, 86].

Efficient methods have been sought to handle algebraic systems arising from discretization of second order partial differential equations having subregions with markedly larger coefficients than the rest of the domain. Several non-multigrid methods are studied in [7, 4, 5, 6]. We present a variant of algebraic multigrid (AMG) designed for this purpose. This work is a joint effort with Petr Vaněk and Marian Brezina.

Aggregation-based multilevel methods [45, 44] using plain aggregation involve low computational complexity. In this chapter, we present an algebraic multilevel method of smoothed aggregation class. Such methods have previously been proposed in [86, 91, 87, 19, 92], and have proved effective in solving a variety of problems ranging from linear elasticity to the Helmholtz problem.

The classical aggregation method is a variational multigrid [9], i.e., the restriction operator is the transpose of the prolongation in which the prolongation operator P_{l+1}^l from the coarse level $l + 1$ to fine level l is constructed based on the decomposition of fine-level nodes into disjoint sets, called aggregates. The aggregates then determine the nonzero structure of the prolongation

for scalar problems, the prolongation operator from coarse level of dimension m to fine-level of dimension n stores exactly n nonzero entries. This sparsity of the transfer operators, together with a fast rate of coarsening, leads to solvers with very low algebraic and geometric complexities, as defined in [80]. These methods, as well as improvements based on them, can be easily generalized to solving both high order [87] and nonscalar [90] problems. This is achieved by associating more than one column with each aggregate, and the resulting method is referred to as the generalized aggregation method.

Unfortunately, the price for simplicity of aggregation methods is that their multilevel convergence is suboptimal, as observed in practical computation.

It is now understood that the reason for this suboptimality is that the disaggregated coarse-level basis functions possess large energy. This understanding has led to development of the smoothed aggregation methods, which address the above deficiency. Characteristic for the smoothed aggregation methods is that their prolongation operators are constructed in two steps. First, a so-called tentative prolongator P_{l+1}^l is constructed as in the generalized aggregation methods. As noted above, this operator has a simple sparsity structure. A smoothing operator S_l is then applied to obtain the final prolongator $I_{l+1}^l = S_l P_{l+1}^l$ used in the method. The hierarchy of coarse problems is then constructed by the recurrence

$$A_{l+1} = (I_{l+1}^l)^T A_l I_{l+1}^l. \quad (2.1)$$

The prolongation smoothing improves energetic properties of the coarse space basis functions and allows for approximation property (2.2) with constants uniform with respect to the level.

Recently, multilevel convergence result has been presented by Vaněk, Brezina and Mandel in [89] for a smoothed aggregation method applied to H^1 -equivalent problems, including nonscalar equations of linear elasticity discretized on unstructured meshes. The key assumption of that theory is satisfaction on all levels of the so-called weak approximation property, which can

be formulated as

$$\forall \mathbf{u} \in V_1 \quad \exists \mathbf{v} \in V_l : \|\mathbf{u} - P_{l+1}^1 \mathbf{v}\| \leq \frac{C}{\sqrt{\varrho(A_l)}} \|\mathbf{u}\|_A. \quad (2.2)$$

Although the method greatly benefits from utilization of the smoothed transfer operators I_{l+1}^l , note that the approximation property under which convergence is proved is formulated in terms of the properties of the tentative prolongators, and the smoothing S_l involved in the definition of I_{l+1}^l is present only in right-hand side in the form of the spectral radius $\varrho(A_l)$. This makes verification of (2.2) easy in practice [89].

In this work we take advantage of the local nature of aggregation basis functions in treatment of problems with coefficient discontinuities. Our algorithm uses a coarsening strategy that respects boundaries of high-coefficient subdomains. As a result, coarsening in the high-coefficient regions will be performed independently of the rest of the domain. Eventually, a high-coefficient region may be represented by a single node on one of the coarse levels. Such node will subsequently be eliminated from the system and no longer considered in the coarsening. In order to accommodate this elimination process, the prolongation smoother S_l has to be carefully designed. We introduce changes to the prolongation smoother considered in [89], which will allow us to retain good computational complexity, and allow us to prove for the problems with coefficient discontinuities the same asymptotic rate of convergence as proved in [89] for problems with uniformly H^1 -equivalent forms.

In our analysis, we will focus mainly on the case of a single high-coefficient region. The method is first presented in an abstract setting, and analyzed for the case of a scalar elliptic problem. Theory applicable to problems with multiple high-coefficient regions is then developed in section 2.8. The key assumption of our theory in sections 2.7 and 2.8 is a modified weak approximation property, which includes prolongator smoother S_l in its left-hand side. We show at the end of section 2.8 that our definition of S_l allows easy verification, in spite of the presence of the smoother S_l in the left-hand side. Finally, section 2.9 concludes with numerical experiments demonstrating the efficacy

of our smoothed aggregation method applied to several model problems.

2.2 Abstract Convergence Theory

In this section we present modification of the abstract theory used in [89] suitable for later application in the analysis of the problems with coefficient discontinuities.

Given a finest level $n_1 \times n_1$ symmetric and positive definite matrix A_1 and recalling the definition of the prolongators of the form $I_{l+1}^l = S_l P_{l+1}^l$, one iteration of the smoothed aggregation multigrid $\mathbf{x} \leftarrow MG(\mathbf{x}, \mathbf{b})$, solving

$$A_1 x = b, \tag{2.3}$$

is described in abstract terms as the following variational multigrid algorithm.

Algorithm 2.1 Let $R_l : \mathbb{R}^{n_l} \rightarrow \mathbb{R}^{n_l}$, $l = 1, \dots, L - 1$ be given smoothers and $\nu, \gamma > 0$ be a given smoothing and cycle parameter, respectively. Set $MG = MG_1$, where $MG_l(\cdot, \cdot)$, $l = 1, \dots, L - 1$ is defined by:

Pre-smoothing: Perform ν iterations of $\mathbf{x}^l \leftarrow (I - R_l A_l) \mathbf{x}^l + R_l \mathbf{b}^l$.

Coarse grid correction:

- Set $\mathbf{b}^{l+1} = (S_l P_{l+1}^l)^T (\mathbf{b}^l - A_l \mathbf{x}^l)$,
- If $l + 1 = L$, solve $A_{l+1} \mathbf{x}^{l+1} = \mathbf{b}^{l+1}$ by a direct method, otherwise set $\mathbf{x}^{l+1} = \mathbf{0}$ and perform γ iterations of $\mathbf{x}^{l+1} \leftarrow MG_{l+1}(\mathbf{x}^{l+1}, \mathbf{b}^{l+1})$,
- Correct the solution on level l by $\mathbf{x}^l \leftarrow \mathbf{x}^l + S_l P_{l+1}^l \mathbf{x}^{l+1}$.

Post-smoothing: Perform ν iterations of $\mathbf{x}^l \leftarrow (I - R_l A_l) \mathbf{x}^l + R_l \mathbf{b}^l$.

As noted in section 2.1, columns of the tentative prolongator P_{l+1}^l will have disjoint nonzero structure. Here and in the following sections we assume

that the columns of P_{l+1}^l have been normalized, so that $P_{l+1}^l : \mathbb{R}^{n_{l+1}} \rightarrow \mathbb{R}^{n_l}$, $n_1 \equiv \text{ord}(A) > n_2 > \dots > n_L$ is a full-rank orthogonal matrix,

$$(P_{l+1}^l)^T P_{l+1}^l = I, \quad l = 1, \dots, L-1. \quad (2.4)$$

Given a prolongator smoother $S_l : \mathbb{R}^{n_l} \rightarrow \mathbb{R}^{n_l}$, the hierarchy of coarse-level problems are constructed by the recurrence (2.1), so we have

$$A_{l+1} = (S_l P_{l+1}^l)^T A_l S_l P_{l+1}^l. \quad (2.5)$$

Following the proof by Vaněk, et al. in [89], we state an abstract convergence result for an AMG method based on smoothed aggregation with prolongator smoothers and tentative prolongators satisfying the requisite properties. The earlier result stipulated a weak approximation property to be satisfied only by the (unsmoothed) tentative prolongators. Here, we incorporate the prolongator smoother into the weak approximation property, noting that the smoother may indeed have a nontrivial kernel. In fact, we will take advantage of this in our analysis. The prolongation smoother will be constructed in a way which renders irrelevant for approximation selected entries of a coarse vector.

We now introduce some notation. Define the composite tentative prolongator, $P_l^1 : \mathbb{R}^{n_l} \rightarrow \mathbb{R}^{n_1}$, by

$$P_l^1 = P_2^1 \dots P_l^{l-1}, \quad P_1^1 = I,$$

and the smoothed composite prolongator $I_l^1 : \mathbb{R}^{n_l} \rightarrow \mathbb{R}^{n_1}$ by

$$I_l^1 = S_1 P_2^1 \dots S_{l-1} P_l^{l-1}, \quad I_1^1 = I.$$

The transfer operators I_l^1 define a hierarchy of coarse spaces $\mathcal{U}_L \subseteq \mathcal{U}_{L-1} \subseteq \dots \subseteq \mathcal{U}_1$ by $\mathcal{U}_l = \text{Range } I_l^1$, with the norm on \mathcal{U}_l induced by the \mathbb{R}^{n_l} -norm $\|\mathbf{x}\|_{\mathbb{R}^{n_l}} = (\mathbf{x}^T \mathbf{x})^{1/2}$,

$$\|\mathbf{u}\|_l = \min\{\|\mathbf{x}\|_{\mathbb{R}^{n_l}} : \mathbf{u} = I_l^1 \mathbf{x}\},$$

and the associated inner product $(\mathbf{u}, \mathbf{v})_l = (\mathbf{x}, \mathbf{y})_{\mathbb{R}^{n_l}}$. Note that by the construction of coarse problems (2.1),

$$\|I_l^1 \mathbf{x}\|_A = \|\mathbf{x}\|_{A_l}. \quad (2.6)$$

Our estimates are based on an abstract regularity-free convergence result proved in [17]. It can be written in our notation as follows:

Lemma 2.2 (Bramble, Pasciak, Wang, Xu [17], Theorem 1). Assume there are linear mappings $Q_l : \mathcal{U}_1 \mapsto \mathcal{U}_l$, $Q_1 = I$ and constants $c_1, c_2 > 0$ such that

- for all $\mathbf{u} \in \mathcal{U}_1$ and every level $l = 1, \dots, L$

$$\|Q_l \mathbf{u}\|_A \leq c_1 \|\mathbf{u}\|_A. \quad (2.7)$$

- for all $\mathbf{u} \in \mathcal{U}_1$ and every level $l = 1, \dots, L - 1$

$$\|(Q_l - Q_{l+1})\mathbf{u}\|_l \leq \frac{c_2}{\sqrt{\varrho(A_l)}} \|\mathbf{u}\|_A. \quad (2.8)$$

Further assume that R_l are symmetric positive definite matrices satisfying

$$\lambda_{\min}(I - R_l A_l) \geq 0 \quad \text{and} \quad \lambda_{\min}(R_l) \geq \frac{1}{c_R^2 \varrho(A_l)} \quad (2.9)$$

with a constant $c_R > 0$ independent of the level.

Then, Algorithm 2.1 satisfies

$$\|\hat{\mathbf{x}} - MG(\mathbf{x}, \mathbf{b})\|_A \leq \left(1 - \frac{1}{c_0(L)}\right) \|\hat{\mathbf{x}} - \mathbf{x}\|_A \quad \forall \mathbf{x} \in \mathcal{U}_1,$$

where $\hat{\mathbf{x}}$ is the solution of (2.3), and $c_0(L) = (1 + c_1 + c_2 c_R)^2 (L - 1)$. Moreover, the preconditioner P defined by the action of $MG(\mathbf{0}, \cdot)$ is symmetric with respect to $(\cdot, \cdot)_{\mathbb{R}^{n_1}}$ and $\text{cond}(A, P) \leq c_0(L)$.

In the following lemma, Assumptions (2.7) and (2.8) of Lemma 2.2 are verified from the properties of S_l and P_{l+1}^l .

Lemma 2.3 Let for every $l = 1, \dots, L-1$, $\bar{\lambda}_l \geq \varrho(A_l)$ and

$$\tilde{Q}_l : \mathcal{U}_1 \rightarrow \mathbb{R}^{n_l}, \quad \tilde{Q}_1 = I, \quad S_l : \mathbb{R}^{n_l} \rightarrow \mathbb{R}^{n_l}$$

be given linear operators. Assume that for some $C_1, C_2 > 0$ and all $l = 1, \dots, L-1$,

$$\|S_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{\mathbb{R}^{n_l}}^2 \leq \frac{C_1^2}{\bar{\lambda}_l} \|\mathbf{u}\|_A^2 \quad \forall \mathbf{u} \in \mathbb{R}^{n_l}, \quad (2.10)$$

$$(P_{l+1}^l)^T P_{l+1}^l = 1, \quad (2.11)$$

$$\|S_l\|_{A_l} \leq 1, \quad (2.12)$$

$$\|(I - S_l)\mathbf{x}\|_{\mathbb{R}^{n_l}}^2 \leq \frac{C_2^2}{\varrho(A_l)} \|\mathbf{x}\|_{A_l}^2 \quad \forall \mathbf{x} \in \mathbb{R}^{n_l}. \quad (2.13)$$

Then, for every $\mathbf{u} \in \mathcal{U}_1$, the mappings $Q_l = I_l^1 \tilde{Q}_l$ satisfy

$$\|Q_l \mathbf{u}\|_A \leq c_1(l) \|\mathbf{u}\|_A, \quad l = 1, \dots, L, \quad (2.14)$$

with $c_1(l) = 1 + C_1(l-1)$, and

$$\|(Q_l - Q_{l+1})\mathbf{u}\|_l \leq c_2(l) \varrho(A_l)^{-1/2} \|\mathbf{u}\|_A, \quad l = 1, \dots, L-1 \quad (2.15)$$

with $c_2(l) = C_1 + C_2 c_1(l)$.

Proof: Let $\mathbf{u} \in \mathcal{U}_1$. From the definitions of Q_{l+1} and I_{l+1}^1 , and (2.6),

$$\|Q_{l+1} \mathbf{u}\|_A = \|I_{l+1}^1 \tilde{Q}_{l+1} \mathbf{u}\|_A = \|I_l^1 S_l P_{l+1}^l \tilde{Q}_{l+1} \mathbf{u}\|_A = \|S_l P_{l+1}^l \tilde{Q}_{l+1} \mathbf{u}\|_{A_l}.$$

Then, using (2.12), (2.10), and (2.6), we have

$$\begin{aligned} \|Q_{l+1} \mathbf{u}\|_A &\leq \|S_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{A_l} + \|S_l \tilde{Q}_l \mathbf{u}\|_{A_l} \\ &\leq \varrho^{1/2}(A_l) \|S_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{\mathbb{R}^{n_l}} + \|\tilde{Q}_l \mathbf{u}\|_{A_l} \\ &\leq \varrho^{1/2}(A_l) C_1 \bar{\lambda}_l^{-1/2} \|\mathbf{u}\|_A + \|Q_l \mathbf{u}\|_A \\ &\leq C_1 \|\mathbf{u}\|_A + \|Q_l \mathbf{u}\|_A \end{aligned}$$

Estimate (2.14) follows by induction with $Q_1 = I$.

To prove (2.15), use the definition of the l -norm, (2.14), (2.13), and (2.6) to get

$$\begin{aligned}
\|(Q_l - Q_{l+1})\mathbf{u}\|_l &\leq \|(\tilde{Q}_l - S_l P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{\mathbb{R}^{n_l}} \\
&= \|S_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u} + (I - S_l)\tilde{Q}_l\mathbf{u}\|_{\mathbb{R}^{n_l}} \\
&\leq \|S_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{\mathbb{R}^{n_l}} + \|(I - S_l)\tilde{Q}_l\mathbf{u}\|_{\mathbb{R}^{n_l}} \\
&\leq \frac{C_1}{\bar{\lambda}_l^{1/2}} \|\mathbf{u}\|_A + \frac{C_2}{\varrho^{1/2}(A_l)} \|\tilde{Q}_l\mathbf{u}\|_{A_l} \\
&\leq \left(\frac{C_1 + C_2 \|Q_l\|_A}{\varrho^{1/2}(A_l)} \right) \|\mathbf{u}\|_A.
\end{aligned}$$

Using (2.14) to see $\|Q_l\|_A \leq c_1(l)$, we get (2.15). \blacksquare

The following convergence result is immediate from above Lemmas 2.2 and 2.3.

Theorem 2.4 Let for every $l = 1, \dots, L - 1$, $\bar{\lambda}_l \geq \varrho(A_l)$ and

$$\tilde{Q}_l : \mathcal{U}_1 \rightarrow \mathbb{R}^{n_l}, \quad \tilde{Q}_1 = I, \quad S_l : \mathbb{R}^{n_l} \rightarrow \mathbb{R}^{n_l}$$

be given linear operators. Assume that for some $C_1, C_2 > 0$ and all $l = 1, \dots, L - 1$,

$$\begin{aligned}
\|S_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{\mathbb{R}^{n_l}}^2 &\leq \frac{C_1^2}{\bar{\lambda}_l} \|\mathbf{u}\|_A^2 \quad \forall \mathbf{u} \in \mathbb{R}^{n_l}, \\
(P_{l+1}^l)^T P_{l+1}^l &= 1, \\
\|S_l\|_{A_l} &\leq 1, \\
\|(I - S_l)\mathbf{x}\|_{\mathbb{R}^{n_l}}^2 &\leq \frac{C_2^2}{\varrho(A_l)} \|\mathbf{x}\|_{A_l}^2 \quad \forall \mathbf{x} \in \mathbb{R}^{n_l}.
\end{aligned}$$

Then, with R_l satisfying (2.9), Algorithm 2.1 satisfies

$$\|\hat{\mathbf{x}} - MG(\mathbf{x}, \mathbf{b})\|_A \leq \left(1 - \frac{1}{c_0(L)} \right) \|\hat{\mathbf{x}} - \mathbf{x}\|_A \quad \forall \mathbf{x} \in \mathcal{U}_1,$$

where $\hat{\mathbf{x}}$ is the solution of (2.3), with $c_1(l) = 1 + C_1(l - 1)$, $c_2(l) = C_1 + C_2 c_1(l)$, and $c_0(L) = (1 + c_1 + c_2 c_R)^2 (L - 1)$.

Moreover, the preconditioner P defined by the action of $MG(\mathbf{0}, \cdot)$ is symmetric with respect to $(\cdot, \cdot)_{\mathbb{R}^{n_1}}$ and $\text{cond}(A, P) \leq c_0(L)$.

2.3 Model Problem

For clarity, we first present and analyze the AMG algorithm in the context of a scalar elliptic model problem with a single high-coefficient region.

Consider the second order scalar elliptic problem,

$$\text{Find } u \in V_h \text{ such that } a(u, v) = f(v) \text{ for every } v \in V_h. \quad (2.16)$$

Here $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ is a bounded domain; $f \in H^{-1}(\Omega)$; and $a(\cdot, \cdot)$ is a coercive and bounded bilinear form on $H^1(\Omega)$ where a high coefficient applies on a single simply connected subdomain, Ω^ε :

$$a(u, v) = \int_{\Omega'} \nabla u \cdot \nabla v \, d\mathbf{x} + \frac{1}{\varepsilon^2} \int_{\Omega^\varepsilon} \nabla u \cdot \nabla v \, d\mathbf{x},$$

with $\Omega' = \Omega \setminus \Omega^\varepsilon$ and $\varepsilon \ll 1$.

For concreteness, we assume a finite element discretization, with τ_h a quasiuniform finite element mesh on Ω , and V_h a P1 or Q1 finite element space associated with τ_h . At some of the boundary nodes, a zero Dirichlet boundary condition is imposed for functions in V_h . We assume the standard scaling of the finite element basis, $\|\varphi_i\|_{L^\infty} = 1$. This yields a linear system, $A\mathbf{x} = \mathbf{b}$.

We will then consider the case where high coefficients apply over a number of simply connected subdomains Ω_k^ε indexed by k .

2.4 Algorithm

The algorithm depends upon the smoothed aggregations concept. Let L designate the coarsest level. On each level l , $l = 1, \dots, L - 1$, nodes are organized in small, disjoint clusters called aggregates, $\{\mathcal{A}_i^l\}_{i=1}^{n_{l+1}}$ where n_{l+1} is the number of aggregates on level l , or equivalently, the number of nodes on level $l + 1$. On the finest level, these clusters have to be specified, e.g., as the sets of degrees of freedom associated with the finite element vertices, while

the coarse-level aggregates are created by our aggregation algorithm. A simple greedy algorithm for generating aggregates based on the structure of the coarse-level matrix is given in [87].

We introduce the composite aggregate $\tilde{\mathcal{A}}_i^l$ which is the aggregate \mathcal{A}_i^l , understood as the corresponding set of degrees of freedom on the finest level. Formally, $\tilde{\mathcal{A}}_i^l$ is defined by

$$\tilde{\mathcal{A}}_i^l = \mathcal{A}_i^{l,1}, \quad \text{where} \quad \mathcal{A}_i^{l,l} = \mathcal{A}_i^l, \quad \mathcal{A}_i^{l,j-1} = \bigcup_{k \in \mathcal{A}_i^{l,j}} \mathcal{A}_k^{j-1}. \quad (2.17)$$

Define the discrete l^2 -(semi)norm of the vector $\mathbf{x} \in \mathbb{R}^{n_1}$ given by

$$\|\mathbf{x}\|_{l^2(\tilde{\mathcal{A}}_i^l)} = \left(\sum_{\text{dofs } k \text{ of } \tilde{\mathcal{A}}_i^l} x_k^2 \right)^{1/2}.$$

Note that since aggregates on each level make a disjoint covering of the nodes,

$$\|\mathbf{x}\|_{\mathbb{R}^{n_1}}^2 = \sum_{i=1}^{n_l} \|\mathbf{x}\|_{l^2(\tilde{\mathcal{A}}_i^{l-1})}^2. \quad (2.18)$$

Let the j th node on level l be denoted $\omega_{l,j} = \tilde{\mathcal{A}}_j^{l-1}$, $j = 1, \dots, n_l$.

The algorithm requires as input the kernel of the stiffness matrix obtained from the finite element model with no essential boundary conditions.

Assume that we are given the system to solve, $Ax = b$, and the representation B of the zero-energy mode(s) of the bilinear form with respect to the finite element basis. For second order scalar problems discretized by standard linear Lagrange finite elements, $B = \mathbf{1}$, a vector of ones.

In order to facilitate the analysis for problems with jumps in coefficients, instead of the original system $Ax = b$, we consider the equivalent problem with diagonally-scaled finest-level matrix $A_1x = D^{-1/2}b$, where $D = \text{diag}(A)$ and $A_1 = D^{-1/2}AD^{-1/2}$. We note that as a consequence of this transformation, the representation of the zero-energy modes changes to $B_1 = D^{1/2}B$.

Our coarsening algorithm starts by aggregating separately in the high-coefficient region. The generalized aggregation coarsening method can be described as follows.

Coarse-level representations of B_1 and of the tentative prolongators P_{l+1}^l can be constructed simultaneously by the following recursive procedure, starting from A_1, B_1 : Based on the current level matrix A_l , the nodes are decomposed into disjoint aggregates. The aggregates determine the nonzero structure of the tentative prolongator P_{l+1}^l . Using the knowledge of B_l and the disjoint nonzero structure of P_{l+1}^l , the values of P_{l+1}^l are computed simultaneously with the coarse-grid representation of the kernel, B_{l+1} by the recurrence [89]

$$P_{l+1}^l B_{l+1} = B_l.$$

This construction is unique due to the normalization (2.4).

As a result of separate coarsening of a high-coefficient region, this region will eventually be represented by a single node on one of the coarse levels. At finer levels, because of the diagonal scaling and separate treatment of high-coefficient subregions, the tentative prolongators P_{l+1}^l obtained are identical to those in the uniform coefficient case.

After the level where the high-coefficient area is first represented by a single node, the separate coarsening forces us to consider the node as a separate one-node aggregate. This is undesirable for two reasons. Applying prolongator smoother to a tentative prolongator with a single-node aggregate leads to extensive overlapping of coarse grid basis functions which mean coarse problem fill-in. Also, the presence of many high-coefficient regions might lead to a proportionately large coarsest grid size. To avoid this situation, single-node aggregates will be eliminated from the system.

These issues also force us to consider a modified prolongator smoother operator. While [89] considers prolongator smoothers of the form

$$S_l = I - \frac{4}{3\varrho(A_l)} A_l, \tag{2.19}$$

we generalize here to a prolongator smoother of the form

$$S_l = (I - P_{A_l, U_l}) \left(I - \frac{4}{3\lambda_{V_l}} P_{V_l} A_l \right) (I - P_{A_l, U_l}) \tag{2.20}$$

where P_{A_l, U_l} denotes a projection onto U_l orthogonal with respect to the A_l -inner product, and P_V denote projection onto V_l orthogonal with respect to the Euclidean inner product.

In order to describe this modification, we introduce for each level subspaces U_l of nodes to be eliminated and V_l of nodes to be smoothed over. Thus, if there is a node j^* that represents a single-node aggregate, we set $V_l = \{x \in \mathbb{R}^{n_l} : x_{j^*} = 0\}$. This choice makes P_{V_l} serve as a filter, and will guarantee that the value of x_{j^*} will not be changed by prolongator smoothing. This averts extensive overlapping of coarse-level basis functions which would otherwise result.

If j^* is a node to be eliminated, we set $U_l = \text{span}\{\alpha \mathbf{e}_{j^*}, \alpha \in \mathbb{R}\}$. With this choice, the prolongator smoother performs a simple harmonic extension into a node being eliminated from its neighbors. This is necessary to guarantee that we can interpolate into all fine-level nodes after the node elimination.

Application of this coarsening-smoothing strategy also requires another concern be taken into account: we must guarantee that the energetic projection is stable in the Euclidean norm. The diagonal scaling implies that the coarse-level basis function corresponding to the single-node aggregate has very little energy. Because of this low energy, the corresponding diagonal entry in the coarse level matrix is small, and the Euclidean norm of the energetic projection may be large. To assure the stability of the energetic projection, provision is made to move a single-node aggregate to coarser levels, independent of smoothing over the rest of the domain. The projection P_{V_l} serves to preserve the diagonal value corresponding to the one-node aggregate. Meanwhile, the spectral radius of the coarse-level matrices are decreasing. When the spectral radius of the coarse-level matrix is comparable to the small diagonal entry associated with the single-node aggregate, the high-coefficient node is eliminated.

2.4.1 Stages of Processing

The processing of a node associated with a high-coefficient subdomain is at one of four stages at each level l of coarsening. The creation of aggregates, the space V_l which is smoothed, and the space U_l of nodes to be eliminated at the current level, all depend on the stage of processing of each high-coefficient region. Spaces V_l and U_l are used to formalize the description of carrying, then eliminating, nodes, and generalize easily to the case of multiple high-coefficient subdomains.

Stage 1. Separate coarsening in Ω^ε and in $\Omega \setminus \Omega^\varepsilon$; standard smoothing over the entire space.

- Aggregates \mathcal{A}_j^l are created so that if $\tilde{\mathcal{A}}_j^l$ contains a node of Ω^ε , all nodes of $\tilde{\mathcal{A}}_j^l$ are in Ω^ε .
- $U_l = \emptyset$, $V_l = \mathbb{R}^{n_l}$.

Ends when there is a j such that $\tilde{\mathcal{A}}_j^l$ contains all nodes in $\bar{\Omega}^\varepsilon$, the closure of Ω^ε in Ω . Assume $j = n_{l+1}$ for convenience.

Stage 2. Single node representing Ω^ε carried to coarser levels by means of single-node aggregate; extensive fill-in of coarse-level matrices prevented by means of "filter" P_{V_l} .

- Create aggregates $\{\mathcal{A}_j^l\}_{j=1}^{n_{l+1}}$ covering the nodes $1, \dots, n_l - 1$ (i.e. skip the node which corresponds to Ω^ε .)
- Set $n_{l+1} = n_{l+1} + 1$
- $\mathcal{A}_{n_{l+1}}^l = \{n_l\}$ (the "last aggregate", for convenience)
- $U_l = \emptyset$; $V_l = \{\mathbf{x} \in \mathbb{R}^{n_l} : \mathbf{x}_{n_l} = 0\}$ to disable smoothing over $\{n_l\}$.

Ends when $(A_{l+1})_{n_{l+1}, n_{l+1}} \geq c_L \bar{\lambda}_{V_{l+1}}$, where c_L is a specified parameter.

Stage 3. Coarsening eliminates single node aggregate representing Ω^ε ; smooth over entire space but with smoother kernel equal to the high-coefficient subspace on level l .

- Create aggregates $\{\mathcal{A}_j^l\}_{j=1}^{n_{l+1}}$ covering the nodes $1, \dots, n_l - 1$ (i.e. skip the node which corresponds to Ω^ε .)

- $U_l = \{\alpha \mathbf{e}^{n_l}, \alpha \in \mathbb{R}\}; V_l = \mathbb{R}^{n_l}$

Stage at single level only.

Stage 4. Standard coarsening; standard smoothing.

- Partition nodes into non-overlapping aggregates $\{\mathcal{A}_i^l\}_{i=1}^{n_{l+1}}$.
- $U_l = \emptyset; V_l = \mathbb{R}^{n_l}$

Ends at coarsest level, L .

Algorithm 2.5 Set $A_1 = D^{-1/2}AD^{-1/2}$, $B^1 = D^{1/2}B$.

Set $l = 1$

Repeat:

- (1) Partition the active nodes into non-overlapping aggregates $\{\mathcal{A}_i^l\}_{i=1}^{n_{l+1}}$.
- (2) Define an $n_l \times n_{l+1}$ matrix P_{l+1}^l

$$(P_{l+1}^l)_{ij} = \begin{cases} (B^l)_i & \text{if } i \in \mathcal{A}_j^l \\ 0 & \text{otherwise.} \end{cases} \quad (2.21)$$

- (3) Scale P_{l+1}^l columnwise and create B^{l+1} :

For $j = 1, \dots, n_{l+1}$

- Set $\alpha_j = \|\text{col } j(P_{l+1}^l)\|$
- Update $\text{col } j(P_{l+1}^l) \leftarrow \frac{1}{\alpha_j} \text{col } j(P_{l+1}^l)$
- Set $(B^{l+1})_j = \alpha_j$

End for

- (4) Define spaces $U_l, V_l \subset \mathbb{R}^{n_l}$ and set

$$S_l = (I - P_{A_l, U_l})(I - \frac{4}{3\bar{\lambda}_{V_l}} P_{V_l} A_l)(I - P_{A_l, U_l}), \quad (2.22)$$

where P_{A_l, U_l} is the A_l -orthogonal projector onto U_l , P_{V_l} is the projector onto V_l which is orthogonal in the Euclidian inner product, and $\bar{\lambda}_{V_l}$ is an upper bound on the spectral radius of $P_{V_l} A_l P_{V_l}$ (see Remark 2.8).

- (5) Create $A_{l+1} \leftarrow (S_l P_{l+1}^l)^T A_l S_l P_{l+1}^l$.
- (6) Set $l \leftarrow l + 1$.

Repeat until n_l is small enough for A_l to be treated by direct solver.

We next show that the prolongator smoothers (2.22) and tentative prolongators (2.21) are such that the assumptions of Lemma 2.3 are satisfied.

2.5 Smoother properties

Prolongator smoother (2.22) is simply of form (2.19) in case all processing is in Stage 1 or Stage 4. The form (2.22) accommodates Stage 2 carrying a high-coefficient node to a coarser level without smoothing and the Stage 3 elimination of a high-coefficient node. Stage 2 reduces $\varrho(A_l)$ while maintaining any diagonal entry associated with a single-node high-coefficient region.

The following lemmas regarding prolongator smoothers (2.22) are stated and proved in sufficient generality for the case of multiple high-coefficient subdomains.

Lemma 2.6 Let A be an $n \times n$ symmetric positive semidefinite matrix and U, V subspaces of \mathbb{R}^n .

Set $\omega = 4/3$ and,

$$S = (I - P_{U,A}) \left(I - \frac{\omega}{\bar{\lambda}_V} P_V A \right) (I - P_{U,A}), \quad (2.23)$$

where $P_{U,A}$ is the projection onto U which is orthogonal in the Hilbert space $\{\mathbb{R}^n, \langle A \cdot, \cdot \rangle\}$, P_V is the projection onto V orthogonal in $\{\mathbb{R}^n, \langle \cdot, \cdot \rangle\}$ and $\bar{\lambda}_V \geq \varrho(P_V A P_V)$.

Then

$$\|S\|_A \leq 1, \quad (2.24)$$

$$\varrho(P_V S^T A S P_V) \leq \frac{1}{9} \bar{\lambda}_V. \quad (2.25)$$

If, in addition, U is nonempty and

$$\min_{\mathbf{x} \in U} \frac{\langle A \mathbf{x}, \mathbf{x} \rangle}{\|\mathbf{x}\|^2} \geq C^2 \bar{\lambda}_V, \quad (2.26)$$

for some $C > 0$, then

$$\|(I - S)\mathbf{x}\| \leq \frac{1}{\sqrt{\bar{\lambda}_V}} \left[\frac{1}{C} + \frac{4}{3} \frac{\varrho(A)}{\bar{\lambda}_V} \left(\sqrt{\frac{\bar{\lambda}_V}{\varrho(A)} + \frac{1}{C}} \right) \right] \|\mathbf{x}\|_A \quad (2.27)$$

for all $\mathbf{x} \in \mathbb{R}^n$ and,

$$\|S\| \leq \left(1 + \frac{4}{3} \frac{\varrho(A)}{\bar{\lambda}_V}\right) \left(1 + \frac{1}{C} \sqrt{\frac{\varrho(A)}{\bar{\lambda}_V}}\right). \quad (2.28)$$

If U is empty, then (2.27) and (2.28) hold with $\frac{1}{C} = 0$.

Proof: Set $S_A = I - \omega \bar{\lambda}_V^{-1} P_V A$ and $P_{U,A}^\perp = I - P_{U,A}$. Since $P_{U,A}^\perp$ is an A -orthogonal projection, norm submultiplicativity gives

$$\|S\|_A \leq \|P_{U,A}^\perp\|_A^2 \cdot \|S_A\|_A = \|S_A\|_A. \quad (2.29)$$

Let $\mathbf{x} \in \mathbb{R}^n$. From the properties of the projection P_V we have $\langle A\mathbf{x}, P_V A\mathbf{x} \rangle = \|P_V A\mathbf{x}\|^2$ and $\|P_V A\mathbf{x}\|_A^2 \leq \bar{\lambda}_V \|P_V A\mathbf{x}\|^2$. Hence

$$\begin{aligned} \|S_A \mathbf{x}\|_A^2 &= \|\mathbf{x}\|_A^2 - 2 \frac{\omega}{\bar{\lambda}_V} \langle A\mathbf{x}, P_V A\mathbf{x} \rangle + \left(\frac{\omega}{\bar{\lambda}_V}\right)^2 \|P_V A\mathbf{x}\|_A^2 \\ &\leq \|\mathbf{x}\|_A^2 - 2 \frac{\omega}{\bar{\lambda}_V} \|P_V A\mathbf{x}\|^2 + \frac{\omega^2}{\bar{\lambda}_V} \|P_V A\mathbf{x}\|^2 \\ &= \|\mathbf{x}\|_A^2 - \frac{\omega}{\bar{\lambda}_V} (2 - \omega) \|P_V A\mathbf{x}\|^2 \leq \|\mathbf{x}\|_A^2. \end{aligned}$$

The statement (2.24) now follows by (2.29).

Next we prove (2.25). Let $\mathbf{x} \in V$. The well-known properties of orthogonal projections $AP_{U,A}^\perp = (P_{U,A}^\perp)^T AP_{U,A}^\perp$, $P_V \mathbf{x} = \mathbf{x}$, $P_V^2 = P_V$ and $P_V^T = P_V$ yield

$$\begin{aligned} \langle S^T A S \mathbf{x}, \mathbf{x} \rangle &= \|P_{U,A}^\perp (I - \omega \bar{\lambda}_V^{-1} P_V A) P_{U,A}^\perp \mathbf{x}\|_A^2 \\ &= \|P_{U,A}^\perp (I - \omega \bar{\lambda}_V^{-1} P_V (P_{U,A}^\perp)^T AP_{U,A}^\perp) \mathbf{x}\|_A^2 \\ &= \|P_{U,A}^\perp P_V (I - \omega \bar{\lambda}_V^{-1} P_V (P_{U,A}^\perp)^T AP_{U,A}^\perp P_V) \mathbf{x}\|_A^2 \\ &= \bar{\lambda}_V \langle p(\tilde{A}) \mathbf{x}, \mathbf{x} \rangle, \end{aligned} \quad (2.30)$$

where p is a polynomial

$$p(t) = \left(1 - \frac{4}{3} t\right)^2 t \quad (2.31)$$

and \tilde{A} is a projected and scaled matrix

$$\tilde{A} = \frac{1}{\bar{\lambda}_V} P_V (P_{U,A}^\perp)^T AP_{U,A}^\perp P_V. \quad (2.32)$$

Further,

$$\varrho(\tilde{A}) = \frac{1}{\bar{\lambda}_V} \max_{\mathbf{x} \in \mathbb{R}^n} \frac{\langle A P_{U,A}^\perp P_V \mathbf{x}, P_{U,A}^\perp P_V \mathbf{x} \rangle}{\|P_V \mathbf{x}\|^2 + \|P_V^\perp \mathbf{x}\|^2} \leq \frac{1}{\bar{\lambda}_V} \max_{\mathbf{x} \in V} \frac{\langle A \mathbf{x}, \mathbf{x} \rangle}{\|\mathbf{x}\|^2} \leq 1.$$

Denoting by $\sigma(\tilde{A})$ the spectrum of \tilde{A} , and using $\varrho(\tilde{A}) \leq 1$, the spectral mapping theorem gives

$$\varrho(p(\tilde{A})) = \max_{t \in \sigma(\tilde{A})} \left(1 - \frac{4}{3}t\right)^2 t \leq \max_{t \in [0,1]} \left(1 - \frac{4}{3}t\right)^2 t = \frac{1}{9}. \quad (2.33)$$

Substituting (2.33) into (2.30) one gets

$$\varrho(P_V S^T A S P_V) = \max_{\mathbf{x} \in V} \frac{\langle A S \mathbf{x}, S \mathbf{x} \rangle}{\|\mathbf{x}\|^2} \leq \frac{1}{9} \bar{\lambda}_V, \quad (2.34)$$

proving (2.25)

To prove (2.27) for nonempty U , we use the identity $(I - P_{U,A})^2 = I - P_{U,A}$, rewrite S as

$$S = I - P_{U,A} - \frac{\omega}{\bar{\lambda}_V} (I - P_{U,A}) P_V A (I - P_{U,A})$$

and estimate for $\mathbf{x} \in \mathbb{R}^n$,

$$\begin{aligned} \|(I - S) \mathbf{x}\| &\leq \|P_{U,A} \mathbf{x}\| + \frac{\omega}{\bar{\lambda}_V} (1 + \|P_{U,A}\|) \|P_V\| \|A^{1/2}\| \|(I - P_{U,A}) \mathbf{x}\|_A \\ &\leq \|P_{U,A} \mathbf{x}\| + \omega \frac{\varrho(A)}{\bar{\lambda}_V} \frac{1}{\sqrt{\varrho(A)}} (1 + \|P_{U,A}\|) \|\mathbf{x}\|_A. \end{aligned} \quad (2.35)$$

By (2.26), it follows that for $\mathbf{x} \in \mathbb{R}^n$,

$$\|P_{U,A} \mathbf{x}\|^2 \leq \frac{1}{C^2 \bar{\lambda}_V} \langle A P_{U,A} \mathbf{x}, P_{U,A} \mathbf{x} \rangle \leq \frac{1}{C^2 \bar{\lambda}_V} \|\mathbf{x}\|_A^2,$$

and, by $\|\mathbf{x}\|_A^2 \leq \varrho(A) \|\mathbf{x}\|^2$, one further gets

$$\|P_{U,A}\| \leq \frac{1}{C} \sqrt{\frac{\varrho(A)}{\bar{\lambda}_V}}.$$

Substituting two estimates above into (2.35) completes the proof of (2.27).

The statement (2.28) follows from (2.27) using inequalities

$$\|S\mathbf{x}\| \leq \|\mathbf{x}\| + \|(I - S)\mathbf{x}\|, \quad \|\mathbf{x}\|_A \leq \sqrt{\varrho(A)}\|\mathbf{x}\|, \quad \mathbf{x} \in \mathbb{R}^n.$$

If U is empty, then $I - P_{U,A} = I$ and both (2.27) and (2.28) again hold with $\frac{1}{C} = 0$. ■

Remark 2.7 We will see from the bounds (2.27) and (2.28) that the parameter c_L in the Stage 2 stopping condition must be chosen to balance conflicting requirements. If this parameter is extremely small, the constant C in (2.26) - (2.28) will be close to zero; but a very large parameter value allows a larger value of the ratio $\frac{\varrho(A)}{\lambda_V}$. Elimination of a single-node aggregate representing a high coefficient region may have to be postponed in order to stabilize the energetic projection which depends on $\frac{1}{C}$.

The following lemma asserts how to calculate $\bar{\lambda}_{V_i}$ used in the Stage 2 stopping criterion based on the value of $\bar{\lambda}_{V_1}$. In order to show that prolongator smoothers and tentative prolongators satisfy convergence assumptions, λ_l must be related to $\bar{\lambda}_{V_i}$. Remark 2.9 discusses how properties of the coarse level matrices, together with the Stage 2 stopping condition provide the necessary control over submatrix eigenvalues. Bounds on λ_l are then formulated in a lemma.

Lemma 2.8 For $\bar{\lambda}_{V_1}$ chosen as

$$\bar{\lambda}_{V_i} := \left(\frac{1}{9}\right)^{l-1} \bar{\lambda}_{V_1}, \quad (2.36)$$

where $\bar{\lambda}_{V_1}$ is a given upper bound of $\varrho(P_{V_1}A_1P_{V_1})$, satisfies

$$\bar{\lambda}_{V_i} \geq \varrho(P_{V_i}A_lP_{V_i}). \quad (2.37)$$

Proof: By construction, P_{l+1}^l of Algorithm 2.5 maps V_{l+1} into V_l and $(P_{l+1}^l)^T P_{l+1}^l = I$. We have $P_{l+1}^l P_{V_{l+1}} \mathbf{x} = P_{V_l} P_{l+1}^l P_{V_{l+1}} \mathbf{x}$ and therefore,

$$(P_{V_{l+1}} A_{l+1} P_{V_{l+1}} \mathbf{x}, \mathbf{x}) = (S_l^T A_l S_l P_{l+1}^l P_{V_{l+1}} \mathbf{x}, P_{l+1}^l P_{V_{l+1}} \mathbf{x})$$

$$\begin{aligned}
&= (P_{V_l} S_l^T A_l S_l P_{V_l} P_{V_{l+1}}^l P_{V_{l+1}} \mathbf{x}, P_{V_{l+1}}^l P_{V_{l+1}} \mathbf{x}) \\
&\leq \varrho(P_{V_l} S_l^T A_l S_l P_{V_l}) \|P_{V_{l+1}}^l P_{V_{l+1}} \mathbf{x}\|^2, \tag{2.38}
\end{aligned}$$

where

$$\begin{aligned}
\|P_{V_{l+1}}^l P_{V_{l+1}} \mathbf{x}\|^2 &= ((P_{V_{l+1}}^l)^T P_{V_{l+1}}^l P_{V_{l+1}} \mathbf{x}, P_{V_{l+1}} \mathbf{x}) \\
&= \|P_{V_{l+1}} \mathbf{x}\|^2 \\
&\leq \|\mathbf{x}\|^2.
\end{aligned}$$

Substituting this estimate into (2.38) gives,

$$\varrho(P_{V_{l+1}} A_{l+1} P_{V_{l+1}}) \leq \varrho(P_{V_l} S_l^T A_l S_l P_{V_l}). \tag{2.39}$$

The inequality (2.37) follows from (2.39) and (2.25) by induction. ■

Remark 2.9 Let $A_{V_l^\perp}$ and A_{U_l} be the nonempty blocks of matrix A_l defined by $(I - P_{V_l})A_l(I - P_{V_l})$ and $P_{U_l}A_lP_{U_l}$, respectively. The purpose of the stopping condition in Stage 2 is to assure that on every level,

$$\bar{\lambda}_{V_l^\perp} \leq C' \bar{\lambda}_{V_l} \quad (\text{if } V_l^\perp \neq \emptyset) \tag{2.40}$$

and at the same time,

$$\lambda_{\min}(A_{U_l}) \geq c^2 \bar{\lambda}_{V_l} \quad (\text{if } U_l \neq \emptyset). \tag{2.41}$$

In case of a single high-coefficient subdomain, $A_{V_l^\perp}$ is a 1 x 1 matrix if $V_l^\perp \neq \emptyset$, and A_{U_l} is a 1 x 1 matrix if $U_l \neq \emptyset$. Hence (2.40) and (2.41) are satisfied trivially.

In the case of multiple high-coefficient subdomains, a straightforward generalization of Algorithm 2.5 applies the stopping criterion to each diagonal entry in $A_{V_l^\perp}$ independently, resulting in a $A_{U_{l+1}}$ which is a submatrix of $A_{V_l^\perp}$, with all of its diagonal entries approximately equal. Since $A_{V_l^\perp}$ and A_{U_l} correspond to mutual interactions of high-coefficient subdomains, they are typically diagonal or sparse, and A_{U_l} is well-conditioned. Since $A_{V_l^\perp}$ is

Gram, the Gershgorin Theorem and the Cauchy-Schwarz Inequality together with the Stage 2 stopping condition guarantee (2.40) with $C' = Cc_L$. For $\lambda_{\min}(A_{U_l}) \geq C \min_i(A_{U_l})_{ii} \geq Cc_L \bar{\lambda}_{V_l}$, we have (2.41). Note that the stopping condition will be satisfied eventually for any $c_L > 0$ since $\bar{\lambda}_{V_l}$ decreases by a factor of 9 each level. To enforce (2.40) and (2.41) without indirect assumptions would require modification of Algorithm 2.5 and theory for A_{U_l} to always be diagonal.

Lemma 2.10 The largest eigenvalue of a coarse level matrix is related to the bound on the spectral radius of submatrix A_{V_l} by

$$\lambda_l \leq (1 + C')^2 \bar{\lambda}_{V_l} \quad (2.42)$$

and

$$\lambda_l \leq \frac{1}{9^{l-1}} (1 + C')^2 \bar{\lambda}_{V_l} \leq \frac{C'}{9^{l-1}} \quad (2.43)$$

for some positive constant C' dependent on A , the Stage 2 stopping condition, and finally, in (2.43), $\bar{\lambda}_{V_l}$.

Proof: For any $\mathbf{x} \in \mathbb{R}^{n_l}$,

$$(A_l \mathbf{x}, \mathbf{x})^{1/2} \leq (A_l P_{V_l} \mathbf{x}, P_{V_l} \mathbf{x})^{1/2} + (A_l P_{V_l^\perp} \mathbf{x}, P_{V_l^\perp} \mathbf{x})^{1/2},$$

so

$$\lambda_l^{1/2} \leq \bar{\lambda}_{V_l}^{1/2} + \bar{\lambda}_{V_l^\perp}^{1/2}.$$

The bound (2.42) follows using (2.40). Substituting first (2.36), then then $\bar{\lambda}_{V_l}$ into (2.42), one gets (2.43). Assuming no single node aggregates on the finest level, $\bar{\lambda}_{V_1} = \varrho(A_1)$. ■

Lemma 2.11 Prolongator smoother (2.22) of Algorithm 2.5 given by

$$S_l = (I - P_{A_l, U})(I - \frac{\omega}{\lambda_l} P_{V_l} A_l)(I - P_{A_l, U}),$$

for $l = 1, \dots, L - 1$ satisfies convergence Lemma 2.3 conditions (2.12)- (2.13) on the prolongator smoother.

Proof: On each level $l = 1, \dots, L - 1$, (2.22) is a smoother to which Lemma 2.6 applies. Then, for every level $l = 1, \dots, L - 1$ (2.24) assures (2.12).

Also, Algorithm 2.5 produces spaces U_l and V_l with the following spectral properties (see Remark 2.9, (2.41), and (2.42)): , either $U_l = \emptyset$, or $\lambda_{\min}(U_l) \geq c^2 \bar{\lambda}_l$;

either $V_l = \mathbb{R}^{n_l}$ and $\frac{\varrho(A)}{\lambda_V} \leq 1$, or $V_l \neq \mathbb{R}^{n_l}$ and $\frac{\varrho(A)}{\lambda_V} \leq (1 + C')^2$.

So if $U_l \neq \emptyset$, (2.26) applies with $\frac{1}{c} = \frac{1}{c}$ in (2.27). If $U_l = \emptyset$, $\frac{1}{c} = 0$ in (2.27). Then for any U_l and V_l , (2.27) and (2.42) give (2.13) with

$$C_2 = (1 + C') \left[\frac{1}{c} + \frac{4}{3} (1 + C')^2 \left(\frac{1}{1 + C'} + \frac{1}{c} \right) \right].$$

Here, C' and c are the positive constants from (2.40) and (2.41), respectively. ■

2.6 Tentative Prolongator

Condition (2.11) of Lemma 2.3 on the tentative prolongator is satisfied by construction. The remaining condition (2.10) is also satisfied merely by the properties of the aggregates, or equivalently, the tentative prolongator.

Because of the separate coarsening in the high- and low-coefficient regions, all finest level nodes of $\omega_{l,i}$ are either in the high-coefficient subdomain or in a low-coefficient region if processing is at Stage 1 or Stage 2 on level l , in which case the coefficient applied to coarse-level basis functions to yield the matrix entry is $\frac{1}{\varepsilon^2}$ or 1. If at Stage 3 or Stage 4 on level l , then the high-coefficient region is being eliminated or has already been eliminated, so the coefficient associated with all nodes involved in coarsening is 1.

Algorithm 2.5 yields aggregates satisfying the following assumptions:

2.6.1 Assumptions on aggregates.

For every level l and aggregate \mathcal{A}_i^l , there exists a ball \mathcal{B}_i^l such that

- (1) All nodes of $\tilde{\mathcal{A}}_i^l$ are located within \mathcal{B}_i^l .
- (2) Every $\mathbf{x} \in \Omega$ belongs to at most κ balls \mathcal{B}_i^l .

(3) For each ball \mathcal{B}_i^l

$$\text{diam}(\mathcal{B}_i^l) \leq Ch3^{l-1}, \quad (2.44)$$

where h is the characteristic diameter of the quasiuniform finite element mesh τ_h .

To discuss further properties of the aggregates, the following definition is required.

Definition 2.12 We say that $\Omega \subset \mathbb{R}^d$ is a shape-regular domain of characteristic diameter H if Ω can be mapped onto unit ball $B \subset \mathbb{R}^d$ using a $W^{1,\infty}$ -diffeomorphism F such that

$$\exists C_1, C_2 > 0 : \frac{C_1}{H} \|\mathbf{x}\| \leq \|(\partial F(\tilde{\mathbf{x}})) \cdot \mathbf{x}\| \leq \frac{C_2}{H} \|\mathbf{x}\| \quad (2.45)$$

almost everywhere on Ω . Here, $\partial F(\tilde{\mathbf{x}})$ denotes the Jacobian of F at $\tilde{\mathbf{x}} \in \Omega$.

Remark 2.13 The shape-regularity condition (2.45) is equivalent to Lipschitz conditions

$$\|F(\mathbf{x}) - F(\mathbf{y})\| \leq \frac{C_2}{H} \|\mathbf{x} - \mathbf{y}\| \quad \forall \mathbf{x}, \mathbf{y} \in \Omega$$

and

$$\|F^{-1}(\hat{\mathbf{x}}) - F^{-1}(\hat{\mathbf{y}})\| \leq \frac{H}{C_1} \|\hat{\mathbf{x}} - \hat{\mathbf{y}}\| \quad \forall \hat{\mathbf{x}}, \hat{\mathbf{y}} \in B,$$

or

$$\frac{C_1}{H} \|\mathbf{x} - \mathbf{y}\| \leq \|F(\mathbf{x}) - F(\mathbf{y})\| \leq \frac{C_2}{H} \|\mathbf{x} - \mathbf{y}\| \quad \forall \mathbf{x}, \mathbf{y} \in \Omega.$$

Proof: Follows by the mean value theorem for a C^1 -diffeomorphism, and its extension for a $W^{1,\infty}$ -diffeomorphism using the density of C^1 in $W^{1,\infty}$. ■

Algorithm 2.5 assures aggregates are one of the following two kinds:

2.6.2 Kinds of aggregates

- (1) All nodes of $\tilde{\mathcal{A}}_i^l$ belong to shape-regular $\mathcal{B}_i^l \cap \Omega^\varepsilon$.
- (2) All nodes of $\tilde{\mathcal{A}}_i^l$ are contained in Ω' . The domain $\mathcal{B}_i^l \cap \Omega'$ is shape-regular.

The shape-regularity assumption on these subdomains provides for use of the scaled Poincaré inequality, stated as follows:

Lemma 2.14 Let Ω be a shape-regular domain of characteristic diameter H . Then, for every $u \in H^1(\Omega)$ there is a constant $q \in \mathbb{R}$ such that

$$\|u - q\|_{L^2(\Omega)} \leq C_p H |u|_{H^1(\Omega)}, \quad \forall u \in H^1(\Omega), \quad (2.46)$$

where C_p depends only on shape-regularity constants in (2.45).

Proof: Since Ω is shape-regular, every $u \in H^1(\Omega)$ can be written as $u(\mathbf{x}) = \hat{u}(F(\mathbf{x}))$, where F is a $W^{1,\infty}$ -diffeomorphism satisfying (2.45). Hence, for every $u \in H^1(\Omega)$,

$$\|u\|_{H^1(\Omega)}^2 = \int_B \|(\partial F) \nabla \hat{u}\|_{\mathbb{R}^d}^2 \det(\partial F^{-1}) dB. \quad (2.47)$$

Letting \mathbf{x} be an eigenvector of ∂F , it follows from (2.45) that all eigenvalues $\lambda \in \sigma(\partial F)$ satisfy $|\lambda| \in \left[\frac{C_1}{H}, \frac{C_2}{H}\right]$, hence

$$(\det \partial F)^{-1} = \det(\partial F)^{-1} \in \left[\left(\frac{H}{C_2}\right)^d, \left(\frac{H}{C_1}\right)^d \right].$$

Substituting the last estimate into (2.47) and using (2.45) again gives

$$cH^{d-2} |\hat{u}|_{H^1(B)}^2 \leq |u|_{H^1(\Omega)}^2 \leq CH^{d-2} |\hat{u}|_{H^1(B)}^2, \quad (2.48)$$

with constants $c, C > 0$ that depend exclusively on C_1 and C_2 in (2.45). By the same argument, one also gets

$$\begin{aligned} CH^d \|\hat{u} - q\|_{L^2(B)}^2 &\leq \|u - q\|_{L^2(\Omega)}^2 \\ &= \int_B (\hat{u} - q)^2 \det(\partial F^{-1}) dB \\ &\leq CH^d \|\hat{u} - q\|_{L^2(B)}^2 \quad \forall q \in \mathbb{R}. \end{aligned} \quad (2.49)$$

The proof is now completed using the Poincaré inequality for a unit ball $B[1]$:

$$\forall \hat{u} \in H^1(B) \quad \exists \hat{q} \in \mathbb{R} : \|\hat{u} - \hat{q}\|_{L^2(B)} \leq C|\hat{u}|_{H^1(B)}, \quad (2.50)$$

where C is a Poincaré constant for the unit ball. Indeed, setting $\hat{q} = q$ and substituting (2.48) and (2.49) into (2.50) gives (2.46). \blacksquare

Remark 2.15 The expression on the left-hand side of (2.46) attains its minimum when q is an $L^2(\Omega)$ -orthogonal projection of u onto the space of constant functions. By well-known arguments this projection is given by

$$q_{min} = \frac{(u, 1)_{L^2(\Omega)}}{\|1\|_{L^2(\Omega)}} = \frac{1}{meas(\Omega)} \int_{\Omega} u \, d\Omega. \quad (2.51)$$

We recall several more well-known concepts used in the following proofs.

For the set of finite element basis functions $\{\varphi_i\}$, define the finite element interpolation operator $\Pi : \mathbb{R}^n \rightarrow V_h$, as $\Pi \mathbf{v} = \sum_{i=1}^n v_i \varphi_i$. Then, for stiffness matrix $A_{i,j} = a(\varphi_j, \varphi_i)$,

$$\|\Pi \mathbf{v}\|_{a(\Omega)} = \|\mathbf{v}\|_A. \quad (2.52)$$

Over a domain Ω with quasiuniform finite element triangulation τ_h with characteristic mesh parameter h , the discrete and continuous L^2 norms are equivalent, with

$$ch^{-d} \|\Pi \mathbf{v}\|_{L^2(\Omega)} \leq \|\mathbf{v}\|_{l^2(\mathbb{R}^n)} \leq Ch^{-d} \|\Pi \mathbf{v}\|_{L^2(\Omega)}. \quad (2.53)$$

2.7 Single subdomain with high-coefficient

For the case of a single high-coefficient region with the single-node aggregate being eliminated at level \hat{l} , define a filter on level \hat{l} by a diagonal matrix $N_{\hat{l}}$ such that

$$(N_{\hat{l}})_{ii} = \begin{cases} 0 & \text{if } i = n_{\hat{l}} \\ 1 & \text{otherwise} \end{cases}. \quad (2.54)$$

Since $B = \mathbf{1}$ for the model problem,

$$B^1 = D^{1/2} \mathbf{1}. \quad (2.55)$$

Also by construction, since the tentative prolongator P_{l+1}^l formed when there is a high-coefficient area represented by a single-node aggregate at Stage 3 at level l has no column corresponding to that node, $P_k^l B^k = N_l B^l$ for every $k > \hat{l}$. This means that $P_k^1 B^k = P_i^1 P_k^l B^k = P_i^1 N_l B^l$ whenever $k > \hat{l}$, or

$$(P_k^1 B^k)_i = \begin{cases} (B^1)_i & \text{for node } i \text{ in low-coefficient region} \\ 0 & \text{otherwise.} \end{cases} \quad (2.56)$$

In the case of a single high-coefficient subdomain, we will now show that the tentative prolongators and prolongator smoothers constructed using Algorithm 2.5 on the model problem together satisfy the weak approximation property (2.10). The result is given in two lemmas: one showing the approximation properties of disaggregated coarse-level vectors over regions not being eliminated, satisfied using only assumptions on the aggregation; a second showing that the smoothed disaggregated vectors indeed satisfy the weak approximation property.

Lemma 2.16 For the single region of high-coefficient case, under Assumptions 2.6.1 and 2.6.2 on aggregates, for every $\mathbf{u} \in \mathbb{R}^{n_l}$, $l = 1, \dots, L-1$ we have,

$$\|N_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{\mathbb{R}^{n_l}}^2 \leq \frac{C_A}{\lambda_l} \|\mathbf{u}\|_{D^{-1/2} A D^{-1/2}}^2.$$

with N_l defined by (2.54), and $\tilde{Q}_l = (P_l^1)^T$

Proof: Since P_l^1 is orthogonal, the j^{th} column of P_l^1 associates the j^{th} degree of freedom on level l with the nodes of $\tilde{\mathcal{A}}_j^{l-1}$; and N_l is the identity except on $\text{span}\{\mathbf{e}^{n_l}\}$ if it is the kernel of N_l , we have

$$\begin{aligned} \|N_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{\mathbb{R}^{n_l}}^2 &= \|P_l^1 N_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{\mathbb{R}^{n_l}}^2 \\ &= \sum_{j=1}^{n_l} \|P_l^1 N_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{l^2(\omega_{l,j})}^2 \end{aligned}$$

$$= \sum_{j=1}^M \|(P_l^1 \tilde{Q}_l - P_{l+1}^1 \tilde{Q}_{l+1}) \mathbf{u}\|_{l^2(\omega_{l,j})}^2,$$

where $M = \begin{cases} n_l & \text{if } N_l = I \\ n_l - 1 & \text{otherwise.} \end{cases}$

Since $P_l^1 \tilde{Q}_l$ is the orthogonal projector onto $\text{Range } P_l^1$, and $\text{Range } P_{l+1}^1 \subset \text{Range } P_l^1$,

$$\begin{aligned} \|N_l(\tilde{Q}_l - P_{l+1}^1 \tilde{Q}_{l+1}) \mathbf{u}\|_{\mathbb{R}^{n_l}}^2 &\leq \sum_{j=1}^M (\|\mathbf{u} - P_l^1 \tilde{Q}_l \mathbf{u}\|_{l^2(\omega_{l,j})}^2 \\ &\quad + \|(P_l^1 \tilde{Q}_l - P_{l+1}^1 \tilde{Q}_{l+1}) \mathbf{u}\|_{l^2(\omega_{l,j})}^2) \\ &= \sum_{j=1}^M \|\mathbf{u} - P_{l+1}^1 \tilde{Q}_{l+1} \mathbf{u}\|_{l^2(\omega_{l,j})}^2. \end{aligned}$$

Using the minimization property of orthogonal projection and (2.55), and letting $\mathbf{v} = D^{-1/2} \mathbf{u}$,

$$\begin{aligned} \|N_l(\tilde{Q}_l - P_{l+1}^1 \tilde{Q}_{l+1}) \mathbf{u}\|_{\mathbb{R}^{n_l}}^2 &\leq \sum_{j=1}^M \min_{r_i} \|\mathbf{u} - r_i B^1\|_{l^2(\omega_{l,j})}^2 \\ &= \sum_{j=1}^M \min_{r_i} \|D^{1/2}(\mathbf{v} - r_i \mathbf{1})\|_{l^2(\omega_{l,j})}^2. \end{aligned}$$

The quasiuniformity of τ_h implies that $D_{ii} = A_{ii} \leq C a_i h^{d-2}$ where h is the characteristic mesh diameter, and $a_i = \begin{cases} \frac{1}{\varepsilon^2} & \text{if } \omega_{l,i} \text{ is in high-coefficient region} \\ 1 & \text{otherwise.} \end{cases}$

Then we have

$$\|N_l(\tilde{Q}_l - P_{l+1}^1 \tilde{Q}_{l+1}) \mathbf{u}\|_{\mathbb{R}^{n_l}}^2 \leq C \sum_{j=1}^M \min_{r_i} a_i h^{d-2} \|\mathbf{v} - r_i \mathbf{1}\|_{l^2(\omega_{l,j})}^2.$$

If Π is the finite element projector, using (2.53), (2.46), and (2.44), we have

$$\begin{aligned} \|N_l(\tilde{Q}_l - P_{l+1}^1 \tilde{Q}_{l+1}) \mathbf{u}\|_{\mathbb{R}^{n_l}}^2 &\leq C \sum_{j=1}^M a_i h^{-2} \|\Pi \mathbf{v} - r_i\|_{L^2(\mathcal{B}_j^l \cap \Omega')}^2 \\ &\leq \sum_{j=1}^M 9^{l-1} C_i a_i |\Pi \mathbf{v}|_{H^1(\mathcal{B}_j^l \cap \Omega')}^2 \\ &\leq \sum_{j=1}^M 9^{l-1} C_i |\Pi \mathbf{v}|_{a(\mathcal{B}_j^l \cap \Omega')}^2. \end{aligned}$$

Finally, the assumption of a bounded number of intersections 2.6.1- 2, (2.52), and (2.43) give

$$\begin{aligned}
\|N_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{\mathbf{R}^{n_l}}^2 &\leq C9^{l-1}|\Pi\mathbf{v}|_{a(\Omega)}^2 \\
&\leq C9^{l-1}\|\mathbf{v}\|_A^2 \\
&\leq \frac{C_A}{\lambda_l}\|\mathbf{v}\|_A^2 \\
&= \frac{C_A}{\lambda_l}\|\mathbf{u}\|_{D^{-1/2}AD^{-1/2}}^2,
\end{aligned}$$

concluding the proof. ■

Lemma 2.17 Assumption (2.10) of convergence Lemma 2.3 is satisfied using tentative prolongators P_{l+1}^l and prolongator smoothers (2.22) constructed using Algorithm 2.5 on the model problem, with $\tilde{Q}_l = (P_l^1)^T$.

Proof: Inequality (2.28) holds for the prolongator smoothers (2.22) with the constant dependent on U_l . Using (2.42) and (2.41), we have for any (empty or nonempty) U_l constructed by Algorithm 2.5,

$$\|S_l\| \leq \left(1 + \frac{4}{3}(1 + C')^2\right) \left(1 + \frac{1}{c}(1 + C')\right).$$

where C' and c are constants from (2.40) and (2.41), respectively.

By construction, $N_l = I$ except on $\text{Ker}(S_l)$. Thus, using Lemma 2.16,

$$\begin{aligned}
\|S_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{\mathbf{R}^{n_l}}^2 &= \|S_l N_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{\mathbf{R}^{n_l}}^2 \\
&\leq \|S_l\|^2 \|N_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{\mathbf{R}^{n_l}}^2 \\
&\leq \frac{C_1^2}{\lambda_l}\|\mathbf{u}\|_{A_1}^2,
\end{aligned}$$

with $C_1 = \sqrt{C_A} \left(1 + \frac{4}{3}(1 + C')^2\right) \left(1 + \frac{1}{c}(1 + C')\right)$. ■

Convergence of the method for the model problem now follows from Lemmas 2.11, 2.17, and orthogonality of the tentative prolongator, by the abstract convergence Theorem 2.4.

Theorem 2.18 When used to solve the model problem on a domain with a single high-coefficient region, with components constructed by Algorithm 2.5, and using appropriate R_l , Algorithm 2.1 converges, and satisfies

$$\|\hat{\mathbf{x}} - MG(\mathbf{x}, \mathbf{b})\|_{A_1} \leq \left(1 - \frac{1}{c_0(L)}\right) \|\hat{\mathbf{x}} - \mathbf{x}\|_{A_1} \quad \forall \mathbf{x} \in V_1,$$

where $A_1 = D^{-1/2}AD^{-1/2}$, and $\hat{\mathbf{x}}$ is the solution of (2.3), with $c_1(l) = 1 + C_1(l-1)$, $c_2(l) = C_1 + C_2c_1(l)$, and $c_0(L) = (1 + c_1 + c_2c_R)^2(L-1)$, for MG as in Algorithm 2.1.

Moreover, the preconditioner P defined by the action of $MG(\mathbf{0}, \cdot)$ is symmetric with respect to $(\cdot, \cdot)_{\mathbb{R}^{n_1}}$ and $\text{cond}(A_1, P) \leq c_0(L)$.

2.8 Multiple subdomains with high-coefficients

As suggested in Remark 2.9, in the case of multiple subdomains with high-coefficients Ω_k^ε each subdomain is treated independently by Algorithm 2.5. That is, when an individual subdomain is represented by a single node, it is processed without smoothing until its respective diagonal entry in A_l is large enough in comparison to the (decreasing with coarsening) spectral radius of A_{V_l} . The node is then eliminated. The subspaces and V_l^\perp and U_l are now potentially multi-dimensional.

Define the index set of nodes eliminated at level l by

$$\mathcal{H}_l^{\text{dof}} = \{j : \omega_{l,j} \text{ at Stage 3.}\}$$

Associated with a $\mathcal{H}_l^{\text{dof}}$, define the index set of high-coefficient areas eliminated at level l , by $\mathcal{H}_l^\Omega = \{k : \bar{\Omega}_k^\varepsilon \text{ is represented by the single node } \omega_{l,j} \text{ for some } j \in \mathcal{H}_l^{\text{dof}}\}$. Algorithm 2.5 should yield aggregates with the property: For every $j \in \mathcal{H}_l^{\text{dof}}$, there is a $k \notin \mathcal{H}_l^\Omega, l' < l$ such that all nodes of $\tilde{\mathcal{A}}_i^j$ belong to $\bar{\Omega}_k^\varepsilon$. The domain $\mathcal{B}_j^l \cap \Omega_k^\varepsilon$ is shape-regular.

Lemmas 2.6 and 2.11 are formulated to include the possibility of multiple high-coefficient subdomains. It remains to be shown that the tentative prolongators provide the needed approximation properties. As with a single

high-coefficient subdomain, the weak approximation property can be satisfied without reference to eliminated high-coefficient regions, as Lemmas 2.19 and 2.20 will demonstrate.

To eliminate a high-coefficient node, define a filter on level l by a diagonal matrix N_l , where

$$(N_l)_{ii} = \begin{cases} 0 & \text{if } i \in \mathcal{H}_l^{\text{dof}} \\ 1 & \text{otherwise} \end{cases} \quad (2.57)$$

Algorithm 2.5 applied to the model problem with multiple regions of high-coefficient yields the following environment:

$$P_{l+1}^l B^{l+1} = N_l B^l \quad (2.58)$$

$$P_{l+1}^l = N_l P_{l+1}^l \quad (2.59)$$

$$P_l^1 = N_1 P_2^1 \cdots N_{l-1} P_l^{l-1}. \quad (2.60)$$

These attributes of the tentative prolongator are used below to show an approximation property of the coarse spaces with respect to the energy norm of the scaled matrix A_1 . The bound is without reference to high-coefficient regions corresponding to eliminated nodes.

Lemma 2.19 Assume there is a positive constant $C_{\mathcal{A}}$ such that for every $\mathbf{v} \in \mathbb{R}^{n_1}$ and all levels $l = 1, \dots, L-1$.

$$\sum_{i=1}^{n_{l+1}} \min_{r_i} \|D^{1/2}(\mathbf{v} - r_i B)\|_{l^2(\tilde{\mathcal{A}}_i^l \setminus \omega_l^\varepsilon)}^2 \leq \frac{C_{\mathcal{A}}}{\lambda_l} \|\mathbf{v}\|_A^2, \quad (2.61)$$

where

$$\omega_l^\varepsilon \equiv \left\{ \bigcup_{k,j} \omega_{k,j}, \quad k = 1, \dots, l, \quad j \in \mathcal{H}_k^{\text{dof}} \right\}.$$

Then there is a sequence of linear mappings

$$\tilde{Q}_l : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{n_l}, \quad l = 1, \dots, L, \quad \tilde{Q}_1 = I,$$

such that

$$\|N_l(\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1})\mathbf{u}\|_{\mathbb{R}^{n_l}}^2 \leq \frac{C}{\lambda_l} \|\mathbf{u}\|_{D^{-1/2}AD^{-1/2}}^2 \quad (2.62)$$

holds for all $\mathbf{u} \in \mathbb{R}^{n_l}$, $l = 1, \dots, L-1$.

Proof: For every level $l = 1, \dots, L$, define an $n_l \times n_l$ diagonal matrix \tilde{N}_l by

$$(\tilde{N}_l)_{ii} = \begin{cases} 0 & \text{for } i \in \omega_{l,k}, \quad k \in \mathcal{H}_l^{dof} \\ 1 & \text{otherwise.} \end{cases}$$

Clearly, $\tilde{N}_1 = N_1$. Further, we show that

$$P_l^1 N_l = \tilde{N}_l P_l^1. \quad (2.63)$$

By the definition of N_l one gets for every $\mathbf{x} \in \mathbb{R}^{n_l}$,

$$P_l^1 N_l \mathbf{x} = \sum_j P_l^1 \mathbf{e}^j x_j, \quad j \notin \mathcal{H}_l^{dof},$$

where \mathbf{e}^j is the j^{th} canonical basis vector of \mathbb{R}^{n_l} .

The nonzero structure of P_l^1 then gives

$$(P_l^1 N_l \mathbf{x})_i = 0 \text{ for all } i \in \omega_{l,k}, \quad k \in \mathcal{H}_l^{dof},$$

proving (2.63).

To determine the range of P_l^1 , we first use (2.58) and (2.63):

$$P_l^1 B^l = P_{l-1}^1 P_l^{l-1} B^l = P_{l-1}^1 N_{l-1} B^{l-1} = \tilde{N}_{l-1} (P_{l-1}^1 B^{l-1}).$$

Hence, by induction with $\tilde{N}_1 = N_1$, it follows that

$$P_l^1 B^l = \tilde{N}_1 \dots \tilde{N}_{l-1} B^1. \quad (2.64)$$

The product $\tilde{N}_1 \dots \tilde{N}_{l-1}$ is an $n_l \times n_l$ diagonal matrix,

$$(\tilde{N}_1 \dots \tilde{N}_{l-1})_{ii} = \begin{cases} 0 & \text{for } i \in \omega_{l-1}^\varepsilon \\ 1 & \text{otherwise.} \end{cases} \quad (2.65)$$

Similarly, the zero structure of the rows of P_l^1 is apparent from

$$P_l^1 = \tilde{N}_1 \dots \tilde{N}_{l-1} P_l^1. \quad (2.66)$$

This is shown using (2.59) and (2.63) to give,

$$P_l^1 = P_{l-1}^1 P_l^{l-1} = P_{l-1}^1 N_{l-1} P_l^{l-1} = \tilde{N}_{l-1} P_{l-1}^1 P_l^{l-1};$$

where by the same argument,

$$P_{l-1}^1 = \tilde{N}_{l-2} P_{l-2}^1 P_{l-1}^{l-2}, \text{ etc. and } P_1^1 = I.$$

For each composite aggregate $\tilde{\mathcal{A}}_i^{l-1}$, define a vector $\mathbf{w}^{l,i} \in \mathbb{R}^{n_1}$ by

$$\mathbf{w}^{l,i} = \begin{cases} (\tilde{N}_1 \dots \tilde{N}_{l-1} B^1)_k & \text{for } k \in \tilde{\mathcal{A}}_i^{l-1} \\ 0 & \text{otherwise.} \end{cases} \quad (2.67)$$

Set W_l to be an $n_1 \times n_l$ matrix consisting of the columns $\mathbf{w}^{l,i}$, $i = 1 \dots, n_l$. Then $P_l^1 \mathbf{e}^i$ is the i^{th} column of P_l^1 , and from (2.67), (2.64) and the nonzero structure of P_l^1 , one gets

$$(P_l^1 \mathbf{e}^i) B_i^l = \mathbf{w}^{l,i}.$$

Hence,

$$\text{Range } W_l \subset \text{Range } P_l^1 \quad (2.68)$$

For each level $l < L$, introduce a seminorm

$$|\cdot|_{\tilde{N},l} : \mathbf{x} \in \mathbb{R}^{n_1} \mapsto \|\tilde{N}_1 \dots \tilde{N}_l \mathbf{x}\|_{\mathbb{R}^{n_1}}.$$

Since $D^{1/2} B = B^1$ by construction, and setting $\mathbf{u} = D^{1/2} \mathbf{v}$, we have from (2.65) and (2.67), for every $\mathbf{r} = (r_1, \dots, r_{n_1})^T$

$$\begin{aligned} \sum_{i=1}^{n_{l+1}} \|D^{1/2}(\mathbf{v} - r_i B)\|_{l^2(\tilde{\mathcal{A}}_i^l \setminus \omega_i^{\tilde{\varepsilon}})}^2 &= \sum_{i=1}^{n_{l+1}} \|(\tilde{N}_1 \dots \tilde{N}_l)(\mathbf{u} - r_i B^1)\|_{l^2(\tilde{\mathcal{A}}_i^l)}^2 \\ &= \sum_{i=1}^{n_{l+1}} \|(\tilde{N}_1 \dots \tilde{N}_l)(\mathbf{u} - r_i \mathbf{w}^{l+1,i})\|_{l^2(\tilde{\mathcal{A}}_i^l)}^2 \\ &= |\mathbf{u} - W_{l+1} \mathbf{r}|_{\tilde{N},l}^2 \end{aligned}$$

So, from (2.68) and (2.66)

$$\begin{aligned}
\sum_{i=1}^{n_{l+1}} \min_{r_i} \|D^{1/2}(\mathbf{v} - r_i B)\|_{l^2(\tilde{\mathcal{A}}_i^l \setminus \omega_i^\varepsilon)}^2 &= \min_{\mathbf{r} \in \mathbb{R}^{n_1}} |\mathbf{u} - W_{l+1} \mathbf{r}|_{\tilde{N}, l}^2 \\
&\geq \min_{\mathbf{r} \in \mathbb{R}^{n_{l+1}}} |\mathbf{u} - P_{l+1}^1 \mathbf{r}|_{\tilde{N}, l}^2 \\
&= \min_{\mathbf{r} \in \mathbb{R}^{n_{l+1}}} \|(\tilde{N}_1 \dots \tilde{N}_l) \mathbf{u} - P_{l+1}^1 \mathbf{r}\|_{\mathbb{R}^A}^2. \tag{2.69}
\end{aligned}$$

Let us set $\tilde{Q}_l = (P_l^1)^T$, $l = 1, \dots, L$. Since $(P_l^1)^T P_l^1 = I$, the mappings $P_l^1 \tilde{Q}_l$ are projections onto $\text{Range } P_l^1$, orthogonal with respect to the Euclidean inner product. We now estimate using (2.69), well-known properties of orthogonal projections, $\text{Range } P_{l+1}^1 \subset \text{Range } P_l^1$, and (2.66), (2.63), and (2.59):

$$\begin{aligned}
\sum_{i=1}^{n_{l+1}} \min_{r_i} \|D^{1/2}(\mathbf{v} - r_i B)\|_{l^2(\tilde{\mathcal{A}}_i^l \setminus \omega_i^\varepsilon)}^2 &\geq \|(I - P_{l+1}^1 \tilde{Q}_{l+1}) \tilde{N}_1 \dots \tilde{N}_l \mathbf{u}\|_{\mathbb{R}^{n_1}}^2 \\
&= \|(I - P_l^1 \tilde{Q}_l) \tilde{N}_1 \dots \tilde{N}_l \mathbf{u}\|_{\mathbb{R}^{n_1}}^2 \\
&\quad + \|P_l^1 (\tilde{Q}_l - P_{l+1}^1 \tilde{Q}_{l+1}) \tilde{N}_1 \dots \tilde{N}_l \mathbf{u}\|_{\mathbb{R}^{n_1}}^2 \\
&\geq \|P_l^1 [(\tilde{N}_1 \dots \tilde{N}_l P_l^1)^T \\
&\quad - P_{l+1}^l (\tilde{N}_1 \dots \tilde{N}_l P_{l+1}^1)^T] \mathbf{u}\|_{\mathbb{R}^{n_1}}^2 \\
&= \|P_l^1 [(P_l^1 N_l)^T - N_l P_{l+1}^l (P_{l+1}^1)^T] \mathbf{u}\|_{\mathbb{R}^{n_1}}^2 \\
&= \|P_l^1 N_l (\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1}) \mathbf{u}\|_{\mathbb{R}^{n_1}}^2 \\
&= \|N_l (\tilde{Q}_l - P_{l+1}^l \tilde{Q}_{l+1}) \mathbf{u}\|_{\mathbb{R}^{n_1}}^2.
\end{aligned}$$

Since $\|\mathbf{v}\|_A^2 = \|\mathbf{u}\|_{D^{-1/2} A D^{-1/2}}^2$, the conclusion now follows. \blacksquare

For the model problem, the relevant kernel consists of the vector of ones. Algorithm 2.5 constructs tentative prolongators which satisfy the assumption of the preceding Lemma 2.19.

Lemma 2.20 Under Assumptions 2.6.1 and 2.6.2, there is a constant $C > 0$ such that for every $u \in \mathbb{R}^{n_1}$ and every level $l = 1, \dots, L - 1$ it holds that

$$\sum_{i=1}^{n_{l+1}} \min_{r_i} \|D^{1/2}(\mathbf{v} - r_i \mathbf{1})\|_{l^2(\tilde{\mathcal{A}}_i^l \setminus \omega_i^\varepsilon)}^2 \leq C \frac{C_A}{\lambda_l} \|\mathbf{v}\|_A^2.$$

Proof: Let \mathcal{A}_i^l be an aggregate of the first kind. Then, there is a domain Ω_k^ε with coefficient ε_k such that all nodes of $\tilde{\mathcal{A}}_i^l$ belong to Ω_k^ε . Hence $D_{jj} \leq C\varepsilon_k^{-2}h^{d-2}$ for all $j \in \tilde{\mathcal{A}}_i^l \setminus \omega_i^\varepsilon = \tilde{\mathcal{A}}_i^l$ and,

$$\begin{aligned} \|D^{1/2}(\mathbf{v} - r_i \mathbf{1})\|_{L^2(\tilde{\mathcal{A}}_i^l \setminus \omega_i^\varepsilon)}^2 &= \|D^{1/2}(\mathbf{v} - r_i \mathbf{1})\|_{L^2(\tilde{\mathcal{A}}_i^l)}^2 \\ &\leq C\varepsilon_k^{-2}h^{d-2} \|\mathbf{v} - r_i\|_{L^2(\tilde{\mathcal{A}}_i^l)}^2 \\ &\leq C\varepsilon_k^{-2}h^{-2} \|\Pi \mathbf{v} - r_i\|_{L^2(\mathcal{B}_i^l \cap \Omega_k^\varepsilon)}^2. \end{aligned}$$

Choosing r_i to be an integral average of $\Pi \mathbf{v}$ over $\mathcal{B}_i^l \cap \Omega_k^\varepsilon$,

$$r_i = \frac{1}{\text{meas}(\mathcal{B}_i^l \cap \Omega_k^\varepsilon)} \int_{\mathcal{B}_i^l \cap \Omega_k^\varepsilon} (\Pi \mathbf{v}) d\mathbf{x},$$

the scaled Poincaré inequality (2.46) together with Assumption 2.6.1- 3 gives

$$\|\Pi \mathbf{v} - r_i\|_{L^2(\mathcal{B}_i^l \cap \Omega_k^\varepsilon)}^2 \leq C9^{l-1}h^2 |\Pi \mathbf{v}|_{H^1(\mathcal{B}_i^l \cap \Omega_k^\varepsilon)}^2.$$

Therefore,

$$\begin{aligned} \min_{r_i} \|D^{1/2}(\mathbf{v} - r_i \mathbf{1})\|_{L^2(\tilde{\mathcal{A}}_i^l \setminus \omega_i^\varepsilon)}^2 &\leq C9^{l-1}\varepsilon_k^{-2} |\Pi \mathbf{v}|_{H^1(\mathcal{B}_i^l \cap \Omega_k^\varepsilon)}^2 \\ &\leq C9^{l-1} |\Pi \mathbf{v}|_{a(\mathcal{B}_i^l \cap \Omega_k^\varepsilon)}^2. \end{aligned} \quad (2.70)$$

If $\tilde{\mathcal{A}}_i^l$ is an aggregate of the second kind, all nodes $j \in \tilde{\mathcal{A}}_i^l \setminus \omega_i^\varepsilon$ are located inside the part of Ω where the coefficient is equal to one. Hence,

$$\begin{aligned} \|D^{1/2}(\mathbf{v} - r_i \mathbf{1})\|_{L^2(\tilde{\mathcal{A}}_i^l \setminus \omega_i^\varepsilon)}^2 &\leq Ch^{d-2} \|\mathbf{v} - r_i \mathbf{1}\|_{L^2(\tilde{\mathcal{A}}_i^l)}^2 \\ &\leq Ch^{-2} \|\Pi \mathbf{v} - r_i\|_{L^2(\mathcal{B}_i^l \cap \Omega')}^2. \end{aligned}$$

Again setting r_i to be an integral average of $\Pi \mathbf{v}$ over $\mathcal{B}_i^l \cap \Omega_k^\varepsilon$, using the scaled Poincaré inequality (2.46), and Assumption 2.6.1- 3, we get

$$\begin{aligned} \min_{r_i} \|D^{1/2}(\mathbf{v} - r_i \mathbf{1})\|_{L^2(\tilde{\mathcal{A}}_i^l \setminus \omega_i^\varepsilon)}^2 &\leq C9^{l-1} |\Pi \mathbf{v}|_{H^1(\mathcal{B}_i^l \cap \Omega')}^2 \\ &\leq C9^{l-1} |\Pi \mathbf{v}|_{a(\mathcal{B}_i^l \cap \Omega')}^2. \end{aligned} \quad (2.71)$$

From (2.70) and (2.71), the conclusion follows from the bounded intersections of the balls 2.6.1- 2, (2.52), and (2.43). \blacksquare

Lemma 2.17 is seen to hold for the multiple region of high-coefficient case, although N_l can now have a multidimensional kernel. As for the case of the single high-coefficient subdomain, convergence of Algorithm 2.1 for the model problem now follows from Lemmas 2.11, 2.17, and orthogonality of the tentative prolongator, using abstract convergence Theorem 2.4. Thus we have:

Theorem 2.21 When used to solve the model problem on a domain with multiple high-coefficient regions, using components constructed by Algorithm 2.5 and appropriate R_l , Algorithm 2.1 converges, satisfying

$$\|\hat{\mathbf{x}} - MG(\mathbf{x}, \mathbf{b})\|_{A_1} \leq \left(1 - \frac{1}{c_0(L)}\right) \|\hat{\mathbf{x}} - \mathbf{x}\|_{A_1} \quad \forall \mathbf{x} \in V_1,$$

where $A_1 = D^{-1/2}AD^{-1/2}$, and $\hat{\mathbf{x}}$ is the solution of (2.3), with $c_1(l) = 1 + C_1(l-1)$, $c_2(l) = C_1 + C_2c_1(l)$, and $c_0(L) = (1 + c_1 + c_2c_R)^2(L-1)$, for MG as in Algorithm 2.1.

Moreover, the preconditioner P defined by the action of $MG(\mathbf{0}, \cdot)$ is symmetric with respect to $(\cdot, \cdot)_{\mathbb{R}^{n_1}}$ and $\text{cond}(A_1, P) \leq c_0(L)$.

2.9 Computational Experiments

In this section, we attempt to demonstrate on several model examples the efficacy of the multilevel method based on the smoothed aggregation approach. The problem solved is (2.16) with domain Ω a unit cube. A Dirichlet boundary condition will be considered over the $x = 0$ face of the cube.

All experiments were run using V-cycles. The iteration of the method was terminated when the Euclidean norm of the initial residual had been reduced by $\varepsilon = 10^{-12}$. In order to observe the behavior of the true error, zero right-hand side and a random initial approximation were chosen. All problems were discretized on 125 subdomains.

We show results for AMG used as a solver, and as a preconditioner. The meshsize is varied so that in each table, each experiment is shown using $20 \times 20 \times 20$, $40 \times 40 \times 40$, and $80 \times 80 \times 80$ elements. Experiments are shown with coefficients varying from $10^{-\sigma}$ to 10^σ for each problem size, with

Table 2.1. Checkerboard pattern, coefficients 10^σ , $10^{-\sigma}$.

dof.	σ	iter	conv. ratio	iter PCG	conv. ratio
9,261	0	12	8.847e-02	10	5.991e-02
9,261	3	11	7.026e-02	11	6.169e-02
9,261	6	14	1.335e-01	12	9.377e-02
68,921	0	15	1.429e-01	10	5.301e-02
68,921	3	12	8.551e-02	10	5.783e-02
68,921	6	13	1.130e-01	13	1.110e-01
531,441	0	15	1.462e-01	10	5.980e-02
531,441	3	14	1.328e-01	12	8.281e-02
531,441	6	18	2.104e-01	18	1.913e-01

the Laplace problem, $\sigma = 0$, listed first in each case. The pattern of the coefficients is a checkerboard in table 2.1, cubes touching at a node in table 2.2, and randomly distributed in table 2.3. Convergence rates are the average of the rates measured on each level.

We note that the method practically used did not strictly adhere to all the assumptions of the theory. The method tends to coarsen separately over regions with very different coefficients, although this is not explicitly enforced.

Table 2.2. Two cubes touching at a node. Coefficient= 10^σ in dark subregions (see Figure 2.1).

dof.	σ	iter	conv. ratio	iter PCG	conv. ratio
9,261	0	12	8.847e-02	10	5.991e-02
9,261	3	13	1.066e-01	11	6.619e-02
9,261	6	14	1.344e-01	11	7.493e-02
68,921	0	15	1.429e-01	10	5.301e-02
68,921	3	16	1.670e-01	11	6.354e-02
68,921	6	16	1.720e-01	11	6.255e-02
531,441	0	15	1.462e-01	10	5.980e-02
531,441	3	15	1.440e-01	11	7.639e-02
531,441	6	15	1.444e-01	11	7.874e-02

2.10 Conclusion

The variant of smoothed aggregation AMG described here is shown to converge for the scalar model problem with a single high-coefficient subdomain. The method is also demonstrated to converge when applied to the problem with multiple high-coefficient subdomains, under certain assumptions. Computational results are given, indicating that even without strict enforcement of the strategies presented here, smoothed aggregations AMG can perform well on problems with discontinuous coefficients.

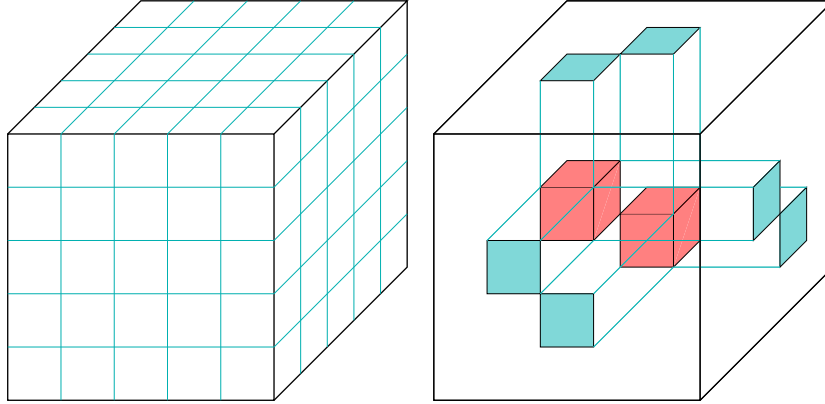


Figure 2.1. Configuration with two high-coefficient subregions touching at a single node.

Table 2.3. Element coefficients random in $(10^{-\sigma}, 10^{\sigma})$.

dof.	σ	iter	conv. ratio	iter PCG	conv. ratio
9,261	0	12	8.847e-02	10	5.991e-02
9,261	3	12	9.871e-02	11	6.415e-02
9,261	6	12	9.871e-02	11	6.430e-02
68,921	0	15	1.429e-01	10	5.301e-02
68,921	3	15	1.511e-01	10	5.506e-02
68,921	6	15	1.511e-01	10	5.530e-02
531,441	0	15	1.462e-01	10	5.980e-02
531,441	3	15	1.440e-01	11	7.365e-02
531,441	6	16	1.624e-01	11	6.921e-02

References

- [1] R.A. Adams. *Sobolev Spaces*. Academic Press, New York, 1975.
- [2] T. Arbogast, A. Chilakapati, and M. F. Wheeler. A characteristic-mixed method for contaminant transport and miscible displacement. In T. Russell et al., editor, *Computational Methods in Water Resources IX*, volume 1, chapter Numerical Methods in Water Resources, pages 77–84. Computational Mechanics Publications and Elsevier Applied Science, London, New York, 1992.
- [3] T. Arbogast and M. F. Wheeler. A characteristics-mixed finite element method for advection-dominated transport problems. *SIAM J. Numer. Anal.*, pages 404–424, 1995.
- [4] N. S. Bakhvalov and A. V. Knyazev. Fictitious domain methods and computation of homogenized properties of composites with a periodic structure of essentially different components. In Gury I. Marchuk, editor, *Numerical Methods and Applications*, pages 221–276. CRC Press, Boca Raton, 1994.
- [5] N. S. Bakhvalov and A. V. Knyazev. Preconditioned iterative methods in a subspace for linear algebraic equations with large jumps in the coefficients. In D. Keyes and J. Xu, editors, *Domain Decomposition Methods in Science and Engineering*, volume 180 of *Contemporary Mathematics*, pages 157–162. American Mathematical Society, Providence, 1994. Proceedings of the Seventh International Conference on Domain Decomposition, October 27–30, 1993, held at the Pennsylvania State University.
- [6] N. S. Bakhvalov, A. V. Knyazev, and G. M. Kobel'kov. Iterative methods for solving equations with highly varying coefficients. In Roland Glowinski, Yuri A. Kuznetsov, Gérard A. Meurant, Jacques Périaux, and Olof Widlund, editors, *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 197–205, Philadelphia, PA, 1991. SIAM.

- [7] N. S. Bakhvalov, A. V. Knyazev, and R. R. Parashkevov. An efficient iterative method for solving Lamé equations for nearly incompressible media and Stokes equations with highly discontinuous coefficients. *Numerische Mathematik*. Submitted. Published as a technical report UCD-CCM 120, 1997, at the Center for Computational Mathematics, University of Colorado at Denver.
- [8] R. Bank, J. Bürger, W. Fichtner, and R. Smith. Some upwinding techniques for finite element approximations of convection diffusion equations. *Numer. Math.*, pages 185–202, 1990.
- [9] R. E. Bank, J. Mandel, S. F. McCormick, and R. E. Bank. Variational multigrid theory. In *Multigrid Methods*, pages 131–178. SIAM, Philadelphia, 1987.
- [10] J. W. Barrett and K. W. Morton. Approximate symmetrization and Petrov-Galerkin methods for diffusion-convection problems. *Comput. Methods Appl. Mech. Engrg.*, pages 97–122, 1984.
- [11] J. P. Benque and J. Ronat. Quelques difficultés des modèles numériques en hydraulique. *Comput. Methods Appl. Mech. Engrg.*, pages 471–494, 1996.
- [12] P. J. Binning. *Modeling unsaturated zone flow and contaminant in the air and water phases*. PhD thesis, Dept. of Civil Engineering and Operational Research, Princeton University, Princeton, New Jersey, 1994. Ph.D. thesis.
- [13] P. J. Binning and M. A. Celia. A finite volume Eulerian-Lagrangian localized adjoint method for solution of the contaminant transport equations in two-dimensional multi-phase flow systems. *Water Resources Research*, pages 103–114, 1996.
- [14] Albrecht Böttcher and Bernard Silberman. *Introduction to Large Truncated Toeplitz Matrices*. Springer-Verlag, New York, 1999.
- [15] E. T. Bouloutas and M. A. Celia. An analysis of some classes of Petrov-Galerkin and optimal test function methods. In M. A. Celia et al., editor, *Proceedings of Seventh International Conference on Computational Methods in Water Resources*, volume 2, pages 15–20. Computational Mechanics Publications, Southampton, England, 1988.

- [16] E. T. Bouloutas and M. A. Celia. An improved cubic Petrov-Galerkin method for simulation of transient advection-diffusion processes in rectangularly decomposable domains. *Comput. Methods Appl. Mech. Engrg.*, pages 289–308, 1991.
- [17] J.H. Bramble, J.E. Pasciak, J. Wang, and J. Xu. Convergence estimates for multigrid algorithm without regularity assumptions. *Math. Comp.*, 57:23–45, 1991.
- [18] A. Brandt, S. F. McCormick, and J. W. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In D. J. Evans, editor, *Sparsity and Its Applications*. Cambridge University Press, Cambridge, 1984.
- [19] Marian Brezina and Petr Vaněk. A black-box iterative solver based on a two-level Schwarz method. *Computing*, 63:233–263, 1999. Dec 07, 1999.
- [20] A. N. Brooks and T. J. R. Hughes. Streamline upwind Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, pages 199–259, 1982.
- [21] R. D. Burnett and E. O. Frind. Simulation of contaminant transport in three dimensions, 2. dimensionality effects. *Water Resources Research*, 23:695–705, 1987.
- [22] C. S. Carrano and G. T. Yeh. A fourier analysis and optimization of the Petrov-Galerkin finite element method. In Alexander Peters et al., editor, *Computational Methods in Water Resources X*, pages 191–198. Kluwer Academic Press, London, 1994.
- [23] M. A. Celia. Eulerian-Lagrangian localized adjoint methods for contaminant transport simulations. In Alexander Peters et al., editor, *Computational Methods in Water Resources X*, pages 207–216. Kluwer Academic Press, London, 1994.
- [24] M. A. Celia and L. A. Ferrand. A comparison of ELLAM formulations for simulation of reactive transport in groundwater. In Wang, editor, *Advances in Hydro-Science and Engineering, 1(B)*, pages 1829–1836. University of Mississippi Press, Jackson, MS, 1993.

- [25] M. A. Celia, I. Herrera, E. T. Bouloutas, and J. S. Kindred. A new numerical approach for the advective-diffusive transport equation. *Numer. Methods Partial Differential Equations*, pages 203–226, 1989.
- [26] M. A. Celia, T. F. Russell, I. Herrera, and R. E. Ewing. An Eulerian-Lagrangian localized adjoint method for the advection-dispersion equation. *Advances in Water Resources*, 13:187–206, 1990.
- [27] M. A. Celia and S. Zisman. An Eulerian-Lagrangian localized adjoint method for reactive transport in groundwater. In G. Gambolati et al., editor, *Computational Methods in Water Resources VII*, pages 383–392. Springer, New York, 1990.
- [28] I. Christie, D. F. Griffiths, A. R. Mitchell, and O. C. Zienkiewicz. Finite element methods for second order differential equations with significant first derivatives. *Internat. J. Numer. Methods Engrg.*, pages 1389–1396, 1976.
- [29] R. A. Cox and T. Nishikawa. A new total variation diminishing scheme for the solution of advective-dominant solute transport. *Water Resources Research*, pages 2645–2654, 1991.
- [30] H. K. Dahle, M. S. Espedal, R. E. Ewing, and O. SÆvareid. Characteristic adaptive subdomain methods for reservoir flow problems. *Numer. Methods Partial Differential Equations*, pages 279–309, 1990.
- [31] H. K. Dahle, R. E. Ewing, and T. F. Russell. Eulerian-Lagrangian localized adjoint methods for a nonlinear convection-diffusion equation. *Comput. Methods Appl. Mech. Engrg.*, pages 223–250, 1995.
- [32] L. Demkowicz and J. T. Oden. An adaptive characteristic Petrov-Galerkin finite element for convection-dominated linear and nonlinear parabolic problems in two space variables. *Comput. Methods Appl. Mech. Engrg.*, pages 63–87, 1986.
- [33] R. E. Ewing. Operator splitting and Eulerian-Lagrangian localized adjoint methods for multiphase flow. In *The Mathematics of Finite Elements and Applications*, pages 215–232. Academic Press, London, 1991.

- [34] R. E. Ewing and H. Wang. Eulerian-Lagrangian localized adjoint methods for linear advection equations. In *Comput. Mech. '91*, pages 245–250. Springer, Berlin, 1991.
- [35] R. E. Ewing and H. Wang. An Eulerian-Lagrangian localized adjoint method for variable-coefficient advection-reaction problems. In Wang, editor, *Advances in Hydro-Science and Engineering, 1(B)*, pages 2010–2015. University of Mississippi Press, Jackson, MS, 1993.
- [36] R. E. Ewing and H. Wang. An Eulerian-Lagrangian localized adjoint method with exponential-along-characteristic test functions for variable-coefficient advective-diffusive-reactive equations. In U. Choi, D. Kwak, and J. Yim, editors, *Proceedings of KAIST Mathematical Workshop. Analysis and Geometry, 8*, pages 77–91. Teajon, Korea, 1993.
- [37] R. E. Ewing and H. Wang. Eulerian-Lagrangian localized adjoint methods for variable-coefficient advective-diffusive-reactive equations in groundwater contaminant transport. In Gomez and Hennart, editors, *Advances in Optimization and Numerical Analysis, Mathematics and Its Applications, Vol. 275*, pages 185–205. Kluwer Academic Publishers, Dordrecht, Netherlands, 1994.
- [38] R. E. Ewing and H. Wang. An optimal order error estimate for Eulerian-Lagrangian localized adjoint methods for variable-coefficient advection-reaction problems. *SIAM J. Numer. Anal.*, pages 318–348, 1996.
- [39] K. Eriksson and C. Johnson. Adaptive streamline diffusion finite element methods for stationary convection-diffusion problems. *J. Math. Comp.*, pages 167–188, 1993.
- [40] M. S. Espedal and R. E. Ewing. Characteristic Petrov-Galerkin subdomain methods for two-phase immiscible flow. *Comput. Methods Appl. Mech. Engrg.*, pages 113–135, 1987.
- [41] R. E. Ewing and M. A. Celia. Numerical methods for reactive transport and biodegradation. In T. F. Russell et al., editor, *Computational Methods in Water Resources IX*, pages 51–58. Computational Mechanics Publications, Southampton, England, 1992.
- [42] R. E. Ewing, T. F. Russell, and M. F. Wheeler. Simulation of miscible

- displacement using mixed methods and a modified method of characteristics. *Society of Petroleum Engineers of AIME*, pages 71–81, 1983.
- [43] R. E. Ewing and H. Wang. Eulerian-Lagrangian localized adjoint methods for linear advection or advection-reaction equations and their convergence analysis. *Comput. Mech.*, pages 97–121, 1993.
- [44] J. Fish and V. Belsky. Generalized aggregation multilevel solver. *Internat. J. Numer. Methods Engrg.*, 40(23):4341–4361, 1997.
- [45] Jacob Fish and Vladimir Belsky. Multigrid method for periodic heterogeneous media. II. Multiscale modeling and quality control in multidimensional case. *Comput. Methods Appl. Mech. Engrg.*, 126(1-2):17–38, 1995.
- [46] L. P. Franca, S. L. Frey, and T. J. R. Hughes. Stabilized finite element methods: I. application to the advective-diffusive model. *Comput. Methods Appl. Mech. Engrg.*, pages 253–276, 1992.
- [47] A. O. Garder, D. W. Peaceman, and A. L. Pozzi. Numerical calculation of multidimensional miscible displacement by the method of characteristics. *Soc. Petroleum Eng. Jour.*, 4:26–36, 1964.
- [48] A. O. Garder, D. W. Peaceman, and A. L. Pozzi. Numerical calculations of multidimensional miscible displacement by the method of characteristics. *Soc. Pet.. Engrg. J.*, pages 26–36, 1964.
- [49] Karl E. Gustafson and Duggirala K. M. Rao. *Numerical Range*. Springer-Verlag, New York, 1996.
- [50] P. Hansbo. The characteristic streamline diffusion method for the time-independent incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, pages 171–186, 1992.
- [51] P. Hansbo and A. Szepessy. A velocity-pressure streamline diffusion finite element method for the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, pages 107–129, 1990.
- [52] A. W. Harbaugh and M. G. McDonald. A modular three-dimensional finite-difference ground-water flow model. Techniques of Water-Resources Investigations Book 6, Chapter A1, U.S. Geological Survey, 1988.

- [53] A. W. Harbaugh and M. G. McDonald. Programmer's documentation for MODFLOW-96, an update to the U.S. Geological Survey modular finite-difference ground-water flow model. Open-File Report 96-486, U.S. Geological Survey, 1996.
- [54] A. W. Harbaugh and M. G. McDonald. User's documentation for MODFLOW-96, an update to the U.S. Geological Survey modular finite-difference ground-water flow model. Open-File Report 96-485, U.S. Geological Survey, 1996.
- [55] R. W. Healy and T. F. Russell. A finite-volume Eulerian-Lagrangian localized adjoint method for solution of the advection-dispersion equation. *Water Resources Research*, 29:2399–2413, 1993.
- [56] R. W. Healy and T. F. Russell. Solution of the advection-dispersion equation in two dimensions by a finite-volume Eulerian-Lagrangian localized adjoint method. *Advances in Water Resources*, pages 11–26, 1998.
- [57] C. I. Heberton, T. F. Russell, L. F. Konikow, and G. Z. Hornberger. A three-dimensional finite-volume Eulerian-Lagrangian localized adjoint method (ELLAM) for solute-transport modeling. Water Resources Investigations Report 00-xxxx, U.S. Geological Survey, 2000. awaiting publication.
- [58] I. Herrera, R. E. Ewing, M. A. Celia, and T. F. Russell. Eulerian-Lagrangian localized adjoint methods: The theoretical framework. *Numer. Methods Partial Differential Equations*, pages 431–458, 1993.
- [59] J. M. Hervouet. Application of the method of characteristics in their weak approximation to solving two-dimensional advection equations on mesh grids. In C. Taylor, J. John, and W. Smith, editors, *Computational Techniques for Fluid Flow, Recent Advances in Numerical Methods in Fluids 5*. SIAM Pineridge Press, Swansea, 1986.
- [60] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, New York, and Oakleigh, Victoria, Australia, 1985.
- [61] P. A. Hsieh. A new formula for the analytical solution of the radial dispersion problem. *Water Resources Research*, 22:1597–1605, 1986.

- [62] T. J. R. Hughes. Multiscale phenomena: Green functions, the dirichlet-to-neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods. *Comput. Methods Appl. Mech. Engrg.*, pages 387–401, 1995.
- [63] T. J. R. Hughes and A. N. Brooks. A multidimensional upwinding scheme with no crosswind diffusion. In *Finite Element Methods for Convection Dominated Flows 34*. ASME, New York, 1979.
- [64] T. J. R. Hughes and M. Mallet. A new finite element formulation for computational fluid dynamics: Iii. the general streamline operator for multidimensional advective-diffusive systems. *Comput. Methods Appl. Mech. Engrg.*, pages 305–328, 1986.
- [65] Jr. J. Douglas and T. F. Russell. Numerical methods for convection-dominated diffusion problems based combining the method of characteristics with finite element or finite difference procedures. *SIAM J. Numer. Anal.*, pages 871–885, 1982.
- [66] C. Johnson. *Numerical Solutions of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, Cambridge, UK, 1987.
- [67] C. Johnson and U. Nävert. An analysis of some finite methods for advection-diffusion equations. In O. Axelsson, L. Frank, and A. van Sluis, editors, *Analytical and Numerical Approaches to Asymptotic Problems in Analysis*. North-Holland, Amsterdam, 1981.
- [68] C. Johnson and A. Szepessy. Adaptive finite element methods for conservation laws based on a posteriori error estimates. *Comm. Pure Appl. Math.*, pages 199–234, 1995.
- [69] C. Johnson, A. Szepessy, and P. Hansbo. On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws. *J. Math. Comp.*, pages 107–129, 1990.
- [70] K. L. Kipp, L. F. Konikow, and G. Z. Hornberger. An implicit dispersive transport algorithm for the U.S. Geological Survey MOC3D solute transport model. Water Resources Investigations Report 98-4234, U.S. Geological Survey, 1998.

- [71] L. F. Konikow, D. J. Goode, and G. Z. Hornberger. A three-dimensional method-of-characteristics solute-transport model (MOC3D). Water Resources Investigations Report 96-4267, U.S. Geological Survey, 1996.
- [72] Erwin Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley & Sons, Inc., New York, 1978.
- [73] D.A. Lavis and B.W. Southern. The inverse of a symmetric banded toeplitz matrix. *Reports on Mathematical Physics*, 39:137–146, 1997.
- [74] G. Lube and D. Weiss. Stabilized finite element methods for singularly perturbed parabolic problems. *Appl. Numer. Math.*, pages 431–459, 1995.
- [75] Jan Mandel, Marian Brezina, and Petr Vaněk. Energy optimization of algebraic multigrid bases. *Computing*, 62:205–228, 1999. June 22, 1999.
- [76] K. W. Morton, A. Priestly, and E. Süli. Stability of the lagrangian-galerkin method with nonexact integration. *Model. Math. Anal. Numer.*, pages 625–653, 1988.
- [77] S. P. Neuman. Adjoint Petrov-Galerkin method with optimal weight and interpolation functions defined on multi-dimensional nested grids. In G. Gambolati et al., editor, *Computational Methods in Water Resources VII*, pages 347–356. Springer, New York, 1990.
- [78] G. F. Pinder and H. H. Cooper. A numerical technique for calculating the transient position of the saltwater front. *Water Resources Research*, pages 875–882, 1970.
- [79] O. Pironneau. On the transport diffusion algorithm and its application to the Navier-Stokes equations. *Numer. Math*, pages 309–332, 1982.
- [80] J. W. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. F. McCormick, editor, *Multigrid Methods*, volume 3 of *Frontiers in Applied Mathematics*, pages 73–130. SIAM, Philadelphia, PA, 1987.
- [81] T. F. Russell. Eulerian-Lagrangian localized adjoint methods for advection-diffusion problems. In D. Griffiths and G. Watson, editors, *Proceedings of the 13th Dundee Conference on Numerical Analysis, 1989*, *Pitman Res. Notes Math, Ser. 228*, pages 206–228. Longman Scientific and Technical, Harlow, UK, 1990.

- [82] T. F. Russell and R. V. Trujillo. Eulerian-Lagrangian localized adjoint methods with variable coefficients in multiple dimensions. In G. Gambolati, editor, *Computational Methods in Surface Hydrology*, pages 357–363. Springer, Berlin, 1990.
- [83] T. F. Russell, M. F. Wheeler, and C. Chiang. Large-scale simulation of miscible displacement by mixed and characteristic finite elements. In W. E. Fitzgibbon, editor, *Mathematical and Computation Methods in Seismic Exploration and Reservoir Modeling*. SIAM, Philadelphia, 1986.
- [84] M. Stynes and T. F. Russell. An optimal-order estimate for the finite volume Eulerian-Lagrangian localized adjoint method for advection problems.
- [85] J. E. Våg, H. Wang, and H. K. Dahle. Eulerian-Lagrangian localized adjoint methods for systems of nonlinear advective-diffusive-reactive equations. *Advances in Water Resources*, pages 297–315, 1996.
- [86] P. Vaněk. Acceleration of convergence of a two-level algorithm by smoothing transfer operator. *Applications of Mathematics*, 37:265–274, 1992.
- [87] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56:179–196, 1996.
- [88] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56:179–196, 1996.
- [89] Petr Vaněk, Marian Brezina, and Jan Mandel. Convergence of algebraic multigrid based on smoothed aggregation. 2000. To appear in *Numer. Math.*
- [90] Petr Vaněk, Marian Brezina, and Radek Tezaur. Two-grid method for linear elasticity on unstructured meshes. *SIAM J. Sci. Comp.*, 21(3):900–923, 1999.
- [91] Petr Vaněk, Jan Mandel, and Marian Brezina. Algebraic multigrid on unstructured meshes. Technical Report 34, UCD/CCM, 1994.
- [92] Petr Vaněk, Jan Mandel, and Marian Brezina. Two-level algebraic

- multigrid for the Helmholtz problem. In Jan Mandel, Charbel Farhat, and Xiao-Chuan Cai, editors, *Domain Decomposition Methods in Science and Engineering*, pages 349–356. American Mathematical Society, Providence, RI, 1998. Proceedings of the 10th International Symposium on Domain Decomposition Methods, Boulder, Colorado, 1997.
- [93] E. Varoglu and W. D. L. Finn. Finite elements incorporating characteristics for one-dimensional diffusion-convection equation. *J. Comp. Phys*, pages 371–389, 1980.
- [94] H. Wang. *Eulerian-Lagrangian localized adjoint methods: analyses, numerical implementations and their applications*. PhD thesis, University of Wyoming, Laramie, Wyoming, 1992. Ph.D. thesis.
- [95] H. Wang and R. E. Ewing. Optimal order convergence rates for ELLAM for reactive transport and contamination in groundwater. *Numer. Methods Partial Differential Equations*, pages 1–31, 1995.
- [96] H. Wang, R. E. Ewing, and T. F. Russell. Eulerian-Lagrangian localized adjoint methods for convection-diffusion equations and their convergence analysis. *IMA J. Numer. Anal.*, pages 405–459, 1995.
- [97] H. Wang, R. C. Sharpley, and S. Man. An ELLAM scheme for advection-diffusion equations in multi-dimensions. In Aldama et al., editor, *Computational Methods in Water Resources XI*, pages 99–106. Computational Mechanics Publications, Southampton, England, 1996.
- [98] J. J. Westerink and D. Shea. Consider higher degree Petrov-Galerkin methods for the solution of the transient convection-diffusion equation. *Internat. J. Numer. Methods Engrg.*, pages 1077–1101, 1989.
- [99] E. J. Wexler. Analytical solutions for one-, two-, and three-dimensional solute transport in ground-water systems with uniform flow. Techniques of Water-Resources Investigations Book 3, Chapter B7, U.S. Geological Survey, 1992.
- [100] M. F. Wheeler and C. N. Dawson. An operator splitting method for advection-diffusion-reaction problems. In *MAFELAP Proceedings 6*, pages 463–482. Academic Press, New York, 1988.
- [101] D. Yang. A characteristic-mixed method with dynamic finite element

space for convection-dominated diffusion problems. *J. Comput. Appl. Math.*, pages 343–353, 1992.

- [102] G. Zhou. *An adaptive streamline diffusion finite element method for hyperbolic systems in gas dynamics*. PhD thesis, University of Heidelberg, Heidelberg, Germany, 1992. Ph.D. thesis.
- [103] G. Zhou. A local L^2 error analysis of the streamline diffusion method for nonstationary convection-diffusion systems. In *Mathematical Modeling and Numerical Analysis*, pages 577–603. RAIRO, 1995.