

1999

A dynamic adaptive grid MPDATA scheme: application to the computational solution of atmospheric tracer transport problems

John Paul Iselin
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Atmospheric Sciences Commons](#), [Geophysics and Seismology Commons](#), and the [Mechanical Engineering Commons](#)

Recommended Citation

Iselin, John Paul, "A dynamic adaptive grid MPDATA scheme: application to the computational solution of atmospheric tracer transport problems " (1999). *Retrospective Theses and Dissertations*. 12137.
<https://lib.dr.iastate.edu/rtd/12137>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI[®]

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

**A dynamic adaptive grid MPDATA scheme:
Application to the computational solution of atmospheric tracer transport problems**

by

John Paul Iselin

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Mechanical Engineering

Major Professors: Joseph M. Prusa and William J. Gutowski Jr.

Iowa State University

Ames, Iowa

1999

Copyright © John Paul Iselin, 1999. All rights reserved.

UMI Number: 9940209

UMI Microform 9940209
Copyright 1999, by UMI Company. All rights reserved.

**This microform edition is protected against unauthorized
copying under Title 17, United States Code.**

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

Graduate College
Iowa State University

This is to certify that the Doctoral dissertation of
John Paul Iselin
has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

~~Co-major Professor~~

Signature was redacted for privacy.

~~Co-major Professor~~

Signature was redacted for privacy.

For the Major Program

Signature was redacted for privacy.

For the Graduate College

DEDICATION

To Ruth Ankony Iselin

TABLE OF CONTENTS

1	INTRODUCTION	1
1.1	Dynamic adaptive grid techniques defined	2
1.2	Motivation for dynamic adaptive grid techniques in atmospheric models	4
1.3	Questions posed during research	5
1.3.1	Overall motivational questions	5
1.3.2	Questions regarding MPDATA on a moving grid	6
1.4	Constraints and test cases	7
1.5	Thesis organization	9
2	A DYNAMIC ADAPTIVE GRID MPDATA SCHEME	10
2.1	A transformation from physical space into computational space	10
2.2	MPDATA	14
2.2.1	Basic MPDATA	15
2.2.2	MPDATA in multiple dimensions on a moving grid	18
2.2.3	Extensions to MPDATA	22
2.3	Dynamic grid adaptation	23
2.3.1	Literature review	24
2.3.2	Introduction to one- and two-dimensional grid generators	32
2.3.3	One-dimensional grid generation	33
2.3.4	Two-dimensional grid generation	38
2.4	Description of the one and two-dimensional models and their results	48
2.4.1	One-dimensional model	49
2.4.2	Two-dimensional model	61
2.4.3	Conclusions	73
3	ATMOSPHERIC TRANSPORT	75
3.1	MM5	76

3.1.1	MM5 coordinate system and governing equations	78
3.1.2	MM5 grid	79
3.1.3	Boundary conditions	81
3.2	Dynamic grid advection component	82
3.2.1	An MM5 compatible advection equation	83
3.2.2	An MM5 compatible MPDATA scheme	84
3.2.3	The dynamic grid advection component grid and domain correspondence with MM5	86
3.2.4	Code details	91
3.3	Interpolation	94
3.3.1	Mapping of grid points into integer space	95
3.3.2	Interpolator implementations	97
3.4	Numerical test cases and results	98
3.4.1	Zonal test case	99
3.4.2	Cylinder advection case	102
3.4.3	Water vapor passive advection case	114
4	CONCLUSIONS	122
4.1	Summary	122
4.2	Unique contributions	126
4.3	Future work	127
4.3.1	Short term extensions	127
4.3.2	Long term extensions	129
APPENDIX A "THIRD-ORDER" MPDATA AND THE RECURSIVE PSEUDO		
	VELOCITY OPTION	131
APPENDIX B DERIVATION OF MM5 GOVERNING EQUATIONS		140
APPENDIX C A HIGHER ORDER MONOTONE INTERPOLATION SCHEME		147
APPENDIX D MM5 PRECISION AND DGAC STRUCTURE		158
BIBLIOGRAPHY		158

LIST OF TABLES

Table 2.1	Errors and costs associated with the two-dimensional rotating cone problem. . .	68
Table 2.2	The effect of additional MPDATA iterations.	70
Table 2.3	The effect of the “third” order correction terms.	71
Table 2.4	The effect of the dynamic adaptive grid	72
Table 3.1	Effect of dynamic grid on number of grid points needed.	114
Table D.1	Logical break down of dynamic grid advection component.	158

LIST OF FIGURES

Figure 2.1	A sample grid in computational space	11
Figure 2.2	Property storage locations for the MPDATA scheme	21
Figure 2.3	Equidistribution illustration	27
Figure 2.4	Grid spacing produced by the one-dimensional grid generator	37
Figure 2.5	Characteristic of the one-dimensional grid	37
Figure 2.6	Poor quality two-dimensional grid	45
Figure 2.7	Improved quality two-dimensional grid	45
Figure 2.8	The effect of varying the stretching parameter λ_0 on a two-dimensional grid.	46
Figure 2.9	Effects of varying λ_0 on a two-dimensional grid.	46
Figure 2.10	Effect of varying the number of points on a two-dimensional grid	47
Figure 2.11	Effect of varying the over-relaxation parameter ω on the two-dimensional grid generation	48
Figure 2.12	Initial condition for the one-dimensional model.	50
Figure 2.13	An example one-dimensional grid.	50
Figure 2.14	Zone of influence for a static and a dynamic grid.	54
Figure 2.15	L_2 error versus safety factor for static uniform grid computations.	57
Figure 2.16	Order of accuracy of schemes without and with third-order terms	58
Figure 2.17	L_2 error versus safety factor s for dynamic grid computations.	59
Figure 2.18	Effects of limiting the grid point movement.	60
Figure 2.19	L_2 error versus computational cost for the one-dimensional model.	61
Figure 2.20	MPDATA compared to a leap frog scheme for the rotating cone problem.	64
Figure 2.21	Results of the rotating cone problem after 6 revolutions without “third” order correction terms.	65
Figure 2.22	Results of the rotating cone problem after 6 revolutions with “third” order correction terms.	66

Figure 3.1	Property storage locations for the MM5 modeling system	80
Figure 3.2	Property storage locations for the MPDATA scheme	87
Figure 3.3	Storage locations for the DGAC grid	88
Figure 3.4	V and W velocity stencils needed to calculate the $\hat{U}^{(k)}$ pseudo velocity at $i + 1/2$	89
Figure 3.5	Correspondence between the physical domains for the MM5 grid and the DGAC grid	90
Figure 3.6	Interpolator performance	95
Figure 3.7	Interpolation from seven points to 50 points	96
Figure 3.8	Initial tracer field for the zonal flow and cylinder advection test cases	100
Figure 3.9	Zonal flow test case results	103
Figure 3.10	MM5 fine grid and MM5 coarse grid correspondence	106
Figure 3.11	Results of the cylinder advection case at standard resolution after 24 hours of simulation	108
Figure 3.12	Results of the cylinder advection case at high resolution (15 km) after 24 hours of simulation	111
Figure 3.13	Differences between standard resolution results and fine resolution MPDATA results on a static grid.	113
Figure 3.14	RMS error as a function of grid points for static and dynamic DGAC	115
Figure 3.15	Average water vapor as a function of height	116
Figure 3.16	Initial condition for five day simulation	117
Figure 3.17	Tracer distribution on $\sigma = 0.825$ after 44 hours of simulation	119
Figure 3.18	Tracer distribution on $\sigma = 0.825$ after 120 hours of simulation	120
Figure 3.19	Horizontal velocity vector plots at $\sigma = 0.825$ after 120 hours of simulation	121
Figure C.1	Effect of flux corrected transport on the MPDATA scheme	155
Figure C.2	Interpolation stencil for the second order two-dimensional interpolator	157
Figure D.1	Comparison of single and double precision computations	159

ACKNOWLEDGMENTS

I would like to thank my wife for her unflagging faith, constant encouragement, and hard work through this doctoral program. My major professors Doctors Joseph M. Prusa and William J. Gutowski Jr. have created a wonderful learning environment through their insight, guidance, cooperation, and patience. It has been a pleasure working for and with each of you. I am appreciative of Doctors Zaito Pan and Zekai Otles for their assistance with the MM5 modeling system, and Doctor William Gallus for his assessment of model results. This work was supported through funding by NSF grant ATM-9619811 to Iowa State University and the Iowa State University Global Change Fellowship.

1 INTRODUCTION

Several springs ago, while working in the Atmospheric Science Department at Iowa State University a tornado was spotted. In contrast to the majority of the city's population, the department members headed to the roof of the building instead of the basement. Several bemoaned the weakness of the storm, commenting that if it had been stronger, they would have headed to their cars and given it a chase.

This experience points out not only why meteorologists are unique, but also the difference between a static observer and one that is willing to move. As we stood on that roof-top we observed the storm pass from behind buildings to our left for a few moments until it was obscured by buildings on the right. Had my colleagues given chase, they would have been able to observe the storm either intensify or decay at least until the storm out-ran them or took a path they could not follow. Their somewhat Lagrangian experience would have been more complete than our Eulerian view from the roof.

The purpose of this work is to demonstrate, in a physically realistic meteorological model, the effect of allowing grid points to move in response to changes within the developing flow field. Analogous to the fixed observers on the roof, the vast majority of the numerical models within the atmospheric science community use evenly spaced static grids. The dynamic adaptive grid (DAG) approach allows the grid points to move and cluster in response to the flow field development, just as those who would have given chase would be responding to the interesting features of the storm.

Passive atmospheric transport, such as water vapor advection, was chosen as the process to model. Atmospheric tracers are typically drawn into long filamentary structures by non-linear atmospheric dynamics. These filaments are difficult to model because of their sharp gradients and the high aspect ratios.

The Multidimensional Positive Definite Advection Transport Algorithm (MPDATA) of Smolarkiewicz (1983) and Smolarkiewicz and Clark (1986) was chosen as the numerical scheme. It is well suited to atmospheric transport problems because it is second-order accurate, has small internal diffusion, and is positive definite. One of the novel aspects of this work is the implementation of MPDATA in a DAG

environment. Chapter 2 is devoted to developing and documenting the characteristic of MPDATA and its various options on a moving grid. This scheme will be referred to as the dynamic MPDATA scheme. Not only does the positive definite aspect of MPDATA make it well suited to model atmospheric advection, but it makes it well suited to the DAG technique because when the background tracer value is zero the scheme is inherently monotonic.

To simulate atmospheric tracer transport, a set of Fortran 90 subroutines called the Dynamic Grid Advection Component (DGAC) was developed and validated. This model, which uses the dynamic MPDATA scheme, was developed to run in conjunction with the Fifth-Generation Penn State/NCAR Mesoscale Model (MM5)(Grell et al.,1995). MM5 functions as the driver providing the velocity fields to the DGAC.

Three progressively more challenging test cases demonstrate that the dynamic MPDATA scheme has less overall diffusion than MPDATA without a dynamic grid (which will be called the static MPDATA scheme) and the centrally differenced leap frog scheme that is the standard MM5 advection scheme. The first of these test cases uses both an idealized analytical tracer field and wind field. The second test case still uses an idealized initial tracer field, but a physically realistic wind field. The third test case uses both physically realistic tracer and wind fields. It is shown that virtually identical results can be achieved with the dynamic MPDATA scheme using a quarter of the grid points of the static MPDATA scheme.

The next section precisely defines what is meant by the dynamic adaptive grid technique. Section 1.2 discusses the motivation for using the DAG technique in atmospheric models. Section 1.3 poses questions that motivated this work and arose during the study. Section 1.4 discusses the constraints that were imposed upon the study and the test cases that were performed. The last section, 1.5, gives more detail about the organization of the thesis.

1.1 Dynamic adaptive grid techniques defined

There are multiple ways to approach or define DAG techniques. In general they involve coupling the grid and the numerical scheme so that the grid changes during the course of a numerical integration to reduce the error involved in the approximation of the solution of an ordinary or partial differential equation. An alternative to DAG is to increase globally the number of grid points or elements within a model and keep their number constant and position static for the duration of the integration using a scheme with a constant order of accuracy. This alternative is more straight forward than DAG because a grid is defined a priori and remains stationary throughout the integration. The disadvantage to globally

increasing the resolution is that the computational cost increases at least to the power of the physical dimension. For example, to double the resolution in each direction on a three-dimensional problem the computational cost would increase at least by a factor of eight. In the case of an explicit finite difference scheme the computational cost would increase by a factor of sixteen, since the time step would need to be halved as well in order to maintain stability. Since only hyperbolic equations are being considered in this study, this estimate does not include problems when elliptic equations (ie pressure fields) need to be calculated. In many problems, globally increasing the grid resolution is excessive since the grid is refined in areas where a coarser grid would be adequate. The DAG technique refines the grid only in critical areas.

According to Kim and Thompson (1990) there are three basic adaptive grid strategies available. These include redistribution of grid points, local refinement, and local increase of algorithm order. Grid point redistribution involves using a fixed number of grid points and allowing them to move so they cluster in certain areas at the expense of having lower resolution in other areas. There are several different approaches to this method. Since this is the method that was used in this work, the different approaches are summarized in greater detail in section 2.2. The major advantages to this method are that the numerical bookkeeping is relatively easy compared to the other two methods, the computational cost of the numerical integration is approximately constant although there is a cost to calculating the grid point movement, the amount of memory required for data storage stays constant, and the method can be applied to multidimensional problems. It was also found in this study that the movement of the grid points can add Lagrangian properties to the computation, which can be advantageous from a physical and numerical viewpoint. This aspect of grid point redistribution is developed in detail in section 2.4.1.2. The disadvantages are that the resolution is decreased in certain areas and the grid cells can become skewed.

The second type of adaptive grid strategies is local refinement. This method starts with a topologically Cartesian grid or an unstructured grid. During the course of the simulation, additional points are inserted in areas that require increased resolution and removed from areas no longer requiring fine resolution. The advantages of this method are that no area has decreased resolution in order to increase the resolution of other areas, cell shapes can be kept from becoming skewed by the method that inserts the points, and the method is applicable to multiple dimensions. The disadvantages are that the data structures involved in the inter-point communications can become complicated and the array sizes are not constant. Dannenhoffer and Baron (1985) and Oden et al. (1987) are examples of this approach.

An approach that may be viewed as a combination of these first two methods was developed by Berger

and Olinger (1984) and applied to atmospheric flows by Skamarock and Klemp (1993) and Skamarock et al. (1994). In this method, finer localized grids are laid over a coarser global grid or other coarser localized grids. These localized grids are allowed to move during the integration period. This method is discussed in greater detail in section 2.3.1.2.

The final type of adaptive technique deals with locally increasing the order of accuracy of the numerical scheme and not adjusting the grid. The advantage to this technique is the grid remains static and of constant size. The major disadvantages are that it is not easily implemented in multiple dimensions, and for unsmooth fields aliasing errors could occur. Kim and Thompson (1990) indicate that this method has had no significant application in multidimensional computational fluid dynamics.

1.2 Motivation for dynamic adaptive grid techniques in atmospheric models

In horizontal directions, atmospheric interactions have a tendency to produce high gradient tracer fields. The nonlinear nature of fluid flow has a tendency to draw tracer fields into long narrow filaments. These structures are difficult to model due to their high spatial aspect ratios. Fine grid-spacing is required to capture the steep gradients across the filaments while large spatial extents are needed to capture the length of the structures. Newell et al. (1992) found evidence of thin horizontal structures of water vapor in the troposphere that they called “tropospheric rivers”.

In the vertical direction, evidence has been found that water vapor appears in thin layers. Newell and Coauthors (1996) found over 500 layers of water vapor and other trace atmospheric components from 105 vertical profiles collected during aircraft flights for the Pacific Exploratory Mission A in the fall of 1991. Over 440 vertical water vapor layers were found by Iselin and Gutowski (1997) in rawinsonde observations during the 45 days of the STORM-FEST project. Although the criteria to become a layer in this study was quite stringent, at least one layer was found in more than half of the soundings. It appeared that the moist layers were formed by rising water vapor that became trapped between air with relatively high static stability at the upper edge of the layer and relatively low static stability at the lower edge of the layer.

Atmospheric flows exhibit important characteristics that have times scales ranging from a few seconds to many days and spatial scales ranging from a few meters to hundreds of kilometers (Skamarock and Klemp, 1993). Localized atmospheric phenomena, like fronts and convective storms, are forced by nonlinear interactions on scales larger than themselves (Dietachmayer and Droegemeier, 1992). The problem of “down scaling” within the atmospheric science community involves methods of imposing larger scale forcing on smaller scale dynamic motion. Numerous methods are being investigated to

solve the down scaling problem. Using dynamic adaptive grids is one method of approaching the problem.

DAG techniques have been successfully used in engineering fluid dynamics problems where sharp gradients form. Examples are in supersonic flows where shock waves and contact discontinuities form and combustion problems where flame propagation fronts appear (Anderson and Rai, 1982 and Dwyer et al., 1982). Although the gradients in atmospheric tracer fields are not as strong as the gradients in these problems, it is reasonable to believe that DAG techniques may be beneficial to the solution of atmospheric tracer transport problems.

1.3 Questions posed during research

Any good research is motivated by questions that are not trivially answered and require some investigation. Additionally, new questions should naturally arise. This research has probably spawned at least as many questions as it has answered. This section explains some of the questions that were originally asked and others that arose through the course of the study. Many of these questions were answered, at least in part, while other still remain to be answered through further study. This section is divided into two parts. Section 1.3.1 introduces overall motivational questions, while section 1.3.2 presents questions dealing with the MPDATA numerical scheme on a moving grid.

1.3.1 Overall motivational questions

The primary questions were:

1. Can DAG techniques yield numerical results superior to a static grid model while using fewer grid points?
2. If so, can the technique be made more computationally efficient, than using more grid points on a static grid?

At least partial or preliminary answers to these questions have been found. During validation tests, approximately equivalent results were obtained in an adaptive grid model using a quarter the number of grid points of a corresponding static grid model. This validation test used an idealized initial condition but realistic velocity profiles. Initial tests indicate the when a realistic initial tracer field is used that improved transport is realized.

The answer to the second question is not completely clear. There is added computational overhead with the DAG method, as additional equations need to be solved in order to determine the movement

of the grid points. Since the grid is forced to conform to the domain boundaries, the problem is inherently elliptic. Elliptic equations are expensive to solve in comparison to the hyperbolic passive tracer advection equation that is ultimately being computed. More computationally efficient numerical methods for moving the grid points in two-dimensions exist than were used in this work. These methods are reviewed in section 2.3.1.1. Because the primary goal of the research was to show that the DAG technique could improve atmospheric tracer transport with fewer grid points, the implementation and evaluation of these techniques were left for future work.

The significance of the computational penalty of solving an additional set of elliptic equations is diminished if the full set of prognostic equations is solved on the dynamic grid. Since the full set of prognostic equations includes mass continuity, three momentum equations, the energy equation, and the tracer advection equation, the computational effort of solving these equations would increase by at a factor of at least six over solving just the tracer advection equation as was done in this study. However, the cost of moving the grid would remain constant. Therefore, the advantages of having fewer grid points would be increased while the penalty of moving the grid would remain constant.

In this work, only the tracer advection equation was solved on a dynamic grid, while the wind fields were solved for on a static grid, so interpolation was used to transfer the wind field from the static grid to the dynamic grid. The second-order, monotone interpolation scheme of Smolarkiewicz and Grell (1992) was used. This interpolator faithfully reproduced the wind field on the dynamic grid, but it was the most expensive part of the DGAC. Solving the full set of prognostic equations on the dynamic grid would eliminate the need for this interpolation.

1.3.2 Questions regarding MPDATA on a moving grid

The MPDATA scheme was chosen as the scheme to compute the tracer advection. Questions concerning MPDATA were:

1. Could MPDATA be made to work with a DAG?
2. How would the explicit nature of the MPDATA scheme and corresponding CFL condition be affected by grid point movement?

The first of these questions was answered. One-, two-, and three-dimensional versions of the MPDATA were developed that worked with moving grids. More advanced options of the MPDATA scheme that are discussed in section 2.3 were also made to work with DAG. To the authors knowledge, there is no published work that uses MPDATA on a moving grid. New insight into when and why the DAG

method fails was found while trying to answer the second of the above questions. For explicit schemes there are spatial limits at each time step that bound the grid point movement. If the grid point moves outside these limits the CFL condition will be violated. It was originally thought that these limits were centered around the grid point location at the beginning of the time step. However, it was found that these limits were centered at the end point of the first order Lagrangian transport of the grid point by the fluid velocity during the time step. This gave the insight that the ideal grid point movement would be along Lagrangian trajectories. If the grid points were not constrained to conform to the boundaries of the domain, and, therefore, were free to follow the flow, Courant numbers significantly larger than one would be permissible. Thus the use of adaptive grid techniques brings a Lagrangian property to a problem formulated in an Eulerian context. Further discussion appears in section 2.4.1.2.

1.4 Constraints and test cases

Initially, a simple one-dimensional advection problem of an ideal profile was computed. From this starting point, two- and three-dimensional problems of increasing complexity were performed, each based on knowledge learned from the preceding simpler problems. The final DGAC model retained certain simplifying assumptions. Some courses of action to relax these constraints are suggested in chapter 4.

This work was restricted to passive scalar advection. Since a scalar equation was solved, the complicating factor of dealing with a stiff system of equations was avoided. In addition, since passive advection was performed, there was no feedback to the driving model that could cause it to become unstable. These simplifications allowed the investigators to concentrate on applying DAG techniques to the atmospheric advection problem.

For the purposes of stretching the grid, the DGAC was designed so the horizontal grid stretching was not a function of height. Not only did this decouple the horizontal and vertical grid stretching, but kept initially vertical columns vertical in anticipation of using parameterization schemes for radiation and convection that assume vertical columns of grid points. The existing three-dimensional atmospheric model, MM5(Grell et al., 1995), that was available and was known to run on the available computer platforms was used to compute the wind fields that drove the DGAC model.

The one-dimensional model was used to determine how to implement the MPDATA scheme with the DAG technique. It was then used to compare the apparent order of accuracy of the dynamic MPDATA scheme to the static MPDATA scheme. Finally, it was used to develop a method of limiting the grid point movement so that the CFL condition would not be violated. The results show that the

the MPDATA scheme is very amenable to the DAG technique and can dramatically improve results using fewer grid points compared to the static MPDATA scheme.

The two-dimensional model solved the rotating cone problem. It was used to learn how to implement the dynamic MPDATA scheme in a multidimensional case. Since, in the DGAC, the horizontal grid stretching was not a function of height, a two-dimensional grid generator could be used to determine the horizontal grid point distribution in this three-dimensional model. This two-dimensional grid generator was initially developed for use with the two-dimensional model. Numerous computations were performed to compare the numerical errors obtained from permutations of different MPDATA options and the dynamic grid option. Several different error measures were considered when evaluating the performance of models. The diffusion error was taken as the difference between the maximum value of the initial condition and the maximum of the computed solution. The phase error was calculated by determining the distance between the maximums in the computed and analytical solutions. The L_2 error norm was also computed. Since the point of the DAG technique is to resolve local features of the prognostic field, the diffusion error was considered a better measure of performance than the L_2 error. Using this measure the DAG technique has less diffusion and uses less CPU cycles to perform a simulation with the same diffusion error than does a model with a static, uniform grid.

The three-dimensional model was initially tested using an idealized zonal flow case and an idealized initial cylindrical tracer field with no topography. The results were compared to results computed with the standard MM5 leap frog scheme. The results indicated the MPDATA scheme is superior to the leap frog scheme and that the dynamic grid computations were an improvement over the static uniform grid computations.

In the second test case, an initial tracer field was advected during a 24 hour simulation using the wind fields from March 6, 1992. This test case was used in a series of computations to compare the performance of the dynamic MPDATA scheme, the static MPDATA scheme, and the standard MM5 advection scheme. Computations were performed at 30 and 15 km resolutions with the static MPDATA scheme and the standard MM5 advection routine to demonstrate that they converged to the same solution as the grid was refined. Comparison of both 30 km results with the 15 km results showed that the 30 km static MPDATA result was closer to the 15 km results than was the 30 km MM5 result. Therefore, the 15 km static MPDATA result was taken as the "exact" solution with which to compare the other results. A comparison of a dynamic MPDATA result using the same number of grid points as the 30 km computations, showed that the same solutions were achieved with the dynamic MPDATA scheme with a quarter the points of the static MPDATA scheme. Additional computations

using progressively fewer grid points showed that the dynamic MPDATA scheme with fewer grid points could consistently produce similar results as the static MPDATA scheme.

The third test case used the initial water vapor field for March 6, 1992, in conjunction with the wind fields from March 6 through March 10, 1992. The domain included the entire continental United States and the simulation ran for 5 days. Although water vapor was advected, there was no mechanism to account for evaporation or precipitation. The DAG results were compared to the static MPDATA calculations and the MM5 leap frog computations for the same time period under the same restrictions. The results indicate that the MPDATA scheme is far superior to the leap frog scheme and that there was a lesser but significant impact from using the dynamic MPDATA scheme over the static MPDATA scheme. Suggestions are given in section 4.3.1 of further possible work using this test case.

1.5 Thesis organization

The next chapter deals with the development of MPDATA with a moving grid, grid generation in one- and two-dimensions, and the development and validation of one- and two-dimensional codes. This second chapter explains the building blocks that were needed for the three-dimensional model. Quantitative results compare the one-dimensional models with static and dynamic grids. Both quantitative and qualitative results are discussed for the two-dimensional model.

Chapter 3 discusses the development and results of the three-dimensional model. MM5 is reviewed briefly to give the reader its salient points. Both the DGAC (the three-dimensional MPDATA code) and MM5 use staggered grids, but they are staggered differently. Additionally, the dynamic grid can obviously move. Therefore, a section on the coupling between these two grids was included. Since the wind fields were computed on the static MM5 grid and used on the DGAC grid, interpolation was used to transfer the wind fields from the static to the dynamic grid. Additionally, the initial condition was interpolated from the MM5 grid to the DGAC grid. In order to minimize the effect of this interpolation a second-order monotone interpolation scheme that was developed by Smolarkiewicz and Grell (1992) was used. Therefore a section is devoted to describing this interpolation scheme. Chapter 3 concludes with a section describing the results of the three-dimensional DGAC test cases.

In chapter 4, the work is summarized, conclusions are drawn, unique contributions are highlighted, and suggestions for future work are given.

2 A DYNAMIC ADAPTIVE GRID MPDATA SCHEME

This chapter describes the MPDATA finite difference scheme in a moving reference frame, the methods used for dynamically adjusting the grid points, and implementation of the scheme to solve one- and two-dimensional problems. These problems provided an environment to study the numerical properties of the dynamic MPDATA scheme, and gain the necessary experience base before the more complicated model was developed.

With this method the prognostic equation is solved in a computational space that has a static, orthogonal, evenly spaced grid. A one- and a two-dimensional time independent version of this space is used in the one- and two-dimension grid generators. Section 2.1 describes the metric terms involved in the transformation. Section 2.2 describes the development of the MPDATA scheme in its most basic form, in a multidimensional form that is applicable to a moving grid, and introduces two options that decrease the diffusion and phase error associated with the scheme. Appendix A contains the full development of these two options. Section 2.3 describes the development of the one- and two-dimensional grid generators that were used to determine the grid point distribution at each time step. Section 2.4 describes the one- and two-dimensional models and presents the effects of varying the number of grid points, the Courant numbers, and the MPDATA options with and without the use of a dynamic grid.

2.1 A transformation from physical space into computational space

Since, in the DAG method, the grid is unevenly spaced and able to move, it is advantageous to transform the equations of motion into a computational domain where the grid is stationary and orthogonal. It is additionally beneficial if the computational grid is in integer space.

The computational grid in integer space has grid spacing of one and the grid coordinates are integer values starting with one and progressing to the maximum number of grid points in that coordinate direction. Figure 2.1 shows a small 6×5 grid that uses integer space. The advantage of integer space is that since the grid spacing is even and of size one, finite difference approximations of derivatives are

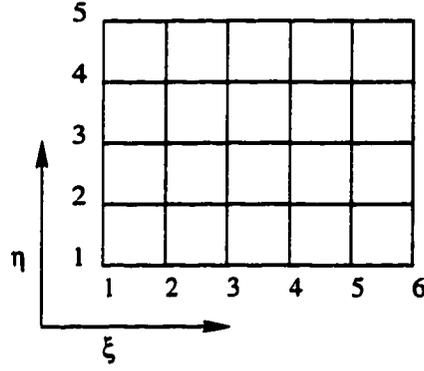


Figure 2.1 A sample grid in computational space.

simple and efficient. All of the denominators are removed from the finite difference computations.

Consider a non-orthogonal grid that is a function of time where the physical coordinates are (t, x, y, z) for time and the three spatial coordinates respectively. The corresponding computational coordinates are designated as (τ, ξ, η, ζ) . The functional relationship between the physical and computation domains is defined as:

$$\begin{bmatrix} t \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \tau \\ x(\tau, \xi, \eta, \zeta) \\ y(\tau, \xi, \eta, \zeta) \\ z(\tau, \xi, \eta, \zeta) \end{bmatrix} \quad (2.1)$$

Derivatives in the physical domain are transformed as

$$\begin{bmatrix} \frac{\partial}{\partial t} \\ \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} = \begin{bmatrix} 1 & \frac{\partial \xi}{\partial t} & \frac{\partial \eta}{\partial t} & \frac{\partial \zeta}{\partial t} \\ 0 & \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} & \frac{\partial \zeta}{\partial x} \\ 0 & \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} & \frac{\partial \zeta}{\partial y} \\ 0 & \frac{\partial \xi}{\partial z} & \frac{\partial \eta}{\partial z} & \frac{\partial \zeta}{\partial z} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial \tau} \\ \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial \zeta} \end{bmatrix} \quad (2.2)$$

The metric terms in the square matrix of equation 2.2 are partial derivatives with respect to the physical domain. These derivatives are evaluated in terms of the transformed coordinates. This was done by inverting the inverse transformation specified in equation 2.2. Writing the inverse transformation as:

$$\begin{bmatrix} \frac{\partial}{\partial \tau} \\ \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} 1 & \frac{\partial x}{\partial \tau} & \frac{\partial y}{\partial \tau} & \frac{\partial z}{\partial \tau} \\ 0 & \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ 0 & \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ 0 & \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial t} \\ \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \quad (2.3)$$

Computing the inverse of the linear system in equation 2.3 yields:

$$\begin{bmatrix} \frac{\partial}{\partial t} \\ \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} = \frac{1}{J_3} \begin{bmatrix} 1 & M_{12} & M_{13} & M_{14} \\ 0 & \frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta} \frac{\partial z}{\partial \eta} & \frac{\partial y}{\partial \zeta} \frac{\partial z}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \zeta} & \frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \xi} \\ 0 & \frac{\partial x}{\partial \zeta} \frac{\partial z}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial z}{\partial \zeta} & \frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \zeta} \frac{\partial z}{\partial \xi} & \frac{\partial x}{\partial \eta} \frac{\partial z}{\partial \xi} - \frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \eta} \\ 0 & \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \zeta} - \frac{\partial x}{\partial \zeta} \frac{\partial y}{\partial \eta} & \frac{\partial x}{\partial \zeta} \frac{\partial y}{\partial \xi} - \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \zeta} & \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial \tau} \\ \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial \zeta} \end{bmatrix} \quad (2.4)$$

where the Jacobian of the transformation is:

$$J_3 = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \zeta} \frac{\partial z}{\partial \eta} + \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \zeta} \frac{\partial z}{\partial \xi} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \zeta} + \frac{\partial x}{\partial \zeta} \frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \eta} - \frac{\partial x}{\partial \zeta} \frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \xi} \quad (2.5)$$

and

$$\begin{aligned} M_{12} &= \frac{\partial x}{\partial \tau} \left(\frac{\partial y}{\partial \zeta} \frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \zeta} \right) + \frac{\partial y}{\partial \tau} \left(\frac{\partial x}{\partial \eta} \frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \zeta} \frac{\partial z}{\partial \eta} \right) + \frac{\partial z}{\partial \tau} \left(\frac{\partial x}{\partial \zeta} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \zeta} \right) \\ M_{13} &= \frac{\partial x}{\partial \tau} \left(\frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta} \frac{\partial z}{\partial \xi} \right) + \frac{\partial y}{\partial \tau} \left(\frac{\partial x}{\partial \zeta} \frac{\partial z}{\partial \xi} - \frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \zeta} \right) + \frac{\partial z}{\partial \tau} \left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \zeta} - \frac{\partial x}{\partial \zeta} \frac{\partial y}{\partial \xi} \right) \\ M_{14} &= \frac{\partial x}{\partial \tau} \left(\frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \eta} \right) + \frac{\partial y}{\partial \tau} \left(\frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial z}{\partial \xi} \right) + \frac{\partial z}{\partial \tau} \left(\frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} - \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} \right) \end{aligned} \quad (2.6)$$

By comparing the elements of the square matrix of equation 2.2 with those of equation 2.4, it is apparent that:

$$\begin{aligned} \frac{\partial \xi}{\partial t} &= \frac{M_{12}}{J_3} \\ \frac{\partial \eta}{\partial t} &= \frac{M_{13}}{J_3} \\ \frac{\partial \zeta}{\partial t} &= \frac{M_{14}}{J_3} \\ \frac{\partial \xi}{\partial x} &= \frac{1}{J_3} \left(\frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta} \frac{\partial z}{\partial \eta} \right) \\ \frac{\partial \eta}{\partial x} &= \frac{1}{J_3} \left(\frac{\partial y}{\partial \zeta} \frac{\partial z}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \zeta} \right) \\ \frac{\partial \zeta}{\partial x} &= \frac{1}{J_3} \left(\frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \xi} \right) \\ \frac{\partial \xi}{\partial y} &= \frac{1}{J_3} \left(\frac{\partial x}{\partial \zeta} \frac{\partial z}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial z}{\partial \zeta} \right) \\ \frac{\partial \eta}{\partial y} &= \frac{1}{J_3} \left(\frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \zeta} \frac{\partial z}{\partial \xi} \right) \\ \frac{\partial \zeta}{\partial y} &= \frac{1}{J_3} \left(\frac{\partial x}{\partial \eta} \frac{\partial z}{\partial \xi} - \frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \eta} \right) \\ \frac{\partial \xi}{\partial z} &= \frac{1}{J_3} \left(\frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \zeta} - \frac{\partial x}{\partial \zeta} \frac{\partial y}{\partial \eta} \right) \\ \frac{\partial \eta}{\partial z} &= \frac{1}{J_3} \left(\frac{\partial x}{\partial \zeta} \frac{\partial y}{\partial \xi} - \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \zeta} \right) \\ \frac{\partial \zeta}{\partial z} &= \frac{1}{J_3} \left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \right) \end{aligned} \quad (2.7)$$

Since all of the derivatives on the right-hand sides of equations 2.5 and 2.7 are with respect to the computational domain, the derivatives are simple and efficient to calculate.

These metric terms are used to develop MPDATA on a moving grid in section 2.2.2, a one-dimensional grid generator in section 2.3.3, and a two-dimensional grid generator in section 2.3.4. However in the case of the grid generators the relations simplify because there is no temporal component to the transformations and the dimensions are one and two respectively. Therefore, for the one-dimensional grid generator, equation 2.7 reduces to:

$$\frac{\partial \xi}{\partial x} = \frac{1}{\frac{\partial x}{\partial \xi}} \quad (2.8)$$

A derivative with respect to the physical coordinate is then:

$$\frac{\partial}{\partial x} = \frac{1}{\frac{\partial x}{\partial \xi}} \frac{\partial}{\partial \xi} \quad (2.9)$$

In the development of the one-dimensional grid generator not only is the first derivative needed but also the second derivative. By differentiating equation 2.9 with respect to the physical coordinate x , the second derivative can be determined as:

$$\frac{\partial^2}{\partial x^2} = \frac{1}{\left(\frac{\partial x}{\partial \xi}\right)^2} \frac{\partial^2}{\partial \xi^2} - \frac{1}{\left(\frac{\partial x}{\partial \xi}\right)^3} \frac{\partial^2 x}{\partial \xi^2} \frac{\partial}{\partial \xi} \quad (2.10)$$

In the two-dimensional grid generation case, since $\frac{\partial z}{\partial \xi} = 1$, $\frac{\partial z}{\partial \eta} = \frac{\partial z}{\partial \xi} = 0$, equation 2.7 reduces to

$$\frac{\partial \xi}{\partial x} = +\frac{1}{J_2} \frac{\partial y}{\partial \eta} \quad (2.11)$$

$$\frac{\partial \eta}{\partial x} = -\frac{1}{J_2} \frac{\partial y}{\partial \xi}$$

$$\frac{\partial \xi}{\partial y} = -\frac{1}{J_2} \frac{\partial x}{\partial \eta}$$

$$\frac{\partial \eta}{\partial y} = +\frac{1}{J_2} \frac{\partial x}{\partial \xi} \quad (2.12)$$

where the two-dimensional Jacobian is:

$$J_2 = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \quad (2.13)$$

Therefore, in the two-dimensional case the derivatives with respect to the physical domain are transformed as:

$$\begin{aligned} \frac{\partial}{\partial x} &= \frac{1}{J_2} \left(\frac{\partial y}{\partial \eta} \frac{\partial}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial}{\partial \eta} \right) \\ \frac{\partial}{\partial y} &= \frac{1}{J_2} \left(\frac{\partial x}{\partial \xi} \frac{\partial}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial}{\partial \xi} \right) \end{aligned} \quad (2.14)$$

Just as in the case of the one-dimensional weight function the second derivatives are needed for the two-dimensional weight function. Taking the derivative of equation 2.14 yields:

$$\begin{aligned}
 \frac{\partial^2}{\partial x^2} &= \frac{1}{J_2^2} \left[\left(\frac{\partial y}{\partial \eta} \right)^2 \frac{\partial^2}{\partial \xi^2} - 2 \left(\frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} \right) \frac{\partial^2}{\partial \xi \partial \eta} + \left(\frac{\partial y}{\partial \xi} \right)^2 \frac{\partial^2}{\partial \eta^2} \right] \\
 \frac{\partial^2}{\partial x \partial y} &= -\frac{1}{J_2} \left[\left(\frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \eta} \right) \frac{\partial^2}{\partial \xi^2} - \left(\frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} \right) \frac{\partial^2}{\partial \xi \partial \eta} + \left(\frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \eta} \right) \frac{\partial^2}{\partial \eta^2} \right] \\
 \frac{\partial^2}{\partial y^2} &= \frac{1}{J_2^2} \left[\left(\frac{\partial x}{\partial \eta} \right)^2 \frac{\partial^2}{\partial \xi^2} - 2 \left(\frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} \right) \frac{\partial^2}{\partial \xi \partial \eta} + \left(\frac{\partial x}{\partial \xi} \right)^2 \frac{\partial^2}{\partial \eta^2} \right]
 \end{aligned} \tag{2.15}$$

2.2 MPDATA

This section describes several variations of the Multidimensional Positive Definite Advection Algorithm (MPDATA) that was developed by Smolarkiewicz (1983). MPDATA is a second-order in time and space finite difference scheme that was originally developed exclusively for tracer transport problems but has since been generalized to deal with all of the advection terms in the equations for fluid motion.

The basic idea of the MPDATA scheme is to use a donor-cell scheme to obtain a first-order accurate solution and then define pseudo velocities that when used as the velocity in conjunction with the donor-cell scheme approximates the error associated with the first-order scheme. This error estimate is then subtracted from the first-order approximation. This creates a scheme that is second-order accurate, sign-preserving, easily implemented, and is computationally efficient.

MPDATA is in the same class of schemes as Lax-Wendroff. Lax-Wendroff, like MPDATA, is developed by subtracting spatial approximations of the higher order error terms from the a first-order solution, thus achieving a higher order scheme. The major difference between the MPDATA scheme and the Lax-Wendroff scheme is that MPDATA uses up-wind approximations for the correction terms, while Lax-Wendroff uses central differences to approximate the correction terms.

MPDATA has a sign-preserving feature that makes it particularly attractive for this work which deals exclusively with tracer transport. Given a tracer quantity like water vapor mixing ratio, that physically can only have positive values, the MPDATA scheme will not produce negative values as long as the Courant-Fredricks-Lewy(CFL) condition is satisfied. When background values of the tracer are zero, it is therefore non-oscillatory which makes the scheme very amenable to dynamic grid adaptation. In section 2.3, it will be seen that the dynamic adaptive grid (DAG) technique clusters grid points around areas of high slope and curvature. If oscillations are permitted by the numerical scheme, the grid points have a tendency to cluster around these ripples due to their high curvature. The resolution

of the grid in the area of the numerical ripples is increased instead of the area of physically relevant features. Since MPDATA does not create any of these background ripples, this difficulty is avoided.

MPDATA is an explicit scheme and therefore subject to the CFL condition which limits the size of the time step in order to maintain stability. When the DAG technique is used, the physical velocity is replaced by the difference between the physical velocity and the velocity of the grid point. If care is not taken to maintain a smoothly varying grid with respect to time, excessive grid point velocity alone can cause CFL violations.

The most basic form of MPDATA uses a first-order predictor step, and then adds second-order anti-diffusion correction terms to it. A quantity called the pseudo velocity is used in the approximation of the correction terms. The pseudo velocity is defined such that when used in place of the physical velocity in a donor-cell step the second-order correction are approximated. This pseudo velocity is a function of the physical velocity and the transport quantity, has units of velocity, but has no physical significance. Since it has the units of velocity there is no reason why a further iteration can not be made to the basic MPDATA scheme using a new pseudo velocity of the same form as the first. This new pseudo velocity is a function of the preceding pseudo velocity and the transport quantity. Therefore, MPDATA can be run with as many iterations as desired. Although the numerical error is reduced with additional iterations, MPDATA remains second order accurate in time and space. Numerical experiments by Margolin and Smolarkiewicz (1989) show that additional iterations beyond four yield insignificant improvement in the solution.

In order to illustrate the development of the scheme, the MPDATA scheme is reviewed in section 2.2.1 for a one-dimension, constant velocity case, for an unspecified number of iterations. Section 2.2.2 reviews the development of MPDATA in multiple dimensions on a moving grid. As indicated by its name, MPDATA is applicable to multidimensions and therefore this aspect is not new. However, the application of MPDATA to a DAG environment is new. Section 2.2.3 introduces two extensions to MPDATA that significantly enhance the schemes accuracy and computational efficiency. The details of these extensions is included in appendix A.

2.2.1 Basic MPDATA

In this section, for illustration purposes only, MPDATA is developed for a one-dimensional flow with constant velocity. These are not necessary constraints for MPDATA, but are assumed in this section only to simplify the initial explanation. The first-order donor-cell scheme is stated; the pseudo velocity for the first MPDATA iteration will be developed; and this expression will then be generalized

for subsequent iterations.

Consider the transport of some non-negative property q in a one-dimensional Eulerian context by a constant velocity flow:

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial x} (uq) = 0 \quad (2.16)$$

where u represents the fluid velocity, t time, and x the spatial location. q can either be an extensive fluid property such that

$$q = \rho\phi \quad (2.17)$$

where ρ is the fluid density and ϕ the intensive property, or since the flow field is constant, it can be an intensive fluid property as long as the the fluid density ρ is a constant.

MPDATA uses the donor-cell scheme to compute a low order solution for the first iteration, calculates a pseudo velocity, and then calculates a second-order correction term using the donor cell scheme with the pseudo velocity. The donor-cell scheme for the first-order predictor step is:

$$q_i^{(1)} = q_i^n - \{F(q_i^n, q_{i+1}^n, U_{i+1/2}) - F(q_{i-1}^n, q_i^n, U_{i-1/2})\} \quad (2.18)$$

where the superscript n is the time level, the subscript i is the spatial index, and $q_i^{(1)}$ is the first order approximation to q_i^{n+1} . Integer values of i represent cell centers and half integer values represent cell faces. The flux function F for the donor-cell scheme is

$$F(q_L, q_R, U) = \max(U, 0) q_L - \min(U, 0) q_R \quad (2.19)$$

where $U = \frac{\Delta t}{\Delta x} u$ is the Courant number. For this section only u is constant; therefore, $U_{i+1/2}^n = U_{i-1/2}^n = U$ in equation 2.18. The following analysis examines this scheme to determine the form of the pseudo velocities $U_{i\pm 1/2}^{(1)}$ so that:

$$q_i^{(2)} = q_i^{(1)} - \left\{ F\left(q_i^{(1)}, q_{i+1}^{(1)}, U_{i+1/2}^{(1)}\right) - F\left(q_{i-1}^{(1)}, q_i^{(1)}, U_{i-1/2}^{(1)}\right) \right\} \quad (2.20)$$

where $q_i^{(2)}$ is a second-order approximation of q_i^{n+1} .

Approximating the temporal derivative using a forward difference, equation 2.16 is:

$$\frac{q^{n+1} - q^n}{\Delta t} + \frac{\partial}{\partial x} (uq^n) = 0 \quad (2.21)$$

Expanding q^{n+1} about time level n , using a Taylor series, yields:

$$\frac{\partial q}{\partial t} + \frac{1}{2}\Delta t \frac{\partial^2 q}{\partial t^2} + \frac{\partial}{\partial x} (uq) = \mathcal{O}(\Delta t^2) \quad (2.22)$$

The partial derivative with respect to time of only the lowest order terms of equation 2.22 yields:

$$\frac{\partial^2 q}{\partial t^2} = -\frac{\partial}{\partial x} \left(u \frac{\partial q}{\partial t} \right) + \mathcal{O}(\Delta t) \quad (2.23)$$

Using the lowest order terms of equation 2.22 and substituting for $\frac{\partial q}{\partial t}$ in equation 2.23:

$$\frac{\partial^2 q}{\partial t^2} = \frac{\partial}{\partial x} \left(u^2 \frac{\partial q}{\partial x} \right) + \mathcal{O}(\Delta t) \quad (2.24)$$

Substitution of equation 2.24 for the second term of equation 2.22 yields the original differential equation on the left hand side and the first-order error term, expressed as a spatial derivative on the right hand side:

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial x} (uq) = -\frac{\partial}{\partial x} \left(\frac{\Delta t}{2} u^2 \frac{\partial q}{\partial x} \right) + \mathcal{O}(\Delta t^2) \quad (2.25)$$

In order to determine the spatial error terms associated with the donor-cell scheme, the spatial derivatives in equation 2.25 are represented as:

$$\begin{aligned} \frac{\partial q}{\partial t} + \frac{u(q_i - q_{i-1})}{\Delta x} &= -\frac{\partial}{\partial x} \left(\frac{\Delta t}{2} u^2 \frac{\partial q}{\partial x} \right) + \mathcal{O}(\Delta t^2) \text{ if } u > 0 \\ \frac{\partial q}{\partial t} + \frac{u(q_{i+1} - q_i)}{\Delta x} &= -\frac{\partial}{\partial x} \left(\frac{\Delta t}{2} u^2 \frac{\partial q}{\partial x} \right) + \mathcal{O}(\Delta t^2) \text{ if } u < 0 \end{aligned} \quad (2.26)$$

If the spatial representations were expanded about the location i and simplified, equation 2.26 becomes:

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial x} (uq) = -\frac{\partial}{\partial x} \left(\left(\frac{\Delta t}{2} u^2 - \frac{\Delta x}{2} |u| \right) \frac{\partial q}{\partial x} \right) + \mathcal{O}(\Delta t^2, \Delta x^2) \quad (2.27)$$

where the absolute values account for the sign of the velocity. The left hand side of equation 2.27 is the original differential equation, while the term on the right hand side represents the first-order error in both space and time.

The terms on the right hand side of equation 2.27 represent the first-order error associated with donor-cell scheme. In order to use equation 2.20, the non-dimensional pseudo velocity $U_{i\pm 1/2}^{(1)}$ needs to be defined. Defining the dimensional pseudo velocity:

$$u^{(1)} \equiv \frac{1}{2} (\Delta x |u| - \Delta t u^2) \frac{1}{q} \frac{\partial q}{\partial x} \quad (2.28)$$

If both the physical velocity and the pseudo velocity are non-dimensionalized using the grid spacing Δx and the time step Δt , equation 2.28 can be written:

$$U^{(1)} = (|U| - U^2) \frac{\Delta x}{2q} \frac{\partial q}{\partial x} \quad (2.29)$$

When this pseudo velocity is used, equation 2.20 can be solved. The numerical approximation of the pseudo velocity in equation 2.29 that was used is:

$$U^{(1)} \approx \frac{q_{i+1}^{(1)} - q_i^{(1)}}{q_{i+1}^{(1)} + q_i^{(1)} + \epsilon} \left(|U_{i+1/2}| - U_{i+1/2}^2 \right) \quad (2.30)$$

where ϵ is a small value to prevent division by zero when both $q_i^{(1)}$ and $q_{i+1}^{(1)}$ are zero. Note that this only works if $q \geq 0$ everywhere or $q \leq 0$ everywhere and remains so throughout the computation.

The MPDATA scheme can be extended to an arbitrary number of iterations by replacing the iterations levels in equation 2.20 with an iteration level (k)

$$q^{(k+1)} = q_i^{(k)} - \left\{ F(q_i^{(k)}, q_{i+1}^{(k)}, U_{i+1/2}^{(k)}) - F(q_{i-1}^{(k)}, q_i^{(k)}, U_{i-1/2}^{(k)}) \right\} \text{ where } k = 0, 1, 2, \dots, K \quad (2.31)$$

where $q^{(0)} = q_i^n$, $U^{(0)} = U$, $q^{(K)} = q^{n+1}$, and the pseudo velocities at each iteration are defined as:

$$U^{(k)} = \left(|U^{(k-1)}| - U^{(k-1)} \right) \frac{\Delta x}{2q^{(k)}} \frac{\partial q^{(k)}}{\partial x} \quad (2.32)$$

When $K = 1$ this collapses to the donor-cell scheme. $K = 2$ is the most basic MPDATA scheme. If additional iterations are taken ($K > 2$) the error decreases, but the scheme remains second-order accurate. The effects of additional iterations are presented in section 2.4.

The MPDATA scheme has been presented in one-dimension only and for a constant velocity flow field. In the next section this presentation will be expanded to account for the multidimensional case on a moving grid for a variable flow field.

2.2.2 MPDATA in multiple dimensions on a moving grid

The previous section introduced the simplest form of the the MPDATA scheme. In this section the analysis is repeated to consider both multiple dimensions and a moving grid. Since the grid is allowed to move, the flow will generally be divergent. Therefore, the MPDATA scheme is reviewed for multiple dimensions, divergent flow, and extended to be used with a moving grid which is a unique contribution of this work.

In section 2.2.2.1 the coordinate transformation that was introduced in section 2.1 is used to rewrite the transport equation in integer space. It is shown that the resulting equation has the same form as the equation in the physical space except that the contravariant velocity replaces the physical velocity. In section 2.2.2.2, MPDATA is developed to solve the equation derived in section 2.2.2.1.

2.2.2.1 Advection in a moving coordinate system

In section 2.2.1 it was irrelevant whether q was an intensive or extensive property. However, it does matter in this case. The form of the equation depends on which is chosen. In both cases, for a moving grid, a divergence term appears in the equation. If q is chosen to be an intensive property, the resulting equations are more similar to the MM5 equations than if it is chosen to be extensive. Therefore, it was

chosen to be an intensive property. The multidimensional equation for q in an Eulerian reference frame if there are no source terms is:

$$\frac{\partial}{\partial t} (\rho q) + \frac{\partial}{\partial x_i} (u_i \rho q) = 0 \quad (2.33)$$

where the subscripts represent tensor indices and summation is assumed to run from 1 to p in \mathcal{R}^p space for repeated indices. Applying the chain rule and continuity equation, equation 2.33 in non-conservative form is:

$$\frac{\partial q}{\partial t} + u_i \frac{\partial q}{\partial x_i} = 0 \quad (2.34)$$

The full three-dimensional time dependent coordinate transformation that was introduced in section 2.1 is used to transform equation 2.34 into the computational domain.

Substitution of the transformation of equation 2.2 into equation 2.34 yields the equation

$$\frac{\partial q}{\partial \tau} + \hat{u}_j \frac{\partial q}{\partial \xi_j} = 0 \quad (2.35)$$

where the contravariant velocity \hat{u}_j is given by

$$\hat{u}_j = \frac{\partial \xi_j}{\partial t} + u_i \frac{\partial \xi_j}{\partial x_i} \quad (2.36)$$

In order to use the MPDATA scheme the left hand side of equation 2.35 needed to be in flux form. Thus equation 2.35 is written as

$$\frac{\partial q}{\partial \tau} + \frac{\partial}{\partial \xi_j} (\hat{u}_j q) = R \quad (2.37)$$

where the divergence term is

$$R = q \frac{\partial \hat{u}_k}{\partial \xi_k} \quad (2.38)$$

The derivatives in the contravariant velocity of equation 2.36 are computed with respect to the transformed independent variables using equation 2.7.

2.2.2.2 MPDATA in multidimensions

MPDATA was presented in section 2.2.1 in one-dimension and for a constant flow. This was done for illustration purposes only. MPDATA is a general scheme applicable to any equations that have advection terms. The most general equation can be written as:

$$\frac{\partial q}{\partial \tau} + \frac{\partial}{\partial \xi_j} (\hat{u}_j q) = R + \tilde{R} \quad (2.39)$$

where the added \tilde{R} term can be considered a combinations of source terms such as pressure gradient or diffusion terms. Smolarkiewicz and Margolin (1998) explain how to incorporate these other source

terms. Application of MPDATA in this work did not require this term and therefore it is not included in the derivation.

The only differences between equations 2.16 and 2.37 are the multidimensional nature and the divergence term R of equation 2.37. In order to determine the correct pseudo velocities for the MPDATA scheme, equation 2.37 is temporally discretized by following the lead of Smolarkiewicz (1983) as:

$$\frac{q^{n+1} - q^n}{\Delta\tau} + \frac{\partial}{\partial\xi_j} \left(\dot{u}_j^{n+1/2} q^n \right) = R^{n+1/2} \quad (2.40)$$

Note that the velocities are specified at the $n + 1/2$ time level. The time level of the physical velocities in the previous section were not specified because the velocity was assumed to be constant. However, that assumption can not be made in this case because of the moving grid. The $n + 1/2$ time level is chosen because several terms cancel later in the derivation that would not otherwise. Expanding the quantities about time level n yields:

$$\frac{\partial q}{\partial\tau} + \frac{1}{2}\Delta\tau \frac{\partial^2 q}{\partial\tau^2} + \frac{\partial}{\partial\xi_j} \left[\left(\dot{u}_j + \frac{1}{2}\Delta\tau \frac{\partial\dot{u}_j}{\partial\tau} \right) q \right] = R + \frac{1}{2}\Delta\tau \frac{\partial R}{\partial\tau} + \mathcal{O}(\Delta\tau^2) \quad (2.41)$$

As in section 2.2.1, the partial derivative with respect to τ of the lowest order terms of equation 2.41 are:

$$\frac{\partial^2 q}{\partial\tau^2} + \frac{\partial}{\partial\xi_j} \left(\dot{u}_j \frac{\partial q}{\partial\tau} + q \frac{\partial\dot{u}_j}{\partial\tau} \right) = \frac{\partial R}{\partial\tau} + \mathcal{O}(\Delta\tau) \quad (2.42)$$

Substituting the first-order terms from equation 2.41 for the $\frac{\partial q}{\partial\tau}$ term of equation 2.42 yields:

$$\frac{\partial^2 q}{\partial\tau^2} = \frac{\partial}{\partial\xi_j} \left(\dot{u}_j \dot{u}_k \frac{\partial q}{\partial\xi_k} + \dot{u}_j q \frac{\partial\dot{u}_k}{\partial\xi_k} - \dot{u}_j R - q \frac{\partial\dot{u}_j}{\partial\tau} \right) + \frac{\partial R}{\partial\tau} + \mathcal{O}(\tau) \quad (2.43)$$

Substitution of equation 2.43 into equation 2.41 yields:

$$\frac{\partial q}{\partial\tau} + \frac{\partial}{\partial\xi_j} (\dot{u}_j q) = R - \frac{1}{2}\Delta\tau \frac{\partial}{\partial\xi_j} \left[\dot{u}_j \dot{u}_k \frac{\partial q}{\partial\xi_k} + \dot{u}_j \left(q \frac{\partial\dot{u}_k}{\partial\xi_k} - R \right) \right] + \mathcal{O}(\Delta\tau^2) \quad (2.44)$$

Note that since $R = q \frac{\partial\dot{u}_k}{\partial\xi_k}$ as defined in equation 2.38 that equation 2.44 becomes:

$$\frac{\partial q}{\partial\tau} + \frac{\partial}{\partial\xi_j} (\dot{u}_j q) = R - \frac{1}{2}\Delta\tau \frac{\partial}{\partial\xi_j} \left(\dot{u}_j \dot{u}_k \frac{\partial q}{\partial\xi_k} \right) + \mathcal{O}(\Delta\tau^2) \quad (2.45)$$

From equation 2.45, it is clear that additional cross terms appeared when $j \neq k$ in the pseudo velocities compared to equation 2.25. The treatment of these terms is the major difference between the one-dimensional MPDATA and multi-dimensional MPDATA schemes. The q quantities are stored at the cell centers while the velocities are stored at the cell faces. Figure 2.2 illustrates the location of the velocity component storage.

There were four cases to consider when determining the spatial correction terms. In the grid space notation the four cases can be represented as:

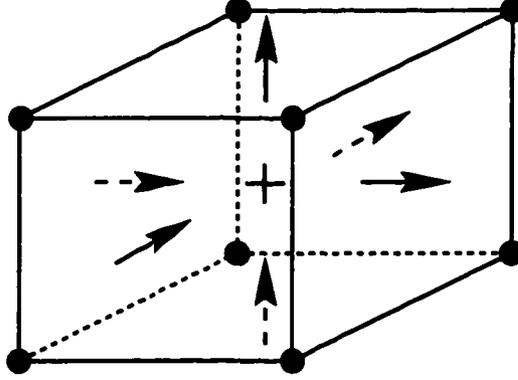


Figure 2.2 Property storage locations for the MPDATA scheme. The cell center is marked with a +. The corner points are marked with the •'s. The →'s, ↗'s and ↑'s mark the $i + 1/2$, $j + 1/2$ and $k + 1/2$ cell face centers respectively.

1. $\hat{U}_{i+1/2} > 0, \hat{U}_{i-1/2} > 0$
2. $\hat{U}_{i+1/2} < 0, \hat{U}_{i-1/2} < 0$
3. $\hat{U}_{i+1/2} > 0, \hat{U}_{i-1/2} < 0$
4. $\hat{U}_{i+1/2} < 0, \hat{U}_{i-1/2} > 0$

Repeating the spatial analysis of section 2.2.1 for cases 1 and 2 while assuming that the $\hat{U}_{i+1/2} \neq \hat{U}_{i-1/2}$, yields the same spatial correction term as before. For case 3 the spatial term does not appear and for case 4 it is twice as large. Numerical experiments indicated that ignoring cases 3 and 4 has negligible effect on the solution, because the only time that either situation occurs is when the flow reverses direction and therefore the velocities and hence the fluxes at both cell faces are small. However, Smolarkiewicz and Margolin (1998) indicate that to protect from the worst case scenario of case 4 that the multidimensional Courant number should be

$$\sum_{i=1}^p |\hat{U}_i| \leq 0.5 \quad (2.46)$$

in a \mathcal{R}^p space for acceptable implementation.

Once the spatial discretization was included equation 2.45 becomes

$$\frac{\partial q}{\partial \tau} + \frac{\partial}{\partial \xi_j} (\hat{u}_j q) = R - \frac{1}{2} \Delta \tau \frac{\partial}{\partial \xi_j} \left(\hat{u}_j \hat{u}_k \frac{\partial q}{\partial \xi_k} \right) + \frac{1}{2} \sum_{j=1}^p \Delta \xi_j \frac{\partial}{\partial \xi_j} \left(|\hat{u}_j| \frac{\partial q}{\partial \xi_j} \right) + \mathcal{O}(\Delta \tau^2, \Delta \xi^2) \quad (2.47)$$

Following the same steps as in section 2.2.1 the pseudo velocities are defined to be

$$\hat{U}_j^{(k+1)} = \left| \hat{U}_j^{(k)} \right| \frac{\Delta \xi_j}{2q} \frac{\partial q^k}{\partial \xi_j} - \sum_{m=1}^p \hat{U}_j^{(k)} \hat{U}_m^{(k)} \frac{\Delta \xi_m}{2q^{(k)}} \frac{\partial q^{(k)}}{\partial \xi_m} \quad (2.48)$$

where $\hat{U}_j^{(k)} = \frac{\Delta r}{\Delta \xi} \hat{u}_j^{(k)}$, where there is no summation over j in the first term on the right hand side. Additionally, the (k) superscripts in the equation 2.48 represent the iteration level. These pseudo velocities are clearer if written explicitly. The pseudo velocity in the first direction is

$$\hat{U}^{(k+1)} = \left(\left| \hat{U}^{(k)} \right| - \hat{U}^{(k)} \hat{U}^{(k)} \right) \frac{\Delta \xi}{2q^{(k)}} \frac{\partial q^{(k)}}{\partial \xi} - \hat{U}^{(k)} \hat{V}^{(k)} \frac{\Delta \eta}{2q^{(k)}} \frac{\partial q^{(k)}}{\partial \eta} - \hat{U}^{(k)} \hat{W}^{(k)} \frac{\Delta \zeta}{2q^{(k)}} \frac{\partial q^{(k)}}{\partial \zeta} \quad (2.49)$$

where the $\hat{U}^{(k)}$, $\hat{V}^{(k)}$ and $\hat{W}^{(k)}$ pseudo velocities are in the ξ , η and ζ direction respectively. The other pseudo velocities are symmetric permutations of equation 2.49.

The pseudo velocities in the ξ direction were approximated as

$$\begin{aligned} \hat{U}_{i+1/2,j,k}^{(k)} &\approx \left(\left| \hat{U}_{i+1/2,j,k}^{(k)} \right| - \hat{U}_{i+1/2,j,k}^{(k)} \hat{U}_{i+1/2,j,k}^{(k)} \right) \frac{q_{i+1,j,k}^{(k)} - q_{i,j,k}^{(k)}}{q_{i+1,j,k}^{(k)} + q_{i,j,k}^{(k)} + \epsilon} \\ &\quad - \frac{1}{2} \hat{U}_{i+1/2,j,k}^{(k)} \hat{V}_{i+1/2,j,k}^{(k)} \frac{q_{i,j+1,k}^{(k)} + q_{i+1,j+1,k}^{(k)} - q_{i,j-1,k}^{(k)} - q_{i+1,j-1,k}^{(k)}}{q_{i,j+1,k}^{(k)} + q_{i+1,j+1,k}^{(k)} + q_{i,j-1,k}^{(k)} + q_{i+1,j-1,k}^{(k)} + \epsilon} \\ &\quad - \frac{1}{2} \hat{U}_{i+1/2,j,k}^{(k)} \hat{W}_{i+1/2,j,k}^{(k)} \frac{q_{i,j,k+1}^{(k)} + q_{i+1,j,k+1}^{(k)} - q_{i,j,k-1}^{(k)} - q_{i+1,j,k-1}^{(k)}}{q_{i,j,k+1}^{(k)} + q_{i+1,j,k+1}^{(k)} + q_{i,j,k-1}^{(k)} + q_{i+1,j,k-1}^{(k)} + \epsilon} \end{aligned} \quad (2.50)$$

where

$$\begin{aligned} \hat{V}_{i+1/2,j,k}^{(k)} &\approx \frac{1}{4} \left(\hat{V}_{i,j+1/2,k}^{(k)} + \hat{V}_{i+1,j+1/2,k}^{(k)} + \hat{V}_{i,j-1/2,k}^{(k)} + \hat{V}_{i+1,j-1/2,k}^{(k)} \right) \\ \hat{W}_{i+1/2,j,k}^{(k)} &\approx \frac{1}{4} \left(\hat{W}_{i,j,k+1/2}^{(k)} + \hat{W}_{i+1,kj+1/2}^{(k)} + \hat{V}_{i,j,k-1/2}^{(k)} + \hat{V}_{i+1,j,k-1/2}^{(k)} \right) \end{aligned} \quad (2.51)$$

$\hat{V}_{i,j+1/2,k}^{(k)}$ and $\hat{W}_{i,j,k+1/2}^{(k)}$ were found by considering symmetric permutations of equations 2.50 and 2.51.

The multidimensional MPDATA scheme for a divergent flow is:

$$\begin{aligned} q_i^{(k+1)} &= q_i^{(k)} - \left\{ F(q_{i,j,k}^{(k)}, q_{i+1,j,k}^{(k)}, \hat{U}_{i+1/2,j,k}^{(k)}) - F(q_{i-1,j,k}^{(k)}, q_{i,j,k}^{(k)}, \hat{U}_{i-1/2,j,k}^{(k)}) \right\} - \\ &\quad \left\{ F(q_{i,j,k}^{(k)}, q_{i,j+1,k}^{(k)}, \hat{V}_{i,j+1/2,k}^{(k)}) - F(q_{i,j-1,k}^{(k)}, q_{i,j,k}^{(k)}, \hat{V}_{i,j-1/2,k}^{(k)}) \right\} - \\ &\quad \left\{ F(q_{i,j,k}^{(k)}, q_{i,j,k+1}^{(k)}, \hat{W}_{i,j,k+1/2}^{(k)}) - F(q_{i,j,k-1}^{(k)}, q_{i,j,k}^{(k)}, \hat{W}_{i,j,k-1/2}^{(k)}) \right\} + R_i^{n+1/2} \end{aligned} \quad (2.52)$$

where $k = 0 \dots K$ and $R_i^{n+1/2}$ is the divergence of the flow when $k = 0$ and zero when $k > 0$. Recall $q^{(0)} = q^n$, $q^{(K)} = q^{n+1}$, and \hat{U}_i quantities are contravariant velocities.

2.2.3 Extensions to MPDATA

The addition of the ‘‘third-order’’ and the recursive pseudo velocity (RPV) options improve the accuracy and the efficiency of the MPDATA scheme, respectively. The ‘‘third-order’’ option only includes some of the ‘‘third-order’’ correction terms, but still has a significant positive impact on the accuracy of the scheme. The RPV option improves the efficiency of the scheme by using one correction

step that is an approximation to an infinite number of correction steps. Both of these options are discussed briefly in this section and are explained in detail in appendix A.

Although MPDATA is capable of handling the divergent flows and source terms, these effects were neglected in the development of the “third-order” option. The scheme remains second-order accurate with the leading order error terms a function of the gradients of velocity and source terms. Although the scheme remains second-order these extra correction terms cause the numerical errors to be distributed more symmetrically and the global error is decreased. This is evident by comparing figure 2.21 to 2.22 in section 2.4.2.2.

In the development of MPDATA the prognostic quantity q was always taken at the newest iteration $q^{(k)}$ when calculating the pseudo velocities as shown in equations 2.32 and 2.48. Formally any value of q between q^n and q^{n+1} could have been used since the correction terms were always calculated using a donor-cell step which is only first-order accurate. The choice to use the newest approximation for q^{n+1} was made because numerical experiments demonstrated that the error of the scheme was the least when this choice was made. However, the option to choose the value q^n for every iteration, allowed Margolin and Smolarkiewicz (1989) to develop a strictly two step scheme. The first step is the first-order donor cell approximation that is implemented in all versions of MPDATA. The pseudo velocities for the second step are calculated so that when used with a donor-cell step the correction is an approximation to an infinite number of MPDATA correction iterations. The results of using the RPV option are approximately equivalent to using four iterations of the MPDATA scheme with a computational cost of 40%.

This concludes the discussion of the MPDATA scheme and its different options. In summary the scheme is second-order accurate in time and space, it is sign-preserving, and computationally efficient. The “third-order” error terms and RPV options were introduced. The scheme was developed assuming a dynamic grid which is a unique contribution of this work. In section 2.4 results from different permutations of the options are compared for one- and two-dimensional models on static and dynamic grids.

2.3 Dynamic grid adaptation

This section deals with how logically to move a finite difference grid so that the global error of a numerical scheme is minimized. Both one- and two-dimensional grid generators are discussed. In section 2.3.1 a literature review covers the various approaches to dynamic grid generation. Section 2.3.2 introduces one- and two-dimensional grid generators. The one-dimensional grid generator is examined

in detail in section 2.3.3. Section 2.3.4 details the two-dimensional version of the grid generator.

2.3.1 Literature review

There has been a large amount of information published about adaptive grid techniques. As explained in chapter 1, there are several different types of techniques that can be called dynamic adaptive grid (DAG) techniques. This review, in section 2.3.1.1 focuses solely the type of method implemented in this work which is the grid point redistribution technique. More extensive reviews of this method can be found in Thompson (1985), Eiseman (1987), and Hawken et al. (1991). In section 2.3.1.2, adaptive grid techniques that have been applied to problems of interest in the atmospheric science community will be reviewed without regard to the type of adaptive technique used.

2.3.1.1 Grid point redistribution techniques

In brief, grid point redistribution techniques are methods that conserve the number of grid points and move them to certain areas to increase the local resolution at the expense of the grid resolution in other areas. One important application of dynamic adaptive grids is the moving boundary problem. For example, problems of freezing and melting have time dependent boundaries. A more extensive review of these problems may be found in Yao and Prusa (1989). Another application is to cluster points in regions where the local error of the numerical solution is large. This often happens in areas of steep gradients where the grid points are not close enough to resolve the gradient, or in areas where the second derivative is large and either large amounts of dissipation or dispersion takes place. This second type of problem is the type that was addressed in this work.

Slater (1992) has categorized adaptive grid techniques according to the type of coupling the solution scheme has with the grid movement scheme. He categorizes all schemes as either completely coupled, strongly coupled, weakly coupled, or uncoupled.

In a completely coupled scheme, the prognostic equations and the equations that determine the grid point placement are viewed as one set of equations and are solved simultaneously. The moving finite-element method is an example of a completely coupled approach. This method was developed by Miller and Miller (1981) and Miller (1981). Only a few one-dimensional examples of completely coupled schemes were found using finite difference type schemes by Winkler et al. (1985) and Dorfi and Drury (1987).

When the strongly coupled approach is used, an approximate solution of the prognostic equations at the new time step is determined using the grid from the old time step. This approximate solution

is used to generate a new grid, which is then used to calculate a better approximation for the solution at the new time step. Strongly coupled systems can be implemented in an iterative fashion where progressively better solution and grid matches are computed at each iteration.

Weakly coupled systems use the solution features at the current time step to determine the grid to be used for the integration from the current to the next time step. This is the type of approach used in this work.

Uncoupled systems are what are typically used to model atmospheric systems. The grid does not respond to features in the flow field and is therefore remains stationary throughout the integration.

There are certain criteria that must be met by any adaptive grid technique. A modified list of these criteria by Thompson (1985) is:

1. Grid points need to be able to move locally while still retaining an orderly pattern. This excludes grids which can fold back on themselves which would be the case if topologically parallel lines were allowed to cross.
2. The grid points need to conform to the boundaries of the domain. The physical domain has to be completely filled by the computational mesh.
3. There must be a means of representing continuous functions on an uneven and temporally changing grid.
4. There must a means of communicating how the grid points move in response to changes in the computed solution.

The first of these criteria dictate that local control of grid point movement is required in order to respond to local phenomena, and the second criteria dictates global quantities need to also be met so that the boundary conditions are satisfied. This restricts possible grid generation techniques to interpolation type techniques that are referred to as algebraic methods or elliptic partial differential equation solvers.

There were several types of grid point redistribution techniques found in the literature. Each of these methods meet the criteria listed above and can be viewed as an equidistribution of some global quantity that is weighted, hopefully, by some measure of the global error associated with the numerical scheme that is being used to compute the solution.

Ideally, given a numerical method that would be used to solve the prognostic equations, the lowest order error terms would be derived, assuming an uneven and time dependent mesh. These error terms could then serve as a weight function that could be equally distributed by some means resulting in a set

of equations, that when solved, would yield a desirable grid. Prusa (1997) analyzed a one-dimensional passive advection problem with centered space and time discretizations. Thirteen unique leading error terms emerged. In general, these terms were non-linear. In the vast majority of the cases found in the literature, a more heuristic measure of the error was used because of the difficulty identifying which of these terms was dominant or including them all. Some of the more popular error measures were some combination of the first and second derivatives of the prognostic variable and the arc length associated with prognostic variable.

Some of the grid generation techniques ultimately result in the same equations, yet were arrived at from different points of view. The following sections list the different techniques and explain their distinguishing characteristics. All of the details about these methods are not given, and the reader is encouraged to examine the references for additional details.

2.3.1.1.1 Equidistribution in one-dimension: Virtually all of the methods for one-dimensional cases are based on equidistribution of some global quantity. Equidistribution means that given some measure of the error as a function of space, called the weight function, the grid points are adjusted so the integrated quantity between any two grid points is a constant. Grid points B and C must be closer together in figure 2.3 than grid points A and B so that the two shaded areas under the weight function curve are the same. Eiseman (1987) explains twelve different ways to view and formulate equidistribution statements in one-dimension. Some of these formulations have advantages over others. Eiseman (1987) points out that the “backward global integral” statement has certain advantages, since it can be solved by quadrature and interpolation and no iteration is required. This is the formulation that is used in the one-dimensional work for this study and is developed in detail in section 2.3.3. Thompson et al. (1985) explain how four of the equidistribution statements can be seen as the Euler-Lagrange equations of a global integral that is a function of the weighting. Variational approaches are described in section 2.3.1.1.4 illustrating the point that although there are different approaches often they result in the same equations from different view points.

2.3.1.1.2 Equidistribution in multiple dimensions: Dwyer et al. (1980), Dwyer et al. (1982), and Dwyer (1984) applied the one-dimensional equidistribution principles along a fixed set of lines in multidimensional problems. Thus the grid only moved in one coordinate direction. Although, in general, grid adaptation may be desirable in the other directions as well, they obtained impressive results in problems dealing with external flow separation, flame propagation, and shock/boundary layer interactions. This method was extended by Gnoffo (1982) and Eiseman (1985) to multiple dimensions.

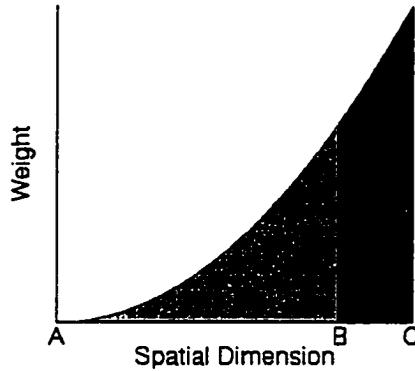


Figure 2.3 Illustration of equidistribution principle. Grid point B must adjust side to side to make both shaded areas under the weight function curve of equal area.

Since each coordinate was treated serially, iteration was required to generate the grid. This method is similar to the alternating direction implicit (ADI) methods for solving elliptic boundary value problems.

2.3.1.1.3 Elliptic grid generation systems: It is explained in Thompson et al. (1985) that a general elliptic partial differential equation that satisfies the necessary conditions is the Poisson equation

$$\nabla^2 \xi^i = g^{jk} P_{jk}^i \quad (2.53)$$

where $\xi^i = \xi^i(x_j)$ is a coordinate in a Cartesian computational domain and x_j is the stretched coordinate in the physical domain. The g^{jk} is the metric tensor for the transformation from the physical to the computational domain. P_{jk}^i are forcing functions that are specified to control the grid stretching. Thompson et al. (1985) give development of appropriate forcing functions to cluster points towards certain points or lines, although Prusa (1999) indicates that it can be difficult to prescribe these functions a priori. This may explain why many authors mention the technique, but few actually specify the forcing functions.

2.3.1.1.4 Variational approaches: Since the goal in DAG techniques is to minimize a weighted global quantity, the use of calculus of variations is applicable.

This approach involves several steps:

1. Identify certain criteria that are desirable properties of the grid.
2. Create an equation that gives some measure of the combination of these properties.

3. Determine the Euler–Lagrange equations to determine the extrema of these properties.
4. Solve the Euler–Lagrange equations to determine the grid point locations.

The first of these criteria is subjective. However, there are certain criteria that must be included. These necessary criteria are grid smoothness and some type of weighting. Without a grid smoothness criteria, grid points would be allowed to collapse to a point and certain regions of the domain would be void of points. Without the weighting an evenly spaced grid with no grid stretching would result as this is the smoothest possible grid, and the grid would not be dynamic since there would be no mechanism to communicate the changes within the solution field to the grid. Other criteria that could be applied to multidimensional grid include a measure of the orthogonality of the grid, a measure of the aspect ratio of grid cells, a measure of the alignment of the grid lines with stream lines, and the Lagrangian transport of a particle in the flow. Winslow (1967) developed a method of grid generation based on grid smoothness and a weighting function for triangular meshes. Brackbill and Saltzman (1982) extended the work to include an orthogonality feature. This is the grid generation technique that was used and will be developed, in detail, in section 2.3.4. Yanenko et al. (1979) included grid smoothness, a weight function and a Lagrangian term. This use of the Lagrangian term, $|U_1 - \frac{\partial X_1}{\partial t}|^2 + |U_2 - \frac{\partial X_2}{\partial t}|^2$ as presented in Hawken et al. (1991), shows insight lacking in other work. It was shown that except to conform to the boundary the “best” grid point movement would be along Lagrangian trajectories. This is discussed further in section 2.4.1.2.

In order to apply the variational approach the problem is transformed into computational space that is uniform and orthogonal. Through this transformation, measures of the grid properties can be expressed and combined in a linear manner. These equations are then minimized by determining the Euler–Lagrange equations. For example, in two dimensions, given a functional $f = f(\xi, \eta, x, x_\xi, x_\eta)$ where $x = x(\xi, \eta)$, $x_\xi = \frac{\partial x}{\partial \xi}$, and $x_\eta = \frac{\partial x}{\partial \eta}$. The global measure of f is

$$\int \int_D f(\xi, \eta, x, x_\xi, x_\eta) d\xi d\eta \quad (2.54)$$

which can be minimized with respect to w by applying the operator

$$\frac{\partial f}{\partial x} - \frac{\partial}{\partial \xi} \left(\frac{\partial f}{\partial x_\xi} \right) - \frac{\partial}{\partial \eta} \left(\frac{\partial f}{\partial x_\eta} \right) = 0 \quad (2.55)$$

This is the Euler–Lagrange equation in multiple dimensions and its development can be found in Weinstock (1974).

Once the Euler–Lagrange equations are found they are discretized and solved numerically to find the location of the grid points such that the weight function is equally distributed among the grid points.

2.3.1.1.5 Grid speed methods: All of the techniques discussed thus far deal with techniques that directly determine the location of the grid points. In these methods a new grid is periodically calculated and either the solution from the old grid is interpolated to the new grid or the dynamic equations are cast into a form to allow solution on a moving grid. In the second case, the grid point velocities are required. This presents several questions:

1. How does one accurately calculate the grid point velocities?
2. How is the grid point movement constrained by the time step?

If only the original and the final grid point locations are known a forward finite difference

$$\frac{\partial x}{\partial t} \approx \frac{x^{n+1} - x^n}{\Delta t} \quad (2.56)$$

will yield an approximation to the grid speed. In a strictly explicit scheme, where the velocities are taken at time level n , or a strictly implicit scheme, where the velocities are taken at $n + 1$, this approximation is only first order accurate. Grid point locations at additional time levels may be retained and used to create a higher order approximation to the grid point velocities, but retaining this additional information is computationally expensive. It was seen in section 2.2.2.2 that the MPDATA scheme used velocities at time level $n + 1/2$, therefore the finite difference approximation became centered and second-order accurate. This is yet another reason why the MPDATA scheme is well suited to DAG.

The answer to the second question above is not as simple to explain. If for some pathological reason the solution changes abruptly between two time steps, there is nothing in the previously discussed techniques preventing a grid point from moving a large distance over the single time step. This is because the methods discussed to this point have no time dependency. They are all strictly elliptic problems and each new grid is completely dependent on the problem solution at some point in time with no influence from the solution at earlier times. If an implicit scheme is used, this does not pose much of a problem. However, if an explicit scheme like MPDATA is used, a violation of the CFL condition can occur. To prevent this a limit on the extent of movement that a grid point can undergo from one time step to next can be imposed. This is discussed in greater detail in section 2.4.1.2.

Another solution to this problem is to use the method of Hindman and Spencer (1983), Holcomb and Hindman (1984), and Slater and Liou (1995) where the equations that specify the grid point locations are differentiated with respect to time. Therefore, instead of solving a set of equations that determine the grid point locations directly, the grid point velocities are solved for. In order to find the location of the grid points, numerical quadrature is used. It was found that with this method the

original, undifferentiated equations had to be used, periodically, in order to keep the clustered points from drifting apart. These methods would appear to answer both of these questions, since a higher order method can be used to solve the grid speed equations and since quadrature is used to determine the grid point locations, as the time step goes to zero, the change in the grid position will also.

2.3.1.1.6 Attraction and repulsion: Another method that yields the grid point velocity instead of the grid point location is that of Anderson and Rai (1982) who used a grid adaptation method based on attractive or repulsive forces determined at each grid point. The forces are determined by comparing the local value of the weight function at a grid point to the average value of the weights over the whole domain. Points that have a higher than average value are attracted while those with less than average are forced apart. Points are prevented from collapsing in on each other because the system becomes very stiff as the points approach each other and therefore in practice they never coalesce. Attractive forces become repulsive forces as the points approach each other. Damping is necessary to prevent grid oscillations.

Another, similar technique by Nakahashi and Deiwert (1986), determines inter-nodal equations as if there were tension springs between the nodes and torsional springs at the nodes. The tension springs control the grid clustering and the torsional springs gives the method of controlling the grid orthogonality. In multiple dimensions, the one-dimensional model is applied iteratively to each coordinate direction in sequence.

2.3.1.2 Dynamic adaptive grids applied to atmospheric science problems

Dietachmayer and Droegemeier (1992) used the variational approach of Brackbill and Saltzman (1982) to solve the viscous Burger's equations in one-dimension, and a frontogenesis problem in two-dimensions for which an analytically solution is known, solid body rotation of four cones, and the evolution of a buoyant thermal. This work showed that dynamic grid adaptation was applicable to atmospheric flows and how to implement it. This work was extended by Dietachmayer (1992) to evaluate the efficiency of the method. A different two-dimensional grid generation technique than was used by Dietachmayer and Droegemeier (1992) was implemented based on equidistribution of an area weighted function. The model solved the shallow water equations initialized with three well spaced vortices. The conclusions were that the orthogonality constraint in the Brackbill and Saltzman (1982) method is not essential for atmospheric flows. The equidistribution grid generator was simpler and faster than more sophisticated grid generation techniques. Essentially equivalent computations were

performed three times faster using adaptive grid techniques over fixed grid methods. By updating the grid periodically, instead of at every time step, modest gains in efficiency were possible.

This work was extended by Fiedler and Trapp (1993). Two- and three-dimensional buoyant thermals were computed in a stably stratified fluid. The dynamic adaptive grid was generated by solving the elliptic equations for a coarse grid and then interpolating the intermediate lines to produce the final fine grid. Fine grid to coarse grid ratios of up to 8:1 were used. Although there was some degradation of the solution as the this ratio became larger, substantial computational savings were achieved.

Prusa et al. (1996) used moving grids in the simulation of propagation and breaking of high altitude gravity waves. A moving boundary technique similar to the methods used for freezing and melting problems that were mentioned in section 2.3.1.1 was used. The lower boundary was a function of space and time and required a time dependent grid. Therefore, the governing equations were transformed from the physical domain into an orthogonal, evenly spaced computational domain. The algebraic transformation used was an extension of the terrain following coordinate system.

Skamarock and Klemp (1993) used moving nested grids as a form of a dynamic adaptive grid. In this method finer Cartesian grids overlay coarser Cartesian grids. The finer grids were not required to be orthogonal with the coarser grids. Grids could be nested inside other nested grids so that very fine resolution was possible. As the integration continued the nested grids could move, change size, be added or removed as directed by the flow attributes. Care had to be taken establishing the boundary conditions for the finer meshes. The abrupt change in the grid resolution posed problems at the boundaries. Conservative interpolation was needed in order to establish the boundary conditions for the finer domains. Flow features of the finer grids were not allowed to propagate onto the coarser grids.

Two-dimensional simulations of a cold bubble and a collapsing cold pool were performed. An algorithm that automatically determined the nested grid movement was implemented. Accurate solutions were obtained using this method although non-physical features also were excited by either the grid scheme or the numerical method. Three-dimensional simulations of super-cell convective storms were performed. In the three-dimensional simulations the grid movements were not fully automatic. Rather the simulation was stopped and the movement of the grids was partially directed through user intervention. Skamarock et al. (1994) used the method while studying the three-dimensional evolution of long-lived squall lines.

Finley et al. (1998a) and Finley et al. (1998b) used moving nested grids to simulate tornados. Although the nested grids had the ability to move, user interaction was required during the simulation to direct the grid movement.

Behrens (1996) used an unstructured grid with local grid refinement and a semi-Lagrangian approach in a parallel computing environment to compute the rotating slotted cylinder about some point that is described by Zalesak (1979). The parallel semi-Lagrangian scheme proved to be efficient but the serial implementation of the grid generation technique was detrimental to computational efficiency.

Tomlin et al. (1997) used unstructured dynamic grids to simulate atmospheric reaction flow problems. The grid used hexagonal elements rather than triangular elements as are often used in unstructured finite volume codes. A linear advection test case and plume dispersions of high NO_x emissions which are typical of a power plant were performed. In conjunction with the advection problem, non-linear source terms associated with chemical reactions were included. It was concluded that these types of problems could not be adequately solved without some mechanism to model the small scale phenomena that were present at the initial emissions. The DAG approach was well suited to model this small scale phenomena.

2.3.2 Introduction to one- and two-dimensional grid generators

The remainder of this section explains, in detail, the techniques that were used to determine the grid point distribution for this work. As state earlier, equidistribution is used in the one-dimensional case while the variational method of Brackbill and Saltzman (1982) is used in the two-dimensional case.

Since the grid is allowed to vary in time, a new grid is generated at each time step. The grid generator clusters grid points in regions where the weight function is large. It has proved impractical to base the weight function on the numerical error of the scheme, as was discussed in section 2.3.1.1. Therefore, it was decided that criteria for producing a "good grid" would be used. Thompson et al. (1985) explain that the solution of the prognostic equation is relatively insensitive to perturbations of the grid points about their optimum location. Therefore, it is not so important that the grid generator provide a very accurate estimate of the numerical error nor obtain a highly precise solution to the equation that specifies the grid distribution. However, it is important to know the location of the grid points to high precision since the gradient of the grid point distribution directly effects the numerical scheme for the advection problem.

The dispersive and diffusive deficiencies of finite difference schemes are typically worst in regions of high gradients and second derivatives. A good grid is defined as one that clusters points in these regions. The grid smoothness, orthogonality, first and second derivatives are the criteria used in the development of this study's grid generators.

One- and two-dimensional grid generators were developed. In the three dimensional atmospheric

tracer model that was coupled with MM5, the vertical grid adjustment was treated independently of the horizontal grid adjustment. The one-dimensional grid generator was applicable to the vertical direction, while the two-dimensional one was used in the horizontal direction. The description of these two grid generators are described in sections 2.3.3 and 2.3.4.

2.3.3 One-dimensional grid generation

The one-dimensional grid generator was used in the early parts of the study to learn about the dynamic grid technique and how to implement it with the MPDATA scheme. This grid generator proved to be very efficient since its solver consisted of a trapezoidal quadrature routine and a one-dimensional interpolator to solve an integral equation rather than an iterative technique to solve a differential equation.

2.3.3.1 Weight function

Both the one- and two-dimensional grids are generated by specifying a weight function that is an approximation of the error associated with numerical scheme being used. Since it was not feasible to directly minimize the truncation error associated with a grid that is non-uniform in space and in time dependent, the sum of the absolute values of the first and second derivatives of the prognostic variable is used. The one-dimensional weight function was defined as:

$$\tilde{w} = \left| \frac{dq}{dx} \right| + \left| \frac{d^2q}{dx^2} \right| \quad (2.57)$$

Using equations 2.9 and 2.10 the derivatives in equation 2.57 were expressed with respect to the computational domain coordinates.

Grid generation is based solely on the prognostic field variable at time step n . However, it is used to integrate forward in time from time step n to $n + 1$. Therefore, the weight function is smoothed to spread the influence of the grid point clustering out in front of the propagating phenomenon. The smoothing is done by iterative applications of a simple Laplacian type function

$$\begin{aligned} \tilde{w}_i^0 &= \tilde{w}_i \\ \tilde{w}_i^k &= \frac{1}{4} (\tilde{w}_{i-1}^{k-1} + 2\tilde{w}_i^{k-1} + \tilde{w}_{i+1}^{k-1}) \text{ where } k = 1 \dots K \\ \tilde{w}_i^n &= \tilde{w}_i^K \end{aligned} \quad (2.58)$$

where \tilde{w}_i represents the smoothed weight function.

As discussed earlier, the lack of temporal influence on the grid generator created a situation where the CFL condition could be violated solely by grid point movement. In section 2.4.1.2 it is shown

that the amount that a grid point is capable of moving without violating the CFL condition is directly proportional to the magnitude of the metric term $\frac{\partial x}{\partial \xi}$ at the grid point. The grid smoothing widens the grid spacing thus increasing this term and allows a grid point more freedom of movement.

Dietachmayer and Droegemeier (1992) indicate that smoothing also prevents grid point oscillations. It was found that smoothing the weight function did prevent grid oscillations in some cases, but not all. These oscillations were present in regions with little grid clustering and may have resulted from numerical errors. When the weight function w was of very small the, $\frac{dx}{dx}$ became large and any error in the numerical solution of equation 2.61 causes large changes in the grid spacing. For the test cases presented, these oscillations were not observed.

The degree of grid stretching is controlled by an input parameter λ . It is desirable to non-dimensionalize both the dependent and independent variables. If this is not done, numerical experiments to determine appropriate values of λ are need for each new application. The physical domain was scaled between 0 and 1. The value of the prognostic variable q was scaled by the maximum value of q at the beginning of the integration and the weight function was scaled between 0 and 1 each time a new grid was generated:

$$\tilde{w}_i = \frac{\tilde{w}_i^n - \min_{\forall i}(\tilde{w}_i^n)}{\max_{\forall i}(\tilde{w}_i^n) - \min_{\forall i}(\tilde{w}_i^n)} \quad (2.59)$$

where \tilde{w} was the smoothed and scaled weight function.

For the test case considered it was found the the weight function would strongly favor the section of the profile with the highest second derivative and virtually neglect lesser local extrema that were still important areas to be resolved. Therefore, after the weight function was scaled between 0 and 1, it was raised to the 1/4 power. This had a tendency compress the local maxima closer to one, rectifying the problem. Finally, it is necessary to translate the weight function so that its maximum value was greater than zero. One was arbitrarily picked for this purpose. Thus the weight function is:

$$w_i = 1 + \lambda \tilde{w}_i^{1/4} \quad (2.60)$$

where λ is the grid stretching parameter specified by the user to control the extent of grid adaptation. If the weight function is allowed to be zero that region could be completely void of grid points, which was obviously undesirable. If λ were set to 0 the weight function is a constant value and no grid stretching occurs. How the grid stretching varies as a function of λ is explained in section 2.3.3.4.

2.3.3.2 Grid generation equation

Thompson et al. (1985) explain a method of grid generation based on equidistribution of a weighted function. The Euler-Lagrange equation for minimization of the given measured quantity can be represented as

$$\frac{dx}{d\xi} w = c \quad (2.61)$$

where c is a constant. The choice of which variable w is a function of, x or ξ , and which variable, x or ξ , is considered dependent and which independent is somewhat arbitrary. However, the choice does impact the solution method. If w is taken as a function of x which is also chosen as the independent variable, the problem can be solved without iterating. Therefore equation 2.61 was rewritten as

$$\frac{d\xi}{dx} = \frac{w}{c} \quad (2.62)$$

2.3.3.3 Grid generation equation solution

Equation 2.62 was solved by the use of numerical integration over the entire domain and I one-dimensional interpolations, where I is the number of grid points in the domain. It is convenient to define a quantity

$$\sigma(x) = \int_0^x w(\tilde{x}) d\tilde{x} \quad (2.63)$$

$w(x)$ is always positive so $\sigma(x)$ has a positive slope over the whole domain which was an important feature as will be seen below. The value of $\sigma_i = \sigma(x_i)$ is determined at the grid points x_i^n from the previous time step. Equation 2.63 is numerically integrated to find the values of σ_i using the trapezoidal rule

$$\begin{aligned} \sigma_1 &= 0 \\ \sigma_i &= \sigma_{i-1} + \frac{1}{2} (w_{i-1} + w_i) (x_i^n - x_{i-1}^n) \quad \text{where } i = 1 \dots I \end{aligned} \quad (2.64)$$

The constant in equation 2.62 is determined by integrating over the entire domain

$$c = \frac{\int_0^L w(x) dx}{\int_1^I d\xi} = \frac{1}{I-1} \int_0^L w(x) dx = \frac{\sigma_I}{I-1} \quad (2.65)$$

Since the values of the computational coordinate at the grid points ξ_i were integers, the problem became finding the new grid points locations x_i^{n+1} in equation 2.62 such that $\xi_i = i$. This is represented as

$$\int_1^i d\xi = \frac{1}{c} \int_0^{x_i^{n+1}} w(x) dx \quad (2.66)$$

The left hand side of equation 2.66 is trivially integrated and the integral on the right hand side is $\sigma(x_i^{n+1})$. Since the constant c was known from equation 2.65, the values of $\sigma(x_i^{n+1})$ are known and

equal to $c(i - 1)$. Since σ is a one-to-one function, as mentioned before, the role of dependent and independent variables are reversed such that

$$x = x(\sigma) \tag{2.67}$$

Since $x_i^n(\sigma)$, σ^n , and σ^{n+1} are known the problem is one of interpolation to determine $x_i(\sigma^{n+1})$ given the function in equation 2.67. This problem was solved using linear interpolation.

2.3.3.4 Characteristics of the one-dimensional grid generator

In order to demonstrate one-dimensional grid generation, an even mesh was initialized with a Gaussian pulse. Several experiments were run to determine the characteristics of the grid generator, such as how the grid generator varied with λ , the number of grid points I , and the number of smoothing passes K in equation 2.58.

Using the left axis, figure 2.4, shows an initial profile on an evenly spaced mesh. The graph of $\frac{dx}{dx}$, using the right axis, shows the grid spacing after the one-dimensional grid generator was run. The grid was generated using values of $\lambda = 100$, $I = 101$ and $K = 4$ smoothing passes. The resulting grid had the desired properties. The grid points clustered around the steep gradients and sharp curvatures of the profile. The minimum grid spacing was centered around the center of the profile where the second derivative with respect to space was highest. One through four and ten smoothing passes were tried (results not shown). There was little difference between using one smoothing pass and four. Ten smoothing passes increased the minimum grid spacing and increased the number of increments with small grid spacing.

Figure 2.5 shows results from holding constant the number of smoothing passes $K = 4$ while varying the value of the stretching parameter λ and the number of grid points I . The same Gaussian profile as in figure 2.4 was used throughout the study. Both the minimum grid spacing and the maximum grid spacing are represented in figure 2.5A. This panel shows that the grid spacing was a strong function of λ up to approximately $\lambda = 100$. Above 100 the stretching parameter decreased the minimum grid spacing but not to nearly as great an extent. The effect on the minimum grid spacing of varying the number of grid points I and the stretching parameter λ is shown in figure 2.5B. Since these curves are approximately linear on a log-log scale a power series represented them well.

A version of the grid generator that solved the elliptic equation by using a point-Jacobi iteration method was developed. Since the same equations were solved this grid generator had the same characteristics as the one described in this section. However, it also took up to ten times longer to generate a grid using the iterative scheme.

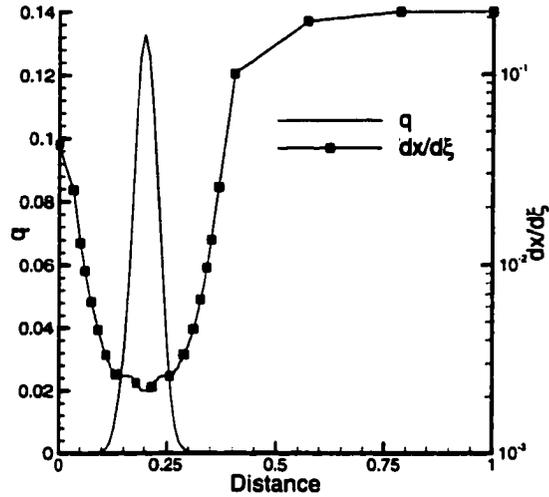


Figure 2.4 Grid spacing produced by the one-dimensional grid generator. q was defined as the Gaussian pulse on an evenly spaced grid, and is shown by the line without symbols. The curve marked with filled boxes shows the grid spacing after the grid points were moved by the one-dimensional grid generator in response to q . The parameters used for this test case were $\lambda = 100$, $I = 101$ and $K = 4$. To prevent obscuring the curve, not all data points are included in the $\frac{dx}{d\xi}$ curve.

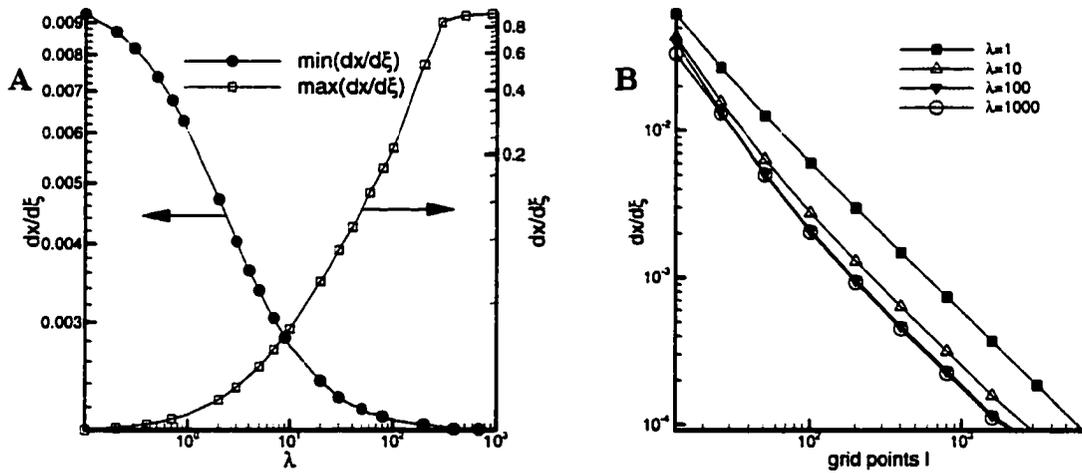


Figure 2.5 Characteristics of the one-dimensional grid. A) Minimum and maximum grid spacing as a function of the λ . Number of grid points $I = 101$, $\lambda = 100$; $K = 4$; B) Minimum grid spacing as a function of number of grid points I and stretching parameter λ . The number of smoothing passes used was $K = 4$.

2.3.4 Two-dimensional grid generation

Since the two-dimensional grid generator was developed using the same ideas as the one-dimensional grid generator it contained the same basic parts. These consisted of establishing a heuristic estimate of the truncation error of the numerical scheme that is then used as a weight function for a minimization problem that resulted in a set of elliptic partial differential equations. Unlike the one-dimensional case this set of partial differential equations was solved using a successive-over-relaxation technique. Like the one-dimensional case the weight function was based on a linear combination of the first and second derivatives of the prognostic variable. The weight function was limited in the two-dimensional case. This limiting was added to the two-dimensional case to prevent excessively large values of the weight functions that appeared when it was used with the three-dimensional code. As in the one-dimensional case the grid and prognostic equation were solved in the transformed domain. Sections 2.3.4.1, 2.3.4.2, 2.3.4.3 and 2.3.4.4 cover the development of the weight function, the derivation of the grid generation equation, the numerical solution of the grid generation equation and some results of the two-dimensional grid generator, respectively.

2.3.4.1 Weight function

Since a heuristic measure of the local numerical error was used as the weight function, there was a great deal of latitude about how to define the weight function. As with the one-dimensional weight function, the first and second derivative of the prognostic variable and grid smoothness were used with the addition of a measure of grid orthogonality for the two-dimensional weight function.

The unscaled weight function \tilde{w} was formed by

$$\tilde{w} = \gamma_1 \left| \frac{\partial q}{\partial x} \right| + \gamma_2 \left| \frac{\partial q}{\partial y} \right| + \gamma_3 \left| \frac{\partial^2 q}{\partial x^2} \right| + \gamma_4 \left| \frac{\partial^2 q}{\partial x \partial y} \right| + \gamma_5 \left| \frac{\partial^2 q}{\partial y^2} \right| \quad (2.68)$$

where the γ_i values were weighting constants. Just as with the one-dimensional grid generator the derivatives in equation 2.68 were expressed in the computational domain coordinates using equations 2.14 and 2.15.

During the development and test of the three-dimensional advection routines that performed the atmospheric tracer transport in conjunction with MM5, it was found that extremely high values of the weight function could occur near the boundaries. It was never determined if the cause resulted from incompatibility of the boundary conditions, the one-sided differencing at the boundaries to determine the weight function, or some other reason. These extremely high weights caused the grid to stretch to this boundary point and ignore the remainder of the domain. Therefore, the weight function was limited

to be no larger than three standard deviations above the average value of the unsmoothed, unscaled weight function. The average value and standard deviation of \bar{w} were calculated as:

$$\bar{w} = \frac{1}{IJ} \sum_{\forall i,j} \bar{w}_{i,j} \text{ where } 1 \leq i, j \leq I, J \text{ respectively} \quad (2.69)$$

$$\sigma_{\bar{w}} = \left(\frac{1}{IJ} \sum_{\forall i,j} (\bar{w}_{i,j} - \bar{w})^2 \right)^{1/2} \text{ where } 1 \leq i, j \leq I, J \text{ respectively} \quad (2.70)$$

The unsmoothed and unscaled weight function was then limited by:

$$\hat{w}_{i,j} = \begin{cases} \bar{w}_{i,j} & \text{if } \bar{w}_{i,j} < \bar{w} + 3\sigma_{\bar{w}} \\ \bar{w} + 3\sigma_{\bar{w}} & \text{if } \bar{w}_{i,j} \geq \bar{w} + 3\sigma_{\bar{w}} \end{cases} \quad (2.71)$$

Once the weight function was limited at every grid point using equations 2.71, the weight function was smoothed as done with the one-dimensional grid generator. The smoothing used repeated operations of:

$$\tilde{w}_{i,j}^{k=0} = \bar{w}_{i,j} \quad (2.72)$$

$$\tilde{w}_{i,j}^k = \frac{1}{8} (4\tilde{w}_{i,j}^{k-1} + \tilde{w}_{i-1,j}^{k-1} + \tilde{w}_{i+1,j}^{k-1} + \tilde{w}_{i,j-1}^{k-1} + \tilde{w}_{i,j+1}^{k-1}) \text{ where } k = 1, K \quad (2.73)$$

$$\tilde{w}_{i,j}^n = \tilde{w}_{i,j}^K \quad (2.74)$$

where the i, j subscripts represented the ξ and η indices respectively, the k superscript represented the iteration level and the n superscript represented the time level. A value of $K = 4$ was used.

After the weight function was smoothed, it was scaled from 0 to 1 by:

$$\hat{w}_{i,j}^n = \frac{\tilde{w}_{i,j}^n - \min_{\forall i,j} (\tilde{w}_{i,j}^n)}{\max_{\forall i,j} (\tilde{w}_{i,j}^n) - \min_{\forall i,j} (\tilde{w}_{i,j}^n)} \quad (2.75)$$

As in the one-dimensional case it is desirable that the weight function be greater than zero. It was also found that often the grid was so sensitive to the largest weightings that it virtually ignored smaller but still significantly weighted regions. Therefore, the weight function was set to:

$$w_{i,j} = 1 + (\hat{w}_{i,j}^n)^m \quad (2.76)$$

where m was set to $\frac{1}{2}$.

2.3.4.2 Grid generation equation

The weight function gives a measure of the amount of stretching that is required as a function of position within the domain. By extremizing a combination of grid smoothness, equidistribution of the

weight function, and the orthogonality of the grid a set of equations has been developed that meet the four requirements listed in section 2.3.1.1. The method used was developed by Brackbill and Saltzman (1982) who extended the work of Winslow (1967) to include a control of the orthogonality of the mesh.

A measure of the variation of the grid spacing over the domain is given by

$$I_s = \int_D [\nabla\xi \cdot \nabla\xi + \nabla\eta \cdot \nabla\eta] dV \quad (2.77)$$

where ∇ is the gradient with respect to the physical coordinates (x, y) . The volume variation of a grid cell is measured by

$$I_v = \int_D w J_2 dV \quad (2.78)$$

where w is a weight function and J_2 is the Jacobian of the transformation matrices as defined in equation 2.13. A measure of the volume weighted orthogonality is given by

$$I_o = \int_D (\nabla\xi \cdot \nabla\eta)^2 J_2^3 dV \quad (2.79)$$

where, according to Thompson et al. (1985) the J_2^3 is "somewhat arbitrary, and caused orthogonality to be emphasized more strongly in the larger cells." The differential volume is given by

$$dV = J_2 d\xi d\eta \quad (2.80)$$

Substitution of 2.80 and equations 2.11 into equations 2.77, 2.78 and 2.79 yields

$$I_s = \int_1^N \int_1^M \frac{1}{J_2} \left[\left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial x}{\partial \eta} \right)^2 + \left(\frac{\partial y}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2 \right] d\xi d\eta \quad (2.81)$$

$$I_v = \int_1^N \int_1^M w J_2^2 d\xi d\eta \quad (2.82)$$

$$I_o = \int_1^N \int_1^M \left(\frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} \right)^2 d\xi d\eta \quad (2.83)$$

A minimization of a linear combination of these three integrals produces a unique grid that has a certain amount of smoothness from equation 2.81, a certain grid spacing as specified by the weight function from 2.82 and a certain degree of orthogonality as given by 2.83. These three minimizations "compete" in the sense that a stretched meshed as specified by the I_v is not the smoothest as specified by I_s and may not be the most orthogonal as specified by I_o . Each integral can be minimized by using the Euler-Lagrange equations as given by Weinstock (1974)

$$\begin{aligned} \left(\frac{\partial}{\partial x} - \frac{\partial}{\partial \xi} \frac{\partial}{\partial x_\xi} - \frac{\partial}{\partial \eta} \frac{\partial}{\partial x_\eta} \right) \bar{I}_l &= 0 \\ \left(\frac{\partial}{\partial y} - \frac{\partial}{\partial \xi} \frac{\partial}{\partial y_\xi} - \frac{\partial}{\partial \eta} \frac{\partial}{\partial y_\eta} \right) \bar{I}_l &= 0 \end{aligned} \quad (2.84)$$

where x_ξ , x_η , y_ξ and y_η represent $\frac{\partial x}{\partial \xi}$, $\frac{\partial x}{\partial \eta}$, $\frac{\partial y}{\partial \xi}$ and $\frac{\partial y}{\partial \eta}$ respectively, \bar{I}_l represents the integrand of the integral I_l and l is either s , v or o .

The indicated operations are performed and the highest derivatives collected. The equations are

$$\begin{aligned} c_{14} \frac{\partial^2 x}{\partial \xi^2} + c_{15} \frac{\partial^2 x}{\partial \xi \partial \eta} + c_{16} \frac{\partial^2 x}{\partial \eta^2} + c_{11} \frac{\partial^2 y}{\partial \xi^2} + c_{12} \frac{\partial^2 y}{\partial \xi \partial \eta} + c_{13} \frac{\partial^2 y}{\partial \eta^2} &= R_{v1} \\ c_{11} \frac{\partial^2 x}{\partial \xi^2} + c_{12} \frac{\partial^2 x}{\partial \xi \partial \eta} + c_{13} \frac{\partial^2 x}{\partial \eta^2} + c_{17} \frac{\partial^2 y}{\partial \xi^2} + c_{18} \frac{\partial^2 y}{\partial \xi \partial \eta} + c_{19} \frac{\partial^2 y}{\partial \eta^2} &= R_{v2} \end{aligned} \quad (2.85)$$

where the coefficients c_{li} are functions of the metric terms and R_{v1} and R_{v2} are source terms from the I_v equation. The I_s and I_o have no source terms associated with them. The l stands for either the s , v or o . The source terms are

$$\begin{aligned} R_{v1} &= -\frac{J_2^2}{2w} \frac{\partial w}{\partial x} = -\frac{J_2}{2w} \left(\frac{\partial w}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial w}{\partial \eta} \frac{\partial y}{\partial \xi} \right) \\ R_{v2} &= -\frac{J_2^2}{2w} \frac{\partial w}{\partial y} = -\frac{J_2}{2w} \left(\frac{\partial w}{\partial \eta} \frac{\partial x}{\partial \xi} - \frac{\partial w}{\partial \xi} \frac{\partial x}{\partial \eta} \right) \end{aligned} \quad (2.86)$$

The derivatives with respect to the physical domain are replaced with derivatives with respect to the computational domain using equations 2.11. The c_s coefficients are

$$\begin{aligned} c_{s1} &= -\frac{1}{J_2^3} \left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \xi} + \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \eta} \right) \left[\left(\frac{\partial x}{\partial \eta} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2 \right] \\ c_{s2} &= \frac{2}{J_2^3} \left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \xi} + \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \eta} \right) \left(\frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} \right) \\ c_{s3} &= -\frac{1}{J_2^3} \left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \xi} + \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \eta} \right) \left[\left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \xi} \right)^2 \right] \\ c_{s4} &= -\frac{2}{J_2^3} \left[\left(\frac{\partial y}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2 \right] \left[\left(\frac{\partial x}{\partial \eta} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2 \right] \\ c_{s5} &= -\frac{2}{J_2^3} \left[\left(\frac{\partial y}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2 \right] \left(\frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} \right) \\ c_{s6} &= \frac{1}{J_2^3} \left[\left(\frac{\partial y}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2 \right] \left[\left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \xi} \right)^2 \right] \\ c_{s7} &= \frac{1}{J_2^3} \left[\left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial x}{\partial \eta} \right)^2 \right] \left[\left(\frac{\partial x}{\partial \eta} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2 \right] \\ c_{s8} &= -\frac{2}{J_2^3} \left[\left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial x}{\partial \eta} \right)^2 \right] \left(\frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} \right) \\ c_{s9} &= \frac{1}{J_2^3} \left[\left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial x}{\partial \eta} \right)^2 \right] \left[\left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \xi} \right)^2 \right] \end{aligned} \quad (2.87)$$

The c_v and c_o coefficients are

$$\begin{aligned}
c_{v1} &= -\frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \eta} & c_{v2} &= \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \eta} & c_{v3} &= -\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \xi} \\
c_{v4} &= \left(\frac{\partial y}{\partial \eta}\right)^2 & c_{v5} &= -2 \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} & c_{v6} &= \left(\frac{\partial y}{\partial \xi}\right)^2 \\
c_{v7} &= \left(\frac{\partial x}{\partial \eta}\right)^2 & c_{v8} &= -2 \frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} & c_{v9} &= \left(\frac{\partial x}{\partial \xi}\right)^2 \\
c_{o1} &= \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \eta} & c_{o2} &= \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \eta} & c_{o3} &= \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \xi} \\
c_{o4} &= \left(\frac{\partial x}{\partial \eta}\right)^2 & c_{o5} &= 2 \left(2 \frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} \right) & c_{o6} &= \left(\frac{\partial x}{\partial \xi}\right)^2 \\
c_{o7} &= \left(\frac{\partial y}{\partial \eta}\right)^2 & c_{o8} &= 2 \left(\frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + 2 \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} \right) & c_{o9} &= \left(\frac{\partial y}{\partial \xi}\right)^2
\end{aligned} \tag{2.88}$$

A linear combination of the three set of equations was chosen as

$$\begin{aligned}
c_4 \frac{\partial^2 x}{\partial \xi^2} + c_5 \frac{\partial^2 x}{\partial \xi \partial \eta} + c_6 \frac{\partial^2 x}{\partial \eta^2} + c_1 \frac{\partial^2 y}{\partial \xi^2} + c_2 \frac{\partial^2 y}{\partial \xi \partial \eta} + c_3 \frac{\partial^2 y}{\partial \eta^2} &= \lambda_v R_{v1} \\
c_1 \frac{\partial^2 x}{\partial \xi^2} + c_2 \frac{\partial^2 x}{\partial \xi \partial \eta} + c_3 \frac{\partial^2 x}{\partial \eta^2} + c_7 \frac{\partial^2 y}{\partial \xi^2} + c_8 \frac{\partial^2 y}{\partial \xi \partial \eta} + c_9 \frac{\partial^2 y}{\partial \eta^2} &= \lambda_o R_{o2}
\end{aligned} \tag{2.89}$$

where

$$c_i = c_{,i} + \lambda_v c_{vi} + \lambda_o c_{oi} \tag{2.90}$$

The constants λ_v and λ_o are positive constants that control the amount of stretching and orthogonality that is imposed. Equations 2.89 are a coupled set of non-linear Poisson equations that can be solved to determine the new grid. The next section discusses the method that was used for solving these equations.

2.3.4.3 Numerical solution of the grid generation equation

The system of equations that needs to be solved is a block penta-diagonal system where each block is a two-by-two matrix that represents the x and y location values for a $(\xi_{i,j}, \eta_{i,j})$ location. The entire system was solved using successive over-relaxation with each individual two-by-two block being solved using Cramer's rule.

The coefficients of the first and second derivatives were represented by second order central differences except the first derivatives at the edges which were represented using one-sided second-order differences. Equations 2.89 are written in finite difference form

$$\begin{bmatrix} (c_4 + c_6) & (c_1 + c_3) \\ (c_1 + c_3) & (c_7 + c_9) \end{bmatrix} \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} R_{x1} + R_{y1} - \lambda_v R_{v1} \\ R_{x2} + R_{y2} - \lambda_o R_{o2} \end{bmatrix} \tag{2.91}$$

where

$$\begin{aligned}
R_{x1} &= c_4 (x_{i-1,j} + x_{i+1,j}) + c_5 (x_{i+1,j+1} + x_{i-1,j-1} - x_{i+1,j-1} - x_{i-1,j+1}) + c_6 (x_{i,j-1} + x_{i,j+1}) \\
R_{y1} &= c_1 (y_{i-1,j} + y_{i+1,j}) + c_2 (y_{i+1,j+1} + y_{i-1,j-1} - y_{i+1,j-1} - y_{i-1,j+1}) + c_6 (y_{i,j-1} + y_{i,j+1}) \\
R_{x2} &= c_1 (x_{i-1,j} + x_{i+1,j}) + c_2 (x_{i+1,j+1} + x_{i-1,j-1} - x_{i+1,j-1} - x_{i-1,j+1}) + c_3 (x_{i,j-1} + x_{i,j+1}) \\
R_{y1} &= c_7 (y_{i-1,j} + y_{i+1,j}) + c_8 (y_{i+1,j+1} + y_{i-1,j-1} - y_{i+1,j-1} - y_{i-1,j+1}) + c_9 (y_{i,j-1} + y_{i,j+1})
\end{aligned} \tag{2.92}$$

This two-by-two linear system shown in equation 2.91 was solved using Cramer's rule yielding an estimate of the new physical grid point locations, \tilde{x} and \tilde{y} . The system was then over-relaxed to increase the convergence of the solution

$$\begin{aligned}
x^{k+1} &= (1 - \omega) x^k + \omega \tilde{x} \\
y^{k+1} &= (1 - \omega) y^k + \omega \tilde{y}
\end{aligned} \tag{2.93}$$

where ω is an over-relaxation parameter between 1 and 2 and the superscript k represents the iteration level.

It is important to remember that this entire iterative process is repeated at each MPDATA time step in order to generate a new grid. Therefore, the value of q is temporally invariant in both the physical and computational spaces during this process. However, since the weights are a function of the physical coordinates (x, y) , which change with each iteration, the value of the weights also changes with each iteration. In order to account for this change the weight function is updated after each iteration by

$$w^{k+1} = w^k + \frac{1}{J_2} \left[\left(\frac{\partial w}{\partial \xi} \frac{\partial y^k}{\partial \eta} - \frac{\partial w}{\partial \eta} \frac{\partial y^k}{\partial \xi} \right) (x^{k+1} - x^k) + \left(\frac{\partial w}{\partial \eta} \frac{\partial x^k}{\partial \xi} - \frac{\partial w}{\partial \xi} \frac{\partial x^k}{\partial \eta} \right) (y^{k+1} - y^k) \right] \tag{2.94}$$

where equations 2.11 are used to interchange the independent and dependent variables of the metric terms. Note that q^n is not known on the grid as it moves and therefore equation 2.68 could not be used to determine the value of w as the x and y coordinates change.

Since the boundary of the domain is forced to remain static, grid points could not move perpendicular to them. However, they were allowed to move parallel to the boundary. The derivative normal to the boundary was made zero for all of the boundaries after the solution of the grid generation equations had converged.

It was desirable to establish a convergence criterion that was independent of the number of grid points. If convergence criterion is not a function of the total number of grid points, as the number of grid points increases less stretching would satisfy the condition. Therefore, the convergence criterion for the grid was set at 2% of the unstretched grid cell width.

2.3.4.4 Characteristics of the two-dimensional grid generator

A number of tests were conducted to verify that the two-dimensional grid generation had the desired properties. The initial condition for the rotating cone problem was used. This consisted of a right circular cone with a radius of 0.15, a height of 4.0, centered at $x = 0.5$ and $y = 0.75$ in a 1×1 domain. The background field was specified as zero. The test consisted of varying the number of grid points in each direction, the stretching and orthogonality parameters (λ_v and λ_o), the number of smoothing passes K , and the over-relaxation parameter ω .

Tests were conducted with different resolutions in the ξ and η directions. It was found that in some cases very poor grids were produced as depicted in figure 2.6. This was caused by boundary grid points being able to move along the boundary and cluster thus producing cells with large aspect ratios. This problem was solved by specifying the boundary point weights proportional to the maximum weight function before smoothing. The value that was used was 80%. The smoothing spread the influence of this weight to the interior and the boundary point areas were kept small. The resulting grid from this is shown in figure 2.7. In addition to solving this problem it provided a more fundamental need. When inflow boundary conditions exist, it is necessary to anticipate that features of the prognostic variable may be introduced at the boundary. By clustering the grid points at the boundary this anticipation is provided for. Another solution to this problem would have been to include a term in equation 2.89 that would minimize the aspect ratio. The option was not implemented because adverse grid aspect ratios were not observed at other locations, significant computational expense is required to add other grid adaptation criterion, and boundary clustering would have been needed regardless.

Figure 2.8 shows the effect of varying the stretching parameter, λ_v . As expected the ratio

$$r_a = \frac{\text{maximum cell area}}{\text{minimum cell area}} \quad (2.95)$$

increased with increasing λ_v . Additionally, the number of iterations to solve for the grid increased as well.

If λ_o is varied, as shown in figure 2.9, a much larger value must be used to effect the grid in any appreciable way. The number of iterations increased as λ_o increased. As expected the maximum cosine of the angle between lines of constant x and y decreased as λ_o increased. λ_o had little effect on the grid cell size ratio r_a .

Figure 2.10 illustrates the behavior of the grid generator as the number of grid points is varied. As expected the number of iterations to converge to a solution increases as the number of grid points increases. The general trend that the grid area aspect ratio would increase is shown. The reason for

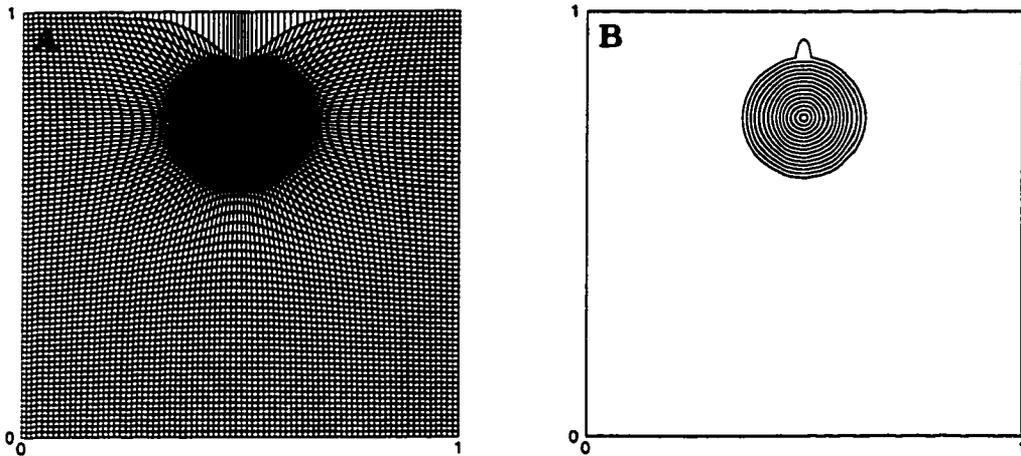


Figure 2.6 Poor quality grid two-dimensional grid. 80×80 grid points with $\lambda_v = 4.0$, $\lambda_o = 0.0$, and $K = 4$ smoothing passes. A) The grid produced with no boundary clustering; B) The profile after it has been reinitialized onto the grid in panel A. This would serve as an initial condition for the rotating cone problem.

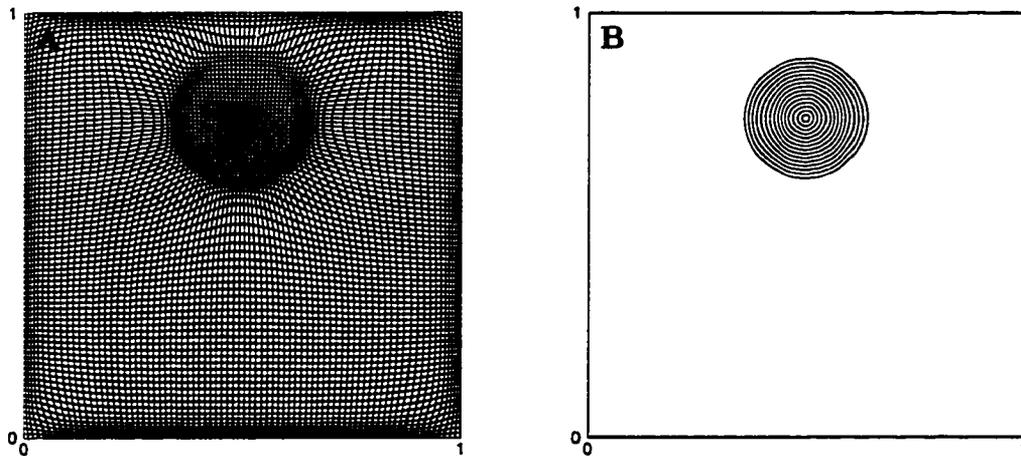


Figure 2.7 Improved quality two-dimensional grid. 80×80 grid points with $\lambda_v = 4.0$, $\lambda_o = 0.0$, and $K = 4$ smoothing passes. A) The grid produced with boundary clustering; B) The profile after it has been reinitialized onto the grid in panel A. This would serve as an initial condition for the rotating cone problem.

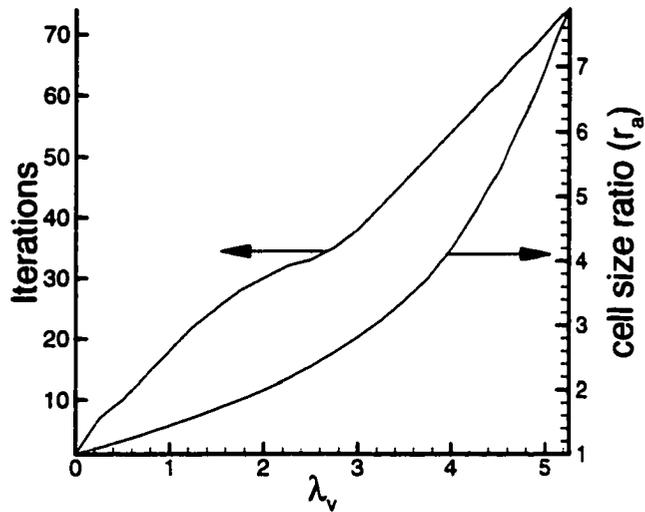


Figure 2.8 The effect of varying the stretching parameter λ_v on a two-dimensional grid. Number of iterations and the cell area ratio r_a . 41×41 grid points with $\lambda_o = 0.0$, and $K = 4$ smoothing passes

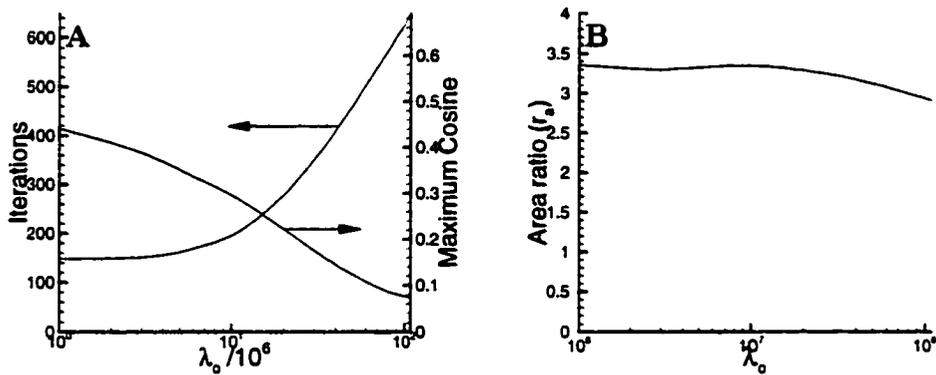


Figure 2.9 Effects of varying λ_o on a two-dimensional grid. 41×41 grid points with $\lambda_v = 4.0$, and 4 smoothing passes. A) Iterations and orthogonality measure; B) Cell area ratio (r_a)

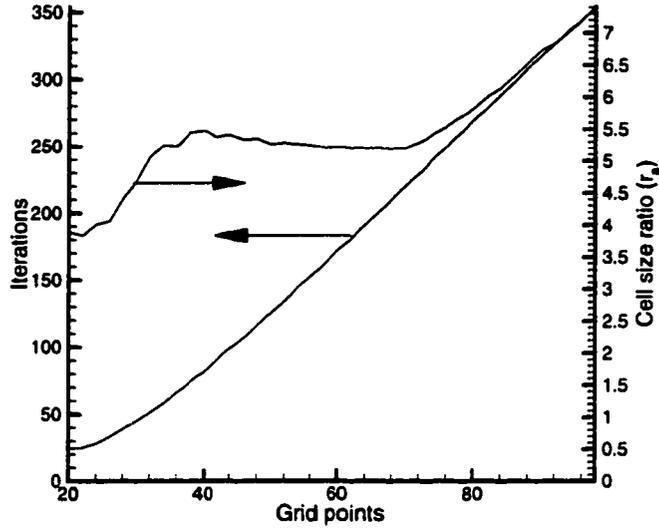


Figure 2.10 Number of iterations and the cell area aspect ratio. $\lambda_v = 4.0$, $\lambda_o = 0.0$, and $K = 4$ smoothing passes

the plateau in the area ratio from about 40 to about 70 grid points is not known.

Figure 2.11 shows the effect on the number of iterations of varying the over-relaxation parameter ω which was used to increase the rate the relaxation scheme converged. Above $\omega = 1.75$ the grid generator became unstable.

The γ values in equation 2.68 were all set to one except for γ_4 which was set to zero. It was found that the cross derivative term caused the weight function to be dependent on the orientation of the grid. Therefore, γ_4 was set to zero eliminating this cross derivative term. It would not be desirable to make $\gamma_1 \neq \gamma_2$ or $\gamma_3 \neq \gamma_5$ because this would cause the weight function to become dependent on direction. Although all the γ 's were set to one with the exception of $\gamma_4 = 0$, greater influence from the prognostic equation could be included in the weight function by considering the spatial scale of the phenomena that the user wants to resolve. It may have been more correct to define

$$\tilde{w} = \frac{1}{L} \left(\gamma_1 \left| \frac{\partial q}{\partial x} \right| + \gamma_2 \left| \frac{\partial q}{\partial y} \right| \right) + \gamma_3 \left| \frac{\partial^2 q}{\partial x^2} \right| + \gamma_5 \left| \frac{\partial^2 q}{\partial y^2} \right| \quad (2.96)$$

where L is a length scale. First of all it would make the contributions of the first and second derivatives dimensionally equivalent. It was found in all the applications of the one- and two-dimensional grid generators that the effect of the second derivative dominated that of the first. Recall that the spatial domains were always scaled to be on the order of one. By ignoring the effect of L , the largest possible scales were implicitly chosen. Since L would always be less than one it would have the effect of increasing the emphasis on the first derivative, balancing the contributions of the first and second

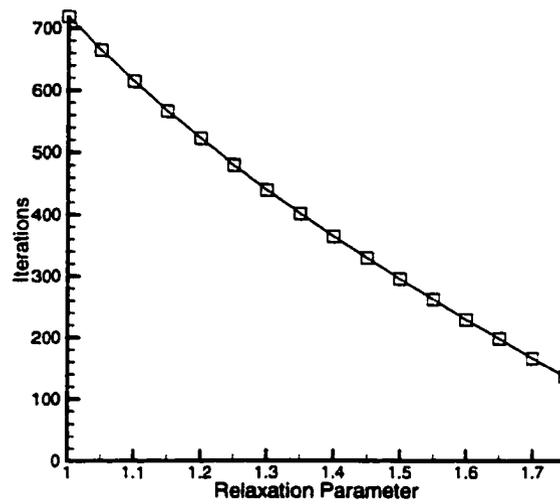


Figure 2.11 Number of iterations for a grid with 41×41 grid points. $\lambda_v = 4.0$, $\lambda_o = 0.0$, and 4 smoothing passes.

derivatives. Investigating this theory is left for future research.

2.4 Description of the one and two-dimensional models and their results

This section describes the one- and two-dimensional models that were used to investigate the performance of the MPDATA scheme with a moving grid. Both models performed passive advection with a time independent velocity field. A Gaussian pulse was transported by a constant velocity field in the one-dimensional case. The rotating cone problem was solved in the two-dimensional case. Since analytical solutions to both of these problems are known, error measures could be determined.

As described earlier, the explicit nature of the MPDATA scheme and the lack of temporal control of the grid could create a situation where the grid point velocity could be so high that the CFL condition would be violated. For the one-dimensional case, a constraint on the grid point movement was developed that would prevent a grid point from moving too far during a single time step. This constraint gave new insight into how and why dynamic adaptive grids work and why they are useful.

Section 2.4.1 describes the one-dimensional model. The section has three subsections that describe the model, the limiting of the grid point movement, and the results of several numerical experiments.

Section 2.4.2 describes the two-dimensional model and results of several numerical experiments. The model is described in section 2.4.2.1. Qualitative as well as quantitative evaluation of the experiments

are presented in section 2.4.2.2.

2.4.1 One-dimensional model

The one-dimensional model was used to develop the necessary experience with the DAG technique and MPDATA before endeavoring to write a multidimensional version. Since it required much less time to perform a calculation than the multidimensional codes, it was used to determine the impact of varying the number of grid points on the static and dynamic MPDATA schemes. Section 2.4.1.1 briefly describes the model. Section 2.4.1.2 explains the limiting of the grid point movement and how the DAG technique includes a Lagrangian aspect to the computation. Results of several numerical experiments are presented in section 2.4.1.3.

2.4.1.1 One-dimensional model description

The one-dimensional advection equation

$$\frac{\partial q}{\partial t} + u \frac{\partial q}{\partial x} = 0 \quad (2.97)$$

was solved on a domain $0 < x < 1$ using a scaled Gaussian pulse of the form

$$q(x) = \frac{1}{3\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-0.2}{0.03}\right)^2} \quad (2.98)$$

as an initial condition. Figure 2.12 shows the initial condition. The velocity field was set to a constant, $u = 1$. The problem was solved up to $t = 0.5$.

Since the MPDATA scheme uses the donor-cell scheme, a staggered grid was used. Figure 2.13 shows a typical grid with only a few points. The \times symbols represent cell centers, while the \bullet symbols represent the cell boundaries. In order to perform the grid stretching, the locations of the cell boundaries were used as the physical points in the grid generator rather than the cell centers. This choice had several advantages. First, since MPDATA requires the velocity values at cell boundaries, the speed of the cell boundaries was required and not the speed of the cell centers. Second, the coordinates of the cell centers were simply the average of the cell boundary coordinates. The first and the last cell boundaries were set to 0 and 1 respectively and served as boundary conditions for the elliptic grid solver. Ghost cell centers were placed on both ends to provide locations for boundary values that could be transported into the domain. These locations were calculated by reflecting the adjacent cell center about the end cell boundary.

The initial condition was created by repeated evaluation of equation 2.98 and implementation of the one-dimensional grid generator as described in section 2.3.3. An evenly spaced grid was initially used

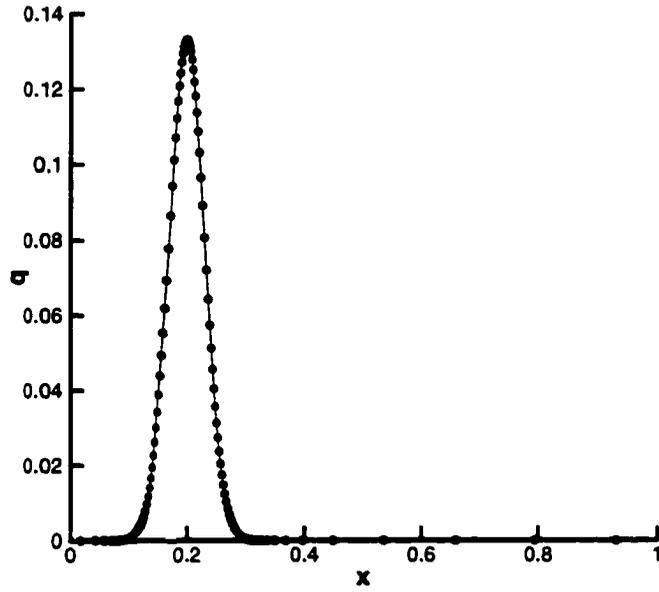


Figure 2.12 Initial condition. Dots represent locations of grid points.

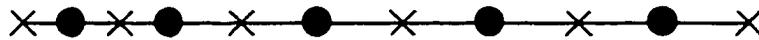


Figure 2.13 Example one-dimensional grid. \times 's represent cell centers, \bullet 's represent cell boundaries.

to start the process. The value of q at these grid points were then determined from equation 2.98. The grid generator then moved the grid points. The new value of $q(x)$ was determined from equation 2.98. The process was then repeated starting with the grid generation. Ten iterations were used to determine the initial grid.

Once the initial grid was calculated, the time step was calculated. The time step was held constant throughout the integration. It was determined as

$$\Delta t = \frac{s}{u} \min_{v_i} \left(\frac{dx}{d\xi} \Big|_i^{n=0} \right) \quad (2.99)$$

where the metric term $\frac{dx}{d\xi}$ was approximated by a central difference expression and the s was a safety factor such that $0 < s \leq 1$.

Each time step consisted of 11 steps. These steps were:

1. An iteration counter k was set to zero.
2. The grid was advanced using the one-dimensional grid generator and the solution at time level n .
3. The grid point velocity was determined at the cell boundaries by using a central difference around the point $n + 1/2$ as:

$$\frac{\partial x}{\partial \tau} \Big|_{i+1/2}^{n+1/2} = \frac{x_{i+1/2}^{n+1} - x_{i+1/2}^n}{\Delta t} \quad (2.100)$$

4. The contravariant velocity from equation 2.36 was approximated as:

$$\hat{u}_{i+1/2}^{n+1/2} = \left(u - \frac{\partial x}{\partial \tau} \Big|_{i+1/2}^{n+1/2} \right) \frac{\partial \xi}{\partial x} \Big|_{i+1/2}^{n+1} \quad (2.101)$$

The contravariant velocity was non-dimensionalized using the grid spacing $\Delta \xi = 1$ and the time step Δt to form the Courant number \hat{U} .

5. The CFL condition was checked. If the Courant number violated the CFL condition, the movement of the grid points was limited using the procedure described later in section 2.4.1.2.

Steps 4 through 5 where repeated until the CFL condition was met. This iterative process was needed since the metric term $\frac{\partial x}{\partial \xi}$ in equation 2.101 was taken at the $n + 1$ time level.

6. Since the donor-cell scheme requires equation 2.97 to be in flux form, a divergence term was needed as described in section 2.2.2. It was approximated as

$$q_i \frac{dU_i}{dx} = q_i^n \left(\hat{U}_{i+1/2}^{n+1/2} - \hat{U}_{i-1/2}^{n+1/2} \right) \quad (2.102)$$

where \hat{U} is the Courant number in the computational domain.

7. The q^{k+1} was calculated using equation 2.52. This step was the first step of the MPDATA iteration loop. If the total number of iterations was complete ($k = K$), q was assigned the value q^{k+1} and the time step was complete.
8. If the RPV option was not used, the non-dimensional pseudo velocity, $\hat{U}^{(k+1)}$, was calculated using equation 2.30. $\hat{U}^{(0)}$ was the Courant number. If the RPV option was used, $\hat{U}^{(k+1)}$ was calculated from equation A.30 in appendix A where the velocity components not in the x -direction were set to zero.
9. If the "third order" option was specified, the extra terms were calculated as described in section A.1.
10. The divergence term was set to zero after the first iteration. Recall from section 2.2.2.2 that the correction term for the divergence term and another correction term identically canceled.
11. The iteration counter k was advanced by one and control was returned to number 7.

The error in the numerical solution was calculated through the use of quadrature. The L_2 norm was defined as

$$E_{L_2} = \left(\frac{1}{L} \int_0^L (\hat{q}(x) - q(x))^2 dx \right)^{1/2} \quad (2.103)$$

where $q(x)$ is the numerical solution and $\hat{q}(x)$ is the exact analytical solution. The trapezoidal rule was used to compute the approximate error.

2.4.1.2 Limiting of the grid point movement

The stability problem created by the lack of temporal control of the grid and the explicit nature of the MPDATA scheme has been mentioned in several sections. In the two- and three-dimensional models, it was only a theoretical difficulty because the grid spacing was large enough that it was not a problem. However, it was a practical problem in the one-dimensional model, because many grid points could be used and thus $\frac{\partial x}{\partial \xi}$ term could become very small.

Much of the literature presents the DAG technique simply in terms of clustering points in critical areas and then allowing movement of the grid so those areas remain highly resolved. While this view is acceptable it is incomplete. Yanenko et al. (1979) includes a Lagrangian transport term in the grid stretching criteria and Thompson et al. (1985) refer to grid points as "observers" that follow the flow in the same manner that the initial illustration in chapter 1 referred to observers of a tornado. Therefore,

others have had the conceptual idea that the DAG technique is a blend between Eulerian and Lagrangian analysis. The following analysis explicitly presents the Lagrangian aspect of the DAG technique.

Through examination of equation 2.101 the CFL condition was

$$-1 \leq \frac{\Delta\tau}{\Delta\xi} \left(u - \frac{\partial x}{\partial\tau} \right) \frac{\partial\xi}{\partial x} \leq 1 \quad (2.104)$$

Substituting the central difference expression in equation 2.100 for $\frac{\partial x}{\partial\tau}$ and noting that $\Delta\xi = 1$

$$-1 \leq (\Delta\tau u - x^{n+1} + x^n) \frac{\partial\xi}{\partial x} \leq 1 \quad (2.105)$$

Solving for x^{n+1} yields an expression for locations of x^{n+1} that will yield a stable condition for an explicit scheme.

$$(x^n + \Delta\tau u) - \frac{\partial x}{\partial\xi} \leq x^{n+1} \leq (x^n + \Delta\tau u) + \frac{\partial x}{\partial\xi} \quad (2.106)$$

This expression illustrates that there is a region in which a grid point can move. The bracketed expression is the first order Lagrangian transport of the grid point by the fluid. Had a higher order finite difference approximation for $\frac{\partial x}{\partial\tau}$ been used this bracketed expression would be higher order. The width of the region is determined by the metric term $\frac{\partial x}{\partial\xi}$. If the grid point is not allowed to move, the expression reverts back to the typical expression for the CFL condition

$$-1 \leq \Delta\tau \frac{\partial\xi}{\partial x} u \leq 1 \quad (2.107)$$

where an approximation to $\frac{\partial\xi}{\partial x}$ is $\frac{1}{\Delta x}$.

Figure 2.14 illustrates the zone of influence for a static grid and one where the grid point is advected with the flow. If the distance a fluid parcel travels over a time step is outside this shaded region the CFL condition is violated and an explicit scheme, like MPDATA, will become unstable. Note that in panel B of figure 2.14 that this advection distance is not equal for flow directions in the negative and positive directions. If the fluid motion is in the same direction as the grid point motion (to the right in figure 2.14), larger time steps, corresponding to larger Courant numbers can be taken because the zone of influence is tilted in that direction. This distance is indicated as x_{ds} in figure 2.14. However, if the grid point motion opposes the flow direction (to the left figure 2.14), smaller time steps are required, since the upper corner of the zone tilts away from the flow direction. This distance is indicated by x_{us} . If the grid point motion was completely determined by the flow advection, infinitely large time steps could be taken because the value of q at the grid point would be constant. However, since the grid is constricted to conform to the boundaries and fill the whole domain, not all the points can travel with the flow.

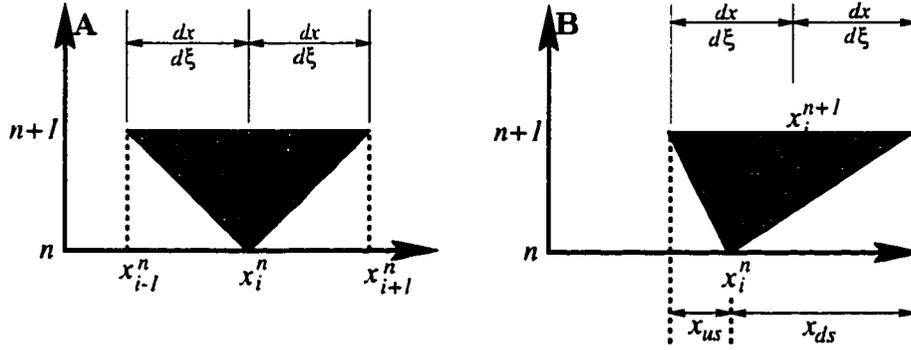


Figure 2.14 A) Zone of influence for a static grid; B) Zone of influence for a dynamic grid; If the advected distance $u\Delta t$ remains within the shaded cone the scheme is stable. In panel B, the x_{us} is this advection distance if the $u < 0$ and x_{ds} is the distance if $u > 0$.

To understand why all points can not travel with the flow, it is helpful to think through a probable path of a grid point to determine when and why a grid point would move upstream. Given a point that is in a region that has not been affected by grid stretching, it is plausible that as a feature with steep gradients and sharp curvature approaches, the point will move upstream toward the phenomenon. As the grid point approaches the feature, it slows and eventually reverses direction and follows the feature, never attaining the same speed as the fluid that is advecting the tracer. It will follow the feature for a distance, passing its original undisturbed location. Gradually, the feature will pass the grid point since the advection speed is greater than the grid speed. Eventually, the smoothness criteria and the boundary conditions will force the point to reverse direction and once again travel upstream, until it has relaxed back to its original undisturbed position.

This scenario illustrates that grid points will travel upstream, but they do so only as an important feature approaches and as it out distances the point. While the feature needs high resolution, the grid point travels along with the feature. Therefore, a grid point has a Lagrangian component during this period. This Lagrangian aspect is one reason why the DAG technique is successful.

However, when upstream motion of grid points is combined with close grid spacing the CFL condition is very restrictive. It was found that when the number of grid points in the one-dimensional model was less than 250, $\frac{\partial x}{\partial \xi}$ was large enough that with a reasonable time step the scheme was always stable. However as the number of grid points increased $\frac{\partial x}{\partial \xi}$ became so small that almost any grid point movement caused a violation of the CFL condition. Two different techniques that used equation 2.106 to ensure stability were attempted. The first technique limited the amount the grid points could move based on the most severe violation the CFL condition. The second technique let the grid points move as directed

by the grid generator and adjusted the time step so that the scheme would be stable. The first technique was successful; the second was not.

With the first technique, when the CFL condition was violated, at any point in the domain, the movement of all the grid points was limited by the worst case scenario. This was done by calculating the maximum allowable movement of each grid point and identifying which exceeded this allowable movement by the greatest percentage. At each grid point, the fraction of the original movement allowed was calculated as

$$\phi_i = \min\left(\frac{\Delta x_{M_i}}{\Delta x_{A_i}}, 1\right) \quad (2.108)$$

where Δx_{M_i} is the maximum allowable distance so the CFL is not violated at point i and Δx_{A_i} is the unlimited distance at point i . In order to ensure that no grid points were allowed to cross, all of the grid point movements were limited by the smallest ϕ_i .

$$\Delta x_i = \Delta x_{A_i} \min(\phi_i) \quad (2.109)$$

Often a couple of iterations of the procedure were required because $\frac{\partial \xi}{\partial x}$ was calculated at time level $n + 1$. Therefore, the limiting would separate the two points that caused the violation but also cause two other points to become so close that their movement needed to be limited.

The second method attempted to adjust the time step size in order to meet the CFL criteria simultaneously at all the grid points. The inequalities in equation 2.106 can be rearranged to yield a minimum and maximum allowable time step for stability. If $u - \frac{\partial x}{\partial \tau} < 0$ the minimum time step is a positive number. These minimum and maximum time steps were computed for each grid point. If the largest minimum time step was less than the shortest maximum time step, the technique worked. However, in practice, situations occurred where the minimum time step of one point was larger than the maximum time step of another point and there was no way to satisfy equation 2.106 for all grid points simultaneously.

Adding a means of temporal control by using either a grid speed method or a attraction and repulsion type of grid generator, as described in sections 2.3.1.1.5 and 2.3.1.1.6, was another option to solve this problem. Recall that these schemes solve for the grid speeds directly and then use quadrature to determine the location of the grid points. Thus, as the time step becomes small so does the grid movement. Since these options may also be more efficient than the two-dimensional grid generator, one of them may be a much better choice than the variational approach that was used. Investigation into their feasibility is left as future work.

2.4.1.3 One-dimensional experimental results

A number of numerical experiments were run to determine the effects of the dynamic grid adaptation, the various MPDATA options, the limiting of grid point movements, and the computational costs. The following sections discuss these topics.

2.4.1.3.1 Dynamic grid adaptation: The \log_2 of the the RMS error was used to evaluate the accuracy of the different schemes. As the number of grid points increased by multiples of two, this measure of the error drops by the order of the scheme. This can be seen by considering the error of two computations, E_1 and E_2 , by some scheme of order p , where E_1 is the error obtained when using uniform grid spacing $2h$ and E_2 is the error obtained when using uniform grid spacing h . Therefore

$$E_1 \sim (2h)^p \quad (2.110)$$

$$E_2 \sim h^p$$

By taking the \log_2 of each and subtracting the second from the first it is seen the difference is the order of the scheme:

$$\log_2 E_1 - \log_2 E_2 = p \log_2 2 + p \log_2 h - p \log_2 h = p \quad (2.111)$$

Initially, the code was run to duplicate the results of Margolin and Smolarkiewicz (1989). This duration of this test case was only 0.02 in non-dimensional time. This corresponded to one time step for the coarsest domain. In order to gain a better understanding of the impact of the dynamic adaptive grid it was decided to extend the duration of the test to $t = 0.5$. Figure 2.15 shows the results for different MPDATA options using a static and uniform grid. Notice that the spacing between the lines in panels A, B, C, and E is two. This indicates that the scheme is second-order accurate, since the change in the \log_2 of the the error is indicative of the schemes order. Similarly, the two panels (D and F) with the “third-order” correction terms, show third-order accuracy. Recall that the “third order” terms made the scheme third order accurate as long as the velocity field is constant as it was in this case. The effect of additional iterations and the recursive pseudo velocities decreases the error only slightly. For a static, uniform grid the safety factor s from equation 2.99 is equivalent to the Courant number.

The differences between the lines for panels B and D of figure 2.15 are shown in figure 2.16. Panel A in this figure is representative of the second-order schemes in figure 2.15 and panel B is representative of the “third-order” schemes. Note that the order of accuracy is not initially apparent but converges as the number of grid intervals is repeated doubled.

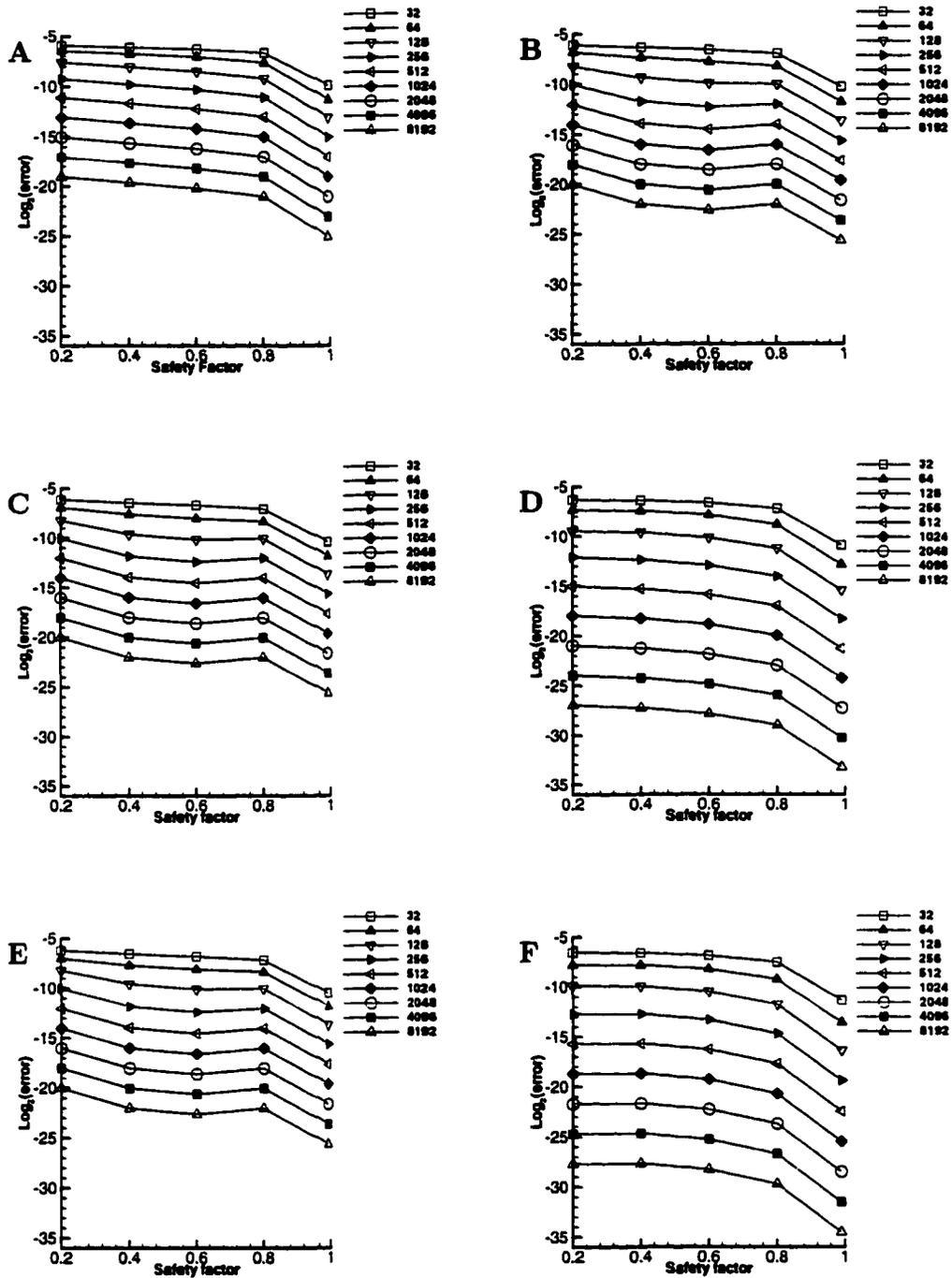


Figure 2.15 L_2 error versus safety factor for static uniform grids A) 2 iterations no 3rd order terms; B) 3 iterations no 3rd order terms; C) 4 iterations no 3rd order terms; D) 3 iterations with 3rd order terms; E) Recursive pseudo velocities no 3rd order terms; F) Recursive pseudo velocities with 3rd order terms; Legend indicates the number of grid points.

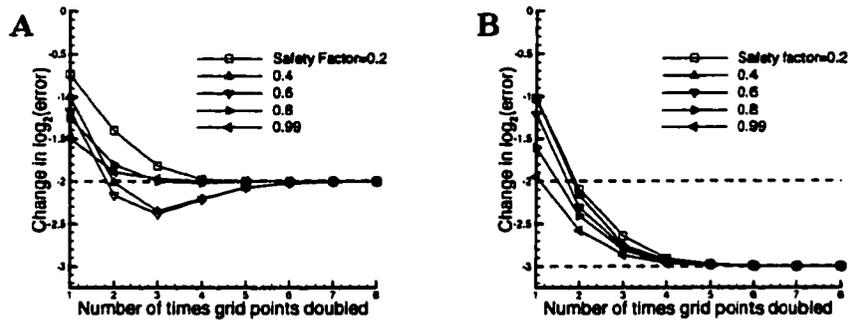


Figure 2.16 Order of accuracy of schemes without and with third-order terms. Lines represent distances between lines depicted in figure 2.15. A) Corresponds to panel B which used 2 iterations and no 3rd order terms B) Corresponds to panel D which used 3 iterations and 3rd order terms.

Figure 2.17 shows the results of the same experiments with a dynamic adaptive grid. Note unlike the static and uniform grid that the safety factor is not equal to the Courant number. The Courant number for a moving grid is a function of the grid speed, and is therefore variable in both space and time for a dynamic grid. Note that the smallest number of grid points used in this case was half the number used in the static grid case and the errors were still less than the static grid case. When small numbers of grid points are used the scheme acts as if it is third or even fourth order accurate. However, the dramatic decrease of error does not continue as more grid points are used nor as the safety factor increased. The errors may actually increase as the number of grid points and/or the safety factor increases. This was due to the need to limit the movement of the grid points in order to keep the method stable. Thus the grid was unable to cluster and follow the phenomena adequately. However, it is more advantageous to have a scheme that has greater accuracy with a small number of grid points than one that requires many points to achieve highly accurate results. The flatness of the curves in figure 2.17 is believed to be due to the movement of the grid points. As the grid points move with the Gaussian pulse the Courant number variation as a function of safety factor was quite small.

2.4.1.3.2 The number of smoothing passes and limitation of the grid point movement:

Figure 2.18, panel A depicts effects on the error of the DAG scheme of the number of smoothing passes of the weight function used in the grid generation procedure. The computation that produced these data used the RPV option without any “third-order” order terms. These results are representative

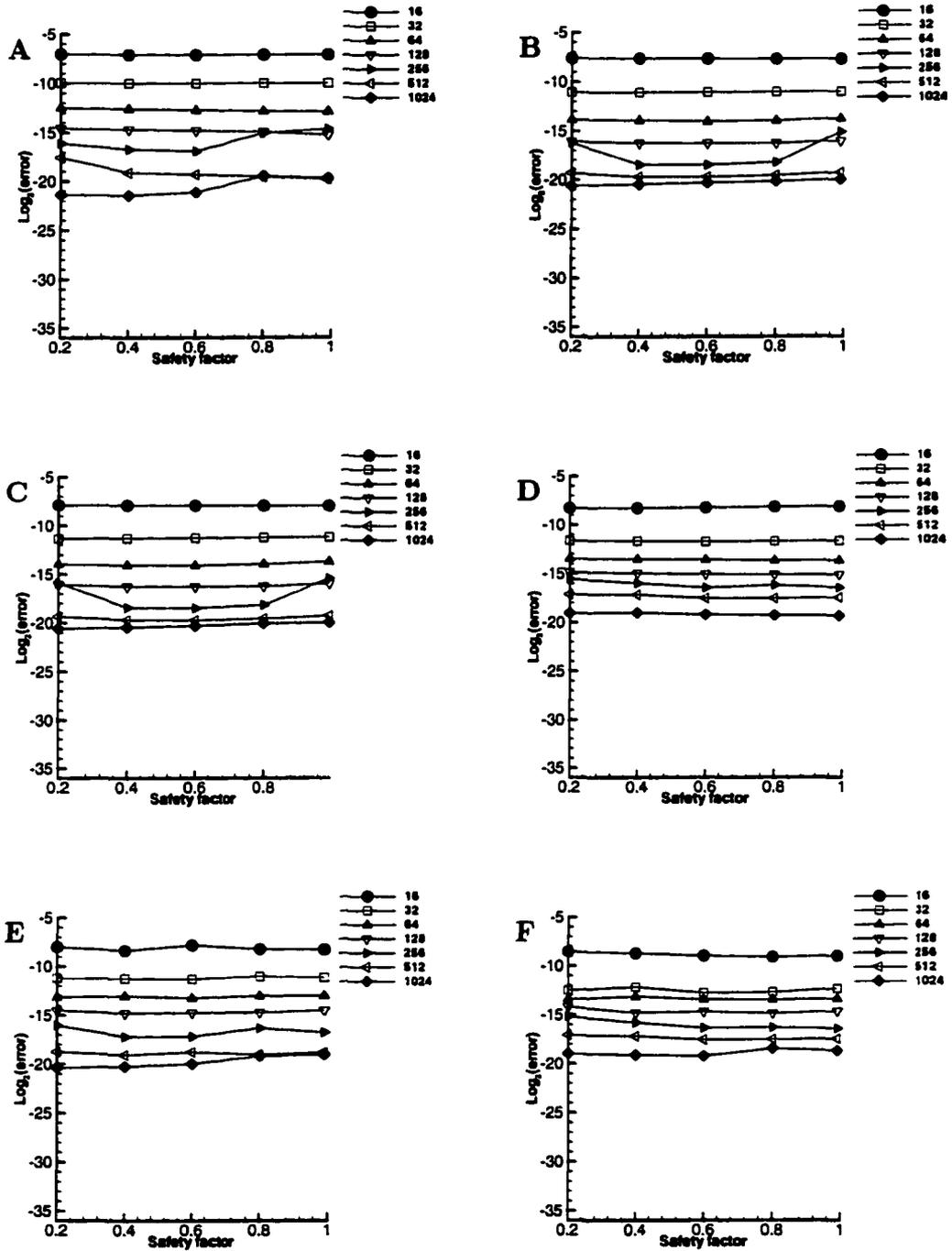


Figure 2.17 L_2 error versus safety factor for dynamic grid A) 2 iterations no 3rd order terms; B) 3 iterations no 3rd order terms; C) 4 iterations no 3rd order terms; D) 3 iterations with 3rd order terms; E) Recursive pseudo velocities no 3rd order terms; F) Recursive pseudo velocities with 3rd order terms; The legend indicates the number of grid points.

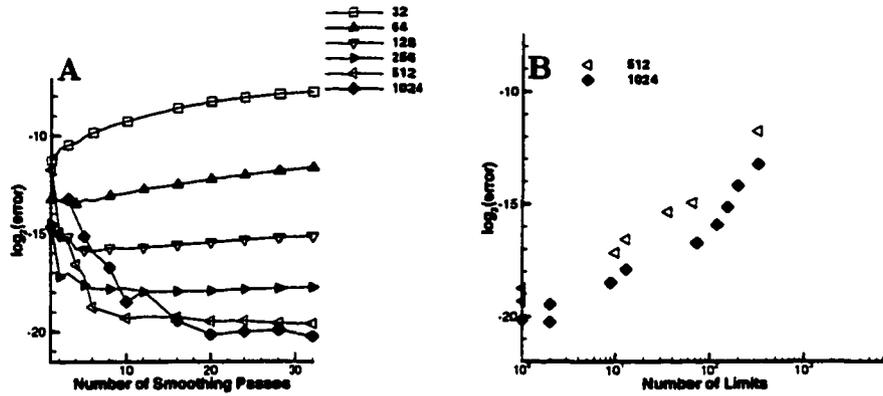


Figure 2.18 The effects of limiting the grid point movement. The options used were RPV, no third order terms, $\lambda_1 = 70$, and $\lambda_2 = 0$. The error used was the L_2 error. The legend indicates the number of grid points.

of the results when other combinations of MPDATA scheme were used. When a small number of grid points was used the effect of increasing the number of smoothing passes was detrimental to the solution. The smoothing passes decreased the variation in the grid metric term $\frac{\partial \xi}{\partial x}$ and thus had a tendency to decluster the grid points. By comparing panels A and B in figure 2.18 it is apparent that when greater numbers of grid points were used the smoothing passes reduced the number of times the grid point movements had to be limited. The grid point limiting had a dramatic detrimental effect on the error as shown in figure 2.18 panel B. The number of smoothing passes increased the $\frac{\partial x}{\partial \xi}$ values and therefore allowed for a greater region in which the grid point could move. The best results were obtained when just enough smoothing passes were used to prevent the limiting of grid point movement.

2.4.1.3.3 Computational cost: Two versions of the one-dimensional code were written. One used a static uniform grid only, while the other had the capability of using a dynamic grid. Steps were taken to optimize the computational efficiency of both codes. Figure 2.19 compares the error as a function of computational cost for both codes.

Although the DAG technique used fewer grid points, the equation for redistributing the grid points was solved at each time step. This added computational expense was nearly the computational savings that were achieved due to the method. Note, however, that the shape of the curves is different between the two panels in figure 2.19. When relatively few grid points were used the slope of the curve in panel B was steeper than the slope in panel A. This indicates that when the number of grid points is small, increasing the number of points in the DAG code is more advantageous than adding grid points to

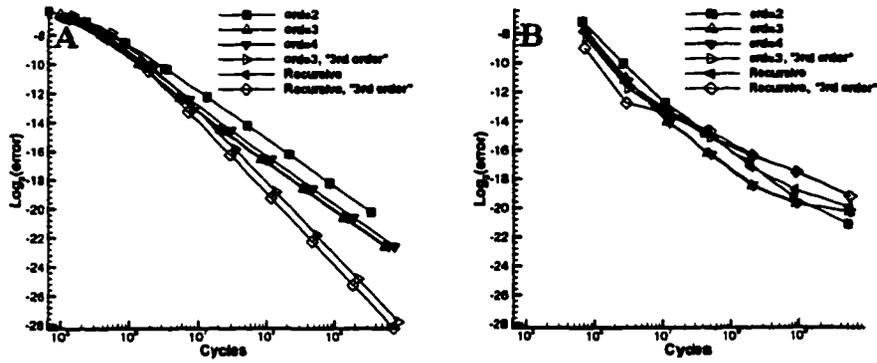


Figure 2.19 L_2 error versus computational cost when $\lambda = 70$; A) Static uniform grid; B) Dynamic grid, $\lambda = 70$

the static grid code. However, as the grid points increase the benefits diminish and eventually become liabilities. The second and third order accuracy are clearly evident in the slopes of the curves in panel A (static uniform grid) for the computations without and with the third order terms.

The requirement of using relatively few grid points in order to gain the benefit of the DAG technique was not seen as a disadvantage. For most atmospheric models, using more than 128 grid points in any one coordinate direction is unusual. Therefore, it was postulated that the benefits of the DAG technique would be more evident as the number of dimensions increased.

2.4.2 Two-dimensional model

A two-dimensional model was initially written to gain experience with the multidimensional MPDATA scheme. The two-dimensional grid generator that was described in section 2.3.4 was initially developed and evaluated with the two-dimensional rotating cone problem. The qualitative results of the two-dimensional code more clearly illustrate the benefits of the different MPDATA options in conjunction with the DAG technique than did the one-dimensional model. Section 2.4.2.1 describes aspects of the two-dimensional models that were not obvious extensions to the one-dimensional case. Section 2.4.2.2 discusses several experiments involving the rotating cone problem. Note that the limiting of the grid point movement was not extended to multidimensions, and therefore is not discussed in this section.

2.4.2.1 Two-dimensional model description

The rotating cone problem was solved in two dimensions. Therefore the equation

$$\frac{\partial q}{\partial t} + u \frac{\partial q}{\partial x} + v \frac{\partial q}{\partial y} = 0 \quad (2.112)$$

was solved on a square domain bounded by $0 \leq x \leq 1$ and $0 \leq y \leq 1$. The value of q was initialized as

$$q = \max \left(4 - \frac{4}{0.15} \sqrt{(x - 0.5)^2 + (y - 0.75)^2}, 0 \right) \quad (2.113)$$

The velocity field was held constant in time and was a prescribed vortex:

$$\begin{aligned} u &= -2\pi (y - 0.5) \\ v &= 2\pi (x - 0.5) \end{aligned} \quad (2.114)$$

As with the one-dimensional grid, the initial grid was generated by ten iterations of initializing the grid with q and then generating a new grid. The time step was held constant throughout the integration, was based on the initial grid spacing, and was calculated as:

$$\Delta t = \frac{s}{\max_{i,j} \left(\left| U_{i+1/2,j}^{n=0} \right| + \left| V_{i,j+1/2}^{n=0} \right| \right)} \quad (2.115)$$

where $U_{i+1/2,j}^{n=0}$ and $V_{i,j+1/2}^{n=0}$ were the initial Courant numbers without consideration of the grid speeds. The contravariant velocities used in calculating $U_{i+1/2,j}^{n=0}$ and $V_{i,j+1/2}^{n=0}$ are given below (equation 2.117).

The order of operations during each time step was the same in the two-dimensional as in the one-dimensional model with the exception of step number 5 in section 2.4.1.1, which performed the limiting of grid point movement. The grid speeds were determined using central differences:

$$\begin{aligned} \left. \frac{\partial x}{\partial t} \right|_{i+1/2,j}^{n+1/2} &= \frac{x_{i+1/2,j}^{n+1} - x_{i+1/2,j}^n}{\Delta t} \\ \left. \frac{\partial y}{\partial t} \right|_{i,j+1/2}^{n+1/2} &= \frac{y_{i,j+1/2}^{n+1} - y_{i,j+1/2}^n}{\Delta t} \end{aligned} \quad (2.116)$$

The contravariant velocities were calculated using the metric terms described in section 2.1:

$$\begin{aligned} \tilde{u}_{i+1/2,j}^{n+1/2} &= \frac{1}{J_{2,i+1/2,j}} \left[\left(u_{i+1/2,j} - \left. \frac{\partial x}{\partial t} \right|_{i+1/2,j}^{n+1/2} \right) \left. \frac{\partial y}{\partial \eta} \right|_{i+1/2,j}^{n+1} - \right. \\ &\quad \left. \left(v_{i+1/2,j} - \left. \frac{\partial y}{\partial t} \right|_{i+1/2,j}^{n+1/2} \right) \left. \frac{\partial x}{\partial \eta} \right|_{i+1/2,j}^{n+1} \right] \\ \tilde{v}_{i,j+1/2}^{n+1/2} &= \frac{1}{J_{2,i,j+1/2}} \left[\left(v_{i,j+1/2} - \left. \frac{\partial y}{\partial t} \right|_{i,j+1/2}^{n+1/2} \right) \left. \frac{\partial x}{\partial \xi} \right|_{i,j+1/2}^{n+1} - \right. \\ &\quad \left. \left(u_{i,j+1/2} - \left. \frac{\partial x}{\partial t} \right|_{i,j+1/2}^{n+1/2} \right) \left. \frac{\partial y}{\partial \xi} \right|_{i,j+1/2}^{n+1} \right] \end{aligned} \quad (2.117)$$

These were non-dimensionalized by $\frac{\Delta t}{\Delta \xi}$ to form the Courant numbers $U_{i+1/2,j}^{n+1/2}$ and $V_{i,j+1/2}^{n+1/2}$. Note that the time level of the physical velocity is not specified in equation 2.117 since it is not a function of time.

As in the one-dimensional case, a divergence term was needed to put equation 2.112 into flux form. Recall that although the physical velocity field is divergent free the movement of the grid points causes the velocity field in the computational domain to be divergent. This divergence term was calculated as

$$q_{i,j} \frac{\partial \hat{U}_i^{(0)}}{\partial \xi_i} = q_{i,j}^n \left(\hat{U}_{i+1/2,j}^{n+1/2} - \hat{U}_{i-1/2,j}^{n+1/2} + \hat{V}_{i,j+1/2}^{n+1/2} - \hat{V}_{i,j-1/2}^{n+1/2} \right) \quad (2.118)$$

Recall that the divergence term is only required for the first MPDATA iteration. The value of the divergence term for additional iterations was set to zero.

Qualitative assessment of the profiles as well as the L_2 error norm, diffusion, and phase error were used in the evaluation of the error associated with two-dimensional rotating cone problem. The L_2 error norm was calculated as

$$E_{L_2} = \left(\frac{1}{A} \int_A (\hat{q}(x, y) - q(x, y))^2 dx dy \right)^{1/2} \quad (2.119)$$

where $\hat{q}(x, y)$ was the analytic solution and $q(x, y)$ was the computed solution. The trapezoidal rule was used to perform the numerical quadrature. E_{L_2} did not give any measure of the diffusion or phase error that existed. The diffusion error was calculated as:

$$E_d = \max_{i,j}(\hat{q}_{i,j}) - \max_{i,j}(q_{i,j}) \quad (2.120)$$

The phase error was determined by calculating the distance between the maximum value of the computed solution and the location of the maximum of the analytical solution.

The L_2 , diffusion error, and phase errors measure different properties. The L_2 error is a global measure of the error. The goal of the DAG technique is to reduce the error in local regions. Therefore, the L_2 error norm may not be the best measure of the DAG performance. The diffusion and phase errors are strictly local measures of the error and therefore are more applicable to evaluate a DAG technique. The reduction of the diffusion error and phase error are of particular importance when modeling atmospheric water vapor transport. If a scheme is too diffusive, critical concentrations of water vapor mixing ratio may not be achieved to cause condensation of water and thus cumulus convection that physically should occur. If a scheme has a significant phase error, precipitation may be predicted in the wrong location. For these reasons, the diffusion and phase errors were considered to be more significant measures of a schemes performance than was the L_2 error.

2.4.2.2 Two-dimensional experimental results

Figure 2.20 shows surface plots of the the initial condition, and the solutions after one revolution. MPDATA with the RPV option and the "third" order terms, a leap frog scheme without artificial

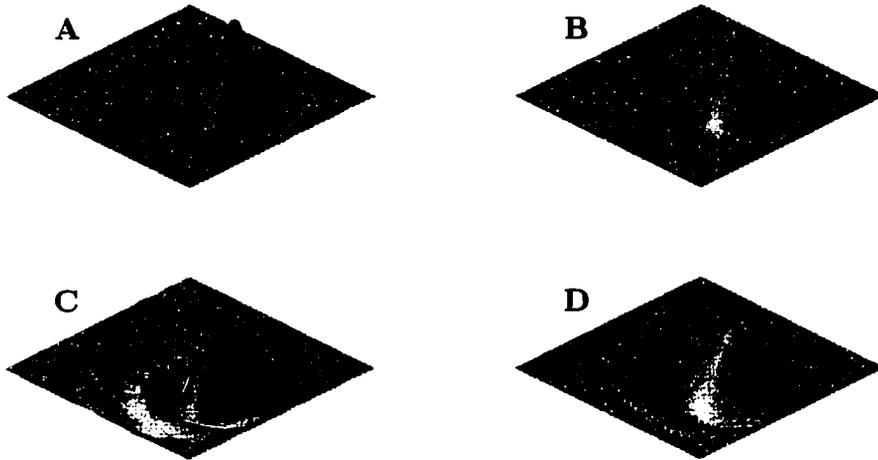


Figure 2.20 MPDATA compared to a leap frog scheme for the rotating cone problem after one revolution. All computations are on a static, uniform grid. A) Initial condition; B) MPDATA scheme with RPV option and “third” order terms; C) Leap frog scheme without artificial dissipation; D) Leap frog scheme with artificial dissipation.

dissipation, and a leap frog scheme with artificial dissipation were used to compute the solutions in panels B, C, and D respectively. The sign preserving nature of the MPDATA scheme is apparent when contrasted to the dispersive ripples present in both of the leap frog schemes. Additionally, the smallness of the internal diffusion of the MPDATA scheme is comparable to that of the leap frog scheme without artificial dissipation. This comparison was made because the standard advection scheme used in MM5 is a leap frog scheme with artificial dissipation. This clearly illustrates the advantages of the MPDATA scheme even without considering the DAG technique.

The advantages of the DAG technique are evident in figures 2.21 and 2.22 which show contour plots in the range $0.25 \leq x \leq 0.75$ and $0.5 \leq y \leq 1.0$. Figure 2.21 shows results without the 3rd order option while figure 2.22 includes the 3rd order terms. All of the results are after 6 revolutions, used 41 grid points in both the x and y directions, stretching parameters of $\lambda_x = 5.0$ and $\lambda_y = 0$, 4 smoothing passes, and a safety factor $s = 0.6$. The effects of the DAG technique and the “third” order terms are apparent. The diffusion error appears to be much less in the cases with the dynamic adaptive grid, while the “third” order correction terms result in a more symmetric distribution of the error and thus less phase error. This symmetric distribution of the error due to the “third” order correction terms is in agreement with the findings of Margolin and Smolarkiewicz (1989).

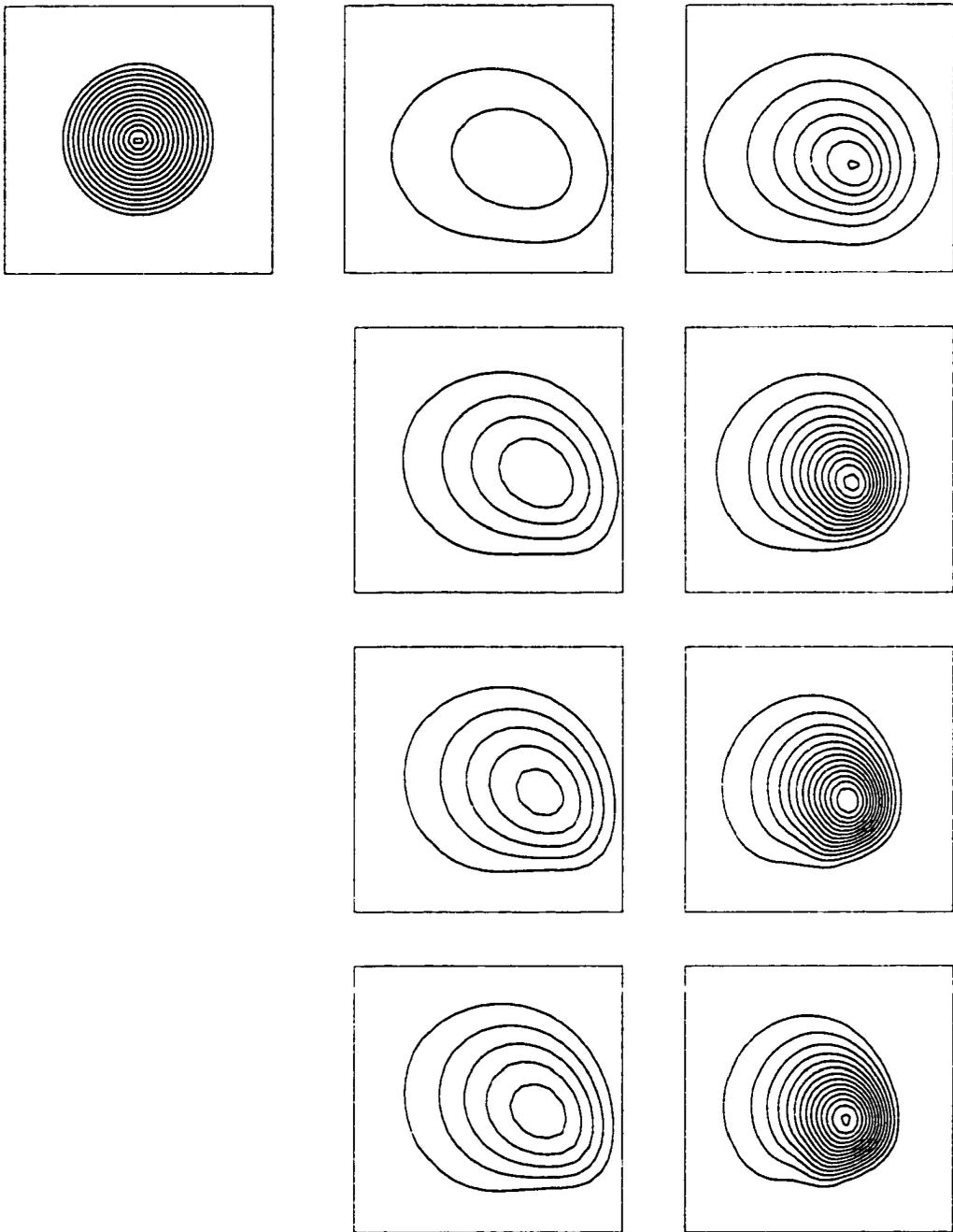


Figure 2.21 Results of the rotating cone problem after 6 revolutions without “third” order correction terms. The upper left panel shows the initial condition, the center column gives results using the static MPDATA scheme, and the right column shows results using the dynamic MPDATA scheme. The results on the first, second, third, and fourth rows show the results with two iterations, three iterations, four iterations, and the RPV option respectively.

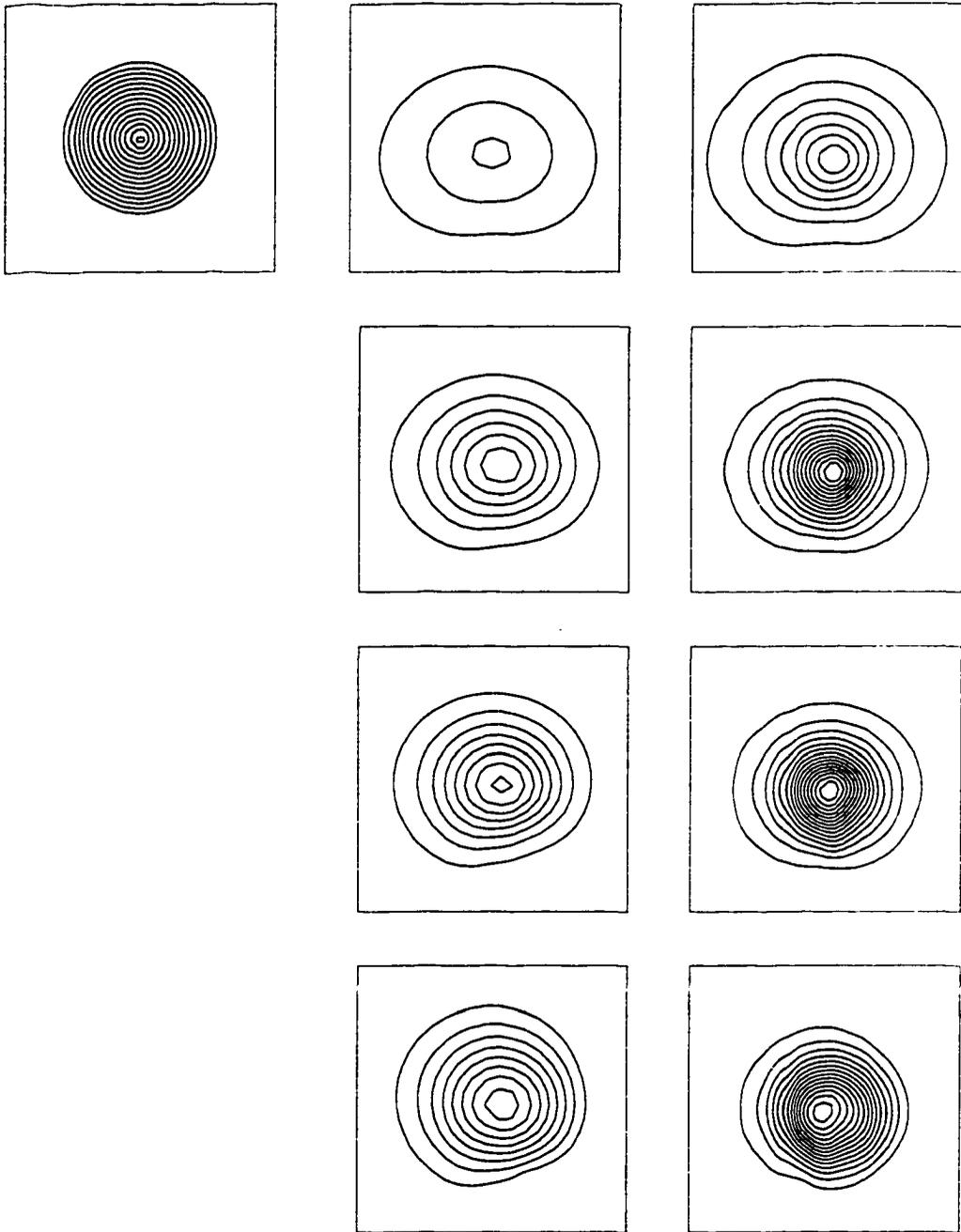


Figure 2.22 Results of the rotating cone problem after 6 revolutions with “third” order correction terms. The upper left panel shows the initial condition, the center row are results using the static MPDATA scheme, and the right column are results using the dynamic MPDATA scheme. The results on the first, second, third, and fourth rows show the results with two iterations, three iterations, four iterations, and the RPV option respectively.

Unlike the one-dimensional test case, additional iterations or alternatively the RPV option and the “third” order correction terms make a significant improvement on the solution. Table 2.1 shows the L_2 , diffusion, and phase errors and the number of machine CPU cycles for the different configurations of the code. The L_2 error norm was calculated by comparing the numerical solution on the final grid to the analytical solution on the final grid.

The magnitude of the initial gradient of q in the area of the cone was 26.7 which was rather steep. Therefore, small perturbations of the grid point where the maximum analytical value of q was located resulted in significant changes of this maximum value. Since the grid was allowed to move, the grid points of the final and initial grids did not coincide. It was very unlikely that the maximum value of the initial condition represented on the final grid would be the same as the maximum value represented on the initial grid. Therefore, the diffusion error was calculated by taking the difference in the maximum value of the analytical solution for q on the initial grid and the maximum value of the numerical solution for q on the final grid. Likewise the phase error was found by finding the distance between the location of the maximum value of the analytical solution on the initial grid and the location of the maximum value of the numerical solution on the final grid. This prevented the possibility that the maximum analytic solution from being different from the maximum value of the initial condition.

The effect of the different options on the numerical error and computational cost was as expected, except for the combination of DAG with the “third” order terms. When four iterations was used, the L_2 error norm was smaller than with three iterations, but the diffusion error was unexpectedly negative. The reason for this is not precisely known. It could have been due to Gibbs phenomena at the peak. This is technically possible since the MPDATA scheme is not monotonic except around the value of zero. Gibbs-type phenomena were observed when the minimum value was greater than zero, but it was never observed at the peak of the cone. Therefore, this negative diffusion was probably not due to a second-order Gibbs-type error. The error was probably due to errors that occurred in the computation of the divergence term. Although the flow was divergence free, an apparent divergence was present due to the movement of the grid points. This divergence was calculated using a second order central difference. Therefore, oscillations could have existed in this field. When the RPV option was used with a DAG and the “third” order terms, positive diffusion was again present and to such an extent that the diffusion error was more than when only two iterations were used.

The even spacing and lack of movement of the grid points caused the phase error of the static grid computations to be quantized by the grid spacing, meaning that the phase error was $m\Delta x + n\Delta y$ where m and n are integers. As a result it was likely that the phase error of the static grid computations

Table 2.1 Errors and costs associated with the two-dimensional rotating cone problem. An R under the number iterations indicates that the recursive pseudo velocity option was used.

Number of Grid Points	Dynamic Grid	Number of Iterations	3rd order Terms	L_2 Error	Diffusion Error	Phase Error	Cycles $\times 10^9$
41 \times 41	No	2	No	0.341	2.814	0.079	1.44
41 \times 41	No	3	No	0.306	2.323	0.079	2.40
41 \times 41	No	4	No	0.300	2.180	0.075	3.36
41 \times 41	No	R	No	0.297	2.132	0.075	2.51
41 \times 41	No	2	Yes	0.322	2.738	0.025	2.08
41 \times 41	No	3	Yes	0.219	1.859	0.000	3.67
41 \times 41	No	4	Yes	0.176	1.449	0.000	5.27
41 \times 41	No	R	Yes	0.158	1.384	0.000	3.15
41 \times 41	Yes	2	No	0.266	2.032	0.067	53.11
41 \times 41	Yes	3	No	0.210	0.703	0.065	61.38
41 \times 41	Yes	4	No	0.196	0.336	0.057	69.29
41 \times 41	Yes	R	No	0.191	0.250	0.053	69.44
41 \times 41	Yes	2	Yes	0.240	1.870	0.045	53.50
41 \times 41	Yes	3	Yes	0.142	0.066	0.023	60.46
41 \times 41	Yes	4	Yes	0.132	-0.178	0.017	65.40
41 \times 41	Yes	R	Yes	0.125	0.166	0.026	66.19
101 \times 101	No	R	Yes	0.038	0.429	0.000	46.39
151 \times 151	No	R	Yes	0.025	0.409	0.007	154.57
201 \times 201	No	R	Yes	0.018	0.300	0.000	363.87

could be zero. This simply required both n and m to be zero. Because the grid was able to move in the DAG computations, it was unlikely that locations of the maximums on the initial and final grids would exactly coincide.

Table 2.2 shows the percentage change in the errors and computational cost of additional iterations or the RPV option compared to the basic MPDATA scheme with two iterations, no “third” order terms, and a static uniform grid. In all cases except one, additional iterations or the RPV option reduced all three error measures. The one exception was the case where the dynamic grid was used in conjunction the “third” order terms. With each subsequent iteration the resulting error reduction became less for all three error measures. However, except in the cases noted above, the RPV option yielded the better results at a lower or comparable cost than using four iterations. The computational penalty of using additional iterations or the RPV option was significantly less for the DAG option. Since the majority of the computational expense of the DAG runs was in the computation of the new grid and the corresponding numerics to account for the moving grid, the relative impact of using the additional iterations or RPV option was less significant.

Tables 2.3 shows the percentage change in the L_2 , diffusion, and phase errors and the computational cost when the “third” order terms were used. Recall from figures 2.21 and 2.22 that the qualitative effect of the “third” order terms was to distribute the error more equally. This is seen in the dramatic reduction of the phase error in table 2.3. Likewise, this would have a positive impact of the L_2 error norm. However, the “third” order correction terms also had a positive effect on the diffusion error as can be seen in the diffusion error. When a dynamic grid was used the cost of the computations with the “third” order error terms was less than those without the “third” order correction terms. One possible explanation for this is that since the error was more evenly distributed when the “third” order error terms were used, the grid generator had to perform fewer iterations to converge. Thus the added cost of calculating the “third” order terms was more than offset by the savings of not having to iterate as many times to determine the new grid.

Table 2.4 shows the impact of the DAG option. The most dramatic improvement is seen in the diffusion error. Reduction of 60 percentage points was not uncommon. The reduction of the L_2 error norm is not as dramatic but is still very significant. The previous discussion should be kept in mind when considering the difference in the phase error. Although it appears that the phase error increases with the use of the DAG option when the “third” order terms are used, the actual effect was minimal. Although dramatic gains were achieved, they were done so at a great computational expense. At least an order of magnitude more computational resources were required to make the calculation with the

Table 2.2 The effect of additional MPDATA iterations. All of the schemes are compared to the MPDATA scheme with 2 iterations, no "3rd" order terms, and a 41×41 grid. Percentages of L_2 , diffusion, and phase errors represent reduction of error. Percentage of cycles represent percentage increase in machine CPU cycles.

Dynamic Grid	Number of Iterations	3rd order Terms	L_2 Error	Diffusion Error	Phase Error	Cycles
No	3	No	10.3%	17.4%	0.0%	66.7%
No	4	No	12.0%	22.5%	5.1%	133.3%
No	R	No	12.9%	24.2%	5.1%	74.3%
No	3	Yes	35.8%	33.9%	100.0%	154.9%
No	4	Yes	48.4%	48.5%	100.0%	266.0%
No	R	Yes	53.7%	50.8%	100.0%	118.8%
Yes	3	No	38.4%	75.0%	17.7%	4,162.5%
Yes	4	No	42.5%	88.1%	27.8%	4,711.8%
Yes	R	No	44.0%	91.1%	32.9%	4,722.2%
Yes	3	Yes	58.4%	97.7%	70.9%	4,098.6%
Yes	4	Yes	61.3%	106.3%	78.5%	4,441.7%
Yes	R	Yes	63.3%	94.1%	67.1%	4,496.5%

DAG option.

When the DAG computations were analyzed, it was found that most of the computational effort was expended solving the elliptic equations to determine the grid point distribution at each time step (78% of the CPU time), generating the weight function from the prognostic variable field (6%), solving the advection problem using MPDATA and controlling the logic of the code(12%), and calculating the contravariant velocities at each step(4%). There were 4734 time steps used in the DAG computations, compared to 2577 time steps in the static uniform grid computations. The time step used in the DAG computations was shorter than the time step for the static grid computations because the minimum grid spacing of the initial grid was smaller in the DAG computations. Adjusting the computational expense of the DAG computations to account for the added number of time steps showed that the effort to perform the MPDATA scheme and the logic of the code was 1.4 times the expense for the static grid computations. Since the same steps were executed for the MPDATA scheme in each, the extra expense was due to added calculations to move the grid points and account for that movement.

Three additional computations shown at the bottom of table 2.1 show the effect of increasing the number of grid points in the static and uniform grid solver while using the RPV and the "third" order terms. The L_2 error norms are the best of any computations performed. However, the diffusion errors

Table 2.3 The effect of the “third” order correction terms. All of the schemes are compared to the MPDATA scheme with 2 iterations, no “3rd” order terms, on a static uniform grid. Percentages of L_2 , diffusion, and phase errors represent reduction of error. Percentage of cycles represent percentage increase in machine CPU cycles.

Dynamic Grid	Number of Iterations	3rd order Terms	L_2 Error	Diffusion Error	Phase Error	Cycles
No	2	No	0.0%	0.0%	0.0%	0.0%
No	2	Yes	5.6%	2.7%	68.4%	44.4%
No	3	No	10.3%	17.4%	0.0%	66.7%
No	3	Yes	35.8%	33.9%	100.0%	154.9%
No	4	No	12.0%	22.5%	5.1%	133.3%
No	4	Yes	48.4%	48.5%	100.0%	266.0%
No	R	No	12.9%	24.2%	5.1%	74.3%
No	R	Yes	53.7%	50.8%	100.0%	118.8%
Yes	2	No	22.0%	27.8%	15.2%	3,588.2%
Yes	2	Yes	34.1%	33.5%	43.0%	3,656.9%
Yes	3	No	38.4%	75.0%	17.7%	4,162.5%
Yes	3	Yes	58.4%	97.7%	70.9%	4,098.6%
Yes	4	No	42.5%	88.1%	27.8%	4,711.8%
Yes	4	Yes	61.3%	106.3%	78.5%	4,441.7%
Yes	R	No	44.0%	91.1%	32.9%	4,722.2%
Yes	R	Yes	63.3%	94.1%	67.1%	4,496.5%

Table 2.4 The effect of the dynamic adaptive grid. All of the schemes are compared to the MPDATA scheme with 2 iterations, no "3rd" order terms, on a static uniform grid. Percentages of L_2 , diffusion, and phase errors represent reduction of error. Percentage of cycles represent percentage increase in machine CPU cycles.

Dynamic Grid	Number of Iterations	3rd order Terms	L_2 Error	Diffusion Error	Phase Error	Cycles
No	2	No	0.0%	0.0%	0.0%	0.0%
Yes	2	No	22.0%	27.8%	15.2%	3,588.2%
No	3	No	10.3%	17.4%	0.0%	66.7%
Yes	3	No	38.4%	75.0%	17.7%	4,162.5%
No	4	No	12.0%	22.5%	5.1%	133.3%
Yes	4	No	42.5%	88.1%	27.8%	4,711.8%
No	R	No	12.9%	24.2%	5.1%	74.3%
Yes	R	No	44.0%	91.1%	32.9%	4,722.2%
No	2	Yes	5.6%	2.7%	68.4%	44.4%
Yes	2	Yes	34.1%	33.5%	43.0%	3,656.9%
No	3	Yes	35.8%	33.9%	100.0%	154.9%
Yes	3	Yes	58.4%	97.7%	70.9%	4,098.6%
No	4	Yes	48.4%	48.5%	100.0%	266.0%
Yes	4	Yes	61.3%	106.3%	78.5%	4,441.7%
No	R	Yes	53.7%	50.8%	100.0%	118.8%
Yes	R	Yes	63.3%	94.1%	67.1%	4,496.5%

were not nearly as good as when the DAG option with RPV and “third” order terms was used on the 41×41 grid. The diffusion error was 16.7% better with the DAG option and the computational cost of the DAG calculation was only 19% of the static and uniform grid calculation with 201×201 grid points.

2.4.3 Conclusions

The use of the MPDATA scheme in one- and two-dimensional models was demonstrated. The one-dimensional code simulated the advection of a Gaussian pulse while the two-dimensional code performed the rotating cone test case. The one-dimensional code was used to determine the characteristics of the MPDATA scheme on a moving grid as the time step and the number of grid points were varied and the effect of limiting the grid point movement. The two-dimensional code illustrated the effect of adding additional iterations or using the RPV option, using the “third” order correction terms, and using a dynamic adaptive grid.

It was found that the one-dimensional DAG code performed as if it were a third or higher order scheme when there were relatively few points. As few as 16 grid points gave reasonable answers when the DAG technique was used. The actual error of the computations when DAG was used was comparable to the results of static, uniform grid calculations with up to 8 times more grid points. Based on the L_2 error norm, when the number of grid points became large, the order of the DAG schemes reduced to first or even zeroth order. This was primarily due to the need to limit the grid point movement in order to prevent violation of the CFL condition.

The criteria for the limiting of the grid points was derived and limitation of the grid points was implemented. It was found that by increasing the number of smoothing passes on the weight function in the grid generator that the number of grid point movement limits could be reduced. The critical determining parameter in the limiting of the grid points was the smallness of $\frac{\partial \tau}{\partial \xi}$. Since the smoothing passes spread the grid points out this had a tendency to increase $\frac{\partial \tau}{\partial \xi}$ and thus reduce the number of times the movement of grid points needed to be limited. The computational efficiency of the one-dimensional grid generator made the DAG option comparable in terms of CPU cycles to the static, uniform grid solver. Note there were 4 to 8 times the number of points in the static uniform computations than the DAG computations but that the expense of generating the grid did not exist. Therefore, the extra expense of adding grid points to the static uniform computations was about the same as moving the grid in the DAG computations.

The two-dimensional code solved the rotating cone problem for six full revolutions. Once again the

DAG technique demonstrated that the L_2 error norm, the diffusion error, and the phase error could all be reduced while using fewer grid points. Additional iterations or the RPV option and the “third” order terms had a significant impact when the DAG option was used. The DAG option had the most significant impact on reducing the diffusion of the cone (90% reduction) although significant decreases in the global L_2 error norm (35% reduction) and in phase error (29% reduction) were also achieved. Additional iterations or using the RPV option reduced the error in the static and DAG cases, but had diminishing benefit as the number of iterations increased. However, the RPV option had lower computational cost associated with it for the static, uniform grid case and comparable computational cost when a dynamic adaptive grid was used. The relative increase in computational cost of adding additional iterations or using the RPV option was much less when the DAG option was compared with the static grid computations, because the majority of the computational cost of the DAG computations was in generating the grid which was not effected by the number of iterations. The “third” order terms greatly reduced the phase error in all the computations. The cost of adding the “third” order terms was more than offset in the computational reduction of generating the grid in the DAG option. The cost of the DAG option was an order of magnitude above the cost of performing the same calculation on a static grid (2.5 gigacycles versus 69.4 gigacycles). However, the computational cost of achieving the same level of diffusion error using a static uniform grid was an order of magnitude higher again (69.4 gigacycles versus 363.9 gigacycles). Since diffusion and phase error were considered better measures of a scheme’s performance the DAG technique was a more computationally efficient technique than simply increasing the resolution of static, uniform grid.

The computations performed in these two-models only solved analytical test cases. If a complete model to simulate fluid flow were developed, the number of equations solved on the dynamic grid would be at least five and probably six since species continuity equations were included. For sub-sonic problems the elliptic pressure equation would be as expensive to solve as the grid movement equations. Thus in more complete models, the relative computational overhead of the DAG technique becomes much smaller, making it even more effective.

3 ATMOSPHERIC TRANSPORT

This chapter presents an application of the MPDATA scheme to the problem of passive atmospheric transport using a set of subroutines called the dynamic grid advection component (DGAC). Note that DGAC is the name of a set of subroutines that were used to compute atmospheric passive tracer transport, whereas DAG is the name a general numerical technique. The DGAC was developed to use the MPDATA scheme on a dynamic adaptive grid. Dynamic MPDATA will refer to computations performed with the DGAC while the grid points are allowed to move. Static MPDATA will refer to computations performed with the DGAC while the grid points remained uniformly spaced in the horizontal directions and static throughout the simulation. This set of routines was written so that it could be driven by a test program to solve analytical problems similar to those used in section 2.3 and by the Fifth-Generation Penn State/NCAR Mesoscale Model (MM5). Relevant discussion of the model's capabilities, coordinate system, governing equations, grid, and boundary conditions are discussed in section 3.1.

The design of the DGAC considered several different issues which are discussed in section 3.2. The advection equation that was used in chapter 2 was not consistent with the MM5 prognostic equations. A consistent form of the advection equation and MPDATA schemes are developed in section 3.2.1. The DGAC grid differs from the MM5 grid and yet they discretize the same physical domain. Thus the correspondence between the two grids is discussed in section 3.2.3. How the DGAC was initialized, how the wind fields were prescribed to it, and how the boundary conditions were implemented conclude section 3.2.

The different grid staggering schemes used in MM5 and the DGAC, the ability for the DGAC grid to adapt dynamically, and the desire to compare MM5 and DGAC results on the same grid necessitated the use of interpolation. In order to minimize the effects of the interpolation, the second order monotone interpolation scheme developed by Smolarkiewicz and Grell (1992) was used, and is discussed in section 3.3.

Section 3.4 describes three test cases and presents results of each. These test cases were progressively

more stringent as well as physically realistic. Initially, an analytical velocity field and tracer field were used to test the MM5/DGAC combination. The second test case used the same idealized tracer field, but physically realistic wind fields. The third test case used a physically realistic wind field and a tracer that was initialized to the same values as the water vapor mixing ratio. This tracer field did not include any effects for precipitation or evaporation, however, and thus had no feedback on MM5 circulation.

3.1 MM5

The model chosen to drive the dynamic grid adaptation component (DGAC) was the Fifth-Generation Penn State/NCAR Mesoscale Model (MM5) which is derived from an atmospheric mesoscale model originally developed by Anthes and Warner (1979). The model has gone through several versions with the latest major revision developed by Dudhia (1993) and Grell et al. (1995). This model is thoroughly explained in these two sources plus Haagenson et al. (1994). The model was written in FORTRAN 77 and contains over 50,000 lines of code. Except for some minor modifications to include the DGAC, to include a component to advect a tracer using the same scheme as MM5 but on a different resolution grid, and to add the capability of advecting an inert tracer, MM5 was not modified. The details of this section were taken from the aforementioned sources and the source code itself.

MM5 is a mesoscale model, meaning that it is designed to simulate atmospheric process on spatial scales on the order of magnitude of hundreds of kilometers. It has many capabilities including:

1. hydrostatic or non-hydrostatic modes
2. nested grids
3. four-dimensional data assimilation
4. parameterizations of
 - (a) horizontal diffusion
 - (b) resolvable scale precipitation processes
 - (c) implicit cumulus convection
 - (d) shallow convection
 - (e) planetary boundary layer
 - (f) atmospheric radiation

The set of prognostic equations that were solved are dependent on whether the hydrostatic or the non-hydrostatic option is used. The hydrostatic option assumes that the atmosphere is always in hydrostatic balance; therefore, there is no explicit prognostic equation for vertical motion. The prognostic equations for the non-hydrostatic option were formulated without this assumption, so there is a prognostic equation for vertical velocity. Only the non-hydrostatic option was used, and the governing equations for it are given in detail in section 3.1.1.

Due to the disparity between the spatial and temporal scales involved in modeling the atmosphere, many of the relevant processes can not be resolved by the finite difference grids that can be effectively run on current computer systems. Therefore, many processes are parameterized. MM5 simulations are strongly dependent on the choice of parameterizations.

The parameterization of horizontal diffusion was included as a means of artificial dissipation, thus providing stability control and anti-aliasing. The scheme used was the fourth-order artificial dissipation technique of Smagorinsky et al. (1965) that takes into account the deformation of a fluid parcel. This parameterization was the only one that had a strong influence on this work. The standard advection scheme within MM5 is a centrally differenced leap-frog scheme. Without some form of artificial dissipation, the scheme produces dispersive ripples that can contaminate the solution. In addition, the leap-frog scheme can cause temporal decoupling. Although horizontal diffusion helps remedy these problems, it also provides dissipation to resolved fields. It will be seen in chapter 4 that the MPDATA is less diffusive than the standard MM5 advection scheme.

Although, the parameterizations of cumulus convection, precipitation processes, shallow convection, planetary boundary layer, and atmospheric radiation contribute to a complete representation of the all atmospheric processes, they are not part of the resolved tracer transport and the specific parameterizations used were not particularly important. Certainly the atmospheric circulations computed here might have been different with a different set of parameterizations, but the conclusions based on the results would have been altered only slightly if at all.

This section gives a brief overview of the capabilities of MM5. Section 3.1.1 describes the coordinate system utilized within MM5. Section 3.1.1 reviews these prognostic equations used to model the atmospheric dynamics within MM5. The staggered grid that is implemented in MM5 is different than the staggered grid used for the MPDATA scheme. Therefore, the MM5 grid is described in detail in section 3.1.2, while the MPDATA grid is described in detail in section 3.2.3.1. The MM5 boundary conditions are reviewed in section 3.1.3.

3.1.1 MM5 coordinate system and governing equations

The coordinate system for the non-hydrostatic version of MM5 is complicated by a transformation from the earth's spherical coordinate system to a horizontal Cartesian system. In addition, topographical effects are dealt with in a vertical coordinate transformation.

The projection from the spherical coordinates to a Cartesian coordinate system introduces metric terms into the horizontal derivatives. The inverses of these metric terms are traditionally referred to as map factors and are given the symbol m . The Lambert conformal projection was used since it is the standard projection for mid-latitude regions which were used in this work. This transformation is a conical transformation and preserves angles between intersecting lines.

The thermodynamic prognostic variables are represented as a perturbations around a time invariant basic state. This basic state is referred to as a reference state in Grell et al. (1995). Therefore, pressure, temperature, and density perturbations are defined by:

$$p(t, x, y, z) = p_0(z) + p'(t, x, y, z) \quad (3.1)$$

$$T(t, x, y, z) = T_0(z) + T'(t, x, y, z) \quad (3.2)$$

$$\rho(t, x, y, z) = \rho_0(z) + \rho'(t, x, y, z) \quad (3.3)$$

where the basic state is in hydrostatic balance:

$$\frac{\partial p_0}{\partial z} = -\rho_0 g \quad (3.4)$$

and g is the acceleration due to gravity. It is common in atmospheric science to use pressure as a vertical coordinate. Vertical hydrostatic basic state properties are used in the non-hydrostatic version of MM5. The model uses a vertical coordinate system, called a σ coordinate, that is normalized by the difference between the surface basic state pressure and the specified pressure at the top of the model (which was 100mb). The advantage of a σ coordinate is that it prevents horizontal layers from intersection the earth's surface. The σ coordinate system is defined as:

$$\sigma = \frac{p_0 - p_t}{p_s - p_t} = \frac{p_0 - p_t}{p^*} \quad (3.5)$$

where p_0 , p_t and p_s are the basic state pressure, the basic state pressure at the model top and the basic state pressure at the surface respectively. p^* is defined as:

$$p^* = p_s - p_t \quad (3.6)$$

The governing prognostic equations of the non-hydrostatic MM5 include momentum equations in three coordinate directions, pressure perturbation equation, energy equation, and species conservation

equations for water vapor, cloud water, and rain water mixing ratios. All of these equations are transformed to σ coordinates and a Lambert conformal projection. The governing equations appear in Grell et al. (1995). Also in appendix B, the governing equations are developed from the continuity, momentum, and energy equations that can be found in any fluid dynamics text.

The pressure perturbation, momentum, and temperature equations are:

$$\begin{aligned} \frac{\partial(p^* p')}{\partial t} &= -m^2 \frac{\partial}{\partial x_i} \left(\frac{p^* p' u_i}{m} \right) - \frac{\partial(p^* p' \dot{\sigma})}{\partial \sigma} + p' \text{DIV} \\ &\quad - m^2 p^* \gamma p \left[\frac{\partial}{\partial x_i} \left(\frac{u_i}{m} \right) - \frac{\sigma}{m p^*} \frac{\partial p^*}{\partial x_i} \frac{\partial u_i}{\partial \sigma} \right] + \gamma p \rho_0 g \frac{\partial w}{\partial \sigma} + p^* \rho_0 g w \end{aligned} \quad (3.7)$$

$$\begin{aligned} \frac{\partial(p^* u_i)}{\partial t} &= -m^2 \frac{\partial}{\partial x_j} \left(\frac{p^* u_i u_j}{m} \right) - \frac{\partial(p^* u_i \dot{\sigma})}{\partial \sigma} + u_i \text{DIV} \\ &\quad - \frac{m p^*}{\rho} \left(\frac{\partial p'}{\partial x_i} - \frac{\sigma}{p^*} \frac{\partial p^*}{\partial x_i} \frac{\partial p'}{\partial \sigma} \right) + p^* f \epsilon_{ij3} u_j \end{aligned} \quad (3.8)$$

$$\begin{aligned} \frac{\partial(p^* w)}{\partial t} &= -m^2 \frac{\partial}{\partial x_i} \left(\frac{p^* w u_i}{m} \right) - \frac{\partial(p^* w \dot{\sigma})}{\partial \sigma} + w \text{DIV} \\ &\quad + p^* g \frac{\rho_0}{\rho} \left[\frac{1}{p^*} \frac{\partial p'}{\partial \sigma} + \frac{T'_v}{T} - \frac{T_0}{T} \frac{p'}{p_0} \right] - g (q_c + q_r) \end{aligned} \quad (3.9)$$

$$\frac{\partial(p^* T)}{\partial t} = -m^2 \frac{\partial}{\partial x_j} \left(\frac{p^* u_j T}{m} \right) - \frac{\partial(p^* \dot{\sigma} T)}{\partial \sigma} + T \text{DIV} + \frac{1}{\rho c_p} \left(p^* \frac{dp'}{dt} - \rho_0 w p^* g \right) + p^* \frac{\dot{Q}}{c_p} \quad (3.10)$$

where repeated indices represent summation from 1 to 2, γ is the ratio of specific heats, T'_v is the perturbation of virtual temperature, c_p is the specific heat at constant pressure for air, q_c and q_r are the mixing ratios for cloud and rain water respectively, \dot{Q} accounts for diabatic heating effects, and ϵ is the permutation symbol. The divergence term DIV , $\dot{\sigma}$, the Coriolis parameter, and the virtual temperature are defined as:

$$\text{DIV} = m^2 \frac{\partial}{\partial x_i} \left(\frac{p^* u_i}{m} \right) + \frac{\partial(p^* \dot{\sigma})}{\partial \sigma} \quad (3.11)$$

$$\dot{\sigma} = m u_i \frac{\partial \sigma}{\partial x_i} + w \frac{\partial \sigma}{\partial z} = -m u_i \frac{\sigma}{p^*} \frac{\partial p^*}{\partial x_i} - \frac{\rho_0 g}{p^*} \quad (3.12)$$

$$f = 2\Omega \sin \phi \quad (3.13)$$

$$T_v = T(1 + 0.608 q_v) \quad (3.14)$$

where Ω is the angular velocity of the earth, ϕ is the the latitude, and q_v is the mixing ratio for water vapor.

3.1.2 MM5 grid

Although MM5 allows multiple nested grids, it was run using only one domain. With this limitation, the MM5 grid was evenly spaced and orthogonal on its projected horizontal plane, but was stretched in the vertical direction. Since the indexing and array usage is non-standard they will be described.

As shown in figure 3.1 the i index increments the grid in the y or North-South direction and is the first dimension of two- and three-dimensional arrays. The j index increments the grid in the x or East-West direction and is the second dimension of two- and three-dimensional arrays. The k index increments the grid in the vertical direction from the top down and is the third dimension in the three-dimensional arrays of MM5. Therefore, although the MM5 coordinate system is left handed, the indices form a right handed system.

The vertical σ coordinate system ranged from zero at the top of the model to one at the surface. The vertical direction was discretized into full σ and half σ levels. All properties were stored at the half σ levels except for the vertical velocity which was stored at the full σ levels.

The horizontal domain is discretized using a type-B staggering (Dudhia, 1993 and Arakawa and Lamb, 1997) which stores all properties except velocities at the cell center referred to in the MM5 literature as cross points (Grell et al., 1995). The horizontal velocities are stored at the middle of the vertical edges of a cell at points called dot points. The vertical velocity components are stored at the cross points and at full σ levels. See figure 3.1.

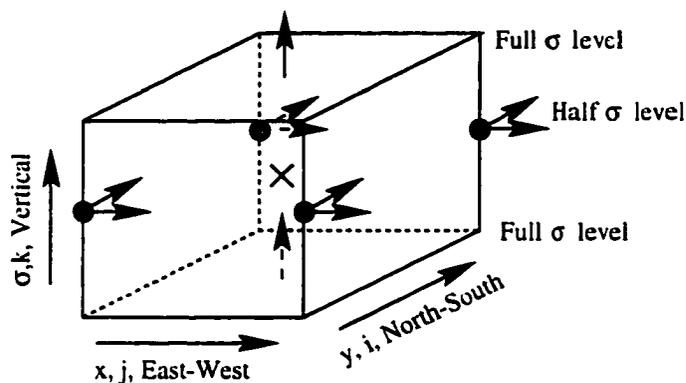


Figure 3.1 Property storage locations for the MM5 modeling system. The \times 's represent the cross points at half σ levels. The coincident arrows (\rightarrow and \nearrow) originate at the dot points and half σ levels, and represent the horizontal velocity components in MM5. The \uparrow 's represents the vertical velocity component at the full σ levels and cross points.

The MM5 grid had one more dot point in each direction than cross point. Likewise, in the vertical direction there was one more full σ level than half σ level. A rather confusing point when specifying the MM5 domains or reading the MM5 documentation is that the horizontal size of the domain was specified by the number of dot points in each direction but the vertical resolution is specified by the number of half σ layers in the vertical direction. Thus, in the horizontal direction the number of cell

boundaries was specified but in the vertical direction the number of cell centers was input.

3.1.3 Boundary conditions

Physically realistic initial and boundary conditions are a complicating factor in modeling the atmosphere. Although high resolution spatial data are available at the surface, upper atmospheric conditions are needed to initialize the model. Temporally varying data both at the surface and in the upper atmosphere are needed to determine the boundary conditions. One problem of using a limited area model like MM5 is that potential inconsistencies between the boundary conditions and the computed interior solutions can cause instabilities and/or discontinuities to form. Therefore, MM5 uses a method of transitioning from the actual boundary condition to the computed values near the boundary. These boundary conditions are discussed in section 3.1.3.1. The NCEP/NCAR 40 year reanalysis data set, discussed in section 3.1.3.2, was used to determine the state of the atmosphere initially over the entire domain and at the boundaries for each time step, thus forming the boundary conditions.

3.1.3.1 Boundary condition implementation within MM5

The boundary conditions used in the MM5 non-hydrostatic version were referred to as relaxation or nudging boundary conditions. The implementation of this type of boundary condition not only includes specifying the variable at the boundary points but also alters the value at interior points within a certain number of points near the boundary. This band of points is referred to as a forcing frame (Grell et al., 1995). The rate of change or tendency of a variable α was prescribed as:

$$\frac{\partial \alpha}{\partial t} = F(n) [F_1 (\alpha_S - \alpha_M) - F_2 (\alpha_S - \alpha_M)] \quad (3.15)$$

where $F(n)$ is a linear function of the number of grid points from the boundary:

$$F(n) = \begin{cases} (\frac{5-n}{3}) & \text{if } n = 2, 3, 4 \\ 0 & \text{if } n > 4 \end{cases} \quad (3.16)$$

where n represents the number of rows from the boundary. F_1 and F_2 are constants:

$$F_1 = \frac{1}{10\Delta t} \quad (3.17)$$

$$F_2 = \frac{\Delta x^2}{50\Delta t} \quad (3.18)$$

which account for damping and diffusion at the boundaries. α_S is the large scale, specified values of variable α , and α_M is the calculated value of α at the previous time step.

This method was applied to all of the variables except for the vertical velocity w . It is not nudged, but rather given a value of zero with zero gradient at the outer edges of the domain.

Temporally varying input data at the frequency required to update the boundary conditions at each time step is generally unavailable. Therefore, MM5 uses boundary condition information at a much lower frequency. For the cases run in this study, the frequency is every six hours. Intermediate times were linearly interpolated from the values at the six-hour intervals.

3.1.3.2 Boundary condition data

The National Centers for Environmental Prediction (NCEP) and the National Center for Atmospheric Research (NCAR) began a project in 1991 to establish a record of global analyses of atmospheric fields from 1957 through 1996 (Kalanay et al., 1996). This project produces a consistent data set that can be used for climate studies as well as provide model initialization without inconsistencies due to changes in reporting practices, data assimilation techniques, and instrumentation.

The NCEP/NCAR reanalysis data used the NCEP global spectral model with four-dimensional data assimilation in order to produce consistent fields. The model used 28 vertical σ levels and had a T62 truncation which roughly corresponds to a horizontal resolution of 210 km. This grid spacing is 7 times the nominal grid spacing used in MM5. Therefore, the reanalysis data provided excessively smooth initial and boundary conditions. Although this was the case it is shown in section 3.4.3 that the internal dynamics of MM5 and the DGAC can produce higher resolution features.

3.2 Dynamic grid advection component

The dynamic grid advection component (DGAC), was written to compute the tracer advection using a dynamic grid. Since the MM5 model equations use the σ coordinate in the vertical direction, have map factors included in the horizontal directions, and are normalized by p^* , it was necessary to cast the advection equation into a form that was compatible with the rest of the MM5 equations and was applicable to a moving grid. This development is contained in section 3.2.1. The development of an MPDATA scheme that solves this advection equation is in section 3.2.2. The complexity of the code was compounded by the different types and resolutions of the grids. The correspondence between the grids and a few details of how the new subroutines were designed is given in section 3.2.3.

3.2.1 An MM5 compatible advection equation

The transport equation appropriate for atmospheric flows on a projected plane is:

$$\frac{\partial(\rho q)}{\partial t} + m^2 \frac{\partial}{\partial x_i} \left(\frac{u_i \rho q}{m} \right) + \frac{\partial}{\partial z} (w \rho q) = 0 \quad (3.19)$$

where x_i ($i = 1, 2$) is the advection in the horizontal x and y directions respectively. Summation over the indices is assumed. q is the intensive property being advected, ρ is the fluid density, m is the map factor which is a metric term that accounts for the projection of the points in a spherical coordinate system to Cartesian coordinates, and z is the vertical direction. Expanding 3.19 using the chain rule:

$$\rho \left(\frac{\partial q}{\partial t} + m u_i \frac{\partial q}{\partial x_i} + w \frac{\partial q}{\partial z} \right) + q \left[\frac{\partial \rho}{\partial t} + m^2 \frac{\partial}{\partial x_i} \left(\frac{\rho u_i}{m} \right) + \frac{\partial(\rho w)}{\partial z} \right] = 0 \quad (3.20)$$

The terms in square brackets constitute the continuity equation and thus are identically zero. Dividing the remaining terms by ρ :

$$\frac{\partial q}{\partial t} + m u_i \frac{\partial q}{\partial x_i} + w \frac{\partial q}{\partial z} = 0 \quad (3.21)$$

Transforming this into terrain following σ coordinates:

$$\frac{\partial \phi}{\partial t} + m u_i \frac{\partial q}{\partial x_i} + m u_i \frac{\partial q}{\partial \sigma} \frac{\partial \sigma}{\partial x_i} + w \frac{\partial q}{\partial \sigma} \frac{\partial \sigma}{\partial z} = 0 \quad (3.22)$$

where σ is defined to be the same as in equation 3.5:

$$\sigma = \frac{p_0 - p_t}{p_s - p_t} = \frac{p_0 - p_t}{p^*} \quad (3.23)$$

Recall that p_0 is the basic state hydrostatic pressure, p_t is the basic state pressure at the top of the model, p_s is the surface basic state pressure, and $p^* = p_s - p_t$. Since p^* is a function of x_i only and p_0 is a function of z only, $\frac{\partial \sigma}{\partial x_i}$, $\frac{\partial \sigma}{\partial z}$, and the velocity in the new vertical coordinate system are

$$\frac{\partial \sigma}{\partial x_i} = -\frac{\sigma}{p^*} \frac{\partial p^*}{\partial x_i} \quad (3.24)$$

$$\frac{\partial \sigma}{\partial z} = \frac{1}{p^*} \frac{\partial p_0}{\partial z} = -\frac{\rho_0 g}{p^*} \quad (3.25)$$

$$\begin{aligned} \dot{\sigma} &= \frac{d\sigma}{dt} \\ &= m u_i \frac{\partial \sigma}{\partial x_i} + w \frac{\partial \sigma}{\partial z} \\ &= -\frac{m \sigma u_i}{p^*} \frac{\partial p^*}{\partial x_i} - \frac{\rho_0 g w}{p^*} \end{aligned} \quad (3.26)$$

where ρ_0 is the basic state density. Note that p^* is not a function of physical time. Substituting these into equation 3.22 and multiplying through by p^* yields

$$p^* \frac{\partial q}{\partial t} + m p^* u_i \frac{\partial q}{\partial x_i} + p^* \dot{\sigma} \frac{\partial q}{\partial \sigma} = 0 \quad (3.27)$$

Assuming an uneven, moving grid and transforming equation 3.27 into an orthogonal computational domain with integer grid spacing:

$$p^* \frac{\partial q}{\partial \tau} + p^* \hat{u}_i \frac{\partial q}{\partial \xi_i} = 0 \quad (3.28)$$

where the contravariant velocity \hat{u}_i is

$$\hat{u}_i = \frac{\partial \xi_i}{\partial t} + m u_j \frac{\partial \xi_i}{\partial x_j} + \dot{\sigma} \frac{\partial \xi_i}{\partial \sigma} \quad (3.29)$$

Putting equation 3.28 into flux form:

$$\frac{\partial(p^* q)}{\partial \tau} + \frac{\partial(p^* \hat{u}_i q)}{\partial \xi_i} = q \left(\frac{\partial p^*}{\partial \tau} + \frac{\partial(p^* \hat{u}_i)}{\partial \xi_i} \right) \quad (3.30)$$

Although p^* is not a function of physical time t , it is a function of time τ in the computational domain. Intuitively, as the physical grid points move, p^* at a given grid point changes because it is a function of the horizontal direction. Hence, in the computational domain p^* is a function of computational time.

3.2.2 An MM5 compatible MPDATA scheme

This development is similar to that presented in section 2.2.2.2 except that the divergence term has a different form and the correction term has a dependence on $\frac{\partial p^*}{\partial \tau}$. As before it is convenient to use equations 2.7 to interchange the dependent and independent variables in the contravariant velocity components.

In order to derive the temporal part of the correction terms to be used in the MPDATA scheme, equation 3.30 is written as

$$\frac{((p^* q)^{n+1} - (p^* q)^n)}{\Delta \tau} + \frac{\partial}{\partial \xi_i} \left((p^* q)^n \hat{u}_i^{n+1/2} \right) = R^{n+1/2} \quad (3.31)$$

where n represents the time level and

$$R = q \left(\frac{\partial p^*}{\partial \tau} + \frac{\partial(p^* \hat{u}_i)}{\partial \xi_i} \right) \quad (3.32)$$

Expanding equation 3.31 using Taylor series

$$\frac{\partial(p^* q)}{\partial \tau} + \frac{1}{2} \Delta \tau \frac{\partial^2(p^* q)}{\partial \tau^2} + \frac{\partial}{\partial \xi_i} \left[p^* q \left(\hat{u}_i + \frac{1}{2} \Delta \tau \frac{\partial \hat{u}_i}{\partial \tau} \right) \right] = R + \frac{1}{2} \Delta \tau \frac{\partial R}{\partial \tau} + \mathcal{O}(\Delta \tau^2) \quad (3.33)$$

Considering only the lowest order terms and differentiating equation 3.33

$$\frac{\partial^2(p^* q)}{\partial \tau^2} = \frac{\partial R}{\partial \tau} - \frac{\partial}{\partial \xi_i} \left(\hat{u}_i \frac{\partial(p^* q)}{\partial \tau} + p^* q \frac{\partial \hat{u}_i}{\partial \tau} \right) + \mathcal{O}(\Delta \tau) \quad (3.34)$$

If only the first order terms of equation 3.33 are considered and it is solved for $\frac{\partial(p^* q)}{\partial \tau}$ and substituted into equation 3.34:

$$\frac{\partial^2(p^* q)}{\partial \tau^2} = \frac{\partial R}{\partial \tau} - \frac{\partial}{\partial \xi_i} \left\{ \hat{u}_i \left[R - \frac{\partial(p^* q \hat{u}_j)}{\partial \xi_j} \right] + p^* q \frac{\partial \hat{u}_i}{\partial \tau} \right\} + \mathcal{O}(\Delta \tau) \quad (3.35)$$

Substituting 3.35 into 3.33 and canceling terms

$$\frac{\partial(p^* q)}{\partial \tau} + \frac{\partial(p^* \hat{u}_i q)}{\partial \xi_i} = R - \frac{1}{2} \Delta \tau \frac{\partial}{\partial \xi_i} \left(p^* \hat{u}_i \hat{u}_j \frac{\partial q}{\partial \xi_j} + \hat{u}_i q \frac{\partial(p^* \hat{u}_j)}{\partial \xi_j} - \hat{u}_i R \right) + \mathcal{O}(\Delta \tau^2) \quad (3.36)$$

If R is substituted, two terms cancel giving

$$\frac{\partial(p^* q)}{\partial \tau} + \frac{\partial(p^* \hat{u}_i q)}{\partial \xi_i} = R - \frac{1}{2} \Delta \tau \frac{\partial}{\partial \xi_i} \left(p^* \hat{u}_i \hat{u}_j \frac{\partial q}{\partial \xi_j} - \hat{u}_i q \frac{\partial p^*}{\partial \tau} \right) + \mathcal{O}(\Delta \tau^2) \quad (3.37)$$

The first correction term in 3.37 is the same temporal correction term derived in equation 2.45 multiplied by p^* . The new correction term accounts for the change in p^* as the grid points move in the horizontal direction.

In the derivation of the spatial correction terms there is a choice of which variable, q or \hat{u}_i , to associate p^* . If it is grouped with the q , the spatial correction terms have the form $|\hat{u}_i| \frac{\partial(p^* q)}{\partial \xi_i}$, whereas if p^* is linked with \hat{u}_i , it has the form $p^* |\hat{u}_i| \frac{\partial q}{\partial \xi_i}$. Since the temporal correction terms have a form similar to the result of the second choice, it was decided to associate p^* with \hat{u}_i . The spatial terms will only be considered in a one-dimensional case, since no new information comes from examining them in a multidimensional one. Consider the donor-cell step of the advection term in equation 3.30.

$$\frac{\partial(p^* \hat{u} q)}{\partial \xi} \approx (p^* \hat{u})_{i+1/2} q_i - (p^* \hat{u})_{i-1/2} q_{i-1} \quad (3.38)$$

where \hat{u} has been assumed to be positive. If the terms on the right hand side are expanded in Taylor series expansions and simplified:

$$\frac{\partial(p^* \hat{u} q)}{\partial \xi} = \frac{\partial(p^* \hat{u} q)}{\partial \xi} - \frac{1}{2} \Delta \xi \frac{\partial}{\partial \xi} \left(p^* \hat{u} \frac{\partial q}{\partial \xi} \right) + \mathcal{O}(\Delta \xi^2) \quad (3.39)$$

If \hat{u} is assumed to be negative and noting that p^* is always positive, the result can be combined with equation 3.39 to yield:

$$\frac{\partial(p^* \hat{u} q)}{\partial \xi} = \frac{\partial(p^* \hat{u} q)}{\partial \xi} - \frac{1}{2} \Delta \xi \frac{\partial}{\partial \xi} \left(p^* |\hat{u}| \frac{\partial q}{\partial \xi} \right) + \mathcal{O}(\Delta \xi^2) \quad (3.40)$$

which is the same spatial correction term derived in equation 2.47 multiplied by p^* . Extending this to the three-dimensional case and combining it with equation 3.37 yields:

$$\begin{aligned} \frac{\partial(p^* q)}{\partial \tau} + \sum_{i=1}^3 \frac{\partial(p^* \hat{u}_i q)}{\partial \xi_i} = R - \frac{1}{2} \sum_{i=1}^3 \frac{\partial}{\partial \xi_i} \left[\Delta \tau \left(p^* \hat{u}_i \sum_{j=1}^3 \hat{u}_j \frac{\partial q}{\partial \xi_j} - \hat{u}_i q \frac{\partial p^*}{\partial \tau} \right) \right. \\ \left. - \Delta \xi_i p^* |\hat{u}_i| \frac{\partial q}{\partial \xi_i} \right] + \mathcal{O}(\Delta \tau^2, \Delta \xi_i^2) \end{aligned} \quad (3.41)$$

where the only summations are those explicitly indicated by \sum . Only one new term appears in the expression of the higher order correction terms. This is the $\frac{\partial p^*}{\partial \tau}$ term.

If the velocities are non-dimensionalized using the computational grid spacing and the time step as $\hat{u}_i = \frac{\Delta \xi_i}{\Delta \tau} \hat{U}_i$ where \hat{U}_i are the contravariant non-dimensional velocities, the correction terms become:

$$\frac{1}{2} \sum_{i=1}^3 \frac{\partial}{\partial \xi_i} \left\{ \frac{\Delta \xi_i^2}{\Delta \tau} \left(|p^* \hat{U}_i| - p^* \hat{U}_i^2 \right) \frac{\partial q}{\partial \xi_i} - \sum_{j=1}^3 p^* \frac{\Delta \xi_i \Delta \xi_j}{\Delta \tau} \hat{U}_i \hat{U}_j \frac{\partial q}{\partial \xi_j} (1 - \delta_{ij}) + \Delta \xi_i \hat{U}_i q \frac{\partial p^*}{\partial \tau} \right\} \quad (3.42)$$

where δ_{ij} is the Kronecker delta. Defining the pseudo velocities as:

$$p^* \hat{u}_i^{(1)} = \frac{\Delta \xi_i^2}{2 \Delta \tau} \left(|p^* \hat{U}_i| - p^* \hat{U}_i^2 \right) \frac{1}{q} \frac{\partial q}{\partial \xi_i} - \sum_{j=1}^3 \frac{p^* \Delta \xi_i \Delta \xi_j}{2 \Delta \tau} \hat{U}_i \hat{U}_j \frac{1}{q} \frac{\partial q}{\partial \xi_j} (1 - \delta_{ij}) + \frac{1}{2} \Delta \xi_i \hat{U}_i \frac{\partial p^*}{\partial \tau} \quad (3.43)$$

Defining a non-dimensional pseudo velocity such that $\hat{u}_i^{(k)} = \frac{\Delta \xi_i}{\Delta \tau} \hat{U}_i^{(k)}$:

$$p^* \hat{U}_i^{(1)} = \frac{1}{2} \Delta \xi_i \left(|p^* \hat{U}_i| - p^* \hat{U}_i^2 \right) \frac{1}{q} \frac{\partial q}{\partial \xi_i} - \frac{1}{2} \sum_{j=1}^3 \Delta \xi_j p^* \hat{U}_i \hat{U}_j \frac{1}{q} \frac{\partial q}{\partial \xi_j} (1 - \delta_{ij}) + \frac{1}{2} \Delta \tau \hat{U}_i \frac{\partial p^*}{\partial \tau} \quad (3.44)$$

Since $\Delta \xi_i = 1$ these become:

$$p^* \hat{U}_i^{(1)} = \frac{1}{2} \left(|p^* \hat{U}_i| - p^* \hat{U}_i^2 \right) \frac{1}{q} \frac{\partial q}{\partial \xi_i} - \frac{1}{2} \sum_{j=1}^3 p^* \hat{U}_i \hat{U}_j \frac{1}{q} \frac{\partial q}{\partial \xi_j} (1 - \delta_{ij}) + \frac{1}{2} \Delta \tau \hat{U}_i \frac{\partial p^*}{\partial \tau} \quad (3.45)$$

Dividing through by p^* :

$$\hat{U}_i^{(1)} = \frac{1}{2} \left(|\hat{U}_i| - \hat{U}_i^2 \right) \frac{1}{q} \frac{\partial q}{\partial \xi_i} - \frac{1}{2} \sum_{j=1}^3 \hat{U}_i \hat{U}_j \frac{1}{q} \frac{\partial q}{\partial \xi_j} (1 - \delta_{ij}) + \frac{1}{2} \Delta \tau \frac{\hat{U}_i}{p^*} \frac{\partial p^*}{\partial \tau} \quad (3.46)$$

This can be generalized to multiple iterations:

$$\hat{U}_i^{(k+1)} = \frac{1}{2} \left(|\hat{U}_i^{(k)}| - \hat{U}_i^{(k)} \hat{U}_i^{(k)} \right) \frac{1}{q} \frac{\partial q}{\partial \xi_i} - \frac{1}{2} \sum_{j=1}^3 \hat{U}_i^{(k)} \hat{U}_j^{(k)} \frac{1}{q} \frac{\partial q}{\partial \xi_j} (1 - \delta_{ij}) + \frac{1}{2} \Delta \tau \frac{\hat{U}_i^{(k)}}{p^*} \frac{\partial p^*}{\partial \tau} \text{ where } k = 0 \dots K \quad (3.47)$$

Recall that $\hat{U}^{(0)} = \hat{U}$, the contravariant velocity.

3.2.3 The dynamic grid advection component grid and domain correspondence with MM5

Both MM5 and the DGAC use staggered grids. However, the staggered grids do not coincide because the DGAC grid is staggered differently, can have a different number of grid points than the MM5 grid, and can be stretched. However, since both of the grids discretize the same physical domain, a correspondence between the two grids is required. The MM5 grid has been described in section 3.1.2. The DGAC grid is described next in section 3.2.3.1. In section 3.2.3.2, the correspondence between the two grids is explained.

3.2.3.1 The dynamic grid advection component grid

Due to the different methods of staggering the grids, locations on the DGAC grid are given different names to differentiate them from points on the MM5 grid. Cell center and corners refer to the three-dimensional center and corners of a cell on the DGAC grid. The cell faces refer to the center of the cell faces. For the purpose of describing the storage locations of the properties on the DGAC grid, the cell center locations will be designated using integer (i) values while the cell faces will be designated using half integer values ($i + 1/2$). In the DGAC the tracer concentration was stored at the cell centers while the three velocity components (u, v, w) were stored at the cell faces ($i + 1/2, j, k$), ($i, j + 1/2, k$), and ($i, j, k + 1/2$) respectively. Figure 3.2 illustrates the locations of these points and the storage locations for the different velocity components for an orthogonal cell.

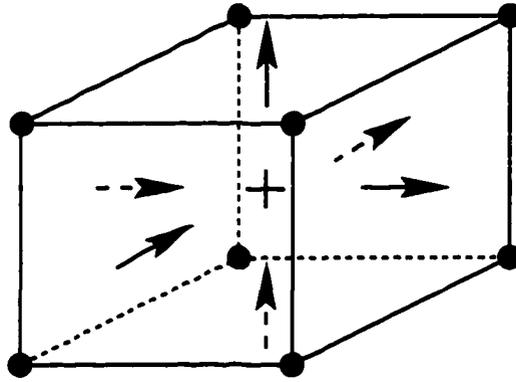


Figure 3.2 Property storage locations for the MPDATA scheme. The cell center is marked with a +. The corner points are marked with the •'s. The →'s, ↗'s and ↑'s mark the $i + 1/2$, $j + 1/2$ and $k + 1/2$ cell face centers respectively.

Although not implemented in the DGAC, physical parameterizations for radiation and cumulus convection are one-dimensional vertical models, and conceivably could be implemented in the future. For these reasons, the horizontal stretching of the grid in the DGAC was chosen not to be a function of height. This ensured the alignment of the grid with the vertical direction and will, in the future, permit the parameterization schemes to be more easily implemented on the stretched grid. This choice also has the considerable benefit of greatly reducing the computational cost of generating the three-dimensional stretched grid and metric stretching terms for the transformation to a computational domain.

When using staggered grids, a choice has to be made as to which type of point, a cell face or cell center, will be at the boundaries of the domain. Unlike the MM5 grid, the DGAC grid has cell centers

at the boundaries as illustrated in figure 3.3. The type of inflow boundary conditions implemented in the DGAC required a value to be upstream of the velocity component, but MM5 implemented the boundary conditions differently and therefore planes of cell boundaries could be at the exterior of the domain.

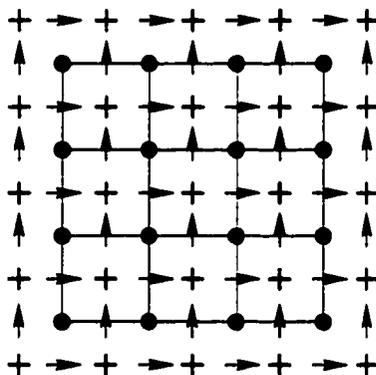


Figure 3.3 Storage locations for the DGAC grid. The + represent corner points, • represent cell center points, → and ↑ represent $i + 1/2$ and $j + 1/2$ cell face center locations respectively.

The pseudo velocities for the MPDATA scheme required all three velocity components at the cell faces. In order to get an estimate of the other two velocity components v and w at location $i + 1/2$ the four values in the same plane but at $(i, j - 1/2)$, $(i, j + 1/2)$, $(i + 1, j - 1/2)$ and $(i + 1, j + 1/2)$ were averaged together as recommended by Smolarkiewicz (1984). See figure 3.4 for an illustration of the stencils to calculate the \hat{U}_i pseudo velocity for a cell. At the boundaries v and w velocities outside of the boundary defined by the cell corners are needed in order to calculate these averages. These velocities were stored at locations midway between the additional center points that were needed for boundary conditions. This accounts for the extra rows of →'s and columns of ↑'s outside the boundaries determined by the corner points (•'s) in figure 3.3.

Unlike the MM5 modeling system, left handed coordinate systems were used for both the coordinate system and the indices. Therefore, within the DGAC the i , j and k indices correspond to the x , y and σ directions respectively.

3.2.3.2 Correspondence between the MM5 and DGAC grids

As mentioned in the introduction to this section, the resolution of the DGAC grid is not a function of the MM5 grid resolution. However, since both of the grids discretize the same physical domain there

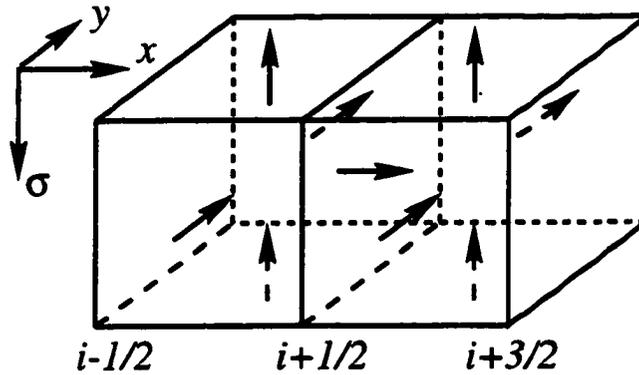


Figure 3.4 V and W velocity stencils needed to calculate the $\tilde{U}^{(k)}$ pseudo velocity at $i+1/2$. The \rightarrow , \nearrow 's and the \uparrow 's represent the non-dimensional velocity components $\tilde{U}^{(k)}$, $\tilde{V}^{(k)}$ and $\tilde{W}^{(k)}$ respectively.

must be a correspondence between the two grids. The horizontal and vertical boundaries were treated slightly differently since the horizontal boundaries had to account for inflow and outflow conditions while the upper and lower boundaries corresponding to the top of the atmosphere and the surface respectively did not permit mass to cross them.

3.2.3.2.1 Horizontal correspondence: The horizontal boundaries were designed to account for the possible inflow of the tracer. The MPDATA scheme requires that an upstream value be present in this case. This requires that the cell center points be the outer most points of the grid. However, on the MM5 grid the dot points (corresponding to the storage locations of the velocity components) are the bounding points. In order to facilitate the interpolation between the grids the DGAC bounding corner points were chosen to lie in planes formed by the MM5 dot points that are set one grid space in from the boundary. This can be seen in figure 3.5 where the intersection of the solid lines represent the DGAC corner points and the filled circles (\bullet) represent the MM5 dot points. During the generation of the dynamic grid these outer corner points were allowed to adjust inside but not outside of the outer planes formed by the MM5 dot points.

Once the DGAC corner points were determined, the DGAC cell centers were calculated. They are represented in figure 3.5 by the open circles (\circ). With the exception of the outer most points, the cell center locations were calculated by computing the arithmetic mean of the the corner points. The bounding cell center locations were determined by reflecting the adjacent cell centers across the planes formed by the bounding DGAC corner points, subject to the constraint that the cell half width was no more than an MM5 cell half width. The standard MM5 uses a five grid buffer region around the outside

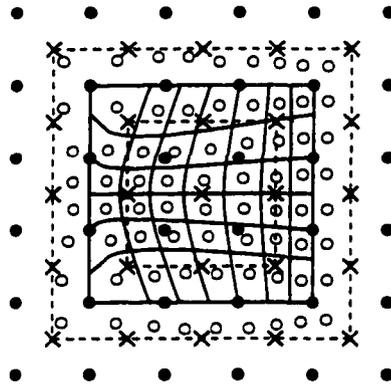


Figure 3.5 Correspondence between the physical domains for the MM5 grid and the DGAC grid. The \bullet 's represent the MM5 dot points. The \times represent the MM5 cross points. The solid lines represent the DGAC grid of corner points. The \circ 's represent the DGAC cell centers. The dashed lines represent the planes of MM5 cross points where observations are known. Note that the outer boundary is treated just as in this figure but the DGAC grid resolution could be varied. Note the DGAC grid has a higher resolution in this illustration. This was not a necessary condition.

of the domain for boundary condition relaxation as discussed in section 3.1.3, observational values were known at the outer two rows of MM5 cross points. Observational values from the two outermost MM5 cross points were used to interpolate to the outermost DGAC center points. These outer two rows of dot points are designated with the dashed lines in figure 3.5.

3.2.3.2.2 Vertical correspondence: The vertical velocities were set to zero at the top and the bottom of the model. In light of this the upper and lower cell corner points were chosen to lie in the planes formed by the upper and lower MM5 dot points respectively. This is in contrast to the horizontal boundaries where the bounding corner points were forced to align with the MM5 dot points one grid space in from the boundary. The σ vertical velocity was set to zero at the top and bottom of the model.

Although there was no possible inflow either through the bottom or the top of the model in order to make the vertical boundaries look like the horizontal boundaries, extra levels of cell center points were placed above the upper σ level and below the lower σ level. Hence, these dummy levels had σ values that were < 0 and > 1 respectively. The value that was placed at these extra levels was unimportant since they were completely isolated from the rest of the model, due to the no flux boundary condition. These extra cell center vertical locations were found by reflecting the adjacent cell centers around the planes

formed by the bounding cell corner points. This allowed the DGAC to treat the vertical direction in exactly the same way the horizontal directions were treated, with the exception that the normal velocity component on the top and bottom boundaries was set to zero.

3.2.4 Code details

MM5 was developed in FORTRAN 77 at the National Center for Atmospheric Research (NCAR) and was initially run on a 64 bit machine. Due to the size (~ 50,000 lines of FORTRAN 77 code) and complexity of MM5, modification of the code was kept to a minimum. This was done for the following reasons:

1. During development of the DGAC, it was run in isolation from MM5 so that computational overhead of running MM5 was avoided.
2. The introduction of inadvertent errors to MM5 was kept to a minimum.
3. The restrictions of adopting MM5 coding paradigms were lifted.

This section describes the DGAC linkage to MM5. The DGAC initialization, the updating of the velocity fields at each time step, and the application of the boundary conditions to the DGAC are discussed in sections 3.2.4.1, 3.2.4.2, and 3.2.4.3 respectively. Additionally, appendix D discusses some numerical aspects of converting MM5 from the CRAY environment, where it was developed, to the DEC Alpha workstations, where it was run, and the means of isolating the DGAC from MM5.

3.2.4.1 DGAC initialization

A number of DGAC calculations were performed prior to the first time step of MM5. These consisted of initializing values that were required from previous time steps, initializing values that remained constant throughout the integration, and determining the appropriate time step. Values that were required from a previous time step were the grid point positions, the corner velocities and the vertical pressure difference p^* at the cell face centers and, of course, the tracer concentration values at the DGAC cell center locations. In the physical domain, the vertical pressure difference p^* were constant with respect to physical time, and therefore, the DGAC was initialized with its value at the MM5 dot points. This eliminated the need to pass these values to the DGAC each time step.

The initial DGAC grid point positions were determined by ten grid-refinement iterations. Each iteration consisted of interpolating the values of the tracer concentration from the the MM5 cross points to the the DGAC corner points and generating a refined grid. The monotonicity feature of

the interpolator was required to obtain a stable grid. If the monotonicity feature was not used, the high second derivatives associated with spurious oscillations caused the grid generator to cluster points around these wiggles. Since the grid was refined in this area, the oscillations would disappear in the next interpolation and the grid would then cluster on either the actual high second derivatives or newly created ripples at different locations. Not only did this make it impossible to create a stable initial grid, but it caused the grid to snap to a new location at the first time step. The resulting high grid speeds violated the CFL condition.

The use of the velocity at the half time step in the MPDATA scheme necessitates the newly computed MM5 velocity u_i^{n+1} and the velocity from the previous time step u_i^n . In order to minimize the amount of information that is passed from MM5 to the DGAC and more importantly the amount of interpolation required, the new velocity u_i^{n+1} and grid point location x_i^{n+1} were saved to be used as u_i^n in the subsequent time step. The vertical pressure difference was treated in a similar way.

The tracer concentration values were computed within the DGAC, therefore its value was stored in a global variable. However, in anticipation of using water vapor mixing ratio as the tracer, it was initialized from the initial tracer concentration values that were added to MM5. This interpolation took place in conjunction with the initial grid refinement described earlier in this section.

The vertical pressure difference p^* was constant with time in the physical domain. The actual variables for horizontal velocity components that were used by MM5 were p^*u and p^*v which were stored at dot points. Since u and v were required for the DGAC and were interpolated to the corner points, it was advantageous to store the temporally constant p^* value at the dot points. Within MM5, p^* was stored at the cross points, and therefore the same method of linear interpolation used in MM5 to compute the p^* at the dot points was used to initialize p^* at the dot points in the DGAC. As a result, these values did not have to be passed from MM5 to the DGAC each time step.

The small grid spacing in the DGAC potentially could limit the appropriate time step size that could be used in the DGAC. Dudhia et al. (1997) recommend that the time step for MM5 be set at 333 m/s. Since the wind speeds were significantly slower than this it was possible to take a single time step with the dynamic MPDATA scheme and not violate the CFL condition, even though in some cases the grid spacing was much smaller than that of MM5.

3.2.4.2 DGAC updating

The correct vertical velocity to use when performing advection calculations with the σ coordinate system is $\dot{\sigma}$. However, the MM5 prognostic vertical velocity variable was w . Therefore w^{n+1} is calculated

at each time step and is known at the end of the time step. σ^{n+1} on the other hand is not calculated until the $n+2$ time step has begun. Therefore, the DGAC was called immediately after σ was calculated in the subroutine `SOLVE3.f` which solves the non-hydrostatic prognostic equations.

The MM5 p^*u , p^*v and σ quantities were passed to the DGAC each time step. Initially, a new grid was generated based on the tracer concentration q^n . p^* was divided out of the p^*u and p^*v quantities to yield the velocity components. The velocities were interpolated from the MM5 velocity storage locations (see figure 3.2) to the DGAC cell corners. The contravariant velocities were calculated at the DGAC cell corners and non-dimensionalized by the time step (recall that the computational grid spacing was one).

At each time step, the contravariant velocities were calculated at the corner points and then interpolated to the DGAC face center locations. The tracer concentrations at the cell boundaries were determined as described in section 3.2.4.2. Finally, the second order MPDATA scheme was used to advance the solution for the tracer concentration forward in time.

3.2.4.3 Boundary conditions

The MM5 boundary conditions were implemented using relaxation boundary conditions (Grell et al., 1995) as described in section 3.1.3. This type of boundary conditions was not used in the DGAC component. As described in section 3.2.3.2.1, the coordination between the DGAC grid and the MM5 grid ensured that the boundary cell center points were contained within the MM5 cross points and could be interpolated from the MM5 boundary information to provide information for inflow boundary cases.

Inflow or outflow boundaries were determined by the sign of the physical velocity at the boundary cell faces. Actually the physical velocity and the contravariant velocities were the same at these points, because they were stationary.

Recall from section 3.1.3.1 that the temporally varying boundary conditions were updated every six hours. Values at intermediate times were determined by linear interpolation. Therefore, the value at a boundary cross point was determined by:

$$q_i^{l+n} = q_i^l + \frac{\partial q_i}{\partial t} n \Delta t \quad (3.48)$$

where the i specifies the location on the boundary, l is the time the last boundary condition was read from the file, the $\frac{\partial q_i}{\partial t}$ is the rate at which the value changes with respect to time and $n \Delta t$ is the time since the last boundary condition update. Use of equation 3.48 was important for inflow boundary conditions.

In the case of the outflow boundary, acceptable results were obtained using elementary linear extrapolation (as for an evenly spaced, orthogonal grid). The stencil used was

$$q_i = 2q_{i\pm 1} - q_{i\pm 2} \quad (3.49)$$

where the i is at the boundary, the $+$ is used for the left and front (West and South) boundaries and the $-$ is used for the right and back boundaries (East and North). If the normal derivative was strictly enforced, a two-dimensional Laplace equation needed to be solved. Results of using the more complicated technique were no better than using the simpler one.

3.3 Interpolation

The computation of tracer advection within the dynamic grid advection component (DGAC) takes place on a separate grid from the other prognostic variables calculated by MM5, so it becomes necessary to interpolate from the MM5 grid to the DGAC grid. The Smolarkiewicz and Grell (1992) interpolator used for this purpose, recasts the problem as a constant velocity Eulerian advection problem. This problem was computationally solved using schemes developed by Tremback et al. (1987) and has a monotone option based on the flux corrected transport of Boris and Book (1973). This interpolation scheme has high (up to 6th) order accuracy and the ability to extrapolate small distances (up to one grid increment) beyond the given domain. The theory of the interpolator, implementation of flux corrected transport which underlies the monotonicity capability, and application to integer space, as defined in section 2.1, is explained in appendix C.

Figure 3.6 shows the results of a test case to demonstrate the characteristics of the interpolator. A sine function was discretized on equally spaced grid points, and then interpolated to the same number of points which were perturbed from their equal distribution locations. Panel A shows the results using 51 grid points. Panel B shows how the RMS error varies with the number of data points for the second-order interpolator and a linear interpolator. Only the second-order interpolation scheme is used due to the high computational cost of the interpolation. Interpolation accounts for 83.5% of the computational cost of computing passive advection with the DGAC.

Since the grid in the dynamic model component is based partially on the curvature of the tracer field, if the interpolated field causes spurious oscillations then dramatic and unwanted redistribution of grid points about these oscillations can occur. The redistribution can be great, resulting in excessive grid speed that violates the CFL condition. These spurious oscillations were avoided by using the monotone option of the interpolator. Figure 3.7 shows the results of a non-monotone and a monotone

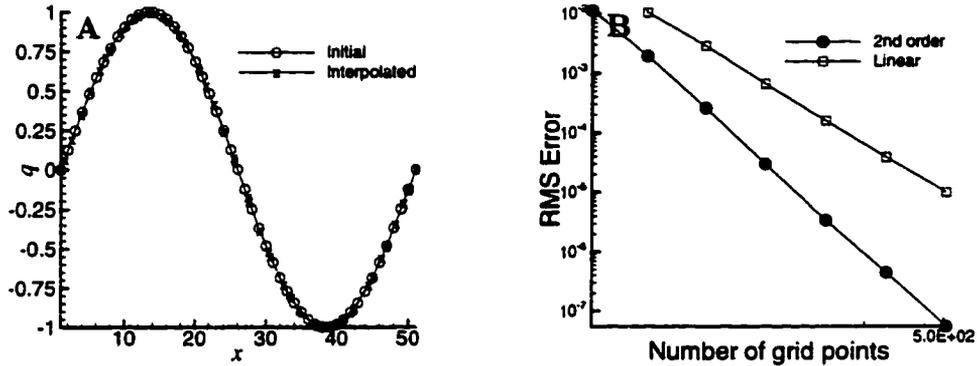


Figure 3.6 Performance of the second-order interpolator. A) Results using 51 points; B) RMS of the second-order interpolator compared to a linear interpolator

interpolation when the number of data points at which a sine function was defined was set to seven, but was interpolated to 51 equally spaced grid points. Note that the monotonicity option has prevented new extrema from developing. Although the non-monotone interpolation has included the maximum and minimum of the sine function which might be desirable for this specific case, in general new extrema lead to numerical oscillations and instability.

In this same figure, discontinuities in the solution are evident at the half integer locations. This is due to the second-order Eulerian solver that is used. If a grid point is just left of $x_{i+1/2}$ the values at x_{i-1} , x_i , and x_{i+1} are used to determine the interpolated value. However, if the interpolation point is just right of $x_{i+1/2}$, the values at grid points x_i , x_{i+1} , and x_{i+2} are used. This sudden shift in the stencil as the interpolated point crosses the $x_{i+1/2}$ location causes this discontinuity. However, under conditions necessary for truncation error of the advection to be small, these discontinuities are negligible.

The specific applications of this interpolation scheme are presented in the rest of this section. The mapping of grid points from physical space to integer space for use in with the MM5-DGAC coupling is described in section 3.3.1. Due to excessively high computational cost, specialized two- and three-dimensional interpolators were developed and are discussed in section 3.3.2.

3.3.1 Mapping of grid points into integer space

MM5 uses physical distances in the horizontal direction and σ coordinates in the vertical direction, but the interpolator was developed with the assumption that the grid was in integer space. Therefore,

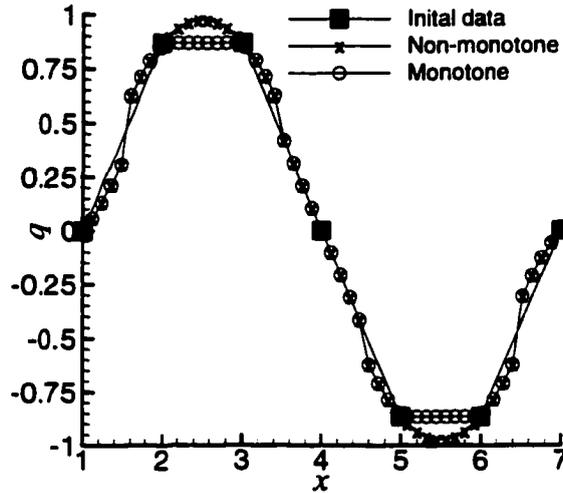


Figure 3.7 Interpolation from seven points to 50 points with and without the monotonicity feature. The discontinuous nature at the half integer locations is evident.

the MM5 domain was transformed into integer space before the interpolation was performed. This was accomplished by determining the metrics of the transformation, determining the nearest MM5 grid point, and then using a Taylor series expansion to determine the value of the interpolation point in integer space.

Initially the metric terms were calculated from the MM5 domain to integer space. Since the MM5 grid was orthogonal, the only non-zero metric terms were $\frac{\partial x}{\partial \xi}$, $\frac{\partial y}{\partial \eta}$, and $\frac{\partial z}{\partial \zeta}$. Additionally, the MM5 grid was evenly spaced in the horizontal directions and therefore $\frac{\partial x}{\partial \xi}$ and $\frac{\partial y}{\partial \eta}$ were constants. $\frac{\partial z}{\partial \zeta}$ was determined from a second-order Taylor expansion.

The most difficult part of this procedure was determining the location of the nearest neighbor. Since the horizontal direction was evenly spaced the horizontal location was found using

$$\hat{\xi} = (\hat{x} - x_1) \frac{\partial \bar{\xi}}{\partial x} + 1 \quad (3.50)$$

where \hat{x} is the x coordinate of the interpolation point in physical space, $\hat{\xi}$ is the interpolation point in integer space, and x_1 is the left most x coordinate. The same method was used to find the interpolation points in the north-south direction.

In the vertical direction the nearest point was found by searching through the array of σ coordinates to find the nearest value. Once the nearest neighbor was found, the coordinate in integer space was

found using

$$\hat{\zeta} = (\hat{z} - z_n) \frac{\partial \bar{\zeta}}{\partial z} + n \quad (3.51)$$

where \hat{z} is the vertical coordinate of the interpolation point, z_n is the vertical coordinate of the point nearest to \hat{z} , $\frac{\partial \bar{\zeta}}{\partial z}$ is evaluated at the point z_n , n is the index of the z_n , and $\hat{\zeta}$ is the coordinate of the interpolation point in integer space.

3.3.2 Interpolator implementations

Since the cost of interpolation was so high, effort was made to minimize it. Therefore, several specialized versions were created for specific tasks. These efforts managed to replace seven three-dimensional interpolations with three two-dimensional interpolations.

The two-dimensional interpolator was used to interpolate p^* at the dot points of the MM5 grid to the corner points of the stretched grid. In addition, this code was used to interpolate the lateral boundary conditions from the MM5 grid to the stretched grid. The mapping routine that was used to determine the location of the interpolation point within integer space allowed for uneven grid spacing in MM5 since the vertical direction was always one of the coordinates when interpolating the boundary conditions.

The three-dimensional interpolator was used in the initialization routine to interpolate the tracer quantity from the cross points on the MM5 grid to the corner points of the stretched grid in order to determine the initial stretched grid. It was used in the initialization routine to establish the original profile on the stretched grid at the cell center points. In the advection routine it was used to interpolate each of the three velocity components from the the MM5 dot points to the stretched grid corner points.

It was decided that the contravariant velocities would be calculated at the cell corners and then interpolated to the cell faces, instead of calculating them directly at the cell faces. If they had been determined at the cell faces a complicated code of determining the locations of the cell faces in the physical domain would have been needed. Additionally, there are more cell faces than cell corners. For a $55 \times 55 \times 23$ domain, there were 193,344 cell face centers and only 69,984 corner points. By choosing to interpolate each velocity component to the cell corners the number of three dimensional interpolation points was reduced by a factor of 2.76. Once the contravariant velocities were calculated at the cell corners, each contravariant component had to be interpolated to its appropriate cell face as shown in figure 3.2. Since the contravariant velocity is the appropriate velocity in the computational domain which is orthogonal with grid spacing of unity, these interpolations were only two-dimensional. Considering that each two-dimensional interpolation and each three-dimensional interpolation required

four and thirteen one-dimensional interpolations respectively, the total costs comparison for the $55 \times 55 \times 24$ domain was 7.5 million one-dimensional interpolations if the contravariant velocities had been calculated at the cell faces with the additional cost of finding the cell face center locations as compared to 3.5 million one-dimensional interpolations with the method that was used. The problem with this was that the "velocity" used in the interpolations was exactly $1/2$, which as seen in figure 3.7 was a point of discontinuity as the nearest grid point switched from the left point to the right point. The arithmetic mean of the values from the left and right points was used.

In order to interpolate p^* from the cell center locations to the cell corner locations, extrapolation was required to the outer edges of corner points. A slightly modified two-dimensional interpolator was developed to perform this. This extrapolation was not a problem since the corner points were not more than a half a grid cell away from the MM5 grid cell centers. If a situation had arisen where extrapolation over a distance greater than a one had occurred, the Tremback schemes would have failed as the CFL condition would have been violated. The use of this type of interpolator was advantageous due to its ability to perform some minor extrapolation.

3.4 Numerical test cases and results

Several numerical experiments were conducted to validate the code. Initially, a program that generated simple analytical domains, velocity fields, tracer initial conditions, and boundary conditions was written and used to drive the DGAC. The linear advection of a Gaussian pulse and the rotating cone problems were replicated in each of the three directions and planes of the DGAC. Additionally, using this test program, the boundary conditions of the DGAC were tested by advecting cylindrical and spherical tracer profiles through the horizontal boundaries. The results of the one- and two-dimensional test cases were reproduced with the three-dimensional code and the boundary conditions permitted inflow and outflow of the tracer profiles with minimal distortion. Since these test cases do not yield any new information will not be discussed further.

Three numerical experiments were run to test the coupling between MM5 and the DGAC. These progressed from a simple case using idealized tracer and wind fields, to using physically realistic tracer fields and wind fields simulated by MM5. The numerical accuracy of the schemes is analyzed. Since the DGAC did not replace any MM5 code, the coupled system took more computational effort than MM5 alone. It was determined that the interpolation scheme took in excess of 80% of the computational effort of the DGAC. Until a new model is developed that uses the DAG technique for all of the prognostic fields, thus eliminating the need to interpolate, computational savings will not be achieved. For this

reason the computational efficiency of the system was not analyzed in detail, though the potential for savings will be explored.

The coupling of the DGAC and MM5 was first tested using a simple zonal advection case. This test case, which is discussed in section 3.4.1, used an analytical tracer profile and wind field. The second test case, which was used extensively to test the new system, used the same analytical tracer profile as the zonal test case, but in conjunction with the wind fields from the March 6, 1992. It was a 24 hour simulation run at varying resolutions. The results show that the error associated with the static and dynamic MPDATA scheme is lower than the error of the standard MM5 leap frog scheme, and the use of the dynamic MPDATA scheme has improved the results compared to the static MPDATA scheme. The third test case, which is described in section 3.4.2, used the water vapor field from March 6, 1992 as the initial tracer field and the wind fields and water vapor from March 6, 1992 through March 11, 1992 as boundary conditions for a five day simulation over the continental United States. This test case is explained in section 3.4.3. All of the test cases used the MPDATA scheme with four iterations and the inclusion of the "third" order correction terms.

3.4.1 Zonal test case

By examining the governing equations that were presented in section 3.1.1 (see appendix B for additional details), initial and boundary conditions were developed that produced a field that was almost constant throughout the domain and was purely zonal flowing from west to east. The only variation was a slight shear in the flow in the meridional (north-south) direction. The development of the initial and boundary conditions are developed in section 3.4.1.1.

Although the results are shown over the central United States, there was no topographical variation for this case, and thus the geopolitical boundaries are shown to provide a sense of scale only. The test case was developed so that an analytical profile would be advected across the domain in approximately 24 hours.

An analytical function was used to produce the initial condition. It was specified on the MM5 Cartesian grid. The function was

$$q(x, y) = \max \left(\frac{4 - 4\sqrt{(x - 350)^2 + (y - 810)^2}}{243}, 0 \right) \quad (3.52)$$

where the reference point was the south-west corner of the domain and the unit of measure is kilometers. The same tracer profile was specified at all the vertical levels of the model. Figure 3.8 shows the initial condition.

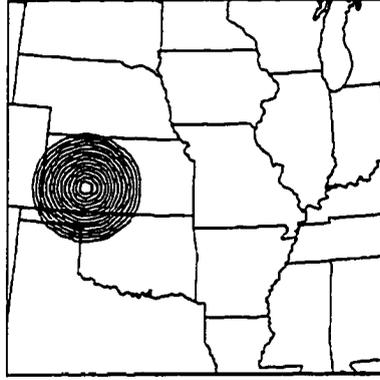


Figure 3.8 Initial tracer field for the zonal flow and cylinder advection test cases.

3.4.1.1 Development of the initial and boundary conditions for the zonal flow case

Equations 3.7 through 3.12 give the equation set for MM5. In order to set up a simple test case with purely zonal flow the following assumptions were made:

- $v = w = 0$
- $u = \frac{c}{f}$, where $c = \text{constant}$
- $T = T_{oo}$, isothermal
- $p^* = \text{constant}$, level terrain
- $m = 1$, no Earth curvature effects on model geometry
- $f = \beta y + f_o$. A β plane approximation uses a uniform gradient approximation ($\beta = \text{constant}$) for the Coriolis term, thus $u = u(y)$ only.
- $q_v = q_c = q_r = 0$, completely dry atmosphere
- $\dot{Q} = 0$, no radiation
- $D_{u_i} = D_w = D_T = D_{p'} = 0$, no molecular viscous diffusion
- $p' = p'(y, \sigma)$ only

Under these assumptions $\dot{\sigma} = 0$ and $\text{DIV} = 0$. The x momentum equation 3.8 when $i = 1$ and the energy equation 3.10 are identically satisfied. The y momentum equation 3.8 when $i = 2$ becomes:

$$\frac{\partial p'}{\partial y} = -c \frac{p_o + p'}{RT_{oo}} \quad (3.53)$$

where the ideal gas relationship has been used to replace the density ρ . The perturbation of virtual temperature is zero since

$$T'_v = T_v - T_o = T(1 + 0.608q_v) - T_o = T - T_o = 0 \quad (3.54)$$

recalling that the atmosphere is isothermal. With this result equation 3.9 reduces to

$$\frac{1}{p'} \frac{\partial p'}{\partial \sigma} = \frac{p^*}{p_o} \quad (3.55)$$

Equations 3.53 and 3.55 stipulate what the p' variation in y and σ must be. Rewriting 3.53 in terms of p where $p = p_o(\sigma) + p'(y, \sigma)$:

$$\frac{\partial p}{\partial y} + \frac{c}{RT_{oo}} p = 0 \quad (3.56)$$

Solving equation 3.56 and substituting back in for p'

$$p' = A e^{-\frac{c}{RT_{oo}} y} - p_o \quad (3.57)$$

where A is a function of only σ . Differentiating this expression with respect to σ and substituting into 3.55 results after simplification in:

$$\frac{\partial A}{\partial \sigma} = \frac{p^*}{p_o} A \quad (3.58)$$

Solving equation 3.5 for p_o and substituting:

$$\frac{\partial A}{A} = \frac{p^* \partial \sigma}{p_o} \quad (3.59)$$

Solving this equation for A and substituting it into equation 3.57:

$$p' = A_s \frac{P_o}{p_{oo}} e^{-\frac{c}{RT_{oo}} y} - p_o \quad (3.60)$$

where A_s is a constant of integration and p_{oo} is the reference pressure at the surface. Setting $p' = 0$ when $\sigma = 1$ and $y = 0$ specifies $A_s = p_{oo}$. Thus 3.60 reduces to:

$$p' = p_o \left(e^{-\frac{c}{RT_{oo}} y} - 1 \right) \quad (3.61)$$

With these specifications made the pressure perturbation equation (3.7) and y momentum equation (3.8 with $i = 2$) are identically satisfied and a complete set of initial and boundary conditions can be developed that produce a flow with no v component, is constant in the x direction and has a slight shear in the y direction.

Initial and boundary conditions files were created using these relationships. The constant c was chosen such that $u = 12.5$ m/s at the central latitude line. Other constants were chosen as $p^* = 900$ mb;

$T_{oo} = 273 \text{ K}$; $p_{oo} = 1000 \text{ mb}$; and $R = 287 \frac{\text{J}}{\text{kg K}}$. The constants in the Coriolis parameter were chosen so that:

$$f = 1.8 \times 10^{-8}y + 7.454 \times 10^{-5} \quad (3.62)$$

where $y = 0$ at the bottom of the domain and is measured in kilometers.

3.4.1.2 Results of the zonal flow case

The zonal flow case was run with and without grid stretching. The grid was 55×55 in the horizontal plane with 23 vertical levels. The static uniform grid had grid spacing of 30 km. In addition to the two MPDATA solutions, a solution using the MM5 central differenced leap frog scheme was also computed. In figure 3.9 results at 20 and 24 hours for comparison without and with boundary condition effects. The data indicate that the DGAC worked very well. Comparison of the MM5 advection scheme and the static MPDATA scheme both at 20 hours indicates that the MPDATA scheme is less diffusive than the standard MM5 scheme. The maximum contour line for the MM5 leap frog scheme was 3.5 while the maximum for the MPDATA scheme was 3.75. When the dynamic MPDATA scheme was used the contours are tighter and the diffusion is less than when the static MPDATA scheme was used. The outflow boundary condition of the two MPDATA schemes caused less distortion to the profile than did the the standard MM5 outflow boundary conditions. Longer simulations were run to demonstrate that the entire profile could pass out of the domain.

3.4.2 Cylinder advection case

The zonal flow test case was a good first test case, but it did not test the system using a divergent flow nor did it contain a variable p^* . In order to include these two effects, a second test case was performed that used the same initial tracer condition as shown in figure 3.8 but for the prognostic MM5 fields used modified initial and boundary conditions for the 24 hour period starting at 0 hours UTC March 6, 1992.

Both MPDATA and the standard MM5 advection routine were used to compute the tracer transport using the wind fields from MM5. In order to verify that the two schemes would converge to the same solution given a fine enough grid, it was necessary to be able to perform a series of simulations with both schemes on different resolution grids. Although it was desirable to change the grid resolution of the advection routine, it was undesirable to have the MM5 wind fields change as it would have been impossible to distinguish if changes were due to the advection routine or changes in the wind field. Initially, it was thought that the MM5 grid resolution would need to be varied and steps were taken to

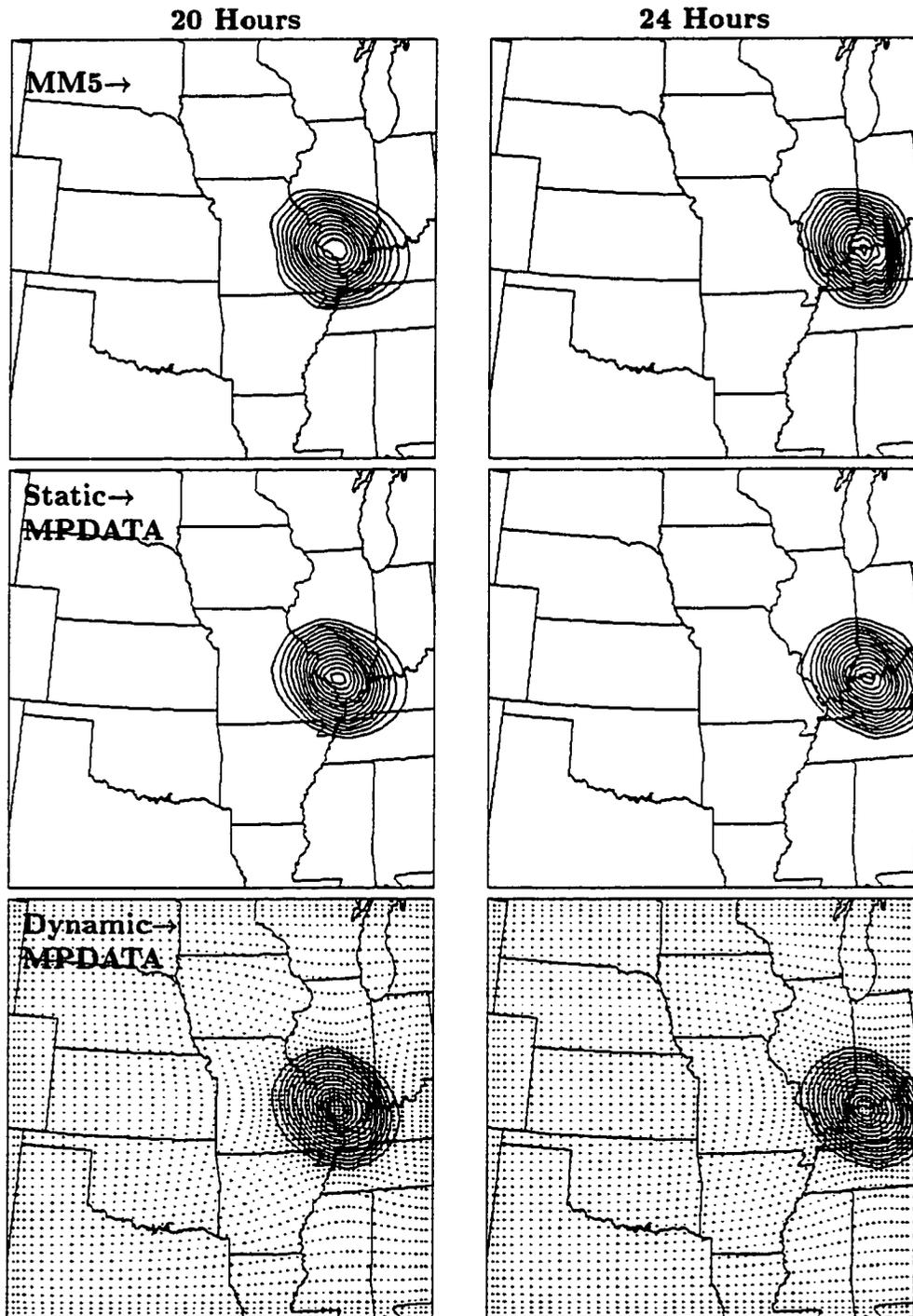


Figure 3.9 Zonal flow test case results. The left and right columns depict results after 20 and 24 hours respectively. The first, second, and third rows represent results from the MM5 advection routine, the static MPDATA scheme, and the dynamic MPDATA scheme respectively. Minimum contour is 0.25 with 0.25 increments.

minimize the effect on the dynamic fields or changing the wind field. Later, in the study a method was developed that allowed the MM5 resolution for the wind field to remain constant while the MPDATA and standard MM5 advection routines were run on different resolution grids tracer advection.

The step that was taken to reduce the effect of changing the MM5 resolution was to remove the effect of latent heat release. Changing the MM5 grid resolution affects the explicit and implicit cumulus parameterizations, which in turn can strongly affect characteristics of the flow field through latent heat release. In order to remove this effect the water vapor variable q_v was set to the insignificantly small value of 1.0×10^{-12} .

The alternative method, that is described in section 3.4.2.1, allowed MM5 to remain at the same resolution while simulating the advection at higher resolution by both the MPDATA scheme and the standard MM5 advection routine. Therefore, in the final analysis the removal of the latent heating effects was not necessary, and although the results would be slightly different had it not been done, it had no meaningful effect on the conclusions.

Several 24 hour numerical experiments were run using the cylinder advection case. These included:

1. Comparing simulations using the MM5 advection scheme, the static MPDATA scheme, and the dynamic MPDATA scheme. All of these simulations were performed on $55 \times 55 \times 23$ grids. This resolution will be called the standard MM5 resolution.
2. Performing the same simulations, with twice as many points in the horizontal directions, using the MM5 advection scheme and the static MPDATA advection scheme.
3. Comparing similar simulations as in item 1, except with progressively fewer grid points in the horizontal directions for the static and dynamic MPDATA schemes. Four test were conducted with 40%, 22%, 11%, and 4.8% of the grid points as in the standard size grid.

To show that the MPDATA and the MM5 leap-frog scheme converged to the same solution it was necessary to perform the same simulation on a relatively fine grid. The DGAC was originally designed so that it could be run at a higher or lower resolution than MM5. Therefore, no modifications were needed in order perform the simulation with MPDATA at a higher resolution. However, it was necessary to develop a method of performing the MM5 advection routine at a resolution independent of the rest of MM5. This set of subroutines is explained in the next section.

3.4.2.1 MM5 leap frog scheme at variable resolution

A set of subroutines that performed tracer advection using the central difference leap-frog scheme on an independent grid was developed by extracting from MM5 the pieces of code that were needed to compute the MM5 tracer advection and implementing them on a grid whose resolution was independent of the rest of MM5. The initialization processes were:

1. Generate an Arakawa B grid that discretized the same domain as the the standard MM5 grid with the appropriate number of grid points.
2. Interpolate the initial tracer profile, p^* , and map factors from the standard MM5 grid to the new high resolution grid. The interpolator of Smolarkiewicz and Grell (1992) was used for this purpose.
3. Calculate the horizontal diffusion coefficients based on the new grid spacing and same time step as the rest of MM5.

In order to make the standard resolution and higher resolution grid match, the outer edge cross points of both grids were chosen to align (the outer dot points could have been chosen to align, but there did not seem to be an advantage of one choice over the other). Figure 3.10 shows the correspondence between the grids. For clarity of presentation, the forcing frames (see section 3.1.3.1 for definition of a forcing frame) for both the coarse grid and the fine grid are shown as being three grid points wide, but the actual width was five grid points. It can be seen that the fine grid forcing frame is completely encompassed within the coarse grid forcing frame. This allowed for easy interpolation of the boundary condition information from the coarse grid forcing frame to the fine grid forcing frame. If the new grid had been allowed to be coarser than the old grid, providing boundary condition information to this wider forcing frame would have been difficult. Because of this difficulty, only new horizontal grids that were finer than the standard 55×55 MM5 grid were used.

During each MM5 time step, a subroutine was called that performed advection on the fine grid. The processes involved were:

1. The velocity variables u , v and σ were interpolated from the coarse grid to the fine grid using the interpolator described in section 3.3.
2. The boundary condition information was interpolated from the MM5 forcing frame to the fine grid forcing frame.
3. Time dependent components of the artificial dissipation were calculated.

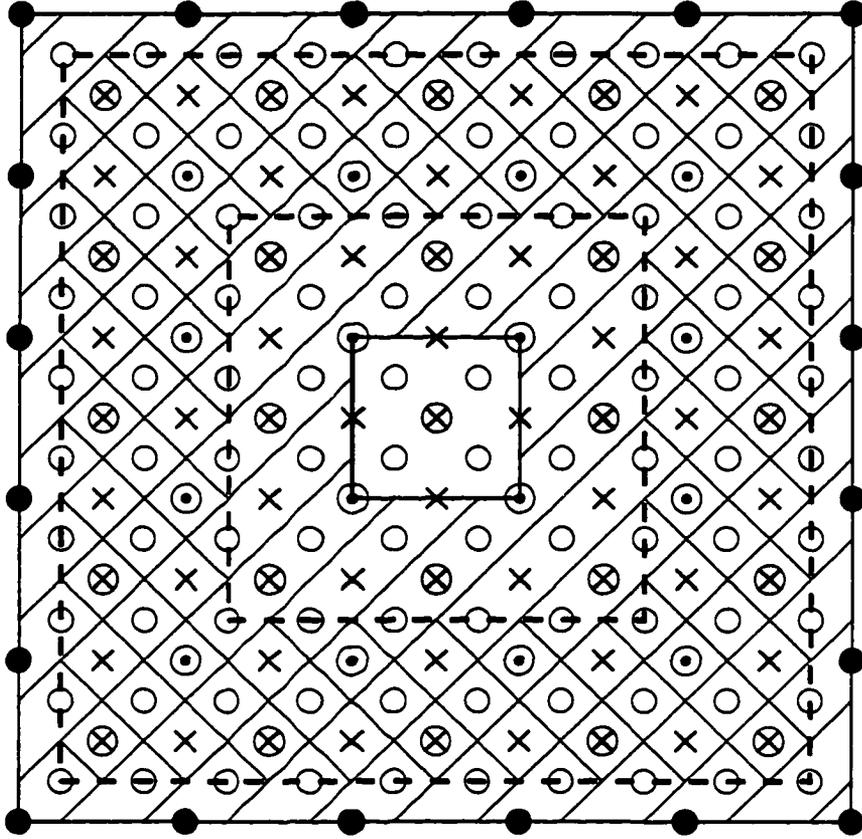


Figure 3.10 MM5 fine grid and MM5 coarse grid correspondence. The coarse grid cross points are represented by \otimes . The coarse grid dot points are represented by both \bullet and \odot . The fine grid dot points are represented by \circ . The fine grid cross points are represented by \times , \otimes , and \odot symbols. For illustration purposes the forcing frame is shown to be three-grid points wide. The actual forcing frame is five grid points wide. The left running hatching shades the forcing frame for the coarse grid, while the cross hatching shades the fine grid forcing frame.

4. The fine grid solution was advanced by calling the MM5 horizontal and vertical advection routines.
5. The boundary conditions were updated.

The fine-grid MM5 advection component was verified by making the resolution identical to the standard MM5 resolution of 55×55 in the horizontal directions and verifying that the tracer advection routine internal to MM5 and the new advection routine produced the same results.

3.4.2.2 Results of the cylindrical advection case at standard resolution

Computations were performed using both static and dynamic MPDATA schemes with 55×55 grid points in the horizontal domain and 23 vertical σ levels. This resolution will be referred to as standard resolution. The static grid had a resolution of 30 km. Figure 3.11 show results from these computations after 24 hours of simulation. The tracer fields in the left column of panels were computed with the MM5 leap frog scheme. The center column results were computed by the static MPDATA scheme, and the right column with the dynamic MPDATA scheme. There are several general observations that can be made concerning these plots. First, it should be noted that the original circular profile has been drawn into a long filamentary structure at mid-tropospheric levels in a short period of time. This illustrates the difficulty in modeling atmospheric tracer transport and underscores the need for this investigation. Second, by comparing the profiles at different levels in the atmosphere, it is apparent that the wind field and its effect on the tracer field is a strong function of height. This posed a problem of how to generate a dynamic grid with no vertical dependency from a profile with strong vertical dependence. This is discussed in section 3.4.2.2.1. Last, comparisons between the MM5 scheme, the static MPDATA scheme, and the dynamic MPDATA scheme can be made. This is discussed in section 3.4.2.2.2

3.4.2.2.1 Composite weight function: Unlike the zonal flow case, the velocity field was dependent on the height in the atmosphere. From all three columns in figure 3.11 it is apparent that the velocity field is a strong function of height σ . At the highest levels (small σ values) the profile was advected to the north east corner of the domain, while the lower level (larger σ values) wind fields blew the tracer towards the south west corner.

Recall that the grid stretching was independent of height to keep columns of grid points vertical. This is evident in grid point distribution for all panels in the right column of figure 3.11. The shearing of the profile with height posed a question on how to stretch the grid. A two-dimensional weight function composite from vertical σ layers was needed. Three different methods of forming this two-dimensional

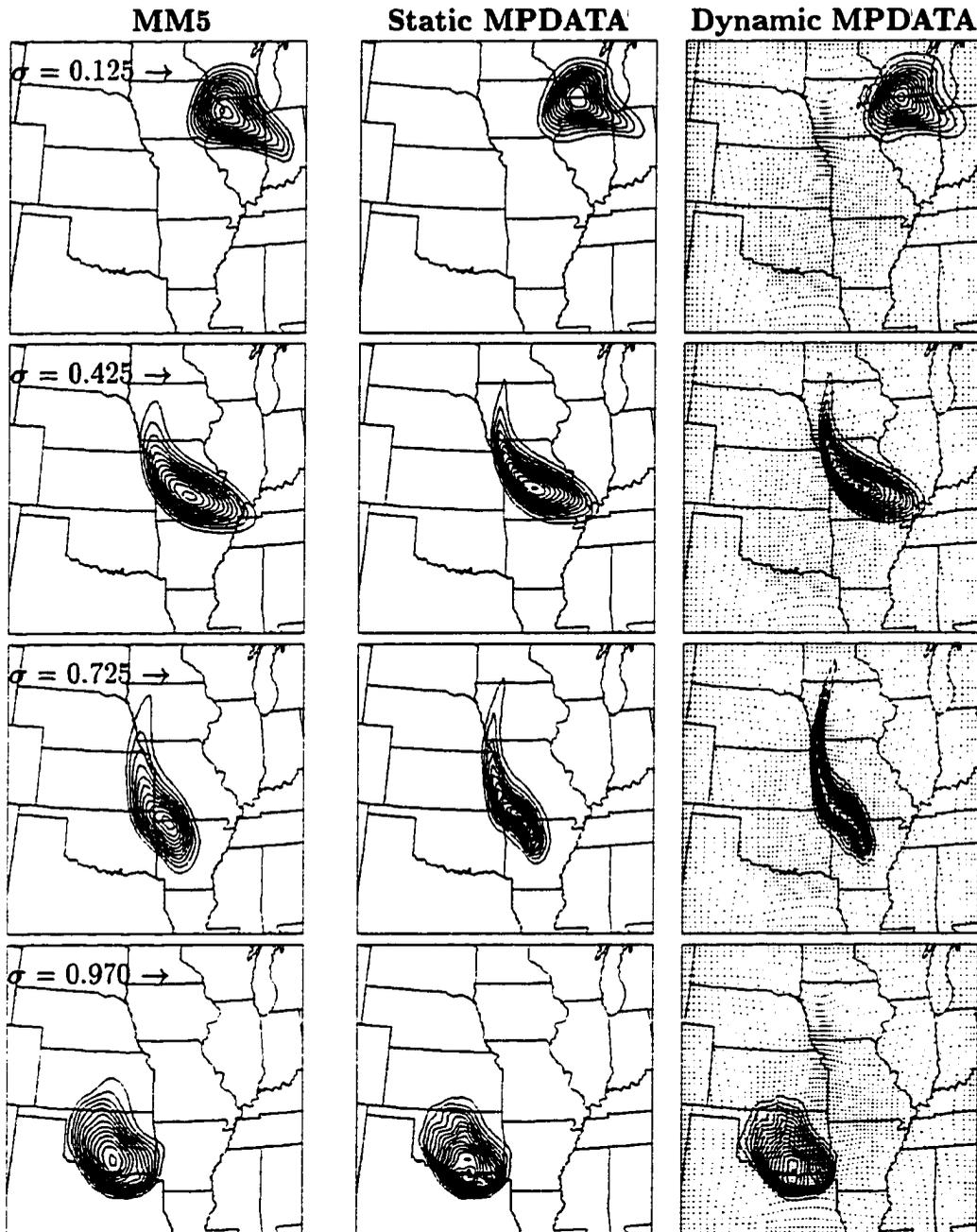


Figure 3.11 Results of the cylinder advection case at standard resolution after 24 hours of simulation. The left, center, and right columns depict results from standard MM5 advection, static MPDATA, and dynamic MPDATA respectively. The first, second, third, and fourth rows show contours at $\sigma = 0.125$, $\sigma = 0.425$, $\sigma = 0.725$, and $\sigma = 0.970$ respectively. Minimum contours are 0.25 with increments of 0.25. Initial tracer field given in equation 3.52 and illustrated in figure 3.8.

horizontal composite were evaluated. The first two of these were not very successful, while the third proved to be satisfactory.

In the first method the composite weight function was formed by computing the sum of the tracer value over a certain range of σ levels for a column of grid points and then calculating the weight function based on the vertically summed tracer field. This created artificially high or low first and second derivatives, and was unsatisfactory.

The second method was similar to the first, except the weight function was calculated at each σ level within a certain range. The sum of these weight functions was then calculated for each column of points. This superposition of the weight functions also created artificially high values for the weight function.

The third method, like the second, computed the weight function at each σ level within a column. Unlike, the second method, the composite weight function was computed by taking the maximum weight function in the column examined. This method proved to be satisfactory.

Compositing was tested for different σ ranges. Generally, values of $\sigma > 0.5$ were considered. This was done in anticipation of using water vapor as the tracer field. Water vapor mixing ratio drops exponentially with height in the atmosphere. So that, the vast majority of water vapor is in the lower half of the atmosphere.

Initially, only the $\sigma = 0.675$ level was used. The grid clustered very well around the tracer profile at the $\sigma = 0.675$ value, but at levels where the tracer concentration was not vertically aligned with the field at $\sigma = 0.675$, the lack of grid resolution degraded the tracer transport.

A computation was performed compositing σ values between 0.5 and 0.85. This improved the tracer transport properties at the added σ levels without serious degradation of the solution at $\sigma = 0.675$. The final computation expanded the range to the Earth's surface ($\sigma = 1.0$). These are the results shown in the right hand column of figure 3.11. Note that at $\sigma = 0.125$ the profile does not coincide with the grid stretching, and at $\sigma = 0.425$ it marginally coincides. However, at σ levels of 0.725 and 0.970 the grid has adapted to resolve large gradients and local maxima.

3.4.2.2 Comparisons of MM5 advection scheme and static MPDATA: Comparing the three simulation results shown in figure 3.11 with each other reveals that the MM5 simulation has the smallest maximums and the shallowest gradients. The dynamic MPDATA scheme had the highest maximums and the steepest gradients. Whereas, the static MPDATA results were between these two extremes. It was not obvious which of these was the best answer. In order to determine which was the

best solution the simulation was repeated at twice the standard resolution using the MM5 and static MPDATA schemes.

3.4.2.3 Results of the cylindrical case at high resolution

In order to verify that the dynamic MPDATA scheme outperformed the static MPDATA scheme and the MM5 scheme, higher resolution computations were performed on static grids. Figure 3.12 shows some of the results of these computations. The panels in the left hand column were computed with the MM5 leap frog scheme at a 15 kilometer resolution. The results depicted in the second column of panels were computed with the static MPDATA scheme also at 15 kilometer resolution. The plots in the third column show the difference between the two plots (MM5 – static MPDATA).

The MM5 results in the left column and the MPDATA results in the second column appear to be more similar when compared visually than the difference plots in the third column would indicate. The very large gradient from negative to positive difference values indicated that there was some phase differences between the two computations. The steep gradients of the tracer field could make the difference plots very sensitive to even small phase errors.

The MPDATA results were shifted in space in an attempt to determine the difference in phase between the two simulations, if any, that was present. Initially, this was done by aligning the locations of the maximum value of the tracer concentration in the MPDATA results with the maximum location of the tracer profile computed with MM5's scheme. Although this may not have been the most optimal method of aligning the profiles, it decreased the magnitude of the differences and the area of the difference contours in 14 of the 23 σ levels. Of the remaining 9 σ levels the magnitude of the differences became more extreme or there was no significant change. The MPDATA results of these 9 σ levels were incrementally shifted in order to align the profiles. In the final analysis the magnitude and the area of coverage of the differences were greatly reduced by shifting the MPDATA results in space. There was systemic variation to the direction the profiles were shifted. The average distances shifted were 17 km in the x -direction and 25 km in the y -direction. Since the grid spacing was 15 km, these represented only a small shift. The fourth column of figure 3.12 shows the difference results, once the contours were aligned.

Comparison of figures 3.11 and 3.12 shows that the MPDATA solution at standard resolution is closer to both the MPDATA and MM5 results at high resolution than was the standard resolution MM5 results. The conclusion was that the MPDATA scheme used in the DGAC was superior to the leap frog scheme employed within MM5. Therefore, it was decided that the MPDATA solution at high

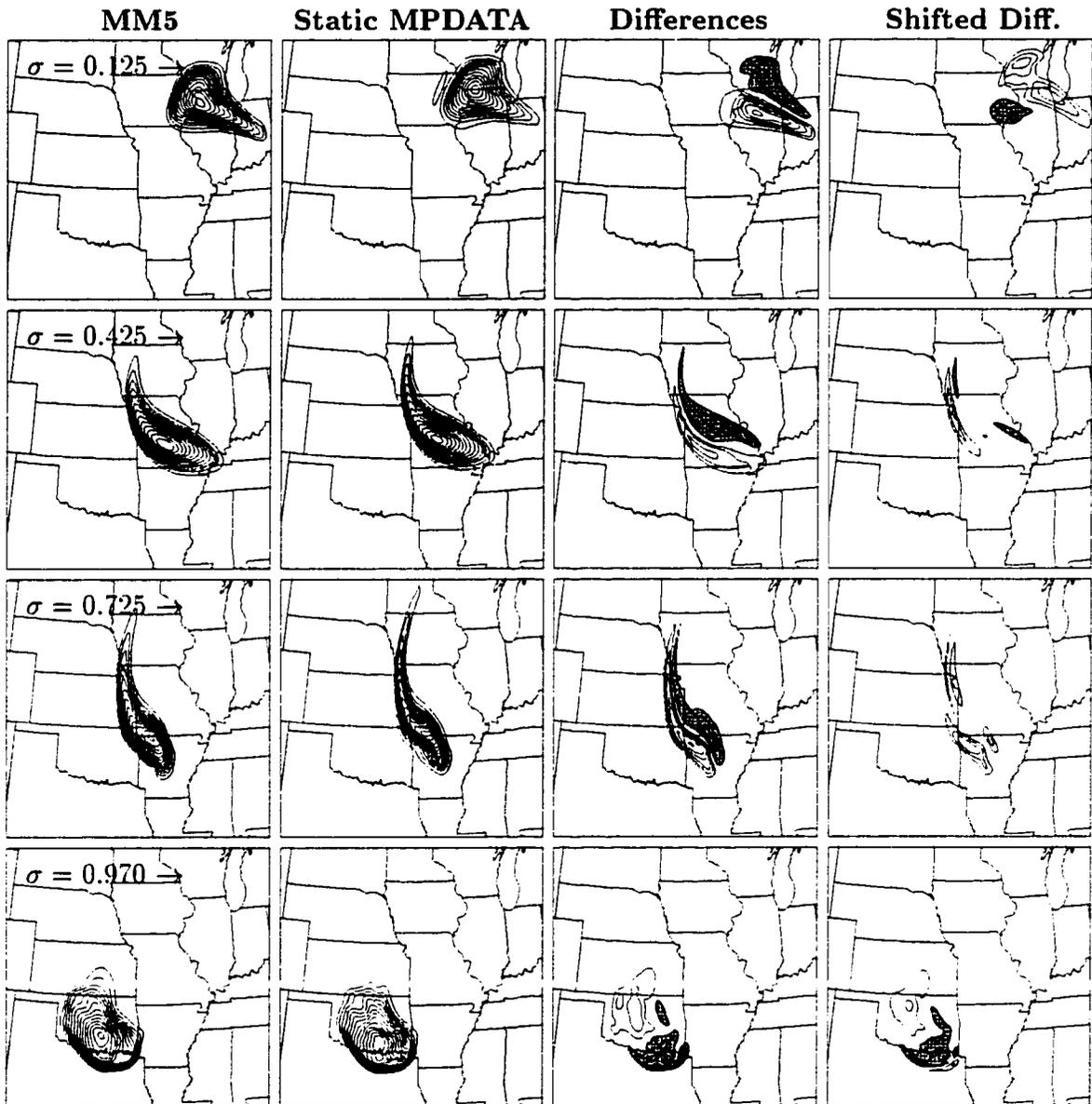


Figure 3.12 Results of the cylinder advection case at high resolution (15 km) after 24 hours of simulation. The four columns from left to right show results from the MM5 advection routine, the results from the static MPDATA scheme, the differences between the static MPDATA scheme and the standard MM5 advection routine before the data was shifted, and the differences between the static MPDATA scheme and the standard MM5 advection routine after the data was shifted respectively. The rows from top to bottom show the results on the $\sigma = 0.125$, $\sigma = 0.425$, $\sigma = 0.725$, and $\sigma = 0.970$ levels respectively. Minimum contours are 0.25 with increments of 0.25. Negative contours are shaded.

resolution would be considered the reference solution for judging the quality of other solutions. The “error” in a computation was computed by interpolating the solution to the static fine grid and then subtracting the fine grid MPDATA solution.

Figure 3.13 shows the difference between the MM5 leap frog scheme at standard resolution (left column), the static MPDATA computation at standard resolution (center column), and the dynamic MPDATA computation (right column). Note that the contour interval in the left column is 0.25 while for the other two columns it is only 0.15.

As expected the MM5 computations showed the most error and the dynamic MPDATA the least. A significant portion of the error in the MM5 computation may have been phase error. By shifting the profiles, the contours may have aligned better and the difference reduced. However, after examining figures 3.11 and the second column of figure 3.12 it was apparent that there was significant diffusion error in the MM5 computation compared to the fine grid MPDATA calculation. Therefore, little benefit was seen in shifting the data.

Referring again to figure 3.13, the dynamic MPDATA computations had the least error. The weight function was determined using the tracer field on σ levels greater than 0.5. The lack of influence of the upper σ resulted in only slightly higher errors at the top half of the model. The pattern shown for the static MPDATA scheme at $\sigma = 0.725$ is indicative of what was found on many of the σ levels for the static grid. The negative quantities in the center indicate that the maximum tracer concentration was low and the positive error around the outside indicates that the profile spread. This combination indicates that the static DGAC computations had a small amount of diffusion associated with them. The lack of this pattern for the dynamic MPDATA scheme at the same level indicates that this diffusion has been virtually eliminated through the use of the dynamic grid.

3.4.2.4 Results of the cylinder advection case at lower resolution

In order to determine the effect of changing the resolution of the dynamic MPDATA scheme, eight additional computations were performed. The number of grid points was varied to determine the impact of using fewer grid points with and without a dynamic grid. In addition to the standard grid of 55×55 , horizontal DGAC grids of 35×35 , 26×26 , 18×18 , and 12×12 were used. All computations were performed with 23 σ levels. Results from each of these computations were compared to the static MPDATA computation with 105×105 . The RMS error was calculated as:

$$E_{\text{rms}}(\sigma) = \left(\frac{1}{IJ} \sum_{i,j=1}^{I,J} (q_{i,j} - \hat{q}_{i,j})^2 \right)^{1/2} \quad (3.63)$$

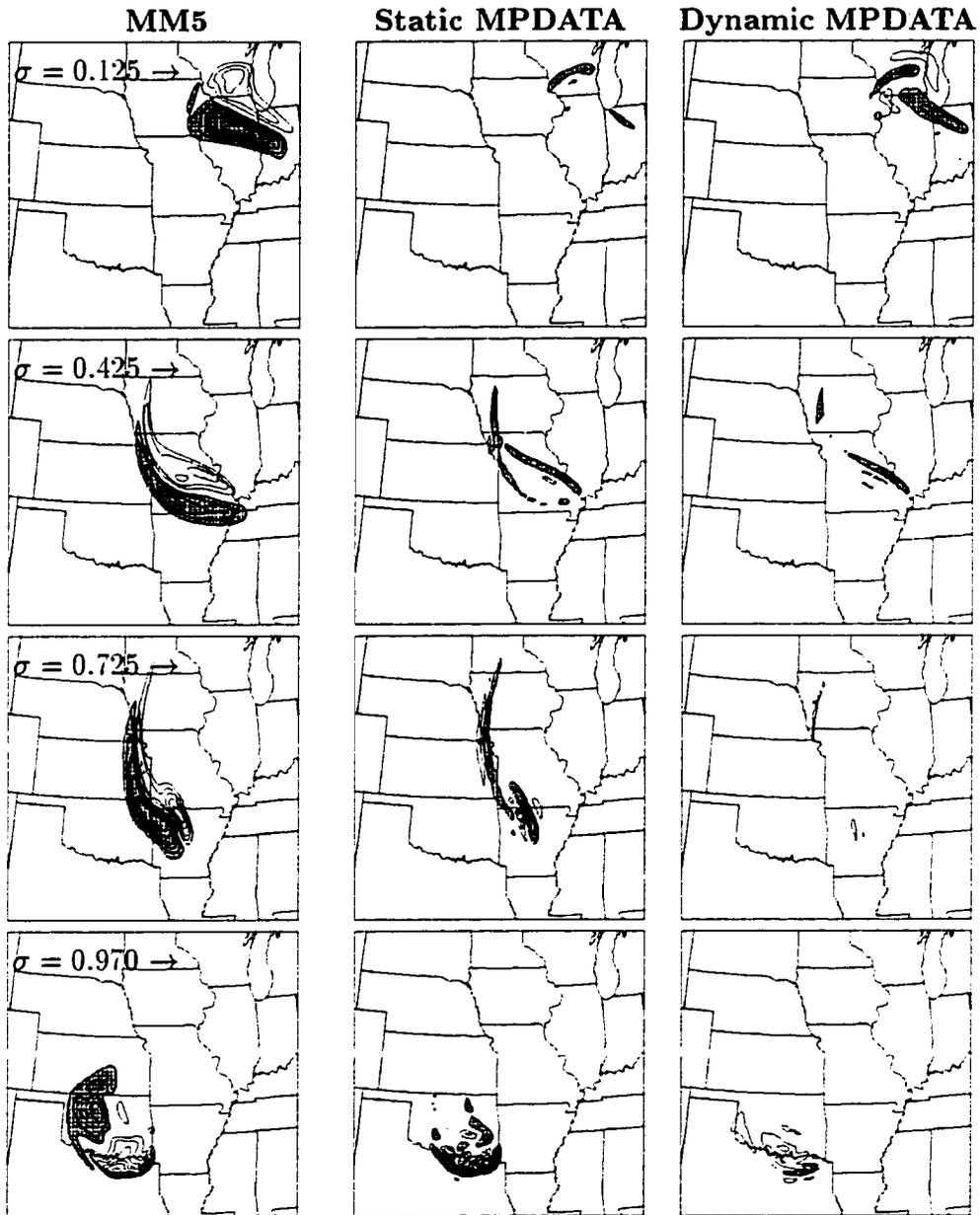


Figure 3.13 Differences between standard resolution (55×55) results and fine resolution (105×105) MPDATA results on a static grid. The left, center, and right columns represent the “errors” in the standard MM5 scheme at standard resolution, the static MPDATA scheme at standard resolution, and the dynamic MPDATA scheme with the same number of grid points as the standard grid. The rows from top to bottom depict contours at $\sigma = 0.125$, $\sigma = 0.425$, $\sigma = 0.725$, and $\sigma = 0.970$ respectively. The contour increments are 0.25 in the left column and 0.15 in the center and right columns. The zero contours are excluded. Negative contours are shaded.

where I and J are the number of fine grid points in the x and y directions respectively, $q_{i,j}$ is the computed solution at a point (i, j) , and $\hat{q}_{i,j}$ is the “exact” solution at the same point. The value at this point was determined through interpolation to the fine grid.

Figure 3.14 shows E_{rms} for the static and dynamic DGAC computations at the different σ levels. All of the dynamic grid computations had smaller errors than did their static grid computations using the same number of grid points with the exception of the 12×12 computations. The results of the dynamic MPDATA computations demonstrate less error at $\sigma > 0.5$. This shows the effect of using the tracer in only the lower portion of the atmosphere to influence the grid stretching. Coincidentally, several of the dynamic error plots roughly correspond to static grid plots at different resolutions. Table 3.1 shows the reduction of grid points that is achieved when the dynamic grid is used.

Table 3.1 Effect of dynamic grid on number of grid points needed. The error on each line was approximately equal.

Approximate Error	Static Grid	Dynamic Grid	% fewer points
0.043–0.053	55×55	35×35	59.5%
0.076–0.085	35×35	26×26	44.8%
0.111–0.131	26×26	18×18	52.1%

3.4.3 Water vapor passive advection case

The zonal flow case and especially the cylindrical tracer cases demonstrated the advantages of the dynamic MPDATA scheme. The simplified initial tracer field, however, had different properties than are typically found in atmospheric tracer flows. In order to step towards physically realistic atmospheric fields, the tracer field was initialized with the water vapor field of March 6, 1992 0 UTC. Although this tracer field was initially identical to the water vapor field, it contained no precipitation or evaporation process, and therefore, the computed quantities differed from the physical water vapor field.

The vertical variation typical of this tracer field appears in figure 3.15 which shows the average water vapor mixing ratio over the continental United States as a function of σ level. This strong vertical variation raised the question of how to best create a two-dimensional composite weight function. Recall that the grid stretching contained no variation in the vertical direction. This required that the three-dimensional tracer field needed to produce a two-dimensional weight function from which the grid point movements could be determined. Unlike the cylinder advection case where the values of the tracer field

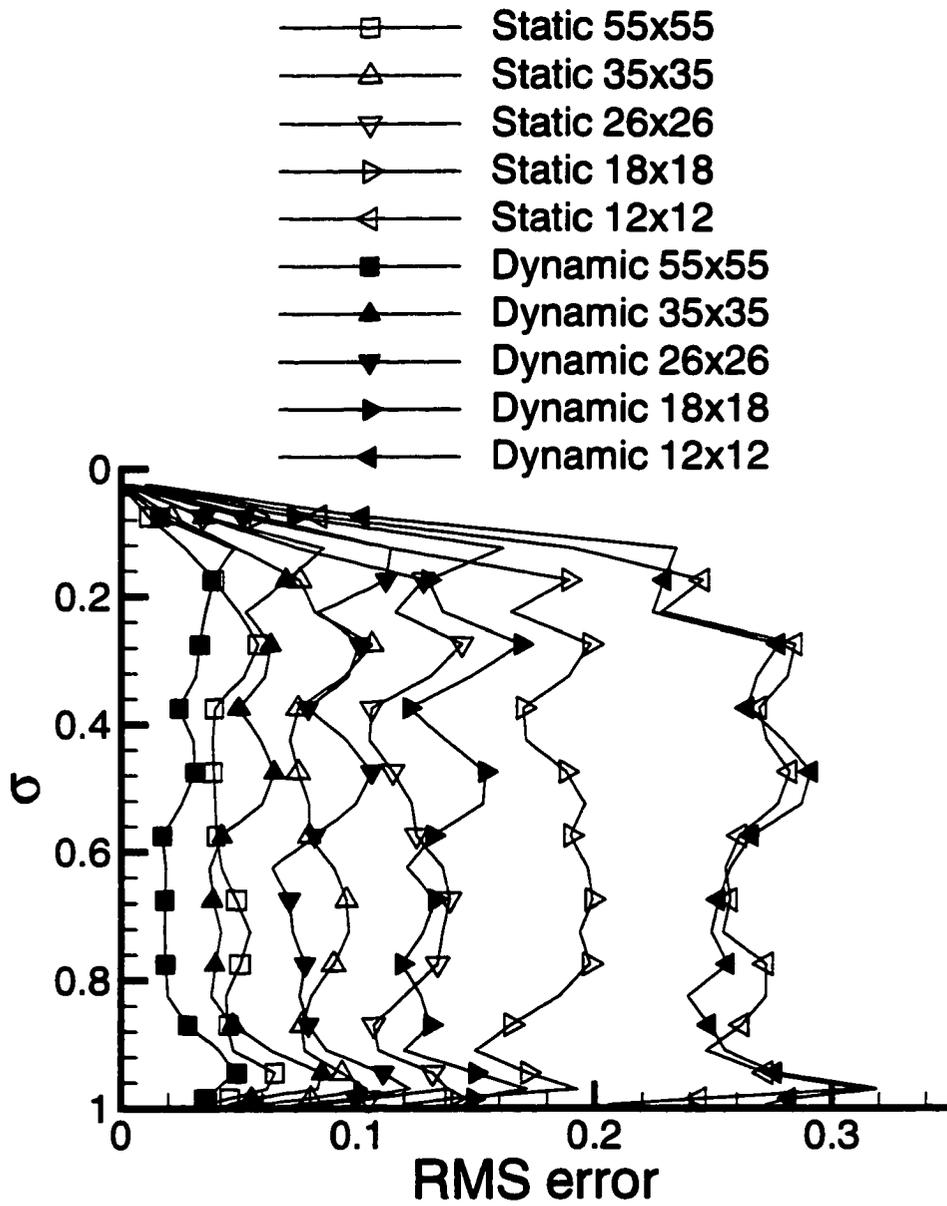


Figure 3.14 RMS error as a function of grid points for static and dynamic DGAC.

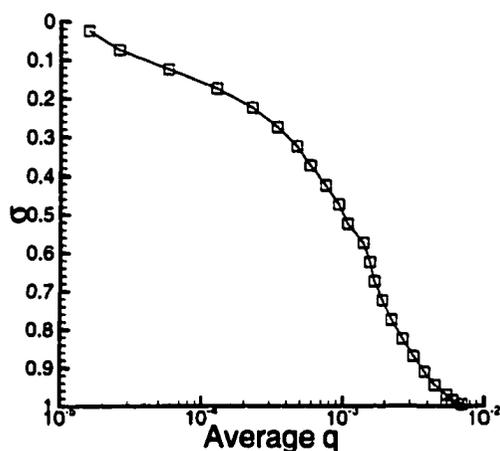


Figure 3.15 The average water vapor q as a function of height. Units are in g of water vapor/kg of air

were of the same order on all σ levels, it was not initially clear how to account for the strong vertical variation in concentration observed in figure 3.15, although the cylinder advection results suggest that tracer simulation is not seriously worsened at levels not used to determine the weight function.

In addition, the range of σ levels on which to base the grid stretching was not clear. The greatest concentrations of the tracer were in the lowest parts of the atmosphere, but this region is strongly affected by the planetary boundary layer and so is not so strongly a product of the resolved transport. Another significant difference between this tracer field and the analytical cylindrical case considered previously was that there was not just a single feature that needed to be resolved.

A series of 24 hour computations were performed on the same horizontal domain as the previous test cases. Both static and dynamic MPDATA simulations were performed along with the standard MM5 scheme. A number of different ranges of σ levels were used to determine the weight function. It appeared that the MPDATA scheme produced sharper gradients and a wider range between maximum and minimum values than did the MM5 scheme. Little difference was observed between the static and dynamic MPDATA schemes. There was very little grid movement observed in the dynamic simulations. Recall that the initial conditions were determined from the NCEP reanalysis data which had a horizontal resolution of about 210 km. These initial tracer fields were much smoother than the analytical profile used previously. It appeared that the simulation period was too short for the internal dynamics to generate any strong gradients, and the domain was so small that before any elongated structures started to form they were transported out of the computational domain. Therefore, additional tracer transport

was simulated for a larger domain for a longer period of time.

Two 5 day simulations over continental United States were conducted using the static and dynamic MPDATA schemes respectively. 101×75 grid points were used with a spacing of 52 km in the MM5 model, with a time step of 90 seconds. The DGAC grid size was set to 100×74 which was the appropriate size so that a static uniform grid had a spacing of 52 km. The MM5 parameterizations remained unchanged, and the initial and boundary condition files were prepared in the same manner as the cylinder advection case. Recall that the water vapor field was set to 1.0×10^{-12} , so that latent heat effects were removed.

Figure 3.16 shows the initial tracer field at $\sigma = 0.825$. The grid stretching was composited in the same manner as in the cylinder advection case except that tracer concentrations from $\sigma = 0.825$ and $\sigma = 0.870$ were used instead of the the level between 0.5 and 1.0. This was done to minimize the effect of the vertical variation of the magnitude of q that was discussed earlier. Suggestions for dealing with this variation are discussed in section 4.3.1.1.

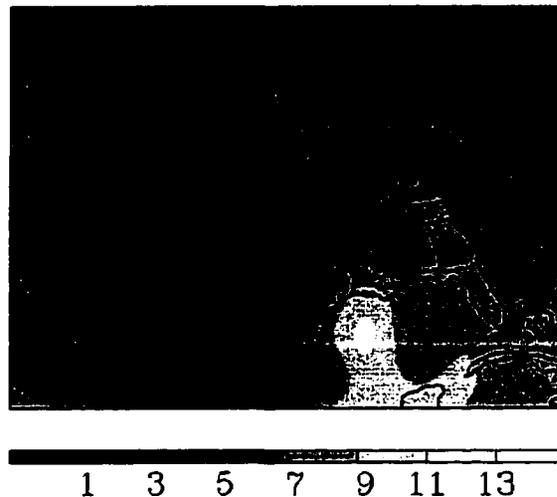


Figure 3.16 Water vapor field at $\sigma = 0.825$ for March 6, 1992 0 UTC. This served as the initial condition for the 5 day simulation of passive water vapor transport. Legend values have been multiplied by 1000 and have units of g/kg .

Figures 3.17 and 3.18 show tracer field on the $\sigma = 0.825$ level after 44 and 120 simulation hours respectively. These two times were chosen because they best illustrate the differences between the MM5 advection scheme, static MPDATA, and dynamic MPDATA. Although the initial condition was characterized by relatively shallow gradients, both the static and dynamic MPDATA schemes as well

as the MM5 scheme all produced structures that were of finer spatial scale than the initial condition. In addition they show long filament type structures that are typical of atmospheric tracer transport as was discussed in section 1.2. In general the dynamic MPDATA scheme resolved gradients and maxima better than the static MPDATA, and they both show much more structure than the MM5 results. As the integration continued, sharper gradients and narrower maximums developed; therefore, greater grid point movement was observed as the dynamic MPDATA computations proceeded. Narrow bands of high q had a tendency to form more often than narrow bands of low q values. Therefore, the grid points had a tendency to cluster around and track along with higher rather than lower concentrations. Grid adaptation occurred around several features simultaneously.

The MM5 computations at 44 hours show much less structure than either the static or dynamic MPDATA schemes. The highest contour level (13×10^{-3}) is not observed in the MM5 computations at all. There are several features in the dynamic MPDATA calculation that appear to be better resolved than in the other two calculations. The filament structure over the Gulf of Mexico has steeper gradients and the highest contour level is continuous. A bimodal pattern in the dynamic MPDATA simulation can be seen south west of the Texas Mexican border that is only hinted at in the static MPDATA run and completely missed in the MM5 run. In addition, the line continues farther north west into New Mexico in the dynamic MPDATA run. The gradients over Cuba and along the east coast states are sharper. The maximum over western Mexico is larger in the MPDATA runs and almost completely missed in the MM5 simulation. The low concentration over Wyoming, Colorado, Kansas, and Oklahoma is more extensive in the static MPDATA computation than in the dynamic. Due to sharper gradients in adjacent regions, grid points did not cluster in this area. For these reasons this low may have been modeled better with the static MPDATA simulation than in the dynamic. Note that grid points did not cluster over the this region due to the sharper gradients in other regions of the domain.

At 120 hours (figure 3.18) the same general conclusions may be drawn as at 44 hours. Specifically, the filament that starts in the Gulf of Mexico continues up into Arkansas and wraps around through Oklahoma and Texas is sharper in the dynamic MPDATA results than in either of the other two results, as is the filament running from Kansas over the Great Lakes into Ontario. Larger maxima occur over Arizona and New Mexico with the dynamic grid.

There is a significant difference at the southern boundary in the Bay of Campeche in the Gulf of Mexico where a maximum can be seen in the MM5 results that does not appear in the MPDATA results. The velocity field on this level ($\sigma = 0.825$) was examined at the 96, 100, 104, 108, 112, 116, and 120 hour points. Figure 3.19 shows this field at 120 hours. This southern boundary features is typical of

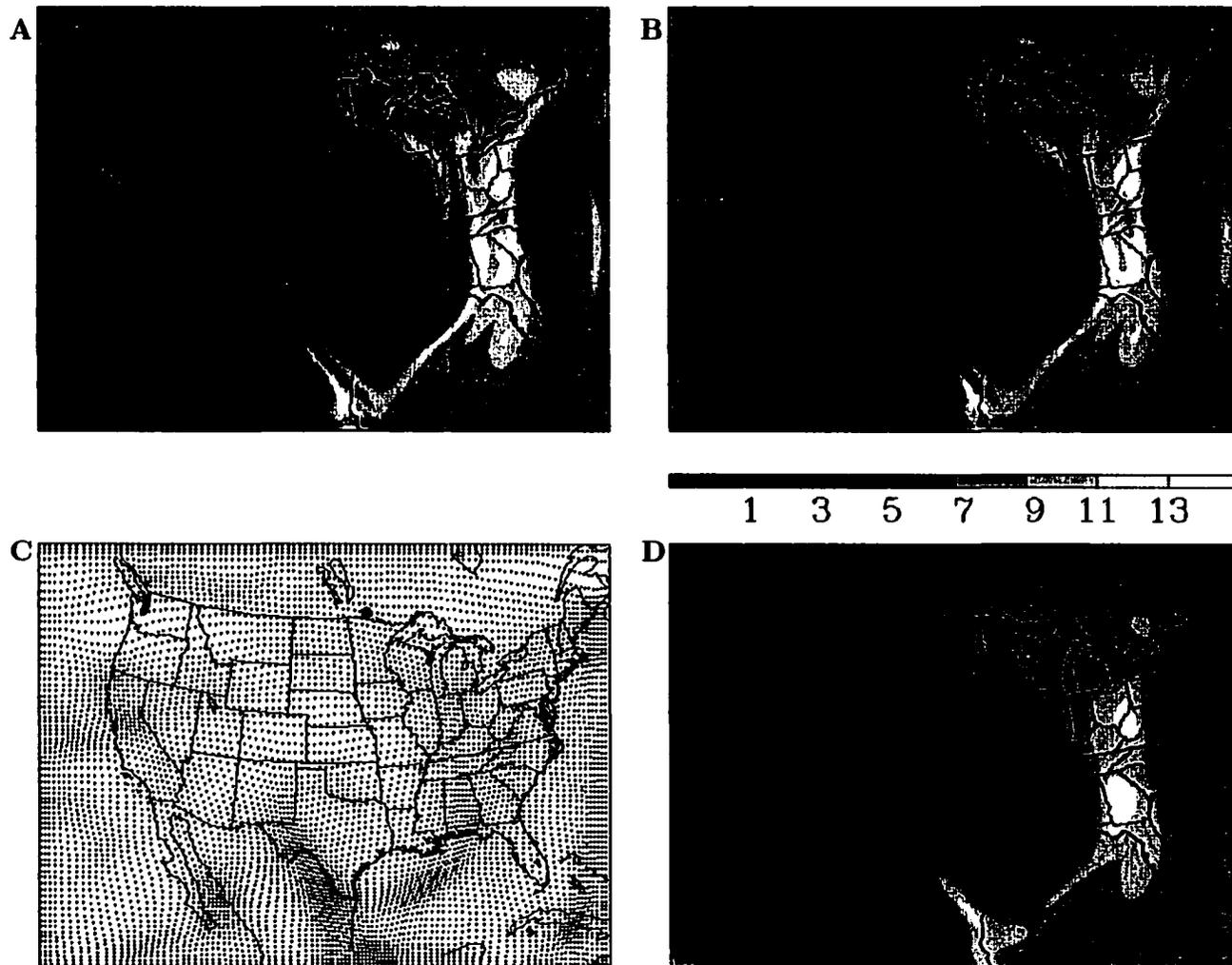


Figure 3.17 Tracer distribution on $\sigma = 0.825$ after 44 hours of simulation. A) dynamic MPDATA B) static MPDATA C) grid for dynamic MPDATA D) MM5 leap frog. Legend values have been multiplied by 1000 and units are g/kg.

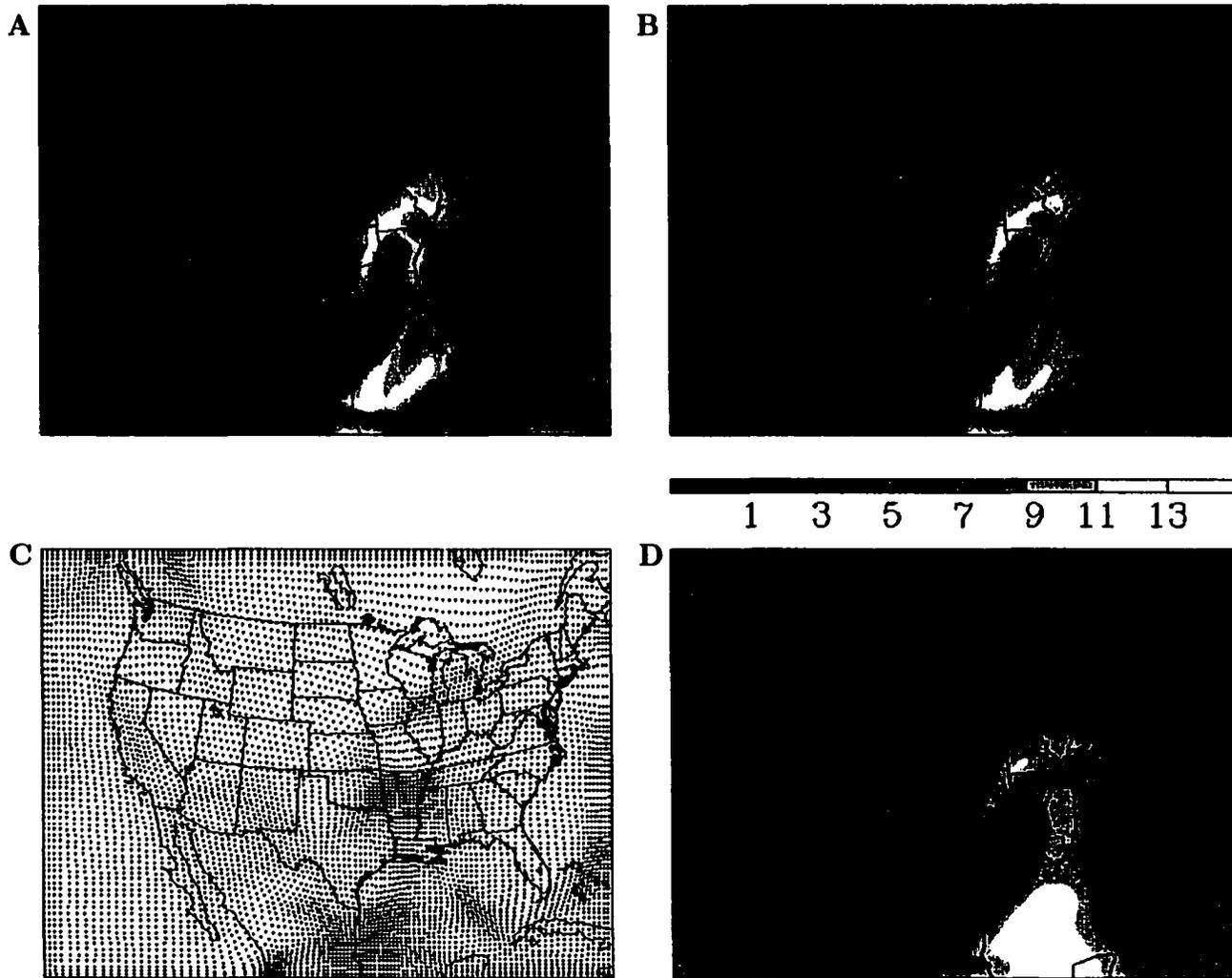


Figure 3.18 Tracer distribution on $\sigma = 0.825$ after 120 hours of simulation. A) dynamic MPDATA B) static MPDATA C) grid for dynamic MPDATA D) MM5 leap frog. Legend values have been multiplied by 1000 and units are g/kg.

what was found for the 24 hours preceding this time. Note the eastern half of the southern boundary had inflow with velocity in the range of 4 m/s. It appears in the MPDATA results that there are high tracer concentrations right at the boundary but not in the interior. Therefore, this maximum could be attributed to advection into the domain. However, the inflow velocity was so weak that the high concentration in the MM5 can not be attributed solely to inflow.

Another explanation is that the relaxation boundary condition discussed in section 3.1.3.1 could allow water vapor to be introduced into the boundary forcing frame. Recall that the MM5 boundary conditions are not applied just at the boundary points but within a forcing frame which consists of a band of 5 grid points at the edge of the domain. The forcing field is strictly imposed only at the outer boundary of the forcing frame. However, the computed solution within the rest of the forcing frame is relaxed towards the prescribed forcing frame values. Even if there was no flow into the domain, these boundary conditions could introduce moisture that was not advected there. Over time nudging could smear these boundary values into the computational domain. This would constitute an unrealistic condition at the boundary. Because of this type of behavior at the boundaries, many investigators place little confidence in the results near the boundaries. Additionally, poorly “tuned” relaxation boundary conditions will cause wave reflections and possible resonance which could add to the inconsistency.

Comparing the other boundaries of the MM5 simulation with the static and dynamic MPDATA simulations indicates that the Eastern and Western boundaries of the MM5 simulation may have numerical inconsistencies. Note that the contour lines are parallel to the boundaries. This indicates that the forcing frame is changing the value rapidly from its exterior to interior edges.

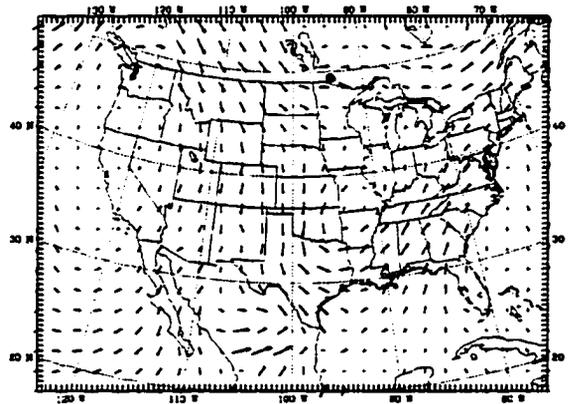


Figure 3.19 Horizontal velocity vector plots at $\sigma = 0.825$ after 120 hours of simulation

4 CONCLUSIONS

This research developed a method of using the Multidimensional Positive Definite Advection Transport Algorithm (MPDATA) in a dynamical adaptive grid (DAG) environment and investigated the application of dynamical adaptive grid techniques to regional modeling of atmospheric tracer transport. One- and two-dimensional models were developed to study properties of the dynamic MPDATA scheme versus a static uniform MPDATA scheme. A three-dimensional tracer advection scheme that was driven by the Fifth-Generation Penn State/NCAR Mesoscale Model (MM5) was developed and tested in several different settings. This concluding chapter summarizes the work and findings, highlights the unique contributions of the work, and gives suggestions for continued investigation.

4.1 Summary

In section 1.3.2 two questions about the use of MPDATA on a moving grid were posed. These questions were:

1. Could MPDATA be made to work in the context of a DAG framework?
2. How would the explicit nature of the MPDATA scheme and the corresponding CFL condition be affected by the grid point movement?

Chapter 2 presents the work that was done to answer these questions.

As background information, the most basic MPDATA scheme was first developed and then a multidimensional version applicable to a moving grid was developed. More sophisticated options of adding “third” order correction terms and using recursive pseudo velocities (RPV) were summarized (details are in appendix A).

A literature review gave the wide range of methods for implementing the DAG technique. The different approaches were broadly categorized into grid point redistribution techniques, local refinement, and local increase of algorithm order. To limit the review to a reasonable size, only grid redistribution techniques were discussed. These techniques included equidistribution techniques in one and multiple

directions, elliptical grid generation systems, variational approaches, grid speed methods, and attraction and repulsion methods. Application of DAG techniques within the atmospheric science community were reviewed without regard to the type of grid adaptation that was used. One- and two-dimensional grid generators based on equidistribution and variational principles respectively were developed in detail. Computational experiments showed that they each had the desired properties of clustering grid points around large first and second derivatives of the prognostic field.

A one-dimensional model was initially developed and used to demonstrate that the dynamic MPDATA scheme indeed does work and that it can produce results that are as accurate as MPDATA scheme with a static uniform grid with many more grid points. As long as the number of grid points was relatively small, the dynamic MPDATA scheme appeared to be third- or even fourth-order accurate, but as the number of grid points increased the scheme could become unstable due to violation of the CFL condition.

To prevent this violation, a method of limiting the movement of the grid points was developed. Analysis of this problem illustrated the Lagrangian aspect of the DAG technique. The grid points can follow targeted features so that they are treated in an approximately Lagrangian manner. As long as the grid points follow the flow the contravariant velocity is small and very large Courant numbers are possible. However, the need for the grid to conform to the boundaries of the domain prevents the grid points from continuing in a Lagrangian fashion.

Once this limiting was applied, the scheme remained stable, but since the grid points could not be clustered closely and their movement was restricted, the order of accuracy of the scheme that was established with fewer grid points did not remain as more grid points were used. The number of times the grid point movement needed to be limited was affected by the number of smoothing passes of the weight function used in the generation of the grid. The smallest errors occurred when the just enough smoothing passes were used to prevent the need to limit the movement of the grid points. When fewer grid points were used, the computational costs of the static and dynamic MPDATA schemes were about equivalent. As the number of grid points increased, the same computational cost was expended, but the error was not reduced as much for the dynamic MPDATA schemes as for the static.

A two-dimensional passive transport model was developed and used to solve the rotating cone problem to gain experience with dynamic MPDATA in a multidimensional case and to better illustrate some of the characteristics of the different MPDATA options. Both the DAG technique and additional iterations or the RPV option reduced the overall diffusion, whereas the use of the "third" order correction terms caused the results to more symmetrical.

Quantitative results using the L_2 , diffusion, and phase errors confirmed the qualitative results. The L_2 error norm defined as

$$E_{L_2} = \left(\frac{1}{L} \int_0^L (\hat{q}(x) - q(x))^2 dx \right)^{1/2} \quad (4.1)$$

where $q(x)$ is the numerical solution and $\hat{q}(x)$ is the exact analytical solution. The diffusion error was defined as

$$E_d = \max_{i,j}(\hat{q}_{i,j}) - \max_{i,j}(q_{i,j}) \quad (4.2)$$

The phase error was determined by calculating the distance between the maximum value of the computed solution and the location of the maximum of the analytical solution.

Although the L_2 error gave a global measure of the error, the purpose of the DAG technique was to reduce the error in certain regions by clustering and moving grid points around and with select tracer field characteristics. For this purpose the diffusion and phase error were better measures of the performance. Grids of up to 201×201 grid points were used in an attempt to produce diffusion error that was as small as the 41×41 grid point dynamic case. This very fine case still had a diffusion error that was almost double and a computational cost of nearly 6 times that of the dynamic case.

Chapter 3 discussed the development of the dynamic grid adaptation component (DGAC) that used the the dynamic MPDATA scheme to compute atmospheric tracer transport. This set of routines was driven by MM5. A review of MM5's capabilities, coordinate system, governing equations, grid, and boundary conditions was given. The details of the DGAC were given, including the development of the advection equation compatible with the MM5 equation set, the modification of the MPDATA scheme to solve this advection equation, the correspondence between the DGAC and the MM5 grids, and the DGAC initialization, updating at each time step, and boundary conditions. The high order monotonic interpolator of Smolarkiewicz and Grell (1992) used to interpolate tracer and wind fields between MM5 and the DGAC was reviewed.

Results of three different test cases were given: a zonal advection test case, a cylinder advection case, and a passive water vapor transport case. All of the test cases were conducted with four iterations of the MPDATA scheme and the inclusion of "third" order correction terms. Each of these cases were progressively more complicated.

The zonal advection test case included analytically determining a set of initial and boundary conditions that when used by MM5 would produce a constant wind field from west to east. These conditions were used to simulate the transport of a vertical cylinder across the domain in a 24 hour period. This was a simple test case due to the relatively simple structure of the wind and initial tracer fields, the lack

of any topographical effects. The simulation was conducted with the standard MM5 leap frog scheme, the static MPDATA scheme, and the dynamic MPDATA scheme. The horizontal grid was 55×55 in each of these simulations and the static uniform grids had a resolution of 30 km. There was less diffusion for the two MPDATA simulations than for the MM5 simulation. In addition, the dynamic MPDATA scheme had less diffusion than the static MPDATA scheme.

The cylinder advection test case used the same initial tracer distribution as the zonal advection test case, but initial and boundary conditions were specified for the prognostic variables from the reanalysis data for 24 hour period of March 6, 1992. Simulations on standard resolution grids of 55×55 , high resolution grids of 109×109 , and lower resolution grids of 35×35 , 26×26 , 18×18 , and 12×12 were computed. The resolution of MM5 remained constant at $55 \times 55 \times 23$ with grid spacing of 30 km for all of these simulations.

The standard resolution case was computed with the MM5 scheme, the static MPDATA scheme and the dynamic MPDATA scheme. As with the rotating cone problem in chapter 2, the two MPDATA computations showed less diffusion than the MM5 scheme and the dynamic MPDATA showed less than the static MPDATA scheme. In order to determine if the MM5 and the static MPDATA schemes would converge to the same solution, the high resolution case was performed

The high resolution computations were performed on static uniform grids only and used the circulation from the standard 30 km case. In order to perform the tracer transport using the MM5 advection scheme on a high resolution grid without increasing the resolution of MM5 circulation itself, a separate set of subroutines was created that used this scheme at a higher resolution than the rest of MM5. Once the results of the simulations with the static MPDATA scheme and the MM5 scheme were shifted slightly to eliminate a slight phase difference, it was apparent that the two solutions were converging as the grid spacing decreased. Comparison of the standard resolution results indicated that the two MPDATA results were more similar to the high resolution results than were the standard resolution MM5 results. Therefore, the high resolution MPDATA was chosen as the reference solution for comparison purposes.

When the standard resolution results were compared with the "exact" solution, the dynamic MPDATA scheme produced virtually the same results with only a quarter the number of grid points. The static MPDATA scheme performed better than the standard MM5 results.

The low resolution test cases were conducted to examine the effect of reducing the number of grid points in the static and dynamic MPDATA simulations. It was found that except at very coarse resolution, the use of the dynamic MPDATA scheme produced superior results compared to the static MPDATA scheme. Even with only 18 points in each horizontal direction, the dynamic MPDATA scheme

was still advantageous. With a 12×12 grid, the dynamic and static MPDATA schemes were comparably accurate.

The passive water vapor advection case provided an initial test case using a physically realistic tracer field that was initialized to be the same as the initial water vapor field at 0 UTC March 6, 1992. Initially, the same domain was used as in the previous two test cases. However, the smoothness of the initial condition combined with the small domain and the short simulation time resulted in few gradients sharp enough to exploit the advantages of the dynamic MPDATA. Therefore, the domain size, grid spacing of the static grids, and the simulation time were all increased to 101×75 , 52 km, and 5 days respectively. This domain encompassed the continental United States. The domain and simulation time were large and long enough that sharp gradients and curvatures developed in the tracer field and significant grid stretching was realized. The grid stretching resolved multiple features of the field simultaneously and apparently superior results were obtained. There are other permutations of this case that could be run and are discussed in more detail in section 4.3.

4.2 Unique contributions

One of the goals of academic research is to gain new knowledge or to implement existing ideas or methods in novel areas. Much was learned by the investigator through this research and several unique contributions met this goal. Briefly these contributions are:

1. The MPDATA scheme was cast into a form applicable to a dynamic adaptive grid environment. Numerical experiments were conducted to gain insight into the properties of this scheme and the impact of the different MPDATA options.
2. An analytical form for limiting the grid point movement in one-dimension was developed and gave insight into the Lagrangian aspect of the dynamic adaptive grid method. This provided a method of maintaining stability for the explicit MPDATA scheme.
3. The dynamic grid adaptation component was developed and used to model passive atmospheric tracer transport. Previous work in the atmospheric science community either had restricted the problems to simple test cases to show that the DAG technique had potential, or used moving nested grids to model small scale phenomena and the grid movement was not fully automatic.

4.3 Future work

This research has produced many new questions and avenues of investigation. This section is included to document some of possibilities so that they may be easily recalled, evaluated, and pursued. Section 4.3.1 gives ideas and motivations for short term investigations that can be readily done with what has been developed. Section 4.3.2 deals with longer term projects that would require significant effort to accomplish.

4.3.1 Short term extensions

There are several tasks that could be done as a direct extension of this work.

4.3.1.1 Variations on the passive water vapor advection case

In the case performed only two of the σ levels were used to compute the composite weight function in part because it was not clear initially how to create the composite weight function from a field that was such a strong function of height. Several possibilities are:

1. The weight function could be calculated by taking the maximum weight function from each column of points. The weight function would be calculated on each σ separately. Since the tracer field is a strong function height this would weight the lower σ levels more heavily than the layers higher in the atmosphere. However, this may be desirable if the goal is to improve the water vapor transport processes.
2. The tracer quantity on each σ level could be scaled by the maximum value on that particular σ level. The composite weight function could then again be taken as the maximum value in any given column of points. This would weight each level equally, which may or may not be desirable.
3. A similar alternative would be to weight the layers by relative humidity instead of water vapor mixing ratio so that upper atmospheric layers close to the saturation point would effect the grid adaptation.
4. Since the goal is to improve the water vapor transport, the average magnitude of the water vapor flux could be used to identify the most important levels on which to base the weighting. This would not necessarily be the levels with the highest tracer values, since the magnitude of the flux is the product of wind speed and the water vapor mixing ratio. A difficulty with this approach is that this average is not known a priori. The fluxes from the first few time steps could be

used, but it would not be known if these levels would continue to have the maximum water vapor flux throughout the entire integration. Changing the levels during the integration is not recommended. Recall that the dependence on time of the grid distribution is only very weakly established through q . If the levels are changed during the integration, the grid points may try to snap to new locations, high grid point velocities could occur, and the CFL condition could be violated.

Although, investigating these different options would not be technically difficult, they would be time consuming if done thoroughly. The five day simulation took a week to run on the current computer arrangement which was a dedicated DEC alpha 3000/900 with a 275 Mhz processor and 256Mb of memory. Jobs could be submitted on this machine for up to 24 hours. Additionally, disk storage before and after the execution of the job was limited to 150 Mb of memory. This was not a week of continuous computing, but included time to stop the simulation, save data to long term storage, and restart the simulation numerous times.

4.3.1.2 Use of the RPV option in the DGAC

Recall that the DGAC was run with four iterations of the MPDATA scheme and the "third" order correction terms. It was shown in chapter 2 that the results using the recursive pseudo velocity (RPV) option were similar to those when four iterations were used and required fewer CPU cycles. Determining the feasibility of the DAG technique in a realistic model was considered a higher priority than implementing this option immediately. Therefore, a version of this option to solve the MM5 advection equation 3.30 was not developed. In the final form, it may be identical to the form given in section A.2, except that the quantity $p^* u_i$ may replace u_i . However, derivation is needed to verify this hypothesis.

4.3.1.3 Vertical grid stretching

Grid adaptation was done only in the horizontal direction. The DGAC was written so that it could perform vertical grid stretching as well. The one-dimensional grid generator from chapter 2 could be used to perform the stretching on each column of points.

The determination of the vertical weight function in each column would need to be made dependent on adjacent columns, probably through some type of averaging procedure. This is necessary to prevent extremely skewed grids in the vertical direction.

It has been suggested by Zapotocny et al. (1994) that large scale adiabatic processes are modeled better when lines of constant potential temperature (equivalently, entropy) are used as the vertical

coordinate system. Unfortunately, in the planetary boundary layer these lines do not remain monotonic, forcing a different coordinate system to be used near the Earth's surface. It could be very useful to determine if an adaptive vertical grid can naturally conform to an isentropic coordinate system away from the planetary boundary layer and in the absence of strong diabatic heating. This would be a way to use an effectively isentropic coordinate system, but only where it is applicable, eliminating the need to transition into another coordinate system near the surface.

4.3.1.4 Implementation of potentially more efficient two-dimensional grid generators

Kim and Thompson (1990) found that the elliptic grid generation system mentioned in section 2.3.1.1.3 was able to generate acceptable grids in one-third the computational time it took variational methods. This suggests that the efficiency of the two-dimensional grid generator can be improved by investigating alternative methods that were discussed in section 2.3.1.1.

4.3.2 Long term extensions

Although the investigations mentioned in the previous section require significant time to complete, there are no foreseen complicating issues that would significantly delay their completion. They are natural extensions of the work already done. It is intended that these will be done in the near future. The suggested projects in this section are much larger and the results are not so certain.

4.3.2.1 Water vapor feedback into MM5

The DGAC was driven by MM5 in a one-way mode only. The wind fields were provided to the DGAC, but the resulting tracer transport did not influence the internal MM5 computations. If the water vapor field was interpolated back to the MM5 grid and used in the MM5 convection, precipitation, evaporation, and radiation schemes, a more physically realistic model could be developed. This process could take place in two steps.

Initially, these physics processes could be performed in parallel with the current MM5 advection scheme. The prediction of precipitation, evaporation, convection, and radiation would be calculated twice. Once using the water vapor transport from the MM5 scheme and the other from the transport calculated by the dynamic MPDATA' scheme. The circulation would continue to be effected by the physics calculations from MM5. However, the precipitation, evaporation, convection, and radiation

quantities from each could be compared and if they did not diverge, it would be plausible to think that feedback from the DGAC would be possible.

The second stage would allow the parameterized quantities from the dynamic MPDATA scheme to influence the energy, pressure perturbation, and vertical momentum equations. At this point feedback from the DGAC to MM5 would be complete. Comparison of model output with the actual observations could be more justified because the DGAC would be influencing observable fields and not just a tracer. Comparisons between the standard MM5 outputs and observations and between the DGAC/MM5 and observations would give provide a measure of the usefulness of the DAG technique. .

4.3.2.2 Develop a new model based on a DAG technique

Ultimately, a whole new model could be developed using a DAG technique. This would allow all of the prognostic variables to be calculated on an adaptive grid. There are several obstacles that would need to be addressed:

1. Which quantity should be used to determine the grid adaptation. In the present work only one field was being simulated; therefore, there was no alternative but to use the tracer field as the determining variable in the grid redistribution. However, if the velocity components, pressure perturbation, temperature, and water vapor variables are all simulated on the dynamic grid, what combination of these should influence grid movement is not obvious.
2. The MPDATA scheme, which is an explicit scheme, was used in this work. MM5 used a short time step so that it could resolve fast moving waves in the model. Since the advective speeds were much smaller than these speeds, the time step could be held constant without violating the CFL condition, even though the grid spacing in some areas was much closer in the DGAC than in MM5. If the full set of equations are solved on the dynamic grid, the explicit MPDATA scheme may require for stability reasons an impractically small time step. Therefore, the MPDATA would possibly have to be modified in some way to incorporate some form of flux splitting.

APPENDIX A “THIRD-ORDER” MPDATA AND THE RECURSIVE PSEUDO VELOCITY OPTION

This appendix details the “third-order” and recursive pseudo velocity (RPV) options of the MPDATA scheme. The “third-order” option significantly reduces the error of the MPDATA option and the RPV option yields significant computational savings.

Not included in this appendix is the implementation of flux corrected transport. If the basic state of the advected field is zero, the MPDATA is monotone. However, if the basic state is not zero, dispersive ripples can develop. For tracer transport the basic state is zero and this option was not required for this work. The interpolator that was used in the three-dimensional model also employed the same monotonicity feature and is explained in section 3.3. The application of this method to MPDATA is described in appendix C.

A.1 “Third-order” MPDATA

Margolin and Smolarkiewicz (1989) and Smolarkiewicz and Margolin (1998) expand the analysis of section 2.2.2 to include some of the third-order correction terms. For the development of the third-order correction terms only, they utilize a constant velocity approximation. Note that the flow was not considered constant during the derivation of the second-order correction terms. Therefore the resulting scheme is second-order accurate because the leading order error terms are a function of the velocity gradient which is assumed to be zero during the derivation. Numerical experiments show that the inclusion of these extra terms causes the MPDATA solution to be more symmetric. These results are evident in section 2.4.1.3. Smolarkiewicz and Margolin (1998) note that this extension of MPDATA is useful in computations of passive scalar advection, and so it is applicable to this research.

Once the contravariant velocity is used in place of the physical velocity the transformed equation 2.37 is identical to an equation in a Cartesian coordinate system. Therefore, this option is developed for a Cartesian domain and can be implemented on a non-Cartesian by replacing the velocities with the

contravariant velocities. The analysis starts with the advection equation:

$$\frac{\partial q}{\partial t} + u_i \frac{\partial q}{\partial x_i} = 0 \quad (\text{A.1})$$

As in the previous two sections, consider first the temporal derivative:

$$\frac{q^{n+1} - q^n}{\Delta t} + u_i \frac{\partial q}{\partial x_i} = 0 \quad (\text{A.2})$$

A Taylor series expansion around q^n is:

$$\frac{\partial q}{\partial t} + u_i \frac{\partial q}{\partial x_i} = -\frac{1}{2}\Delta t \frac{\partial^2 q}{\partial t^2} - \frac{1}{6}\Delta t^2 \frac{\partial^3 q}{\partial t^3} + \mathcal{O}(\Delta t^3) \quad (\text{A.3})$$

Spatial approximations for $\frac{\partial^2 q}{\partial t^2}$ to $\mathcal{O}(\Delta t^2)$ and $\frac{\partial^3 q}{\partial t^3}$ to $\mathcal{O}(\Delta t)$ are needed. To develop $\frac{\partial^2 q}{\partial t^2}$ to $\mathcal{O}(\Delta t)$, equation A.3 to first-order was used:

$$\frac{\partial q}{\partial t} = -\frac{\partial}{\partial x_i} (u_i q) + \mathcal{O}(\Delta t) \quad (\text{A.4})$$

Differentiation with respect to time yields:

$$\frac{\partial^2 q}{\partial t^2} = -\frac{\partial}{\partial x_i} \left(u_i \frac{\partial q}{\partial t} \right) + \mathcal{O}(\Delta t) \quad (\text{A.5})$$

Substituting equation A.4 for $\frac{\partial q}{\partial t}$:

$$\frac{\partial^2 q}{\partial t^2} = \frac{\partial}{\partial x_i} \left(u_i u_j \frac{\partial q}{\partial x_j} \right) + \mathcal{O}(\Delta t) \quad (\text{A.6})$$

Differentiation with respect to time and substitution of equation A.4 once more yields:

$$\frac{\partial^3 q}{\partial t^3} = -\frac{\partial}{\partial x_i} \left(u_i u_j u_k \frac{\partial^2 q}{\partial x_j \partial x_k} \right) + \mathcal{O}(\Delta t) \quad (\text{A.7})$$

To develop $\frac{\partial^2 q}{\partial t^2}$ to $\mathcal{O}(\Delta t^2)$, equation A.3 is taken to second-order:

$$\frac{\partial q}{\partial t} = -u_i \frac{\partial q}{\partial x_i} - \frac{1}{2}\Delta t \frac{\partial^2 q}{\partial t^2} + \mathcal{O}(\Delta t^2) \quad (\text{A.8})$$

Substituting equation A.6 for $\frac{\partial^2 q}{\partial t^2}$ in equation A.8:

$$\frac{\partial q}{\partial t} = -\frac{\partial}{\partial x_i} (u_i q) - \frac{1}{2}\Delta t \frac{\partial}{\partial x_i} \left(u_i u_j \frac{\partial q}{\partial x_j} \right) + \mathcal{O}(\Delta t^2) \quad (\text{A.9})$$

Differentiating equation A.9 with respect to time:

$$\frac{\partial^2 q}{\partial t^2} = \frac{\partial}{\partial x_i} \left(-u_i \frac{\partial q}{\partial t} \right) - \frac{1}{2}\Delta t \frac{\partial}{\partial x_i} \left(u_i u_j \frac{\partial}{\partial x_j} \left(\frac{\partial q}{\partial t} \right) \right) + \mathcal{O}(\Delta t^2) \quad (\text{A.10})$$

Use of equations A.9 and A.4 to substitute for $\frac{\partial q}{\partial t}$ in the first and second terms respectively on the right hand side of equation A.10 yields:

$$\frac{\partial^2 q}{\partial t^2} = \frac{\partial}{\partial x_i} \left(u_i u_j \frac{\partial q}{\partial x_j} \right) + \Delta t \frac{\partial}{\partial x_i} \left(u_i u_j u_k \frac{\partial^2 q}{\partial x_j \partial x_k} \right) + \mathcal{O}(\Delta t^2) \quad (\text{A.11})$$

Equations A.7 and A.11 are used to substitute for the temporal derivatives in equation A.3:

$$\frac{\partial q}{\partial t} + u_i \frac{\partial q}{\partial x_i} = -\frac{1}{2} \Delta t \frac{\partial}{\partial x_i} \left(u_i u_j \frac{\partial q}{\partial x_j} \right) - \frac{1}{3} \Delta t^2 \frac{\partial}{\partial x_i} \left(u_i u_j u_k \frac{\partial^2 q}{\partial x_j \partial x_k} \right) + \mathcal{O}(\Delta t^3) \quad (\text{A.12})$$

If approximations for the spatial derivatives to high enough order are used, equation A.12 yields a third-order scheme in time. $u_i \frac{\partial q}{\partial x_i}$ needs to be approximated to $\mathcal{O}(\Delta x^3)$, $\frac{\partial}{\partial x_i} \left(u_i u_j \frac{\partial q}{\partial x_j} \right)$ to $\mathcal{O}(\Delta x^2)$ and since $\frac{\partial}{\partial x_i} \left(u_i u_j u_k \frac{\partial^2 q}{\partial x_j \partial x_k} \right)$ is already third-order accurate a $\mathcal{O}(\Delta x)$ approximation is adequate. Dropping the index notation in place of the (x, y, z) , (u, v, w) notation and considering the spatial discretization of the first component of $u_i \frac{\partial q}{\partial x_i}$ and $\frac{\partial}{\partial x_i} \left(u_i u_j \frac{\partial q}{\partial x_j} \right)$ separately, the donor cell approximations for these two terms are:

$$u \frac{\partial q}{\partial x} = \frac{u}{\Delta x} (q_{i,j,k} - q_{i-1,j,k}) \quad (\text{A.13})$$

$$\begin{aligned} u \frac{\partial}{\partial x} \left(u \frac{\partial q}{\partial x} + v \frac{\partial q}{\partial y} + w \frac{\partial q}{\partial z} \right) &= \frac{u^2}{\Delta x^2} (q_{i,j,k} - q_{i-1,j,k} - q_{i-1,j,k} + q_{i-2,j,k}) \\ &+ \frac{uv}{\Delta x \Delta y} (q_{i,j,k} - q_{i,j-1,k} - q_{i-1,j,k} - q_{i-1,j-1,k}) \\ &+ \frac{uw}{\Delta x \Delta z} (q_{i,j,k} - q_{i,j,k-1} - q_{i-1,j,k} - q_{i-1,j,k-1}) \end{aligned} \quad (\text{A.14})$$

where the velocities are assumed to be positive. A Taylor series expansion of equation A.13 yields:

$$\begin{aligned} \frac{u}{\Delta x} (q_{i,j,k} - q_{i-1,j,k}) &= \frac{\partial}{\partial x} (uq) - \frac{1}{2} \Delta x \frac{\partial}{\partial x} \left(u \frac{\partial q}{\partial x} \right) + \frac{1}{6} \Delta x^2 \frac{\partial}{\partial x} \left(u \frac{\partial^2 q}{\partial x^2} \right) \\ &+ (\text{v permutations}) + (\text{w permutations}) + \mathcal{O}(\Delta x^3, \Delta y^3, \Delta z^3) \end{aligned} \quad (\text{A.15})$$

When equation A.14 is expanded in a Taylor series:

$$\begin{aligned} &\frac{u^2}{\Delta x^2} (q_{i,j,k} - q_{i-1,j,k} - q_{i-1,j,k} + q_{i-2,j,k}) + \frac{uv}{\Delta x \Delta y} (q_{i,j,k} - q_{i,j-1,k} - q_{i-1,j,k} - q_{i-1,j-1,k}) \\ &+ \frac{uw}{\Delta x \Delta z} (q_{i,j,k} - q_{i,j,k-1} - q_{i-1,j,k} - q_{i-1,j,k-1}) = u^2 \left(\frac{\partial^2 q}{\partial x^2} - \Delta x \frac{\partial^3 q}{\partial x^3} \right) \\ &+ uv \left(\frac{1}{2} \frac{\partial^2 q}{\partial x \partial y} - \frac{1}{6} \Delta x \frac{\partial^3 q}{\partial x^2 \partial y} - \frac{1}{6} \Delta x \frac{\partial^3 q}{\partial x \partial y^2} \right) + uw \left(\frac{1}{2} \frac{\partial^2 q}{\partial x \partial z} - \frac{1}{6} \Delta x \frac{\partial^3 q}{\partial x^2 \partial z} - \frac{1}{6} \Delta x \frac{\partial^3 q}{\partial x \partial z^2} \right) \\ &+ (\text{v permutations}) + (\text{w permutations}) + \mathcal{O}(\Delta x^3, \Delta y^3, \Delta z^3) \end{aligned} \quad (\text{A.16})$$

where the terms not explicit given are symmetric permutations of the terms that are explicitly given in equations A.15 and A.16. When these two equations and their permutations are substituted into equation A.12 the resulting equation is:

$$\begin{aligned} \frac{\partial q}{\partial t} + u \frac{\partial q}{\partial x} + v \frac{\partial q}{\partial y} + w \frac{\partial q}{\partial z} &= \frac{1}{2} \left(|u| \Delta x \frac{\partial^2 q}{\partial x^2} + |v| \Delta y \frac{\partial^2 q}{\partial y^2} + |w| \Delta z \frac{\partial^2 q}{\partial z^2} \right) \\ &- \frac{1}{6} \left(\Delta x^2 u \frac{\partial^3 q}{\partial x^3} + \Delta y^2 v \frac{\partial^3 q}{\partial y^3} + \Delta z^2 w \frac{\partial^3 q}{\partial z^3} \right) - \frac{1}{2} \Delta t \left(u^2 \frac{\partial^2 q}{\partial x^2} + v^2 \frac{\partial^2 q}{\partial y^2} + w^2 \frac{\partial^2 q}{\partial z^2} \right. \\ &\left. + 2uv \frac{\partial^2 q}{\partial x \partial y} + 2uw \frac{\partial^2 q}{\partial x \partial z} + 2vw \frac{\partial^2 q}{\partial y \partial z} \right) + \frac{1}{2} \Delta x \Delta t \left(|u| u \frac{\partial^3 q}{\partial x^3} + |u| v \frac{\partial^3 q}{\partial x^2 \partial y} + |u| w \frac{\partial^3 q}{\partial x^2 \partial z} \right) \end{aligned}$$

$$\begin{aligned}
& + \frac{1}{2} \Delta y \Delta t \left(|v| v \frac{\partial^3 q}{\partial x^3} + |v| u \frac{\partial^3 q}{\partial y^2 \partial x} + |v| w \frac{\partial^3 q}{\partial y^2 \partial z} \right) + \frac{1}{2} \Delta z \Delta t \left(|w| w \frac{\partial^3 q}{\partial x^3} + |w| u \frac{\partial^3 q}{\partial z^2 \partial x} + |w| v \frac{\partial^3 q}{\partial z^2 \partial y} \right) \\
& - \frac{1}{3} \Delta t^2 u^2 \left(u \frac{\partial^3 q}{\partial x^3} + v \frac{\partial^3 q}{\partial x^2 \partial y} + w \frac{\partial^3 q}{\partial x^2 \partial z} \right) - \frac{1}{3} \Delta t^2 v^2 \left(v \frac{\partial^3 q}{\partial y^3} + u \frac{\partial^3 q}{\partial y^2 \partial x} + w \frac{\partial^3 q}{\partial y^2 \partial z} \right) \\
& - \frac{1}{3} \Delta t^2 w^2 \left(w \frac{\partial^3 q}{\partial z^3} + v \frac{\partial^3 q}{\partial z^2 \partial y} + u \frac{\partial^3 q}{\partial z^2 \partial x} \right) - 2 \Delta t^2 uvw \frac{\partial^3 q}{\partial z \partial y \partial x} + \mathcal{O}(\Delta t^3, \Delta x^3, \Delta y^3, \Delta z^3) \quad (\text{A.17})
\end{aligned}$$

where the terms with the absolute values were determined by repeating this analysis while assuming that the velocities were negative instead of positive. When defining the pseudo velocities there is a decision made concerning the cross derivative terms of the form $-\Delta t uv \frac{\partial^2 q}{\partial x \partial y}$ and $-2 \Delta t^2 uvw \frac{\partial^3 q}{\partial x \partial y \partial z}$. In each case a free parameter exists that allows different amounts of the terms to be accounted for in different pseudo velocities. Following the lead of Smolarkiewicz and Margolin (1998) these are chosen to be divided equally between the two and three pseudo velocities respectively. Therefore, the pseudo velocity in the x direction were defined as:

$$\begin{aligned}
U^{(1)} &= \frac{1}{2} (|U| - U^2) \frac{\Delta x}{q} \frac{\partial q}{\partial x} - \frac{1}{2} U \left(V \frac{\Delta y}{q} \frac{\partial q}{\partial y} + W \frac{\Delta z}{q} \frac{\partial q}{\partial z} \right) + \frac{U}{6} (3|U| - 2U^2 - 1) \frac{\Delta x^2}{q} \frac{\partial^2 q}{\partial x^2} \\
&+ \frac{V}{2} (|U| - 2U^2) \frac{\Delta x \Delta y}{q} \frac{\partial^2 q}{\partial x \partial y} + \frac{W}{2} (|U| - 2U^2) \frac{\Delta x \Delta z}{q} \frac{\partial^2 q}{\partial x \partial z} - \\
&\frac{2UVW}{3} \frac{\Delta y \Delta z}{q} \frac{\partial^2 q}{\partial y \partial z} \quad (\text{A.18})
\end{aligned}$$

These pseudo velocity terms can be generalized to any number of MPDATA iterations by replacing $U^{(1)}$, U , V , and W with $U^{(k+1)}$, $U^{(k)}$, $V^{(k)}$, and $W^{(k)}$ respectively. Recall when the iteration level is zero the physical velocities are used.

Finite difference approximations to $\frac{\Delta x}{2q} \frac{\partial q}{\partial x}$, $\frac{\Delta y}{q} \frac{\partial q}{\partial y}$, $\frac{\Delta z}{q} \frac{\partial q}{\partial z}$, $\frac{\Delta x^2}{q} \frac{\partial^2 q}{\partial x^2}$, $\frac{\Delta x \Delta y}{q} \frac{\partial^2 q}{\partial x \partial y}$, $\frac{\Delta x \Delta z}{q} \frac{\partial^2 q}{\partial x \partial z}$ and $\frac{\Delta y \Delta z}{q} \frac{\partial^2 q}{\partial y \partial z}$ are needed at the $i + 1/2, j, k$ location. These terms were approximated as:

$$\left. \frac{\Delta x}{2q} \frac{\partial q}{\partial x} \right|_{i+1/2, j, k} \approx \frac{q_{i+1, j, k} - q_{i, j, k}}{q_{i+1, j, k} + q_{i, j, k} + \epsilon} \quad (\text{A.19})$$

$$\left. \frac{\Delta y}{q} \frac{\partial q}{\partial y} \right|_{i+1/2, j, k} \approx \frac{q_{i+1, j+1, k} + q_{i, j+1, k} - q_{i+1, j-1, k} - q_{i, j-1, k}}{q_{i+1, j+1, k} + q_{i, j+1, k} + q_{i+1, j-1, k} + q_{i, j-1, k} + \epsilon} \quad (\text{A.20})$$

$$\left. \frac{\Delta x^2}{q} \frac{\partial^2 q}{\partial x^2} \right|_{i+1/2, j, k} \approx 2 \frac{q_{i-1, j, k} - q_{i, j, k} - q_{i+1, j, k} + q_{i+2, j, k}}{q_{i-1, j, k} + q_{i, j, k} + q_{i+1, j, k} + q_{i+2, j, k} + \epsilon} \quad (\text{A.21})$$

$$\left. \frac{\Delta x \Delta y}{q} \frac{\partial^2 q}{\partial x \partial y} \right|_{i+1/2, j, k} \approx 2 \frac{q_{i+1, j+1, k} - q_{i, j+1, k} - q_{i+1, j-1, k} + q_{i, j-1, k}}{q_{i+1, j+1, k} + q_{i, j+1, k} + q_{i+1, j-1, k} + q_{i, j-1, k} + \epsilon} \quad (\text{A.22})$$

$$\left. \frac{\Delta y \Delta z}{q} \frac{\partial^2 q}{\partial y \partial z} \right|_{i+1/2, j, k} \approx \frac{q_{i+1/2, j+1, k+1} - q_{i+1/2, j-1, k+1} - q_{i+1/2, j+1, k-1} + q_{i+1/2, j-1, k-1}}{q_{i+1/2, j+1, k+1} + q_{i+1/2, j-1, k+1} + q_{i+1/2, j+1, k-1} + q_{i+1/2, j-1, k-1} + \epsilon} \quad (\text{A.23})$$

where $q_{i+1/2, j, k}$ is the average value of $q_{i, j, k}$ and $q_{i+1, j, k}$ and ϵ is a small number to prevent division by zero when all the q values in the stencil are zero. The $\frac{\Delta z}{q} \frac{\partial q}{\partial z}$ and $\frac{\Delta x \Delta z}{q} \frac{\partial^2 q}{\partial x \partial z}$ factors are calculated

as in equations A.20 and A.22 respectively except that in the z direction instead of the y direction. In addition to approximating the values of the derivatives and q at the cell faces, the orthogonal velocities need to be approximated as well. This is done as

$$V_{i+1/2,j,k} = \frac{1}{4} (V_{i,j+1/2,k} + V_{i+1,j+1/2,k} + V_{i,j-1/2,k} + V_{i+1,j-1/2,k}) \quad (\text{A.24})$$

$$W_{i+1/2,j,k} = \frac{1}{4} (W_{i,j,k+1/2} + W_{i+1,j,k+1/2} + W_{i,j,k-1/2} + W_{i+1,j,k-1/2}) \quad (\text{A.25})$$

A.2 The recursive pseudo velocities option

As mentioned in section 2.2.1, any number greater or equal to two iterations of MPDATA could be performed. Each iteration consists of computing a new pseudo velocity and then subtracting a first-order accurate donor cell approximation. Although the order of accuracy does not increase, the error decreases and an improved answer is computed.

In the original MPDATA scheme Smolarkiewicz (1983) used the value of q from the most recent iteration in both the calculation of the pseudo velocity and the donor-cell approximation to calculate the anti-diffusive correction flux. This can be seen in equation 2.31 by the q at the k iteration level in the computation of the fluxes. This value of q is an approximation to the value of q at time level $n + 1$. Since the donor cell scheme is only first-order accurate any value of q between time levels n and $n + 1$ could be used without a loss of accuracy. Although numerical experiments show that for the two-pass MPDATA scheme using the newest value of q produced the best results, there is a reason to consider using q^n .

Because of this option, Margolin and Smolarkiewicz (1989) were able to evaluate an approximation to the sum of an infinite number of MPDATA iterations. Using this recursive pseudo velocity (RPV) form of MPDATA, a strictly two-pass scheme was developed. The first pass used the physical velocity to create a first-order approximation of the solution. A pseudo velocity that was an approximation to the sum of an infinite number of the pseudo velocities, was calculated. A donor-cell scheme was used to correct the solution in the second pass. This scheme had some attributes that made it desirable for this research that are evident in the in section 2.4.2.2.

Mathematically, the differences between the pseudo velocities of the original MPDATA scheme and the one used for the RPV is shown in the next three sets of equations. In order to implement the RPV option the pseudo velocities are not defined as in equation 2.48, rather they are redefined as:

$$U^{(k+1)} = \left(|U^{(k)}| - U^{(k)} \right) U^{(k)} A - U^{(k)} V^{(k)} B - U^{(k)} W^{(k)} C \quad (\text{A.26})$$

$$V^{(k+1)} = \left(|V^{(k)}| - V^{(k)} \right) V^{(k)} B - V^{(k)} U^{(k)} A - V^{(k)} W^{(k)} C$$

$$W^{(k+1)} = \left(|W^{(k)}| - W^{(k)}W^{(k)} \right) C - W^{(k)}U^{(k)}A - W^{(k)}V^{(k)}B$$

where the quantities A , B and C were finite difference representations at location $i + 1/2, j, k$ such that

$$\begin{aligned} A &= \frac{1}{2q^n} \frac{\partial q^n}{\partial x} \approx \frac{q_{i+1,j,k}^n - q_{i,j,k}^n}{q_{i+1,j,k}^n + q_{i,j,k}^n + \epsilon} \\ B &= \frac{1}{2q^n} \frac{\partial q^n}{\partial y} \approx \frac{q_{i,j+1,k}^n + q_{i+1,j+1,k}^n - q_{i,j-1,k}^n - q_{i+1,j-1,k}^n}{2 \left(q_{i,j+1,k}^n + q_{i+1,j+1,k}^n + q_{i,j-1,k}^n + q_{i+1,j-1,k}^n + \epsilon \right)} \\ C &= \frac{1}{2q^n} \frac{\partial q^n}{\partial z} \approx \frac{q_{i,j,k+1}^n + q_{i+1,j,k+1}^n - q_{i,j,k-1}^n - q_{i+1,j,k-1}^n}{2 \left(q_{i,j,k+1}^n + q_{i+1,j,k+1}^n + q_{i,j,k-1}^n + q_{i+1,j,k-1}^n + \epsilon \right)} \end{aligned} \quad (\text{A.27})$$

where ϵ is a small number that prevents division by zero in the case of all the $q^n = 0$. The A , B and C quantities at the other cell faces for the V and W pseudo velocities have finite difference representations that are symmetric permutations of equations A.27. Note that the q values are always taken at the n time level instead of the current iteration level as in equation 2.48.

Three important criteria are needed in order to ensure the validity of the following derivation. It is important to notice that the A , B and C values are bounded between -1 and 1 . A second point is that the magnitude of each $U^{(k+1)}$, $V^{(k+1)}$ and $W^{(k+1)}$ is less than its predecessor. The third point is that for every iteration (k) that $U^{(k)}$, $V^{(k)}$ and $W^{(k)}$ have the same sign as A , B and C respectively. The first two points are necessary to show that the sum converges. The third point is necessary so that the absolute value sign can be transferred from the velocity to the A , B and C quantities, which is necessary in order to compute the summations. Margolin and Smolarkiewicz (1989) proved the third point to be true. Therefore, equations A.26 are written as:

$$\begin{aligned} U^{(k+1)} &= |A| U^{(k)} - A U^{(k)} U^{(k)} - B U^{(k)} V^{(k)} - C U^{(k)} W^{(k)} \\ V^{(k+1)} &= |B| V^{(k)} - B V^{(k)} V^{(k)} - A V^{(k)} U^{(k)} - C V^{(k)} W^{(k)} \\ W^{(k+1)} &= |C| W^{(k)} - B W^{(k)} W^{(k)} - A W^{(k)} U^{(k)} - C W^{(k)} V^{(k)} \end{aligned} \quad (\text{A.28})$$

The goal of this section is to determine an approximation for S_u , S_v and S_w which are defined by:

$$\begin{aligned} S_u &= |A| \sum_{k=1}^{\infty} U^{(k)} - A \sum_{k=1}^{\infty} U^{(k)} U^{(k)} - B \sum_{k=1}^{\infty} \left(U^{(k)} V^{(k)} \right) - C \sum_{k=1}^{\infty} \left(U^{(k)} W^{(k)} \right) \\ S_v &= |B| \sum_{k=1}^{\infty} V^{(k)} - B \sum_{k=1}^{\infty} V^{(k)} V^{(k)} - A \sum_{k=1}^{\infty} \left(V^{(k)} U^{(k)} \right) - C \sum_{k=1}^{\infty} \left(V^{(k)} W^{(k)} \right) \\ S_w &= |C| \sum_{k=1}^{\infty} W^{(k)} - C \sum_{k=1}^{\infty} W^{(k)} W^{(k)} - A \sum_{k=1}^{\infty} \left(W^{(k)} U^{(k)} \right) - B \sum_{k=1}^{\infty} \left(W^{(k)} V^{(k)} \right) \end{aligned} \quad (\text{A.29})$$

These quantities represent the sum of an infinite number of pseudo velocity. When these quantities are used as the velocity in the donor cell step the second-order error terms are approximated. Therefore, the RPV option of the MPDATA scheme involves three steps:

1. Calculate a first-order approximation $q^{(1)}$ using the donor-cell scheme with q^n and the physical velocity.
2. Calculate S_u , S_v , and S_w using q^n and the physical velocity.
3. Calculate q^{n+1} using the donor-cell scheme with the S 's playing the role of the velocity and $q^{(1)}$.

According to Margolin and Smolarkiewicz (1989) the sums in equations A.29 can not be exactly evaluated, but they can be approximated using a power series representation in a small variable. Therefore, the following definitions are made

$$\begin{aligned}
 U^{(k)} &= a_{uu1}^{(k)}\varepsilon_u + a_{uu2}^{(k)}\varepsilon_u^2 + a_{uu3}^{(k)}\varepsilon_u^3 + \dots + \\
 &\quad a_{uv1}^{(k)}\varepsilon_u\varepsilon_v + a_{uv2}^{(k)}\varepsilon_u^2\varepsilon_v + a_{uv3}^{(k)}\varepsilon_u\varepsilon_v^2 + \dots + \\
 &\quad a_{uw1}^{(k)}\varepsilon_u\varepsilon_w + a_{uw2}^{(k)}\varepsilon_u^2\varepsilon_w + a_{uw3}^{(k)}\varepsilon_u\varepsilon_w^2 + \dots \\
 V^{(k)} &= a_{vv1}^{(k)}\varepsilon_v + a_{vv2}^{(k)}\varepsilon_v^2 + a_{vv3}^{(k)}\varepsilon_v^3 + \dots + \\
 &\quad a_{vu1}^{(k)}\varepsilon_v\varepsilon_u + a_{vu2}^{(k)}\varepsilon_v^2\varepsilon_u + a_{vu3}^{(k)}\varepsilon_v\varepsilon_u^2 + \dots + \\
 &\quad a_{vw1}^{(k)}\varepsilon_v\varepsilon_w + a_{vw2}^{(k)}\varepsilon_v^2\varepsilon_w + a_{vw3}^{(k)}\varepsilon_v\varepsilon_w^2 + \dots \\
 W^{(k)} &= a_{ww1}^{(k)}\varepsilon_w + a_{ww2}^{(k)}\varepsilon_w^2 + a_{ww3}^{(k)}\varepsilon_w^3 + \dots + \\
 &\quad a_{wu1}^{(k)}\varepsilon_w\varepsilon_u + a_{wu2}^{(k)}\varepsilon_w^2\varepsilon_u + a_{wu3}^{(k)}\varepsilon_w\varepsilon_u^2 + \dots + \\
 &\quad a_{wv1}^{(k)}\varepsilon_w\varepsilon_v + a_{wv2}^{(k)}\varepsilon_w^2\varepsilon_v + a_{wv3}^{(k)}\varepsilon_w\varepsilon_v^2 + \dots
 \end{aligned} \tag{A.30}$$

where ε_u , ε_v and ε_w are the small parameters chosen to be $U^{(1)}$, $V^{(1)}$, and $W^{(1)}$ respectively, and the sum of the a coefficients are to be determined. Not all of the possible terms were included in equation A.30 due to the form of the recursion relation A.28 and the initial conditions according to Margolin and Smolarkiewicz (1989). Because of the choice to use the first pseudo velocities as the perturbation parameters the initial conditions are:

$$a_{uu1}^{(1)} = a_{vv1}^{(1)} = a_{ww1}^{(1)} = 1 \tag{A.31}$$

while the rest are zero.

Only the sums of the a_{uu} and a_{uv} coefficients are derived as the other coefficients are symmetric permutations of these. Substitution of equations A.30 into the recursion relation A.28:

$$\begin{aligned}
U^{(k+1)} = & |A| a_{uu1}^{(k)} \varepsilon_u + \left(|A| a_{uu2}^{(k)} - A a_{uu1}^{(k)} a_{uu1}^{(k)} \right) \varepsilon_u^2 + \left(|A| a_{uu3}^{(k)} - 2A a_{uu1}^{(k)} a_{uu2}^{(k)} \right) \varepsilon_u^3 + \\
& \left(|A| a_{uv1}^{(k)} - B a_{uu1}^{(k)} a_{vv1}^{(k)} \right) \varepsilon_u \varepsilon_v + \left(|A| a_{uv2}^{(k)} - 2A a_{uu1}^{(k)} a_{uv1}^{(k)} - B a_{uu1}^{(k)} a_{vv1}^{(k)} - B a_{uu2}^{(k)} a_{vv1}^{(k)} \right) \varepsilon_u^2 \varepsilon_v + \\
& \left(|A| a_{uv3}^{(k)} - B a_{uu1}^{(k)} a_{vv2}^{(k)} - B a_{uv1}^{(k)} a_{vv1}^{(k)} \right) \varepsilon_u \varepsilon_v^2
\end{aligned} \tag{A.32}$$

Since $U^{(k+1)}$ can also be expanded as in equation A.30 the coefficients are matched:

$$\begin{aligned}
a_{uu1}^{(k+1)} &= |A| a_{uu1}^{(k)} \\
a_{uu2}^{(k+1)} &= |A| a_{uu2}^{(k)} - A a_{uu1}^{(k)} a_{uu1}^{(k)} \\
a_{uu3}^{(k+1)} &= |A| a_{uu3}^{(k)} - 2A a_{uu1}^{(k)} a_{uu2}^{(k)} \\
a_{uv1}^{(k+1)} &= |A| a_{uv1}^{(k)} - B a_{uu1}^{(k)} a_{vv1}^{(k)} \\
a_{uv2}^{(k+1)} &= |A| a_{uv2}^{(k)} - 2A a_{uu1}^{(k)} a_{uv1}^{(k)} - B a_{uu1}^{(k)} a_{vv1}^{(k)} - B a_{uu2}^{(k)} a_{vv1}^{(k)} \\
a_{uv3}^{(k+1)} &= |A| a_{uv3}^{(k)} - B a_{uu1}^{(k)} a_{vv2}^{(k)} - B a_{uv1}^{(k)} a_{vv1}^{(k)}
\end{aligned} \tag{A.33}$$

Summation of the recursion relation A.33 and application of some obvious properties yields

$$\sum_{k=1}^{\infty} a_{uu1}^{(k+1)} = |A| \sum_{k=1}^{\infty} a_{uu1}^{(k)} = \sum_{k=1}^{\infty} a_{uu1}^{(k)} - a_{uu1}^{(1)} = \sum_{k=1}^{\infty} a_{uu1}^{(k)} - 1 \tag{A.34}$$

Therefore, the infinite sum of equation A.33 is

$$\sum_{k=1}^{\infty} a_{uu1}^{(k)} = \frac{1}{1 - |A|} \tag{A.35}$$

A similar technique is used to solve for the infinite sum of a_{uu2} except that the second term of equation A.34 needs to be determined first. The infinite sum of the product of equation A.33 with itself is

$$\sum_{k=1}^{\infty} a_{uu1}^{(k+1)} a_{uu1}^{(k+1)} = A^2 \sum_{k=1}^{\infty} a_{uu1}^{(k)} a_{uu1}^{(k)} = \sum_{k=1}^{\infty} a_{uu1}^{(k)} a_{uu1}^{(k)} - 1 \tag{A.36}$$

Therefore

$$\sum_{k=1}^{\infty} a_{uu1}^{(k)} a_{uu1}^{(k)} = \frac{1}{1 - A^2} \tag{A.37}$$

Substituting equation A.37 into the infinite sum of equation A.34 yields

$$\sum_{k=1}^{\infty} a_{uu2}^{(k+1)} = |A| \sum_{k=1}^{\infty} a_{uu2}^{(k)} - \frac{A}{1 - A^2} \tag{A.38}$$

Since the initial condition for $a_{uu2} = 0$

$$\sum_{k=1}^{\infty} a_{uu2}^{(k+1)} = \sum_{k=1}^{\infty} a_{uu2}^{(k)} \tag{A.39}$$

Therefore equation 2.33 is solved for

$$\sum_{k=1}^{\infty} a_{uu2}^{(k)} = \frac{-A}{(1-A^2)(1-|A|)} \quad (\text{A.40})$$

Using these same techniques the rest of the coefficients in equation A.33 were found

$$\begin{aligned} \sum_{k=1}^{\infty} a_{uu3}^{(k)} &= \frac{2|A^3|}{(1-|A|)(1-A^2)(1-|A^3|)} \\ \sum_{k=1}^{\infty} a_{uv1}^{(k)} &= \frac{-B}{(1-|A|)(1-|AB|)} \\ \sum_{k=1}^{\infty} a_{uv2}^{(k)} &= \frac{2AB}{(1-|A|)(1-A^2|B|)} \left\{ \frac{|A|}{1-A^2} + \frac{|B|}{1-|AB|} \right\} \\ \sum_{k=1}^{\infty} a_{uv3}^{(k)} &= \frac{B^2(|A|+|B|)}{(1-|A|)(1-|AB|)(1-B^2|A|)} \end{aligned} \quad (\text{A.41})$$

The values for the a_{uw} coefficients are found from permutations of the a_{uv} coefficients. Likewise the a_v and a_w coefficients are found from the permutations of the a_u coefficients.

Unlike the iterative MPDATA scheme there is no guarantee the magnitude of the recursive pseudo velocities will be less than one. This can cause the explicit donor-cell scheme to become unstable. This can be seen in the a_u terms, since each has a $(1-|A|)$ in the denominator and $|A|$ can become close to one. Therefore, the recursive pseudo velocities S_u , S_v and S_w need to be limited. Following Margolin and Smolarkiewicz (1989) this is done as:

$$S_u = \min(|U|, |S_u|) \text{sign}(S_u) \quad (\text{A.42})$$

where the S_v and S_w recursive pseudo velocities are limited in the same way. Therefore, all of the terms in equation A.29 can be approximated. The sums of the pseudo velocities are used themselves as the pseudo velocities in the second iteration of the MPDATA scheme. This renders the scheme a strictly two step predictor corrector scheme and has results that are approximately equivalent of taking four MPDATA iterations but with fewer CPU clock cycles.

APPENDIX B DERIVATION OF MM5 GOVERNING EQUATIONS

The governing equations that are used in the MM5 model are not trivially developed. A significant amount of time was spent deriving these equations. Much was learned from the exercise and hopefully the reader will benefit from the development.

Starting with the continuity, horizontal momentum, vertical momentum, and energy equations in an evenly spaced, orthogonal projection from Haltiner (1971):

$$\frac{\partial \rho}{\partial t} + m^2 \frac{\partial}{\partial x_i} \left(\frac{\rho u_i}{m} \right) + \frac{\partial(\rho w)}{\partial z} = 0 \quad (\text{B.1})$$

$$\frac{\partial(\rho u_i)}{\partial t} + m^2 \frac{\partial}{\partial x_j} \left(\frac{\rho u_i u_j}{m} \right) + \frac{\partial(\rho u_i w)}{\partial z} = -m \frac{\partial p}{\partial x_i} + \rho f \epsilon_{ij3} u_j \quad (\text{B.2})$$

$$\frac{\partial(\rho w)}{\partial t} + m^2 \frac{\partial}{\partial x_i} \left(\frac{\rho u_i w}{m} \right) + \frac{\partial(\rho w^2)}{\partial z} = -\frac{\partial p}{\partial z} - \rho g (1 + q_c + q_r) \quad (\text{B.3})$$

$$c_p \frac{dT}{dt} - \frac{1}{\rho} \frac{dp}{dt} = \dot{Q} \quad (\text{B.4})$$

where repeated subscripts indicate summation, $i = 1, 2$, and $(x_1, x_2) = (x, y)$. The m 's are the inverse of the metric terms to account for the projection from the spherical coordinate system to the projected plane. These are traditionally referred to as map factors. The q_r and q_c are the mixing ratios of the condensed rain and cloud water suspended in the air parcel. The f is the Coriolis parameter and is defined as:

$$f = 2\Omega \sin \phi \quad (\text{B.5})$$

where Ω and ϕ are the angular velocity of the earth and latitude respectively. The ϵ_{ij3} is the permutation symbol. The total derivative is:

$$\frac{d}{dt} = \frac{\partial}{\partial t} + m u_i \frac{\partial}{\partial x_i} + w \frac{\partial}{\partial z} \quad (\text{B.6})$$

The pressure, temperature, and density variables are defined in terms of a basic state and perturbations around this basic state:

$$p(t, x, y, z) = p_o(z) + p'(t, x, y, z) \quad (\text{B.7})$$

$$T(t, x, y, z) = T_o(z) + T'(t, x, y, z) \quad (\text{B.8})$$

$$\rho(t, x, y, z) = \rho_o(z) + \rho'(t, x, y, z) \quad (\text{B.9})$$

where the basic state is in hydrostatic balance:

$$\frac{\partial p_o}{\partial z} = -\rho_o g \quad (\text{B.10})$$

MM5 utilizes a σ vertical coordinate system that has the advantage that it automatically accounts for topographic effects. Consider the transformation from the orthogonal coordinate system (t, x_i, z) into a σ coordinate system (τ, ξ_i, σ) , where σ is defined as:

$$\tau = t \quad (\text{B.11})$$

$$\xi_i = x_i \quad (\text{B.12})$$

$$\sigma = \frac{p_o - p_t}{p_s - p_t} = \frac{p_o - p_t}{p^*} \quad (\text{B.13})$$

where $p_o = p_o(z)$ is a basic state pressure, $p_s = p_s(x_i)$ is the basic state surface pressure, $p_t = \text{constant}$ is the prescribed pressure at the top of the model, and $p^* = p^*(x_i) = p_s - p_t$. Noting that σ is not a function of time:

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \tau} \quad (\text{B.14})$$

$$\frac{\partial}{\partial x_i} = \frac{\partial}{\partial \xi_i} + \frac{\partial \sigma}{\partial \xi_i} \frac{\partial}{\partial \sigma} = \frac{\partial}{\partial \xi_i} - \frac{p_o - p_t}{q^{*2}} \frac{\partial p^*}{\partial x_i} \frac{\partial}{\partial \sigma} = \frac{\partial}{\partial \xi_i} - \frac{\sigma}{p^*} \frac{\partial p^*}{\partial x_i} \frac{\partial}{\partial \sigma} \quad (\text{B.15})$$

$$\frac{\partial}{\partial z} = \frac{\partial \sigma}{\partial z} \frac{\partial}{\partial \sigma} = \frac{1}{p^*} \frac{\partial p_o}{\partial z} \frac{\partial}{\partial \sigma} = \frac{-\rho_o g}{p^*} \frac{\partial}{\partial \sigma} \quad (\text{B.16})$$

B.1 Pressure perturbation equation

The development of the pressure perturbation equation starts with equation B.1 and the ideal gas law:

$$\rho = \frac{p}{RT} \quad (\text{B.17})$$

Substituting equation B.17 into B.1:

$$\frac{\partial}{\partial t} \left(\frac{p}{RT} \right) + m^2 \frac{\partial}{\partial x_i} \left(\frac{p u_i}{mRT} \right) + \frac{\partial}{\partial z} \left(\frac{p w}{RT} \right) = 0 \quad (\text{B.18})$$

Applying the chain rule:

$$\frac{1}{T} \left[\frac{\partial p}{\partial t} + m^2 \frac{\partial}{\partial x_i} \left(\frac{p u_i}{m} \right) + \frac{\partial (p w)}{\partial z} \right] + p \left[\frac{\partial}{\partial t} \left(\frac{1}{T} \right) + m u \frac{\partial}{\partial x_i} \left(\frac{1}{T} \right) + w \frac{\partial}{\partial z} \left(\frac{1}{T} \right) \right] = 0 \quad (\text{B.19})$$

Performing the differentiations of $\frac{1}{T}$ in the second square bracketed term:

$$\frac{\partial p}{\partial t} + m^2 \frac{\partial}{\partial x_i} \left(\frac{p u_i}{m} \right) + \frac{\partial (p w)}{\partial z} - \frac{p}{T} \frac{dp}{dt} = 0 \quad (\text{B.20})$$

If the diabatic heating term \dot{Q} is neglected in equation B.4 it can be written as:

$$\frac{dT}{dt} = \frac{1}{\rho c_p} \frac{dp}{dt} \quad (\text{B.21})$$

Substitution of equation B.21 and the ideal gas model into equation B.20 gives:

$$\frac{\partial p}{\partial t} + m^2 \frac{\partial}{\partial x_i} \left(\frac{p u_i}{m} \right) + \frac{\partial(pw)}{\partial z} - \frac{R}{c_p} \frac{dp}{dt} = 0 \quad (\text{B.22})$$

Recall:

$$\frac{R}{c_p} = \frac{c_p - c_v}{c_p} = 1 - \frac{1}{\gamma} = \frac{\gamma - 1}{\gamma} \quad (\text{B.23})$$

where $\gamma = \frac{c_p}{c_v}$. Substitution of equation B.23 into B.22 then yields:

$$\frac{\partial p}{\partial t} + m^2 \frac{\partial}{\partial x_i} \left(\frac{p u_i}{m} \right) + \frac{\partial(pw)}{\partial z} + \frac{1 - \gamma}{\gamma} \frac{dp}{dt} = 0 \quad (\text{B.24})$$

Applying the chain rule:

$$\frac{\partial p}{\partial t} + m^2 p \frac{\partial}{\partial x_i} \left(\frac{u_i}{m} \right) + m u_i \frac{\partial p}{\partial x_i} + p \frac{\partial w}{\partial z} + w \frac{\partial p}{\partial z} + \frac{1 - \gamma}{\gamma} \frac{dp}{dt} = 0 \quad (\text{B.25})$$

which can be written as:

$$\frac{dp}{dt} + m^2 p \frac{\partial}{\partial x_i} \left(\frac{u_i}{m} \right) + p \frac{\partial w}{\partial z} + \frac{1 - \gamma}{\gamma} \frac{dp}{dt} = 0 \quad (\text{B.26})$$

which can be rearranged as:

$$\frac{dp}{dt} + \gamma p \left[m^2 \frac{\partial}{\partial x_i} \left(\frac{u_i}{m} \right) + \frac{\partial w}{\partial z} \right] = 0 \quad (\text{B.27})$$

Using equations B.7 and B.10, equation B.27 becomes:

$$\frac{dp'}{dt} + \gamma p \left[m^2 \frac{\partial}{\partial x_i} \left(\frac{u_i}{m} \right) + \frac{\partial w}{\partial z} \right] - \rho_o g w = 0 \quad (\text{B.28})$$

Transforming equation B.28 into the σ coordinate system using equations B.14 through B.16 and multiplying by p^* :

$$\begin{aligned} & p^* \frac{\partial p'}{\partial \tau} + m u_i p^* \frac{\partial p'}{\partial \xi_i} + m u_i p^* \frac{\partial \sigma}{\partial x_i} \frac{\partial p'}{\partial \sigma} + w p^* \frac{\partial \sigma}{\partial z} \frac{\partial p'}{\partial \sigma} + \\ & p^* \gamma p \left[m^2 \frac{\partial}{\partial \xi_i} \left(\frac{u_i}{m} \right) + m^2 \frac{\partial \sigma}{\partial x_i} \frac{\partial}{\partial \sigma} \left(\frac{u_i}{m} \right) + \frac{\partial \sigma}{\partial z} \frac{\partial w}{\partial \sigma} \right] - p^* \rho_o g w = 0 \end{aligned} \quad (\text{B.29})$$

Noting that $p^* = p^*(x_i)$ only, it follows that:

$$\frac{\partial p^*}{\partial \sigma} = \frac{1}{\frac{\partial \sigma}{\partial z}} \frac{\partial p^*}{\partial z} = 0 \quad (\text{B.30})$$

Using the chain rule equation B.29 can be written:

$$\frac{\partial(p^* p')}{\partial \tau} + m^2 \frac{\partial}{\partial \xi_i} \left(\frac{p^* p' u_i}{m} \right) - m^2 p' \frac{\partial}{\partial \xi_i} \left(\frac{p^* u_i}{m} \right) +$$

$$\begin{aligned}
& \frac{\partial}{\partial \sigma} \left(m u_i p^* p' \frac{\partial \sigma}{\partial x_i} \right) - p' \frac{\partial}{\partial \sigma} \left(m u_i p^* \frac{\partial \sigma}{\partial x_i} \right) + \\
& \quad \frac{\partial}{\partial \sigma} \left(w p^* p' \frac{\partial \sigma}{\partial z} \right) - p' \frac{\partial}{\partial \sigma} \left(w p^* \frac{\partial \sigma}{\partial z} \right) + \\
p^* \gamma p \left[m^2 \frac{\partial}{\partial \xi_i} \left(\frac{u_i}{m} \right) + m^2 \frac{\partial \sigma}{\partial x_i} \frac{\partial}{\partial \sigma} \left(\frac{u_i}{m} \right) + \frac{\partial \sigma}{\partial z} \frac{\partial w}{\partial \sigma} \right] - p^* \rho_0 g w &= 0 \tag{B.31}
\end{aligned}$$

Rearranging and collecting terms:

$$\begin{aligned}
& \frac{\partial(p^* p')}{\partial \tau} + m^2 \frac{\partial}{\partial \xi_i} \left(\frac{p^* p' u_i}{m} \right) + \frac{\partial}{\partial \sigma} \left[p^* p' \left(m u_i \frac{\partial \sigma}{\partial x_i} + w \frac{\partial \sigma}{\partial z} \right) \right] \\
& \quad - p' \left\{ m^2 \frac{\partial}{\partial \xi_i} \left(\frac{p^* u_i}{m} \right) + \frac{\partial}{\partial \sigma} \left[p^* \left(m u_i \frac{\partial \sigma}{\partial x_i} + w \frac{\partial \sigma}{\partial z} \right) \right] \right\} \\
+ p^* \gamma p \left[m^2 \frac{\partial}{\partial \xi_i} \left(\frac{u_i}{m} \right) + m^2 \frac{\partial \sigma}{\partial x_i} \frac{\partial}{\partial \sigma} \left(\frac{u_i}{m} \right) + \frac{\partial \sigma}{\partial z} \frac{\partial w}{\partial \sigma} \right] - p^* \rho_0 g w &= 0 \tag{B.32}
\end{aligned}$$

Noting that:

$$\frac{d\sigma}{dt} = \dot{\sigma} = m u_i \frac{\partial \sigma}{\partial x_i} + w \frac{\partial \sigma}{\partial z} = -m u_i \frac{\sigma}{p^*} \frac{\partial p^*}{\partial x_i} - \frac{\rho_0 g}{p^*} \tag{B.33}$$

and defining:

$$\text{DIV} = m^2 \frac{\partial}{\partial \xi_i} \left(\frac{p^* u_i}{m} \right) + \frac{\partial(p^* \dot{\sigma})}{\partial \sigma} \tag{B.34}$$

one can then substitute equations B.33 and B.34 into equation B.32:

$$\begin{aligned}
& \frac{\partial(p^* p')}{\partial \tau} + m^2 \frac{\partial}{\partial \xi_i} \left(\frac{p^* p' u_i}{m} \right) + \frac{\partial(p^* p' \dot{\sigma})}{\partial \sigma} - p' \text{DIV} + \\
p^* \gamma p \left[m^2 \frac{\partial}{\partial \xi_i} \left(\frac{u_i}{m} \right) + m^2 \frac{\partial \sigma}{\partial x_i} \frac{\partial}{\partial \sigma} \left(\frac{u_i}{m} \right) + \frac{\partial \sigma}{\partial z} \frac{\partial w}{\partial \sigma} \right] - p^* \rho_0 g w &= 0 \tag{B.35}
\end{aligned}$$

Using equations B.15 and B.16:

$$\frac{\partial \sigma}{\partial x_i} = -\frac{\sigma}{p^*} \frac{\partial p^*}{\partial x_i} = -\frac{\sigma}{p^*} \left(\frac{\partial p^*}{\partial \xi_i} + \frac{\partial \sigma}{\partial x_i} \frac{\partial p^*}{\partial \sigma} \right) = -\frac{\sigma}{p^*} \frac{\partial p^*}{\partial \xi_i} \tag{B.36}$$

$$\frac{\partial \sigma}{\partial z} = -\frac{\rho_0 g}{p^*} \tag{B.37}$$

Substituting equations B.36 and B.37 into equation B.35 yields the form of the pressure perturbation equation found in Grell et al. (1995):

$$\begin{aligned}
\frac{\partial(p^* p')}{\partial \tau} &= -m^2 \frac{\partial}{\partial \xi_i} \left(\frac{p^* p' u_i}{m} \right) - \frac{\partial(p^* p' \dot{\sigma})}{\partial \sigma} + p' \text{DIV} \\
&\quad - m^2 p^* \gamma p \left[\frac{\partial}{\partial \xi_i} \left(\frac{u_i}{m} \right) - \frac{\sigma}{m p^*} \frac{\partial p^*}{\partial \xi_i} \frac{\partial u_i}{\partial \sigma} \right] + \gamma p \rho_0 g \frac{\partial w}{\partial \sigma} + p^* \rho_0 g w \tag{B.38}
\end{aligned}$$

B.2 Horizontal momentum equations

Starting with equation B.2 and expanding it via the chain rule:

$$\rho \left[\frac{\partial u_i}{\partial t} + m u_j \frac{\partial u_i}{\partial x_j} + w \frac{\partial u_i}{\partial z} \right] + u_i \left[\frac{\partial \rho}{\partial t} + m^2 \frac{\partial}{\partial x_j} \left(\frac{\rho u_j}{m} \right) + \frac{\partial(\rho w)}{\partial z} \right] = -m \frac{\partial p}{\partial x_i} + \rho f \epsilon_{ij3} u_j \tag{B.39}$$

The terms in the second set of square brackets is the continuity equation and is identically zero:

$$\frac{\partial u_i}{\partial t} + m u_j \frac{\partial u_i}{\partial x_j} + w \frac{\partial u_i}{\partial z} = -\frac{m}{\rho} \frac{\partial p}{\partial x_i} + f \epsilon_{ij3} u_j \quad (\text{B.40})$$

Differentiating equation B.7 and recalling that $p_o = p_o(z)$ only:

$$\frac{\partial p}{\partial x_i} = \frac{\partial p_o}{\partial x_i} + \frac{\partial p'}{\partial x_i} = \frac{\partial p'}{\partial x_i} \quad (\text{B.41})$$

Therefore, equation B.40 becomes:

$$\frac{\partial u_i}{\partial t} + m u_j \frac{\partial u_i}{\partial x_j} + w \frac{\partial u_i}{\partial z} = -\frac{m}{\rho} \frac{\partial p'}{\partial x_i} + f \epsilon_{ij3} u_j \quad (\text{B.42})$$

Converting to σ coordinates and following the same procedure on the advective terms as was done for the pressure equation in equations B.29 through B.35, yields:

$$\frac{\partial(p^* u_i)}{\partial \tau} + m^2 \frac{\partial}{\partial \xi_j} \left(\frac{p^* u_i u_j}{m} \right) + \frac{\partial(p^* u_i \dot{\sigma})}{\partial \sigma} - u_i \text{DIV} = -\frac{m p^*}{\rho} \left(\frac{\partial p'}{\partial \xi_i} + \frac{\partial \sigma}{\partial \xi_i} \frac{\partial p'}{\partial \sigma} \right) + p^* f \epsilon_{ij3} u_j \quad (\text{B.43})$$

where $\dot{\sigma}$ and DIV are defined in equations B.33 and B.34 respectively. Using equation B.36 to substitute for $\frac{\partial \sigma}{\partial x_i}$ and rearranging yields the equation in Grell et al. (1995):

$$\frac{\partial(p^* u_i)}{\partial \tau} = -m^2 \frac{\partial}{\partial \xi_j} \left(\frac{p^* u_i u_j}{m} \right) - \frac{\partial(p^* u_i \dot{\sigma})}{\partial \sigma} + u_i \text{DIV} - \frac{m p^*}{\rho} \left(\frac{\partial p'}{\partial \xi_i} - \frac{\sigma}{p^*} \frac{\partial p^*}{\partial \xi_i} \frac{\partial p'}{\partial \sigma} \right) + p^* f \epsilon_{ij3} u_j \quad (\text{B.44})$$

B.3 Vertical momentum equation

Care must be taken in deriving the vertical momentum equation because the terms involved are typically much smaller than terms in the other prognostic equations. The effects of water vapor, although ignored in the other equations, can be significant in the vertical momentum equation, thus recognizing the addition of the cloud and rain water in equation B.3.

By removing continuity from the vertical momentum equation in the same manner as it was removed from the horizontal momentum equation, equation B.3 becomes:

$$\frac{\partial w}{\partial t} + m u_i \frac{\partial w}{\partial x_i} + w \frac{\partial w}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial z} - g(1 + q_c + q_r) \quad (\text{B.45})$$

Differentiating equation B.7:

$$\frac{\partial p}{\partial z} = \frac{\partial p_o}{\partial z} + \frac{\partial p'}{\partial z} \quad (\text{B.46})$$

Substitution into equation B.45 yields:

$$\frac{\partial w}{\partial t} + m u_i \frac{\partial w}{\partial x_i} + w \frac{\partial w}{\partial z} = -\frac{1}{\rho} \frac{\partial p_o}{\partial z} - \frac{1}{\rho} \frac{\partial p'}{\partial z} - g - g(q_c + q_r) \quad (\text{B.47})$$

Several first order approximations are used in the development of the basic state pressure gradient term. Considering equation B.9 and the $-\frac{1}{\rho} \frac{\partial p_o}{\partial z}$ term:

$$-\frac{1}{\rho} \frac{\partial p_o}{\partial z} = -\frac{1}{\rho_o + \rho'} \frac{\partial p_o}{\partial z} \approx -\frac{\rho_o - \rho'}{\rho_o^2} \frac{\partial p_o}{\partial z} \quad (\text{B.48})$$

Substituting the hydrostatic relationship in equation B.10:

$$-\frac{1}{\rho} \frac{\partial p_o}{\partial z} \approx \left(1 - \frac{\rho'}{\rho_o}\right) g \approx \left(1 - \frac{\rho'}{\rho}\right) g = g - \frac{\rho'}{\rho} g = g + \left(\frac{\rho_o}{\rho} - 1\right) g \quad (\text{B.49})$$

In order to account for the gaseous mixture of water vapor and air, the ideal gas law based on the virtual temperature T_v can be used.

$$\rho = \frac{p}{RT_v} \quad (\text{B.50})$$

According to Wallace and Hobbs (1977) the virtual temperature is a fictitious temperature that permits using the gas constant for dry air in the ideal gas law, yielding the same density as if one used the actual temperature and gas constant for dry air. Virtual temperature can be approximated as:

$$T_v \approx T(1 + 0.608q_v) \quad (\text{B.51})$$

where q_v is the mixing ratio of water vapor. Substituting equation B.50:

$$\begin{aligned} -\frac{1}{\rho} \frac{\partial p_o}{\partial z} &\approx g + \left(\frac{p_o T_v}{T_o p} - 1\right) g \\ &= g + \frac{p_o}{p} \left(\frac{T_v}{T_o} - \frac{p}{p_o}\right) g \\ &= g + \frac{\rho_o T_o}{\rho T} \left(1 + \frac{T'_v}{T_o} - 1 - \frac{p'}{p_o}\right) g \\ &= g + \frac{\rho_o}{\rho} \left(\frac{T'_v}{T} - \frac{T_o p'}{T p_o}\right) g \end{aligned} \quad (\text{B.52})$$

Where T'_v is defined as

$$T_v = T_o + T'_v \quad (\text{B.53})$$

Substitution of equation B.53 into equation B.47 yields:

$$\frac{\partial w}{\partial t} + m u_i \frac{\partial w}{\partial x_i} + w \frac{\partial w}{\partial z} = -\frac{1}{\rho} \frac{\partial p'}{\partial z} + g \frac{\rho_o}{\rho} \left[\frac{T'_v}{T} - \frac{T_o p'}{T p_o}\right] - g(q_c + q_r) \quad (\text{B.54})$$

Transforming equation B.54 into σ coordinates, multiplying by p^* , and applying the chain rule (as was done for the advection terms in the pressure and horizontal momentum equations) yields the form of the vertical momentum equation given in Grell et al. (1995):

$$\frac{\partial(p^* w)}{\partial \tau} = -m^2 \frac{\partial}{\partial \xi_i} \left(\frac{p^* w u_i}{m}\right) - \frac{\partial(p^* w \dot{\sigma})}{\partial \sigma} + w \text{DIV} + p^* g \frac{\rho_o}{\rho} \left[\frac{1}{p^*} \frac{\partial p'}{\partial \sigma} + \frac{T'_v}{T} - \frac{T_o p'}{T p_o}\right] - g(q_c + q_r) \quad (\text{B.55})$$

B.4 Temperature equation

The prognostic equation for temperature was developed from the energy equation B.4. Rearranging this equation, expanding the total derivative $\frac{dT}{dt}$ and substituting equation B.7:

$$\frac{\partial T}{\partial t} + mu_i \frac{\partial T}{\partial x_i} + w \frac{\partial T}{\partial z} = \frac{1}{\rho c_p} \frac{dp_o}{dt} + \frac{1}{\rho c_p} \frac{dp'}{dt} + \frac{\dot{Q}}{c_p} \quad (\text{B.56})$$

Expanding $\frac{dp_o}{dt}$, recalling that $p_o = p_o(z)$ only and using the fact that basic state is in hydrostatic equilibrium as noted in equation B.10:

$$\frac{dp_o}{dt} = \frac{\partial p_o}{\partial t} + mu_i \frac{\partial p_o}{\partial x_i} + w \frac{\partial p_o}{\partial z} = w \frac{\partial p_o}{\partial z} = -\rho_o g w \quad (\text{B.57})$$

Equation B.57 is substituted into equation B.56. The same steps that were applied to the advection terms in all of the other equations are once again applied. These include: transformation into σ coordinates; multiplication by p^* ; and application of the chain rule. This results in the equation reported by Grell et al. (1995):

$$\frac{\partial(p^*T)}{\partial \tau} = -m^2 \frac{\partial}{\partial \xi_j} \left(\frac{p^* u_j T}{m} \right) - \frac{\partial(p^* \dot{\sigma} T)}{\partial \sigma} + T \text{DIV} + \frac{1}{\rho c_p} \left(p^* \frac{dp'}{d\tau} - \rho_o w p^* g \right) + p^* \frac{\dot{Q}}{c_p} \quad (\text{B.58})$$

where

$$\frac{d}{d\tau} = \frac{\partial}{\partial \tau} + mu_i \frac{\partial}{\partial \xi_i} + \dot{\sigma} \frac{\partial}{\partial \sigma} \quad (\text{B.59})$$

APPENDIX C A HIGHER ORDER MONOTONE INTERPOLATION SCHEME

The interpolator used in this work was developed by Smolarkiewicz and Grell (1992) and versions of the code that they wrote were examined to determine how to practically implement it. Modification of the interpolator to meet the needs of this work required understanding it thoroughly. In order to aid those that might wish to use this interpolator for future work, its detailed development of it is included here.

C.1 Theory

Given a field $q(x_i)$ on an evenly spaced grid \bar{x}_i with grid spacing of unity, the goal of any interpolation scheme is to represent this field on some other set of points \hat{x}_i that lie within the region bounded but not necessarily coincident with \bar{x}_i . By applying Stoke's theorem, the difference between the field value at the desired location $q(\hat{x})$ and an adjacent known location $q(\bar{x})$ is:

$$q(\hat{x}_i) - q(\bar{x}_i) = \int_C \frac{\partial q(x_i)}{\partial x_j} dx_j \quad (\text{C.1})$$

where C represents an arbitrary path connecting the two points \hat{x}_i and \bar{x}_i . This arbitrary contour can be taken as a line segment:

$$x_i(\tau) = -(\bar{x}_i - \hat{x}_i)\tau + \bar{x}_i \quad (\text{C.2})$$

where the parameter τ varies from 0 to 1. Note that $x_i(\bar{x}_i, 0) = \bar{x}_i$ and $x_i(\hat{x}_i, 1) = \hat{x}_i$. Defining $\bar{q}(\tau) = q(x(\bar{x}_i, \tau))$ and taking the derivative of equation C.2 converts equation C.1 into:

$$\bar{q}(1) = \bar{q}(0) + \int_0^1 \frac{\partial \bar{q}(\tau)}{\partial x_j} (\hat{x}_j - \bar{x}_j) d\tau \quad (\text{C.3})$$

Noting that $(\hat{x}_j - \bar{x}_j)$ is a constant, equation C.3 can be written as

$$\bar{q}(1) = \bar{q}(0) + \int_0^1 \frac{\partial}{\partial x_j} [\bar{q}(\tau) (\hat{x}_j - \bar{x}_j)] d\tau \quad (\text{C.4})$$

Defining $U_i = (\bar{x}_i - \hat{x}_i)$ and noting that

$$\bar{q}(1) - \bar{q}(0) = \int_0^1 \frac{\partial \bar{q}}{\partial \tau} d\tau \quad (\text{C.5})$$

C.4 is the formal integral of the equation

$$\frac{\partial \bar{q}}{\partial \tau} + \frac{\partial}{\partial x_i} (U_i \bar{q}) = 0 \quad (\text{C.6})$$

This casts the interpolation problem into a constant velocity advection problem. Forward in time schemes can solve such problems with zero phase error. One such scheme is the second-order Lax-Wendroff scheme as described in Smolarkiewicz and Grell (1992). In the atmospheric science community, constant velocity advection on a staggered grid it is referred to as the Crowley (1968) advection scheme. Tremback et al. (1987) has extended these schemes up to tenth order.

C.1.1 Second order Tremback scheme

Consider a one-dimensional advection equation with constant velocity

$$\frac{\partial q}{\partial t} = -u \frac{\partial q}{\partial x} = -\frac{\partial F}{\partial x} \quad (\text{C.7})$$

where $F = qu$. Note that the conservative and non-conservative forms are equivalent since the flow is non-divergent. Using a second order finite difference approximation of this equation with $U_i = \frac{\Delta t}{\Delta x} u$ will solve the interpolation problem. An advection scheme for constant velocity flow that is second-order in both space and time problem can be developed by considering the discretization:

$$q^{n+1} = q^n - \frac{\Delta t}{\Delta x} (F_{i+1/2}^{n+1/2} - F_{i-1/2}^{n+1/2}) + \mathcal{O}(\Delta t^2, \Delta x^2) \quad (\text{C.8})$$

where the fluxes $F_{i+1/2}^{n+1/2}$ and $F_{i-1/2}^{n+1/2}$ are the fluxes at the right and left boundaries respectively. Note that they are taken at time level $n + 1/2$ so that the approximation of the time derivative is second order accurate. Expressing the $F^{n+1/2}$ in a Taylor series expansion

$$F^{n+1/2} = F^n + \frac{1}{2} \Delta t \frac{\partial F}{\partial t} + \mathcal{O}(\Delta t^2) \quad (\text{C.9})$$

Since u is constant, the original equation C.7 can be multiplied by u to yield

$$\frac{\partial F}{\partial t} = -u \frac{\partial F}{\partial x} \quad (\text{C.10})$$

Substituting equation C.10 into C.9

$$F^{n+1/2} = F^n - \frac{1}{2} \Delta t u \frac{\partial F}{\partial x} + \mathcal{O}(\Delta t^2) \quad (\text{C.11})$$

Expressing $\frac{\partial F}{\partial x}$ in a second-order central difference about $i - 1/2$ and using the average value of F_{i-1} and F_i , which is a second-order approximation to $F_{i-1/2}$, gives the left flux as:

$$F_{i-1/2}^{n+1/2} = \frac{1}{2} (F_i^n + F_{i-1}^n) - \frac{1}{2} \frac{\Delta t}{\Delta x} u (F_i^n - F_{i-1}^n) + \mathcal{O}(\Delta t^2, \Delta x^2) \quad (\text{C.12})$$

The right flux $F_{i+1/2}^{n+1/2}$ can be obtained by adding one to the spatial index i in equation C.12. With the left and right fluxes determined, equation C.8 was used to solve for

$$q_i^{n+1} = q_i^n - \left(F_{i+1/2}^{n+1/2} - F_{i-1/2}^{n+1/2} \right) \quad (\text{C.13})$$

Therefore, this scheme is second order accurate in time and space.

The interpolator was written such that the known values are in integer space, $\Delta x_i = \Delta x = 1$. Since $\Delta t = 1$ when interpolating the value q_i^{n+1} , $\frac{\Delta t}{\Delta x} u = U_i$ of equation C.6.

The central differences of this scheme are dispersive in nature and therefore not necessarily monotonic. In order to achieve monotonicity the flux corrected transport method of Boris and Book (1973) was used.

C.2 Flux corrected transport

The flux corrected transport method of achieving monotonicity is explained for a two-dimensional case, since this will minimize the complexity, but not neglect the extensions of the multidimensional problem. Any first order solution to the tracer transport problem is diffusive and thus monotone. The problem of spurious oscillations comes from higher order corrections to the first order solutions. The advection problem is represented as:

$$q^{n+1} = q^n - (F_{i+1/2,j} - F_{i-1/2,j} + F_{i,j+1/2} - F_{i,j-1/2}) \quad (\text{C.14})$$

Where the $F_{i,j}$ terms represent the numerical fluxes at the cell boundaries. The fluxes can be represented as

$$F_{i,j} = FL_{i,j} + A_{i,j} \quad (\text{C.15})$$

where $FL_{i,j}$ and $A_{i,j}$ represent first order and the higher order corrections or "anti-diffusive" fluxes respectively. Spurious oscillations occur when a scheme overestimates the $A_{i,j}$ flux. Therefore, if this flux can be limited, any scheme can be made monotonic. The problem can be viewed as

$$q_{i,j}^{n+1} = Q_{i,j}^{n+1} - (A_{i+1/2,j} - A_{i-1/2,j} + A_{i,j+1/2} - A_{i,j-1/2}) \quad (\text{C.16})$$

where Q^{n+1} is the first order solution of the problem at the new time level. Since first order schemes are monotonic it is assumed that

$$q_{i,j}^{\max} \geq Q_{i,j}^{n+1} \geq q_{i,j}^{\min} \quad (\text{C.17})$$

where q^{\max} and q^{\min} are the limiters of Zalesak (1979):

$$q_{i,j}^{\max} = \max(Q_{i,j}^{n+1}, Q_{i-1,j}^{n+1}, Q_{i+1,j}^{n+1}, Q_{i,j-1}^{n+1}, Q_{i,j+1}^{n+1}, q_{i,j}^n, q_{i+1,j}^n, q_{i-1,j}^n, q_{i,j+1}^n, q_{i,j-1}^n) \quad (\text{C.18})$$

$$q_{i,j}^{\min} = \min(Q_{i,j}^{n+1}, Q_{i-1,j}^{n+1}, Q_{i+1,j}^{n+1}, Q_{i,j-1}^{n+1}, Q_{i,j+1}^{n+1}, q_{i,j}^n, q_{i+1,j}^n, q_{i-1,j}^n, q_{i,j+1}^n, q_{i,j-1}^n)$$

The problem has been reduced to limiting the A fluxes to some value \bar{A} , such that

$$q_{i,j}^{\max} \geq Q_{i,j}^{n+1} - (\bar{A}_{i+1/2,j} - \bar{A}_{i-1/2,j} + \bar{A}_{i,j+1/2} - \bar{A}_{i,j-1/2}) \geq q_{i,j}^{\min} \quad (\text{C.19})$$

where

$$\bar{A}_{i,j} = c_{i,j} A_{i,j} \text{ where } 0 \leq c_{i,j} \leq 1 \quad (\text{C.20})$$

The divergence of the \bar{A} fluxes can be written as

$$- (\bar{A}_{i+1/2,j} - \bar{A}_{i-1/2,j} + \bar{A}_{i,j+1/2} - \bar{A}_{i,j-1/2}) = \bar{A}_{i,j}^{\text{in}} - \bar{A}_{i,j}^{\text{out}} \quad (\text{C.21})$$

where $\bar{A}_{i,j}^{\text{in}}$ and $\bar{A}_{i,j}^{\text{out}}$ are the sums of the incoming and outgoing limited fluxes respectively:

$$\begin{aligned} \bar{A}_{i,j}^{\text{in}} &= c_{i-1/2,j} \max(A_{i-1/2,j}, 0) + c_{i,j-1/2} \max(A_{i,j-1/2}, 0) - \\ & c_{i+1/2,j} \min(A_{i+1/2,j}, 0) - c_{i,j+1/2} \min(A_{i,j+1/2}, 0) \end{aligned} \quad (\text{C.22})$$

$$\begin{aligned} \bar{A}_{i,j}^{\text{out}} &= c_{i+1/2,j} \max(A_{i+1/2,j}, 0) + c_{i,j+1/2} \max(A_{i,j+1/2}, 0) - \\ & c_{i-1/2,j} \min(A_{i-1/2,j}, 0) - c_{i,j-1/2} \min(A_{i,j-1/2}, 0) \end{aligned}$$

Therefore equation C.19 can be written

$$q_{i,j}^{\max} \geq Q_{i,j}^{n+1} + \bar{A}_{i,j}^{\text{in}} - \bar{A}_{i,j}^{\text{out}} \geq q_{i,j}^{\min} \quad (\text{C.23})$$

Since $\bar{A}_{i,j}^{\text{in}}$ can only increase the value of $Q_{i,j}^{n+1}$ and $\bar{A}_{i,j}^{\text{out}}$ only decrease it, an equivalent statement to equation C.23 is

$$\begin{aligned} q_{i,j}^{\max} - Q_{i,j}^{n+1} &\geq c_{i-1/2,j} \max(A_{i-1/2,j}, 0) + c_{i,j-1/2} \max(A_{i,j-1/2}, 0) - \\ & c_{i+1/2,j} \min(A_{i+1/2,j}, 0) - c_{i,j+1/2} \min(A_{i,j+1/2}, 0) \end{aligned} \quad (\text{C.24})$$

$$\begin{aligned} Q_{i,j}^{n+1} - q_{i,j}^{\min} &\geq c_{i+1/2,j} \max(A_{i+1/2,j}, 0) + c_{i,j+1/2} \max(A_{i,j+1/2}, 0) - \\ & c_{i-1/2,j} \min(A_{i-1/2,j}, 0) - c_{i,j-1/2} \min(A_{i,j-1/2}, 0) \end{aligned} \quad (\text{C.25})$$

where the right hand side of the inequalities are $\bar{A}_{i,j}^{\text{in}}$ and $\bar{A}_{i,j}^{\text{out}}$ respectively. The inequalities will still hold if the maximum $c_{i,j}$ is taken in each equation above. Defining

$$A_{i,j}^{\text{in}} = \max(A_{i-1/2,j}, 0) + \max(A_{i,j-1/2}, 0) - \min(A_{i+1/2,j}, 0) - \min(A_{i,j+1/2}, 0) \quad (\text{C.26})$$

$$A_{i,j}^{\text{out}} = \max(A_{i+1/2,j}, 0) + \max(A_{i,j+1/2}, 0) - \min(A_{i-1/2,j}, 0) - \min(A_{i,j-1/2}, 0)$$

where care should be taken not to confuse the $\tilde{A}_{i,j}^{\text{in/out}}$ with the $A_{i,j}^{\text{in/out}}$ quantities. Using these definitions, inequalities C.24 and C.25 are equivalent to

$$\begin{aligned} q_{i,j}^{\text{max}} - Q_{i,j}^{n+1} &\geq \max(c_{i-1/2,j}, c_{i,j-1/2}, c_{i+1/2,j}, c_{i,j+1/2}) A_{i,j}^{\text{in}} \\ Q_{i,j}^{n+1} - q_{i,j}^{\text{min}} &\geq \max(c_{i-1/2,j}, c_{i,j-1/2}, c_{i+1/2,j}, c_{i,j+1/2}) A_{i,j}^{\text{out}} \end{aligned}$$

Dividing through by $A_{i,j}^{\text{in}}$ and $A_{i,j}^{\text{out}}$ respectively

$$\begin{aligned} \frac{q_{i,j}^{\text{max}} - Q_{i,j}^{n+1}}{A_{i,j}^{\text{in}} + \epsilon} &\geq \max(c_{i-1/2,j}, c_{i,j-1/2}, c_{i+1/2,j}, c_{i,j+1/2}) \quad (\text{C.27}) \\ \frac{Q_{i,j}^{n+1} - q_{i,j}^{\text{min}}}{A_{i,j}^{\text{out}} + \epsilon} &\geq \max(c_{i-1/2,j}, c_{i,j-1/2}, c_{i+1/2,j}, c_{i,j+1/2}) \end{aligned}$$

where a small number ϵ has been added to $A_{i,j}^{\text{in}}$ and $A_{i,j}^{\text{out}}$ in case either or both is zero. Defining two new positive definite quantities

$$\begin{aligned} \beta_{i,j}^{\uparrow} &= \frac{q_{i,j}^{\text{max}} - Q_{i,j}^{n+1}}{A_{i,j}^{\text{in}} + \epsilon} \quad (\text{C.28}) \\ \beta_{i,j}^{\downarrow} &= \frac{Q_{i,j}^{n+1} - q_{i,j}^{\text{min}}}{A_{i,j}^{\text{out}} + \epsilon} \end{aligned}$$

therefore from inequalities C.27

$$\max(c_{i-1/2,j}, c_{i,j-1/2}, c_{i+1/2,j}, c_{i,j+1/2}) \leq \min(\beta_{i,j}^{\uparrow}, \beta_{i,j}^{\downarrow}) \quad (\text{C.29})$$

However, $c_{i+1/2,j} \equiv c_{i+1-1/2,j}$ and $c_{i-1/2,j} \equiv c_{i-1+1/2,j}$ and likewise in the j direction. This means that

$$\begin{aligned} c_{i+1/2,j} &\leq \min(1, \beta_{i,j}^{\uparrow}, \beta_{i,j}^{\downarrow}, \beta_{i+1,j}^{\uparrow}, \beta_{i+1,j}^{\downarrow}) \quad (\text{C.30}) \\ c_{i,j+1/2} &\leq \min(1, \beta_{i,j}^{\uparrow}, \beta_{i,j}^{\downarrow}, \beta_{i,j+1}^{\uparrow}, \beta_{i,j+1}^{\downarrow}) \end{aligned}$$

The maximum is made 1 because of the restriction in equation C.20. If $A_{i+1/2,j}$ or $A_{i,j+1/2}$ is positive it does not contribute to $A_{i,j}^{\text{in}}$ and subsequently to $\beta_{i,j}^{\uparrow}$. Nor does it contribute to $A_{i+1,j}^{\text{out}}$, $A_{i,j+1}^{\text{out}}$, $\beta_{i+1,j}^{\downarrow}$ and $\beta_{i,j+1}^{\downarrow}$. Therefore, these can be eliminated from inequalities C.30:

$$\begin{aligned} c_{i+1/2,j} &= \min(1, \beta_{i,j}^{\downarrow}, \beta_{i+1,j}^{\uparrow}) \text{ if } A_{i+1/2,j} > 0 \quad (\text{C.31}) \\ c_{i,j+1/2} &= \min(1, \beta_{i,j}^{\downarrow}, \beta_{i,j+1}^{\uparrow}) \text{ if } A_{i,j+1/2} > 0 \end{aligned}$$

Likewise, if $A_{i+1/2,j}$ or $A_{i,j+1/2}$ is negative

$$\begin{aligned} c_{i+1/2,j} &= \min(1, \beta_{i,j}^{\uparrow}, \beta_{i+1,j}^{\downarrow}) \text{ if } A_{i+1/2,j} < 0 \quad (\text{C.32}) \\ c_{i,j+1/2} &= \min(1, \beta_{i,j}^{\uparrow}, \beta_{i,j+1}^{\downarrow}) \text{ if } A_{i,j+1/2} < 0 \end{aligned}$$

Combining these results yields the corrected anti-diffusive flux as

$$\begin{aligned}\tilde{A}_{i+1/2,j} &= \min\left(1, \beta_{i,j}^{\downarrow}, \beta_{i+1,j}^{\uparrow}\right) \max\left(A_{i+1/2,j}, 0\right) + \min\left(1, \beta_{i,j}^{\uparrow}, \beta_{i+1,j}^{\downarrow}\right) \min\left(A_{i+1/2,j}, 0\right) \quad (\text{C.33}) \\ \tilde{A}_{i,j+1/2} &= \min\left(1, \beta_{i,j}^{\downarrow}, \beta_{i,j+1}^{\uparrow}\right) \max\left(A_{i,j+1/2}, 0\right) + \min\left(1, \beta_{i,j}^{\uparrow}, \beta_{i,j+1}^{\downarrow}\right) \min\left(A_{i,j+1/2}, 0\right)\end{aligned}$$

C.2.1 Application of flux corrected transport to the interpolator

In the case of the interpolator, since the velocity is always constant (non-divergent) and also one-dimensional, the limiters can be chosen as

$$\begin{aligned}q_{i,j}^{\max} &= \max\left(Q_{i,j}^{n+1}, q_{i,j}^n, q_{i+1,j}^n, q_{i-1,j}^n\right) \quad (\text{C.34}) \\ q_{i,j}^{\min} &= \min\left(Q_{i,j}^{n+1}, q_{i,j}^n, q_{i+1,j}^n, q_{i-1,j}^n\right)\end{aligned}$$

The first order flux was calculated using the donor cell scheme.

$$\begin{aligned}Q_{i-1/2,j}^{n+1} &= \max\left(u_{i-1/2}, 0\right) q_{i-1}^n + \min\left(u_{i-1/2}, 0\right) q_i^n \quad (\text{C.35}) \\ Q_{i+1/2,j}^{n+1} &= \max\left(u_{i+1/2}, 0\right) q_i^n + \min\left(u_{i+1/2}, 0\right) q_{i+1}^n\end{aligned}$$

The second order Tremback et al. (1987) scheme is used to calculate a higher order fluxes, $FH_{i-1/2,j}^{n+1}$. The $A_{i+1/2}$ anti-diffusive fluxes are obtained by subtracting the first-order solution from the second order solution

$$A_{i+1/2} = FH_{i-1/2,j}^{n+1} - Q_{i-1/2,j}^{n+1} \quad (\text{C.36})$$

Once the anti-diffusive fluxes are calculated, they are limited using the above procedure and then added back to the first order solution yielding an interpolated value with no over- or undershoots.

C.2.2 Application of flux corrected transport to MPDATA

Although the option of using flux corrected transport (FCT) with the MPDATA scheme was available. implemented, tested and verified to work, it was an expensive option that was found to be unnecessary for the cases studied. However, for other fields that do not have a basic state value of zero the MPDATA is not monotone and FCT is necessary if monotonicity is to be preserved.

The pseudo velocities $U_{i+1/2,j,k}^{(1)}$, $V_{i,j+1/2,k}^{(1)}$ and $W_{i,j,k+1/2}^{(1)}$ defined in equations 2.48 are modified to ensure that the second-order corrections are not too great. The higher order correction terms are easy to identify since the first order solution was calculated first and then the second order corrections are made to it. Therefore, the higher order fluxes for the MPDATA scheme are

$$A_{i+1/2,j,k} = \max\left(U_{i+1/2,j,k}^{(1)}, 0\right) Q_{i,j,k}^{n+1} + \min\left(U_{i+1/2,j,k}^{(1)}, 0\right) Q_{i+1,j,k}^{n+1} \quad (\text{C.37})$$

$$A_{i,j+1/2,k} = \max(V_{i,j+1/2,k}^{(1)}, 0) Q_{i,j,k}^{n+1} + \min(V_{i,j+1/2,k}^{(1)}, 0) Q_{i,j+1,k}^{n+1} \quad (\text{C.38})$$

$$A_{i,j,k+1/2} = \max(W_{i,j,k+1/2}^{(1)}, 0) Q_{i,j,k}^{n+1} + \min(W_{i,j,k+1/2}^{(1)}, 0) Q_{i,j,k+1}^{n+1} \quad (\text{C.39})$$

Since the corrected anti-diffusive flux is

$$\tilde{A}_{i+1/2,j,k} = U_{i+1/2,j,k}^{(1)} Q_{i,j,k}^{n+1} \quad (\text{C.40})$$

$$\tilde{A}_{i,j+1/2,k} = V_{i,j+1/2,k}^{(1)} Q_{i,j,k}^{n+1} \quad (\text{C.41})$$

$$\tilde{A}_{i,j,k+1/2} = W_{i,j,k+1/2}^{(1)} Q_{i,j,k}^{n+1} \quad (\text{C.42})$$

where once again $Q_{i,j,k}^{n+1}$ is the first-order solution at time step $n + 1$, the $Q_{i,j,k}^{n+1}$ can be divided out of the equation, resulting in the flux-corrected pseudo-velocities,

$$\begin{aligned} [U_{i+1/2,j,k}^{(1)}]^{\text{mon}} &= \min(1, \beta_{i,j,k}^{\downarrow}, \beta_{i+1,j,k}^{\uparrow}) \max(U_{i+1/2,j,k}^{(1)}, 0) \\ &+ \min(1, \beta_{i,j,k}^{\uparrow}, \beta_{i+1,j,k}^{\downarrow}) \min(U_{i+1/2,j,k}^{(1)}, 0) \end{aligned} \quad (\text{C.43})$$

$$\begin{aligned} [V_{i,j+1/2,k}^{(1)}]^{\text{mon}} &= \min(1, \beta_{i,j,k}^{\downarrow}, \beta_{i,j+1,k}^{\uparrow}) \max(V_{i,j+1/2,k}^{(1)}, 0) \\ &+ \min(1, \beta_{i,j,k}^{\uparrow}, \beta_{i,j+1,k}^{\downarrow}) \min(V_{i,j+1/2,k}^{(1)}, 0) \end{aligned} \quad (\text{C.44})$$

$$\begin{aligned} [W_{i,j,k+1/2}^{(1)}]^{\text{mon}} &= \min(1, \beta_{i,j,k}^{\downarrow}, \beta_{i,j,k+1}^{\uparrow}) \max(W_{i,j,k+1/2}^{(1)}, 0) \\ &+ \min(1, \beta_{i,j,k}^{\uparrow}, \beta_{i,j,k+1}^{\downarrow}) \min(W_{i,j,k+1/2}^{(1)}, 0) \end{aligned} \quad (\text{C.45})$$

where $\beta_{i,j,k}^{\uparrow}$ and $\beta_{i+1,j,k}^{\downarrow}$ are

$$\beta_{i,j,k}^{\uparrow} = \frac{q_{i,j,k}^{\text{max}} - q_{i,j,k}^n}{A_{i,j,k}^{\text{in}} + \epsilon} \quad (\text{C.46})$$

$$\beta_{i,j,k}^{\downarrow} = \frac{q_{i,j,k}^n - q_{i,j,k}^{\text{min}}}{A_{i,j,k}^{\text{out}} + \epsilon} \quad (\text{C.47})$$

and $A_{i,j,k}^{\text{in}}$ and $A_{i,j,k}^{\text{out}}$ are

$$\begin{aligned} A_{i,j,k}^{\text{in}} &= \max(U_{i-1/2,j,k}^{(1)}, 0) Q_{i-1,j,k}^{n+1} + \max(V_{i,j-1/2,k}^{(1)}, 0) Q_{i,j-1,k}^{n+1} \\ &+ \max(W_{i,j,k-1/2}^{(1)}, 0) Q_{i,j,k-1}^{n+1} - \min(U_{i+1/2,j,k}^{(1)}, 0) Q_{i+1,j,k}^{n+1} \\ &- \min(V_{i,j+1/2,k}^{(1)}, 0) Q_{i,j+1,k}^{n+1} - \min(W_{i,j,k+1/2}^{(1)}, 0) Q_{i,j,k+1}^{n+1} \end{aligned} \quad (\text{C.48})$$

$$\begin{aligned} A_{i,j,k}^{\text{out}} &= \max(U_{i+1/2,j,k}^{(1)}, 0) Q_{i,j,k}^{n+1} + \max(V_{i,j+1/2,k}^{(1)}, 0) Q_{i,j,k}^{n+1} \\ &+ \max(W_{i,j,k+1/2}^{(1)}, 0) Q_{i,j,k}^{n+1} - \min(U_{i-1/2,j,k}^{(1)}, 0) Q_{i-1,j,k}^{n+1} \\ &- \min(V_{i,j-1/2,k}^{(1)}, 0) Q_{i,j-1,k}^{n+1} - \min(W_{i,j,k-1/2}^{(1)}, 0) Q_{i,j,k-1}^{n+1} \end{aligned} \quad (\text{C.49})$$

MPDATA will not produce undershoots as long as the background field value is zero, due to the nature of the donor-cell scheme. However, if the background field is a positive constant higher order

wiggles can appear. If the monotonicity feature is enabled, these oscillations can be eliminated. This is illustrated in figure C.1 through the use of the two dimensional rotating cone problem. Panel A shows the initial profile. The rest of the panels depict the solution after one revolution. Panel B shows the solution when a background value of zero is used while the monotonicity feature has not been enabled. In this case, no dispersive ripples are evident. The computational result in panel C shows dispersive ripples because the background field was set at 1.0 and the flux corrected transport was not employed. The last panel (D) shows the results a simulation with the background value set to 1.0 and the monotonicity feature enabled. The dispersive ripples have not occurred in this last panel. Additionally, the solution in panel D has less diffusion than the MPDATA solution in panel B where the background value was zero. For this reason Smolarkiewicz and Margolin (1998) advocates adding a large constant to the tracer field, computing the advection with FCT and then subtracting the same constant. However, due to the extra computational time required to add FCT (12% to compute the results in panel D compared to those in panel B), this option was not implemented in this work.

C.2.3 Interpolator implementation in integer space

Although the theory behind the interpolator is straight forward, the implementation is somewhat complicated especially in two- and three-dimensions. This section will describe the implementation of the interpolator in integer space.

C.2.4 One-dimensional interpolator

Although a one-dimensional interpolator is not used anywhere in the model, the two- and three-dimensional interpolators use the logic of the one-dimensional interpolator. Therefore, for simplicity the details will be described for the one-dimensional version.

Given values on an evenly spaced grid \bar{x} with grid spacing of one, the goal is to interpolate to a point \hat{x} that lies somewhere in the domain, but not necessarily on any grid point. Since the nearest point could be close to the boundary of the domain, the boundary is extended by a distance that is half the order of the interpolation scheme. Therefore, if a second-order scheme is used, one extra point is added to the beginning and the end of the domain. The value at these points is set equal to the value at the boundary so as to avoid creating new extrema from extrapolation.

Since the known points are in integer space, the `nint` intrinsic Fortran 90 function was used to determine the array index of the nearest point. The “velocity” is computed by subtracting the interpolation point from this nearest data point.

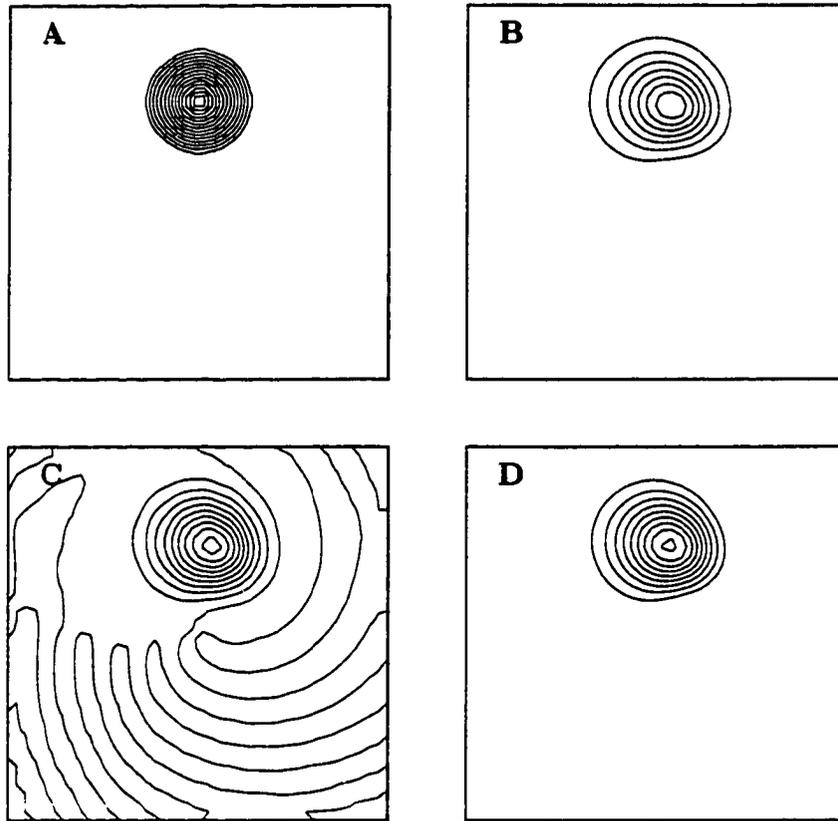


Figure C.1 Effect of flux corrected transport on the MPDATA scheme. The cone height in all figures is four greater than the base value. All solutions are after one revolution. (A)Initial Condition. (B)Solution with base values of zero without flux corrected transport. (C)Solution with base value of one without flux corrected transport. (D)Solution with base value of one with flux corrected transport.

If the monotonicity constraint is not enforced the Tremback et al. (1987) scheme is used to determine the left and right boundary fluxes and the final solution is simply the value at the nearest point minus the difference between the right and left boundary fluxes as shown in equation C.11.

If the monotonicity constraint is enforced, the first-order and high order fluxes are calculated using the donor-cell and Tremback schemes respectively. The anti-diffusive fluxes are calculated as the differences between the first-order and the high order fluxes. The low order solution is determined from the value at the nearest point minus the difference between the right and left low order fluxes. From this low order solution the value at the nearest grid point along with the values at the points adjacent to the nearest grid point the q^{\max} and q^{\min} are determined. This is followed by the determination of the β^{\uparrow} and β^{\downarrow} values. With these values the anti-diffusive fluxes are limited as described in the previous section. Finally the interpolated value are determined as the low order solution less the difference between the right and left fluxes. The code is written so a variable field can be interpolated to an arbitrary number of points with one call to the interpolator.

C.2.5 Two and three-dimensional interpolators

The implementation of the two-dimensional grid generator is built upon successive implementations of the one-dimensional interpolator. In order to determine the value at a point on a two-dimensional plane, the values along a line passing through this point are required. This is illustrated in figure C.2. Initially the values at the intermediate points (o's) are determined and then these values are used to interpolate to the final \hat{x} points (x's). The size of the stencil depended on the order of the interpolation scheme. Since all the interpolations are second-order accurate, the nine point stencil depicted in figure C.2 is used. Four one-dimensional interpolations are required for each two-dimensional interpolation.

The three-dimensional interpolation is implemented just like the two-dimensional except two sets of intermediate points are required. The first set of interpolations is to a plane. The second set of is to a line and the final interpolation is to the point. Since second-order interpolation was used, a 27 point stencil is needed. The number of one-dimensional interpolations involved in a three-dimensional domain is 13. Nine of these interpolations are in the first set onto a plane. Four are in the second set to a line. The final one is from the line to a point. Therefore, the cost of interpolating in three-dimensions is 13 times higher than in one-dimension and 3.25 times higher than two-dimensions. Therefore, effort was made to minimize the three-dimensional interpolations that were required.

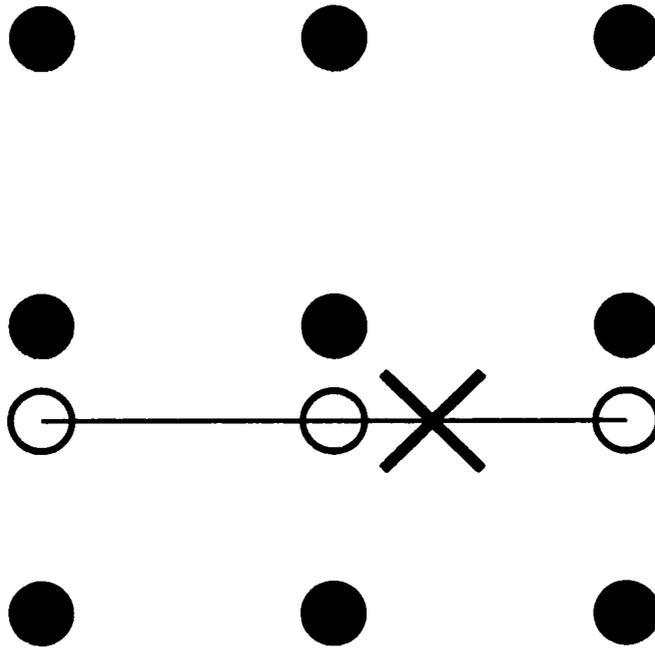


Figure C.2 Interpolation stencil for the second order two-dimensional interpolator. ●'s represent known points, × represents the point to be interpolated to and the ○'s represent temporary interpolation points

APPENDIX D MM5 PRECISION AND DGAC STRUCTURE

Although MM5 was developed in FORTRAN 77 (F77) the improved syntax of Fortran 90 (F90) made its use desirable for the development of the DGAC. In order to eliminate possible inconsistencies with calling a F90 subroutine from an F77 program, MM5 was compiled using an F90 compiler. A comparison between the results of a 24 hour simulation where MM5 was compiled under F77 and the same simulation using the F90 compiler showed no differences between the binary output files.

Since MM5 was originally run on a Cray computer which uses a 64-bit single-precision real variable, MM5 was written using single precision reals. However, this work was run a 32-bit DEC alpha 3000/900 computer with a 275 Mhz processor and 250 Mb of memory. Figure D.1 shows the vertical velocity and temperature time series for a point in the center of the domain for single and double precision calculations. Although the differences are not extreme, in order to be prudent, it was decided to perform all calculations using the same precision as on the Cray (double precision). Differences will grow with time as would be expected when a numerical forecast is perturbed about a reference simulation.

Table D.1 describes the modules that made up the DGAC. The `advect_mod.f90` component consisted of an initialization, boundary condition, restart and update subroutines. These were the only DGAC subroutines that could be called from MM5.

Table D.1 Logical break down of dynamic grid advection component.

Module Name	Purpose
<code>pr_mod.f90</code>	Defines the precision of the real variables and contains a subroutine to calculate a small number ϵ using that precision.
<code>param_mod.f90</code>	Size parameters of the MM5 domain and dynamic grid advection component.
<code>metric_mod.f90</code>	Routines to calculate the metric terms and finite differences for derivative approximations.
<code>interp_mod.f90</code>	Interpolation routines
<code>grid_mod.f90</code>	Grid generation routines
<code>advect_mod.f90</code>	Dynamic grid adaptation component initialization and driver
<code>dvlpmnt.f90</code>	A test and validation program.

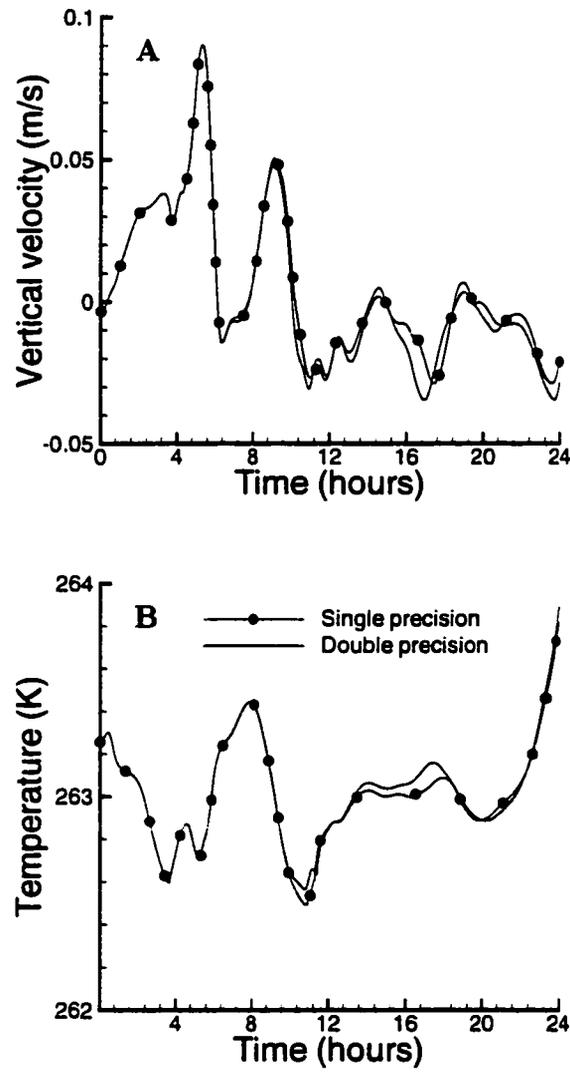


Figure D.1 Comparison of single and double precision computations of the center point for a 24 hour simulation starting at 0UTC on March 6, 1992. A) Vertical velocity; B) temperature

BIBLIOGRAPHY

- Anderson, D. A. and Rai, M. M. (1982). The use of solution adaptive grids in solving partial differential equations. In Thompson, J. F., editor, *Numerical Grid Generation*, New York. North-Holland.
- Anthes, R. A. and Warner, T. T. (1979). Development of hydrodynamic models suitable for air pollution and other mesometeorological studies. *Monthly Weather Review*, 106:1045-1078.
- Arakawa, A. and Lamb, V. R. (1977). Computational design of the basic dynamical process of the ucla general circulation model. *Methods in Computational Physics*, 31:173-265.
- Behrens, J. (1996). An adaptive semi-lagrangian advection scheme and its parallelization. *Monthly Weather Review*, 124:2386-2395.
- Berger, M. and Oliger, J. (1984). Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53:484-512.
- Boris, J. P. and Book, D. L. (1973). Flux-corrected transport, I. SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics*, 11:38-69.
- Brackbill, J. U. and Saltzman, J. S. (1982). Adaptive zoning and singular problems in two dimensions. *Journal of Computational Physics*, 46:342-368.
- Crowley, C. J. (1968). Numerical advection experiments. *Monthly Weather Review*, 96(1):1-11.
- Dannenhoffer, J. I. and Baron, J. R. (1985). Grid adaption for the 2-D Euler equations. Technical Report Paper 85-0484, American Institute of Aeronautics and Astronautics.
- Dietachmayer, G. S. (1992). Application of continuous dynamic grid adaptation techniques to meteorological modeling. Part II: Efficiency. *Monthly Weather Review*, 120(8):1707-1722.
- Dietachmayer, G. S. and Droegemeier, K. K. (1992). Application of continuous dynamic grid adaptation techniques to meteorological modeling. Part I: Basic formulation and accuracy. *Monthly Weather Review*, 120(8):1675-1706.

- Dorfi, E. and Drury, L. O. (1987). Simple adaptive grids for 1-d initial value problems. *Journal of Computational Physics*, 69:175-195.
- Dudhia, J. (1993). A nonhydrostatic version of the penn state-ncar mesoscale model: Validation tests and simulation of an atlantic cyclone and cold front. *Monthly Weather Review*, 121:1493-1513.
- Dudhia, J., Gill, D., Guo, Y.-R., Hansen, D., and Manning, K. (1997). *PSU/NCAR Mesoscale Modeling System Tutorial Class Notes: MM5 Modeling System Version 2*. Mesoscale and Microscale Meteorological Division; National Center for Atmospheric Research, Boulder.
- Dwyer, H. (1984). Grid adaption for problems in fluid dynamics. *AIAA Journal*, 22:1705-1712.
- Dwyer, H. A., Kee, R. J., and Sanders, B. R. (1980). Adaptive grid methods for problems in fluid mechanics and heat transfer. *AIAA Journal*, 10:1205-1212.
- Dwyer, H. A., Smooke, M. D., and Kee, R. J. (1982). Adaptive gridding for finite difference solutions to heat and mass transfer problems. In Thompson, J. F., editor, *Numerical Grid Generation*, New York. North-Holland.
- Eiseman, P. R. (1985). Alternating direction adaptive grid generation. *AIAA Journal*, 23:551-560.
- Eiseman, P. R. (1987). Adaptive grid generation. *Computer methods in applied mechanics and engineering*, 64:321-376.
- Fiedler, B. H. and Trapp, R. J. (1993). A fast dynamic grid adaptation scheme for meteorological flows. *Monthly Weather Review*, 121:2879-2888.
- Finley, C. A., Cotton, W. R., and A., P. R. (1998a). Numerical simulation of two tornadoes produced by a high-precipitation supercell. In *19th conference on severe local storms*, pages 206-209. American Meteorological Society. Boston.
- Finley, C. A., Cotton, W. R., and A., P. R. (1998b). Secondary vortex development in a tornado vortex produced by a simulated supercell thunderstorm. In *19th conference on severe local storms*, pages 359-362. American Meteorological Society. Boston.
- Gnoffo, P. (1982). Vectorized, finite-volume, adaptive-grid algorithm for navier-stokes. In Thompson, J. F., editor, *Numerical Grid Generation*, New York. North-Holland.

- Grell, G. A., Dudhia, J., and Stauffer, D. R. (1995). A description of the fifth-generation Penn State/NCAR mesoscale model (MM5). Technical Report 398, National Center for Atmospheric Research, Boulder.
- Haagenson, P. L., Dudhia, J., Stauffer, D. R., and Grell, G. A. (1994). The Penn State/NCAR mesoscale model (MM5) source code documentation. Technical Report 392, National Center for Atmospheric Research, Boulder.
- Haltiner, G. J. (1971). *Numerical Weather Prediction*. John Wiley and Sons, Inc., New York.
- Hawken, D., Gottlieb, J., and Hansen, J. (1991). Review of some adaptive node-movement techniques in finite-element and finite-difference solutions of partial differential equations. *Journal of Computational Physics*, 95:254-302.
- Hindman, R. G. and Spencer, J. (1983). A new approach to truly adaptive grid generation. Technical Report Paper 83-0450, American Institute of Aeronautics and Astronautics. Presented at the AIAA 21st Aerospace Sciences Meeting, Reno, Nevada, January 10-13, 1983.
- Holcomb, J. and Hindman, R. G. (1984). Development of a dynamically adaptive grid method for multidimensional problems. Technical Report Paper 84-1668, American Institute of Aeronautics and Astronautics. Presented at the AIAA/ASME/SAE/ASEE 24th Joint Propulsion Conference, Boston, Massachusetts, July 11-13, 1988.
- Iselin, J. P. and Gutowski, W. J. (1997). Water vapor layers in storm-fest rawinsonde observations. *Monthly Weather Review*, 125:1954-1963.
- Kalnay, E., Kanamitsu, R., and R., K. (1996). The NCEP/NCAR 40-year reanalysis project. *Bulletin of the American Meteorological Society*, 77:437-471.
- Kim, J. K. and Thompson, J. R. (1990). Three-dimensional adaptive grid generation on a composite-block grid. *AIAA Journal*, 38(8):470-477.
- Margolin, L. G. and Smolarkiewicz, P. K. (1989). Antidiffusive velocities for multipass donor cell advection. Technical Report UCID-21866, Lawrence Livermore National Laboratory, Livermore, CA.
- Miller, K. (1981). Moving finite elements II. *SIAM Journal on Numerical Analysis*, 18(6):1033-1057.

- Miller, K. and Miller, R. N. (1981). Moving finite elements I. *SIAM Journal on Numerical Analysis*, 18(6):1019–1032.
- Nakahashi, K. and Deiwert, G. S. (1986). Three-dimensional adaptive grid method. *AIAA Journal*, 24:948–954.
- Newell, R. and Coauthors (1996). Vertical fine-scale atmospheric structure measured from NASA DC-8 during PEM-West A. *Journal of Geophysical Research*, 101:1943–1960.
- Newell, R., Newell, N. E., Zhu, Y., and Scott, C. (1992). Tropospheric rivers?—a pilot study. *Geophysical Research Letters*, 12:2401–2404.
- Oden, J., Stouboulis, T., and Devloo, P. (1987). An adaptive finite element strategy for complex flow problems. Technical Report Paper 87-0557, American Institute of Aeronautics and Astronautics.
- Prusa, J. M. (1997). Personal communication.
- Prusa, J. M. (1999). Personal communication.
- Prusa, J. M., Smolarkiewicz, P. K., and Garcia, R. R. (1996). Propagation and breaking at high altitudes of gravity waves excited by tropospheric forcing. *Journal of the Atmospheric Sciences*, 53(1):2186–2216.
- Skamarock, W. and Klemp, J. B. (1993). Adaptive grid refinement for two-dimensional and three-dimensional nonhydrostatic atmospheric flow. *Monthly Weather Review*, 121:788–804.
- Skamarock, W., Weisman, M. L., and Klemp, J. B. (1994). Three-dimensional evolution of simulated long-lived squall lines. *Journal of The Atmospheric Sciences*, 51:2563–2583.
- Slater, J. W. (1992). *A dynamically adaptive mesh method for internal flows*. PhD dissertation, Iowa State University, Ames, IA.
- Slater, J. W. and Liou, M. S. (1995). Approach for dynamic grids. *AIAA Journal*, 33:63–68.
- Smagorinsky, J., Manabe, S., and Hailoway Jr., J. L. (1965). Numerical results from a nine-level general circulation model of the atmosphere. *Monthly Weather Review*, 93:727–768.
- Smolarkiewicz, P. K. (1983). A simple positive definite advection scheme with small implicit diffusion. *Monthly Weather Review*, 111:479–486.

- Smolarkiewicz, P. K. (1984). A fully multidimensional positive definite advection transport algorithm with small implicit diffusion. *Journal of Computational Physics*, 54:325–362.
- Smolarkiewicz, P. K. and Clark, T. L. (1986). The multidimensional positive definite advection transport algorithm: Further development and applications. *Journal of Computational Physics*, 67:396–438.
- Smolarkiewicz, P. K. and Grell, G. A. (1992). A class of monotone interpolation schemes. *Journal of Computational Physics*, 101(2):431–440.
- Smolarkiewicz, P. K. and Margolin, Len, G. (1998). MPDATA: A finite-difference solver for geophysical flows. *Journal of Computational Physics*, 140:459–480.
- Thompson, J. F. (1985). A survey of dynamically-adaptive grids in the numerical solution of partial differential equations. *Applied Numerical Mathematics*, 1:3–27.
- Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W. (1985). *Numerical Grid Generation: Foundations and Applications*. Elsevier Science Publishing Co., Inc., New York.
- Tomlin, A., Berzens, M., Ware, J., Smith, J., and Pilling, M. (1997). On the use of adaptive gridding methods for modelling chemical transport from multi-scale sources. *Atmospheric Environment*, 31:2945–2959.
- Tremback, C. J., Powell, J., Cotton, W. R., and Pielke, R. A. (1987). The forward-in-time upstream advection scheme: Extension to higher orders. *Monthly Weather Review*, 115:540–555.
- Wallace, J. M. and Hobbs, Peter, V. (1977). *Atmospheric Science, An Introductory Survey*. Academic Press. Inc.. San Diego.
- Weinstock, R. (1974). *Calculus of Variations*. Dover Publications, Inc., New York.
- Winkler, K.-H. A., Mihalas, D., and Norman, M. L. (1985). Adaptive-grid methods with asymmetric time-filtering. *Computer Physics Communications*, 36:121–140.
- Winslow, A. M. (1967). Numerical solution of the quasilinear Poisson equation in a nonuniform triangular mesh. *Journal of Computational Physics*, 1:149–172.
- Yanenko, N., Lisseikin, V. D., and Kovenia, V. M. (1979). The method of the solution of gas dynamical problems in moving meshes. *Lecture Notes in Physics*, 91:48–61.
- Yao, L. S. and Prusa, J. (1989). Melting and freezing. *Advances in Heat Transfer*, 19:1–95.

- Zalesak, Steven, T. (1979). Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31:335-362.
- Zapotocny, T. H., Johnson, D. R., and Reames, F. M. (1994). Development and initial test of the University of Wisconsin global isentropic-sigma model. *Monthly Weather Review*, 122:2160-2178.