Inner Products For Discrete Vector Fields and Accuracy of Mimetic Finite Difference Methods

James Hyman, Mikhail Shashkov
Los Alamos National Laboratory
T-7, MS-B284
Los Alamos NM 87545
and
Stanly Steinberg
Department of Mathematics and Statistics
University of New Mexico
Albuquerque NM 87131

January 20, 2000

Contents

1 Introduction **4**

2 Motivation **6**
2.1 SOM in One Dimension 6
2.1.1 A New Inner Product 8
2.1.2 Accuracy Analysis 8
2.2 SOM in Tensor Product Grids 10
2.2.1 Accuracy of the Tensor Grid Gradient 12

3 The Inner Products in General 2-D Cells **13**
3.1 Bilinear Interpolation of a Cell 14
3.2 Cell Geometry 15
3.3 Vector Fields in Local Coordinates 17
3.4 Degrees of Freedom and Inner Product 17
3.5 The Standard Inner Product 18
3.6 The New Inner Product 19
3.6.1 The Interpolation Problem 19
3.6.2 Formulas for the Cell Inner Product 21

4 SOM in General Grids **22**

5 Numerical Results **24**

List of Tables

5.1 Problem 1: Accuracy for Smooth Solution and Smooth Grid 27
5.2 Problem 2: Accuracy for Smooth Solution and Piece-Wise Smooth Grid . . 27
5.3 Problem 3: Accuracy for Smooth Solution and Random Grid 27
5.4 Problem 4: Accuracy for Piece-Wise Smooth Solution and Smooth Grid . . 27
5.5 Problem 5: Accuracy for Piece-Wise Smooth Solution and Random Grid . . 28

List of Figures

2.1 The Data on a Cell 11
3.1 A Two-Dimensional Cell 13
3.2 Tangent and Normal Vectors 15
3.3 Local Basis of Tangents and Normals 16
3.4 Triangles 18
4.1 Stencils for standard and new discrete gradients 23
5.1 The Test Grids 25

Abstract

The support operators method of discretizing partial differential equations produces discrete analogs of continuum initial boundary value problems that exactly satisfy discrete conservation laws analogous to those satisfied by the continuum system. Thus the stability of the method is assured, but currently there is no theory that predicts the accuracy of the method. In this paper, we numerically investigate how the accuracy, particularly the accuracy of the fluxes, depends on the definition of the inner product for discrete vector fields.

We introduce two different discrete inner products, the standard inner product that we have used previously and a new more accurate inner product. The definitions of these inner products are based on interpolation of the fluxes of vector fields. The derivation of the new inner product is closely related to the use of the Piola transform in mixed finite elements. Computing the formulas for the new accurate inner product requires a non-trivial use of computer algebra. From the results of our numerical experiments we can conclude that using more accurate inner product produces a method with the same order of convergence as the standard inner product but the constant in error estimate is about three times less. However, the method based on the standard inner product is easier to compute with and less sensitive to grid irregularities, so we recommend its use for rough grids.

1 Introduction

The *support-operators method* (SOM) [6, 21, 22, 23] for discretizing partial differential equations takes advantage of the fact that most partial differential equations of importance in mathematical physics and engineering can be formulated in terms the invariant differential operators divergence, gradient, and curl. The SOM provides a systematic approach to spatial differencing of partial differential equations by constructing discrete analogs of these invariant operators that *exactly* satisfy discrete analogs of important differential and integral identities satisfied by the invariant continuum operators. From the discrete identities, in direct analogy with the continuum, one can then derive *exact* discrete conservation laws and *exact* analogs of other important physical principles, which in turn assures the stability and robustness of the method. The SOM method has been developed for and applied to a wide range of problems [3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 18, 19, 20, 21, 22].

In the version of the SOM studied here, the main steps are: choose a discretization of the scalar and vector fields; choose a discretization for the divergence ($\nabla\cdot$); then choose discrete inner products for discrete scalar and vector fields; and then use a discrete analog of the *Divergence Theorem* to determine the discrete gradient (∇). The Divergence Theorem says that

$$\int_\Omega \nabla \cdot \vec{v}\, f\, dV + \int_\Omega \vec{v}\, \nabla\, f\, dV = \int_{\partial\Omega} f\, \vec{v} \cdot \vec{n}\, dS\,, \tag{1.1}$$

where Ω is some smooth region, $\partial\Omega$ is the boundary of the region, $\vec{n}$ is an outward normal to the boundary, f is a smooth scalar function defined on the closure of the region, and $\vec{v}$ is a smooth vector field defined on the closure of the region (see [23]). So if f and g are scalar fields and if $\vec{v}$ and $\vec{w}$ are vector fields, then relevant continuum inner products for scalars and vectors are

$$\langle f, g\rangle = \int_\Omega f\, g\, dV\,, \quad \langle \vec{v}, \vec{w}\rangle = \int_\Omega \vec{v} \cdot \vec{w}\, dV\,, \tag{1.2}$$

and then (1.1) can be written

$$\langle \nabla \cdot \vec{v}, f\rangle + \langle \vec{v}, \nabla\, f\rangle = \int_{\partial\Omega} f\, \vec{v} \cdot \vec{n}\, dS\,. \tag{1.3}$$

Previously, natural geometric ideas have been used to discretize these inner products, while standard finite-volumes are used to discretize the divergence. The discrete analog of the gradient is derived from the discrete analog of the (1.3).

In this paper, we show how to improve the accuracy of the standard discrete inner product for *vectors*, and thereby improve the accuracy of the SOM, particularly the accuracy of the fluxes. There is no rigorous theory that can predict the accuracy of the SOM, and in particular, the accuracy of the gradient, from the accuracy of the inner products and the accuracy of the divergence, but experience shows that improvements in the accuracy of the inner products improves the accuracy of the SOM.

Our interest in this problem was motivated by the mixed finite-element method (e.g. see [1, 2]). The new inner product is intimately connected to the Piola transform used in the mixed methods to define the spaces of vector function elements, however our derivation is distinctly different from that used in the mixed method.

The many applications of the SOM has shown that when one is interested in long-time simulations, solving problems in non-isotropic heterogeneous materials, or using grids that may not be smooth, the best results are usually obtained when the numerical approximations of the fundamental operators have *exact* discrete analogs of the important properties of the continuum operators. Because the SOM discretizations exactly satisfy discrete conservation laws, the solutions of the discrete problem are stable because the discrete solutions will satisfy bounds analogous to the continuum solution. This stability is the source of the strengths of the SOM. However, the accuracy of the SOM is not guaranteed and depends in a nontrivial way on the details of the construction of the discrete analogs of the invariant operators and the discrete inner products.

In Section 2 we illustrate the SOM using a one-dimensional example. For this example we give the usual "standard" inner product and then introduce a "new" inner product based on linear interpolation of vector fields and then extend these inner products to two-dimensional tensor product grids. These examples clearly show how improving the inner product for vectors improves the accuracy of the gradient. However, the example accurately illustrates how improving the inner product makes the SOM more complicated. For smooth grids, the gradients based on the standard and new inner products are second-order accurate and, for rough grids, they are both first-order accurate when we used a *point projection* to compare the discrete to the continuum (see (2.1.2) and (2.18) below). A clue that the new inner product is better comes from using an *integral-average projection*, which shows that the new inner product produces a gradient that is then second-order accurate in arbitrary grids, while the standard inner product is still only first-order accurate. From the a finite-volume point of view, it is natural to use integral-average projections to access the accuracy of these methods.

In Section 3 we define cell inner products of vector fields on a general quadrilateral cell in the plane. We begin by describing the cell using bilinear interpolation of its corners and then use this description to compute the geometric properties of the cell. Using this, we can easily describe the standard inner product and introduce the new inner product. The crucial point in this section is to show that it is natural to define the inner product by linearly interpolating the fluxes corresponding the vector field rather than the vector field itself. This produces an interpolation for vector fields that is intimately connected to the Piola transform used in the mixed finite-element method. The resulting inner product is so complex that we used computer algebra techniques in a nontrivial way to compute its formula. The new inner product has defied our attempts at analysis, but the numerical tests indicate some advantages.

The crucial issue for us is not how much the gradient improves, but how much does the new inner product improve the accuracy of the solution of Laplace's equation, particularly the accuracy of the fluxes? The SOM has been well tested, so Section 5 is devoted to using a few numerical examples to check this question. For smooth grids both the new and standard inner products produce second-order accurate solutions and fluxes, and for rough grids the solution is still second-order accurate for both inner products, but the fluxes are only first-order accurate. However, the total errors for both the solution and the fluxes are smaller for the new inner product.

Another important question for us is what is the best way to create a vector inner product in three dimension: the standard inner product doesn't have a clear generalization while the new one does, although the inner product will be quite complicated. So it is important to be able to numerically evaluate the new inner product quickly, and there are a number of ideas used in finite-elements that could bring the cost within reason. Because the new inner product has a clear finite element interpretation, ideas from this area will certainly help. Anyway, further work in this direction needs to be done before we can routinely use the new inner product, even in two dimensions. As a final note, these inner products capture considerable geometric detail about the cell on which the vector field is to be interpolated, and consequently cannot be simple!

Our understanding of the use of the Piola transform in the mixed finite-element method and of the importance of interpolation for defining inner products was considerably influenced by conservations with colleagues and students. Thanks to T. Arbogast, J.M. Morel, N. Robidoux, T.F. Russell, M. Wheeler, and I. Yotov.

2 Motivation

In this section, we present a one dimensional example to illustrate how the SOM works and how the method depends on the inner product of the vector fields. We then describe the extension of the one-dimensional case to tensor-product grids in two dimensions,

2.1 SOM in One Dimension

In one dimension, the Divergence Theorem (1.1) becomes integration by parts

$$\int_a^b v' \, f \, dx + \int_a^b v \, f' \, dx = \left. f \, v \right|_a^b \, , \tag{2.4}$$

but when discretizing this formula we will need to keep track of which functions are the analogs of the scalars and which are analogs of vectors and which operators are the analogs of the divergence and gradient. So in this section we will use functions v and w as analogs of the vector fields and functions f and g as analogs of the scalar fields, while $\mathbf{D}$ is the discrete analog of the divergence and $\mathbf{G}$ is the discrete analog of the gradient. In this paper, we will not analyze the accuracy of the boundary conditions, so we will assume that all functions (and their derivatives, if necessary) are zero at the boundary. Under this assumption, we can write (2.4) as

$$\langle v', f \rangle + \langle v, f' \rangle = 0 \tag{2.5}$$

where the inner products are

$$\langle f, g \rangle = \int_a^b f \, g \, dx \, , \quad \langle v, w \rangle = \int_a^b v \, w \, dx \, . \tag{2.6}$$

In fact, in the continuum in one dimension, the inner products are the same, but they will be discretized differently.

The discretizations is generated on a one-dimensional grid, $\{x_i; i = 0, \ldots, N\}$, where $x_0 = a$, $x_N = b$, $N > 0$, $x_{i+\frac{1}{2}} = (x_{i+1} + x_i)/2$, and $h_{i+\frac{1}{2}} = x_{i+1} - x_i$. We choose a cell-centered discretization for the space $\mathcal{S}$ of scalar fields: $f_{i+\frac{1}{2}}$, $i = 0, \cdots N-1$, and a nodal discretization for the space $\mathcal{V}$ of vector fields: v_i, $i = 0, \cdots N$.

There are natural discretizations of the inner products (2.6). We write our discrete inner products as a cell inner product and then sum the cell inner-products over the grid. For the discrete scalar functions the mid-point rule is natural:

$$\langle f, g\rangle_{i+\frac{1}{2}} = f_{i+\frac{1}{2}}\, g_{i+\frac{1}{2}}\, h_{i+\frac{1}{2}}\,, \tag{2.7}$$

with

$$\langle f, g\rangle = \langle f, g\rangle_{\mathcal{S}} = \sum_{i=0}^{N-1} \langle f, g\rangle_{i+\frac{1}{2}}\,. \tag{2.8}$$

For the discrete vector functions the trapezoid rule is natural:

$$\langle v, w\rangle_{i+\frac{1}{2}} = \frac{v_{i+1}\, w_{i+1} + v_i\, w_i}{2}\, h_{i+\frac{1}{2}}\,, \tag{2.9}$$

with

$$\langle v, w\rangle = \langle v, w\rangle_{\mathcal{V}} = \sum_{i=0}^{N-1} \langle v, w\rangle_{i+\frac{1}{2}}\,. \tag{2.10}$$

The natural finite-volume discretization for divergence is

$$(\mathbf{D}\; v)_{i+\frac{1}{2}} = \frac{v_{i+1} - v_i}{h_{i+\frac{1}{2}}}\,. \tag{2.11}$$

In the SOM, the discrete gradient is constructed as negative adjoint to $\mathbf{D}$ in sense that for the just defined discrete inner products $\mathbf{G} = -\mathbf{D}^*$, that is,

$$\langle \mathbf{G}\; f, v\rangle_{\mathcal{V}} = -\langle f, \mathbf{D}\; v\rangle_{\mathcal{S}}\,. \tag{2.12}$$

The definitions of the two inner products, the definition of the divergence, the summation by parts formula (2.12), and the the fact that f and v are arbitrary gives the expression

$$\frac{h_{i+\frac{1}{2}} + h_{i-\frac{1}{2}}}{2} (\mathbf{G}\, f)_i = f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}\,, \tag{2.13}$$

which means the discrete gradient is

$$(\mathbf{G}\, f)_i = \frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{(h_{i+\frac{1}{2}} + h_{i-\frac{1}{2}})/2}\,. \tag{2.14}$$

This *standard* gradient is usual discretization in non-uniform grids.

2.1.1 A New Inner Product

Let us introduce a *new* more accurate inner product in space of discrete vector functions. To do this, we assume that vector fields varies linearly between nodes, that is,

$$v(x) = v_i + \frac{v_{i+1} - v_i}{x_{i+1} - x_i}\,(x - x_i)\,, \quad x_{i+1} \geq x \geq x_i\,, \tag{2.15}$$

on the $i + 1/2$ cell. If v and w are both defined in a cell by linear interpolation then we define the cell inner product as

$$\begin{aligned} \langle v,\, w\rangle_{i+\frac{1}{2}} &= \int_{x_i}^{x_{i+1}} v(x)\, w(x)\, dx \\ &= h_{i+\frac{1}{2}} \left(\frac{1}{3}\, v_i\, w_i + \frac{1}{6}\,(v_i\, w_{i+1} + v_{i+1}\, w_i) + \frac{1}{3}\, v_{i+1}\, w_{i+1} \right) . \end{aligned} \tag{2.16}$$

As above, the global inner product in space of vector field is the sum of cell inner products over all cells (2.10).

Again, as above, the new definition of the inner product for vectors yields a new gradient that, analogous to (2.13), satisfies the following system of equations

$$\frac{h_{i-\frac{1}{2}}}{6}\,(\mathbf{G}\, f)_{i-1} + \frac{h_{i+\frac{1}{2}} + h_{i-\frac{1}{2}}}{3}\,(\mathbf{G}\, f)_i + \frac{h_{i+\frac{1}{2}}}{6}\,(\mathbf{G}\, f)_{i+1} = f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}\,,\,, \quad 1 \leq i \leq N-1\,. \tag{2.17}$$

This new discrete gradient has the same form as the compact finite-difference approximation to the first derivative and the finite element approximations that involve the mass matrix.

2.1.2 Accuracy Analysis

The accuracy analysis is based on projecting smooth functions onto the grid and then using Taylor series to compute the truncation error of the differential operators. Because we are using a *staggered* grid, the natural projection of a smooth vector function $v(x)$ onto the grid is given by the point projection

$$v_i = v(x_i)\,,$$

while for scalar functions there are two natural projections onto the grid, the point projection and the integral-average projection:

$$f_{i+\frac{1}{2}} = f(x_{i+\frac{1}{2}}) \quad \text{or} \quad f_{i+\frac{1}{2}} = \frac{1}{h_{i+\frac{1}{2}}} \int_{x_i}^{x_{i+1}} f(\xi)\, d\xi\,. \tag{2.18}$$

The choice of projection will affect the accuracy analysis.

The analysis of the inner-products shows that for both projections the cell inner products are third-order accurate and the global inner products are second-order accurate. More importantly, by construction, the new inner product of vector fields is exact on linear functions, while the standard is not. It is in this sense the new inner product is more accurate than the standard inner product.

The accuracy of the divergence is evaluated using Taylor series centered at the cell centers,

$$v(x) = \sum_{0}^{\infty} v^{(n)}(x_{i+\frac{1}{2}}) \frac{(x - x_{i+\frac{1}{2}})^n}{n!}\,, \tag{2.19}$$

because the divergence is evaluated at the cell centers. Using the point projection for v' and v gives the truncation error as

$$v'(x_{i+\frac{1}{2}}) - \frac{v(x_{i+1}) - v(x_i)}{h_{i+\frac{1}{2}}} \approx -\frac{1}{24} h_{i+1/2}^2 v^{(3)}(x_{i+\frac{1}{2}})\,, \tag{2.20}$$

which is the standard second-order estimate for the centered difference. If we use the integral-average projection for v' and the point projection for v, then the fundamental theorem of calculus gives the truncation error as

$$\frac{1}{h_{i+\frac{1}{2}}} \int_{x_i}^{x_{i+1}} v'(\xi)\, d\xi - \frac{v(x_{i+1}) - v(x_i)}{h_{i+\frac{1}{2}}} \equiv 0\,, \tag{2.21}$$

that is, the divergence is exact!

The gradient is evaluated at the nodes, so to analyze its truncation error, f is expanded at x_i, and then we use the point projection for f', but either the point or integral-average projection for f. The point projection for f gives

$$\begin{aligned} f'(x_i) - \frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{(h_{i+\frac{1}{2}} + h_{i-\frac{1}{2}})/2} \quad &\approx \quad \frac{1}{4}\left(h_{i+\frac{1}{2}} - h_{i-\frac{1}{2}}\right) f^{(2)}(x_i) \\ &- \quad \frac{1}{24}\left(h_{i+\frac{1}{2}}^2 - h_{i+\frac{1}{2}} h_{i-\frac{1}{2}} + h_{i-\frac{1}{2}}^2\right) f^{(3)}(x_i)\,. \end{aligned} \tag{2.22}$$

The integral-average projection for f gives

$$\begin{aligned} f'(x_i) - \frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{(h_{i+\frac{1}{2}} + h_{i-\frac{1}{2}})/2} \quad &\approx \quad \frac{1}{3}\left(h_{i+\frac{1}{2}} - h_{i-\frac{1}{2}}\right) f^{(2)}(x_i) \\ &- \quad \frac{1}{12}\left(h_{i+\frac{1}{2}}^2 - h_{i+\frac{1}{2}} h_{i-\frac{1}{2}} + h_{i-\frac{1}{2}}^2\right) f^{(3)}(x_i)\,. \end{aligned} \tag{2.23}$$

So in both cases, the gradient is first-order accurate in general grids and second-order accurate in uniform grids.

As we do not have an explicit expression for the new gradient, we will analyze the residual of the equation the defines the gradient. For the purposes of comparison, we rewrite the truncation error for the standard gradient using the residual of Equation (2.14) which is the analog of formula (2.17) for the new gradient. So we set

$$R_i^{\mathrm{std}} = \frac{(h_{i+\frac{1}{2}} + h_{i-\frac{1}{2}})}{2} f'(x_i) - \left(f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}\right) \tag{2.24}$$

and then, as above, expand at x_i and then use the point projection for f', but either the point or integral-average projection for f. The point projection for f gives

$$R_i^{\rm std} \approx \frac{1}{8}\left(h_{i+\frac{1}{2}}^2 - h_{i-\frac{1}{2}}^2\right) f^{(2)}(x_i) - \frac{1}{48}\left(h_{i+\frac{1}{2}}^3 + h_{i-\frac{1}{2}}^3\right) f^{(3)}(x_i)\,, \tag{2.25}$$

while the integral average projection gives

$$R_i^{\rm std} \approx \frac{1}{6}\left(h_{i+\frac{1}{2}}^2 - h_{i-\frac{1}{2}}^2\right) f^{(2)}(x_i) - \frac{1}{24}\left(h_{i+\frac{1}{2}}^3 + h_{i-\frac{1}{2}}^3\right) f^{(3)}(x_i)\,. \tag{2.26}$$

For the new gradient (2.17) given by the new inner product, the residual is

$$R_i^{\rm new} = \frac{h_{i-\frac{1}{2}}}{6}(\mathbf{G}\,f)_{i-1} + \frac{h_{i+\frac{1}{2}} + h_{i-\frac{1}{2}}}{3}(\mathbf{G}\,f)_i + \frac{h_{i+\frac{1}{2}}}{6}(\mathbf{G}\,f)_{i+1} - \left(f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}\right)\,. \tag{2.27}$$

The point projection for f gives

$$R_i^{\rm new} \approx \frac{1}{24}\left(h_{i+\frac{1}{2}}^2 - h_{i-\frac{1}{2}}^2\right) f^{(2)}(x_i) - \frac{1}{48}\left(h_{i+\frac{1}{2}}^3 + h_{i-\frac{1}{2}}^3\right) f^{(3)}(x_i)\,, \tag{2.28}$$

while the integral-average projection for f gives

$$R_i^{\rm new} \approx \frac{1}{24}\left(h_{i+\frac{1}{2}}^3 + h_{i-\frac{1}{2}}^3\right) f^{(3)}(x_i) - \frac{1}{120}\left(h_{i+\frac{1}{2}}^3 - h_{i-\frac{1}{2}}^3\right) f^{(4)}(x_i)\,. \tag{2.29}$$

So, in a uniform grid, all of the above residuals are third order, implying the gradient is second order. In non-uniform grids, the integral average projection error for the equation defining the *new gradient* is third order, while all others are second order. So in arbitrary grids, the new gradient is second-order in the integral average projection, while all other truncation errors are first order. This is the main motivation for extending these ideas to higher dimensions.

In summary, in all cases the inner products for scalars and vectors are third-order accurate on each cell and globally second-order accurate. The new inner product for vectors is exact on linear functions, but the standard is not. For the point projection, the divergence is second-order accurate on general grids, and for the integral-average projection, the divergence is exact. All truncation errors for the gradient are second-order accurate in uniform grids and first order accurate in general grids with the important exception that using the integral average projections gives a second-order accurate gradient in general grids. These results have been confirmed by numerical experiment.

2.2 SOM in Tensor Product Grids

A tensor product grid is given by the points (x_i, y_j), $0 \le i \le N$, $0 \le j \le M$ where the x_i and y_j give grids in one dimension. The cells in the grid are labeled by their mid-points $x_{i+\frac{1}{2},j+\frac{1}{2}}$, and if $\Delta x_{i+\frac{1}{2}} = x_{i+1} - x_i$ and $\Delta y_{j+\frac{1}{2}} = y_{j+1} - y_j$ then the area of such a cell is given by

$$A_{i+\frac{1}{2},j+\frac{1}{2}} = \Delta x_{i+\frac{1}{2}} \Delta y_{j+\frac{1}{2}}\,. \tag{2.30}$$

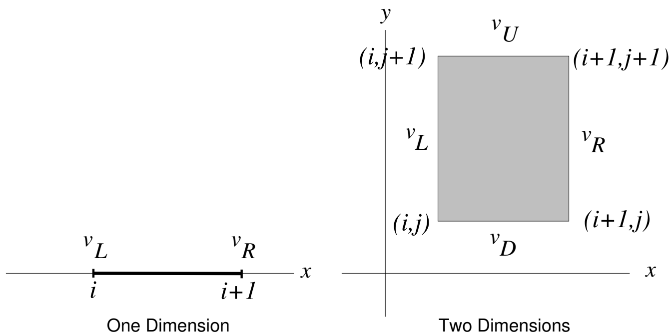


Figure 2.1: The Data on a Cell

Now scalar functions are discretized as $f_{i+\frac{1}{2},j+\frac{1}{2}}$ and vector fields $v = (v^{(1)}, v^{(2)})$ are given by their normal Cartesian component at the centers of the sides of cell:

$$v^{(1)}_{i,j+\frac{1}{2}}\,, \quad 0 \le i \le N\,, \, 0 \le j \le M-1\,, \quad v^{(2)}_{i+\frac{1}{2},j}\,, \quad 0 \le i \le N-1\,, \, 0 \le j \le M\,.$$

The notation can be quite cumbersome, and because we work mostly with a single cell as shown in Figure 2.1, we introduce the cell-based notation

$$v_L = v^{(1)}_{i,j+\frac{1}{2}}\,, \quad v_R = v^{(1)}_{i+1,j+\frac{1}{2}}\,, \quad v_D = v^{(2)}_{i+\frac{1}{2},j}\,, \quad v_U = v^{(2)}_{i+\frac{1}{2},j+1}\,,$$

and then set $\bar{v} = (v_L, v_R, v_D, v_U)$. Of course, the subscripts L, R, D, U stand for left, right, up and down, dx and dy give the lengths of the sides of the cell, and A denotes the area of the cell.

The results for tensor-product grids are essentially the same as for one-dimensional grids, so here we just look at the new gradient to see that its accuracy is preserved. If $\bar{v}$ and $\bar{w}$ are discrete vector fields on a cell, then the generalization of the standard cell inner product (2.9) to a tensor product grid is

$$\langle \bar{v}, \bar{w} \rangle_{\text{std}} = \left(\frac{v_L\, w_L + v_R\, w_R}{2} + \frac{v_D\, w_D + v_U\, w_U}{2} \right) A\,, \tag{2.31}$$

while new inner product (2.16) becomes

$$\begin{aligned} \langle \bar{v}, \bar{w} \rangle_{\text{new}} \;\; = \;\; & \left(\left(\frac{1}{3}\, v_L\, w_L + \frac{1}{6}\, (v_L\, w_R + v_R\, w_L) + \frac{1}{3}\, v_R\, w_R \right) \right. \\ & \left. + \left(\frac{1}{3}\, v_D\, w_D + \frac{1}{6}\, (v_D\, w_U + v_U\, w_D) + \frac{1}{3}\, v_U\, w_U \right) \right) A\,. \end{aligned} \tag{2.32}$$

In any dimension, the global inner products are obtained by summing the cell inner products over all of the cells.

The sum of the fluxes out of the cell divided by the area of the cell gives the natural finite volume divergence on the cell:

$$(\mathbf{D}\ V) = \frac{(v_R \Delta y - v_L \Delta y + v_U \Delta x - v_D \Delta x)}{A}\,, \tag{2.33}$$

and because of the formula (2.30) for the area of cell, this can be rewritten as

$$(\mathbf{D}\ V) = \frac{v_R - v_L}{\Delta x} + \frac{v_U - v_D}{\Delta y}\,. \tag{2.34}$$

Again, as in one dimension, combining the definitions of the inner products and the divergence and a double summation by parts tells us that the new gradient of a scalar is a vector,

$$\mathbf{G}\ f = (\mathbf{G}_x f, \mathbf{G}_y f)\,,$$

where

$$\begin{aligned}
\frac{\Delta x_{i-\frac{1}{2}}}{6}(\mathbf{G}_x f)_{i-1,j+\frac{1}{2}} + \frac{\Delta x_{i+\frac{1}{2}} + \Delta x_{i-\frac{1}{2}}}{3}(\mathbf{G}_x f)_{i,j+\frac{1}{2}} + \frac{\Delta x_{i+\frac{1}{2}}}{6}(\mathbf{G}_x f)_{i+1,j+\frac{1}{2}} &= \\
f_{i+\frac{1}{2},j+\frac{1}{2}} - f_{i-\frac{1}{2},j+\frac{1}{2}}\,, & \\
\frac{\Delta y_{j-\frac{1}{2}}}{6}(\mathbf{G}_y f)_{i+\frac{1}{2},j-1} + \frac{\Delta y_{j+\frac{1}{2}} + \Delta y_{j-\frac{1}{2}}}{3}(\mathbf{G}_y f)_{i+\frac{1}{2},j} + \frac{\Delta y_{j+\frac{1}{2}}}{6}(\mathbf{G}_y f)_{i+\frac{1}{2},j+1} &= \\
f_{i+\frac{1}{2},j+\frac{1}{2}} - f_{i+\frac{1}{2},j-\frac{1}{2}}\,. &
\end{aligned} \tag{2.35}$$

We have used index notation here because the gradient is not cell based.

2.2.1 Accuracy of the Tensor Grid Gradient

Here we are only interested in seeing if we obtain a gradient with improved accuracy using the integral average projection. So, as is natural, the integral average projection of a smooth scalar field f is

$$f_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{A_{i+\frac{1}{2},j+\frac{1}{2}}} \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} f(x,y)\,dx\,dy \tag{2.36}$$

As we will see from the accuracy results, a good projection to use for a smooth vector field $\vec{v} = (v_x(x,y), v_y(x,y))$ is the integral average of the normal component of the vector field:

$$v_{i,j+\frac{1}{2}} = \frac{1}{\Delta y_{j+\frac{1}{2}}} \int_{y_j}^{y_{j+1}} v_x(x_i, y)\,dy\,, \tag{2.37}$$

$$v_{i+\frac{1}{2},j} = \frac{1}{\Delta x_{i+\frac{1}{2}}} \int_{x_i}^{x_{i+1}} v_y(x, y_j)\,dx\,. \tag{2.38}$$

To analyze the error in the x-component of the gradient, as before, we set the residual to

$$\begin{aligned}
R_i^x = \frac{\Delta x_{i-\frac{1}{2}}}{6}(\mathbf{G}_x f)_{i-1,j+\frac{1}{2}} + \frac{\Delta x_{i+\frac{1}{2}} + \Delta x_{i-\frac{1}{2}}}{3}(\mathbf{G}_x f)_{i,j+\frac{1}{2}} + \frac{\Delta x_{i+\frac{1}{2}}}{6}(\mathbf{G}_x f)_{i+1,j+\frac{1}{2}} \\
-(f_{i+\frac{1}{2},j+\frac{1}{2}} - f_{i-\frac{1}{2},j+\frac{1}{2}})\,.
\end{aligned} \tag{2.39}$$

Now in (2.37) we replace v_x by $\partial f/\partial x$, and then in (2.39) we replace $(\mathbf{G}_x f)_{i,j+\frac{1}{2}}$ by $v_{i,j+\frac{1}{2}}$ (also with i replaced by $i-1$ and $i+1$) and $f_{i+\frac{1}{2},j+\frac{1}{2}}$ (also with i replaced by $i-1$) by

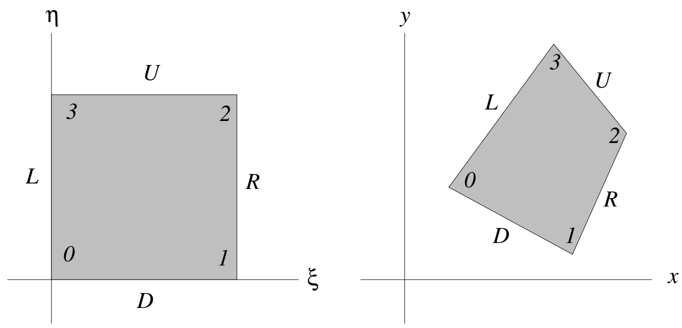


Figure 3.1: A Two-Dimensional Cell

the integral average projection of f (2.36). All terms in the resulting formula contain an integral with respect to y that can be factored out. The integrand is the same as in the one-dimensional residual with f' replaced by $\partial f/\partial x$ and all functions now parameterized by y. So the same analysis as in one dimension gives

$$\begin{aligned} R_i^x \;\approx\; & \frac{1}{24}\left(h_{i+\frac{1}{2}}^3 + h_{i-\frac{1}{2}}^3\right) \frac{1}{\Delta y_{i+\frac{1}{2}}} \int_{y_i}^{y_{i+1}} f^{(3)}(x_i, y)\, dy \\ - \; & \frac{1}{120}\left(h_{i+\frac{1}{2}}^3 - h_{i-\frac{1}{2}}^3\right) \frac{1}{\Delta y_{i+\frac{1}{2}}} \int_{y_i}^{y_{i+1}} f^{(4)}(x_i, y)\, dy\,, \end{aligned}$$

with a similar result for $\mathbf{G}_y$. Thus the gradient in tensor product grids has the same accuracy as the one-dimensional analog of the gradient provided that we use an appropriate integral average of normal components projection for vector fields. Numerical tests confirm that the order of accuracy using both point and integral average projections is the same as in one dimension.

3 The Inner Products in General 2-D Cells

In this section we introduce two inner products for discrete vector fields defined on grids of quadrilateral cells: the standard inner product that is described in many of the referenced papers on support-operators methods (e.g. [22]), and the new inner product. We describe both inner products from unified viewpoint based on the idea that cells are defined by mappings from a unit square (see Figure 3.1). In the discrete case, vector fields are described by their projections onto the unit normals to cell faces [22]. An inner product for discrete vector fields must be a symmetric and positive-definite bilinear form with respect to the components of a vector. Our inner products are cell based: they can be defined on a single grid cell (see Figure 3.1) and then summed or all cell to make a global inner product. The process of constructing the global inner product from the cell inner products is similar to assembling a global mass matrix from the local mass matrices in the finite element method.

3.1 Bilinear Interpolation of a Cell

In one dimension, a cell is described by giving the end points of an interval, and then the full cell can be given using linear interpolation. In two dimensions, a cell is describe by giving its four corners and then the full cell is given by bilinear interpolation (Figure 3.1). So if the four points are (x_0, y_0), (x_1, y_1), (x_2, y_2), and (x_3, y_3), then a unit square can be mapped to the cell using a bilinear map:

$$\begin{aligned} x(\xi,\eta) &= x_0\,(1-\xi)\,(1-\eta) + x_1\,\xi\,(1-\eta) + x_2\,\xi\,\eta + x_3\,(1-\xi)\,\eta\,, \\ y(\xi,\eta) &= y_0\,(1-\xi)\,(1-\eta) + y_1\,\xi\,(1-\eta) + y_2\,\xi\,\eta + y_3\,(1-\xi)\,\eta\,, \end{aligned} \tag{3.40}$$

where the corners are mapped as follows:

$$(0,0) \to (x_0, y_0)\,, \quad (1,0) \to (x_1, y_1)\,, \quad (0,1) \to (x_2, y_2)\,, \quad (1,1) \to (x_3, y_3)\,. \tag{3.41}$$

To describe the inner products, we need detailed formulas for the Jacobian, and the tangent and normal vectors to the lines $\xi = const$ and $\eta = const$. The Jacobian of a bilinear map is linear in ξ and η and has the form

$$J = J(\xi,\eta) = a + b\,\xi + c\,\eta\,, \tag{3.42}$$

where

$$a = (x_1 - x_0)(y_3 - y_0) - (x_3 - x_0)(y_1 - y_0)\,, \tag{3.43}$$

$$\begin{aligned} b &= (x_1 - x_0)(y_2 - y_0) + (x_3 - x_0)(y_1 - y_0) \\ &\quad -(x_2 - x_0)(y_1 - y_0) - (x_1 - x_0)(y_3 - y_0)\,, \end{aligned} \tag{3.44}$$

$$\begin{aligned} c &= (x_2 - x_0)(y_3 - y_0) + (x_3 - x_0)(y_1 - y_0) \\ &\quad -(x_3 - x_0)(y_2 - y_0) - (x_1 - x_0)(y_3 - y_0)\,. \end{aligned} \tag{3.45}$$

If we fix η and vary ξ, then $(x(\xi,\eta), y(\xi,\eta)$ is a straight line, conversely if we fix ξ and vary η, the we also get a straight line. These straight lines form a coordinate system in the the two-dimensional cell that is not necessarily orthogonal. As illustrated in Figure 3.2, at a point where a line from each family intersects, we define the tangent and normal vectors: $\vec{T}_\xi = (x_\xi, y_\xi)$; $\vec{T}_\eta = (x_\eta, y_\eta)$; $\vec{N}_\xi = (y_\eta, -x_\eta)$; $\vec{N}_\eta = (-y_\xi, x_\xi)$. The notation is arranged so that $\vec{T}_\xi$ and $\vec{N}_\xi$ point in the direction of increasing ξ, while $\vec{T}_\eta$ and $\vec{N}_\eta$ point in the direction of increasing η. Consequently, $\vec{T}_\xi = \vec{N}_\xi$ and $\vec{T}_\eta = \vec{N}_\eta$ if the coordinate lines are orthogonal, that is, the cell is a rectangle.

For a bilinear map, the tangent and normal vectors are linear functions of a single variable:

$$\vec{T}_\xi(\eta) = \vec{T}_D\,(1-\eta) + \vec{T}_U\,\eta\,, \tag{3.46}$$

$$\vec{T}_\eta(\xi) = \vec{T}_L\,(1-\xi) + \vec{T}_R\,\xi\,, \tag{3.47}$$

$$\vec{N}_\xi(\xi) = \vec{N}_L\,(1-\xi) + \vec{N}_R\,\xi\,, \tag{3.48}$$

$$\vec{N}_\eta(\eta) = \vec{N}_D\,(1-\eta) + \vec{N}_U\,\eta\,, \tag{3.49}$$

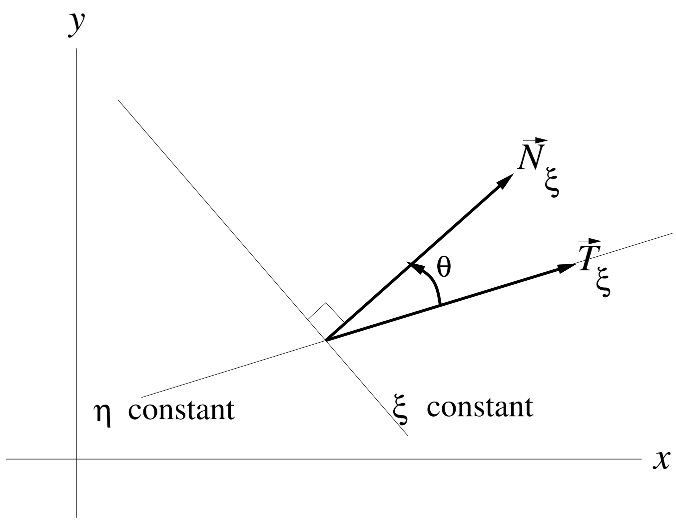


Figure 3.2: Tangent and Normal Vectors

where

$$\begin{aligned}
\vec{T}_D &= (x_1 - x_0, y_1 - y_0)\ , \qquad & \vec{N}_D &= (y_0 - y_1, x_1 - x_0)\ , \\
\vec{T}_U &= (x_2 - x_3, y_2 - y_3)\ , \qquad & \vec{N}_U &= (y_3 - y_2, x_2 - x_3)\ , \\
\vec{T}_L &= (x_3 - x_0, y_3 - y_0)\ , \qquad & \vec{N}_L &= (y_3 - y_0, x_0 - x_3)\ , \\
\vec{T}_R &= (x_2 - x_1, y_2 - y_1)\ , \qquad & \vec{N}_R &= (y_2 - y_1, x_1 - x_2)\ ,
\end{aligned}$$

normal and tangential vectors to the sides of the cell.

3.2 Cell Geometry

Before we can describe the inner products, we need to know a bit more about the geometry of the cell. First, it is easy to check that:

$$\begin{aligned}
\vec{T}_\xi \cdot \vec{N}_\xi &= J\,, \qquad & \vec{T}_\xi \cdot \vec{N}_\eta &= 0\,, \\
\vec{T}_\eta \cdot \vec{N}_\xi &= ,0 \qquad & \vec{T}_\eta \cdot \vec{N}_\eta &= J\,,
\end{aligned} \tag{3.50}$$

and consequently the tangent and normal vectors are biothogonal. Moreover

$$|\vec{N}_\xi| = |\vec{T}_\eta|\,, \quad |\vec{N}_\eta| = |\vec{T}_\xi|\,. \tag{3.51}$$

If we introduce the angle θ between the tangent and normal vectors (see Figure 3.2), and use the biorthogonality of the tangent and normal vectors (3.50), then

$$\cos(\theta) = \frac{\vec{T}_\xi \cdot \vec{N}_\xi}{|\vec{T}_\xi|\,|\vec{N}_\xi|} = \frac{J}{|\vec{T}_\xi|\,|\vec{N}_\xi|} = \frac{J}{|\vec{N}_\eta|\,|\vec{T}_\eta|} = \frac{\vec{T}_\eta \cdot \vec{N}_\eta}{|\vec{T}_\eta|\,|\vec{N}_\eta|}\,. \tag{3.52}$$

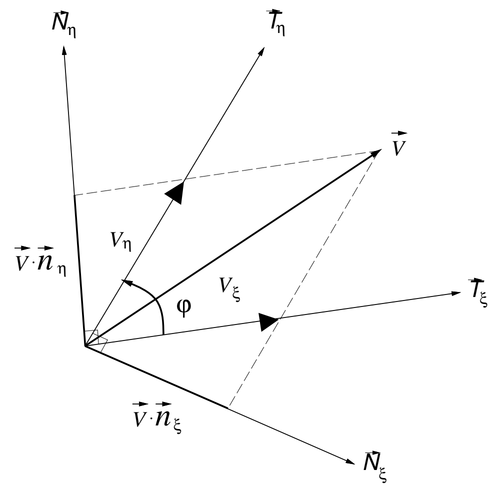


Figure 3.3: Local Basis of Tangents and Normals

We will need the the unit tangent and normal vectors:

$$\vec{t}_\xi = \frac{\vec{T}_\xi}{|\vec{T}_\xi|}\,; \qquad \vec{t}_\eta = \frac{\vec{T}_\eta}{|\vec{T}_\eta|}\,; \tag{3.53}$$

$$\vec{n}_\xi = \frac{\vec{N}_\xi}{|\vec{N}_\xi|}\,; \qquad \vec{n}_\eta = \frac{\vec{N}_\eta}{|\vec{N}_\eta|}\,, \tag{3.54}$$

which are bi-orthonormal:

$$\begin{array}{ll} \vec{t}_\xi \cdot \vec{n}_\xi = \cos(\theta)\,; & \vec{t}_\xi \cdot \vec{n}_\eta = 0\,; \\ \vec{t}_\eta \cdot \vec{n}_\xi = 0\,; & \vec{t}_\eta \cdot \vec{n}_\eta = \cos(\theta)\,. \\ |\vec{t}_\xi| = 1 & |\vec{t}_\eta| = 1 \end{array} \tag{3.55}$$

As shown in Figure 3.3, we introduce the angle ϕ between the two tangent vectors,

$$\vec{t}_\xi \cdot \vec{t}_\eta = \cos(\phi)\,, \tag{3.56}$$

and note that if θ is the angle between the tangents and normals as given in (3.52) and shown in Figure 3.2, then $\phi + \theta = \pi/2$, so $\cos(\theta) = \sin(\phi)$.

We also need the lengths of the sides of the cells:

$$S_D = |\vec{N}_\eta(0)| = |\vec{T}_\xi(0)| \;=\; \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}\,, \tag{3.57}$$

$$S_U = |\vec{N}_\eta(1)| = |\vec{T}_\xi(1)| \;=\; \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2}\,, \tag{3.58}$$

$$S_L = |\vec{N}_\xi(0)| = |\vec{T}_\eta(0)| \;=\; \sqrt{(x_3 - x_0)^2 + (y_3 - y_0)^2}\,, \tag{3.59}$$

$$S_R = |\vec{N}_\xi(1)| = |\vec{T}_\eta(1)| \;=\; \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}\,. \tag{3.60}$$

3.3 Vector Fields in Local Coordinates

As shown in Figure 3.3, if $\vec{V}$ is an arbitrary vector field defined on the cell, then it is convenient to write the vector field as a liner combination of the tangent vectors:

$$\vec{V} = V_\xi \, \vec{t}_\xi + V_\eta \, \vec{t}_\eta \,, \tag{3.61}$$

which implies that the normal components of $\vec{V}$ are given by

$$\vec{V} \cdot \vec{n}_\xi = V_\xi \left(\vec{t}_\xi \cdot \vec{n}_\xi \right) \,, \quad \vec{V} \cdot \vec{n}_\eta = V_\eta \left(\vec{t}_\eta \cdot \vec{n}_\eta \right) \,, \tag{3.62}$$

and that the tangential coefficients are

$$\begin{aligned} V_\xi &= \frac{\vec{V} \cdot \vec{n}_\xi}{\sin(\phi)} = \frac{\left(\vec{V} \cdot \vec{n}_\xi\right) |\vec{T}_\xi| \, |\vec{N}_\xi|}{J} \,, \\ V_\eta &= \frac{\vec{V} \cdot \vec{n}_\eta}{\sin(\phi)} = \frac{\left(\vec{V} \cdot \vec{n}_\eta\right) |\vec{T}_\eta| \, |\vec{N}_\eta|}{J} \,. \end{aligned} \tag{3.63}$$

The desired form of the vector field is then

$$\begin{aligned} \vec{V} &= \frac{\vec{V} \cdot \vec{n}_\xi}{\sin(\phi)} \, \vec{t}_\xi + \frac{\vec{V} \cdot \vec{n}_\eta}{\sin(\phi)} \, \vec{t}_\eta \\ &= \frac{\left(\vec{V} \cdot \vec{n}_\xi\right) \, |\vec{N}_\xi| \, \vec{T}_\xi + \left(\vec{V} \cdot \vec{n}_\eta\right) \, |\vec{N}_\eta| \, \vec{T}_\eta}{J} \\ &= \frac{\left(\vec{V} \cdot \vec{N}_\xi\right) \, \vec{T}_\xi + \left(\vec{V} \cdot \vec{N}_\eta\right) \, \vec{T}_\eta}{J} \,. \end{aligned} \tag{3.64}$$

It is important to note that all the expression in the previous formula are know geometric quantities except for $\vec{V} \cdot \vec{N}_\xi$ and $\vec{V} \cdot \vec{N}_\eta$, which are known as the fluxes. These fluxes will play a critical role in the interpolation, because if we know them everywhere in the cell, then we know the vector field everywhere.

3.4 Degrees of Freedom and Inner Product

The degrees of freedom we have to define the inner products are the normal components of a vector on the cell faces. More precisely, we assume we are given, on each face of the cell, the normal components of two vector fields $\vec{V}$ and $\vec{W}$:

$$\begin{aligned} V_D &= \vec{V} \cdot \vec{n}_\eta(0) \,, & W_D &= \vec{W} \cdot \vec{n}_\eta(0) \,, \\ V_U &= \vec{V} \cdot \vec{n}_\eta(1) \,, & W_U &= \vec{W} \cdot \vec{n}_\eta(1) \,, \\ V_L &= \vec{V} \cdot \vec{n}_\xi(0) \,, & W_L &= \vec{W} \cdot \vec{n}_\xi(0) \,, \\ V_R &= \vec{V} \cdot \vec{n}_\xi(1) \,, & W_R &= \vec{W} \cdot \vec{n}_\xi(1) \,. \end{aligned} \tag{3.65}$$

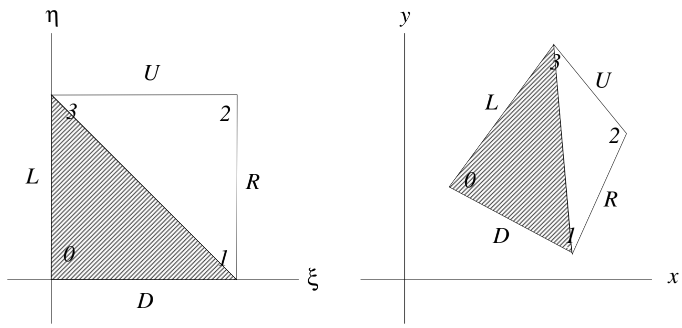


Figure 3.4: Triangles

Our plan is to interpolate the given normal components of the two vector fields $\vec{V}$ and $\vec{W}$ to the interior of the cell and then compute the discrete cell inner product by computing continuum cell inner product for the interpolated fields:

$$\langle \vec{V}, \vec{W} \rangle = \int_{\text{cell}} \vec{V}(x,y)\, \vec{W}(x,y)\, dx\, dy = \int_0^1 \int_0^1 \vec{V}(\xi,\eta)\, \vec{W}(\xi,\eta)\, J(\xi,\eta)\, d\xi\, d\eta\,.$$

Notice that the interpolation formulas for the vector field must be linear in the degrees of freedom - V_L, V_R, V_D, V_U - so that the integral will be bilinear. In fact, we will be able to compute these integrals exactly, and the new inner product is given by the exact integral, while the standard inner product is given by a simple quadrature rule applied to the integral.

3.5 The Standard Inner Product

If we are given the data of the normal components (3.65) of two vector fields on the faces of a cell:

$$\bar{V} = (V_D, V_U, V_L, V_R)\,, \quad \bar{W} = (W_D, W_U, W_L, W_R)\,,$$

then we need to construct a cell inner product $\langle \bar{V}, \bar{W} \rangle$ that is given by a symmetric positive-definite quadratic form in the given data. The standard inner product is constructed by interpolating the data to a constant vector field in the four triangles given by any three of the corner points in the cell (see Figure 3.4), and then integrating the interpolated vector field over each triangle, and then averaging these results.

Let us look at the triangle given by deleting the point labeled by 2 from the cell. This triangle is defined by the tangent vectors $\vec{T}_\xi(0)$ and $\vec{T}_\eta(0)$ given in (3.46) and (3.47). We will assume that the vector field $\vec{V}$ is constant in this triangle and defined by last expression in (3.64) at $\xi = \eta = 0$:

$$\vec{V} = \frac{\left(\vec{V} \cdot \vec{N}_\xi(0)\right) \vec{T}_\xi(0) + \left(\vec{V} \cdot \vec{N}_\eta(0)\right) \vec{T}_\eta(0)}{J(0,0)}\,. \tag{3.66}$$

If two vector fields $\vec{V}$ and $\vec{W}$ have this representation, then the definition of the unit tangents (3.54), the biorthogonality conditions (3.55), the definition of ϕ in (3.56), and the degrees of freedom given in (3.65) give following expression for the inner product of two vectors at $\xi = \eta = 0$:

$$\vec{V} \cdot \vec{W} = \frac{V_L\, W_L + V_D\, W_D + (V_L\, W_D + V_D\, W_L)\cos(\phi_0)}{\sin^2(\phi_0)} \tag{3.67}$$

Integration of this constant expression over triangle will gives $A_{LD}\, \vec{V} \cdot \vec{W}$ where A_{LD} is area of triangle having sides L and D (see Figure (3.4)). If we proceed to the triangle obtained by deleting point 3 from the cell, then we replace D by U and then the angle that appears in the formula is $\pi - \phi_1$. But $\cos(\pi - \phi_1) = -\cos(\phi_1)$ and $\sin(\pi - \phi_1) = \sin(\phi_1)$. So if we run over the four corners and take an average we get

$$\begin{aligned} \langle \bar{V}, \bar{W} \rangle \; = \; & \left(A_{LD} \frac{V_L\, W_L + V_D\, W_D + (V_L\, W_D + V_D\, W_L)\cos(\phi_0)}{\sin^2(\phi_0)} \right. \\ + \; & A_{UL} \frac{V_U\, W_U + V_L\, W_L - (V_U\, W_L + V_L\, W_U)\cos(\phi_1)}{\sin^2(\phi_1)} \\ + \; & A_{RU} \frac{V_R\, W_R + V_U\, W_U + (V_R\, W_U + V_U\, W_R)\cos(\phi_2)}{\sin^2(\phi_2)} \\ + \; & \left. A_{DR} \frac{V_D\, W_D + V_R\, W_R - (V_D\, W_R + V_R\, W_D)\cos(\phi_3)}{\sin^2(\phi_3)} \right) \Big/ \, 2 \,. \end{aligned}$$

An analysis of this expression is given in [22] and shows that if the a cell is convex then the cell inner product is symmetric and positive definite.

3.6 The New Inner Product

The new inner product is also based on an interpolation, but an interpolation that is more complex than that used for the standard standard inner product, which hopefully produces a more accurate inner product.

3.6.1 The Interpolation Problem

The problem is that we are given the normal components of a vector field on the four faces of the cell, that is, we are given V_D, V_U, V_L and V_R and we are required to find a vector field $\vec{V}(\xi, \eta)$ such that

$$\vec{V}(0,\eta) \cdot \vec{n}_\xi(0) = V_L \,, \qquad \vec{V}(1,\eta) \cdot \vec{n}_\xi(1) = V_R \,, \tag{3.68}$$

$$\vec{V}(\xi,0) \cdot \vec{n}_\eta(0) = V_D \,, \qquad \vec{V}(\xi,1) \cdot \vec{n}_\eta(1) = V_U \,. \tag{3.69}$$

The flux of the vector field through an face of the cell is given by the product of the length of the side of the cell times the normal component of the vector field. Now if each equation in (3.68) is multiplied by the length of an appropriate normal vector, then the flux of the desired vector field must satisfy

$$\vec{V}(0,\eta) \cdot \vec{N}_\xi(0) = S_L\, V_L \,, \qquad \vec{V}(1,\eta) \cdot \vec{N}_\xi(1) = S_R\, V_R \,, \tag{3.70}$$

$$\vec{V}(\xi,0) \cdot \vec{N}_\eta(0) = S_D\, V_D \,, \qquad \vec{V}(\xi,1) \cdot \vec{N}_\eta(1) = S_U\, V_U \,. \tag{3.71}$$

A critical observation is that (3.48) and (3.49) give $\vec{N}_\xi(\xi)$ linear in ξ and $\vec{N}_\eta(\eta)$ linear in η. So if the vector field $\vec{V}(\xi,\eta) = \vec{V}$ is constant, that is, if in Cartesian coordinates, $\vec{V} = (V_x, V_y)$ with V_x and V_y constant, then the normal fluxes $\vec{V}\cdot\vec{N}_\xi(\xi)$ and $\vec{V}\cdot\vec{N}_\eta(\eta)$ are linear functions, which is certainly not true for the normal components of the vector field itself. So we will use linear interpolation on the fluxes so that our interpolation is exact for constant vector fields:

$$\vec{V}(\xi,\eta)\cdot\vec{N}_\xi(\xi) = S_L\,V_L\,(1-\xi) + S_R\,V_R\,\xi\,, \tag{3.72}$$
$$\vec{V}(\xi,\eta)\cdot\vec{N}_\eta(\eta) = S_D\,V_D\,(1-\eta) + S_U\,V_U\,\eta\,. \tag{3.73}$$

Now using the last expression in (3.64) we can represent interpolated vector field in following form

$$\vec{V}(\xi,\eta) = \frac{(S_L\,V_L\,(1-\xi) + S_R\,V_R\,\xi)\;\vec{T}_\xi(\eta) + (S_D\,V_D\,(1-\eta) + S_U\,V_U\,\eta)\;\vec{T}_\eta(\xi)}{J(\xi,\eta)}\,. \tag{3.74}$$

We wrote the interpolated vector field in this form so that it is clear that the formula contains no radicals; the interpolated field is a rational function of the logical space variables, and linear function of our degrees of freedom - V_L, V_R, V_D, V_U. Note that the interpolation (3.74) is exact, not just for constant vector fields, but for any vector fields that have linear flux, as in (3.72).

It is useful to interpret the derived interpolation formula in terms of the theory of mixed finite elements. When constructing finite element spaces for the mixed finite element method, it is common to use the *Piola* transform to define the Raviart-Thomas space of elements. The purpose of the Piola transform is to map vectors in logical space to vectors in physical space in a way that preserves normal fluxes (see paper [1] by Arbogast, Dawson, Keenan, Wheeler, and Yotov).

We will translate the results [1] to our notation and show that there is is simple correspondence between the elements they choose and our interpolated vector fields. We begin by assuming that we are given the components of a vector (W_ξ, W_η) in logical space and then define the components of the vector (W_x, W_y) in physical using the Piola transform given in Formula (4.1) of Section 4 of [1]. In our notation, this is

$$\begin{bmatrix} W_x \\ W_y \end{bmatrix} = \frac{1}{J}\begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}\begin{bmatrix} W_\xi \\ W_\eta \end{bmatrix}. \tag{3.75}$$

This can be rewritten as

$$\vec{W} = \frac{W_\xi\,\vec{T}_\xi + W_\eta\,\vec{T}_\eta}{J} \tag{3.76}$$

Because the Piola transform preserves normal fluxes and W_ξ and W_η are the normal fluxes in logical space,

$$W_\xi = \left(\vec{W}\cdot\vec{n}_\xi\right)|\vec{N}_\xi|\,, \quad W_\eta = \left(\vec{W}\cdot\vec{n}_\eta\right)|\vec{N}_\eta| \tag{3.77}$$

and then (3.76) becomes

$$\vec{W} = \frac{\left(\vec{W} \cdot \vec{n}_\xi\right) \, |\vec{N}_\xi| \, \vec{T}_\xi + \left(\vec{W} \cdot \vec{n}_\eta\right) \, |\vec{N}_\eta| \, \vec{T}_\eta}{J} \tag{3.78}$$

This is the same as our formula (3.64). In [1] W_ξ and W_η are given on the faces of the cell and then W_ξ is interpolated linearly with respects to ξ and W_η is interpolated linearly with respects to η. Then the finite elements in physical space are obtained from the Piola transform.

Our degrees of freedom are the normal components of vectors given on the faces of a cell as in Formula (3.68) and then we linearly interpolate the fluxes as in Formula (3.72). Because of (3.77), this is the same as linearly interpolating W_ξ and W_η which is what is done in [1]. Therefore our our representation of the vector field in the cell in terms of the given data is the same as that obtained using the Piola transform in finite elements. So if we were to use the Piola transform to define the vector fields in physical space, then use the usual dot product for these vector fields, then the resulting dot product will be the same as ours.

Our hope is that using new inner product based on (3.74) in framework of SOM will give us more accurate finite difference method. In the next section we test these ideas numerically.

3.6.2 Formulas for the Cell Inner Product

If $\vec{V}(\xi, \eta)$ is a vector field interpolated from the data $\bar{V}$ using (3.74), and $(\xi(x, y), \eta(x, y))$ is the inverse transformation of (3.40), then $\vec{V}(x, y) = \vec{V}(\xi(x, y), \eta(x, y))$ is the interpolated vector field in physical space. If $\vec{W}$ is another such field, then the cell inner product is

$$\langle \vec{V}, \vec{W} \rangle = \int_{\text{cell}} \vec{V}(x, y) \cdot \vec{W}(x, y) \, dx \, dy = \int_0^1 \int_0^1 \vec{V}(\xi, \eta) \cdot \vec{W}(\xi, \eta) \, J(\xi, \eta) \, d\xi \, d\eta \, . \tag{3.79}$$

This expression is a symmetric bilinear form in terms of our degrees of freedom - V_L, V_R, V_D, V_U and W_L, W_R, W_D, W_U - and thus has at most ten independent components. Expression (3.74) explicitly gives

$$\begin{aligned} \langle \vec{V}, \vec{W} \rangle \;&=\; \left(S_L^2 \int_0^1 \int_0^1 \frac{(1-\xi)^2 \, |\vec{T}_\xi(\eta)|^2}{J(\xi, \eta)} d\xi \, d\eta \right) V_L \, W_L \\ &+ \;\; 2 \left(S_L \, S_R \int_0^1 \int_0^1 \frac{(1-\xi) \, \xi \, |\vec{T}_\xi(\eta)|^2}{J(\xi, \eta)} d\xi \, d\eta \right) V_L \, W_R \\ &+ \;\; 2 \left(S_L \, S_D \int_0^1 \int_0^1 \frac{(1-\xi) \, (1-\eta) \, (\vec{T}_\xi(\eta) \cdot \vec{T}_\eta(\xi))}{J(\xi, \eta)} d\xi \, d\eta \right) V_L \, W_D \\ &+ \;\; \ldots \end{aligned} \tag{3.80}$$

Formula (3.79) makes it clear that the bilinear form is positive definite.

Formula (3.80) shows that all of the integrands have the form

$$\frac{P(\xi,\eta)}{J(\xi,\eta)} \tag{3.81}$$

where the Jacobian J is linear with respect to ξ and η and P is at most a quartic polynomial in ξ and η. Such integrals can be done in closed form and, in fact, P is far from a general quartic polynomial.

What we need for our numerical code is a subroutine that has output the coefficients of the bilinear form and input the coordinates of the vertices of the cell. We created a computer algebra program to derive the formulas for the coefficients of the bilinear form. Attempting a direct computation easily overwhelms two computer algebra systems, so the computation is done by significantly simplifying the integrand before doing the integrals. Also, because the Jacobian has the particular linear form (3.42), the integrals will have a different form if $b = 0$ or $c = 0$ or if both are zero. Because we assume that $J(\xi, \eta)$ is greater than zero on the closed unit square, $a > 0$, so we set $r = b/a$ and $s = c/a$. If we assume that $r\,s \neq 0$, and then the ten resulting integrals have the form

$$A + B\,\log(1+r) + C\,\log(1+s) + D\,\log(1+r+s)\,, \tag{3.82}$$

where A, B, C, and D are polynomials in the coordinates of the corner point of the cells. In fact, because the formulation is translation invariant, the coefficients are polynomials in the squares of the distances between the corner points of the cells. A subroutine based on these formulas suffers seriously from catastrophic cancellation when r, s, or both are small. Consequently, we also derived formulas for these cases using Taylor series in r and s to remove the singularities. The careful simplification of the resulting formulas is critical for creating a numerically stable subroutine.

4 SOM in General Grids

The importance of the new inner product is its possible use in more accurately solving boundary value problems, so we will compare the accuracy of the solution and fluxes for the Poisson equation when we use the standard and new inner products in the SOM. To apply the SOM, we write the Poisson equation as a first order system equations

$$\mathbf{div}\,\vec{W} = f\,, \tag{4.83}$$

$$\vec{W} = -K\,\mathbf{grad}\,u\,, \tag{4.84}$$

and then use Dirichlet boundary conditions on u to define the the boundary value problem. Here $\vec{W}$ is the flux, f is the source, K is the conductivity tensor, and u is the potential.

Recall the the SOM need us to define two discrete inner products and a discretization of the divergence operator. We have defined two inner product for vector fields, and the cell inner product for two scalars f_C and g_C, defined at a cell center, is

$$f_C\,g_C\,A \tag{4.85}$$

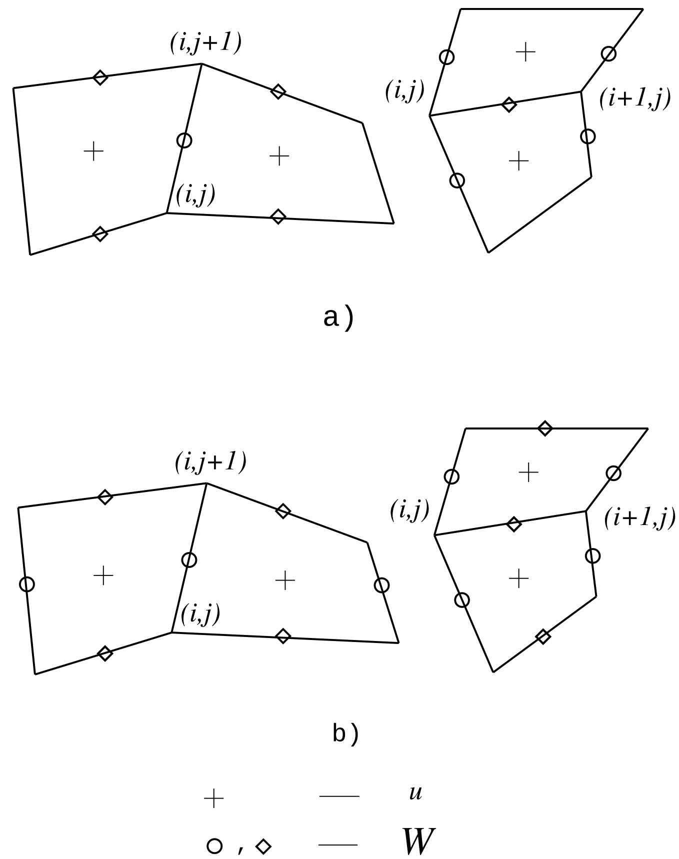


Figure 4.1: Stencils for standard and new discrete gradients

where A is the area of the cell. In terms of the normal components of vectors, the discretization of the divergence is

$$\mathbf{div}\,\vec{W} \approx \frac{(W_R\,S_R - W_L\,S_L) + (W_U\,S_U - W_D\,S_D)}{A}\,. \tag{4.86}$$

The discretization of the balance equation (4.83) is especially simple and similar to one for tensor product grids (2.33)

$$\frac{(W_R\,S_R - W_L\,S_L) + (W_U\,S_U - W_D\,S_D)}{A} = f_C\,,$$

where A is area of the quadrilateral and f_C is the value of function f at the center of the cell. Given the discrete inner products and divergence, the SOM automatically generates a discretization of the flux operator $-K\,\mathbf{grad}$.

The structure of the discrete analog of flux equation (4.84) for general grid is also similar to one for tensor product grid (2.35), however now the two equations defining the flux contain fluxes from both the up and down sides and the left and right sides. In Figure 4.1 we display the stencils for computing the fluxes. In both a) and b) the left figure gives the stencil of the first equation while the right figure gives the stencil for the left equation (the equation at the central circle of diamond involves all of the marked points). So the stencil for the new inner product is only a slight extension of the stencil for the standard inner product.

We have an existing code that implements the SOM in general grids that is described in detail in [20]. This code only needs the stencils for the inner products for scalars, vectors, and the divergence and automatically derives the needed information about the flux. So no closed-form formula is needed for the flux. Our computer algebra program provided the formulas for and a subroutine to evaluate the stencils for the new inner product.

5 Numerical Results

Our final goal is not to just have a more accurate inner product or gradient, but to have an more accurate solution of Laplace equation based on these ingredients, so we tested the new ideas on five Laplace equation examples. Note that the SOM method has been extensively tested in [21, 22], so this section only contains a few examples to illustrate the convergence rates for the fluxes and to compare the two inner products. The test problems involve three grids, a smooth grid, a randomly disturbed grid, and a non-differentiable grid. There are two test solutions, one that is smooth and one that is piecewise smooth, continuous but not continuously differentiable.

A smooth grid is obtained by mapping a of uniform grid in the unit cube $[0\,,1] \times [0\,,1]$ in the $(\xi,\, \eta)$ computational space into the unit cube in the (x, y) physical space using

$$x(\xi,\, \eta) = \xi + 0.1\, \sin(2\,\pi\,\xi)\, \sin(2\,\pi\,\eta)\,, \quad y(\xi,\, \eta) = \eta + 0.1\, \sin(2\,\pi\,\xi)\, \sin(2\,\pi\,\eta)\,. \tag{5.87}$$

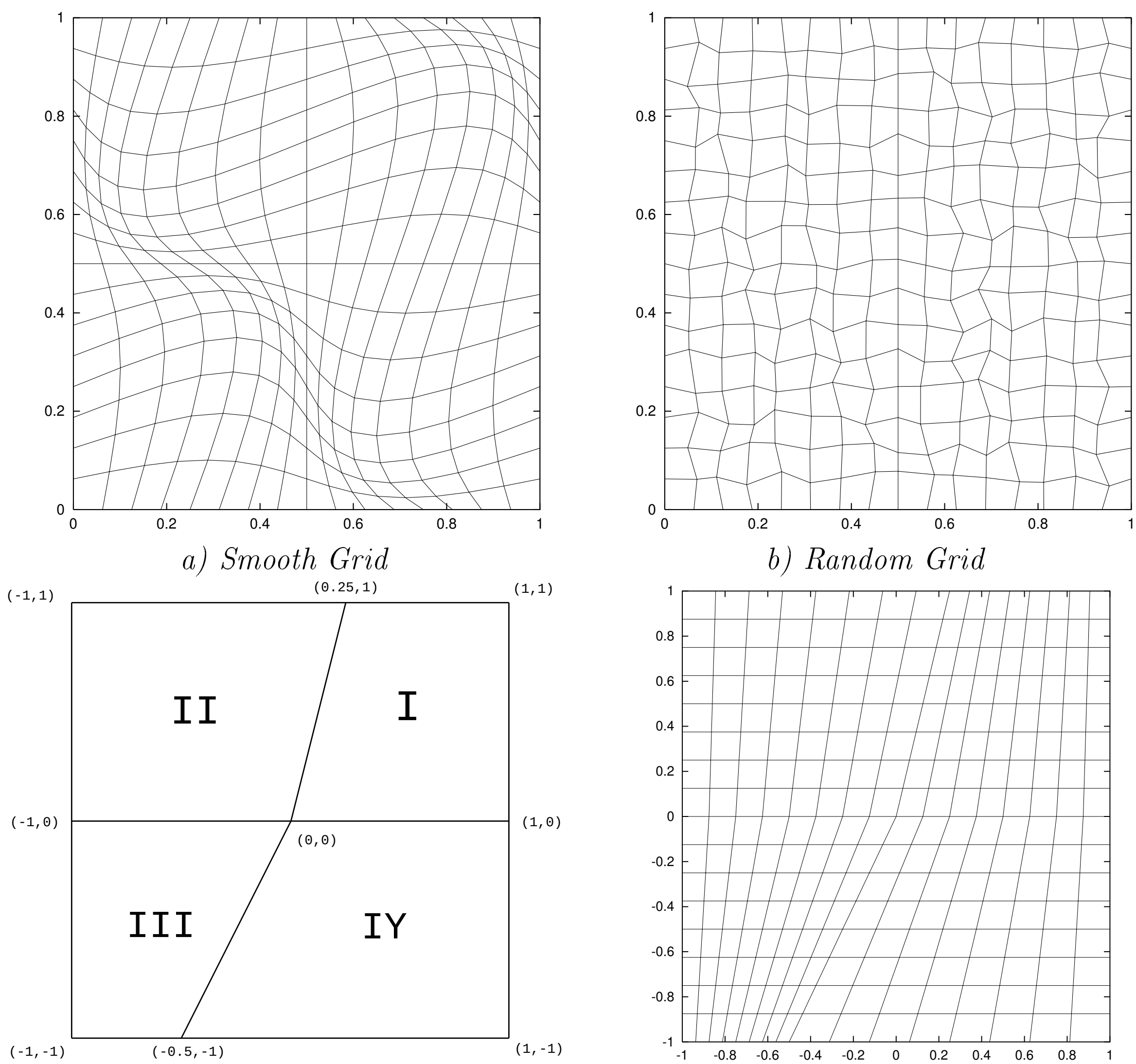
A random (non-smooth) grids such as the one shown in Figure 5.1 b) are obtained by moving the nodes of a uniform square grid in uniformly-distributed random directions and with amplitude uniformly distributed in an interval 20% of initial grid size.

The piece-wise smooth grid given in Figure 5.1 d) was used in the paper [2]. Note that block grids play an important role in many applications where the material changes abruptly, such as in ground-water modeling. This grid obtained by dividing the computational domain into four blocks as shown on Figure 5.1 c) and then a grid is generated on each block using transfinite interpolation [15]. Consequently, except in trivial cases, the global map of unit square to computational domain is continuous but not differentiable on the internal boundaries of each block.

The test problems use the inhomogeneous steady-state diffusion equation with inhomogeneous Dirichlet boundary conditions, where the inhomogeneous terms are computed from a given exact solution. The smooth solution is

$$u(x, y) = x^2 + y^2\,, \tag{5.88}$$

in domain $[0, 1] \times [0, 1]$ and constant diffusion coefficient. The piecewise-smooth test problem is taken from MacKinnon and Carey [17] (see also [22]), where the diffusion equation has a



a) Smooth Grid

b) Random Grid

b) Blocks

d) Piecewise Smooth Block Grid

Figure 5.1: The Test Grids

constant right-hand side of one, and the diffusion coefficient is equal to k_1 for $x < 0$ and to k_2 for $x > 0$ (we use $k_1 = 1$ and $k_2 = 5$). The exact solution is one dimensional:

$$u(x) = \begin{cases} a_1 \frac{x^2}{2} + b_1 x \,, & 0 \le x \le \frac{1}{2} \,, \\ a_2 \frac{x^2}{2} + b_2 x + c_2 \,, & \frac{1}{2} \le x \le 1 \,, \end{cases} \,, \tag{5.89}$$

where

$$a_i = \frac{-1}{k_i} \,, \quad b_1 = -0.25\,(3\,a_2 + a_1)\,\frac{k_2}{k_1 + k_2} \,, \quad b_2 = \frac{k_1}{k_2}\,b_1 \,, \quad c_2 = -(b_2 + 0.5\,a_2) \,. \tag{5.90}$$

The errors for both the solution and the flux are computed using L_2 norms. The continuum solution was discretized by taking the function values at the center of the cell to be the average of the function over the cell, while the normal flux was taken as point values of continuum normal flux at the center of a cell face. We now give the data for five test problems:

- Problem 1 (Table 5.1) uses the smooth grid 5.1 a) and the smooth solution (5.88).
- Problem 2 (Table 5.2) uses the piece-wise smooth grid 5.1 d) and the smooth solution (5.88).
- Problem 3 (Table 5.3) uses the random grid 5.1 b) and the smooth solution (5.88).
- Problem 4 (Table 5.4) uses the smooth grid 5.1 a) and the piece-wise smooth solution (5.89).
- Problem 5 (Table 5.5) uses the random grid 5.1 b) and the piece-wise smooth solution (5.89).

The conclusions are that the solution values have a second-order convergence rate for both inner products but the new inner product is 2-3 times more accurate. On smooth and piece-wise smooth grids the fluxes have a second-order convergence rate for both inner products. On random grids the fluxes have a first-order convergence rate for both inner products. In both cases the new inner product gives more accurate fluxes. Note that second-order accuracy for the solutions using the standard inner product was confirmed in [22].

References

[1] T. Arbogast, C.N. Dawson, P.T. Keenan, M.F. Wheeler, and I. Yotov, *Enhanced Cell-Centered Finite Differences for Elliptic Equations on General Geometry*, SIAM Journal of Scientific Computing, **18** (1997), 1-32.

[2] Z. Cai, J.E. Jones, S.F. McCormick and T.F. Russel, *Control-volume mixed finite element methods*, Computational Geosciences, **1** (1997), 289-315.

Method	M	L_2-norm for function	L_2-norm for flux
Low	9	4.86E-3	3.13E-2
	17	1.29E-3	9.29E-3
	33	3.33E-4	2.44E-3
High	9	2.89E-3	1.11E-2
	17	7.79E-4	2.95E-3
	33	1.98E-4	7.50E-4

Table 5.1: Problem 1: Accuracy for Smooth Solution and Smooth Grid

Method	M	L_2-norm for function	L_2-norm for flux
Low	9	3.51E-2	4.30E-2
	17	8.78E-3	1.22E-2
	33	2.19E-3	3.38E-3
High	9	9.95E-3	2.19E-2
	17	2.50E-4	7.53E-3
	33	6.27E-4	1.94E-3

Table 5.2: Problem 2: Accuracy for Smooth Solution and Piece-Wise Smooth Grid

Method	M	L_2-norm for function	L_2-norm for flux
Low	9	4.19E-3	1.65E-2
	17	1.07E-3	8.94E-3
	33	2.84E-4	4.55E-3
High	9	1.63E-3	8.17E-3
	17	4.45E-4	4.18E-3
	33	9.91E-5	2.20E-3

Table 5.3: Problem 3: Accuracy for Smooth Solution and Random Grid

Method	M	L_2-norm for function	L_2-norm for flux
Low	9	1.27E-3	1.56E-2
	17	3.57E-4	4.71E-3
	33	9.35E-5	1.25E-3
High	9	4.51E-4	4.08E-3
	17	1.22E-4	1.13E-3
	33	3.11E-5	2.94E-4

Table 5.4: Problem 4: Accuracy for Piece-Wise Smooth Solution and Smooth Grid

Method	M	L_2-norm for function	L_2-norm for flux
Low	9	1.05E-3	9.09E-3
	17	2.83E-4	3.75E-3
	33	7.04E-5	1.89E-3
High	9	3.47E-4	3.67E-3
	17	9.58E-5	1.48E-3
	33	2.33E-5	7.17E-4

Table 5.5: Problem 5: Accuracy for Piece-Wise Smooth Solution and Random Grid

[3] E.J. Caramana, D.E. Burton, M. Shashkov, and P.P. Whalen, *The Construction of Compatible Hydrodynamics Algorithms Utilizing Conservation of Total Energy*, Journal of Computational Physics, 146 (1998), 227-262.

[4] J. Castillo, J.M. Hyman, M. Shashkov and S. Steinberg, *The Sensitivity and Accuracy of Fourth Order Finite-Difference Schemes on Nonuniform Grids in One Dimension*, An International Journal of Computers & Mathematics with Applications, **30** (1995), 41-55.

[5] J. Castillo, J. Hyman, M. Shashkov and S. Steinberg, *High-Order Mimetic Finite Difference Methods on Nonuniform Grids*, in Proceedings of Third International Conference on Spectral and High-Order Methods, Houston, Texas June 5-9, 1995, Edited by A. V. Ilin and R. R. Scott, Published by Houston Journal of Mathematics, 1996, 347-361.

[6] A. Favorskii, A. Samarskii, M. Shashkov and V. Tishkin, *Operational Finite-Difference Schemes*, Differential Equations, **17**, (1981), 854-862.

[7] J.M. Hyman, R.J. Knapp, and J.C. Scovel, *High Order Finite Volume Approximations of Differential Operators on Nonuniform Grids*, Physica D **60** (1992), 112-138.

[8] J. Hyman, M. Shashkov and S. Steinberg, *Problems with Heterogeneous and Non-Isotropic Media on Distorted Grids*, Proceedings of First International Symposium on "Finite Volumes for Complex Applications", Problems and Perspectives, July 15-18, 1996, Rouen, France, Edited by F. Benkhaldoun and R. Vilsmeier, Hermes, Paris, 1996, 249-260.

[9] J.M. Hyman and M. Shashkov, *Natural Discretizations for the Divergence, Gradient, and Curl on Logically Rectangular Grids*, International Journal of Computers & Mathematics with Applications, **33** (1997), 81-104.

[10] J.M. Hyman, M. Shashkov and S. Steinberg, *The Numerical Solution of Diffusion Problems in Strongly Heterogenous Non-Isotropic Materials*, Journal of Computational Physics, **132** (1997), 130-148.

[11] J.M. Hyman and M. Shashkov, *The Adjoint Operators for Natural Discretizations for the Divergence, Gradient and Curl on Logically Rectangular Grids*, IMACS Journal - Applied Numerical Mathematics, **25** (1997), 413-442.

[12] J.M. Hyman and M. Shashkov, *Mimetic Discretizations for Maxwell's Equations and Equations of Magnetic Diffusion*, in "Mathematical and and Numerical Aspects of Wave Propagations", Edited by J.A. DeSanto, Proceedings of the Fourth International Conference on Mathematical and Numerical Aspects of Wave Propagation, Golden, Colorado, June 1-5, 1998, SIAM, Philadelphia, 561-563.

[13] J.M. Hyman and M. Shashkov, *The Approximation of Boundary Conditions for Mimetic Finite Difference Methods*, Computers and Mathematics with Applications, **36** (1998), 79-99.

[14] J. Hyman and M. Shashkov, *Mimetic Discretizations for Maxwell's Equations*, Journal of Computational Physics, **151** (1999), 881-909.

[15] P.M. Knupp and S. Steinberg, *The Fundamentals of Grid Generation*, CRC Press, Boca Raton, 1993.

[16] P.M. Knupp, *On the Invertibility of the Isoparametic Map*, Computational Methods Applied to Mechanical Engineering, **78** (1990), 313-329.

[17] R. J. MacKinnon and G. F. Carey. *Analysis of Material Interface Discontinuities and Superconvergent Fluxes in Finite Difference Theory*, Journal of Computational Physics, **75** (1988), 151-167.

[18] L. Margolin and M. Shashkov, *Using a Curvilinear Grid to Construct Symmetry-Preserving Discretization for Lagrangian Gas Dynamics*, Journal of Computational Physics, **149** (1999), 389-417.

[19] L. Margolin, P. Smolarkiewicz and M. Shashkov, *A Discrete Operator Calculus for Finite Difference Approximations*, Report LA-UR-98-2835 of Los Alamos National Laboratory, Los Alamos, New Mexico, USA, to appear at Journal of Computer Methods in Applied Mechanics and Engineering.

[20] J.E. Morel, R.M. Roberts, and M. Shashkov, *A Local Support-Operator Diffusion Discretization Scheme for Quadrilateral r-z Meshes*, Journal of Computational Physics, **144** (1998), 17-51.

[21] M. Shashkov, and S. Steinberg, *Support-Operator Finite-Difference Algorithms for General Elliptic Problems*, Journal of Computational Physics, **118** (1995), 131-151.

[22] M. Shashkov, and S. Steinberg, *Solving Diffusion Equations with Rough Coefficients in Rough Grids*, Journal of Computational Physics, **129** (1996), 383-405.

[23] M. Shashkov, *Conservative Finite-Difference Methods on General Grids*, CRC Press, Boca Raton, 1995.