Figure 7: Isotherms for steady state heat conduction solution on fin geometry and grid of Fig. 5 with $N = 5$

Figure 8: Channel flow solution superimposed on the triangular grid using four elements

Figure 5: Spectral element grid for a fin geometry using 34 elements

Figure 6: Convergence plot for solution of Poisson's equation on a triangular grid

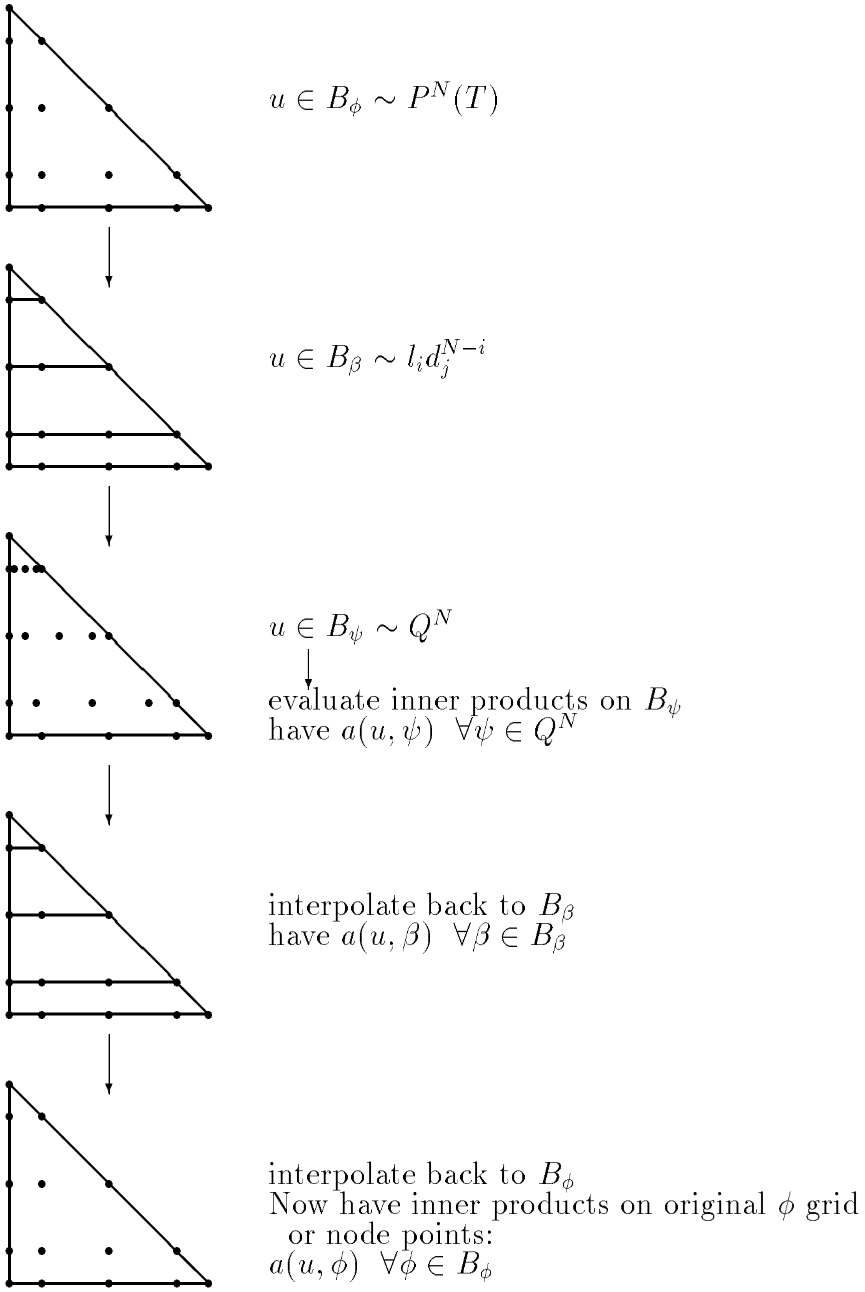
$u \in B_\phi \sim P^N(T)$
$u \in B_\beta \sim l_i d_j^{N-i}$
$u \in B_\psi \sim Q^N$
evaluate inner products on B_ψ
have $a(u,\psi)$ $\forall \psi \in Q^N$
interpolate back to B_β
have $a(u,\beta)$ $\forall \beta \in B_\beta$
interpolate back to B_ϕ
Now have inner products on original ϕ grid
or node points:
$a(u,\phi)$ $\forall \phi \in B_\phi$

Figure 4: Total residual evaluation procedure

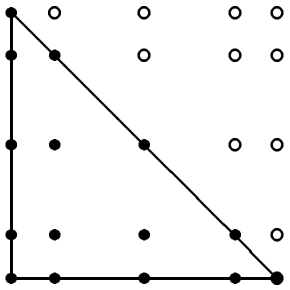

Figure 2: Nodal points used on the triangular spectral elements

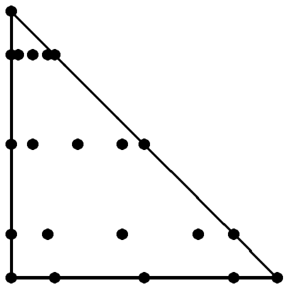

Figure 3: Quadrature points used on the triangular spectral elements

Figure 1: Elemental mappings for a) standard quadrilateral spectral elements and b) triangular spectral elements

REFERENCES

1. Patera, A. T., A spectral element method for fluid dynamics: Laminar flow in a channel expansion, *Journal of Computational Physics*, 54, 1984, pp. 468.

2. Fischer, P. F. and Rønquist, E. M., Spectral element methods for large-scale parallel Navier-Stokes calculations, *Computer Methods in Applied Mechanics and Engineering*, to appear.

3. Maday, Y., Mavriplis, C., and Patera, A. T., Nonconforming mortar element methods: application to spectral discretizations, *Domain Decomposition Methods*, SIAM, Philadelphia, 1989, pp.392.

4. Mavriplis, D., *AIAA Paper 91-1549CP*, 1991.

5. Babushka, I. and Aziz, A.K., *On the Angle Condition in the Finite Element Method*, SIAM Journal of Numerical Analysis, 13, 2, 1976.

6. Dubiner, M., Spectral Methods on Triangles and Other Domains, *Journal of Scientific Computing*, 6, 4, 1991, pp. 345-390.

7. Funaro, D., Pseudospectral Approximation of a PDE Defined on a Triangle, *Applied Mathematics of Computation*, 42, 1991, pp. 121-138.

6. TWO-DIMENSIONAL FLUID FLOW ILLUSTRATION

The Navier-Stokes triangular spectral element solver is illustrated in Figure 8 by a solution for a laminar channel flow. The four element triangular grid is shown along with the velocity profile. The velocity profile is exact (parabolic). This solution was obtained from an initial solution which was perturbed by 10% from the exact solution and allowed to march in time until it stabilized itself to the exact laminar parabolic velocity solution. While the geometry is simple, more complex geometry solutions will be available by the conference.

7. CONCLUSION

In this paper, we have generalized the spectral element method to grids containing triangular elements. Triangular elements are important since they provide greater geometric flexibility than quadrilateral elements. They have, however, not been used previously since efficient tensor product algorithms were not available for triangles. The algorithms given here remedy this, allowing one to design efficient spectral element programs using triangular elements. The order of complexity of our algorithms for triangular elements is identical to that for quadrilateral elements, though the constants in these complexity bounds are poorer. Thus a natural approach is to combine triangular and quadrilateral spectral elements using the efficient quadrilateral elements in the domain interior, and switching to triangles as needed along complex boundaries for geometric flexibility in modeling incompressible fluid flow.

$$\nabla \cdot \vec{u} = 0,$$

where $\vec{u}$ is the velocity vector, p is the pressure, and R is the Reynolds number, are solved by a fractional time-stepping scheme as follows:

$$\frac{\hat{u} - \vec{u}^n}{\Delta t} = \vec{u} \cdot \nabla \vec{u}$$

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot \hat{u}$$

$$\frac{\hat{\hat{u}} - \hat{u}}{\Delta t} = -\nabla p$$

$$\frac{\vec{u}^{n+1} - \hat{\hat{u}}}{\Delta t} = \frac{1}{R} \nabla^2 \vec{u}.$$

This scheme permits the decoupling of the pressure and velocity, which, conveniently, can both be solved by Poisson solvers. The nonlinear convective terms are treated explicitly by a third order Adams-Bashforth scheme as follows:

$$(\vec{u} \cdot \nabla \vec{u})^n = \sum_{q=0}^{2} \alpha_q (\vec{u}^{n-q} \cdot \nabla \vec{u}^{n-q})$$

$$\alpha_0 = 23/12 \quad \alpha_1 = -16/12 \quad \alpha_2 = 5/12.$$

And hence the viscous terms are solved implicitly. The discretized Poisson equations are solved iteratively by a preconditioned conjugate gradient method. The preconditioners are simply the diagonal matrices.

All equations are solved in variational form since derivatives cannot be taken on the nodal point basis. Hence, for example, the pressure Poisson solver may be written as

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot \hat{u} = \frac{1}{\Delta t} \nabla \cdot (\vec{u}^n + \Delta t \;\; \vec{u} \cdot \nabla \vec{u})$$

which taken term by term gives:

$$\nabla^2 p \Longrightarrow (\nabla^2 p, \phi) = -(\nabla p, \nabla \phi) + \text{boundary terms}$$

$$\frac{1}{\Delta t} \nabla \cdot u^n \Longrightarrow \frac{1}{\Delta t} (\nabla \cdot u^n, \phi)$$

$$\nabla \cdot (\vec{u} \cdot \nabla \vec{u}) \Longrightarrow (\nabla \cdot (\vec{u} \cdot \nabla \vec{u}), \phi) = -(\vec{u} \cdot \nabla u, \phi_x) - (\vec{u} \cdot \nabla v, \phi_y) + \text{b.t.}$$

where $\vec{u} = u\vec{i} + v\vec{j}$. Again, all these calculations are implemented with isoparametric mappings for arbitrarily shaped triangular elements.

order	time to compute element residual	nodes in element	time per node divided by order
2	2.2 ms	6	.183
4	8.5 ms	15	.139
8	48 ms	45	.133
16	338 ms	153	.139

Table 1: Execution time for residual calculation on triangular spectral elements

order of the polynomial. This verifies that the calculation requires $O(N^3)$ operations: to be exact the number of nodes is $(N+1)(N+2)/2$ so the number of operations is $O(N(N+1)(N+2)/2) = O((N^3+3N^2+2N)/2)$.

4. VALIDATION

The triangular spectral element Poisson solver is validated on the grid shown in Figure 5. The grid contains 34 triangular elements of varying shapes. Deformed elements are mapped to the right triangle T via isoparametric mappings. Figure 6 represents a convergence plot for the solution of Poisson's equation $\nabla^2 u = f$ on the grid of Figure 5 with Dirichlet boundary conditions and $f = -3.25\pi^2 \sin \pi x \sin \frac{3}{2}\pi y$. The straight line convergence in the log-linear plot of error versus N the order of the polynomial indicates that exponential convergence has been achieved. Hence we have developed suitable triangular elements which preserve the exponential accuracy of spectral type methods.

Figure 7 illustrates a steady state heat conduction solution in a fin using the same grid with $N = 5$ (Fig. 5): $\nabla^2 T = q$ where $q = 0$, the upper and side boundaries of the fin root are held at $T_1 = 1$, the lower fin root boundaries are linearly varying from $T_1 = 1$ to $T_o = 0$, and the fin surfaces are held at $T_o = 0$. The solution is shown in terms of isotherms.

5. NAVIER-STOKES TRIANGULAR SPECTRAL ELEMENT IMPLEMENTATION

Direct simulation of incompressible flow is achieved by the high accuracy solution of the full unsteady non-linear Navier-Stokes equations using triangular spectral elements. The scheme follows standrad spectral element procedures described in[1,2,3], hence the discussion will remain brief. The Navier-Stokes equations

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\nabla p + \frac{1}{R}\nabla^2 \vec{u}$$

Finally, for the tensor product polynomials Q^N we define the basis functions

$$\psi_{ij} \equiv l_i \, q_j$$

and the basis

$$B_\psi = \{ \psi_{ij} \}_{(i,j) \in \square}$$

where $\square$ was defined in Equation (2).

<u>Outline of Computation</u>

Given the values of a trial function u at the nodes of the triangle T, the goal is to evaluate the stiffness inner products:

$$\{ a(u, \phi_{ij}) \}_{(i,j) \in \Delta}.$$

This could obviously be done by direct quadrature of the trial solution u with each of these test functions but the cost would be $O(N^4)$. There are $(N+1)^2$ quadrature points and $N^2 + 3N + 3$ test functions. The alternative is to convert from the basis B_ϕ to a more convenient basis, then compute quadratures by a fast tensor product algorithm. The resulting algorithm is more complicated, but requires only $O(N^3)$ operations, as in the case of quadrilateral spectral elements.

The first step is to convert the trial solution u expressed in the basis B_ϕ for $P^N(T)$ to the basis B_ψ for Q^N via the operations

$$B_\phi \Rightarrow B_\beta \Rightarrow B_\psi.$$

This can always be done, since $P^N \subset Q^N$. Next we evaluate the stiffness inner products:

$$\{ a(u, \psi_{ij}) \}_{(i,j) \in \square}.$$

Finally, these inner products are converted back to the basis B_β and then to B_ϕ:

$$\{ a(u, \psi_{i,j}) \}_{(i,j)\in\square} \Rightarrow \{ a(u, \beta_{ij}) \}_{(i,j)\in\Delta} \Rightarrow \{ a(u, \phi_{ij}) \}_{(i,j)\in\Delta}$$

so that we have the stiffness inner products $\{ a(u, \phi_{ij}) \}_{(i,j)\in\Delta}$ as required.

This is, in outline, the residual calculation. The total residual evaluation may be summarized in graphical format as in Figure 4.

The total cost here is $O(N^3)$, since each step is a tensor product operation requiring $O(N^3)$ operations. The constant in this $O(N^3)$ bound is, however, larger than in the case of quadrilateral elements. The total number of arithmetic operations is about double that of quadrilateral elements of order N. Table 1 presents timings for the calculation of one residual on triangular spectral elements. The time per node is shown to be proportional with the then

$$p(x) \;=\; \sum_{i=0}^{N} a_i \; l_i(x).$$

In particular, since

$$d_i^k(\zeta_j) = \delta_{ij}, \quad 0 \leq j \leq k$$

we have

$$d_i^k(x) \;=\; \sum_{j=0}^{m} h_{ji} \; l_j(x).$$

Polynomial Bases in Two Dimensions

In two dimensions there are two polynomial spaces of interest, the space of polynomials of degree N, and the space of tensor product polynomials:

$$\begin{aligned} P^N(T) &\;=\; \text{polynomials of order } N \text{ on } T \\ Q^N &\;=\; \text{tensor product polynomials of order } N \end{aligned}$$

where T is the right triangle of side one shown in Figure 1b. Q^N is the space of polynomials of degree N in x and y separately. Thus we have the natural embedding:

$$P^N(T) \;\subset\; Q^N$$

The notation $P^N(T)$ for the space of polynomials in two dimensions is used to avoid confusion with our notation P^N, for the space of polynomials in one dimension.

Two different sets of bases functions are used for the space $P^N(T)$. First, given our choice of the tensor product Gauss Lobatto points $\{(\zeta_i, \zeta_j) \;\mid\; (i,j) \in \Delta\}$ as nodes (see Equation (1) and Figure 2), we define "nodal" basis functions:

$$\phi_{ij} \;\;\equiv\;\; p \in P^N(T) \;\mid\; p(\zeta_l, \zeta_k) = \delta_{il}\;\delta_{jk}.$$

The corresponding basis for $P^N(T)$ is

$$B_\phi = \{\; \phi_{i,j} \;\}_{\;(i,j) \,\in\, \Delta}$$

and is the most natural basis here, though it is awkward for computation. An alternate set of basis functions for $P^N(T)$ is the set of polynomials

$$\beta_{ij} \;\equiv\; l_i \; d_j^{N-i}$$

with the corresponding basis:

$$B_\beta = \{\; \beta_{ij} \;\}_{\;(i,j) \,\in\, \Delta}.$$

Polynomial Bases in One Dimension

Let N be the order of the spectral element. Define the one dimensional polynomial spaces:

$$\begin{aligned} P^N \quad &= \quad \text{polynomials on } [0,1] \text{ of order } N \\ P^k,\ k \le N \quad &= \quad \text{polynomials of order } k \end{aligned}$$

The N-th order Lagrangian interpolation polynomials with zeroes at the Gauss Lobatto points are

$$q_i \quad \equiv \quad p \in P^N \mid p(\zeta_j) = \delta_{ij},\ 0 \le j \le N$$

where δ_{ij} is the Kronecker delta.

We also need the Lagrange basis functions for P^k with nodes at the first k of the Gauss Lobatto points $\{\ \zeta_i\ \}_{i=0}^N$

$$d_i^k \quad \equiv \quad p \in P^k \mid p(\zeta_j) = \delta_{ij},\ 0 \le j \le k.$$

Finally define the polynomials

$$l_i \quad = \quad d_i^i$$

Then we have the relations:

$$q_i \quad = \quad d_i^N$$

$$P^N \quad = \quad \text{span}\{\ q_i\ \}_{i=0}^N \quad = \quad \text{span}\{\ l_i\ \}_{i=0}^N$$

$$P^k \quad = \quad \text{span}\{\ d_i^k\ \}_{i=0}^k \quad = \quad \text{span}\{\ l_i\ \}_{i=0}^k$$

For later use, we form the matrix whose elements are the values of the basis functions $\{l_k\}_{k=0}^N$ at the Gauss Lobatto points. Define the matrix G having elements:

$$g_{ik} \ = \ l_k(\zeta_i), \quad \forall\ i,k, \quad 0 \le i,k \le N.$$

Each l_k is zero on $\{\zeta_i\}_{i=0}^{k-1}$, one at ζ_k, and greater than one at the remaining Gauss Lobatto points. Thus G is a lower triangular matrix with unit diagonal.

Since G is a nonsingular triangular matrix, its inverse is easy to compute. Let $H = \{h_{ij}\}$ be the inverse of G. The matrix H gives the transformation from function values at the Gauss Lobatto points to polynomials expressed in the basis $\{l_k\}_{k=0}^N$. That is, for any $p \in P^N$, if

$$a_i \ = \ \sum_{k=0}^N h_{ik}\ p(\zeta_k),$$

Compatibility of this triangular element with the quadrilateral spectral elements is apparent for the edges lying along the lines $x = 0$ and $y = 0$. We also have the Gauss Lobatto point spacing along the line $x + y = 1$, so this edge is also compatible with the quadrilateral elements, and with any edge of other triangular elements.

Quadrature Points

There is some degree of latitude in the selection of quadrature points within the triangular elements. Any choice that guarantees accurate integration of polynomials of sufficient order will be adequate. There are many families of quadrature formulas designed specifically for triangles[5]. However, formulas specific to triangles do not have the regular pattern of quadrature points needed for exploitation of tensor product structure. We choose instead to use the tensor product Gauss Lobatto formulas. While this approximately doubles the number of quadrature points used, as shown in Figure 3, it does allow exploitation of tensor product structure. In the case of high order elements, exploitation of tensor product structure is crucial. Figure 3 illustrates the choice of quadrature points on the reference triangle T. They are the points

$$\{(\eta_i^j, \zeta_j)\}_{(i,j)\in\square}$$

where the y coordinates (ζ_j) are the same as for the nodal points and the x coordinates (η_i^j) are different for each j or constant y line. On each j line the coordinates of the quadrature points are

$$\eta_i^j = (1 - \zeta_j)\ \zeta_i$$

which are just the usual Gauss Lobatto points on the square mapped into our triangular domain.

3. RESIDUAL CALCULATION

The key to the efficiency of the spectral element method is the existence of fast tensor product algorithms for evaluation of residuals. In this section we describe our tensor product algorithm for the residual calculation on triangles. We first describe the polynomial bases needed and outline the algorithm. Then we describe each of the steps involved in relative detail.

In extending the spectral element method to triangular elements, it is natural to design a nodal element method, since this is simpler than directly enforcing the C^0 continuity condition between elements, and simplifies boundary treatments. Moreover, by choosing a set of nodes compatible with those used for quadrilateral spectral elements, we can make a triangular element that can be freely intermixed with quadrilateral elements.

For quadrilateral elements, one defines a standard element on the unit square, as shown in Figure 1a. Curvilinear elements needed in the grid are then constructed by mapping to this standard element. The same approach is followed with triangles, as shown in Figure 1b: arbitrary triangles are mapped to the right triangle T of sides $[0, 1]$.

Nodal Points

We take as nodal points the tensor product Gauss Lobatto points lying within this triangle, as shown in Figure 2. If the quadrature points for the $2N$-order Gauss Lobatto formula on the interval $[0, 1]$ uses points

$$Z = \{\zeta_i\}_{i=0}^N$$

we take as nodes, the points:

$$\{(\zeta_i, \zeta_j) \mid 0 \leq i, j \leq N, \quad i + j \leq N\}.$$

To denote these points on the triangle, we will use the notation $(\zeta_i, \zeta_j)_{(i,j)\in\Delta}$ where

$$\Delta = \{(i, j) \mid 0 \leq i, j \leq N, \quad i + j \leq N\}. \tag{1}$$

In order to avoid confusion we will adopt the notation $(\zeta_i, \zeta_j)_{(i,j)\in\square}$, where

$$\square = \{(i, j) \mid 0 \leq i, j \leq N\} \tag{2}$$

for the usual index range.

Let Γ be the unit interval $[0, 1]$. The polynomial basis functions on Γ are chosen to be the Lagrangian interpolants $h_i \in P^N(\Gamma)$ satisfying:

$$h_i(\zeta_j) = \delta_{i,j}$$

where $\delta_{i,j}$ is the Kronecker delta. They have the explicit representation:

$$h_i(z) = -\frac{1}{N(N+1)L_N(\zeta_i)} \frac{(1 - z^2)L_N'(z)}{z - \zeta_i},$$

where L_N is the Legendre polynomial of order N and ζ_i, $i = 0, \ldots, N$ are the roots of L_N'. The representation of these nodal points on the reference triangle T is given in Figure 2.

imum vertex angle is bounded from below, if the minimum angle approaches zero as the grid is refined, inconsistency occurs. Even when quadrilaterals maintain consistency, the discretization error using triangular elements is always better when small angles occur.

The geometric flexibility and favorable approximation properties of triangular elements are widely appreciated in the context of finite elements. However, since the fast tensor product algorithms used with quadrilateral elements do not apply to triangular elements, triangular spectral elements do not seem to have been used until now. A fast spectral element method for triangles and other regions was derived previously by Dubiner[6], but is quite complex, and is rarely used. Also, Funaro[7] developed a technique for treating triangles for spectral multidomain methods but does not advocate its use.

Dubiner's approach is based on construction of an orthonormal basis for the polynomials on a triangle. He shows that there are orthonormal bases that are both well conditioned, and have sparse stiffness matrices, leading to fast algorithms. However, to provide an easy means of enforcing inter-element continuity, his basic method needs to be modified. He thus constructs a new basis consisting of interior shape functions, vanishing on the element boundary, together with boundary functions used to enforce C^0 interelement continuity.

Our approach and Dubiner's both yield compution cost $O(N^3)$ to evaluate the residual on a triangle with $(N+1)N/2$ degrees of freedom, and both appear relatively stable and well conditioned. The central advantage of our approach is its comparative simplicity. In particular, incorporation of the new triangular elements into existing spectral element codes is relatively straightforward.

The remainder of the paper is organized as follows. In the next section we describe the design of the triangular elements to be used in conjunction with our spectral discretization. Section 3 outlines the residual calculation with fast tensor products on the triangular elements. An example calculation is then given in section 4 to show that spectral accuracy is retained on the new triangular elements. Section 5 describes the implementation of triangular elements in the solution of the incompressible Navier-Stokes calculations, and a fluid flow example follows.

2. DESIGN OF TRIANGULAR ELEMENTS

The spectral element method is a high order generalization of finite elements. The standard C^0 quadrilateral finite element method takes as trial space the tensor product space Q^N of polynomials with terms of degree at most N in either x or y. For triangular elements, one takes instead the space P^N of polynomials in x and y of degree N. Since quadrilateral spectral elements are based on the trial space Q^N too, we use the space P^N for triangular spectral elements. It seems to be the only natural choice.

1. INTRODUCTION

The spectral element method is a relatively new method which provides both the geometric flexibility of finite element methods and the exponential accuracy of spectral methods[1,2,3]. It is currently being used for direct simulation of incompressible fluid flow. With this method, the full Navier-Stokes equations can be solved in relatively complex geometries. Our goal is to increase the flexibility of the method, to enable high accuracy direct simulation of flows in complex geometries to be done efficiently.

The basic idea of the spectral element method is to use a tensor product basis of orthogonal polynomials on fairly large curvilinear quadrilateral subdomains mapped to the square $[-1,1]^2$. The biggest difference between this approach and standard finite elements is that with spectral elements, the elements are much larger and the number of elements is smaller, allowing the use of high order basis functions (polynomials of order five to fifteen) on each element. Further, the stiffness matrix quadratures are done "on the fly" during residual calculations as part of the solution procedure. This can be quite efficient, through the use of the tensor product representation for polynomials.

The spectral element method provides the same exponential convergence to the solution as spectral or pseudo-spectral discretizations. However, since the grid in the spectral element method can be formed by joining together any number of mapped quadrilateral subdomains, this approach provides far greater geometric flexibility than use of a single spectral domain. With the spectral method, only smooth mappings of the entire domain are admissible without sacrificing the convergence rate. The spectral element method represents a liberation from this restriction, providing great geometric flexibility without sacrificing spectral convergence to the true solution.

However, while the spectral element method provides much greater geometric flexibility than spectral methods, the restriction to quadrilateral spectral elements does limit geometric flexibility. Quadrilateral domains are awkward, especially in regions of high curvature, which are typically also regions requiring fine resolution. Further, irregular geometries such as rough walls of irregular shape are not easily gridded by quadrilateral elements. Triangulation, used in unstructured finite volume and finite element methods, (see, for example [4]), represents a far easier method to create grids in such cases.

In addition to convenience and geometric flexibility, triangular elements have a fundamental advantage over quadrilateral elements. It is well known that the conditioning of quadrilateral finite elements degenerates as the angles at their vertices approach 0 or 180 degrees. However with triangles, in the limit as the smallest angle goes to zero one retains a convergent discretization[5]. This is not the case for quadrilaterals using the standard tensor product polynomial bases. While quadrilaterals give consistent discretizations if the min-

TRIANGULAR SPECTRAL ELEMENTS FOR INCOMPRESSIBLE FLUID FLOW

C. Mavriplis*
The George Washington University
Washington, DC 20052

John Van Rosendale*
ICASE, NASA Langley Research Center
Hampton, VA 23681

ABSTRACT

In this paper we discuss the use of triangular elements in the spectral element method for direct simulation of incompressible flow. Triangles provide much greater geometric flexibility than quadrilateral elements and are better conditioned and more accurate when small angles arise. We employ a family of tensor product algorithms for triangles, allowing triangular elements to be handled with comparable arithmetic complexity to quadrilateral elements. The triangular discretizations are applied and validated on the Poisson equation. These discretizations are then applied to the incompressible Navier-Stokes equations and a laminar channel flow solution is given. These new triangular spectral elements can be combined with standard quadrilateral elements, yielding a general and flexible high order method for complex geometries in two dimensions. The natural generalization to tetrahedral elements in three dimensions will be described in a future work.

*This research was partially supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the authors were in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001, and by the National Science Foundation under grant No. ECS-9209347 awarded to the first author.