

Triangular and Tetrahedral Spectral Elements

S.J. Sherwin*

G.E. Karniadakis*

Abstract

In this paper we present a new unstructured spectral element method for use on triangular and tetrahedral subdomains. The algorithm maintains the accuracy and efficiency of standard quadrilateral/hexahedral spectral element methods but offers greater adaptivity. Standard unstructured meshes can be used and the order of the polynomial expansion can be varied within each triangular/tetrahedral sub-domain. To determine the time step restrictions when explicitly solving a convectively dominated flow we numerically analyse the linear advection equation. The formulation is then applied to the incompressible Navier-Stokes equations in the new spectral element code $\mathcal{N}\epsilon\kappa\mathcal{T}\alpha r$.

Key words: unstructured, spectral, adaptive.

AMS subject classifications: 65N30, 65N35, 76D05.

1 Introduction

The current generation of spectral elements uses quadrilaterals and hexahedral elemental subdomains for discretization. As the complexity of the geometries we want to discretise increases the quadrilateral and hexahedral mesh design becomes notably more difficult. This issue has been partly addressed by using a non-conforming, *mortar*, discretisation [1, 2]. This technique has been implemented on quadrilateral domains but until recently had not been implemented on hexahedra due to its complexity. The mortar method also allows the polynomial order to be varied in each elemental domain by using a projection operator at the interface. Another non-conforming mesh approach is the modal *hp* finite element discretisation on quadrilateral domains [3, 4]. In this approach the polynomial expansion can also be varied but unlike the mortar method

the expansion is conforming over elemental edges. The nature of the modal discretisation also allows the polynomial order to be varied in each elemental domain in a more implicit fashion. The trend of both the spectral element and *hp* finite element methods has been towards greater adaptivity. Both these methods have adopted non-conforming discretisations allowing greater mesh adaptivity and they both permit variable order polynomial order discretisations. This requirement for greater adaptivity is motivated by recognition that for many practical problems the solution has a localised structure. An alternative approach to achieve this adaptivity is to use an unstructured discretisation.

In this paper we present an unstructured spectral elements basis which can be applied to triangular and tetrahedral domains. The construction was motivated by theoretical work by Dubiner [5] in two-dimensions, and can be considered as a triangular/tetrahedral extension of the *hp* finite element modal approach. An important property of this new expansion is that it maintains a tensor product type form allowing the use of sum factorisation to maintain efficiency. The algorithms have an operation count of $O(L^{D+1})$ where L is the expansion order and D is the space dimension. As is demonstrated by a numerical investigation of the linear advection equation the new basis has an explicit time step restriction which scales as $O(L^2)$ which is similar to the standard spectral and spectral element method. Both these characteristics have allowed an efficient implementation of the incompressible Navier-Stokes equations in the new unstructured spectral element code $\mathcal{N}\epsilon\kappa\mathcal{T}\alpha r$.

The paper is organized as follows: In the next section we present the tetrahedral expansion basis which is reduced to the two-dimensional basis as a degenerate case. In section 3 we present a review of the basic operations, in matrix form, applied to the new basis. In section 4 we elaborate on the linear advection equation, and in section 5 we present numerical results.

*Center for Fluid Mechanics, Division of Applied Mathematics, Brown University, Providence, R.I. 02912, Corresponding Author

2 Tetrahedral expansion basis

We wish to formulate a tetrahedral expansion basis for the solution of the Navier-Stokes equations extending the two-dimensional expansions proposed by Dubiner [5]. A detailed description of this formulation in two-dimensions is given in [6] and the three-dimensional basis formulation can be found in [7]. Here we outline the basic formulation by introducing the co-ordinate system and describe various properties of the tetrahedral expansion.

2.1 Co-ordinate system

To introduce the co-ordinate system we must first consider a basic mapping as illustrated in figure 1. Here we see the mapping of a rectangular domain in the (Φ, Ψ) space to a triangle in the (ϕ, ψ) space. Although the expansion is not associated with any specific set of nodes, the co-ordinate mapping shown in figure 1 forms a more convenient set of co-ordinates from the computational viewpoint.

We define the standard triangle and rectangle as:

$$T^2 \equiv \{(\phi, \psi) | -1 \leq \phi, \psi; \phi + \psi \leq 0\}$$

$$R^2 \equiv \{(\Phi, \Psi) | -1 \leq \Phi, \Psi \leq 1\}.$$

The rectangular domain R^2 can be mapped into the triangular domain T^2 by the following transformation:

$$\psi = \Psi$$

$$\phi = \frac{(1 + \Phi)(1 - \Psi)}{2} - 1,$$

and similarly the triangular domain T^2 can be mapped into the rectangular domain R^2 by the inverse transformation:

$$\Psi = \psi$$

$$\Phi = 2 \frac{(1 + \phi)}{(1 - \psi)} - 1. \tag{1}$$

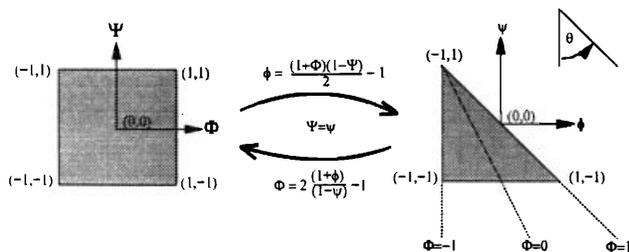


Figure 1: Rectangle to triangle transformation.

As indicated in figure 1, within T^2 the co-ordinate Φ has a value of -1 along the line $\phi = -1$ and a value of 1 along

the line $\phi + \psi = 0$ except at the point $(\phi = -1, \psi = 1)$ where Φ is multi-valued. We know that Φ is bounded in R^2 and the same is true in T^2 . It might appear strange to use a co-ordinate system which has a singular point but it should be noted that the singularity in the co-ordinates does not imply that the expansion is singular. We also recall that both cylindrical and spherical co-ordinate systems have multi-valued co-ordinates at the origin.

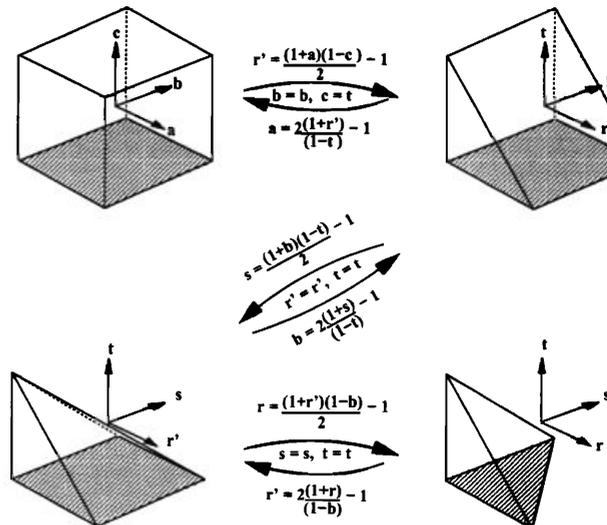


Figure 2: Transformation from hexahedral co-ordinate system to tetrahedral co-ordinate system.

This mapping is the foundation for constructing a co-ordinate system in the tetrahedral domain T^3 starting from a co-ordinate system in the hexahedral domain R^3 . To achieve this, we repeatedly apply the two-dimensional transformation $(\Phi, \Psi) \leftrightarrow (\phi, \psi)$ in three steps. Schematically this transformation is shown in figure 2: In the first step we map R^3 into a triangular prism. Here the Cartesian coordinates (a, b, c) define the domain $R^3 = \{(a, b, c) | -1 \leq a, b, c \leq 1\}$. In the second step, we map the prism into a square based pyramid and finally, we map the pyramid into the tetrahedron, T^3 . The local coordinates (r, s, t) define the space $T^3 = \{-1 \leq r, s, t; r + s + t \leq -1\}$. The three-dimensional basis can be expressed via the initial set of coordinates (a, b, c) as we demonstrate in section 2.2. In summary, we can write the hexahedral co-ordinates (a, b, c) in terms of the tetrahedral co-ordinates (r, s, t) by repeatedly applying the inverse transformation to arrive at:

$$a = 2 \frac{(1 + r)}{(-s - t)} - 1, \quad b = 2 \frac{(1 + s)}{(1 - t)} - 1, \quad c = t.$$

For $t = -1$ we recover the two-dimensional mapping.

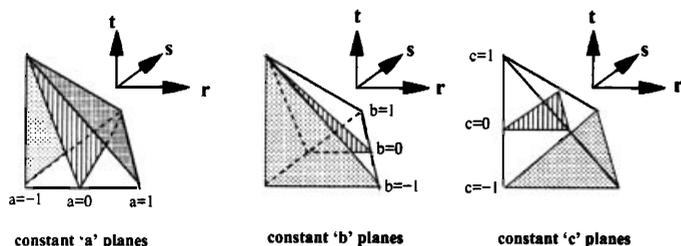


Figure 3: Constant planes of the co-ordinates a, b and c on the standard tetrahedron.

The constant planes represented by (a, b, c) in the tetrahedral space T^3 space are shown in figure 3. We note the degeneracy of the coordinate system in the T^3 space. Planes of constant 'a' remain planes as the coordinate varies from $a = -1$ to $a = 1$ and are dependent on all of the basic coordinates r, s and t . However, planes of constant 'b' degenerate to a line as this coordinate varies from $b = -1$ to $b = 1$ although these planes only depend on the basic coordinates s and t . Finally, planes of constant 'c' degenerate to a point as this coordinate varies from $c = -1$ to $c = 1$ and these planes only depend on the basic coordinate t .

2.2 Expansion basis

We define a polynomial basis, denoted by $g_{lmn}(r, s, t)$, so that we can approximate the function $f(x, y, z)$ by a C^0 continuous expansion over 'K' tetrahedra by the form:

$$f(x, y, z) = \sum_k \sum_l \sum_m \sum_n \hat{f}_{lmn}^k g_{lmn}^k(r, s, t).$$

Here \hat{f}_{lmn}^k is the expansion coefficient corresponding to the expansion polynomial g_{lmn}^k in the k^{th} tetrahedron; (x, y, z) are the global spatial co-ordinates and (r, s, t) are the local co-ordinates within any given tetrahedron.

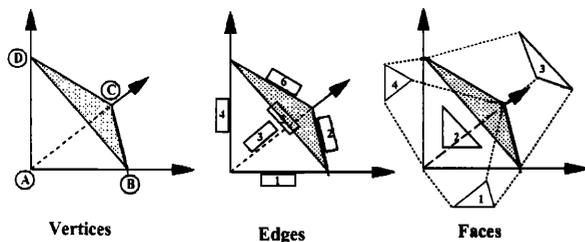


Figure 4: Tetrahedron notation

Having defined the co-ordinates (a, b, c) in the previous section we can now describe the expansion basis. Using the notation given in figure 4 the basis is described by:

• Vertex modes

$$\begin{aligned} g_{100}^{vert-A} &= \left(\frac{1-a}{2}\right)\left(\frac{1-b}{2}\right)\left(\frac{1-c}{2}\right) \\ g_{100}^{vert-B} &= \left(\frac{1+a}{2}\right)\left(\frac{1-b}{2}\right)\left(\frac{1-c}{2}\right) \\ g_{010}^{vert-C} &= \left(\frac{1+b}{2}\right)\left(\frac{1-c}{2}\right) \\ g_{001}^{vert-D} &= \left(\frac{1+c}{2}\right), \end{aligned}$$

• Edge modes ($2 \leq l; 1 \leq m, n \mid l < L; l + m < M; l + m + n < N$)

$$\begin{aligned} g_{l00}^{edge-1} &= \left(\frac{1+a}{2}\right)\left(\frac{1-a}{2}\right)P_{l-2}^{1,1}(a)\left(\frac{1-b}{2}\right)^l\left(\frac{1-c}{2}\right)^l \\ g_{1m0}^{edge-2} &= \left(\frac{1+a}{2}\right)\left(\frac{1+b}{2}\right)\left(\frac{1-b}{2}\right)P_{m-1}^{1,1}(b)\left(\frac{1-c}{2}\right)^{m+1} \\ g_{1m0}^{edge-3} &= \left(\frac{1-a}{2}\right)\left(\frac{1+b}{2}\right)\left(\frac{1-b}{2}\right)P_{m-1}^{1,1}(b)\left(\frac{1-c}{2}\right)^{m+1} \\ g_{10n}^{edge-4} &= \left(\frac{1-a}{2}\right)\left(\frac{1-b}{2}\right)\left(\frac{1+c}{2}\right)\left(\frac{1-c}{2}\right)P_{n-1}^{1,1}(c) \\ g_{10n}^{edge-5} &= \left(\frac{1+a}{2}\right)\left(\frac{1-b}{2}\right)\left(\frac{1+c}{2}\right)\left(\frac{1-c}{2}\right)P_{n-1}^{1,1}(c) \\ g_{01n}^{edge-6} &= \left(\frac{1+b}{2}\right)\left(\frac{1+c}{2}\right)\left(\frac{1-c}{2}\right)P_{n-1}^{1,1}(c). \end{aligned}$$

• Face modes ($2 \leq l; 1 \leq m, n \mid l < L; l + m < M; l + m + n < N$)

$$\begin{aligned} g_{lm0}^{face-1} &= \left(\frac{1+a}{2}\right)\left(\frac{1-a}{2}\right)P_{l-2}^{1,1}(a) \\ &\quad \left(\frac{1+b}{2}\right)\left(\frac{1-b}{2}\right)^l P_{m-1}^{2l-1,1}(b)\left(\frac{1-c}{2}\right)^{l+m} \\ g_{l0n}^{face-2} &= \left(\frac{1+a}{2}\right)\left(\frac{1-a}{2}\right)P_{l-2}^{1,1}(a) \\ &\quad \left(\frac{1-b}{2}\right)^l \left(\frac{1+c}{2}\right)\left(\frac{1-c}{2}\right)^l P_{n-1}^{2l-1,1}(c) \\ g_{1mn}^{face-3} &= \left(\frac{1+a}{2}\right)\left(\frac{1+b}{2}\right)\left(\frac{1-b}{2}\right)P_{m-1}^{1,1}(b) \\ &\quad \left(\frac{1+c}{2}\right)\left(\frac{1-c}{2}\right)^{m+1} P_{n-1}^{2m+1,1}(c) \\ g_{1mn}^{face-4} &= \left(\frac{1-a}{2}\right)\left(\frac{1+b}{2}\right)\left(\frac{1-b}{2}\right)P_{m-1}^{1,1}(b) \\ &\quad \left(\frac{1+c}{2}\right)\left(\frac{1-c}{2}\right)^{m+1} P_{n-1}^{2m+1,1}(c). \end{aligned}$$

• Interior modes ($2 \leq l; 1 \leq m, n \mid l < L; l + m < M; l + m + n < N$)

$$\begin{aligned} g_{lmn}^{interior} &= \left(\frac{1+a}{2}\right)\left(\frac{1-a}{2}\right)P_{l-2}^{1,1}(a)\left(\frac{1+b}{2}\right)\left(\frac{1-b}{2}\right)^l P_{m-1}^{2l-1,1}(b) \\ &\quad \left(\frac{1+c}{2}\right)\left(\frac{1-c}{2}\right)^{l+m} P_{n-1}^{2l+2m-1,1}(c). \end{aligned}$$

where the indices 'lmn' in g_{lmn} refer to the order of the principal polynomial in r, s and t respectively, and L, M, N define the total number of modes. We note that when $l = m + 1 = n + 1$ the edge and face modes have the same shape which allows the basis to be combined into a C^0 expansion by matching the expansion coefficients of these modes. However, expansion coefficients of odd order modes may need to have their sign changed to ensure this continuity.

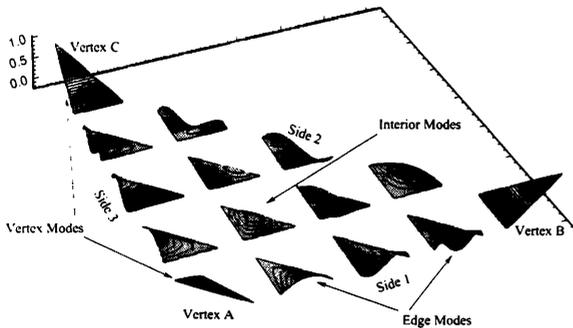


Figure 5: Isocontours for a distribution of two-dimensional modes at an expansion order of $L = M = 5$. Each vertex mode has a unit value at one vertex and varies linearly to zero at the other vertices. The edge modes have a non-zero magnitude along one edge and are zero along all other edges and vertices. The interior modes (which become face modes in the three-dimensional expansion) are zero along all boundaries.

The plane ' $c = -1$ ' defines the two-dimensional expansion (i.e. face 1). The shape of all the two-dimensional modes when $L = M = 5$ can be seen in figure 5. This figure illustrates the general form of the boundary and interior modes. The interior face modes are zero along the edges and vertices. Edge modes only have magnitude along one edge and are zero on all the other edges. Finally, the vertex modes are identical to the linear finite element modes having a unit magnitude at one vertex and decaying to zero at the other vertices. The same structure of boundary and interior modes exists in the three dimensional expansion shown above except that the boundary modes now include vertices, edges and faces. We note that every mode is a polynomial in (a, b, c) as well as (r, s, t) .

To enforce C^0 continuity we require that the number of modes along any edge or with any face are the same between two connecting elements. This condition can be satisfied by setting $L = M = N$ and for an L order expansion there are $L(L+1)(L+2)/6$ modes in three-dimensions. The expansion space is complete in a polynomial space \mathcal{P}_L defined by:

$$\mathcal{P}_L = \text{Span}\{r^l s^m t^n\}_{(lmn) \in p}$$

where

$$p = \{(l, m, n) | 0 \leq l, m, n; l + m + n < L = M = N\}.$$

The order of the two- and three-dimensional expansions can be varied within an element. This is possible since we only require that the number of modes at connecting faces and edges are the same. However, this requirement does not restrict us from allowing the expansion to change at every face and edge or even in the interior of the expansion. For example, it is possible to have a quadratic type expansion on one face and a quartic expansion on another. Therefore a conforming variable order expansion can be constructed in a very natural fashion.

3 Basic operations

Some operations required in the following algorithms are integration and differentiation. In this section we describe these operations as well as introduce some notation that will be adopted in the next section.

3.1 Integration

Since we choose to expand the solution as a polynomial, our primary concern is how to accurately integrate polynomials within T^3 . Recalling that the expansion is a polynomial in both the (r, s, t) space as well as the (a, b, c) space we can exactly evaluate these integrals by using Gaussian quadrature. It is difficult to use one-dimensional Gauss quadrature in the (r, s, t) space since the integration bounds are not constant but this is not the case in the (a, b, c) space where the integration of $u(r, s, t)$ has the form:

$$\int_{T^3} u(r, s, t) dr ds dt = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 u(a, b, c) \frac{\partial(r, s, t)}{\partial(a, b, c)} da db dc.$$

Here $\frac{\partial(r, s, t)}{\partial(a, b, c)}$ is the Jacobian and can be expressed in terms of (a, b, c) as:

$$\frac{\partial(r, s, t)}{\partial(a, b, c)} = \left(\frac{1-b}{2}\right) \left(\frac{1-c}{2}\right)^2.$$

The general form of the Gaussian integration of a polynomial function $f(z)$ with respect to an integration factor $(1-z)^\alpha(1+z)^\beta$ can be written as

$$\int_{-1}^1 (1-z)^\alpha (1+z)^\beta f(z) dz = \sum_{i=0}^{N-1} f(z_i^{\alpha, \beta}) w_i^{\alpha, \beta},$$

where $z_i^{\alpha, \beta}$ and $w_i^{\alpha, \beta}$ are the zeros and weights, respectively, related to the Jacobi polynomial $P_N^{\alpha, \beta}(z)$ [8, 9]. The most standard form of Gauss quadrature is Gauss-Legendre where $\alpha = 0, \beta = 0$. However, we would like

to include the Jacobian, $(\frac{1-b}{2})(\frac{1-c}{2})^2$, in the quadrature weights. This is advantageous since it avoids explicit evaluation of this Jacobian and minimises the order of the quadrature required for exact evaluation of the integral. Therefore we choose to use a quadrature rule with $\alpha = 0, \beta = 0$ in the 'a' direction, a quadrature rule with $\alpha = 1, \beta = 0$ in the 'b' direction and a quadrature rule with $\alpha = 2, \beta = 0$ in the 'c' direction. The integration rule therefore becomes:

$$\int_{-1}^1 \left\{ \int_{-1}^1 \int_{-1}^1 u(a, b, c) \left(\frac{1-b}{2}\right) da db \right\} \left(\frac{1-c}{2}\right)^2 dc = \sum_{k=0}^{q_c-1} \left\{ \sum_{j=0}^{q_b-1} \sum_{i=0}^{q_a-1} f(a_i^{0,0}, b_j^{1,0}, c_k^{2,0}) w_i^{0,0} \hat{w}_j^{1,0} \right\} \hat{w}_k^{2,0},$$

where

$$\hat{w}_j^{1,0} = \frac{w_j^{1,0}}{2}, \quad \hat{w}_j^{2,0} = \frac{w_j^{2,0}}{4}.$$

Here q_a, q_b, q_c are the number of quadrature points in the a, b and c directions respectively. The two-dimensional rule is shown in curly brackets.

Unlike standard quadrilateral spectral elements we are free to choose any type of quadrature rule (i.e. Gauss, Gauss Lobatto or Gauss Radau) with the same cost. Gauss-Lobatto is convenient as it has zeros at the ends of the integration domain allowing the boundary conditions to be easily evaluated and imposed. Nevertheless, this means that we have multiple quadrature points at vertex C and D as well as along edge 6 (see figure 4). This is undesirable because of the redundancy of quadrature points and the fact that it is more difficult to evaluate the derivative at these points. To circumvent this problem we choose to use Gauss-Radau quadrature in the 'b' and 'c' directions.

The quadrature points for the standard triangle T^2 and tetrahedron T^3 are shown in figure 6. The Gauss-Radau quadrature we are using includes a zero at $s = -1$ and therefore we do not have any quadrature points at vertices C and D or along edge 6. We choose to use Gauss-Lobatto integration in the 'a' directions and thus have quadrature points along the boundaries of T^2 and T^3 except at vertices C,D and along edge 6.

3.2 Local projection and notation

In order to define the forward and backward transformations and thereby describe the projection operator we wish to introduce some notation to help simplify the description. To this end, we consider the collocation evaluation

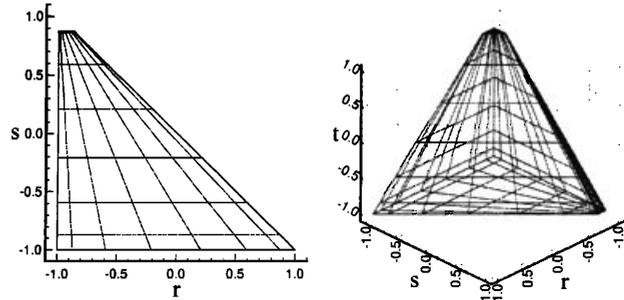


Figure 6: Quadrature points in the standard triangle and tetrahedron space. In the 'a' direction a Gauss-Lobatto distribution is used and in the 'b' and 'c' directions a Gauss-Radau distribution is used.

of a function $f(a, b, c)$ at the quadrature points a_i, b_j, c_k . Note that we have dropped the superscripts which denote the form of the quadrature as explained in the previous section. We shall use the notation \underline{f} to represent a vector of the function evaluation $f(a_i, b_j, c_k)$ at the quadrature points where we will assume that the i index runs fastest followed by the j index and then k . Similarly we shall let $\hat{\underline{f}}$ denote a vector of expansion coefficients \hat{f}_{lmn} . Here we shall use the convention that the vertices are initially stored followed by the edges, then the faces and finally the interior. In each group we assume that the index n runs fastest followed by m and then l . This convention is necessary for the sparsity of the expansion to be evident.

To complement the vectors \underline{f} and $\hat{\underline{f}}$ we introduce the matrices W and G . W is a diagonal matrix containing the quadrature weights required to integrate \underline{f} over T^3 . G is a matrix whose columns are the discrete values of the expansion modes at the quadrature points. These matrices have the form:

$$W = \begin{bmatrix} w_0^{0,0} \hat{w}_0^{1,0} \hat{w}_0^{2,0} & 0 & 0 \\ 0 & \vdots & 0 \\ 0 & 0 & w_{q_a-1}^{0,0} \hat{w}_{q_b-1}^{1,0} \hat{w}_{q_c-1}^{2,0} \end{bmatrix},$$

$$G = \begin{bmatrix} g_{100}(a_0, b_0, c_0) & \cdots & g_{lmn}(a_0, b_0, c_0) \\ \vdots & \vdots & \vdots \\ g_{100}(a_{q_a-1}, b_0, c_0) & \cdots & g_{lmn}(a_{q_a-1}, b_0, c_0) \\ \vdots & \vdots & \vdots \\ g_{100}(a_{q_a-1}, b_{q_b-1}, c_{q_c-1}) & \cdots & g_{lmn}(a_{q_a-1}, b_{q_b-1}, c_{q_c-1}) \end{bmatrix}.$$

Given $\underline{f}, \hat{\underline{f}}, W$ and G we can now define the discrete for-

ward and backward transformations. The discrete backward transformation is very simple as given a set of expansion coefficients \hat{u}_{lmn} the function at the quadrature points is evaluated by the summation:

$$(2) \quad u(a_i, b_j, c_k) = \sum_{lmn} \hat{u}_{lmn} g_{lmn}(a_i, b_j, c_k).$$

This summation can be written in matrix notation as:

$$\underline{u} = G\hat{\underline{u}}.$$

To define the forward transformation we first introduce the discrete inner product in matrix notation (recalling that W is symmetric):

$$(u(a, b, c), v(a, b, c))_{T^3} = (W\underline{u})^T \underline{v} = \underline{u}^T W\underline{v}.$$

This operation is exact if the functions $u(a, b, c), v(a, b, c)$ lie within the polynomial expansion space and the quadrature order is consistent with the approximation. Similarly, the discrete inner product of the expansion basis g_{lmn} with a function $u(a, b, c)$ can be written:

$$(g_{lmn}, u(a, b, c))_{T^3} = G^T W\underline{u}.$$

To determine the expansion coefficients and thus define the forward transformation we take the inner product of equation (2) with respect to the expansion basis. This gives us the matrix equation

$$G^T W\underline{u} = G^T W G \hat{\underline{u}}.$$

Since the matrix $G^T W G$ is square and invertible we define the forward transformation by:

$$\hat{\underline{u}} = (G^T W G)^{-1} G^T W\underline{u}.$$

The discrete projection operator, P , is the projection of a given function into the expansion space and is simply a forward and backward transformation so

$$Pu(a, b, c) = G\hat{\underline{u}} = G(G^T W G)^{-1} G^T W\underline{u}.$$

The projection operator can be interpreted as a collocation operator, which is used to evaluate the function at the quadrature points, followed by a Galerkin projection.

The matrix notation describes the above operations at the quadrature points; however it should be appreciated that the approximation of the function is a continuous polynomial and not a discrete representation at the quadrature points. Nevertheless, we frequently only require the values of our approximation at the quadrature points and therefore find it convenient to consider most operations in this matrix formulation.

3.3 Differentiation

Differentiation may be performed in either the transformed or physical space. However, when dealing with terms like the quadratic convection operator that appears in the Navier-Stokes equation it is efficient to perform the derivative in physical space and then evaluate the convection operator in a collocation manner. Differentiation in physical space is possible since the space defined by Lagrange polynomials through the quadrature points contains the expansion space $\mathcal{P}_{\mathcal{L}}$. A function in $\mathcal{P}_{\mathcal{L}}$ can therefore be represented as:

$$u(a, b, c) = \sum_{ijk} u(a_i, b_j, c_k) h_i^{qa-1}(a) h_j^{qb-1}(b) h_k^{qc-1}(c)$$

where h_i^N is the N^{th} order Lagrange polynomial which has a unit value at the i^{th} quadrature point and is zero at the other quadrature points. Due to the Kronecker Delta property of the Lagrange polynomial (i.e $h_i(a_j) = \delta_{ij}$) differentiation at the quadrature points is very efficient, taking $O(q)$ operations to evaluate the derivative one point where q is the one-dimensional quadrature order. The local gradient operator in terms of (r, s, t) can then be recovered by use of the chain rule since

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{pmatrix} = \begin{pmatrix} \frac{4}{(1-b)(1-c)} \frac{\partial}{\partial a} \\ \frac{2(1+a)}{(1-b)(1-c)} \frac{\partial}{\partial a} + \frac{2}{(1-c)} \frac{\partial}{\partial b} \\ \frac{2(1+a)}{(1-b)(1-c)} \frac{\partial}{\partial a} + \frac{(1+b)}{(1-c)} \frac{\partial}{\partial b} + \frac{\partial}{\partial c} \end{pmatrix}$$

where

$$a = 2 \frac{(1+r)}{(-s-t)} - 1 \quad b = 2 \frac{(1+s)}{(1-t)} - 1.$$

There is potentially a problem when $b = 1$ or $c = 1$ since the factors, $\frac{1}{(1-b)(1-c)}, \frac{1}{(1-b)}$, become infinite. This was part of the motivation for using Gauss Radau quadrature in the b and c directions since this means that we do not need to evaluate the derivatives at these points. However, the derivatives are well defined in this region but not in this co-ordinate system. To evaluate the derivative, one can take the derivative of the expansion basis in (r, s, t) and multiply it by the expansion coefficients.

We can represent the partial differentiation with respect to a, b and c at the quadrature points by the matrix operations:

$$\frac{\partial \underline{u}}{\partial a} = D_a \underline{u} \quad \frac{\partial \underline{u}}{\partial b} = D_b \underline{u} \quad \frac{\partial \underline{u}}{\partial c} = D_c \underline{u}.$$

These matrices are fairly sparse and, from an implementation point of view, are most efficiently evaluated as a

series of one-dimensional operations. If we use the notation $A[ijk][pqr]$ to represent the matrix entry $A[i + q_a j + (q_a q_b)k][p + q_a q + (q_a q_b)r]$ then the form of these matrices is given by :

$$\begin{aligned} D_a[ijk][pqr] &= \frac{dh_i(a_p)}{da} h_j(b_q) h_k(c_r) \\ D_b[ijk][pqr] &= h_i(a_p) \frac{dh_j(b_q)}{db} h_k(c_r) \\ D_c[ijk][pqr] &= h_i(a_p) h_j(b_q) \frac{dh_k(c_r)}{dc} \end{aligned}$$

where we recall that the i and p indices run fastest. We can represent the partial differentiation with respect to r, s and t using these matrices as:

$$\begin{aligned} D_r &= \Lambda \left(\frac{4}{(1-b)(1-c)} \right) D_a \\ D_s &= \Lambda \left(\frac{2(1+a)}{(1-b)(1-c)} \right) D_a + \Lambda \left(\frac{2}{(1-c)} \right) D_b \\ D_t &= \Lambda \left(\frac{2(1+a)}{(1-b)(1-c)} \right) D_a + \Lambda \left(\frac{(1+b)}{(1-c)} \right) D_b + D_c. \end{aligned}$$

Here we have adopted the nomenclature that $\Lambda(f(a, b, c))$ is a diagonal matrix whose components are the values of $f(a, b, c)$ evaluated at the quadrature points.

4 Linear advection equation

The three-dimensional linear advection equation for the scalar quantity, $u(x, y, z; t)$, can be written:

$$(3) \quad \frac{\partial u}{\partial t} + Lu = \frac{\partial u}{\partial t} + (\mathbf{V} \cdot \nabla)u = 0$$

$$\mathbf{V} = [a, b, c]^T; \quad u(x, y, z; 0) = u_0(x, y, z).$$

The initial condition $u_0(x, y, z)$ is considered smooth. The propagation velocity \mathbf{V} is real and will typically be taken as constant and divergence free (i.e. $\nabla \cdot \mathbf{V} = 0$). The equation is also supplemented with appropriate boundary conditions.

We will discretise this equation in a Galerkin fashion by considering the weak form of equation (3). Therefore, if we take the inner product with respect to the function $v(x, y, z)$ we obtain:

$$(4) \quad \left(v, \frac{\partial v}{\partial t} \right)_\Omega + (v, Lu)_\Omega = 0$$

4.1 Discretisation

We consider the solution in an finite domain Ω which is fixed in space and has a boundary $\partial\Omega$. The domain is assumed to be split into K tetrahedral subdomains denoted

by T_k^3 each of which has a local boundary ∂T_k^3 . The union of the K sumdomains T_k^3 is equal to Ω , i.e.

$$\Omega = \sum_k^K T_k^3$$

and the domain boundary, $\partial\Omega$, is a subset of the union of all the local boundaries, ∂T_k^3

$$\partial\Omega \subset \bigcup_k^K \partial T_k^3.$$

If we initially consider a single element and let $v(x, y, z) = g_{lmn}(x, y, z)$ then equation (4) becomes:

$$\left(g_{lmn}, \frac{\partial u}{\partial t} \right) + (g_{lmn}, Lu) = 0 \quad \forall(lmn).$$

Since the solution is approximated as a polynomial function we can represent the solution at the quadrature points by \underline{u} and write the previous equation in matrix form (see section 3.2) as:

$$(5) \quad G^T B \underline{u}_t + G^T B L \underline{u} = 0,$$

where $\underline{u}_t = \frac{\partial \underline{u}}{\partial t}$ and that $B = WJ$ (J being a diagonal matrix containing the value of the Jacobian at the quadrature points). Finally we can represent the solution vector in terms of the expansion coefficients since $\underline{u} = G \hat{\underline{u}}(t)$. Substituting this into equation (5) and noting that

$$\frac{\partial \underline{u}}{\partial t} = G \frac{d\hat{\underline{u}}}{dt},$$

we obtain the Galerkin approximation to equation (3) within an element in terms of the expansion coefficients, $\hat{\underline{u}}$:

$$G^T B G \frac{d\hat{\underline{u}}}{dt} + G^T B L G \hat{\underline{u}} = 0.$$

If this equation was to be solved on a local element then boundary conditions would need to be imposed. This would involve condensing the system of any Dirichlet boundaries. However at present, we are interested in constructing the global system for multiple elements. We can represent the local approximation to equation (3) over all elements as :

$$(6) \quad (G^T B G) \frac{d\hat{\underline{u}}_l}{dt} + (G^T B L G) \hat{\underline{u}}_l = 0,$$

where $\hat{\underline{u}}_l$ denotes a vector containing all the expansion coefficient from every element $\hat{\underline{u}}^k$. The underlined matrix denotes a block diagonal system made from the K local

matrices. In order to define the global expansion as a union of C^0 continuous expansions we need to construct the global matrix systems. The operation of expressing the local non-unique expansion coefficients, \hat{u}_l , in terms of the global unique expansion coefficients, \hat{u}_g , can be represented by the mapping matrix Z (see [6, 10]) i.e.,

$$\hat{u}_l = Z\hat{u}_g.$$

Substituting this relation into equation (6) and premultiplying the whole equation by Z^T we obtain the global matrix system:

$$Z^T(G^T BG)Z \frac{d\hat{u}_g}{dt} + Z^T(G^T BLG)Z\hat{u}_g = 0.$$

If we invert the global mass matrix $(Z^T(G^T BG)Z)^{-1}$ we obtain the semi-discrete system for the solution of the global expansion coefficients:

$$(7) \quad \frac{d\hat{u}_g}{dt} = - (Z^T(G^T BG)Z)^{-1} Z^T(G^T BLG)Z\hat{u}_g.$$

The complete discretisation involves the solution of this system of ordinary differential equations in time. The temporal discretisation we have adopted is the Adams Bashforth multi-step scheme.

The system may be solved with a range of appropriate boundary conditions. Dirichlet boundaries are imposed by performing a local boundary transformation (see [10]) on any modes which have a non-zero component along the Dirichlet boundary and then *condensing* these modes out of the global mass matrix $Z^T(G^T BG)Z$.

To complete the discretisation we need to describe the form of the operator $L \equiv (\mathbf{V} \cdot \nabla)$. The differential component of the operator acts on the expansion basis which is in the polynomial space, \mathcal{P}_L . Therefore, it can be evaluated exactly by differentiating the Lagrange polynomials through the quadrature points as explained in section 3.3. The operator in full is defined:

$$L \equiv a(x, y, z) \frac{\partial}{\partial x} + b(x, y, z) \frac{\partial}{\partial y} + c(x, y, z) \frac{\partial}{\partial z}.$$

If we express the partial derivatives in x, y and z in terms of partial derivatives in r, s and t using the chain rule we obtain:

$$L = \left(a \frac{\partial r}{\partial x} + b \frac{\partial r}{\partial y} + c \frac{\partial r}{\partial z} \right) \frac{\partial}{\partial r} + \left(a \frac{\partial s}{\partial x} + b \frac{\partial s}{\partial y} + c \frac{\partial s}{\partial z} \right) \frac{\partial}{\partial s} + \left(a \frac{\partial t}{\partial x} + b \frac{\partial t}{\partial y} + c \frac{\partial t}{\partial z} \right) \frac{\partial}{\partial t}.$$

Therefore, the discrete version of the L operator acting at the quadrature points can be written as:

$$L = RD_r + SD_s + TD_t$$

where

$$\begin{aligned} R &= \Lambda \left(a \frac{\partial r}{\partial x} + b \frac{\partial r}{\partial y} + c \frac{\partial r}{\partial z} \right) \\ S &= \Lambda \left(a \frac{\partial s}{\partial x} + b \frac{\partial s}{\partial y} + c \frac{\partial s}{\partial z} \right) \\ T &= \Lambda \left(a \frac{\partial t}{\partial x} + b \frac{\partial t}{\partial y} + c \frac{\partial t}{\partial z} \right) \end{aligned}$$

and D_r, D_s, D_t as well as $\Lambda()$ were as in section 3.3. For straight-sided tetrahedral elements the geometric factors $\frac{\partial r}{\partial x}, \frac{\partial s}{\partial x}, \frac{\partial t}{\partial x}, \dots$ are constants and so if $a(x, y, z), b(x, y, z)$ and $c(x, y, z)$ are also constants, then R, S and T are simply scalars.

4.2 Spectrum of the weak advection operator

The main purpose of this investigation is determining what time step restrictions are imposed by explicit treatment of the advection terms. Considering the semi-discrete form of the weak advection equation as given in equation (7) we note that for a time stepping scheme to be stable we require the eigenvalues of $\Delta t \cdot (Z^T(G^T BG)Z)^{-1} Z^T(G^T BLG)Z$ lie within stability region of the time stepping scheme. Here Δt is the time step and clearly this must decay at the same rate that the largest eigenvalue grows in order for the spectrum to remain within the stability region.

The matrix $G^T BLG$ represent the discrete form of the inner product of the expansion basis with the advection operator acting on the expansion basis i.e. (g_{lmn}, Lg_{pqr}) . Now since we have assumed that the propagation velocity \mathbf{V} is divergence-free, we can write:

$$Lg_{lmn} = (\mathbf{V} \cdot \nabla)g_{lmn} = \nabla \cdot (\mathbf{V}g_{lmn}).$$

Since g_{lmn}, g_{pqr} are both scalar fields we can apply the vector identity $(\phi \nabla \cdot \mathbf{V} = \nabla \cdot (\phi \mathbf{V}) - \nabla \phi \cdot \mathbf{V})$ to show:

$$(g_{lmn}, Lg_{pqr}) = -(Lg_{lmn}, g_{pqr}) + \int_{T^3} \nabla \cdot (g_{lmn} \mathbf{V}g_{pqr}) \delta V.$$

Applying the divergence theorem to the last term we arrive at:

$$(g_{lmn}, Lg_{pqr}) = -(Lg_{lmn}, g_{pqr}) + \int_{\partial T^3} g_{lmn} g_{pqr} \mathbf{V} \cdot \mathbf{n} \delta A$$

where \mathbf{n} is the outward normal along the boundaries of a tetrahedron. This last result demonstrates that the operator is skew symmetric if the surface integral is zero. As there is no approximation error in evaluating the L operator, the matrix representation of the discrete operator will also be skew symmetric. The surface integral is zero if either of the expansion modes g_{lmn} or g_{pqr} is an interior

mode since interior modes are zero along the boundaries by design. We are also imposing C^0 continuity and assuming a conforming discretisation, so the surface integral along interfaces between two elements will be equal and opposite. When the system is globally assembled there will be zero net contribution from elemental interfaces. The only remaining contribution from the surface integral is therefore along the boundaries of the solution domain. If the boundary conditions are skew symmetric (i.e. periodic or zero Dirichlet) then the global operator will be skew symmetric.

Since the global mass matrix $(Z^T(G^T BG)Z)$ is symmetric when it premultiplies the weak advection operator $Z^T(G^T BLG)Z$ (with skew symmetric boundary conditions) the matrix $(Z^T(G^T BG)Z)^{-1} Z^T(G^T BLG)Z$ will be skew symmetric. Therefore, the eigenvalues of $(Z^T(G^T BG)Z)^{-1} Z^T(G^T BLG)Z$ must be purely imaginary. This means that we require a time stepping scheme with a stability region encompassing the imaginary axis. The third order Adams Bashforth scheme has a stability region which crossed the imaginary axis at:

$$\Delta t \cdot \lambda_{max} \simeq 0.723,$$

where λ_{max} is the maximum permissible eigenvalue for the scheme to be stable.

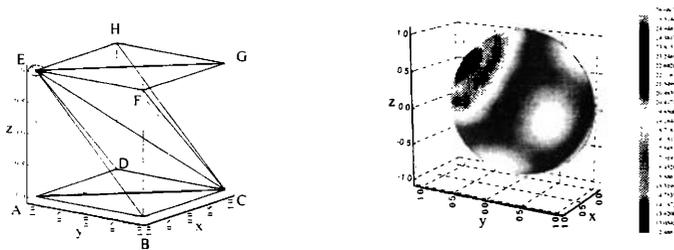


Figure 7: Periodic domain containing six tetrahedral elements as shown on the left where the triangle and circle indicate vertices C and D respectively (see figure 4). For a range of propagation velocities at $L = 10$ the maximum eigenvalue of the matrix $(Z^T(G^T BG)Z)^{-1} Z^T(G^T BLG)Z$ was calculated and is shown on the right. The propagation velocities have a unit magnitude and are parallel to the vector connecting the origin of the right plot to a point on the hemisphere.

In order to determine the maximum eigenvalue of the matrix $(Z^T(G^T BG)Z)^{-1} Z^T(G^T BLG)Z$ we choose a periodic domain formed from six tetrahedra as shown in figure 7. The domain spans the space $\{-1 \leq x, y, z \leq 1\}$ and

as indicated by the circles in the plot, the vertex D of all tetrahedral elements is placed at point $E (1, -1, 1)$. The triangles in this figure indicate the location of vertex A (see figure 4). Since the advection operator L is a function of the propagation velocity \mathbf{V} the matrix $(Z^T(G^T BG)Z)^{-1} Z^T(G^T BLG)Z$ must also be a function of \mathbf{V} . The variation of maximum eigenvalues as a function of the propagation velocity is shown in figure 7. In constructing this plot we have assumed that the propagation velocity has a unit magnitude and is oriented in the direction given by a vector connecting the origin to a point on the surface of the hemisphere. It is only necessary to determine this range of propagation velocities since propagation in the $[1, 1, 1]$ direction forms a matrix which is the negative of the matrix due to propagation in the $[-1, -1, -1]$ direction and so they will have the same maximum eigenvalues. Since we have restricted the propagation velocity to be of unit magnitude it is possible to describe any vector by two spherical angles ϕ and θ as shown in figure 8. In this figure we see the absolute maximum eigenvalues as shown in figure 7 parametrised with the spherical angles ϕ and θ . We have reversed the direction of the θ axis to make the plot consistent with figure 7.

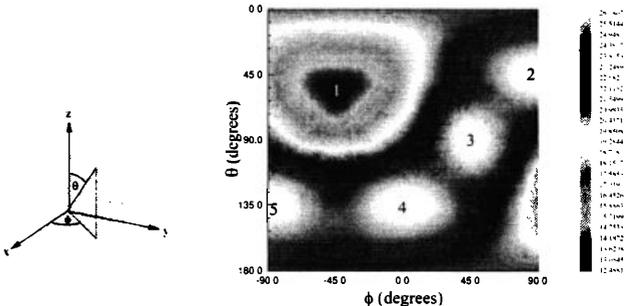


Figure 8: Definition of spherical angles ϕ and θ (left) and plot of absolute maximum eigenvalues for a unit propagation vector (right). This is the same distribution as shown in figure 7.

The values of ϕ and θ at the nearest calculated position to the extrema shown in figure 8 are given in table 1. The position and form of the extrema can be attributed to the structure of the tetrahedral domain shown in figure 7. For example the global minimum (extremum 1) corresponds to a unit propagation velocity in the $[\sqrt{1/3}, -\sqrt{1/3}, \sqrt{1/3}]$ direction which is parallel to the diagonal bisector of the whole domain running from position C to E in figure 7. The triangular structure of this minimum is consistent with the directions in which the tetrahedra bisect the faces of

	Type	ϕ (degrees)	θ (degrees)	Eig. value
1	minimum	-45.3	56.2	12.35
2	maximum	90.0	45.3	26.65
3	maximum	45.3	90.0	26.65
4	maximum	0.0	135.2	26.65
5	maximum	-90.0	135.2	26.65

Table 1: Table of position and type of turning point in absolute maximum eigenvalue distribution shown in figure 8.

the domain. These bisectors run in the directions from B to E , C to F and G to E which correspond to the spherical angles $(\phi = -180, \theta = 45)$, $(\phi = 0, \theta = 45)$ and $(\phi = -45, \theta = 180)$. Figure 8 demonstrates that these angles make a triangle of similar orientation to the region surrounding the minimum extremum at position 1. Let us now consider vectors which bisect the faces in the other directions i.e.; from A to F , G to B and D to B which correspond to the spherical angles $(\phi = 90, \theta = 45)$, $(\phi = 0, \theta = 135)$ and $(\phi = 45, \theta = 90)$. We see from table 1 these angles correspond to the maximum extrema 2, 4 and 3. Maximum 5 corresponds to a propagation velocity of equal and opposite direction to maximum 2. Finally the local minima between the maxima 2-3, 3-4 and 4-5 can be attributed to the three remaining diagonal bisectors of the complete domain i.e. from D to F , H to B and G to A . We see that propagation velocities aligned with the element edges lead to minima in the eigenvalue distribution whilst maxima correspond to propagation in the orthogonal direction to those that lead to the minima. This type of behaviour has also been observed in the two-dimensional expansion [6].

We are now in a position to determine the maximum eigenvalue growth rate as a function of expansion order. The growth rate for three propagation directions is shown in figure 9. The most critical directions are those corresponding to the maximum extrema of the eigenvalues in figure 8 and we see that the growth rate in the $(\phi = 45, \theta = 90)$ direction is asymptotically faster than the other sampled directions but is still bounded by a slope of 2. If we consider the last three points of these curves we find that the slopes are 1.88, 1.79 and 1.82 for the propagation velocities of $(\phi = 45, \theta = 90)[D - B]$, $(\phi = 45, \theta = 56.2)[D - F]$ and $(\phi = -45, \theta = 56.2)[C - E]$ respectively. Therefore, the growth rate is bounded by L^2 .

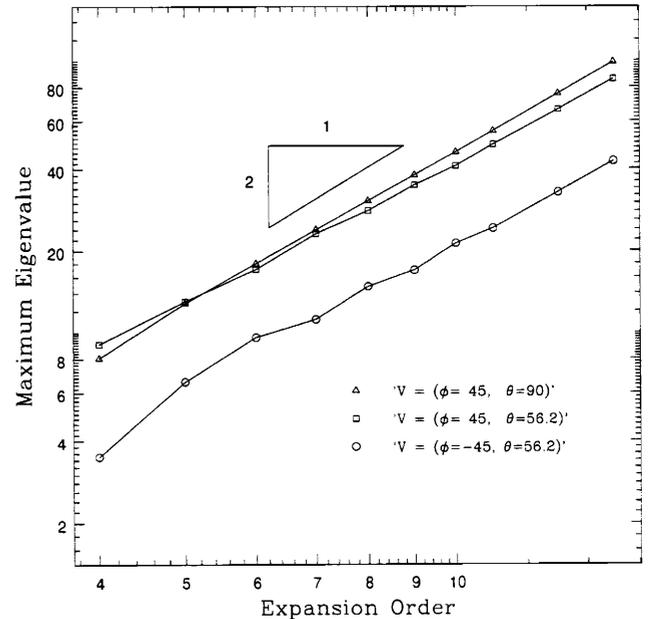


Figure 9: Growth of the maximum eigenvalue of the matrix $(Z^T(\underline{G}^T \underline{B} \underline{G})Z)^{-1} Z^T(\underline{G}^T \underline{B} \underline{L} \underline{G})Z$ as a function of expansion order for propagation velocities of magnitude $|\mathbf{V}| = \sqrt{3}$ with directions of $(\phi = 45, \theta = 90)[D - B]$, $(\phi = 45, \theta = 56.2)[D - F]$ and $(\phi = -45, \theta = 56.2)[C - E]$.

5 Numerical examples

Extensive numerical results that verify flexibility and exponential convergence for the triangular and tetrahedral spectral elements can be found in [6], [7], and [10]. Here we include two examples: One on the elliptic Helmholtz equation, and the second one on a standard Navier-Stokes problem.

The elliptic Helmholtz equation $(\nabla^2 - 1)u = f$ was discretized on the “helix” domain shown in figure 10. This domain is formed by rotating a circle with a triangle coming out of it about the axis through the center of the circle. Also included in the figure is the crosssection that generates the three-dimensional domain by propagation along the axis. This 3D mesh generation algorithm from 2D templates is described in [11]. The forcing is chosen so that the exact solution is

$$u(x, y, z) = \frac{1}{(Y - R)^2 + 1}$$

$$Y = -x \sin\left(\frac{\pi z}{5}\right) + y \cos\left(\frac{\pi z}{5}\right),$$

where $R = 1$ is the radius of the cylinder. The full domain

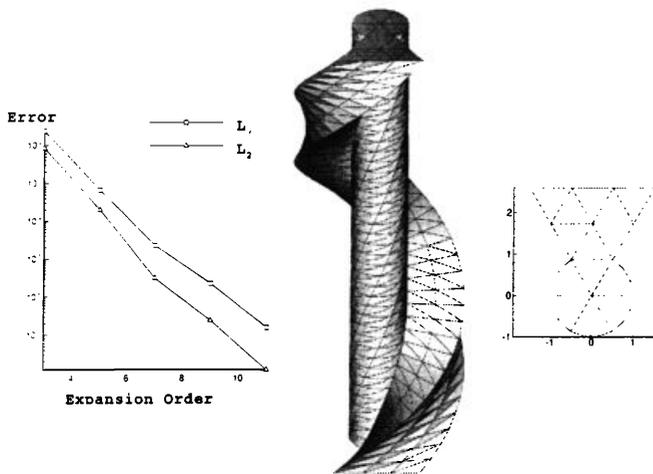


Figure 10: Tetrahedral spectral element mesh for the Helmholtz problem. $K = 1680$ tetrahedra were used in the discretization.

is discretized using $K = 1680$ elements. For the convergence plot (figure 10) only an eighth of the mesh length was considered corresponding to $K = 210$; this was necessary due to accommodate memory requirements on an SGI workstation. The curved surfaces include the cylinder and the two helicoids bounding the triangle. An isoparametric representation is used in the formulation, i.e. the geometry is interpolated with the same accuracy as the solution. Exponential convergence is realized in the p-refinement study shown in figure 10. In other numerical experiments with simpler domains we have seen a faster decay rate. At present, it is not well understood if there are some non-negligible quadrature errors that slow convergence, but current work addresses this point.

The Navier-Stokes results are based on a high-order splitting scheme that treats separately advection, incompressibility constraint, and viscous correction [14], [10]. The flow we consider is in a pipe expansion in a mesh shown in figure 11. Although the solution is azimuthally symmetric, we use a full tetrahedral discretisation in a Cartesian co-ordinate system and so the Navier-Stokes solution is three dimensional. Here we see half the computational domain split down the centerline of the cylinders. The complete domain was discretised using $K = 600$ tetrahedral elements. The smaller pipe has a diameter of $D_0 = 1$ and the larger cylinder has a diameter of $D_1 = 2$. The Hagen-Poiseuille solution with unit magnitude at the centerline is imposed as the inflow and the outflow is treated

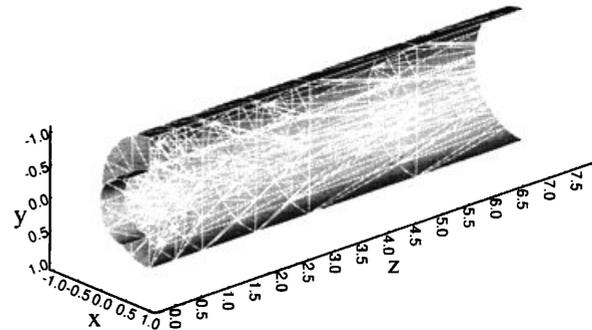


Figure 11: Discretisation of the pipe expansion using 600 tetrahedral elements. Here we see half of the domain. The pipe expands from a diameter of 1 to a diameter of 2 and the complete domain is 7.5 units long of which 0.5 units is the smaller pipe.

as parallel. The Reynolds number here is defined as

$$Re = \frac{W_0 D_0}{\nu},$$

where W_0 is the mean velocity at the inflow and D_0 is the diameter of the smaller pipe. The flow in a pipe expansion was experimentally studied by Macagno & Hung [13]. The re-attachment position at low Reynolds number is of interest since this length is a linear function of Reynolds number as documented by Macagno & Hung: this was computed in our simulations using change of sign in vorticity at the wall. We also determined the location of the center of a “captive eddy” (also visualized in the experiments) as a function of Reynolds number; both results are shown in figure 12. In this plot L is the streamwise distance to the re-attachment point from the shoulder in the conduit and L_2 is the streamwise distance from the shoulder to the center of the eddy. Most of the results were obtained with $L = 7$. The calculation was performed with an expansion order of $L = 9$ for the case $Re = 50$ and no significance difference was observed. Using a quadrilateral spectral element code formulated in cylindrical co-ordinates [12] the calculation was repeated and the results are recorded in figure 12. Also shown in this plot are experimental points from Macagno & Hung’s paper determined from photographic information and visual observation of dyed oil leaked into the flow through a series of small orifices in the cylinder wall. As

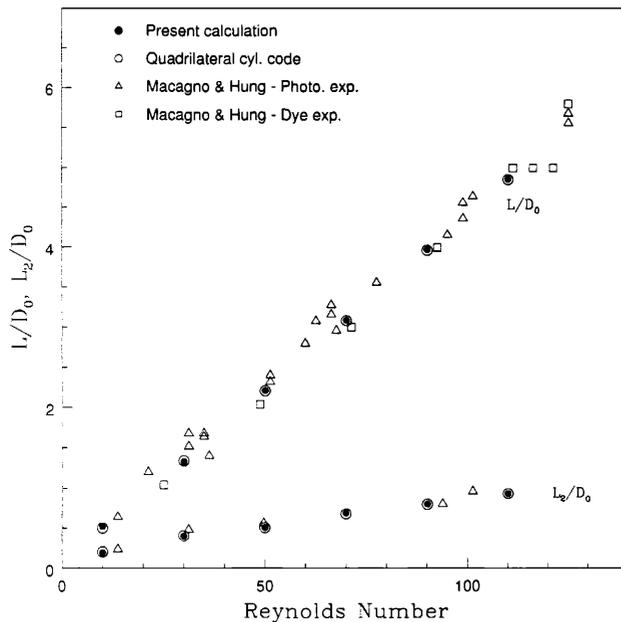


Figure 12: Growth rate as a function of Reynolds number of the streamwise distance to the re-attachment point L and eddy center L_2 from the expansion shoulder. Also shown in this plot is a comparison with a spectral element calculation in cylindrical co-ordinates [12] and experimental data determined by Macagno & Hung [13].

can be seen our calculations are in good agreement with both the experimental data as well as the quadrilateral spectral element code.

6 Acknowledgements

We would like to thank T.C. Warburton of Brown University for his assistance with the Helmholtz problem. This work was sponsored by AFOSR, ONR, and DOE. Computations were performed at the Pittsburgh Supercomputing Center and at NCSA University of Illinois.

References

[1] C. Bernardi, Y. Maday, and A.T. Patera. A new conforming approach to domain decomposition: the mortar element method. In *Nonlinear Partial Differential Equations and Their Applications*, Pitman and Wiley, 1992.

- [2] R.D. Henderson and G.E. Karniadakis. Unstructured spectral element methods for simulation of turbulent flows. *in press, J. Comp. Phys.*, 1995.
- [3] J.T. Oden W.Wu and V.Legat. An hp adaptive strategy for finite element approximation of the navier-stokes equations. In *Finite Elements in Fluids - New trends and applications*, page 32. Pineridge Press, 1993.
- [4] L. Demkowicz, J.T. Oden, W. Rachowicz, and O. Hardy. Toward a universal h-p adaptive finite element strategy, Part 1. constrained approximation and data structure. *Comp. Meth. Appl. Mech. Eng.*, 77:79, March 1989.
- [5] M. Dubiner. Spectral methods on triangles and other domains. *J. Sci. Comp.*, 6:345, 1991.
- [6] S.J. Sherwin and G.E. Karniadakis. A triangular spectral element method; applications to the incompressible Navier-Stokes equations. *in press, Comp. Meth. Appl. Mech. Eng.*, 1995.
- [7] S.J. Sherwin and G.E. Karniadakis. A new triangular and tetrahedral basis for high-order finite element methods. *in press, Int. J. Num. Meth. Eng.*, 1995.
- [8] C. Canuto, M.Y. Husanini, A. Quarteroni, and T.A. Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, 1988.
- [9] A. Ghizzetti and A. Ossicini. *Quadrature Formulae*. Academic Press, 1970.
- [10] S.J. Sherwin and G.E. Karniadakis. Tetrahedral hp finite elements: Algorithms and flow simulations. *in press, J.C.P.*, 1995.
- [11] T.C. Warburton, S.J. Sherwin, and G.E. Karniadakis. Extrusion of 3d meshes from 2d templates for computational mechanics problems. *in preparation*.
- [12] D.J. Newman and G.E. Karniadakis. Navier-Stokes formulation in cylindrical co-ordinates of the prism code. 1994, Brown University, unpublished report.
- [13] E.O. Macagno and T. Hung. Computational and experimental study of a captive annular eddy. *J. Fluid Mech.*, 28:43, 1967.
- [14] G.E. Karniadakis, M. Israeli, and S.A. Orszag. High-order splitting methods for the incompressible Navier-Stokes equations. *J. Comp. Phys.*, 97:414, 1991.