FINITE DIFFERENCE OPERATORS ON UNSTRUCTURED TRIANGULAR MESHES

G. Erlebacher
NASA Langley Research Center
Hampton, Va 23665

The solution to partial differential equations (PDE's) on increasingly complicated two-dimensional geometries has for a long time held the interest of the computational fluid dynamics community. One of the major difficulties encountered is the discretization of the PDE's on the grid underlying the physical domain in such a way that the resulting numerical scheme is stable and has good convergence properties. To this end, much effort is currently being devoted to constructing grids which simplify the discretization process, without sacrificing their ability to cover general domains and without compromizing the convergence and stability of the resulting numerical schemes. These considerations have led to research on zonal gridding[1], global cartesian gridding,[2] and embedded grids[3] just to name a few. These techniques are generally applied to curvilinear grids for which a vast store of numerical knowledge has been accumulated over the last decade.

An alternative to one or more curvilinear meshes is to triangulate the physical domain. This approach has the advantage that any two-dimensional geometrical configuration, however complicated, can be gridded with well-behaved, nearly equilateral triangles[4]. Both structured and unstructured triangular grids can be used. They are distinguished by the fact that whereas an interior node on the former is always common to six edges, there is no such restriction on the latter. Of course, structured triangular grids retain some of the regular features of curvilinear grids.

Once the physical domain is triangulated, one must derive the discrete form of the PDE's, or of the individual operators. Finite-volume techniques are ideally suited to discretize hyperbolic systems of PDE's (notably the Euler equations) conservatively[5]. The method is not limited to triangles or quadrilaterals, which leaves the possibi-

lity open for grids with mixed cell types. Integrated forms of the conservation laws are evaluated over a triangle area or an appropriate cell area surrounding a triangle node (depending on whether the variables are evaluated at triangle centroids or at the nodes). Conservation of energy and mass impose additional constraints on the difference formulas[6].

An alternative to treating integral conservation laws is to discretize each operator of the differential form of the equations separately. This can also be accomplished with the help of finite-volume techniques, or by using a variational principle. For example, the latter method can be used to effectively generate discrete forms of the Poisson operator[4]. One of its chief assets is that only operators of lower order than the original must be discretized, which is clearly a simpler task.

In this work, a new form of the Laplace operator is derived which is shown to be second-order accurate when calculated at triangle nodes if the nodes have the property that their corresponding cells possess a high degree of symmetry. On the other hand, it is shown that the Laplace operator directly derived from Stokes' integral theorem is only zeroth-order accurate on these same cells. The reason the zero order approximation has been successfully used in the past is that the error most probably arises from the discretization of the cell area, which often multiplies the equation that is being solved, thereby cancelling the error.

The other major topic of this paper is a discussion of the construction of gradient and Laplace operators on unstructured triangular grids, mainly from the variational point of view. Two discrete representations of the Laplacian are compared to each other, and conclusions are drawn regarding their accuracy on a pointwise basis. Finally, geometric relations valid on arbitrary polygons are given for completeness in the appendix. These were derived in the course of a truncation error analysis.

TRIANGULAR GRIDS

Two-dimensional triangular grids have been used for a number of years, mostly within the context of finite-element theory[6-8], because the formulas they lead to are simpler and easier to handle than the ones derived on curvilinear quadrilaterals. They allow the treatment of many different types of geometries with equal ease. On the other hand, the corresponding triangular gridded finite-difference tech-

niques are still quite new, and as yet, not well documented. The purpose of this paper is to partially remedy this situation by presenting several methods for deriving difference operators on 2-D triangular grids.

The paper begins with a review of the notation used, followed by a derivation of a series of relationships on rectilinear, non-orthogonal coordinate systems. These operators are then compared with the corresponding finite-difference operators on triangular grids. This comparison provides a basis for estimation of maximum bounds on truncation errors.

Triangles

A triplet of non-colinear points in the plane defines a triangle. The coordinates of the 3 vertices are referenced by their x,y coordinates in a cartesian system x,y,z where the unit vector $\hat{z}$ is normal to the x,y plane. These points (also called nodes, or vertices) are referred to by radius vectors subscripted with lowercase letters i,j,k, numbered counterclockwise as one moves around the triangle's perimeter. The difference between two such vectors, $(\vec{r}_j - \vec{r}_i)$ is denoted by $\vec{r}_{i,j}$. Triangles are indicated by capitalized superscripts. When summing over the vertices of a triangle B, the symbol $\sum_{i(B)}$ is used. Unit vectors are recognized by the presence of a carat. Consistent with this notation, the outward unit normal vector to a triangle edge i,i+1 is

$$\hat{n}_{i,i+1} = \frac{\vec{r}_{i,i+1}}{r_{i,i+1}} \times \hat{z} = \hat{r}_{i,i+1} \times \hat{z}$$

where $r_{i,i+1}$ is the length of $\vec{r}_{i,i+1}$. The normal vector is outward because of the counterclockwise vertex ordering and positive z direction. Occasionally $r_{i+1/2}$ is substituted for $r_{i,i+1}$ when the meaning is clear from the context. Otherwise, the original convention is assumed. Each node is connected to a set of nodes which form a corresponding polygon. Its nodes are numbered in a counterclockwise order (fig. 1). The center node is generally labelled c, and for simplicity, expressions such as $f_i - f_c$ are replaced by F_i. Here, f_i is the value of the analytical function f (typically C^2) to which the operators are applied.

Cells

Fundamental to the construction of discrete operators on unstructured grids is the concept of a cell which satisfies the following three conditions:

1. Each grid point has a cell associated with it.

2. A cell contains one and only one grid point.

3. An arbitrary point in the computational domain is either included in a single cell, or lies on a cell boundary.

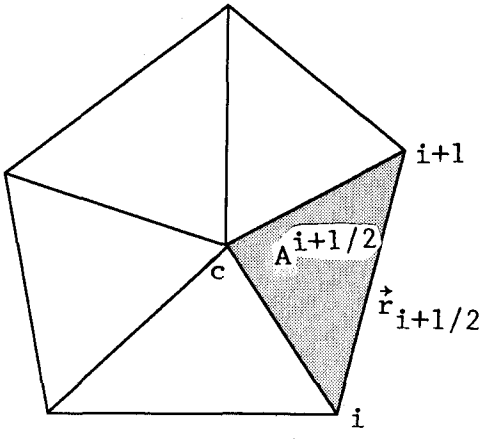


Fig. 1: Polygon surrounding node c

The third condition immediately implies that the domain of intersection between two cells is at most an edge, which guarantees that cells do not overlap. Thus, the area of the computational domain is just the sum of the cell areas. Moreover, since each cell is associated with a unique (interior or boundary) mesh point, it follows that the sum can be indexed by the mesh points or simply by the corresponding mesh point index. The computational domain can also be interpreted as a fluid modeled by a finite number of particles (lattice points) of finite volume (cell areas).

Consider now the construction of a cell from a given triangular lattice. Two approaches are commonly referred to: the Voronoi cell,

and the median based cell. The Voronoi cell (fig. 2) associated with a node c is by definition the collection of points that lie closer to c than to any other node[9]. By virtue of its definition, it is evident that this construction is unique. The major advantage of this type of cell is that its volume varies continuously as the mesh nodes move across the physical domain. Furthermore, they are inherently convex, which guarantees that they always contain their centroid. This can have an impact on certain adaptive techniques[10]. The associated triangular mesh is called the Delaunay mesh. An alternative to the Voronoi cells are the median-based cells which are used throughout this text. The construction of such a cell is made clear in fig. 3. Medians to the triangles that have c as one of their nodes,

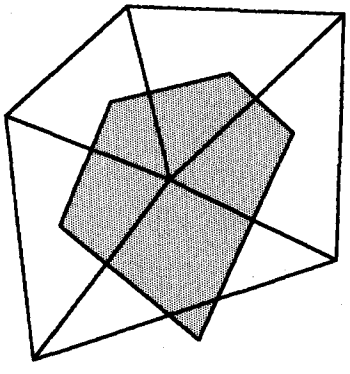


Fig. 2: Voronoi cell

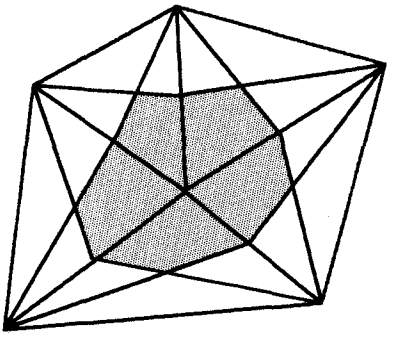


Fig. 3: Median-based cell

define the perimeter of the cell associated with c. It is also easily shown that the cell area is given by

$$A_c = \frac{1}{3} \sum_{B(c)} A^B \qquad (2)$$

The cells just defined interlock like a jigsaw puzzle and can partition every bounded planar region.

Non-Orthogonal Rectilinear Coordinate Systems

A cartesian coordinate system is not always the most convenient system in which to work. For example, in order to estimate truncation errors of operators on triangular meshes, Taylor expansions of their finite-difference representations are required. On a cartesian grid, an expansion in the x,y directions would explicitly involve the various angles in the triangle. However, simpler formulas result if

the expansion is done along two preferred directions. From symmetry considerations, the most natural point about which to expand is the triangle's centroid. The independent variables used are, therefore, chosen to be the distances along 2 of the triangle's 3 medians. In general, consider a positive oriented coordinate system $\hat{u}_1$, $\hat{u}_2$, along two arbitrary directions such that $\hat{u}_1 \cdot \hat{u}_2 = 0$ (fig. 4). It is desirable to derive an analytical expression for the gradient operator in these coordinates. Since the gradient operator is two-dimensional, it can be expressed as a linear combination of the independent unit operators ∂_1, ∂_2, where the partial derivatives are respectively taken along the u_1 and u_2 axes. A simple calculation leads to the formula

$$\nabla = \hat{z} \times (-\hat{u}_1 \partial_2 + \hat{u}_2 \partial_1) \; / \; \sin\theta \qquad (3)$$

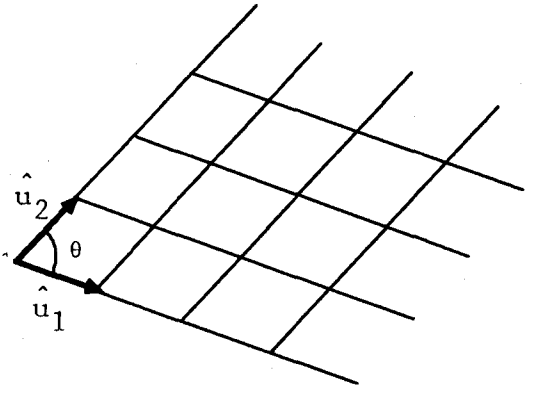


Fig. 4. Non-orthogonal rectilinear basis

Since the angle between coordinate lines is constant over the entire domain, the Laplace operator is simply the scalar product of the gradient operator with itself, which immediately yields the formula

$$\nabla^2 = \frac{1}{\sin^2\theta} (\partial_{11} - 2 \cos\theta \; \partial_{12} + \partial_{22}) \qquad (4)$$

This clearly reduces to the standard form on a cartesian grid in the limit $\theta = \frac{\pi}{2}$.

The next step is to derive an exact formula for the gradient

operator at a cell center c (fig. 5). The expression above is obviously still exact if one works in a basis defined along any two directions, but we wish to take all the neighboring vertices of c into account. To do so, equation (3) is written for all consecutive pairs of nodes (1,2),(2,3),(3,4),...(n-1,n):

$$(\nabla_E)_c^{i+1/2} = \hat{z} \times (-\hat{r}_{c,i}\ \vartheta_{i+1} + \hat{r}_{c,i+1}\ \vartheta_i) \ /\ \sin\theta_{i+1/2} \qquad (5)$$

where $(\nabla_E)_c^{i+1/2}$ is the gradient calculated at c, in the rectilinear

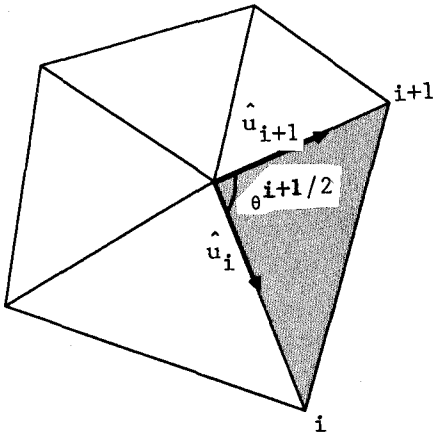


Fig. 5. Non-orthogonal basis superimposed on a triangular mesh

basis $\vec{r}_{c,i}$, $\vec{r}_{c,i+1}$. The gradient operator can be written as a weighted sum of these functions if the weighting coefficients sum up to unity. The coefficients are chosen to be proportional to the triangle areas, which leads to

$$\nabla = \frac{1}{6} A_c\ \hat{z} \times \sum_{i(c)} \vec{r}^{\,i}\ (\vec{r}_{c,i} \cdot \nabla) \qquad (6)$$

where the notation $\vec{r}^{\,i} = \vec{r}_{i-1,i+1}$ has been adopted. The expression above is consistent with the intuitive requirement that triangles with zero area should not contribute to the gradient. Note that other choices for the coefficients are possible. For example, Crowley[11] reports successful computations when the coefficients are proportional

to the product of the triangle area and the angle $\theta_{i+1/2}$.

Numerical operators

When a computational mesh has an irregular structure, it has been found that discrete approximations to a number of differential operators can be obtained from integral relations such as Stokes' and Gauss' theorems. The gradient and divergence operators are respectively determined by

$$\int_{\sigma} \nabla f \, da = \oint_{\partial\sigma} f \, \hat{n} \, dr \tag{7}$$

$$\int_{\sigma} \nabla.\vec{v} \, da = \oint_{\partial\sigma} \vec{v}.\hat{n} \, dl \tag{8}$$

where the notation is clear from fig. 6a.

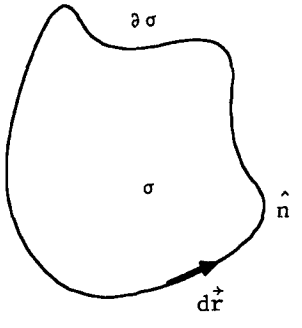


Fig. 6a. Domain σ with C^1 boundary $\partial\sigma$.

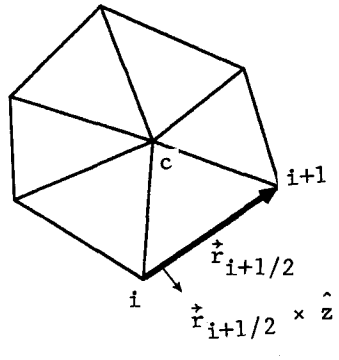


Fig. 6b: Polygonal boundary.

From the mean-value theorem, there exists points p_1, p_2, in σ such that

$$\int_{\sigma} \nabla f \, da = A_{\sigma} \, \nabla f_{p_1} \tag{9}$$

$$\int_{\sigma} \nabla.\vec{v} \, da = A_{\sigma} \, \nabla.\vec{v}_{p_2} \tag{10}$$

where A_{σ} is the area of the domain σ , and ∇f_p is the analytical gradient calculated at the point p. Since these relations are valid on all simply-connected domains, they must also be satisfied on domains with polygonal boundaries. To derive finite-difference

approximations to (9-10), the points p_i are identified with the cell center c, and the integrals on the left-hand side are approximated with a box integration scheme. Equations (9-10) respectively become

$$\nabla f_c = \frac{1}{3A_c} \sum_{i(c)} f_{i,i+1} \hat{n}_{i,i+1} r_{i,i+1} \tag{11}$$

$$\nabla.\vec{v}_c = \frac{1}{3A_c} \sum_{i(c)} \vec{v}_{i,i+1} \cdot \hat{n}_{i,i+1} r_{i,i+1} \tag{12}$$

which can be seen from figure 6b. First order operators can also be defined at the barycenter (not a grid point) of a triangle if $3A_c$ is replaced by the triangle area A^B. A knowledge of a variable U at the barycenters of a cell's triangles enables it to be calculated at the cell's center from the area weighted average

$$U_c = \frac{1}{3A_c} \sum_{B(c)} A^B U^B \tag{13}$$

For example, with $U=\nabla f$, an estimate for ∇f at the cell center is given by

$$3 A_c \nabla f_c = \sum_{B(c)} A^B (\nabla f)^B \tag{14}$$

$$= \sum_{i(c)} f_{i+1/2} \vec{r}_{i+1/2} + f_{i+1} \vec{r}_{i+1,c} + f_{c,i} \vec{r}_{c,i}) \times \hat{z} \tag{15}$$

$$= \sum_{i(c)} f_{i+1/2} \vec{r}_{i+1/2} \times \hat{z} \quad . \tag{16}$$

Expressions (16) and (11) are identical, which justifies the use of (14) as an interpolation formula to derive a discrete representation of the Laplace operator. Other interpolation schemes have also been considered[11] where $A^B/3A_c$ is replaced by $A^B \alpha^B/(6A_c \pi)$. Here, α^B is the angle centered at c within triangle B. These are not pursued further in this paper.

A rather simple form of the gradient is obtained if $\vec{r}^i = \vec{r}_{i-1,i+1}$ is substituted for $\vec{r}_{i-1,i+1}$ and when $f_{i,i+1}$ is expanded in terms of f_i and f_{i+1}. After simplifications, The gradient becomes

$$\nabla f_c = \frac{1}{6A_c} \sum_{i(c)} f_i \vec{r}^i \times \hat{z} \quad . \tag{17}$$

In the same manner, vectors are defined at a triangle's barycenter by the arithmetic average of its value at the 3 triangle vertices, which allows the gradient operator at c to be expressed, not as a function

of the variable evaluated at the vertices, but as a function of its value at the barycenters. Starting from (16), $f_{i+1/2}$ is expanded to obtain

$$3A_c \nabla f_c = -\frac{1}{2} \hat{z} \times \sum_{i(c)} (f_i + f_{i+1}) \vec{r}_{i+1/2} \tag{18}$$

which can be simplified by observing that

$$\sum_{i(c)} \vec{r}_{i+1/2} = 0 \tag{19}$$

around a closed contour. One can therefore add f_c to the expression in parenthesis in (18) to obtain

$$\nabla f_c = -\frac{1}{2A_c} \hat{z} \times \sum_{i(c)} f^{i+1/2} \vec{r}_{i,i+1} \tag{20}$$

where $f^{i+1/2}$ is the value of f at the centroid of triangle B, with vertices i,i+1,c. In terms of the outward normals $\hat{n}_{i+1/2}$, the cell centered gradient reduces to

$$\nabla f_c = \frac{1}{2A_c} \sum_{i(c)} f^{i+1/2} \hat{n}_{i+1/2} r_{i+1/2} \quad . \tag{21}$$

A similar expression can also be derived for the divergence operator by multiplying both sides of equation (20) by 3/2 and replacing $\vec{v}_{i,i+1}$ by

$$\vec{v}^{i+1/2} = \frac{1}{3} (\vec{v}_i + \vec{v}_{i+1} + \vec{v}_c)$$

Equations (11) and (21) are not exactly identical since (13) is only an approximation. Nevertheless, they do agree to first order. These same integral techniques are employed to derive a discrete representation for the Laplace operator:

$$\oint_\sigma \nabla^2 f \, da = \oint_{\partial\sigma} (\nabla f) \hat{n} \, dl \tag{22}$$

where $\hat{n} \, dl = d\vec{l} \times \hat{z}$ and with an assumed knowledge of the gradient operator. The domain of integration of (22) is the (hatched) cell area in fig. 3. The approximate formula

$$A_c \nabla^2 f_c = \frac{1}{2} \sum_{i(c)} (\nabla f)^{i+1/2} (\vec{r}_{i+1/2} \times \hat{z}) \tag{23}$$

is then immediately deduced from the mean value theorem. One then

obtains for the Laplacian,

$$\nabla^2 f_c = -\frac{1}{4A_c A^{i+1/2}} \sum_{i(c)} \vec{r}_{i+1/2} \cdot [\vec{r}_{i+1,c} F_i + \vec{r}_{c,i} F_{i+1}] \tag{24}$$

where $F_i = f_i - f_c$. This formula is well known, but it is found to be only zeroth order accurate on most cell shapes. Nonetheless it is often used to solve elliptic problems when a stationary solution is sought. The error is primarily a result of the approximation of the line integral in (22). In the next section, a more accurate discrete representation of the Laplace operator is derived, based on a variational principle. The difference between the two formulations will be seen to originate from alternate partitionings of the physical domain. The generalization of (24) to the operator $\nabla.(\sigma\nabla)$ is immediate:

$$\nabla.(\sigma\nabla f)_c = -\frac{1}{4A_c A^{i+1/2}} \sum_{i(c)} \sigma^{i+1/2} r_{i,i+1} \cdot (\vec{r}_{i+1,c} F_i + \vec{r}_{c,i} F_{i+1}) \tag{25}$$

where $\sigma^{i+1/2}$ is a spatial function evaluated at the barycenter of triangle i+1/2.

Variational Principles

In the previous section, an expression for the discrete Laplace operator was derived with the help of integral theorems. Unfortunately, as will be shown later, it is only zeroth order accurate on many cell geometries. The objective here is to derive an improved, more accurate version of the discrete Laplace operator. Rather than resort to the methods presented in the previous section, variational methods are chosen for their flexibility and ease of use[12-13]. The method is mesh independent, invariant with respect to a change of variables, and relies on the discretization of an operator of lower order, which is one of its chief assets. Some authors find the variational approach a useful tool to develop finite-difference approximations to systems of equations which can be derived from a minimization principle. Discrete natural boundary conditions can also be deduced from this approach in a natural way. Even though some equations do not possess a minimization principle, it still makes sense to separate out the differential operators that do, and to derive some finite-difference formulas for them.

Formulation

A map I(f) from the space of C^2 functions to the real line is called a functional. Given a set of functions that all satisfy identical boundary conditions, which one(s) make I(f) stationary with respect to arbitrary perturbations of f? To answer this question, an arbitrary perturbation δf is applied to f:

$$f \rightarrow f + \lambda\ \delta f \tag{26}$$

where λ is a scalar parameter. I(f) then becomes $I(f + \lambda\ \delta f)$, and the functions that satisfy

$$\frac{dI(\lambda)}{d\lambda}\ \Big|_{\lambda=0} = 0 \tag{27}$$

make the functional stationary. In what follows, only functionals of the form

$$I(f) = \int_{\sigma} L(f,\ \nabla f,\ \vec{r})\ d\vec{r} \tag{28}$$

integrated over a fixed two-dimensional domain σ are considered. All the concepts described below can be extended to the treatment of moving boundaries, natural boundary conditions, and different types of Lagrangian functions. The objective is to calculate which function(s) among those with identical boundary conditions on $\partial\sigma$ make I(f) stationary with respect to perturbations of f. When written out, the derivative of I(f) with respect to λ becomes

$$\frac{dI(\lambda)}{d\lambda} = \frac{d}{d\lambda} \int_{\sigma} L(f + \lambda\ \delta f,\ \nabla f + \lambda\ \nabla\delta f,\ \vec{r})\ d\vec{r} \quad , \tag{29}$$

and the first variation of I is simply

$$\delta I = \frac{dI(\lambda)}{d\lambda}\ \Big|_{\lambda=0} = \int_{\sigma} \left(\frac{\partial L}{\partial f} - \frac{d}{d\vec{r}}\ \frac{\partial L}{\partial \nabla f}\right)\ \delta f\ d\vec{r} \tag{30}$$

where the boundary terms drop out since δf vanishes on $\partial\sigma$. The functions that make the functional stationary must therefore satisfy the Euler-Lagrange equations. For notational simplicity, let $\Lambda(f)$ denote the Euler-Lagrange operator acting on f. Equation (30) then becomes

$$\delta I = \int_{\sigma} \Lambda(f)\ \delta f\ d\vec{r} \tag{31}$$

The inherent flexibility of this approach lies in the fact that the accuracy of the discrete Euler-Lagrange operator, Λ_h , is the result of two approximations: L_h and I_h. The former is equivalent to the requirement that the calculation be based on a good representation of the lower order differential operator. This will be addressed in this section. On the other hand, the latter is directly affected by the integration scheme employed to approximate the functional. Of course, more accurate schemes will lead to more complicated formulas for Λ_h . In the following, we confine ourselves to a box integration scheme. It is interesting to note that while I(f) may be approximated on a mesh $M(f_i,A_i)$, its first variation can be evaluated on a second mesh $M'(f_j,A_j)$, after L_h has been appropriately interpolated onto it. All the formulas remain invariant. This property is often used to express differential operators in various coordinate systems. Examples of mesh switching are given in the next section when the formalism is applied to triangular grids. The discrete representation of (31) is obtained as follows. First, the functional I is made discrete:

$$I_h(f) = \sum_{i(c)} L_{h_i}(f)\, A_i \quad . \tag{32}$$

Next, f is perturbed according to (29) and δI_h is calculated on the mesh M:

$$\delta I_h = \sum_{i(c)} \frac{\partial L_{h_i}(f+\lambda\delta f)}{\partial \lambda}\Big|_{\lambda=0} A_i \tag{33}$$

$$= \sum_{j(c)} \left(\sum_{i(c)} \frac{\partial L_{h_i}}{\partial f_j} \frac{A_i}{A_j} \right) \delta f_j\, A_j \tag{34}$$

where the derivatives are evaluated on the mesh M'. Comparing equations (31) and (34), $\Lambda_h(f)$: becomes

$$\Lambda_{h_j}(f) = \sum_{i(c)} \frac{\partial L_{h_i}}{\partial f_j} \frac{A_i}{A_j} \tag{35}$$

at each nodal point j on the mesh M'. When these results are specialized to the functional

$$I(f) = -\frac{1}{2} \int_\sigma (\nabla f)^2 \, d\vec{r} \quad , \tag{36}$$

it is easy to verify that its first variation is

$$\delta I = \int_{\sigma} \nabla^2 f \; \delta f \; d\vec{r} \tag{37}$$

if $\delta f=0$ on $\partial\sigma$. Equation (37) direcly leads, after the identification $\Lambda_{h_j} = \nabla_h^2 f_j$, to the discrete approximation

$$\nabla_h^2 f_i = - \frac{1}{2A_i} \sum_{j(i)} \frac{\partial (\nabla_h f)_i^2}{\partial f_j} A_j \tag{38}$$

where the sum extends over all nodes j which partake in the definition of $\nabla_h f_i$. As expected, the discrete representation of the Laplacian is solely dependent on a knowledge of the discrete gradient. By merely switching from one discrete gradient formulation to another, on various sub-meshes, a variational approach immediately results in a variety of discrete approximations. Two versions of the discrete Laplacian on arbitrary triangular meshes are now presented.

Triangular Grids

A discrete representation of the Laplacian of a scalar f is now derived from two variational principles that differ from each other in the definition of the mesh M over which the gradient is approximated. The grid points of M' at which the Lapacian is evaluated are the same in both cases; it is defined by triangle vertices. The domain is then partitioned into median-based cells of area A_c.

Version 1

A typical cell from the mesh M is illustrated as a hatched region in fig. 1. The nodes on the mesh M, defined by the triangle centroids, lead to a functional $I_h(f)$ of the form

$$I_h(f) = - \frac{1}{2} \sum_{B(\sigma)} (\nabla_h f^B)^2 \; A^B \tag{39}$$

which, after substitution of the discrete gradient operator (18) becomes

$$I_h(f) = - \frac{1}{2} \sum_{B(\sigma)} \frac{1}{A^B} \sum_{\substack{i(B) \\ j(B)}} (f_{i+1/2} \; f_{j+1/2} \; \vec{r}_{i+1/2}) . \vec{r}_{j+1/2} \tag{40}$$

on the mesh M'. Rearranging the terms, and noting that

$$f_{i,i+1}=(f_i+f_{i+1})/2 \quad ,$$

it follows that

$$I_h(f) = -\frac{1}{8} \sum_{B(\sigma)} \frac{1}{A^B} f_i \, f_j \, \vec{r}^{\,i} \cdot \vec{r}^{\,j} \tag{41}$$

which is a quadratic expression in the $f_i f_k$. The coefficients of δf_c in the first variation

$$\delta I_h(f) = \sum_{c(\sigma)} \frac{\partial I_h}{\partial f_c} \delta f_c \tag{42}$$

are identified with the right-hand side of equation (35) after substitution of $\nabla_h^2 f_j$ for Λ_{h_j} , which leads to

$$\nabla_h^2 f_c = \frac{1}{A_c} \frac{\partial I_h}{\partial f_c} \tag{43}$$

for the approximate Laplace operator. The derivative of I_h with respect to f_c must now be evaluated. From (41)

$$\frac{\partial I_h}{\partial f_c} = \frac{1}{4} \sum_{B(\sigma)} \frac{\vec{r}_{c-1,c+1}}{A^B} \cdot \sum_{j(B)} f_j \, \vec{r}_{j-1,j+1} \tag{44}$$

where all superscripts refer to the triangle B with vertices c,i,i+1 (the last two lie on the polygon's boundary) and area $A^{i+1/2}$ (fig. 1). Expand the sum over the triangle's vertices in (44) yields

$$\nabla_h^2 f_c = -\frac{1}{4A_c} \sum_{B(c)} \frac{\vec{r}_{i+1/2}}{A_{i+1/2}} \cdot (f_c \vec{r}_{i+1/2} + f_i \, \vec{r}_{i+1,c} + f_{i+1} \vec{r}_{c,i}) \tag{45}$$

which is identical to the formulas derived from integral theorems in the previous section. Henceforth, this form of the Laplacian is referred to as FL.

In a later section, it will be shown that this representation of the Laplace operator is zeroth order accurate on most cell shapes except for those that tile planar domains in a regular fashion. Usually a scalar coefficient multiplies a first order accurate expression. This is probably due to the approximation of the cell area in the integration scheme. Previous authors, when solving time-independent elliptic problems involving the Laplace operator, missed this error, assuming that the expression was first order accurate.

Moreover, Fritts[7] has indicated that on average, the discrete algorithm is slightly less than second order accurate. A second formula, second-order accurate on well-behaved cells, is now derived.

Version 2

The mesh M' is unchanged (triangle vertices), whereas the mesh M is now the union of the vertices of M' and the midpoints of all its edges as illustrated in fig. 7, which depicts a cell of mesh M as a hatched area identified with node k. Let k be an arbitrary vertex from M with a cell area of A_{k^M} where the suffix M is present as a reminder of the mesh currently in use. The discrete functional is

$$I_h(f) = -\frac{1}{2} \sum_{k(\sigma)} (\nabla f_k)^2 \, A_{k^M} \tag{46}$$

Expanding the gradients, the functional becomes

$$I_h(f) = -\frac{1}{2} \sum_{k(\sigma)} \frac{1}{9A_{k^M}} \sum_{\substack{i(k) \\ j(k)}} f_{i,i+1} \, f_{j,j+1} \, \vec{r}_{i+1/2} \cdot \vec{r}_{j+1/2} \tag{47}$$

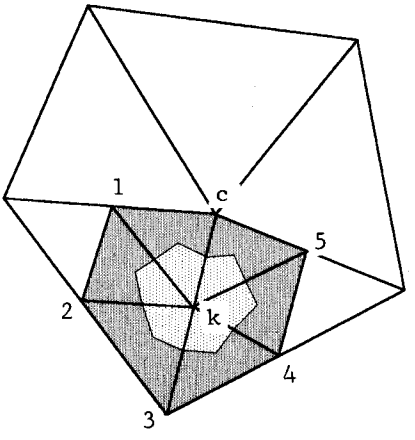


Fig. 7: Mesh M (node k) and mesh M' (node c)

Using the following relation true at all the interior nodes k,

$$\sum_{i(k)} f_{i,i+1} \vec{r}_{i,i+1} = \frac{1}{2} \sum_{i(k)} f_i \, \vec{r}^{\,i} \tag{48}$$

to express (47) as a quadratic function of $f_i f_j$, results in the simplified expression

$$I_h(f) = -\frac{1}{72} \sum_k \frac{1}{A_{k^M}} \sum_{i,j} f_i f_j \vec{r}^i \cdot \vec{r}^j \qquad (49)$$

formally identical to (41). Equation (49) is now differentiated with respect to f_c to obtain

$$\frac{\partial I_h}{\partial f_c} = -\frac{1}{36} \sum_{k(c)} \frac{1}{A_{k^M}} \vec{r}^c \cdot \sum_{\substack{i(k) \\ j(k)}} f_i \vec{r}^i f_j \vec{r}^j \qquad (50)$$

where $\vec{r}^c$ is defined in figure 8. Note that the sum only includes the polygons k connected to node c, for these are the only ones which give a non-zero contribution to (50). All that remains to be done now is to interpolate all the variables in (50) onto the grid M'. For

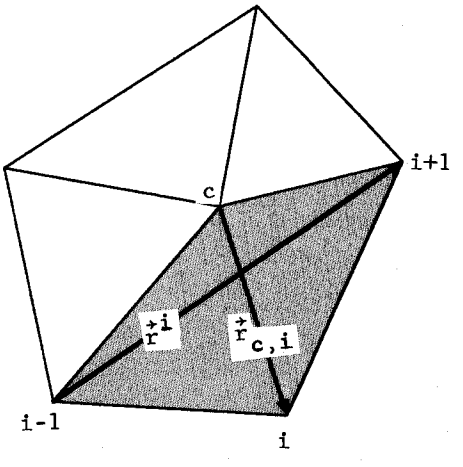


Fig. 8: Notation for equation (50)

maximum clarity, the sum over i is expanded, noting that there are always six terms as can be seen from figure 7:

$$f_i \vec{r}^i = f^c \vec{r}_{5,1} + f_1 \vec{r}_{c,2} + f_2 \vec{r}_{1,3} + f_3 \vec{r}_{2,4} + f_4 \vec{r}_{3,5} + f_5 \vec{r}_{4,1} \quad .$$

Every term in (51) on the fine mesh M is now interpolated onto the coarser mesh M' by assuming that the value of f at an edge midpoint is the average of its value at the extremities. Equation (51) becomes after some straighforward algebra

$$f_i \vec{r}^{\,i} = \frac{3}{4} [(f_i - f_c)\, \vec{r}^{\,i} - (f_{i+1} - f_{i-1})\, \vec{r}_{c,i}] \quad . \tag{52}$$

The area of the quadrilateral c,i-1,i,i+1 is denoted by A^i, (fig. 8) and A_{k^M} is expressed as a function of A^i:

$$A_{k^M} = \frac{A^i}{4} \quad . \tag{53}$$

Substitution of (52) and (53) into (50) leads to

$$\frac{\partial I_h}{\partial f_c} = \frac{1}{24} \sum_{i(c)} \frac{\vec{r}^{\,i}}{A^i} \cdot [(f_i - f_c)\, \vec{r}^{\,i} - (f_{i+1} - f_{i-1})\, \vec{r}_{c,i}] \tag{54}$$

where $\vec{r}^{\,c}$ has been replaced by $-\frac{\vec{r}^{\,i}}{2}$.

Finally equation (54), inserted into (43) for the Laplacian gives the desired result

$$\nabla_h^2 f_c = \frac{1}{6A_c} \sum_{i(c)} \frac{\vec{r}^{\,i}}{A^i} \cdot [(f_i - f_c)\, \vec{r}^{\,i} - (f_{i+1} - f_{i-1})\, \vec{r}_{c,i}] \tag{55}$$

This form of the Laplacian is henceforth referred to as GL. In a later section, the accuracy of GL and FL will be compared to one another on various cell shapes, and it will be concluded that GL is often more accurate than FL on a pointwise basis.

The generalization of (55) to approximate $\nabla.(\sigma\nabla f)$ is straightforward:

$$\nabla.(\sigma\nabla f) = \frac{1}{6A_c} \sum_{i(c)} \frac{\sigma^i\, \vec{r}}{A^i} \cdot [(f_i - f_c)\, \vec{r}_i - (f_{i+1} - f_{i-1})\, \vec{r}_{c,i}] \tag{56}$$

where $\sigma^i = \frac{\sigma_i + \sigma_c}{2}$.

Formulas for the second partial derivatives with respect to x and y may also be derived from appropriate functionals[4].

This section treated interior points only. If natural boundary conditions must be calculated, the procedure is identical. The function f is perturbed by an increment $\lambda\delta f$ among the set of all functions differentiable to second order. The first variation of the functional is calculated, this time including the boundary terms. It is then made discrete. At the same time, the functional itself (same as previously) is made discrete, and differentiated with respect to the nodal values of the function. The finite-difference formulas

corresponding to the natural boundary conditions are then determined from equation (35).

Cell Geometry

In this section, the explicit expressions for GL and FL are compared to one another, first at triangle centroids, and then on more general cell shapes. It is demonstrated that the the best linear approximation to the Laplace operator estimated at the centroid of arbitrary triangles is obtained when they are equilateral, and that even then, only GL provides first order accuracy. A discrete representation of an operator L is considered n^{th} order accurate if

$$L_h(f) = L_E(f) + O(h^n) \tag{57}$$

where h is a typical scale length, for example the average length of a triangle edge.

Arbitrary Triangular cell

From basic geometry, it is well known that all the $A^{i+1/2}$ in an arbitrary triangle B with centroid c are equal to each other. This simplifies the formulas considerably. It is easy to apply formulas (45) and (55) to this cell (which respectively correspond to FL and GL). They reduce to the single formula

$$\nabla_h^2 f_c = \alpha_i \frac{\left(\sum_i r_{c,i}^{+2}\right)}{4 A_c^2} \sum_j F_j \tag{58}$$

where $A_c = A^B/3$ is the cell area. The scale factors that correspond to GL and FL are denoted by α_i , i=FL,GL. They are given by $\alpha_{GL} = \frac{1}{3}$ and $\alpha_{FL} = 1$. This difference in values immediately implies that FL and GL cannot simultaneously be first order accurate on any given triangle.

Under what conditions (if any) does there exist a linear finite-difference formula in the f_i of higher accuracy than zero on a triangle? The most general linear expression for $\nabla_h^2 f_c$ as a function of f at the triangle barycenter c and nodes i is

$$\nabla_h^2 f_c = \beta_c f_c + \sum_{i=1}^{3} \beta_i f_i \tag{59}$$

where β_i, β_c are 4 undetermined constants. A Taylor expansion of f_i about f_c at node c up to second order leads to

$$f_i = f_c + \vec{r}_{c,i} \cdot \nabla f_c + \frac{1}{2} \vec{r}_{c,i}\vec{r}_{c,i} : \nabla\nabla f + O(r^2_{c,i}) \qquad (60)$$

Equation (59) is required to be exact for constant functions. Furthermore, (59) and (60) together imply that

$$\beta_c = - \sum_i \beta_i \quad . \qquad (61)$$

Substituting (59) into (60), and choosing a function f linear in x,y, all the terms of second and higher order derivatives drop out and there remains

$$\sum_i \beta_c \vec{r}_{c,i} \cdot \nabla f = 0 \qquad (62)$$

which must be satisfied for all linear functions f. Expressing $\vec{r}_{c,3}$ as the linear combination

$$\vec{r}_{c,3} = -(\vec{r}_{c,1} + \vec{r}_{c,2}) \qquad (63)$$

and substituting this into (62) implies that

$$(\beta_1 - \beta_3)\, \partial_1 f + (\beta_2 - \beta_3)\, \partial_2 f = 0 \qquad (64)$$

where the partials are taken along the corresponding vectors $\vec{r}_{c,i}$. The derivatives are acting along two independent directions; therefore their coefficients must vanish. As a consequence,

$$\beta_1 = \beta_2 = \beta_3 = \beta \quad . \qquad (65)$$

With this simplification (59) reduces to

$$\nabla^2_h f_c = \beta \sum_i F_i \qquad (66)$$

which is structurally identical to (57). The β_i have now been chosen to cancel out the zeroth and first order terms in the Taylor expansion of (59) for all possible functions. This leaves the second order term, plus a third order remainder:

$$\nabla_h^2 f_c = \frac{\beta}{2} \left(\sum_{i(c)} r_{c,i}^2 \, \partial_{ii} f_c \right) + \beta \, O(r_{c,i}^3) \quad . \tag{67}$$

After expressing ∂_{33} as a function of ∂_{11}, ∂_{12}, and ∂_{22} and substituting the result in the expression for the discrete Laplace operator, (66) becomes

$$\nabla_h^2 f_c = \beta \left(r_{c,1}^2 \partial_{11} + r_{c,2}^2 \partial_{22} + \vec{r}_{c,1} \cdot \vec{r}_{c,2} \, \partial_{1,2} \right) f_c + \beta \, O(r_{c,i}^3) \tag{68}$$

which approximates the exact Lapacian to first order at node c if $r_{c,i} = r_{c,2}$, $-2 \cos\theta = 1$, and $\beta = 4/(3h^2)$ where $h = \vec{r}_{c,i}$ (see (4)). The first two conditions imply that the triangle must be equilateral. Finally, the discrete representation of the Laplacian is found by substituting β back into (67),

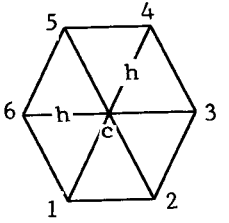
$$\nabla_h^2 f_c = \frac{4}{3h^2} \sum_i F_i \quad . \tag{69}$$

Therefore, FL gives a zeroth order accuracy, whereas GL is first order accurate, on equilateral triangles. This is easily seen after expressing h as a function of A_c. On polygons with more than 4 edges, a sufficient number of degrees of freedom (>6) are available to theoretically insure the existence of a first order accurate discrete Laplace operator. In the next section, it is shown that GL is zeroth order accurate on asymmetrical cells, and second order on highly symmetric ones. It is conjectured that when solving a time-dependent equation that involves the Laplace operator, that these zero order errors average out and that the actual error is closer to second order. This is borne out by work done by Fritts[6]. Finally, note that if the triangle is not equilateral, (14) and (68) cannot be identified with each other, and zeroth order accuracy is the best that can be expected from any linear formula.

More general shapes

Table I lists several cell shapes, along with the corresponding discrete formulas derived from GL and FL. The discrete formulas are equated to the exact Laplacian operator acting upon the function f with the addition of an error term of order n. Several facts emerge from a close inspection of table I. First of all, there are cells (i.e. cell 7) for which both GL and FL are simultaneously zeroth order accurate. Accordingly, they do not produce the correct values when applied to a function f as the cell size decreases to zero. Second,

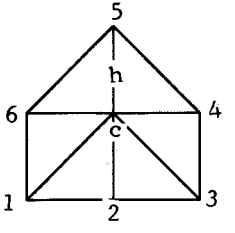
Cell Geometries with discrete GL and FL



Cell 1

$$\nabla_h^2 f_c^{FL} = \nabla_h^2 f_c^{GL} = \frac{2}{3h^2} \sum_{i(c)} F_i = \nabla_E^2 f_c + O(h^2)$$

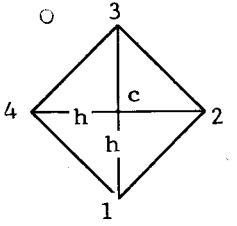
Cell 2



$$\nabla_h^2 f_c^{FL} = \frac{1}{h^2} (F_2 + F_4 + F_5 + F_6) = \nabla_E^2 f_c + O(h^2)$$

$$\nabla_h^2 f_c^{GL} = \frac{5}{6h^2} [F_4 + F_6 + F_1 + F_3 + 4\,F_2 + 6\,F_5] = \nabla_E^2 f_c + O(h)$$

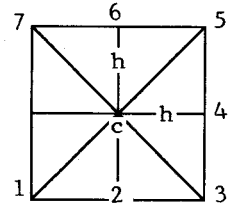
Table I



Cell 3

$$\nabla_h^2 f_c^{FL} = \frac{3}{4h^2} \sum_{i(c)} F_i = \nabla_E^2 f_c + O(1)$$

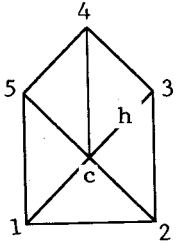
$$\nabla_h^2 f_c^{GL} = \frac{1}{h^2} \sum_{i(c)} F_i = \nabla_E^2 f_c + O(h^2)$$



Cell 4

$$\nabla_h^2 f_c^{FL} = \frac{3}{4h^2} \sum_{\text{even } i} F_i = \nabla_E^2 f_c + O(1)$$

$$\nabla_h^2 f_c^{GL} = \frac{1}{4h^2} \left(2 \sum_{\text{odd } i} + \sum_{\text{even } i}\right) F_i = \nabla_E^2 f_c + O(h^2)$$



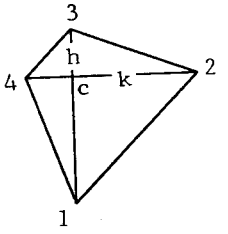
Cell 5

$$\nabla_E^2 f_c^{FL} = \frac{6}{5h^2} \sum_{i(c)} F_i = \nabla_E^2 f_c + O(1)$$

$$\nabla_h^2 f_c^{GL} = \frac{1}{h^2} \sum_{i(c)} F_i = \nabla_E^2 f_c + O(h^2)$$

Table I (cont.)

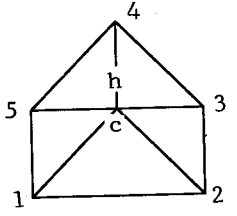
Cell 6



$$\nabla_h^2 f_c^{FL} = \frac{6}{h+k}\left[\frac{F_1 + F_2}{k} + \frac{F_3 + F_4}{h}\right] = \nabla_E^2 f_c + O(1)$$

$$\nabla_h^2 f_c^{GL} = \frac{2}{h+k}\left[\frac{F_1 + F_2}{k} + \frac{F_3 + F_4}{h}\right] = \nabla_E^2 f_c + O(h)$$

Cell 7



$$\nabla_h^2 f_c^{FL} = \frac{1}{2h^2}\left[F_1 + F_2 + F_3 + F_5\right] = \nabla_E f_c^2 + O(1)$$

$$\nabla_h^2 f_c^{GL} = \frac{1}{18h^2}\left[13(F_3 + F_5) + 9(F_1 + F_2) + 18\,F_4\right] = \nabla_E^2 f_c + O(1)$$

Table I (cont.)

FL is only second order accurate on cells 1 and 2 which share the interesting property that they can both tile the plane. This fact has been verified on every cell shape possessing this property that has been tried to date[4]. However this has not been formally demonstrated. This property is not shared by GL since it is only first order accurate on cell 2. GL produces second order truncation errors on cells 3 and 4, contrary to FL which, surprisingly enough is only zeroth order accurate. A more detailed look at the formulas reveals however that they are proportional to the exact Laplacian. Although cells 3 and 4 cannot tile the plane individually, they can do so together. This, along with the fact that the ratio of GL to FL is smaller and greater than unity on cells 3 and 4 respectively, suggests that in some sense, the truncation errors are closer to first or second order, as was pointed out by Fritts.

In the context of adaptive grids, it is very important to use accurate pointwise approximations to operators that provide information about surface topology, in order to pull nodes to the areas where they are really needed[10]. This requirement is not as severe when approximating equations which contain this operator, because they are often multiplied out by the cell area which has the effect of guaranteeing the consistency of the discrete differential equations.

Truncation error of the gradient operator at a cell node

To date, very little is known about the accuracy of discrete operators constructed on arbitrary triangular grids, with the exception of the work done in finite-element theory[14]. Their philosophy however is somewhat different from the one adopted in finite-difference methods. In the former, basis vectors are typically defined at each mesh point, and the solution vector is linearly expanded in this basis with arbitrary coefficients that are chosen to make a functional stationary. This eliminates the need to approximate the differential operators since they can be directly applied to the original function. In contrast, finite-difference theory constructs these operators from the unknown solution at the grid points. Despite these differences, there exists a similar dependence of the truncation error bound of the gradient operator on the geometry of the cell, which brings both methodologies closer together. The motivation for a truncation error analysis is mainly to sharpen our instincts regarding the choice of an optimal cell or triangle geometry. It seems intuitively obvious that a grid composed of equilateral triangles is the best in the sense that it leads to the lowest maximum bound on

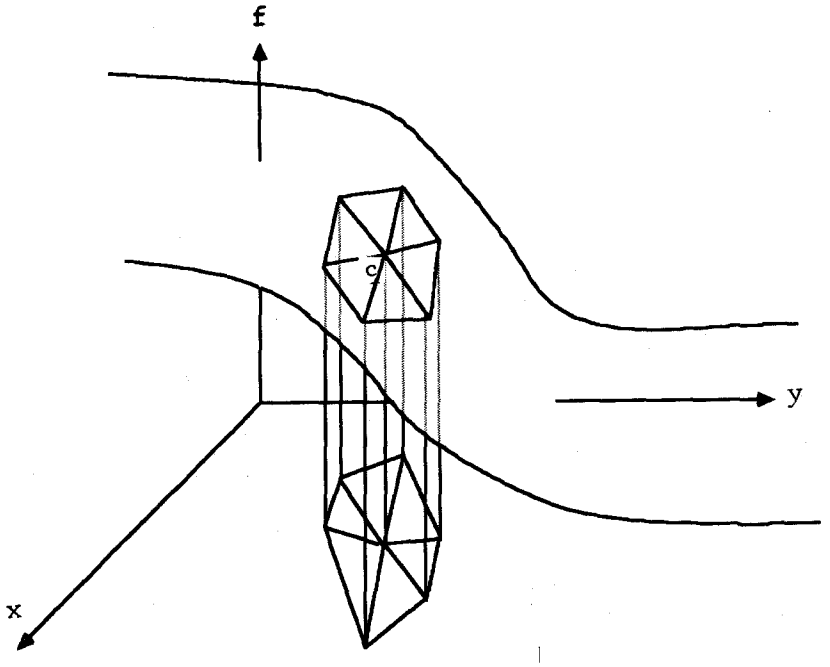


Fig. 9. The triangular grid is lifted up onto the solution surface. Well-behaved surface triangles in high gradient and curvature regions of the solution result in skewed elements when projected back onto the physical plane

truncation errors. However, note that cells built from isoceles triangles may be strong contenders for optimality, since these triangles are so closely related to the cartesian grid. Experience with curvilinear grids has proven that extreme skewness of the cells increases the inaccuracy of a computation. This is a consequence of an inverse dependence of the error on the Jacobian of the transformation between the computational and the physical domain. The calculation of truncation error estimates typically leads to formulas for maximum error bounds of the form[4]

$$\text{error} \leq K(f) \frac{h^n}{\text{minimum angle in triangle}} \tag{71}$$

where h is a typical scale length (e.g. average edge length). K is an analytic operator that acts on f. These maximum bounds are often much larger than the actual error incurred during the numerical computation. For example, if the triangle becomes very skewed, the denominator in (71) goes to zero, which magnifies the error bound. However, this does not necessarily imply that the error is large. Consider a triangular grid, not in the physical plane, but on the surface f itself (fig. 9). If f has very high curvature or gradient regions, triangles that are well structured on the surface will project down to the physical plane and produce skewed elements. But the following question must be posed: since the errors should intuitively be minimum when the surface is well resolved, shouldn't the well-behaved triangles lie on this surface, rather than on the x-y plane? The maximum error bounds will still take on a form similar to (71), but both h and the minimum triangle angle will contain information about the function f, which will probably have the effect of reducing the magnitude of K(f), and provide tighter error bounds. This speculative conclusion still requires verification.

Conclusion

In this paper, attention has been focused on the construction of discrete representations of the gradient and Laplace operators on unstructured triangular grids. Emphasis was placed on the derivation of a discrete Laplacian which was at least first order accurate. This objetive was not achieved. However, with the help of a variational principle, a new discrete approximation to the Laplacian (GL) was constructed, and shown to be at least first order accurate on simple cell shapes. The standard Laplacian (FL) was found to be zeroth order accurate on most cell shapes, except on those which tile the plane.

This substantiates, in an empirical manner, the observation that it is second order accurate in some average sense.

One of the main reasons for constructing operators which have a high order of accuracy (on a pointwise basis) is to correctly estimate the properties of the solution surface (e.g. curvature). These quantities are crucial within the context of adaptive grid strategies. Therein, nodes are dynamically moved, added, or deleted from the mesh according to the properties of the solution surface evolving in time.

Ongoing research is being focused on the construction of higher order accurate operators, as well as to sharpening the maximum bounds on truncation error estimates. The approach to the latter is based on performing a truncation error analysis on the solution surface rather than in the physical plane.

Appendix

For the interested reader, some geometrical relationships are presented that were derived as byproducts of the truncation error analysis[4]. These derivations demonstrate some of the subtelties involved in index manipulation when working on unstructured grids. The first formula provides a relation between the lengths of the sides of an arbitrary triangle, and the length of its medians. With the notation of figure 10, let

$$I = \sum_{i(c)} r_{i+1/2}^2 \quad \text{and} \quad J = \sum_{i(c)} r_{c,i}^2$$

where c is the triangle's centroid. The radius vector $\vec{r}_{i+1/2}$ is expanded along the two median vectors $\vec{r}_{c,i}$ and $\vec{r}_{c,i+1}$. I becomes

$$I = \sum_{i(c)} (\vec{r}_{i,c} + \vec{r}_{c,i+1})^2$$

$$= 2\,J + 2\,K$$

where

$$K = \sum_{i(c)} \vec{r}_{i,c} \cdot \vec{r}_{c,i+1} \quad .$$

To evaluate K as a function of J, let $\vec{r}_{c,i+1} = \vec{r}_{i,c} + \vec{r}_{i-1,c}$ which implies that

$$K = \sum_{i(c)} r_{i,c}^2 + \sum_{i(c)} \vec{r}_{i,c} \cdot \vec{r}_{i-1,c}$$

This leads to the desired result

$$\sum_{i(c)} r_{i+1/2}^2 = 3 \sum_{i(c)} r_{c,i}^2$$

The second formula presented is a consequence of the fact that the FL applied to linear functions is zero. Starting from FL evaluated at a

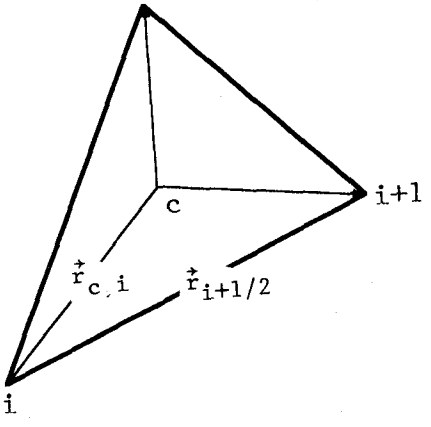


Fig. 10

cell center, the linear function values f^i are Taylor expanded:

$$f_i = f_c + \vec{r}_{c,i} \cdot \nabla f_c \quad .$$

Therefore FL becomes

$$\nabla_h^2 f_c^{FL} = -\frac{1}{2A_c} \sum_{i(c)} \frac{\vec{r}_{i+1/2} \cdot [\vec{r}_{i+1,c} \vec{r}_{c,i} - \vec{r}_{c,i} \vec{r}_{i+1,c}]}{(\vec{r}_{c,i} \times \vec{r}_{i+1,c}) \cdot \hat{z}} \cdot \nabla f_c \quad \text{(A1)}$$

where the triangle area $A^{i+1/2}$ has been replaced by a vector product. The numerator is a standard cross-product formula; therefore equation (A1) simplifies to

$$\nabla_h^2 f_c^{FL} = -\frac{1}{2A_c} \sum_{i(c)} \frac{\vec{r}_{i+1/2} \times (\vec{r}_{c,i} \times \vec{r}_{i+1,c})}{(\vec{r}_{c,i} \times \vec{r}_{i+1,c}) \cdot \hat{z}} \cdot \nabla f_c$$

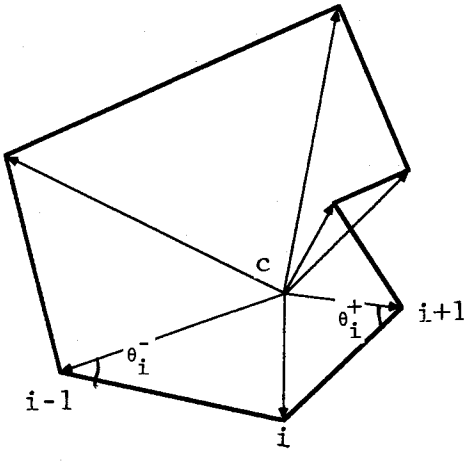


Fig. 11.

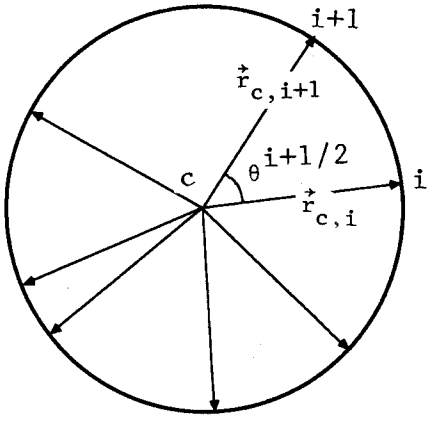


Fig. 12.

$$= - \frac{1}{2A_c} \sum_{i(c)} (\vec{r}_{i+1/2} \times \hat{z}) \cdot \nabla f_c = 0$$

which proves that the FL is exact for linear functions. Equation (A1) can be rewritten in terms of cotangents and set to zero to obtain the geometrical formula

$$\sum_{i(c)} (\mathrm{ctg}\theta_c^+ + \mathrm{ctg}\ \theta_c^-)\ \vec{r}_{c,i} = 0 \qquad \text{(A2)}$$

which is valid on any convex or concave polygon, with c in its interior or exterior. The notation is defined in fig. 11. A particular case is obtained if the nodes i lie on a circumference of radius $r_{c,i}$ with c at its center (fig. 12). Equation (A2) then reduces to

$$\sum_{i(c)} (\mathrm{tg}\ \frac{\theta_i^+}{2} + \mathrm{tg}\ \frac{\theta_i^-}{2})\ \vec{r}_{c,i} = 0 \quad .$$

References

1. Hessenius, K. A.; and Rai, M. M.: Applications of a Conservative Zonal Scheme to Transient and Geometrically Complex Problems. AIAA paper 84-1532.

2. Wedan, B.; and South, J. C.: A Method for Solving the Transonic Full-Potential Equation for General Configurations. AIAA paper 83-1889.

3. Berger, M.; Gropp, W., and Oliger, J.: Grid Generation for Time-Dependent Problems: Criteria and Methods. Numerical Grid Generation Techniques, Proc. of NASA Langley Research Workshop, Va, 181-188, (1980).

4. Erlebacher, G.: Solution Adaptive Triangular Meshes with Application to the Simulation of Plasma Equilibrium. Ph. D. Thesis, Columbia University, N.Y., (1984).

5. Pelz, R. B. and Jameson, A.: Transonic Flow Calculations using Triangular Finite Elements. AIAA Computational Fluid Dynamics Conf., 253-260, (1983)

6. Fritts, M. J. and Boris, J. P.: The Lagrangian Solution to transient Problems in Hydrodynamics using a Triangular Mesh. J. Comp. Phys. 31, 173-215, (1979)

7. Miller, K. and Miller, R. N.: Moving Finite Elements I. Siam J. Num. Anal., vol 18, No.6, 1019-1032, (1981)

8. Miller, K.: Moving Finite-Element Methods II. Siam J. Num. Anal., 18, No.6, 1033-1057, (1981)

9. Dukowicz, J. K.: Generalized Grids: and Application of the Voronoi Mesh. Adaptive Grid Workshop, Los Alamos

10. Erlebacher, G. and Eiseman, P. R.: Adaptive Triangular Mesh Generation, AIAA-84-1607, AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers conference.

11. Crowley, W. P., Free Lagrange Methods for Compressible Flows. Free-Lagrange Methods Conference, Hilton Head Island, SC, (1985)

12. Salzman, J.: A Variational Method for generating Multidimensional Adaptive Grids. Ph.D thesis, New York University, (1981)

13. Bauer, F., Betancourt, O., and Garabedian, P.: A Computational Methods in Plasma Physics. Springer Verlag, New York (1978)

14. Strang, G. and Fix, G.: An Analysis for the Finite-Element Methods. Prentice Hall, Englewook Cliffs, N.J., (1973)