Journal of Computational Physics **151,** 684–727 (1999)
Article ID jcph.1999.6214, available online at http://www.idealibrary.com on IDEAL®

Local Rectangular Refinement with Application to Nonreacting and Reacting Fluid Flow Problems

Beth Anne V. Bennett and Mitchell D. Smooke

Department of Mechanical Engineering, Yale University, New Haven, Connecticut 06520

Received September 18, 1997; revised January 22, 1999

A new solution-adaptive gridding method has been developed for the solution of discretized systems of coupled nonlinear elliptic partial differential equations on rectangular domains. Such a method is required for the numerical solution of realistic combustion problems, in which physical quantities may vary by orders of magnitude over one-tenth of a millimeter at atmospheric pressure, or over micrometers at higher pressures. The local rectangular refinement (LRR) method maintains orthogonality at grid-line intersections but lifts the tensor product restriction common to traditional grids, producing unstructured grids. Governing equations are discretized throughout the domain using newly derived forms, and Newton's method is used to solve the resulting system. On a simple test case with a known solution, the LRR method and its new discretizations are found to be more accurate than gridding methods representative of those appearing previously in the literature. For the more realistic problem of nonreacting driven square cavity flow, the LRR solution agrees very well with previously published data. When the LRR method is applied to a practical reacting flow (a rich axisymmetric laminar Bunsen flame with complex chemistry, multicomponent transport, and an optically thin radiation submodel), grid spacing highly influences the inner flame's position, which stabilizes only with adequate refinement. The vorticity–velocity formulation of the governing equations is shown to produce valid results when used in conjunction with the LRR gridding technique. Furthermore, each LRR grid is used to form a nonuniform equivalent tensor product (ETP) grid and also, in most cases, an equispaced fully refined (FR) grid; these additional grids are supersets of the LRR grids and thus contain refinement in exactly the same regions. Performance comparisons between the LRR, ETP, and FR grids indicate that the LRR method provides substantial savings in execution time and computer memory requirements, without compromising solution accuracy.

Key Words: adaptive methods; mesh refinement; finite difference discretizations; square cavity flow; axisymmetric laminar flame.

0021-9991/99 $30.00

1. INTRODUCTION

Combustion, a thermochemical and fluid dynamic process in which fuel and oxidizer meet and energy is released, is of fundamental importance. Despite technological advances in computer speed and size, it is still not possible to model accurately flames with both complex chemistry *and* complex geometry, because of the large computational grids and memory required by conventional methods. Memory demands are particularly high when the governing equations are solved using a coupled approach necessitated by the highly reactive nature of combustion problems, instead of the field-by-field (or "one variable at a time") solution procedure often employed in nonreacting fluid flow problems. This size constraint generates a need for numerical techniques which will solve a given problem to a desired degree of accuracy, using fewer grid points than a traditional mesh, while at the same time incorporating efficient use of memory and displaying competitive solution times. The combination of sophisticated adaptive gridding and realistic combustion modeling is the focus of the current work. The present paper develops a new adaptive gridding method and validates it using three test problems (two nonreacting flows and one reacting flow); additional combustion applications are explored elsewhere [4, 5, 46], with the latter reference being the first author's dissertation.

One of the most important components of the numerical solution process is the grid chosen to discretize the domain. Ideally, grid points should cluster in regions of high solution activity, but elsewhere only a few points might be needed to resolve the solution. A mesh exhibiting these characteristics can be produced via an iterative adaptive process which incorporates several ingredients, such as the weight functions for equidistribution; choice of mechanism for point addition or point moving; retention or nonretention of tensor product structure; maintenance of orthogonality at all, some, or none of the grid-line intersections; various mesh constraints; the discretization forms; and the adaption termination criteria. Not all choices will produce viable methods. Existing techniques can be classified as belonging to one of three categories, as follows.

Globally refined rectangular solution-adaptive gridding results in nonuniformly spaced meshes with a tensor product structure, meaning that every grid point is located at the intersection of two grid lines, each of which extends from one side of the domain to that opposite. This technique is rooted in one-dimensional mesh equidistribution and its supporting theory [8, 24, 28, 37]. A major advantage to this type of gridding is that the derivatives remain easy to calculate. In various applications (for example, [39, 42]), grid spacing is continually altered, based on weight functions comprising the gradient and curvature of the solution iterate [40]. Unfortunately, each time a grid line is added with the intent of reducing the error in a particular region, points are unnecessarily introduced at each intersection of this new line with all perpendicular mesh lines.

In the second category of solution-adaptive grid formulation, intersecting grid lines are no longer constrained to meet orthogonally, and the curvilinear grids in physical space may be mapped to simple tensor product grids in computational space. Weight functions are again employed in forming the mesh, but extreme deviation from orthogonality must be avoided in order to limit truncation error. Existing work can be divided into two groups: alternating adaption and solution, which has been applied to droplet burning and one-dimensional premixed flames [17], and to premixed flame propagation in two dimensions [11]; and simultaneous computation of node positions and solution, in which the physical coordinates become two additional dependent variables. The latter techniques were pioneered by [7];

more recently, similar methods have been applied to basic test functions in one and three dimensions [32]. Unfortunately, such methods make linear problems nonlinear and nonlinear problems much more nonlinear, limiting their practicality.

The third category of solution-adaptive grid construction is that of locally refined rectangular gridding. The grid is refined through cell subdivision, which occurs without the global introduction of many unnecessary points—a major advantage. Although the derivatives do not require transformation to a computational space, the loss of the tensor product structure increases the difficulty of finding neighboring points. Another challenge lies in properly treating points at the interfaces between different refinement levels of the mesh. Despite such complexities, these methods have been utilized in a variety of applications. A two-dimensional local mesh refinement method consisting of a base grid with superimposed finer submeshes was developed and applied to hyperbolic conservation laws [6]; this method was later extended to three-dimensional hyperbolic problems [3]. In such applications, the submesh solutions are advanced in time independently of one another, and values are periodically updated at points common to more than one mesh. Similar methods have been used for incompressible flows [21] and for premixed Bunsen flames with one-step chemistry approximations [14, 31]. Such methods, however, cannot be reasonably applied to fully elliptic *realistic* combustion problems, in which simultaneous solution at *all* grid points is necessary for obtaining *rapidly converging*, accurate solutions [48]. This disadvantage is overcome by the local rectangular refinement (LRR) method [46], which incorporates the simultaneous solution of all governing equations at all points in the domain by a damped, modified Newton's method. (It should be noted that while other combustion researchers have successfully implemented field-by-field solvers on unstructured grids, they have thus far solved only diffusion flame problems with one- or two-reaction chemistry approximations [34]. Spatial gradients of minor chemical species not present in such approximations are an order of magnitude larger than gradients of the major species; gradients of minor species in *premixed* flames (such as Bunsen flames) are larger by yet another order of magnitude. The LRR method has been used successfully to solve both diffusion flames and Bunsen flames incorporating complex chemical mechanisms [4, 5, 46], including the final application in the present paper.)

The remainder of the paper is organized as follows. The LRR solution-adaptive gridding method is examined in detail in Section 2. Section 3 develops new discretizations and new coarse–fine interface treatments, the accuracies of which have been tested against traditional discretizations appearing in the literature. Section 4 discusses the nonlinear system solver and also explores the nonstandard sparsity structure of the Newton's method Jacobian matrix, produced by the new inter-point couplings of the discretizations. In Section 5, three application problems are posed and solved: numerical results are evaluated for the temperature field in a rectangular heated plate; the temperature and flow fields of thermally and dynamically driven square cavity flows; and the temperature, flow, and chemical species fields in a rich Bunsen flame. Each of the latter two problems is governed by a different set of coupled, nonlinear, elliptic partial differential equations, the vorticity–velocity formulations of which are seen to produce valid results when used in conjunction with the new gridding technique. In addition, performance and accuracy comparisons are made among LRR solutions, solutions found on equivalently refined tensor product (ETP) grids (formed by extending grid lines of the LRR grids to each domain boundary), and solutions found on equispaced fully refined (FR) grids. All calculations presented in this paper were performed on an IBM RS/6000-590 computer. Finally, Section 6 draws conclusions and outlines future work.

2. LRR SOLUTION-ADAPTIVE GRIDDING ALGORITHM

The first step in any computational method is the discretization of the physical domain. In general, however, the purpose of grid adaption is defeated if too unsuitable a grid is imposed on a problem at the start, since the legacy of the initial grid may be too strong for the adaption process to overcome. A balance must be struck between too few and too many points, which can often be done only through experience. In the LRR solution-adaptive gridding method, the initial grid may be an equispaced or nonequispaced orthogonal tensor product grid, subject to two constraints. First, no part of the domain may become entirely devoid of points, so an upper limit (0.25) is imposed on the ratio of the maximum mesh spacing to the domain length, in each coordinate direction. Second, the mesh is buffered by restricting the ratio between adjacent point spacings, which, in turn, limits the truncation error; for the current applications, the ratio must fall between 2.5^{-1} and 2.5. It should be noted that while the nonreacting flows explored in the present paper begin with *equispaced* starting grids, the LRR method has been developed for the general case of *nonequispaced* starting grids. As will be demonstrated via the rich Bunsen flame application of the present paper, the latter grids are indispensable for initiating the adaptive solution of full-chemistry combustion problems [4, 5, 46], in which a small grid spacing near the burner mouth is a key component in obtaining a stable converged solution, regardless of the spacing in the rest of the domain.

2.1. *Mesh Structure*

The LRR starting grid is referred to as the Adaption 0 grid or base grid, because its points will be present in all successive adaptions. The cells, or boxes, of this grid will be labeled Level 0 boxes. This mesh is composed of two types of points, as shown in Fig. 1: interior points which have nine-point stencils associated with them (indicated by (a)), and external boundary points which have either six-point (b) or four-point (c) stencils. The members of each stencil will be referred to by their compass direction relation to the central point P (W, E, S, N, SW, SE, NW, or NE).

During the adaption process, individual boxes of the grid are flagged (as will be detailed later) for subsequent refinement, which entails placing a nine-point stencil within each box. The existing corners of the box become points SW, SE, NW, and NE of the new stencil. The central point is added at the box's center, and, if necessary, some or all of the W, E, S, and N points are also added, depending upon the configuration of the surrounding mesh. Thus, the refinement process subdivides the original Level 0 box into four smaller Level 1 boxes of equal size. The stencil which has been added during this process is said to be of Level 1. It is evident that a Level L box will have dimensions equal to $(\frac{1}{2})^L$ times the dimensions of the Level 0 box within which it is located.

Points in an adapted LRR grid commonly fall into three categories: regular interior points (points with nine-point stencils, at which four boxes adjoin); internal boundary points (points at the intersection of three boxes); and external boundary points (points touching fewer than three boxes). Internal boundary points lack a full computational stencil, since they lie on an interface between two different levels of refinement. Discretizations at such points are greater sources of error, especially when interpolation is substituted for governing equation discretization. In order to limit the rapidity with which grid spacing changes and, in turn, control the truncation error, the grids are forbidden to have two adjacent internal boundary points.

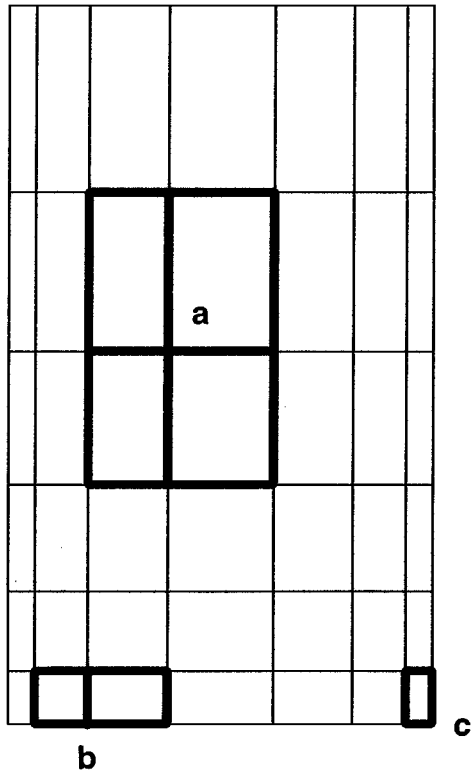

FIG. 1. Example of a nonequispaced initial grid for the LRR method, indicating a nine-point stencil at an interior point (a), a six-point stencil at an external boundary point (b), and a four-point stencil at an external boundary corner point (c).

The effect of this constraint is shown in Fig. 2. The first grid (a) contains two occurrences of pairs of adjacent internal boundary points (circled) and therefore violates the previously stated grid rule. This situation is remedied by refining additional cells, as demonstrated in (b); however, the resulting mesh is now subject to a second constraint, namely that the boxes meeting at any single point cannot differ by more than one level from each other. The circled point is located at the convergence of four boxes, but the box to its upper left is two levels coarser than that to its lower right. Therefore, one more cell must be refined before the grid is acceptable, as in (c). In fact, all grids which obey the second grid rule automatically satisfy the first. Application of these two grid rules results in a limited

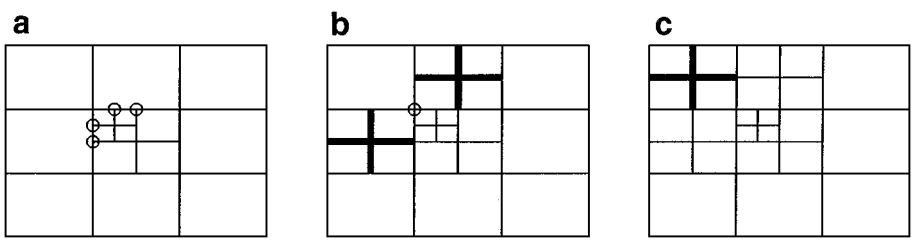

FIG. 2. Violation (a) and repair (b) of adjacent internal boundary points, producing an adjacent box-level violation, which is then repaired (c).

number of allowable mesh configurations, each consisting of one, two, three, or four boxes which meet to define a central point P of a computational stencil. There are 15 allowable regular interior configurations, 4 internal boundary configurations, and 16 external boundary configurations; all configurations are composed of reflections and/or rotations of 9 basic configurations. For further details, as well as configuration diagrams, see [46]. No other configurations are admitted by the grid rules.

By examining several configurations, as has been done in [46], it becomes apparent that the number of boxes which must be refined due to the addition of a single new stencil is bounded. This limit exists because the original grid, prior to that first added stencil, must have already contained a certain degree of refinement in order to satisfy all grid constraints. The least upper bound (LUB) on the number of possible refinements caused by the creation of a single stencil of Level M is

$$\text{LUB} = 3\max(M-1, 0) + 5\max(M-2, 0). \tag{1}$$

It is important to be able to derive such a limit, both to ensure that the grid constraints do not force complete refinement of the entire grid and to aid in programming the algorithm.

2.2. *Solution-Adaptive Refinement*

One of the four fundamental components of adaptive gridding, as described in [45], is "a means for communicating the need for a redistribution of points in the light of the error evaluation, and a means of controlling this redistribution." For two-dimensional local gridding, various researchers have tried several methods of flagging cells for refinement. The least automated method is that employed by [9] in the solution of an axisymmetric laminar diffusion flame: manual refinement of cells known *a priori* to be located near the flame front. A more practical method for targeting cells is the equidistribution of a positive weight function, which is equivalent to minimization of a grid property. Points will be made to cluster where the weight function is large, so it must contain some measure of the rapidity of change of the solution. The weight functions of [14, 31] contain linear combinations of first and second derivatives of some of the dependent variables, since these best approximate trends in the truncation error. In [49], circulation around each cell is examined, and refinement occurs when a preset value is exceeded; this technique is equivalent to using a weight function.

In the present method, weight functions are used to determine which grid cells are to be refined and to test grid termination criteria. First, N_{dep} weight functions are formed at the center of each box, where N_{dep} is the number of dependent variables present in the physical problem. These weight functions are then used to create N_{dep} new grids, the union of which forms the actual single new grid. The use of the N_{dep} weight functions could be interpreted as the employment of a single weight function which, in each box, is the maximum of the existing N_{dep} weight functions. However, the normalization procedures used here negate the exact equivalence of these approaches.

Because each of the N_{dep} weight functions is to be formed at each box's center, it will use dependent variable values from each of the four corners of that particular box, since those values can be easily accessed. Thus, only first derivatives will be present in $\mathcal{W}$, since higher derivatives require values at additional grid points. It should be noted that weight functions based only on first derivatives may contain some numerical fluctuation stemming from

local solution error within the bounds of the tolerance to which that solution was previously determined. However, the weight functions are first smoothed, as will be described shortly, and then globally subequidistributed, with both processes serving to minimize greatly the effect of any spurious local fluctuations. It is also true that the weight functions do not contain a direct approximation of truncation error as is sometimes the case in adaptive methods, because such an approximation would require at least 10 times as much computation as the current approach (for a practical calculation containing 10 to 20 derivative terms in each of its governing equations, on average); the additional computational cost is not warranted, given the manner in which the weight functions are employed.

The following form has been chosen for the weight function,

$$\mathcal{W}^{\mathrm{U}}_{k,b}(x,y) = 1 + \left\{ \frac{\alpha_k \left| \sqrt{\left(\left.\frac{\partial f_k}{\partial x}\right|_b \right)^2 + \left(\left.\frac{\partial f_k}{\partial y}\right|_b \right)^2} \right|}{\max\limits_b \left| \sqrt{\left(\left.\frac{\partial f_k}{\partial x}\right|_b \right)^2 + \left(\left.\frac{\partial f_k}{\partial y}\right|_b \right)^2} \right|} \right\}, \tag{2}$$

where the f_k ($k = 1, 2, \ldots, N_{\mathrm{dep}}$) are the dependent variables, subscript b denotes the box number, and superscript U stands for "unsmoothed." If any dependent variable happens to be constant over the entire grid, the denominator becomes zero, provisions for which have been made in the program. The weight function includes the additive constant "1" to temper solution adaptivity with grid uniformity. The derivatives in (2) are discretized as follows, where Δx and Δy are the box dimensions, and new subscripts represent lower left, lower right, upper left, and upper right:

$$\begin{aligned} \left[\!\left[\frac{\partial f_k}{\partial x} \right]\!\right]_{\mathrm{B}} &= \frac{1}{2}\left[\frac{(f_{k,\mathrm{UR}} - f_{k,\mathrm{UL}})}{\Delta x} + \frac{(f_{k,\mathrm{LR}} - f_{k,\mathrm{LL}})}{\Delta x} \right] \\ \left[\!\left[\frac{\partial f_k}{\partial y} \right]\!\right]_{\mathrm{B}} &= \frac{1}{2}\left[\frac{(f_{k,\mathrm{UR}} - f_{k,\mathrm{LR}})}{\Delta y} + \frac{(f_{k,\mathrm{UL}} - f_{k,\mathrm{LL}})}{\Delta y} \right]. \end{aligned} \tag{3}$$

The error terms (not shown) are second order in Δx and Δy.

The overall character of adaptive grids reflects the nature of the weight functions; thus, smoothing the weight functions indirectly smooths variations in the grid spacing. A Laplace-type filter is used, discretized differently than other LRR derivative discretizations, since the weight functions are associated with the *boxes* rather than with the grid points themselves. At horizontal (vertical) boundaries, $\partial^2\mathcal{W}/\partial y^2$ ($\partial^2\mathcal{W}/\partial x^2$) is approximated by zero. This approximation means that the weight functions at the boundaries are smoothed using information from along the boundaries only, which has, in practice, no effect on the amount of refinement at the domain boundaries. The grid rules allow 36 interior box configurations, 28 boundary box configurations, and 32 corner box configurations. All of these configurations are superpositions of rotations and/or reflections of only five base configurations, and a Laplacian discretization has been derived for each, as detailed in [46]. Discretization of the zero Laplacian is simply a mechanism through which to create a smoothing filter for the weight function values. It can be recast into a more useful form by solving for the value of the weight function at the center of the central box, which is then averaged with the unsmoothed value to produce the smoothed weight function. This process, which constitutes a single smoothing pass, is performed on each weight function in each box, with

the exception of the four domain corner boxes, whose weight functions remain unchanged. The number of smoothing passes, N_{smth}, must be specified and is typically 8 to 10.

During the adaption process, the goal is the creation of a grid on which the weight functions are *sub*equidistributed, meaning that the following inequality will be satisified for each dependent variable f_k, for each box b in the grid:

$$\int\int_{\text{Box } b} \mathcal{W}_{k,b}(x, y)\, dx\, dy \leq C_k. \tag{4}$$

This expression can be rewritten in a more useful form as

$$\frac{1}{C_k}\int\int_{\text{Box } b} \mathcal{W}_{k,b}(x, y)\, dx\, dy \leq 1. \tag{5}$$

The double integral is approximated by multiplying the integrand, evaluated at the box center, by the area of the box. If the left side of (5) exceeds unity, then the factor by which it is in excess equals the number of boxes into which box b must be divided in order to achieve subequidistribution. However, any given box is not allowed to undergo more than one refinement during a single sweep through the grid, so if the left side exceeds 4, a state of subequidistribution will not immediately be reached when the box is quartered. If, instead, the left side is less than 4, refinement *will* result in local subequidistribution.

In practice, the constants C_k are found first, by integrating $\mathcal{W}_{k,b}$ over the entire domain and dividing by the number of boxes N_{bxs}, so that the quantity on the left side of inequality (5) can be calculated for each value of k and b. This value is then divided by $\varepsilon_{\text{equi}}$, a user-specified parameter typically ranging from 1.5 to 3; $\varepsilon_{\text{equi}}$ is chosen as 2 for the application problems presented here. The purpose of this division is to avoid overdoing the refinement process, which may occur if several of the left-side values are only slightly greater than 1, and it also allows the adaption to proceed in a more controlled manner. Interpolation onto the new points is a simple averaging process, since any new W, E, S, or N point lies halfway between existing points. Dependent variable values at the central point of the new stencil are approximated by averaging the values from the four corners of the refined box.

2.3. *Increasing Grid Robustness*

Not only the magnitudes of dependent variables at given locations may change as the grid is altered. If the grid has been greatly refined in the region where the high gradients were originally located and then the very act of refining (and then re-solving) causes those gradients to migrate, successive grids will contain unnecessary points in the region that was initially refined. Therefore, remeshing is performed at the start of each LRR adaption, which removes any unnecessary refinement. The remeshing time depends solely on N_{dep} and N_{bxs}, so for a given problem size, the more complex the governing equations (and thus the longer the solution process), the smaller the percentage of the total run time that is occupied by remeshing. In the nonreacting flow applications examined in the present paper, remeshing consumes about 0.2% of the total CPU time. However, in a practical combustion calculation such as the rich Bunsen flame explored in the present paper or the flames in [4, 5, 46], for example, remeshing occupies roughly 0.001% of the total CPU time, with the remainder spent on the Newton solver (Jacobian formation: approximately 90%; solution of the resulting linear system: approximately 10%).

If too small an area is refined, the front of interest may try to equilibrate at a position beyond the refined area, becoming falsely trapped at the coarse–fine interface. In addition,

because the discretization errors scale as spatial derivatives of the dependent variables, error at internal boundary points could be reduced if interfaces were made to occur further from the region of high gradients. To remedy both problems, the region of refinement is extended in a controlled manner by a process referred to here as *layering*. At the start of each adaption, certain cells are flagged for refinement based on weight function excesses, as described above. Then, any cells which are within $N_{\text{layer},\mathcal{A}}$ boxes of these are also refined, thus surrounding the originally refined area with $N_{\text{layer},\mathcal{A}}$ layers of refined cells, where $\mathcal{A}$ refers to the adaption number.

The layering parameter $N_{\text{layer},\mathcal{A}}$ is calculated based on the user-specified number of layers of refinement $N_{\text{layer},1}$ desired during the first adaption. Typical values of $N_{\text{layer},1}$ are less than 5. The number of layers with which eligible cells will be surrounded during Adaption $\mathcal{A}$ can be defined recursively as

$$N_{\text{layer},\mathcal{A}} = \min[0,\, 2 \cdot (N_{\text{layer},\mathcal{A}-1} - 1)]; \tag{6}$$

the equivalent nonrecursive definition is given by

$$N_{\text{layer},\mathcal{A}} = \min\left[0,\, 2^{(\mathcal{A}-1)} \cdot (N_{\text{layer},1} - 2) + 2\right]. \tag{7}$$

Had $N_{\text{layer},\mathcal{A}}$ been chosen independent of the adaption number $\mathcal{A}$, then the actual width of the buffering layer would decrease by a factor of 2 with each successive adaption; thus, to maintain adequate layering, the parameter $N_{\text{layer},\mathcal{A}}$ must depend on $\mathcal{A}$. However, it is important to have the layered region from one adaption fall within the layering from the previous adaption, because this means that successively smaller areas are being refined due to layering. That this important criterion is met can be seen in recursive definition (6). A second key feature of the $N_{\text{layer},\mathcal{A}}$ definition is that as $\mathcal{A} \to \infty$, the layering region thickness approaches $\frac{1}{2}(N_{\text{layer},1} - 2)$, as measured in terms of Level 0 box widths. More generally, the layering region thickness approaches a nonzero limit, and it approaches this limit *from above*. Such behavior is required in order to avoid the placement of coarse–fine grid interfaces infinitesimally close to the region of high solution activity.

An example of layering is shown in Fig. 3, in which the number of layers is 1, for simplicity. A single cell (referred to as a primary cell) has undergone refinement because of weight function excesses, shown in bold. Without layering, the final state of the grid would be as pictured in (a). However, with layering, a few additional cells (secondary cells) are refined (b), to move the grid interfaces further from the region of high solution activity, thus reducing the error and aiding the convergence process. Dotted lines indicate a cell refinement performed so that the grid obeys all grid constraints. To avoid unnecessary refinement, no secondary cell is refined if it is already of a finer level than the primary cell.

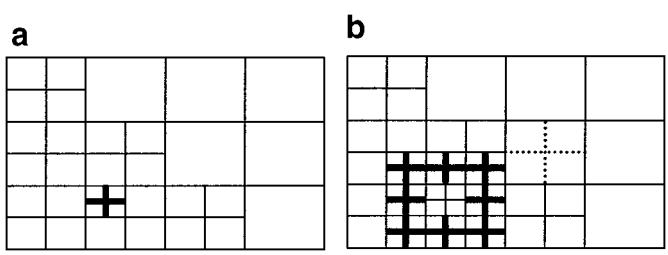

FIG. 3. Grid before (a) and after (b) layering. Dotted lines in (b) indicate additional refinement forced by grid constraints.

Layers are put around only those primary cells being refined during the *current* adaption; if each adaption were to include the layering of previously refined cells, then eventually the entire grid might be refined.

Finally, the mesh must undergo one last sweep to remove single-box holes, which occur when all boxes surrounding a given box are refined, but the central box has not been flagged for refinement. This situation does not violate any prescribed grid criteria, but if the central box remains unrefined, the midpoints of its sides are four internal boundary points, at which the discretizations are less accurate than at regular interior points. If, on the other hand, the central box is refined, entailing the addition of only one point, the four internal boundary points are transformed into regular interior points. Clearly, the benefits of eliminating four internal boundary points far outweigh the cost of adding a single point. This algorithm adds fewer points than that employed by some finite element researchers [1, 2]. Single-box hole removal is also performed on boxes along the external domain boundary, as long as all surrounding boxes belong to a finer level than the box under consideration.

Interpolation is used to form an initial guess on the new grid, which will then undergo an iterative solution process (i.e., Newton's method); interpolation is never used to produce a final solution. The overall adaption process terminates when the smoothed weight functions are subequidistributed to within 5%.

3. DISCRETIZATION TECHNIQUES

Given a particular point P, how are the remaining points of its computational stencil chosen? In the majority of the literature, finite difference discretizations of derivatives on unstructured rectangular grids use single-scale stencils (see, for example, [3, 6, 14, 21, 31, 44]). In such an approach, the chosen stencil is the smallest such that

$$\begin{aligned} x_{\mathrm{NE}} - x_{\mathrm{N}} = x_{\mathrm{SE}} - x_{\mathrm{S}} = \Delta x_+, \qquad & y_{\mathrm{NE}} - y_{\mathrm{E}} = y_{\mathrm{NW}} - y_{\mathrm{W}} = \Delta y_+, \\ x_{\mathrm{N}} - x_{\mathrm{NW}} = x_{\mathrm{S}} - x_{\mathrm{SW}} = \Delta x_-, \qquad & y_{\mathrm{E}} - y_{\mathrm{SE}} = y_{\mathrm{W}} - y_{\mathrm{SW}} = \Delta y_-, \end{aligned} \tag{8}$$

while at the same time at least one of the four rectangular "macro-cells" comprising the stencil (P–W–SW–S, P–E–NE–N, P–S–SE–E, P–N–NW–W) must have no internal grid structure, as shown in Fig. 4. Derivatives are discretized in the same way as on globally refined rectangular grids. In general, such discretizations are first order in the grid spacing,

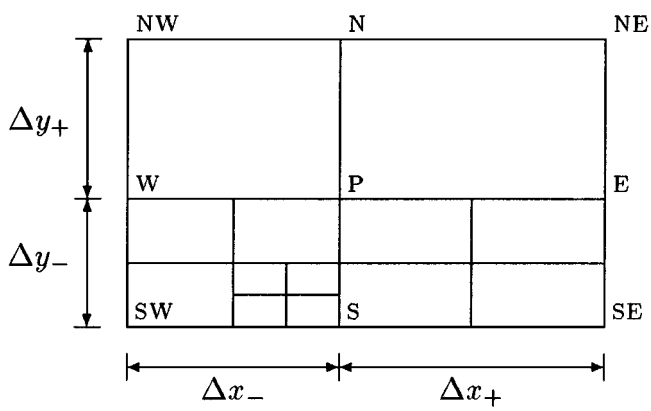

FIG. 4. Example of a single-scale computational stencil.

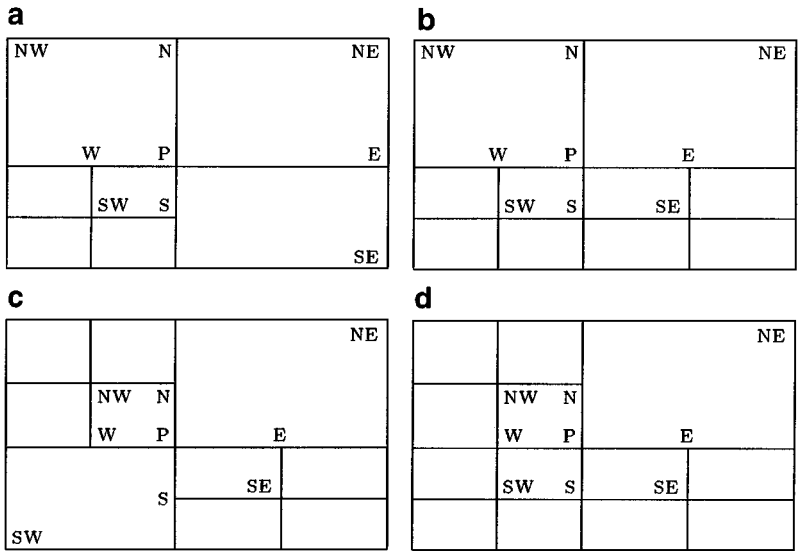

FIG. 5. Examples of multiple-scale computational stencils.

although there do exist nonuniform grid spacings for which the error is better than first order. However, only in the equispaced case is the error truly second order in the grid spacing. The largest drawback to single-scale discretizations is that their use precludes the conservation of fluxes across grid interfaces.

In the current research, however, derivatives at regular interior points (and domain boundary points) are discretized in a novel multiple-scale way, which couples the variable values at points belonging to different levels of the mesh. Given a point P, W is the first point encountered when proceeding westwards from P; E, S, and N are the first points encountered when traveling in their respective compass directions from P. Point SW is the lower left corner of the box to the lower left of P, SE is the lower right corner of the box to the lower right of P, and so on. Thus, the points closest to central point P make up the multiple-scale stencil. Four examples are shown in Fig. 5, and [46] details the full range of multiple-scale stencils, including the corresponding derivative discretizations; representative discretizations are also given in the appendices of [4].

In addition to the natural coupling between information of differing length scales, multiple-scale discretizations more firmly tie together the solution values at the internal boundary points with those at regular interior points. This connection helps to overcome what is traditionally a weak link in unstructured rectangular gridding: inaccurate determination of the solution at internal boundary points. Discretizations, presented in [4, 46], have been derived by combining function-value Taylor expansions at neighboring points about point P and then eliminating as many unwanted terms as possible. In 9 of the 15 possible allowable mesh configurations around regular interior points, the multiple-scale and single-scale discretizations are of the same order. Moreover, the multiple-scale truncation errors are all smaller than the corresponding single-scale ones by a factor of either 2 or 4; examples are given in Appendix A of [4]. In the remaining six cases, the multiple-scale stencils produce truncation errors which are first order, regardless of the spacing of the underlying grid. However, this behavior is far outweighed by the benefits of numerically coupling information from different length scales, as will be seen later.

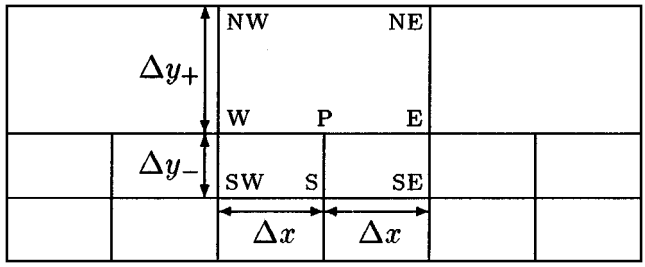

FIG. 6. Internal boundary point lacking a northern neighbor.

Treatment of the internal boundary points is much less straightforward. The possibilities examined here will focus on an internal boundary point P lacking a north neighbor, shown in Fig. 6; although the displayed base grid is uniformly spaced in each coordinate direction, such spacing is not required. The majority of the literature dealing with finite differencing on unstructured grids uses a two-point linear interpolant (see, for example, [3, 6, 21, 49]) to replace the governing equation for each variable at P. This two-sided interpolant (W and E) is accurate to first order, with a second-order error. However, when P appears within a discretization scheme centered at one of its neighbors, this error contributes first- or zeroth-order error if the discretization is that of a first or second derivative, respectively. A smaller group of researchers (see [13, 14, 31, 44]) uses a five-point interpolant (W, E, S, SW, and SE), accurate to second order, which contributes either second- or first-order error to first- or second-derivative discretizations at neighboring points, respectively. Despite this improvement, the five-point interpolant's one-sidedness can produce unwanted solution oscillations. Higher-order interpolating polynomials involve more points, destroy the structure of the Newton's method Jacobian matrix (producing more than nine nonzero blocks per block row), and are even more oscillatory.

The current research explores discretization of the actual governing equations at internal boundary points as an alternative to interpolants. For the configuration in Fig. 6, x-derivatives can be discretized using traditional three-point centered differences, which are flux-conservative. The discretization of $\partial f/\partial y$, however, can be used to illustrate a new set of discretizations, to be referred to as a "pseudo-nine-point" treatment, which has been derived and is presented in detail in [46] and in Appendix B of [4]. By averaging values from points NW and NE, a pseudo-north neighbor can be created for internal boundary point P. Such a pseudo-point will appear neither in the grid nor in the discretized governing equations, but, in discretization forms normally used for *globally* refined rectangular grids, dependent variable values at what *would* be point N are replaced by the pseudo-north neighbor's values. The first-derivative discretization becomes

$$\left[\!\left[\frac{\partial f}{\partial y}\right]\!\right]_{\mathrm{P}} = \frac{\frac{1}{2}(f_{\mathrm{NE}} + f_{\mathrm{NW}}) - f_{\mathrm{S}}}{(\Delta y_+ + \Delta y_-)}$$

$$\text{Error} \approx \begin{cases} \dfrac{-4 f_{xx}(\Delta x)^2 - 3 f_{yy}(\Delta y)^2}{12(\Delta y)}, & \text{if } \Delta y_+ = 2\Delta y_- \equiv \Delta y; \\ \dfrac{-f_{xx}(\Delta x)^2 - f_{yy}[(\Delta y_+ + \Delta y_-)(\Delta y_+ - \Delta y_-)]}{2(\Delta y_+ + \Delta y_-)}, & \text{otherwise.} \end{cases} \tag{9}$$

This error term is first order in the grid spacing, and the presence of f_{xx} comes from the x-direction interpolation used to produce values at the pseudo-north point. Fortunately, the coarser areas of the grid (i.e., the northern end of this stencil) have less sharply varying

solution derivatives. Furthermore, because of the use of layering, most internal boundary points occur in noncrucial areas of the grid. Therefore, the error is small, in practice. The derivative $\partial^2 f/\partial y^2$ is treated in a similar manner.

A disadvantage to the pseudo-nine-point discretizations is their lack of flux conservation, which is especially important at grid interfaces, where the length-scale change artifically imposed by the grid may introduce spurious phenomena. The remedy lies in choosing an imaginary control volume associated with point P, and applying Green's theorem in the plane to an appropriate integral. The resulting flux-conservative expression for the first derivative has not yet undergone completion of testing within the algorithm. However, preliminary results indicate that this discretization may merit additional investigation. Aside from the results presented in Fig. 13 (in which various discretizations and interpolants are examined), all calculations presented here employ pseudo-nine-point discretizations at the internal boundary points.

Important to further exploration of flux-conserving discretizations is the concept of grid link reciprocity: if point A is involved in point B's computational stencil, then B is also a member of A's stencil, and if A is *not* in B's computational stencil, then B will *not* be in A's stencil. Flux conservation requires the representation of a flux leaving one grid point and entering the next to be formulated in exactly the same manner within discretizations at each of the two points involved. The multiple-scale approach for the regular interior points and domain boundary points, coupled with the two- or five-point interpolant at internal boundary points, preserves grid link reciprocity. However, use of the interpolants confounds any attempt at flux conservation, a situation which can be improved only by discretizing the governing equations at the internal boundary points. Unfortunately, utilizing the pseudo-nine-point discretizations destroys the grid link reciprocity. Since it is not possible, within the nine-point stencil constraint, to conserve flux at all points in an LRR grid, a compromise is made. During x-derivative discretization, the left (right) face of each control volume is located halfway between points P and W (E); for y-derivative discretization, the bottom (top) face is halfway between points P and S (N). No control volumes protrude beyond the domain boundary. Near points other than internal boundary points, these two sets of control volumes intersect and yet neither overlap nor leave gaps in the domain, so only at the internal points boundary are two sets truly used.

Convective terms are discretized differently than other terms involving first derivatives, in order to avoid the unphysical oscillations or iterative-method divergence commonly caused by the use of centered differences on these terms. Traditional upwinding schemes can introduce artificial viscosity effects [35] at high Reynolds numbers [16] or for insufficiently refined grids. The generalized discretizations derived here retain this feature, which can be minimized with a fine enough grid. Discretizations of convective derivatives on LRR grids are a generalization of traditional upwind differencing. For regular interior points, the traditional scheme itself is used. At internal boundary points, discretizations utilizing the philosophy of the pseudo-nine-point approach have been derived.

As an example, an internal boundary point P lacking a north neighbor is again considered, shown in Fig. 6. Values at the NW and NE points are averaged and then used in place of the N values in the traditional scheme, resulting in

$$\left[\left[v\frac{\partial f}{\partial y}\right]\right]_{\mathrm{P}} = \begin{cases} \dfrac{\frac{1}{2}\left[v_{\mathrm{P}} + \frac{1}{2}(v_{\mathrm{NE}} + v_{\mathrm{NW}})\right]\left[\frac{1}{2}(f_{\mathrm{NE}} + f_{\mathrm{NW}}) - f_{\mathrm{P}}\right]}{\Delta y_+}, & \text{if } \frac{1}{2}\left[v_{\mathrm{P}} + \frac{1}{2}(v_{\mathrm{NE}} + v_{\mathrm{NW}})\right] < 0, \\ \dfrac{\frac{1}{2}(v_{\mathrm{P}} + v_{\mathrm{S}})(f_{\mathrm{P}} - f_{\mathrm{S}})}{\Delta y_-}, & \text{if } \frac{1}{2}(v_{\mathrm{P}} + v_{\mathrm{S}}) > 0, \end{cases} \tag{10}$$

which has a first-order error associated with it. The interpolation error, which is responsible for a term in the truncation error involving Δx, is of the same order as the rest of the error and thus does not heavily influence the accuracy. The one drawback to this discretization, however, is that it is not flux-conservative. More recently, a flux-conservative form has been derived by the authors and shows promise (please see Appendix C of [4]); this form, along with the flux-conservative first-derivative discretization mentioned above, may be examined in future work.

Even the potential inclusion of the two flux-conservative forms currently under investigation, mentioned above, will not render the LRR method a completely flux-conservative one; as stated earlier, it is impossible to conserve flux at all points in an LRR grid while using only nine-point stencils. However, as pointed out in [27], it is not possible to achieve simultaneously both flux conservation *and* accuracy. This reference states that, at grid interfaces, "conservation proves to be an important property mostly in cases of moving shocks for accurate prediction of their location and speed. Conversely, accuracy is more of an issue in a boundary layer, since the second-order derivatives (viscous terms) are important ..." [27, p. 129]. The reader is reminded that the LRR method has been developed with a view toward low Mach number laminar flame applications [4, 5, 46], for which accuracy is crucial.

4. NUMERICAL SOLUTION PROCEDURE

Discretization of the governing equations produces a system of N_{eq} coupled nonlinear equations, where the number of equations N_{eq} equal N_{pts} multiplied by the number of dependent variables N_{dep} at each grid point. This discretized system can be written in residual form as

$$F(U) = 0, \tag{11}$$

where U is the unknown vector. Newton's method will be used to seek a solution U^* to this system, given an initial guess U^0. At the nth iteration of the method, the $n+1$st solution iterate is formed from the nth iterate through the equation

$$J(U^n)(U^{n+1} - U^n) = -\lambda^n F(U^n), \qquad n = 0, 1, 2, \ldots. \tag{12}$$

The Jacobian matrix is given by $J(U^n) = \partial F(U^n)/\partial U$, and the parameter λ^n $(0 < \lambda^n \leq 1)$ multiplicatively damps the Newton correction [12].

If the solution is converging quickly enough (as quantified in [38] via a combination of correction step norms, preset constants, and iteration number), then the previous Jacobian can be reused. This modified Newton's method displays approximately linear convergence, but the computational time may be shorter overall simply because of the reduction in the number of time-consuming Jacobian evaluations. The Newton iteration is considered to have converged when the 2-norm of a scaled correction vector is less than a preset, problem-specific tolerance. Several steady-state solves may be performed consecutively, with successively tighter tolerances. The spatially elliptic discretized governing equations can be made parabolic in time by appending the term $\partial U/\partial t$ to the residual form of the equations [41]. Inclusion of the transient terms makes the Jacobian more diagonally dominant, generally improving convergence of the linear algebra solver. In addition, the same Jacobian is used for several time steps, saving computational time. The time step is chosen

adaptively, as in [42]. Depending upon the difficulty of the application problem, the time-relaxation procedure may be employed for a specified number of time steps, prior to the solution of the steady-state equations.

Regardless of the Jacobian structure, which will be addressed below, application of Newton's method has linearized the original system of equations. Based on the results of a linear system solver study [20], the bi-conjugate gradient stabilized (Bi-CGSTAB) method [47] has been chosen, with a Gauss–Seidel preconditioner. The linear system solver is considered to have converged when the 2-norm of the scaled residual vector is less than or equal to one-tenth of the tolerance used in Newton's method.

In practice, since all computational stencils used here employ at most nine points, the Jacobian matrix will contain a maximum of $9N_{\text{dep}}^2 N_{\text{pts}}$ nonzero entries. Their location depends upon the numbering scheme by which the equations are ordered, which is usually the same as the grid point numbering scheme. A closer examination of Jacobian structure requires the concept of the "Jacobian ordering" of grid points (the ordering by which the corresponding equations are arranged in the Jacobian), which, in the current research, may differ from the actual grid point numbering scheme (to be referred to as the "physical grid point ordering"). Because the grids produced via the LRR method are unstructured, the grid points are numbered in the sequence in which they are added to the grid. This physical grid point ordering is used solely in the storage and accessing of values in the FORTRAN arrays. To avoid excessive rearrangement of array entries, linked lists are used to switch between the Jacobian and physical grid point orderings. Since different stencils involve different groups of grid points, the choice of stencil type, (i.e., multiple-scale or single-scale) affects the structure of the Jacobian, as well as the Jacobian ordering of the points. The following two examples and ordering choices are based upon the new multiple-scale stencils.

The first scheme, implemented on the grid in Fig. 7a, numbers all of the points in the base grid, moving from the lower left to the upper right, followed by all of the points belonging to the next finest grid level (again from lower left to upper right), and so on. In Fig. 7b, the sparsity structure of the corresponding Jacobian is presented. Each box in the matrix is actually an $N_{\text{dep}} \times N_{\text{dep}}$ block, and boxes occupied by black dots denote nonzero blocks. The Jacobian's most noticeable characteristic is its lack of the traditional block nine-diagonal form. Although a block nine-diagonal structure is somewhat present in the upper left two-thirds of the matrix, the rest does not follow any obvious pattern. A major drawback to this ordering is the large bandwidth (19 blocks, in this case), which will probably slow the convergence of the linear algebra solver. Nonreciprocity of grid links produces a lack of structural symmetry.

The bandwidth can be greatly reduced by choosing a point ordering in which points are close in sequence to the members of their computational stencils. This strategy motivates the next scheme, incorporated in the LRR method and shown in Fig. 8a. The grid is the same, but the numbering now begins in the lower left corner and ends in the upper right. The Jacobian structure produced by this new ordering appears in Fig. 8b. The bandwidth has decreased to only 13 blocks, a marked improvement over the previous scheme and one which is even more substantial for larger grids. The overall Jacobian structure in Fig. 8b is much closer in appearance to the traditional block nine-diagonal form. A major advantage to the second point-ordering scheme is that the main block diagonal is immediately adjacent to an upper and a lower block diagonal. This characteristic, necessary for implementation of the linear algebra system solver, is one which the first scheme does not possess. However, even with the desired placement of these diagonals, the standard solver still has required

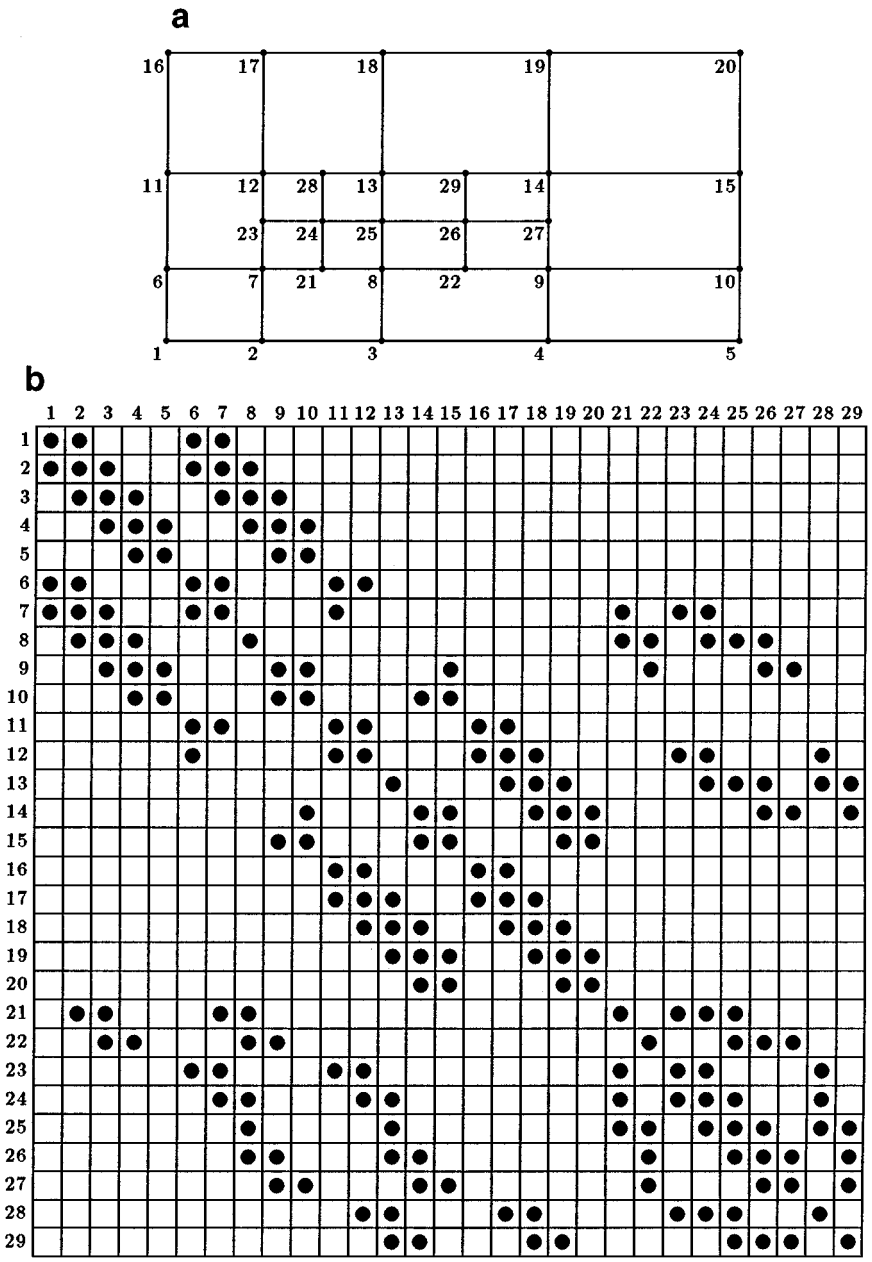

FIG. 7. First grid numbering scheme: Jacobian ordering (a) and sparsity structure of resulting Jacobian matrix (b).

modification to accommodate the placement of the remaining block "diagonals," because points frequently participate in more than nine stencils. In the extreme case, a single point may participate in as many as *17* stencils, a rare occurrence reflected by a block column containing 17 nonzero blocks.

One disadvantage lies in the impact of the unstructured nature of LRR grids on the efficiency with which the Jacobian matrix can be evaluated. For *structured* grids (and the resulting structured Jacobians), one can simultaneously evaluate groups of independent

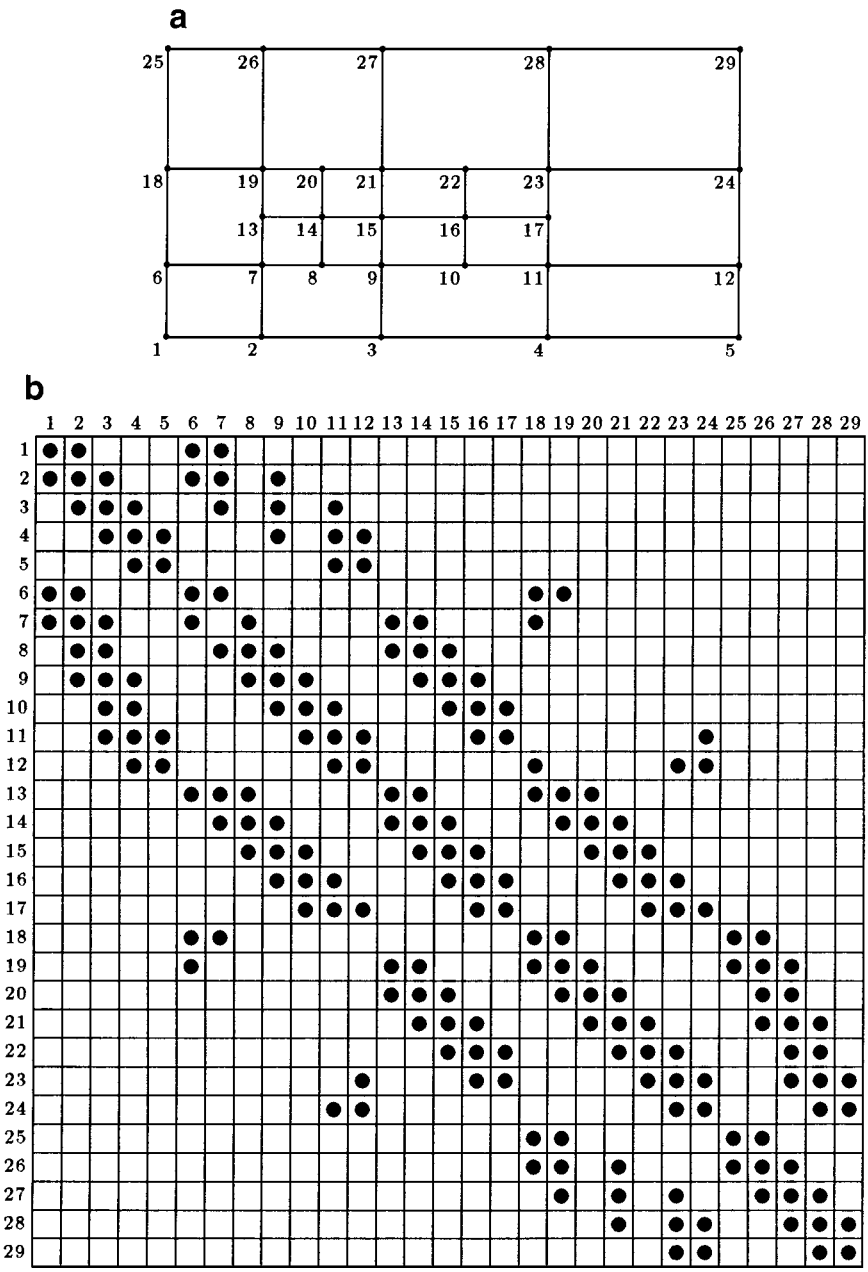

FIG. 8. Second grid numbering scheme: Jacobian ordering (a) and sparsity structure of resulting Jacobian matrix (b).

columns of the Jacobian with a minimum of subroutine calls [10]; the formation of the column groups is based on the computational stencil width [42]. Unfortunately, lack of structure in the LRR grids dictates that this efficient technique cannot be applied and instead necessitates writing two governing equations subroutines: one which evaluates all N_{dep} residuals at all grid points and one which does so at only a single grid point. The Jacobian is still formed block column by block column, but evaluation requires one call

to the former routine and N_{eq} calls to the latter; Jacobian formation on a tensor product grid using the aforementioned efficiencies requires solely $\frac{1}{9}N_{eq}+1$ calls to the former routine.

5. NUMERICAL RESULTS AND DISCUSSION

5.1. *Rectangular Heated Plate*

The LRR method is first applied to the problem of finding the temperature distribution in a rectangular heated metal plate. The extremely thin plate, oriented in a horizontal plane, is considered two-dimensional and gravitational effects are neglected. Its physical properties are temperature-invariant, and it is in contact with a heat source. Air, taken as incompressible, flows over the plate with a velocity $\mathbf{v}_{air} = (u_{air}, v_{air})$, in which $u_{air} = v_{air}$. A top view of the plate is displayed in Fig. 9. Because the region of high solution activity is oriented obliquely with respect to the boundary, as shown, this problem strongly tests the capabilities of any gridding method.

By assuming a Peclet number of unity, an elliptic partial differential equation is derived from the energy equation, which, when nondimensionalized, becomes

$$\frac{\partial T}{\partial x} + \frac{\partial T}{\partial y} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \dot{q}(x, y), \tag{13}$$

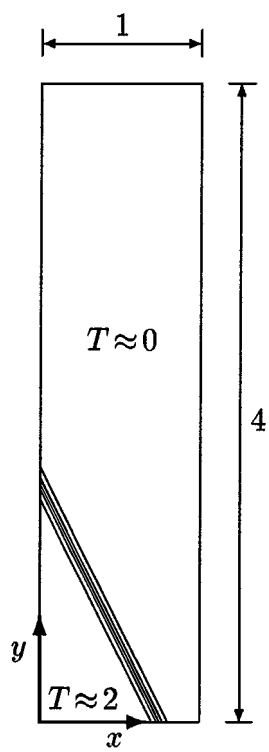

FIG. 9. Physical configuration for the rectangular heated plate problem, including five solution contours ($T = 0.2$, $T = 0.6$, $T = 1.0$, $T = 1.4$, and $T = 1.8$).

where $\dot{q}(x, y)$ and the boundary conditions are all chosen to give an analytical solution of

$$T(x, y) = \tanh[\mathcal{S}(-4x - 2y + 3)] + 1. \tag{14}$$

Parameter $\mathcal{S}$ controls the steepness of the gradients; when $\mathcal{S} = 5$, as for the results presented here, T changes from 10 to 90% of its maximum value over the space of 0.098 length units in a domain measuring 1×4 units, as illustrated in Fig. 9. To start the iterative solution process on the initial grid, the temperature distribution is initially guessed to be planar.

Error is analyzed via a grid-independent estimate over the two-dimensional physical domain Ω:

$$\text{Error} = \int\int_{\Omega} |T_{\text{calculated}} - T_{\text{analytic}}| \, d^2\Omega. \tag{15}$$

The calculated error does not depend on the grid itself, in the sense that point locations do not bias the result. Points which are closer together have smaller areas associated with them and thus contribute less to the overall integral than points which are farther apart.

The rectangular heated plate application demonstrates the flexibility of the LRR method and the effects of different parameter choices. In all cases, the initial grid is an equispaced 6×21 mesh. Since solution contour plots for the final grids of the various runs are too similar to be distinguished among by the human eye, accuracy is expressed via the error norm given above. No time stepping precedes the solution of the steady-state problem, and each adaption requires only a single steady-state solve. A Newton tolerance of 1×10^{-6}, difficult to attain in practical applications, is employed here to ensure that the posed systems are solved to high accuracy; any change in solution accuracy is a result of method parameter variation. Adaption terminates either when the termination criterion is satisfied or when the number of grid points exceeds 50,000. While parameter settings could have been easily chosen to produce numerical experiment runs requiring fewer than 50,000 points, such runs either would have produced obviously inadequate results (for example, with large $\varepsilon_{\text{equi}}$) or would have required so few adaptions that there would have been little timing and error data with which comparisons among the various runs could have been made. In fact, for more accurate timing data, each case is run 10 times and the computing times are averaged.

5.1.1. Effect of LRR Method Parameter Variation. Three parameters of the LRR method are now varied, one at a time, to examine their effect and purpose, beginning with $N_{\text{layer},1}$, the parameter controlling the number of layers of refined cells added around previously refined cells. Final grids produced with different values of the layering parameter $N_{\text{layer},1}$ are shown in Fig. 10, starting with the case of no layering. As the number of layers increases, the interfaces between the variously refined regions of the grids fall further and further away from the solution's high activity areas, as anticipated. In Fig. 11, a log–log plot of error as a function of the reciprocal of the smallest grid spacing, pth-order accuracy is reflected by a slope of $-p$. Therefore, it is apparent that for an adequate amount of layering (in this case, $N_{\text{layer},1} \geq 2$), the LRR method provides second-order accuracy, indicated by the slope of -2 on the plot. As predicted, the error indeed decreases as the number of layers increases. The refined region is sufficiently wide in Fig. 10c, as compared to the region of high activity, and the final error is smallest, so the parameter setting $N_{\text{layer},1} = 3$ is retained.

The second parameter to be studied is N_{smth}, the number of weight function smoothing passes. It is expected that the more the weight functions are smoothed, the more evenly

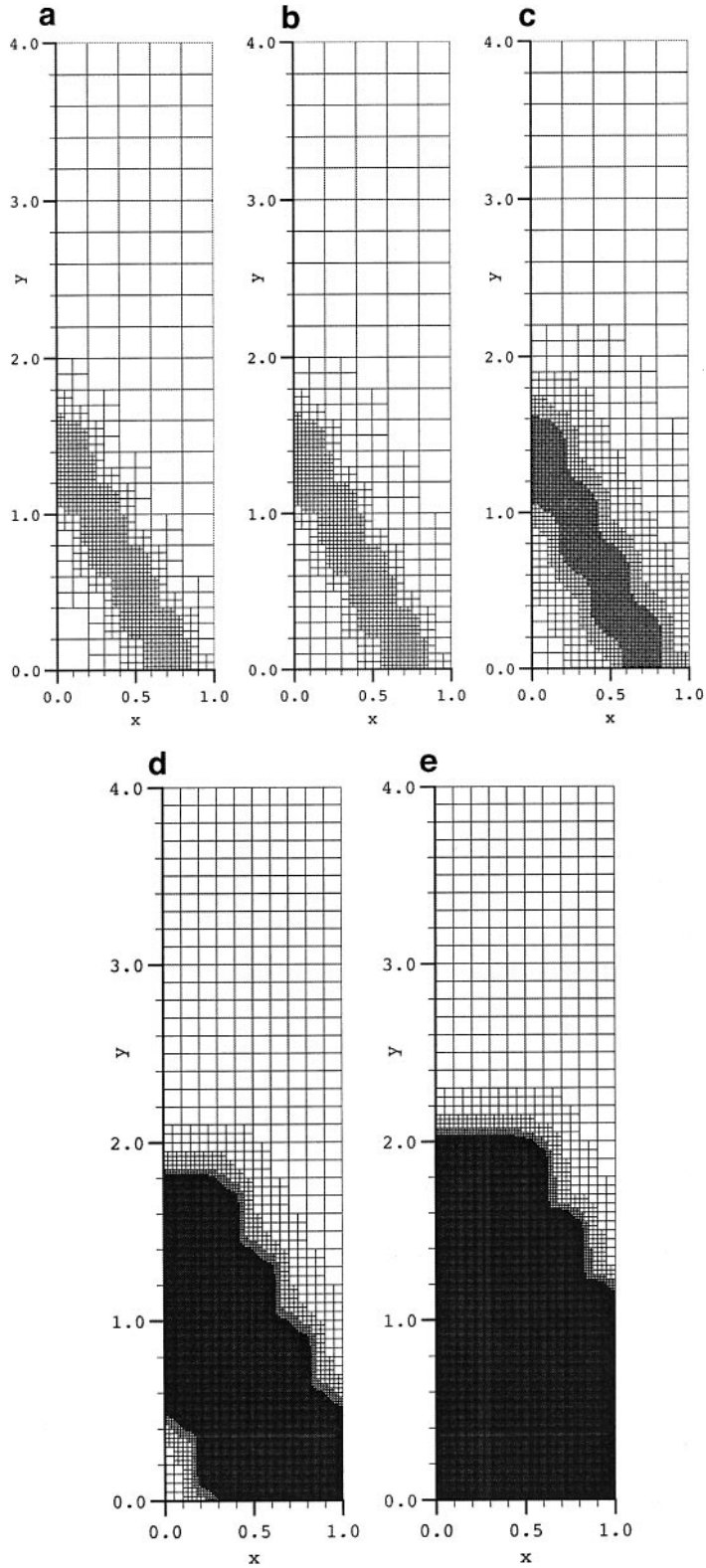

FIG. 10. Final LRR grids for the rectangular heated plate problem, with $N_{\text{layer},1} = 0$ (a); $N_{\text{layer},1} = 1$ (b); $N_{\text{layer},1} = 2$ (c); $N_{\text{layer},1} = 3$ (d); and $N_{\text{layer},1} = 4$ (e).

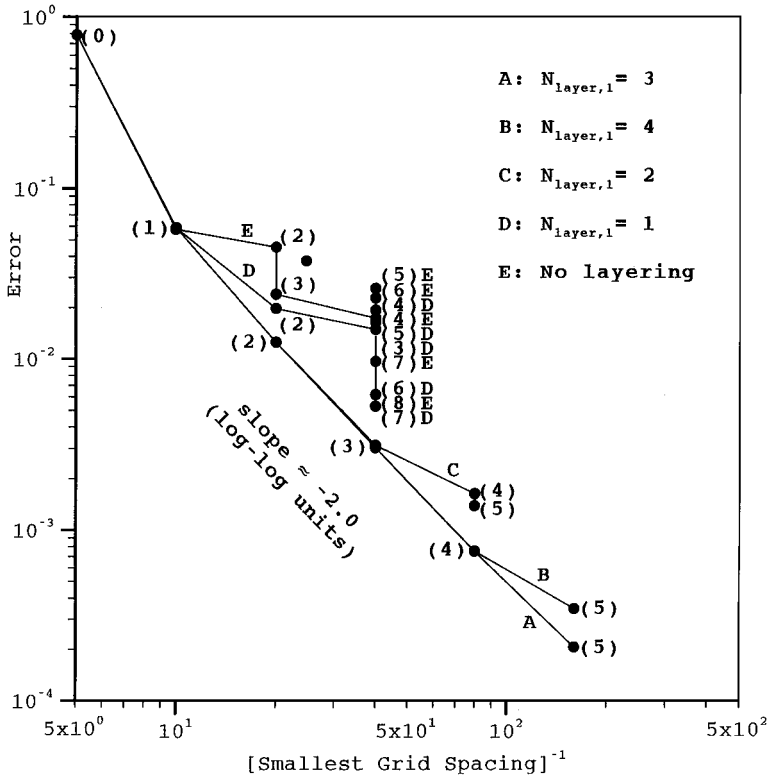

FIG. 11. Parameter study using rectangular heated plate problem: variation in the number of added refinement layers, as controlled by $N_{\text{layer},1}$. Numbers in parentheses refer to the LRR grid adaption number.

spatially distributed they should become. Thus, when they are used as an indicator for grid refinement, the regions of high gradients and curvatures should be perceived by the method as being less sharp, and, consequently, fewer points should be added. In practice, however, the number of points does not vary greatly from one trial to the next. This behavior can be explained by the current setting of $N_{\text{layer},1} = 3$, which encourages point addition and thus masks any effect of varying the number of weight function smoothing passes. This evidence indicates interaction among various method parameters; the alteration of any single parameter may or may not have a noticeable effect, depending upon other parameter settings. Figure 12 again illustrates the second-order accuracy of the LRR method and the very small differences in overall error as N_{smth} is changed. Parameter setting $N_{\text{smth}} = 10$ is retained.

The third and final aspect of the LRR method to be examined is the choice of discretizations used on the regular interior points and external boundary points, combined with the internal boundary point treatment. Not every combination is a viable one; for example, single-scale stencils at regular interior grid points can be used only in conjunction with interpolants at internal boundary points. For the current application, the number of points in a given adaption is observed to be independent of the discretization and the internal boundary point treatment, simply because no major solution differences among the methods are apparent for the first few adaptions.

Three of the five curves in Fig. 13 display results obtained with multiple-scale stencils and various internal boundary point treatments; the remainder cover single-scale stencils used with interpolants—schemes representative of those prominent in the local grid refinement

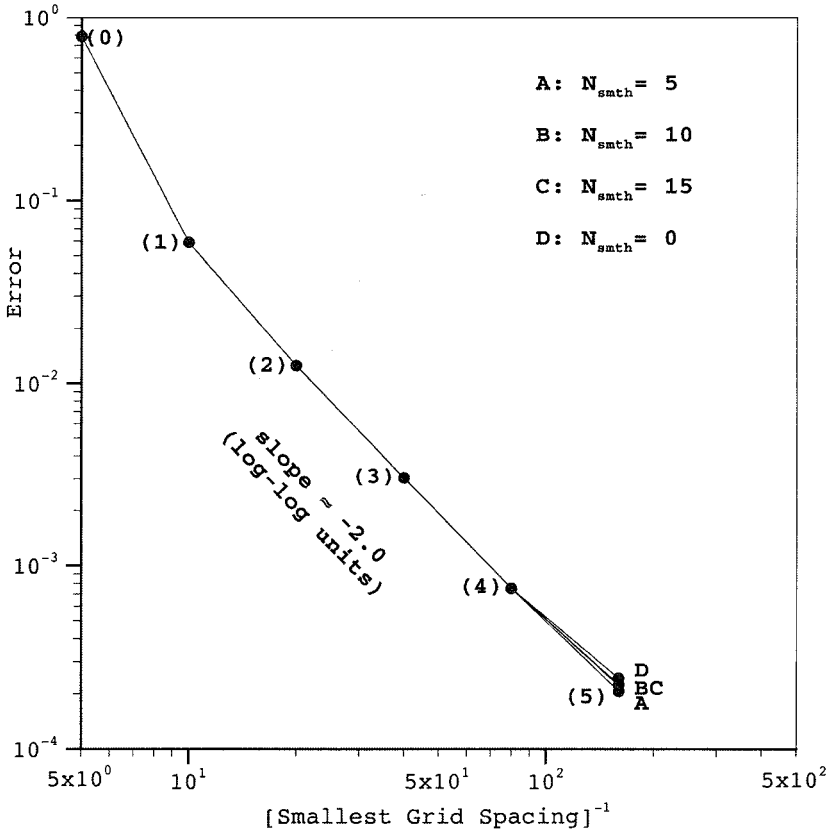

FIG. 12. Parameter study using rectangular heated plate problem: variation in the number of smoothing passes, N_{smth}. Numbers in parentheses refer to the LRR grid adaption number.

literature. Once again, the plot illustrates that the LRR method produces second-order accuracy, as shown by the -2 slope of the data, during the first four adaptions. In Adaption 5, the error data from the runs employing multiple-scale stencils in combination with either pseudo-nine-point stencils (curve A) or five-point interpolants (curve B) continue to fall on the line of slope -2. However, error measurements for the trial using multiple-scale discretizations with two-point interpolants at internal boundary points (curve C) begin to deviate from the baseline of slope -2; this behavior can be attributed to the lower order of the two-point interpolant, as compared with the five-point interpolant.

The largest deviation from the baseline is displayed by the two trials using single-scale interpolants (curves D and E). The Adaption 5 error for these trials is almost 40% larger than the Adaption 5 error for curves A and B, and the line segment connecting the Adaptions 4 and 5 data for curves D and E has an approximate slope of -1.4. These results are not surprising, since the grid spacings used in the single-scale discretizations can be as large as Δx and Δy of the base grid, resulting in poor accuracy, despite the fact that the single-scale stencils are equispaced. Another drawback of the single-scale solutions is that they require several Newton iterations, increasing their execution time, versus only one Newton iteration for the multiple-scale solutions. Thus, multiple-scale stencils are preferred and are therefore employed in all remaining LRR runs, as are pseudo-nine-point stencils at the internal boundary points.

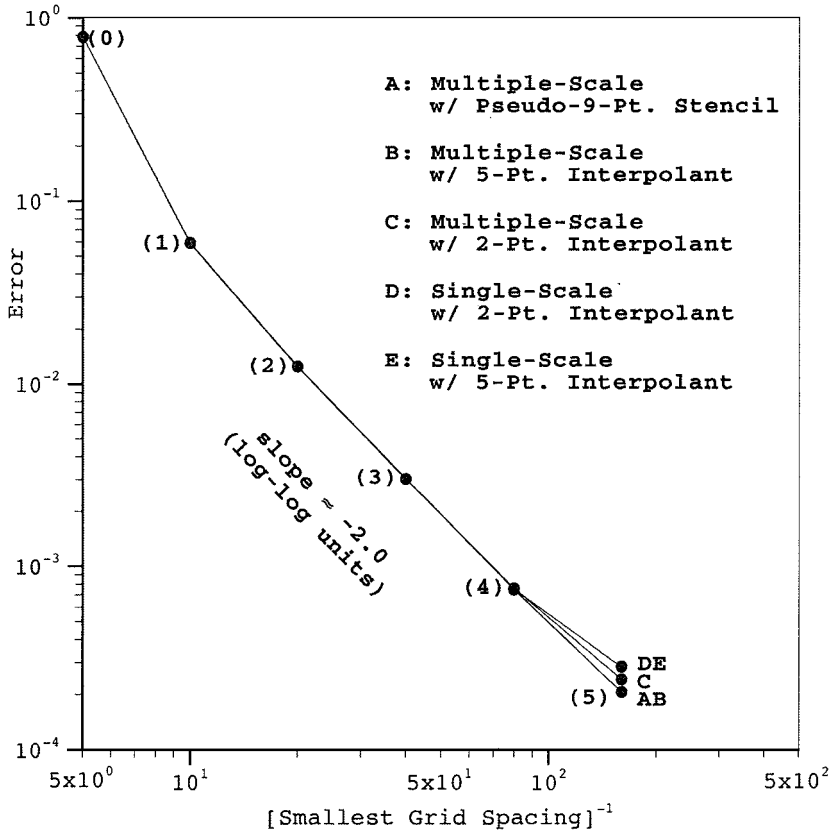

FIG. 13. Parameter study using rectangular heated plate problem: variation in discretizations and internal boundary point treatments. Numbers in parentheses refer to the LRR grid adaption number.

5.1.2. Comparison of LRR and traditional gridding methods. The base case for the LRR method is now compared with the solutions computed on two series of globally refined rectangular grids: equivalent tensor product (ETP) grids and fully refined (FR) grids. The ETP grids result from extending all grid lines of a given LRR grid out to the edges of the domain; such grids represent those available to researchers using adaptive globally refined rectangular grids. The FR grids result from using the smallest grid spacing in a given LRR grid to create a uniform, equispaced grid. Both the ETP and FR grids thus contain exactly the same refinement areas as the corresponding LRR grids, since a given LRR grid is a subset of the ETP grid, which, in turn, is a subset of the FR grid. In the ETP and FR trials, the entire solution method is the same as that for the LRR trials (i.e., Newton's method with nested BiCG-STAB linear algebra solver). The reported times are 10-run averages and include both the time required to interpolate a guess onto the new grid from the solution of the previous grid and the time to form and solve the governing equation on the new grid.

Table I contains the results from the three different gridding methods. The LRR trial's errors are of comparable magnitude to those of the ETP and FR trials, with the LRR grids having slightly smaller errors throughout. In the earlier adaptions, the LRR method requires similar solution times per point, as compared to the FR trials, while later LRR adaptions require considerably more time per point than the FR trials. The ETP trials take consistently

TABLE I
Rectangular Heated Plate: Comparison of LRR Method with Traditional Gridding Methods

Method	Adaption	N_{pts}	Error	t (s)	t/N_{pts}
LRR	0	126	0.78641	0.08	0.00063
	1	304	0.05907	0.14	0.00046
	2	906	0.01250	0.54	0.00060
	3	2,795	0.00303	1.88	0.00067
	4	9,353	0.00075	11.41	0.00122
	5	33,719	0.00021	75.60	0.00224
ETP	0	$6 \times 21 = 126$	0.78641	0.08	0.00063
	1	$11 \times 32 = 352$	0.05910	0.37	0.00105
	2	$21 \times 54 = 1{,}134$	0.01251	0.98	0.00086
	3	$41 \times 93 = 3{,}813$	0.00304	3.71	0.00097
	4	$81 \times 167 = 13{,}527$	0.00076	13.23	0.00098
	5	$161 \times 321 = 51{,}681$	0.00023	50.78	0.00098
FR	0	$6 \times 21 = 126$	0.78641	0.08	0.00063
	1	$11 \times 41 = 451$	0.05910	0.28	0.00062
	2	$21 \times 81 = 1{,}701$	0.01251	1.14	0.00067
	3	$41 \times 161 = 6{,}601$	0.00304	5.27	0.00080
	4	$81 \times 321 = 26{,}001$	0.00077	23.40	0.00090
	5	$161 \times 641 = 103{,}201$	0.00026	87.60	0.00085

longer per point than the FR trials, most likely because the ETP grids contain cells of extreme aspect ratios which worsen the conditioning of the Jacobian matrix, leading to slower convergence. The overall execution time is, of course, smallest for each LRR adaption as compared to the corresponding ETP and FR solutions, because of the small number of grid points in the LRR trials. The Adaption 5 LRR grid, for example, contains 0.65 and 0.33 times as many points as its ETP and FR counterparts, respectively, but produces no significant difference in accuracy.

The implementations of LRR, ETP, and FR methods require the following storage, expressed in terms of the number of dependent variables per grid point and the total number of points:

$$\begin{aligned} \text{Storage}_{\text{LRR}} &= \left(9N_{\text{dep}}^2 + 21N_{\text{dep}} + 26\right) N_{\text{pts}} \\ \text{Storage}_{\text{ETP or FR}} &= \left(9N_{\text{dep}}^2 + 17N_{\text{dep}} + 10.375\right) N_{\text{pts}}. \end{aligned} \tag{16}$$

In each case, the leading term represents the memory required for the Newton's method Jacobian matrix, and when there are more than one or two unknowns per point, this term dominates. For the current problem, $N_{\text{dep}} = 1$, resulting in

$$\begin{aligned} \text{Storage}_{\text{LRR}} &= 56N_{\text{pts}} \\ \text{Storage}_{\text{ETP or FR}} &= 36.375N_{\text{pts}}. \end{aligned} \tag{17}$$

Therefore, when an LRR grid contains fewer than 0.65 times as many points as the corresponding ETP or FR grid (as is the case for some of the above trials), the solution process

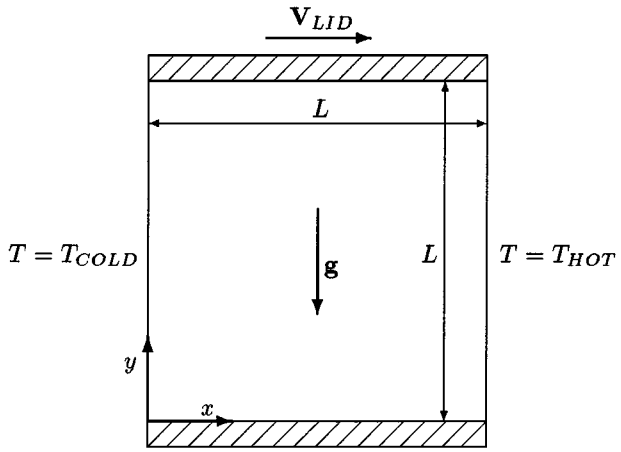

FIG. 14. Physical configuration for the square cavity problem.

on the LRR grid requires less memory overall. Significantly greater storage savings for the LRR method are demonstrated via the remaining application problems.

5.2. *Thermally and Dynamically Driven Square Cavity Flow*

The second problem to be investigated is that of flow within a differentially heated square cavity, with either a stationary or a moving lid. This problem has been widely studied for several different combinations of Grashof, Prandtl, and Reynolds numbers (see, for example, [15, 22, 33, 36], and references therein). The cavity, shown in Fig. 14, is filled with air (Prandtl number $\mathrm{Pr} = 0.72$), and it has two insulated horizontal walls and two uninsulated vertical walls. The left and right vertical walls are maintained at low and high temperatures, respectively, which will set into motion a counterclockwise circulation, as long as the lid velocity is zero. When the lid is moving to the right, the thermally driven circulation must compete with the dynamically forced clockwise circulation.

The steady-state flow is governed by four coupled nonlinear equations, involving unknowns of horizontal and vertical velocity (u and v), vorticity ω, and temperature T; the derivation can be found in [22]. Kinematic viscosity ν and thermal conductivity λ are assumed constant. The equations are nondimensionalized such that the domain becomes a unit square and the temperature varies from zero to one, producing

$$\begin{aligned} &\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -\frac{\partial \omega}{\partial y}, \qquad \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = \frac{\partial \omega}{\partial x}, \\ &\frac{\partial}{\partial x}(u\omega) + \frac{\partial}{\partial y}(v\omega) = \frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} + \mathrm{Gr}\frac{\partial T}{\partial x}, \\ &\frac{\partial}{\partial x}(uT) + \frac{\partial}{\partial y}(vT) = \frac{1}{\mathrm{Pr}}\left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right], \end{aligned} \tag{18}$$

where

$$\mathrm{Gr} = \frac{g\beta(T_{\mathrm{HOT}} - T_{\mathrm{COLD}})L^3}{\nu^2}, \qquad \mathrm{Pr} = \frac{\rho \nu c_p}{\lambda}, \qquad \text{and} \qquad \mathrm{Re} = \frac{\rho V_{\mathrm{LID}} L}{\nu}.$$

Variables u, v, ω, and T are now dimensionless. The boundary conditions are simple: no-slip conditions on all walls; definition of vorticity invoked along the entire boundary; and temperatures are either prescribed directly (vertical walls) or else the fluxes vanish (horizontal walls). Physical intuition indicates counterclockwise flow, at least when the lid is stationary, which is reflected in the initial guess for the velocity fields, as follows:

$$\begin{aligned} u &= \sqrt{2}\,\mathrm{Gr}^{1/3}\sin(\pi x)\cos(\pi y) \\ v &= -\sqrt{2}\,\mathrm{Gr}^{1/3}\cos(\pi x)\sin(\pi y). \end{aligned} \tag{19}$$

Vorticity is initially chosen to be zero, and temperature is guessed to vary linearly between the vertical boundaries.

This second application problem does not have an analytical solution, but it does provide a more realistic application for the gridding methods. The effects of adapting based on different variables are demonstrated: first, adaption is done based on gradients of ω only, then on gradients of ω and T, and finally on gradients of all four variables (for brevity, the latter results are not shown). For all of the following results, many of the method parameter settings are the same as those for the previous application problem, with a few exceptions. The number of weight function smoothing passes N_{smth} is set to 2, because the gradients are not as sharp as those for the previous problem. Now, a maximum of five adaptions is allowed, although the maximum number of grid points (for LRR) is still 50,000. For all cases, the initial grid is an equispaced 41×41 mesh, with $\Delta x = \Delta y = 0.025$. On each adaptive grid, the solution process consists of 150 adaptively chosen time steps, followed by one steady-state solve. Although the time steps are not critical to achieving convergence in several of the adaptions, standardization of the method is necessary in order to compare equal amounts of computation among the methods and from one adaption to the next. To lower execution times, the Jacobian matrix is evaluated only at every 10th time step.

5.2.1. Comparison of LRR and traditional gridding methods for the case of natural convection: $Ra = 10^4$. For the problem of thermally driven flow in a square cavity with a stationary lid ($\mathrm{Re} = 0$), the solution strategy includes a single steady-state solve on the initial grid, and the Newton tolerance for all cases is 1×10^{-5}. There can be no error measurement since the analytical solution is not known. However, a chosen physical quantity, such as the maximum absolute value of vorticity, denoted by $|\omega|_{\max}$, can be monitored for each set of results. The Rayleigh number ($\mathrm{Ra} = \mathrm{Gr} \cdot \mathrm{Pr}$) is chosen as 10^4, which allows comparison with published data [15, 22]; note that some data [15] have been adjusted so that they are in accordance with the current nondimensionalization.

Figure 15 displays profiles of u and v along the cavity vertical and horizontal midplanes, for an LRR Adaption 5 grid and the corresponding ETP and FR grids, along with a benchmark solution [15] and data from a vorticity–velocity formulation [22]. The latter two are both calculated on equispaced rectangular grids. While the LRR data shown here are from adapting based on ω, there are no visible differences between these solutions and those obtained on any of the final adapted grids in which different adaption variables were chosen. The figure shows excellent agreement among the different datasets. As expected, a counterclockwise flow pattern has developed.

Figure 16 shows vorticity contours calculated on LRR Adaption 5 (adaption based on ω); it is clear that refinement indeed occurs in regions of rapid change of ω. That the flow is primarily circulating, with boundary layers along the walls, is evidenced by the large

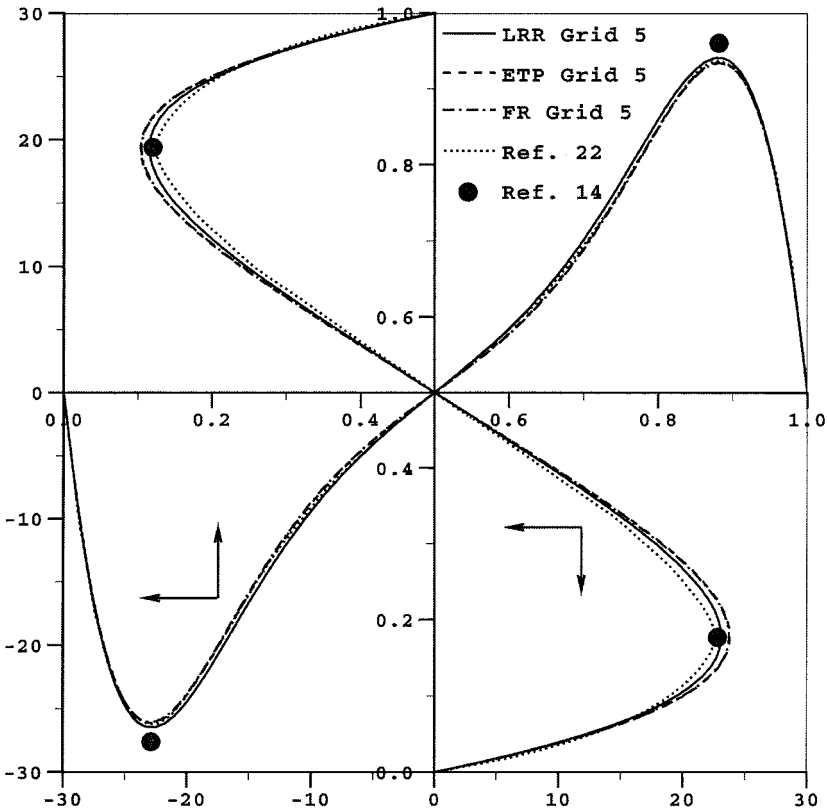

FIG. 15. Profiles of u and v along the cavity vertical and horizontal mid-planes ($x = 0.5$ and $y = 0.5$, respectively) for $\mathrm{Ra} = 10^4$ (natural convection).

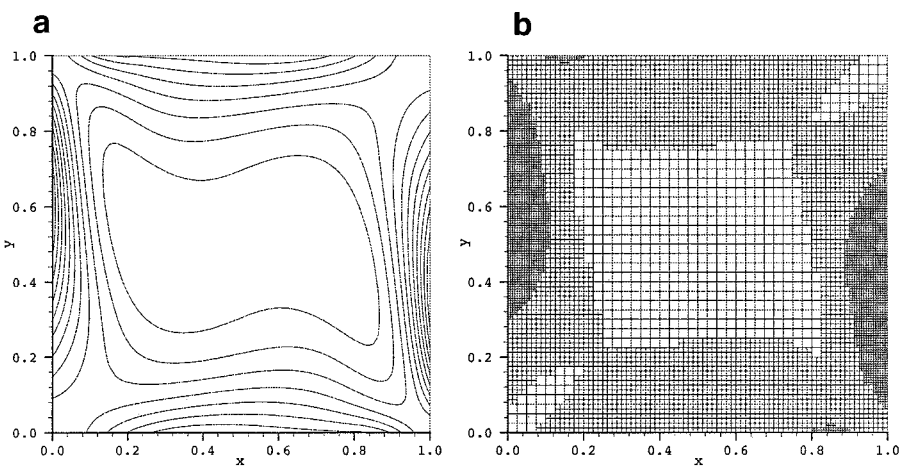

FIG. 16. Numerical isopleths of vorticity (ω) for $\mathrm{Ra} = 10^4$ (natural convection) (a), calculated on LRR Adaption 5, formed by adapting based on ω (b).

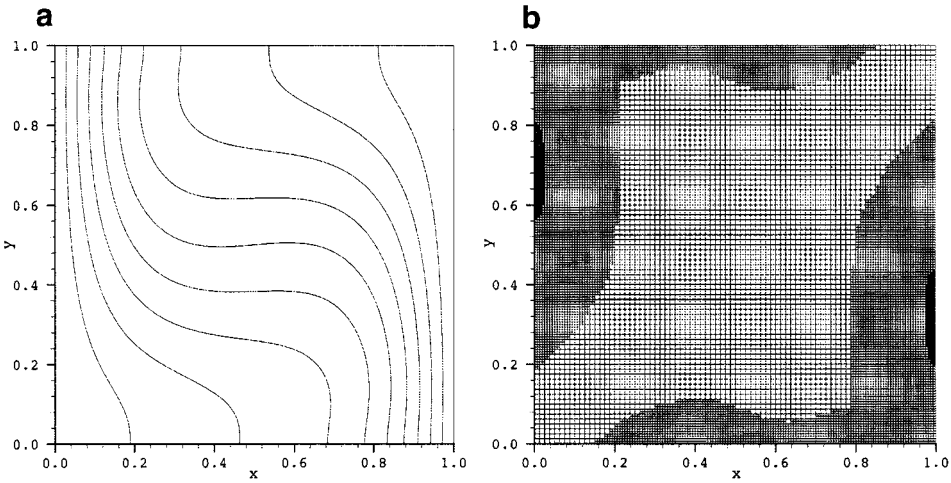

FIG. 17. Numerical isotherms for $\mathrm{Ra} = 10^4$ (natural convection) (a), calculated on LRR Adaption 5, formed by adapting based on ω and T (b).

negative vortex appearing near the center and the smaller positive vortices along each of the four walls. Figure 17 displays isotherms obtained on LRR Adaption 5 (adaption based on ω and T), along with the grid itself. The grid undergoes refinement along the walls to resolve the boundary layers, and additional refinement occurs in regions where the vorticity and/or temperature is rapidly changing.

Table II presents the results obtained by adapting based on ω using the LRR method, as well as results on corresponding ETP and FR grids. Immediately apparent is the fact that the LRR $|\omega|_{\max}$ values are extremely close to the ETP and FR values: within 2% for Adaption 1 and within 1% for all subsequent adaptions. This high degree of accuracy is

TABLE II
Square Cavity with Ra = 10^4 (Natural Convection)
Adaption on ω: Comparison of LRR Method with Traditional Gridding Methods

Method	Adaption	N_{pts}	$\lvert\omega\rvert_{\max}$	t (s)	t/N_{pts}
LRR	0	1,681	424.5	44	0.026
	1	3,063	508.3	333	0.109
	2	4,949	547.1	767	0.155
	3	5,919	547.8	671	0.113
	4	6,585	547.6	686	0.104
	5	7,143	545.6	830	0.116
ETP	0	$41 \times 41 = 1{,}681$	424.5	44	0.026
	1	$65 \times 81 = 5{,}265$	500.2	592	0.112
	2	$95 \times 139 = 13{,}205$	552.6	2358	0.179
	3	$95 \times 145 = 13{,}775$	552.5	794	0.058
	4	$97 \times 149 = 14{,}453$	552.7	1022	0.071
	5	$113 \times 155 = 17{,}515$	544.0	2545	0.145
FR	0	$41 \times 41 = 1{,}681$	424.5	44	0.026
	1	$81 \times 81 = 6{,}561$	498.9	706	0.108
	2, 3, 4, 5	$161 \times 161 = 25{,}921$	543.2	3618	0.140

TABLE III
Square Cavity with Ra = 10^4 (Natural Convection)
Adaption on ω and T: Comparison of LRR Method with Traditional Gridding Methods

Method	Adaption	N_{pts}	$\|\omega\|_{max}$	t (s)	t/N_{pts}
LRR	0	1,681	424.5	44	0.026
	1	3,181	508.6	347	0.109
	2	5,371	548.8	816	0.152
	3	8,189	550.9	994	0.121
	4	11,123	546.2	1,473	0.132
	5	15,299	568.6	2,660	0.174
ETP	0	$41 \times 41 = 1{,}681$	424.5	44	0.026
	1	$65 \times 81 = 5{,}265$	500.2	617	0.117
	2	$95 \times 141 = 13{,}395$	552.6	2,412	0.180
	3	$97 \times 161 = 15{,}617$	542.2	2,026	0.130
	4	$153 \times 161 = 24{,}633$	543.4	2,263	0.092
	5	$169 \times 239 = 45{,}461$	572.3	8,853	0.195
FR	0	$41 \times 41 = 1{,}681$	424.5	44	0.026
	1	$81 \times 81 = 6{,}561$	498.9	706	0.108
	2, 3, 4	$161 \times 161 = 25{,}921$	543.2	3,618	0.140
	5	$321 \times 321 = 103{,}041$	567.4	20,851	0.202

achieved on LRR grids with 0.58, 0.37, 0.43, 0.46, and 0.41 as many points as the traditional ETP grids, for Adaptions 1 through 5, respectively; the LRR grids have 0.47, 0.19, 0.23, 0.25, and 0.28 as many points as the equispaced FR grids. For any given adaption, the LRR *per-point* solution times are of the same order as those for the corresponding ETP and FR grids, with the only exceptions occurring when the ETP grids do not change greatly from one adaption to the next (and thus those solutions converge more quickly). The overall LRR solution times are much shorter than those for the ETP and FR runs.

When ω and T are the adaption variables, as in Table III, there is again very little difference among the $|\omega|_{max}$ values obtained using the LRR, ETP, and FR grids. For Adaptions 1 through 3, the LRR values are within 2% of those obtained on the corresponding ETP and FR grids, and for Adaptions 4 and 5 the values differ by less than 1%. At the same time, the LRR grids have significantly fewer points than the other grids; for example, the Adaption 5 grid has 0.34 times as many points as its ETP counterpart and 0.15 times as many points as its FR counterpart. Because the LRR per-point solution times are comparable to, or sometimes even faster than, the per-point solution times on the ETP and FR grids, the LRR method finds the solution much more quickly than the other methods. Finally, the LRR grids (and thus, the ETP and FR grids) contain more points than when ω was the only adaption variable, which is to be expected.

Adaption has also been performed on all variables simultaneously, and those results display characteristics similar to the above (a very accurate representation of $|\omega|_{max}$, while using a fraction of the number of points used in the corresponding ETP and FR grids). Regardless of the adaption variable(s) chosen, plots of the final solutions are indistinguishable from one another. Memory requirements are examined at the end of the next section.

5.2.2. Comparison of LRR and traditional gridding methods for the case of mixed convection: $Ra = 10^4$ and $Gr/Re^2 = 0.5$. Because the problem of thermally *and* dynamically

driven flow in a square cavity is more challenging than the previous one, the solution strategy now includes 150 time steps on the initial grid, as well as on all subsequent grids. The reader is reminded that these time steps are not actually needed for convergence on some of the earlier adaptions, but that valid comparisons require standardization of the amounts of calculation. The Newton tolerance is 1×10^{-4}, except for some ETP computations (indicated by table footnotes) which are solved to 2×10^{-4}; these would have required further time stepping in order to have met the original tolerance. As in the natural convection case, there is no analytical solution, but a physical quantity (here, $v_{\max}$, since $|\omega|_{\max}$ is infinite) is used to gauge solution accuracy. The choices of $\mathrm{Ra} = 10^4$ and $\mathrm{Gr}/\mathrm{Re}^2 = 0.5$ ($\mathrm{Re} = 166.667$) permit direct comparison with published data [22].

Some minor differences exist among the solutions found on grids generated via different adaption variables, but the general shapes of the solution profiles are the same. Figure 18 displays the profile of u along the cavity's vertical mid-plane, using results from LRR Adaption 4 based on ω and T, solutions from the corresponding ETP and FR runs, as well as data from [22]. (Adaption 4 results are illustrated instead of Adaption 5 results, simply because the FR calculation corresponding to LRR Adaption 5 exceeds the 2 GB RAM limit.) The datasets show good agreement, given the sharpness of the gradients as compared to the natural convection case. It is evident that the counterclockwise circulation generated by buoyancy effects of the differentially heated vertical walls survives only in the lower part of the cavity, while the rapid movement of the top wall controls flow in the upper part.

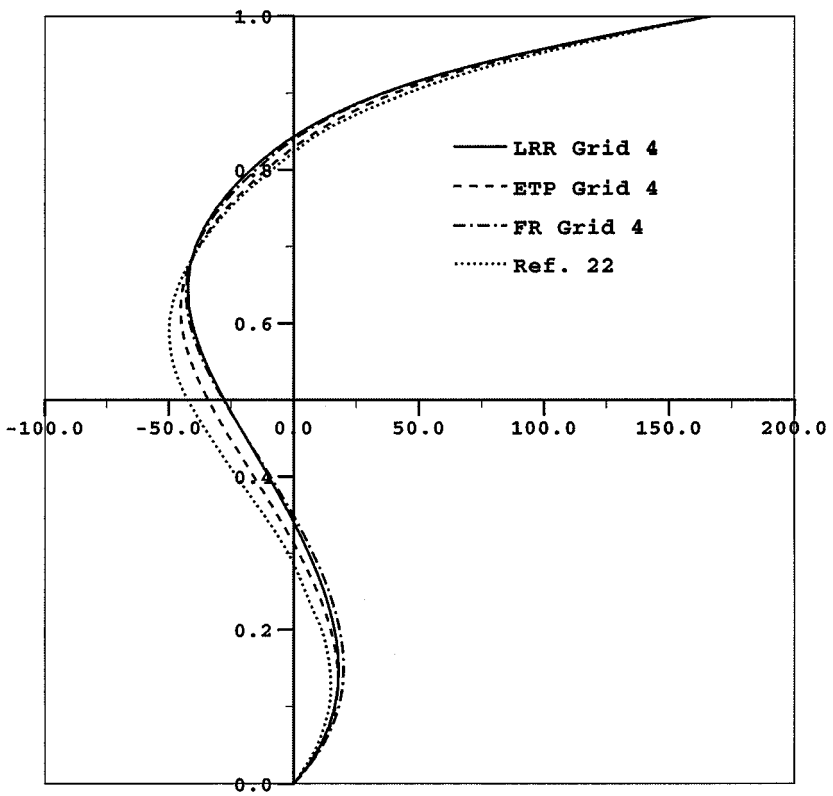

FIG. 18. Profiles of u along the cavity vertical mid-plane ($x = 0.5$) for $\mathrm{Ra} = 10^4$ and $\mathrm{Gr}/\mathrm{Re}^2 = 0.5$.

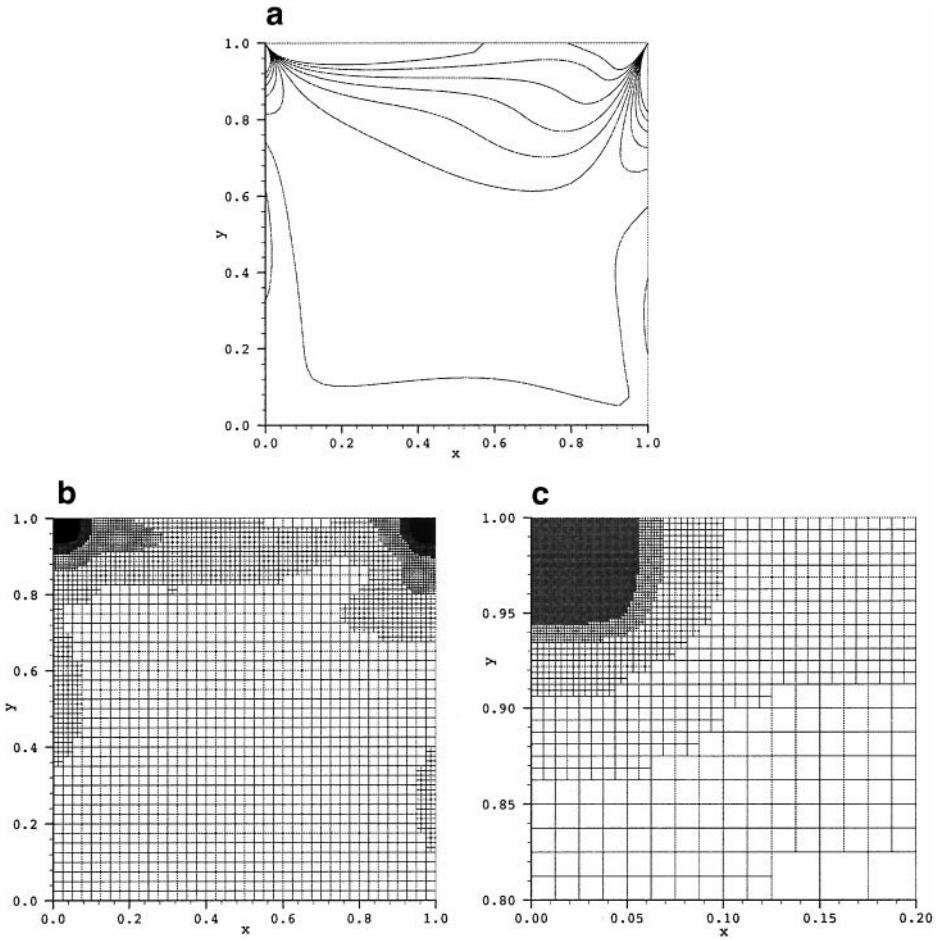

FIG. 19. Numerical isopleths of vorticity (ω) for $\mathrm{Ra} = 10^4$ and $\mathrm{Gr}/\mathrm{Re}^2 = 0.5$ (a), calculated on LRR Adaption 5, formed by adapting based on ω (b). Close-up of upper left corner of domain (c) displays base grid plus five levels of refinement.

Figure 19 displays vorticity contours calculated on LRR Adaption 5 (adaption based on ω), along with the grid itself. As expected, the grid is highly refined in the upper left and right corners, and the close-up in (c) shows the presence of the base cell size plus five refinement levels. Figure 20 shows isotherms calculated on LRR Adaption 5 (adaption based on ω and T), and effect of the cavity lid motion is indeed apparent. The grid, shown in (b), displays refinement in the regions of high ω and T activity, and it again comprises the base grid plus five levels of refinement. Although not shown here, results from adapting based on all four dependent variables display similar trends.

Table IV presents the results from LRR adaptions performed based on ω only, along with results from corresponding ETP and FR grids. Excellent agreement (within 1% from Adaption 2 on) is observed among the $v_{\max}$ value on each LRR grid and its ETP and FR counterparts, but the values are ever-increasing, symptomatic of the fact that the adaption process is not complete by Adaption 5. The LRR Adaption 5 grid contains roughly one-third as many points as its ETP counterpart, and for Adaptions 1 through 5, respectively, the LRR grids contain only 0.38, 0.14, 0.05, 0.02, and 0.01 times as many points as the corresponding

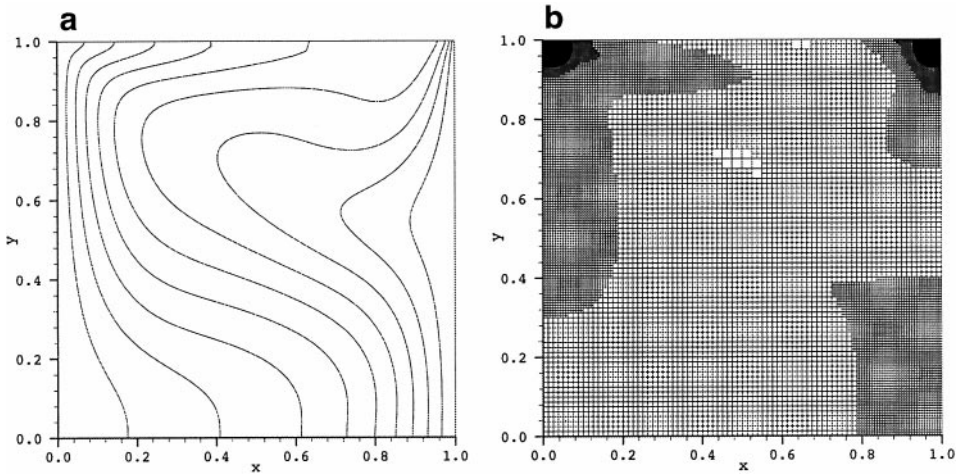

FIG. 20. Numerical isotherms for $\mathrm{Ra} = 10^4$ and $\mathrm{Gr}/\mathrm{Re}^2 = 0.5$ (a), calculated on LRR Adaption 5, formed by adapting based on ω and T (b).

equispaced FR grids. Comparisons of timings with ETP grids for later adaptions are not valid because of the looser ETP tolerance, which requires less computational effort. The difficulty experienced by these ETP runs in meeting the original 1×10^{-4} tolerance most likely stems from cells of extremely high or low aspect ratio, present in the ETP grids only and leading to poorly conditioned Jacobian matrices. LRR and FR per-point timings are

TABLE IV
Square Cavity with $\mathrm{Ra} = 10^4$ and $\mathrm{Gr}/\mathrm{Re}^2 = 0.5$
Adaption on ω: Comparison of LRR Method with Traditional Gridding Methods

Method	Adaption	N_{pts}	$v_{\max}$	t (s)	t/N_{pts}
LRR	0	1,681	22.07	128	0.076
	1	2,517	21.34	284	0.113
	2	3,593	29.26	333	0.093
	3	5,074	35.57	540	0.106
	4	7,792	40.72	1,002	0.129
	5	15,451	45.21	3,895	0.252
ETP	0	$41 \times 41 = 1{,}681$	22.07	128	0.076
	1	$77 \times 55 = 4{,}004$	23.15	429	0.107
	2	$112 \times 72 = 8{,}064$	29.13	825	0.102
	3	$146 \times 98 = 14{,}308$	35.73	1,626	0.114
	4	$191 \times 126 = 24{,}066$	41.22[a]	2,776[a]	0.115[a]
	5	$265 \times 167 = 44{,}255$	45.59[a]	5,562[a]	0.126[a]
FR	0	$41 \times 41 = 1{,}681$	22.07	128	0.076
	1	$81 \times 81 = 6{,}561$	22.06	663	0.101
	2	$161 \times 161 = 25{,}921$	29.15	2,637	0.102
	3	$321 \times 321 = 103{,}041$	35.76	14,770	0.143
	4	$641 \times 641 = 410{,}881$	41.17	26,608	0.065
	5	$1281 \times 1281 = 1640{,}961$	—[b]	—[b]	—[b]

[a] Converged to a tolerance of only 2×10^{-4}.
[b] Size of problem exceeded 2 GB RAM limit.

TABLE V
Square Cavity with Ra = 10^4 and Gr/Re^2 = 0.5
Adaption on ω and T: Comparison of LRR Method with Traditional Gridding Methods

Method	Adaption	N_{pts}	v_{max}	t (s)	t/N_{pts}
LRR	0	1,681	22.07	128	0.076
	1	3,275	23.77	372	0.114
	2	4,881	29.22	455	0.093
	3	7,833	35.62	830	0.106
	4	13,403	40.81	1,866	0.139
	5	24,959	45.36	4,597	0.184
ETP	0	$41 \times 41 = 1{,}681$	22.07	128	0.076
	1	$77 \times 78 = 6{,}006$	22.04	626	0.104
	2	$117 \times 97 = 11{,}349$	29.16	1,077	0.095
	3	$157 \times 118 = 18{,}526$	35.75[a]	1,785[a]	0.096[a]
	4	$212 \times 187 = 39{,}644$	41.20[a]	4,302[a]	0.109[a]
	5	$308 \times 244 = 75{,}152$	45.79[a]	10,145[a]	0.135[a]
FR	0	$41 \times 41 = 1{,}681$	22.07	128	0.076
	1	$81 \times 81 = 6{,}561$	22.06	663	0.101
	2	$161 \times 161 = 25{,}921$	29.15	2,637	0.102
	3	$321 \times 321 = 103{,}041$	35.76	14,770	0.143
	4	$641 \times 641 = 410{,}881$	41.17	26,608	0.065
	5	$1281 \times 1281 = 1{,}640{,}961$	—[b]	—[b]	—[b]

[a] Converged to a tolerance of only 2×10^{-4}.
[b] Size of problem exceeded 2 GB RAM limit.

roughly comparable in all adaptions except Adaption 4, in which the LRR per-point timing is nearly twice as large. Comparison of LRR Adaption 5 results cannot be made with the corresponding FR grid, since the FR problem is too large for the available computational resources. Overall, the LRR runs consistently require a small fraction of the CPU time of the ETP and FR runs, while maintaining accuracy to within 1%, as measured by values of v_{max}.

When ω and T are the adaption variables, as in Table V, the values of v_{max} produced using the LRR method again mirror those obtained on the corresponding ETP and FR grids, to within 1% from Adaption 2 on. (The continual increase of v_{max} with grid number again reflects the artificially imposed termination of the adaption process after the fifth grid.) This excellent agreement is achieved on LRR grids with 0.50, 0.19, 0.08, 0.03, and 0.02 times as many points as their FR counterparts, with much shorter overall solution times for LRR. As before, the FR counterpart of LRR Adaption 5 is too large for the available 2 GB RAM workstation. Also, because of ETP convergence problems, timing comparisons can be made only for the adaptions in which the tolerances are the same among LRR, ETP, and FR; in these cases, the per-point solution times are comparable. In general, the LRR method consumes much less CPU time than the other methods; for example, LRR Adaption 4 takes 0.07 as much time as its equispaced FR counterpart.

The storage amounts needed for each of the methods, presented in Eq. (18), are written in terms of N_{dep}, the number of dependent variables per point, and N_{pts}, the number of points. When $N_{dep} = 4$, as in the square cavity problem (for either mixed or natural convection), the requirements simplify to

$$\begin{aligned} \text{Storage}_{\text{LRR}} &= 254 N_{pts} \\ \text{Storage}_{\text{ETP or FR}} &= 222.375 N_{pts}, \end{aligned} \tag{20}$$

which do not differ as widely as in the rectangular heated plate problem. Whenever an LRR grid has fewer than 0.88 times as many points as its ETP or FR counterparts, the LRR solution process consumes less memory than the ETP or FR solution. Therefore, *all* LRR solutions presented above for the two types of square cavity flow not only consume much less CPU time, but LRR memory usage ranges from approximately 50% down to, in some cases, 1% that of the traditional gridding methods, as represented by the ETP and FR grids.

5.3. *Rich Axisymmetric Laminar Bunsen Flame with Complex Chemistry*

The third problem to which the LRR solution-adaptive gridding method is applied is that of a rich full-chemistry axisymmetric laminar Bunsen flame, which is assumed to have reached a steady state at atmospheric pressure. Like the square cavity flow, this third application does not have an analytical solution, but it is again a realistic problem and poses a difficult challenge for adaptive gridding techniques. In general, the major species mass fraction gradients in axisymmetric laminar *premixed* flames are an order of magnitude greater than those in axisymmetric laminar *diffusion* flames. Moreover, the *minor* species gradients in premixed flames are larger by yet another order of magnitude. The sharpness of these gradients, as well as the fact that they all occur in the same spatial region (i.e., near the flame front), make the axisymmetric laminar Bunsen flame an ideal subject for adaptive gridding.

The burner consists of a central jet, from which a homogeneous methane–air mixture issues, and a surrounding concentric jet, from which pure air flows. The composition of the methane–air mixture is 11.5 molar percentage CH_4 and 88.5 molar percentage air, which results in an overall equivalence ratio of $\Phi = 1.243$. Because the fuel-to-oxidizer ratio is rich, the premixed flame which forms atop the coflowing jets (often referred to as a "Bunsen cone" [30]) is accompanied by a diffusion flame "halo" which forms further downstream. As illustrated in Fig. 21, the premixed fuel–air jet's inner radius is $r_1 = 0.5$ cm

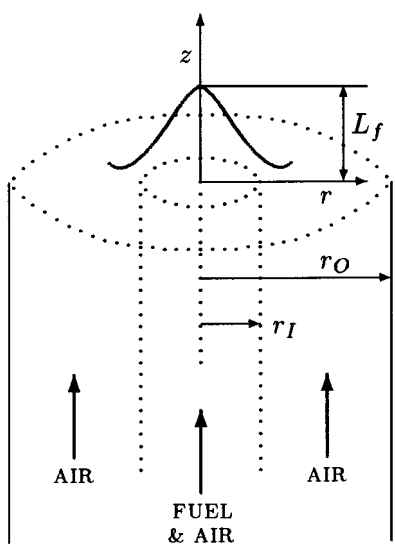

FIG. 21. Physical configuration for the axisymmetric Bunsen flame.

and its wall thickness is $w_{\mathrm{JET}} = 0.05$ cm. The coflowing jet's inner radius is $r_{\mathrm{O}} = 3.0$ cm, marking the radial extent of the computational domain. Computations are performed in a two-dimensional domain with axial boundaries at $z = 0$ and $z = 25$ cm; when choosing boundary conditions, the latter can be considered "infinitely" far from the flame, since the flame length does not exceed 1 or 2 cm.

The velocity profile across the inner jet exit is parabolic, with the average axial velocity $v_{z,\mathrm{I}} = 50$ cm s^{-1}. The v_z profile across the outer jet exit increases from an innermost minimum of zero to $v_{z,\mathrm{O}} = 50$ cm s^{-1}, simulating a plug flow with a thick inner boundary layer [5]. Across the entire burner surface, radial velocity $v_r = 0$ cm s^{-1}, while *both* v_r and v_z vanish across the thickness of the jet wall. The Reynolds number within the inner jet is approximately 494. The present model employs a C_1 chemical mechanism involving 16 species and 46 reactions [43]. All thermodynamic, chemical, and transport properties are evaluated using subroutine libraries [23]. Forward reaction rate constants are determined from modified Arrhenius expressions containing exponential temperature dependence, and reverse rate constants are calculated from the corresponding forward rate constants and the equilibrium constants [29]. The present model also includes an optically thin radiation submodel [18, 25, 26].

As detailed in [4, 5], the governing equations are formulated via a vorticity–velocity approach [19], in which the pressure gradient is eliminated by taking the curl of the momentum equation. Advantages of this formulation, as well as further examples of its application to combustion problems, may be found in [4, 5, 19] and the references therein. The governing equations, not repeated here, are a set of 20 highly nonlinear, strongly coupled, elliptic partial differential equations, originally derived from the conservation equations for mass, momentum, energy, and individual species mass. They involve 20 dependent variables at each grid point: radial velocity v_r; axial velocity v_z; vorticity ω; temperature T; and mass fractions Y of the 16 species. The boundary conditions, aside from the higher v_z values for the inner and outer jets, are the same as those for the rich flame in [5]. Because of the governing equations' extreme nonlinearity, the full-chemistry solution on the starting grid is formed with the aid of a one-step chemistry starting estimate and a time-relaxation process (see [4, 5, 41]). Execution times for all cases computed here *begin* from the converged starting-grid solution, and all are normalized by the time taken to reconverge initially on that solution (i.e., the Jacobian evaluation time).

For this application problem, the effects of adapting based on different variables are displayed: first, adaption is performed based on gradients of $Y_{\mathrm{CH_4}}$ only (sharp gradients of $Y_{\mathrm{CH_4}}$ occur at the premixed flame front), then on gradients of $Y_{\mathrm{CH_4}}$ and T, and finally on gradients of all 20 variables (for brevity, the latter results are not shown). Many of the parameter settings are retained from the previous problems, with the following differences. Because of the sharp gradients and the tendency of the flame front to migrate (during the earlier adaptions), N_{smth} is reset to 10 and $N_{\mathrm{layer},1}$ is set to 5. For the 2 GB RAM IBM RS/6000-590 workstation used, the limiting number of grid points for this 20-variable problem is 62,000. For all cases, the initial grid is a nonequispaced tensor product mesh of size 58×70. Knowledge that a flame will form somewhere above the inner jet is used to impose a mesh which is uniformly spaced with $\Delta r = 0.02$ cm for $0.00 \le r \le 0.80$ cm, with increasingly larger spacing for $0.80 \le r \le 3.00$ cm; it is also uniformly spaced with $\Delta z = 0.02$ cm from $z = 0.00$ to $z = 0.80$ cm, with wider spacing for $0.80 \le z \le 25.00$ cm. (A small enough spacing is necessary to sustain the flame, but an equispaced FR mesh cannot be used because it would contain too many points.)

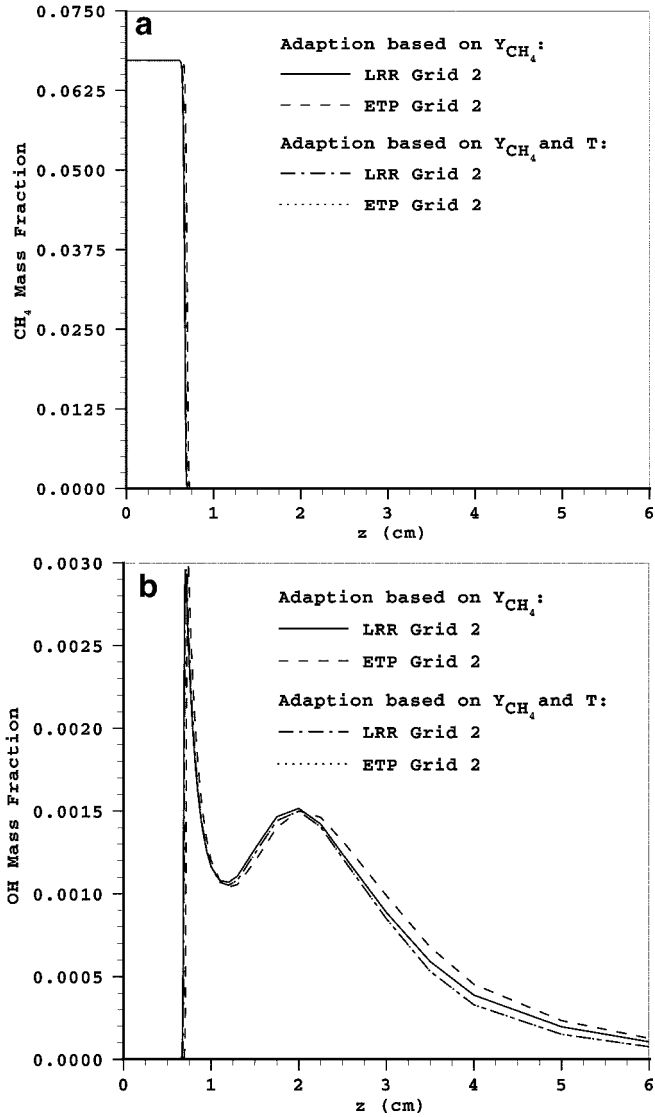

FIG. 22. Along the axis of symmetry, Y_{CH_4} (a) and Y_{OH} (b) as functions of distance from the burner surface, calculated on various grids, for the axisymmetric Bunsen flame. Only a portion of the domain is shown.

For each noninitial grid, the solution iterate is advanced in time via 200 adaptively chosen time steps, beginning with $\Delta t = 1 \times 10^{-8}$. As before, these time steps are not *required* on every grid to ensure convergence, but they are taken anyway to standardize the amount of computation and validate comparisons. In practice, the LRR grids usually require only 50 to 100 time steps to bring the solution iterate into the convergence domain of the steady-state Newton solve, but the ETP grid solutions often *do* require the full 200 time steps. If certain criteria are met, each formed Jacobian is used for three successive time steps. The time steps on each noninitial grid are followed by steady-state Newton's method, with the final solution computed to a tolerance of 1×10^{-4}, which is quite reasonable for a practical combustion problem.

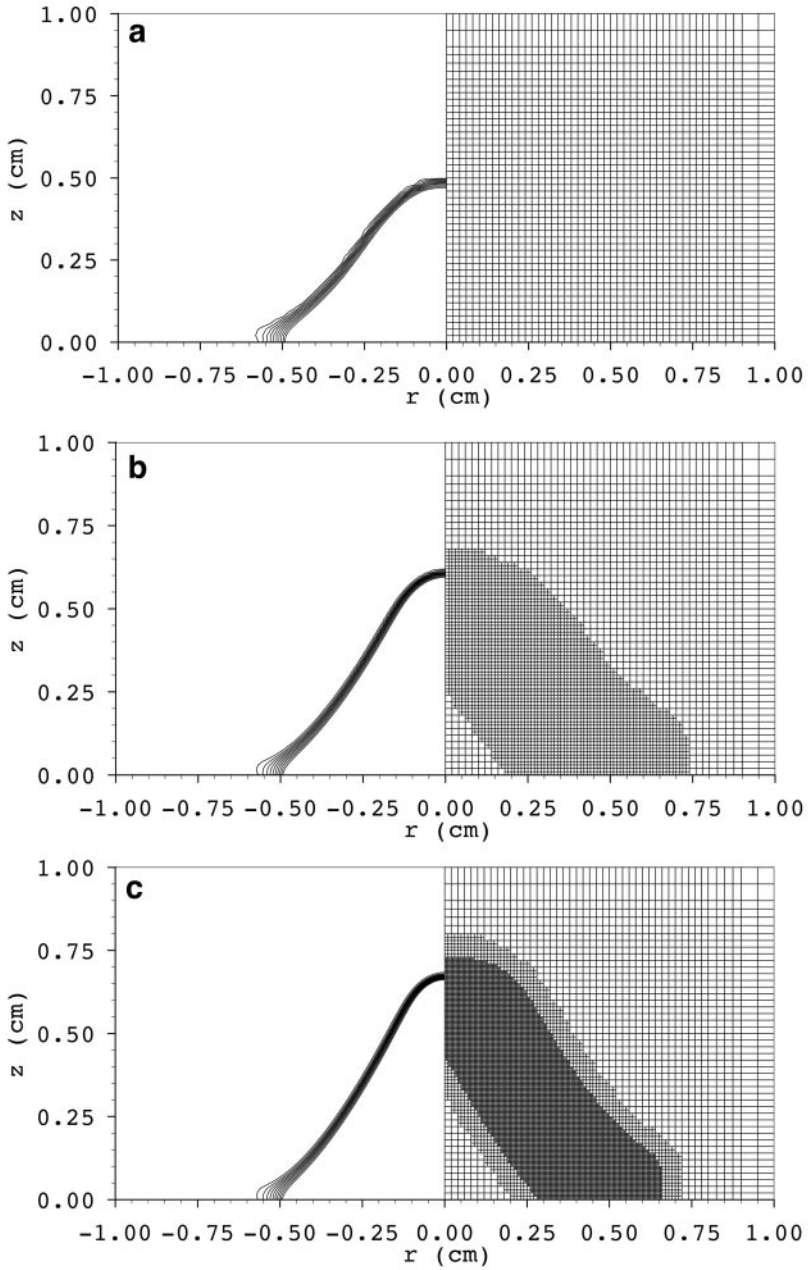

FIG. 23. In the axisymmetric laminar Bunsen flame, Y_{CH_4} isopleths have been computed on a series of LRR adapted grids, formed with Y_{CH_4} as the adaption variable. Shown is the region near the inner jet exit (only a portion of the computational domain but the part in which all of the adaption occurs), for (a) Adaption 0, (b) Adaption 1, (c) Adaption 2, and (d) Adaption 3. A close-up of the Adaption 3 grid is displayed in (e).

5.3.1. Comparison of LRR and traditional gridding methods. As in the square cavity flow problem, there can be no error measurement since the analytical solution is not known. However, the flame length L_f, defined as the z-axis position at which Y_{CH_4} first drops below a preset small value (10^{-4}), is monitored. Although v_r, v_z, ω, T, and the mass fractions of all 16 chemical species have been calculated for each run, only results for T, Y_{CH_4}, and

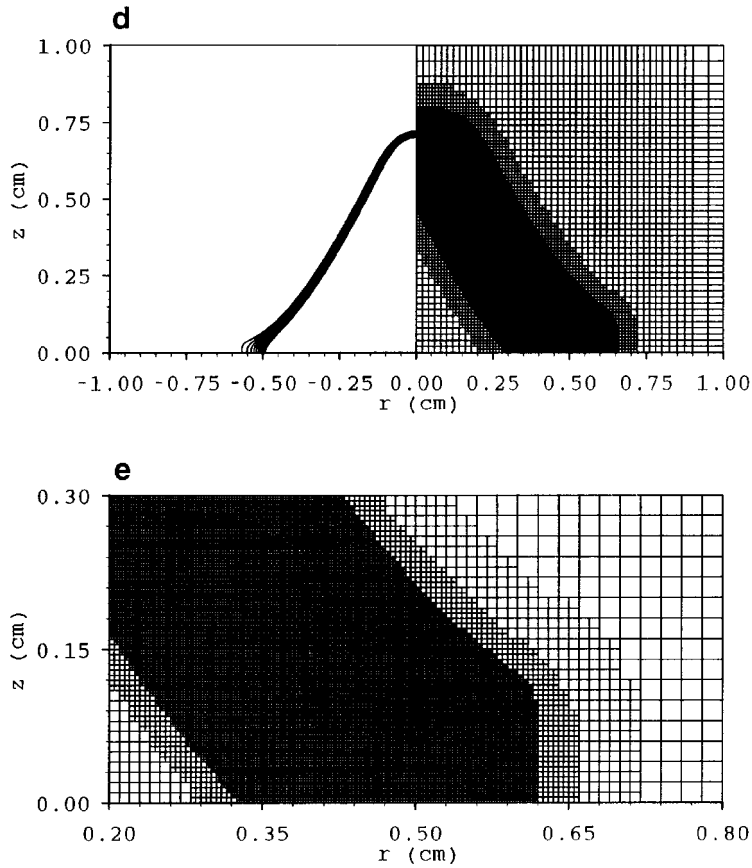

FIG. 23—*Continued*

Y_{OH} are presented in order to give a flavor of the degree of problem difficulty and the level of solution accuracy. Readers interested in the physical interpretation of results for similar flames may pursue [4, 5].

The acute need for adaptive gridding in modeling the axisymmetric Bunsen flame is apparent from Fig. 22, which depicts Y_{CH_4} and Y_{OH} along the flame's axis of symmetry as functions of axial distance above the burner. In Fig. 22a, at the premixed flame front, the Y_{CH_4} profile plummets from 99% of its peak value (at $z = 0.6300$ cm) to less than 1% of peak (at $z = 0.6950$ cm) over an axial distance of only 0.065 cm. This rapid decrease occurs not only on the centerline of the flame but also across the entire surface of the Bunsen cone. For each of the four datasets shown, the flame thickness is spanned axially by 14 grid points, and in LRR Adaption 3 (not shown), it is covered by 27 points. To attain the latter resolution with a fully refined grid would require over 12 million points! In Fig. 22b, the Y_{OH} profile increases steeply near the region in which the Y_{CH_4} profile undergoes its rapid drop. Overall, the Y_{OH} profile contains two peaks: a spike located near 0.80 cm, signifying the presence of a premixed flame front; and a gradual rise and fall centered around 2.0 cm, indicating the diffusion flame halo. Not unexpectedly, the Y_{OH} gradients in the premixed flame are an order of magnitude larger than those in the diffusion flame. Through examination of the governing equations' truncation errors [5], it can be determined that, in order to compute flame positions correctly, the premixed flame front requires significantly higher resolution

TABLE VI
Rich Axisymmetric Laminar Bunsen Flame with Complex Chemistry
Adaption on Y_{CH_4}: Comparison of LRR Method with Traditional Gridding Methods

Method	Adaption	N_{pts}	L_f (cm)	t	t/N_{pts}
LRR	0	4,060	0.520	1.0	0.0002
	1	6,522	0.640	37.9	0.0058
	2	14,499	0.700	95.6	0.0066
	3	41,802	0.738	384.6	0.0092
ETP	0	$58 \times 70 = 4{,}060$	0.520	1.0	0.0002
	1	$95 \times 104 = 9{,}880$	0.650	59.6	0.0060
	2	$160 \times 183 = 29{,}280$	0.730	218.2	0.0075
	3	$284 \times 347 = 98{,}548$	—[a]	—[a]	—[a]
FR	0	$151 \times 1251 = 188{,}901$	—[a]	—[a]	—[a]
	1	$301 \times 2501 = 752{,}801$	—[a]	—[a]	—[a]
	2	$601 \times 5001 = 3{,}005{,}601$	—[a]	—[a]	—[a]
	3	$1201 \times 10001 = 12{,}011{,}201$	—[a]	—[a]	—[a]

[a] Size of problem exceeded 2 GB RAM limit.

than the diffusion flame. Thus, all profiles in Figs. 22a and 22b coincide at the premixed front, where they have the same high level of resolution, with minor differences apparent in the post-diffusion-flame region, where dataset resolutions differ.

Figure 23 illustrates a series of LRR grids, portions of which are shown in the right halves of (a) through (d), generated via adaption on Y_{CH_4}, as well as the corresponding Y_{CH_4} isopleths, whose mirror images are shown in the left halves. The isopleths become smoother and better resolved as the grid is progressively refined, and it is indeed clear that refinement is occurring in the region of high Y_{CH_4} activity. The Adaption 3 grid, for example, appearing in (d) and in greater detail in (e), contains the base grid plus three levels of refinement. The most obvious feature, however, as the grid undergoes refinement, is that the Bunsen flame's length L_f increases from 0.520 cm on the initial grid, to 0.640 and 0.700 cm on successive grids, ending with 0.738 cm on the LRR Adaption 3 grid. The *rate* of

TABLE VII
Rich Axisymmetric Laminar Bunsen Flame with Complex Chemistry
Adaption on Y_{CH_4} and T: Comparison of LRR Method with Traditional Gridding Methods

Method	Adaption	N_{pts}	L_f (cm)	t	t/N_{pts}
LRR	0	4,060	0.520	1.0	0.0002
	1	10,087	0.640	60.2	0.0060
	2	29,262	0.710	200.8	0.0069
ETP	0	$58 \times 70 = 4{,}060$	0.520	1.0	0.0002
	1	$115 \times 132 = 15{,}180$	0.650	84.5	0.0056
	2	$225 \times 260 = 58{,}500$	0.675	341.8	0.0058
FR	0	$151 \times 1251 = 188{,}901$	—[a]	—[a]	—[a]
	1	$301 \times 2501 = 752{,}801$	—[a]	—[a]	—[a]
	2	$601 \times 5001 = 3{,}005{,}601$	—[a]	—[a]	—[a]

[a] Size of problem exceeded 2 GB RAM limit.

increase, however, slows as the grids become more refined, indicating that a threshold level of refinement exists; on grids refined beyond this level, L_f will remain unchanged. (Computer memory constraints prevented the computation of LRR solutions beyond Adaption 3.) This phenomenon of increasing flame length is simply explained in [4, 46] for a Bunsen flame modeled with one-step chemistry, and a similar explanation is given in [5] for Bunsen flames with complex chemical mechanisms. These references conclude that the most accurate L_f is that found on the most refined grid (in this case, that of Adaption 3).

Table VI presents the results obtained by adapting based on Y_{CH_4} using the LRR method, as well as results on corresponding ETP grids; the FR grids contained far too many points, exceeding the available computational resources. Immediately obvious is the fact that, even for the ETP grids, the flame length L_f increases as the mesh spacing becomes finer, with the *rate* of increase progressively lessening. As an indication of the precision of the LRR method, the values for L_f computed on the LRR Adaptions 1 and 2 grids are within 1.5 and 4.1% of their ETP counterparts, with the LRR grids containing 0.66 and 0.50 as many points as their ETP counterparts. (The LRR Adaption 3 grid contains 0.42 as many points as its too-large ETP counterpart.) It is no surprise that the FR grids exceed available memory, since the LRR grids contain 0.02, 0.009, 0.005, and 0.003 times as many points as their FR counterparts! The LRR solution times are roughly half those of the corresponding ETP grids, with the per-point solution times smallest for the LRR grids. In Table VII, for adaption based on both Y_{CH_4} and T, similar behaviors are observed, except that the per-point timings for the LRR grids are slightly longer than those for the corresponding ETP grids, although overall the LRR solutions are still considerably faster because they employ fewer points.

Figure 24 compares the appearances of the grids when adaption occurs only on Y_{CH_4}, as shown in (a), and when adaption occurs on both Y_{CH_4} and T, as shown in (b). In each case,

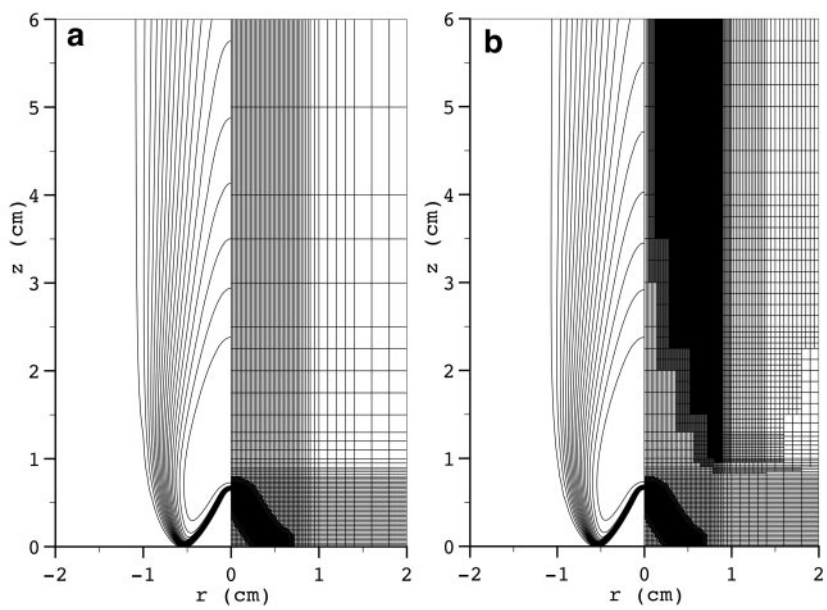

FIG. 24. In the axisymmetric laminar Bunsen flame, computed isotherms are displayed along with portions of LRR grids, formed with Y_{CH_4} as the adaption variable (a), and formed with Y_{CH_4} and T as the adaption variables (b).

a mirror image of the computed isotherms is displayed in the figure's left half. The grid in (b) contains additional refinement in the region of the diffusion flame halo, downstream of the premixed flame front. Despite this difference, the locations of the isotherms (and isopleths of other dependent variables) do not change significantly. This conclusion also holds when examining the results obtained from adaption performed on all 20 dependent variables (not shown).

The storage requirements for the basic implementations of each of the methods have been given earlier in Eq. (16). Substituting $N_{\text{dep}} = 20$ into (16) and adding the extra memory required for the thermodynamic and transport properties, namely $205N_{\text{pts}}$, produce final storage requirements of

$$\begin{aligned} \text{Storage}_{\text{LRR}} &= 4251N_{\text{pts}} \\ \text{Storage}_{\text{ETP or FR}} &= 4155.375N_{\text{pts}} \end{aligned} \tag{21}$$

for the axisymmetric laminar Bunsen flame. For a given number of grid points, these values indicate that the LRR solution process requires only 1.023 times as much memory as on an ETP or FR grid *with the same number of points*. Therefore, whenever an ETP or FR grid requires more than 1.023 times the number of points as the LRR grid from which it was formed, the LRR grid will consume less memory, as is the case for *all* LRR/ETP/FR grid triples presented. It is worth noting that *without* the LRR method, the level of refinement present in the Adaption 3 grid (adaption based on Y_{CH_4}) would not have been possible because of the storage requirements of the corresponding ETP and FR grids. The remainder of Tables VI and VII contain further examples of memory savings with the LRR method.

6. CONCLUSIONS AND FUTURE DIRECTIONS

This paper has presented the LRR solution-adaptive gridding method, which has been developed for the solution of discretized systems of coupled nonlinear elliptic partial differential equations. New discretizations have been derived for use on its unstructured grids, and the resulting discretized set of governing equations has been solved using a damped, modified Newton's method. The adaptive technique has as its basis the principle of weight function equidistribution, upon which traditional tensor product (globally refined rectangular) adaptive gridding methods are also founded. The LRR method has been applied first to a simple test problem to which the analytical solution is known. The new multiple-scale discretizations have been seen to produce a smaller overall error than the single-scale discretizations commonly used on unstructured grids, and the layering technique has also reduced errors while increasing grid robustness. LRR results are comparable in accuracy to those obtained on larger equivalent tensor product (ETP) and equispaced fully refined (FR) grids.

A realistic nonreacting application, that of flow in a thermally and dynamically driven square cavity, has also been examined using the LRR method, as well as the corresponding ETP and FR grids. Regardless of whether the cavity lid is stationary (natural convection case) or moving (mixed convection case), the LRR, ETP, and FR results display excellent agreement among values of the monitor quantities, and mid-plane velocity profiles are also very close to previously published data, especially in the natural convection case. For all results, the LRR grid solutions converge more quickly than their ETP and FR counterparts,

and in several cases even the *per-point* solution times are shorter for LRR. LRR grids used one-half to one-third the number of points as their ETP counterparts, and, on average, the LRR grids consumed half the total computer memory of their ETP equivalents. Further comparison reveals that LRR grids used one-half to *one-hundredth* as many points as their FR counterparts, with commensurate memory savings.

The final application—a rich axisymmetric laminar Bunsen flame encompassing complex chemistry, multicomponent transport, and an optically thin radiation submodel—once again demonstrates the critical need for a method such as the LRR method, since neither the ETP nor FR grids are capable (within the computer memory constraints) of the level of refinement achieved with the LRR grids. While the rich Bunsen flame studied here comprises both a premixed flame *and* a diffusion flame, the gradients present in premixed flames are an order of magnitude larger than those found in diffusion flames. Therefore, very fine spacing is seen to be especially important near the premixed flame front, since the latter's adequate resolution strongly influences flame length. Layering also plays an important role because of the migration of the premixed flame front during the refinement process. In general, the LRR method used less than half the memory and CPU time of that required by the ETP grids, without compromising solution accuracy. (The ETP grid corresponding to LRR Adaption 3 was too large for the computational resources available.) In addition, the results are seen to be largely independent of the choice of adaption variable(s). Comparisons were not possible with the FR grids, each of which greatly exceeded computational resources, containing from 50 to 300 times as many points as their LRR counterparts.

The list of practical problems to which the LRR method can be successfully applied is lengthy. In [4, 5, 46], the new adaptive technique is applied to simple-chemistry and full-chemistry axisymmetric laminar Bunsen flames, both lean and rich, as well as to an axisymmetric diffusion flame with several different chemical mechanisms. In the future, flames including better radiation submodels, in which the number of operations needed to compute the radiation terms scales as the square of the number of grid points, can be examined more efficiently using LRR grids. Soot models, which consume large amounts of computer time with traditional adaptive gridding methods, may then be added. Furthermore, aside from an expected increase in the level of complexity of the discretizations and the unstructured nature of the grid, extension of the LRR method to three dimensions appears feasible. In this latter context, the benefits of using LRR solution-adaptive gridding, measured in terms of both decreased computer storage requirements and lower execution times, should be even greater than those in two dimensions.

ACKNOWLEDGMENTS

This work was supported by the Office of Naval Research (Grant N00014-95-1-0412) and the Department of Energy Office of Basic Energy Sciences (Grant DE-FG02-88ER13836). The first author also acknowledges the previous support of a National Defense Science and Engineering Graduate Fellowship administered through the Army Research Office (Grant DAAH04-93-G-0280).

REFERENCES

1. S. Adjerid and J. E. Flaherty, Finite-element method for two-dimensional parabolic systems, *SIAM J. Sci. Stat. Comput.* **9**, 792 (1988).
2. R. E. Bank and A. H. Sherman, An adaptive, multi-level refinement method for time-dependent partial differential equations, *Computing* **26**, 91 (1981).

3. J. Bell, M. Berger, J. Saltzman, and M. Welcome, Three-dimensional adaptive mesh refinement for hyperbolic conservation laws, *SIAM J. Sci. Comput.* **15**, 127 (1994).
4. B. A. V. Bennett and M. D. Smooke, Local rectangular refinement with application to axisymmetric laminar flames, *Combust. Theory Modelling* **2**, 221 (1998).
5. B. A. V. Bennett and M. D. Smooke, A comparison of the structures of lean and rich axisymmetric laminar Bunsen flames: Application of local rectangular refinement solution-adaptive gridding, submitted for publication.
6. M. J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* **82**, 64 (1989).
7. J. U. Brackbill and J. S. Saltzman, Adaptive zoning for singular problems in two dimensions, *J. Comput. Phys.* **46**, 342 (1982).
8. K. Chen, Error equidistribution and mesh adaption, *SIAM J. Sci. Comput.* **15**, 798 (1994).
9. P. J. Coelho and J. C. F. Pereira, Calculation of a confined axisymmetric laminar diffusion flame using a local grid refinement technique, *Combust. Sci. Tech.* **92**, 243 (1993).
10. A. R. Curtis, M. J. D. Powell, and J. K. Reid, On the estimation of sparse Jacobian matrices, *J. Inst. Math. Appl.* **13**, 117 (1974).
11. A. Dervieux, B. Larrouturou, and R. Peyret, On some adaptive numerical approaches of thin flame propagation problems, *Comput. Fluids* **17**, 39 (1989).
12. P. Deuflhard, A modified Newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting, *Numer. Math.* **22**, 289 (1974).
13. H. C. de Lange, *Modelling of Premixed Laminar Flames*, Ph.D. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands (1992).
14. H. C. de Lange and L. P. H. de Goey, Numerical flow modelling in a locally refined grid, *Int. J. Numer. Methods Eng.* **37**, 497 (1994).
15. G. de Vahl Davis, Natural convection of air in a square cavity: A bench mark numerical solution, *Int. J. Numer. Methods Fluids* **3**, 249 (1983).
16. G. de Vahl Davis and G. D. Mallinson, An evaluation of upwind and central difference approximations by a study of recirculating flow, *Comput. Fluids* **4**, 29 (1976).
17. H. A. Dwyer, M. D. Smooke, and R. J. Kee, Adaptive gridding for finite difference solutions to heat and mass transfer problems, in *Numerical Grid Generation*, edited by J. F. Thompson (North-Holland, Amsterdam, 1982), p. 339.
18. D. K. Edwards, Molecular gas band radiation, *Adv. Heat Transfer* **12**, 115 (1976).
19. A. Ern, C. C. Douglas, and M. D. Smooke, Detailed chemistry modelling of laminar diffusion flames on parallel computers, *Int. J. Supercomput. Appl.* **9**, 167 (1995).
20. A. Ern, V. Giovangigli, D. E. Keyes, and M. D. Smooke, Towards polyalgorithmic linear system solvers for nonlinear elliptic problems, *SIAM J. Sci. Comput.* **15**, 681 (1994).
21. L. Fuchs, A local mesh-refinement technique for incompressible flows, *Comput. Fluids* **14**, 69 (1986).
22. T. Fusegi and B. Farouk, Predictions of fluid flow and heat transfer problems by the vorticity–velocity formulation of the Navier–Stokes equations, *J. Comput. Phys.* **65**, 227 (1986).
23. V. Giovangigli and N. Darabiha, Vector computers and complex chemistry combustion, in *Mathematical Modeling in Combustion and Related Topics*, edited by C.-M. Brauner and C. Schmidt-Lainé (Nijhoff, Dordrecht, 1988), p. 491.
24. V. Giovangigli and M. D. Smooke, Adaptive continuation algorithms with application to combustion problems, *Appl. Numer. Math.* **5**, 305 (1989).
25. R. J. Hall, The radiative source term for plane-parallel layers of reacting combustion gases, *J. Quant. Spectrosc. Radiat. Transfer* **49**, 517 (1993).
26. R. J. Hall, Radiative dissipation in planar gas–soot mixtures, *J. Quant. Spectrosc. Radiat. Transfer* **51**, 635 (1994).
27. Y. Kallinderis, Numerical treatment of grid interfaces for viscous flows, *J. Comput. Phys.* **98**, 129 (1992).
28. J. Kautsky and K. Nichols, Equidistributing meshes with constraints, *SIAM J. Sci. Stat. Comput.* **1**, 499 (1980).

29. R. J. Kee, F. M. Rupley, and J. A. Miller, *The Chemkin Thermodynamic Database*, Sandia National Laboratory Report SAND87-8215 (1987).
30. B. Lewis and G. von Elbe, *Combustion, Flames and Explosion of Gases* (Academic Press, New York, 1987).
31. R. M. M. Mallens, *Stabilisation of Laminar Premixed Methane/Air Flames*, Ph.D. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands (1996).
32. K. Matsuno and H. A. Dwyer, Adaptive methods for elliptic grid generation, *J. Comput. Phys.* **77**, 40 (1988).
33. P. Orlandi, Vorticity–velocity formulation for high Re flows, *Comput. Fluids* **15**, 137 (1987).
34. R. B. Pember, L. H. Howell, J. B. Bell, P. Colella, W. Y. Crutchfield, W. A. Fiveland, and J. P. Jessee, An adaptive projection method for the modeling of unsteady, low-Mach number combustion, in *Western States Section of the Combustion Institute, 1997 Fall Meeting, Diamond Bar, CA*, Paper WSS/CI 97F–125 (1997).
35. P. J. Roache, On artificial viscosity, *J. Comput. Phys.* **10**, 169 (1972).
36. C.-Y. Shen, H. L. Reed, and T. A. Foley, Shepard's interpolation for solution-adaptive methods, *J. Comput. Phys.* **106**, 52 (1993).
37. M. D. Smooke, Solution of burner-stabilized premixed laminar flames by boundary value methods, *J. Comput. Phys.* **48**, 72 (1982).
38. M. D. Smooke, Error estimate for the modified Newton method with applications to the solution of nonlinear, two-point boundary value problems, *J. Optim. Theory Appl.* **39**, 489 (1983).
39. M. D. Smooke and M. L. Koszykowski, Two-dimensional fully adaptive solutions of solid–solid alloying reactions, *J. Comput. Phys.* **62**, 1 (1986).
40. M. D. Smooke and R. M. M. Mattheij, On the solution of nonlinear two-point boundary value problems on successively refined grids, *Appl. Numer. Math.* **1**, 463 (1985).
41. M. D. Smooke, R. E. Mitchell, and R. J. Kee, Solution of premixed and counterflow diffusion flame problems by adaptive boundary value methods, in *Numerical Boundary Value ODEs*, edited by U. M. Ascher and R. D. Russell (Birkhäuser, Boston, 1985), p. 303.
42. M. D. Smooke, R. E. Mitchell, and D. E. Keyes, Numerical solution of two-dimensional axisymmetric laminar diffusion flames, *Combust. Sci. Tech.* **67**, 85 (1989).
43. M. D. Smooke, I. K. Puri, and K. Seshadri, A comparison between numerical calculations and experimental measurements of the structure of a counterflow diffusion flame burning diluted methane in diluted air, *21st Int. Symp. on Combustion* (Combustion Institute, Pittsburgh, PA, 1986), p. 1783.
44. L. T. Somers and L. P. H. de Goey, A numerical study of a premixed flame on a slit burner, *Combust. Sci. Tech.* **108**, 121 (1995).
45. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation* (North-Holland, New York, 1985).
46. B. A. Valdati, *Solution-Adaptive Gridding Methods with Application to Combustion Problems*, Ph.D. thesis, Yale University, New Haven, Connecticut (1997).
47. H. A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* **13**, 631 (1992).
48. Y. Xu and M. D. Smooke, Application of a primitive variable Newton's method for the calculation of an axisymmetric laminar diffusion flame, *J. Comput. Phys.* **104**, 99 (1993).
49. R. Zurn, Local gridding, unpublished (1992).