Preprint of a Paper Submitted for Presentation at the
2001 ANS International Meeting on
Mathematical Methods for Nuclear Applications
Salt Lake City, Utah, U. S. A., September 9-13, 2001

ADAPTIVE MESH REFINEMENT FOR THE TIME-DEPENDENT NODAL INTEGRAL METHOD

Allen J. Toreja[1, 2] and Rizwan-uddin[2]
Department of Nuclear, Plasma, and Radiological Engineering
University of Illinois at Urbana-Champaign
214 Nuclear Engineering Laboratory MC 234
103 South Goodwin Avenue
Urbana, Illinois, 61801 U.S.A.

Also to be considered for the M&C 2001 student paper contest

[1]Corresponding author. E-mail to atoreja@uiuc.edu

[2]Also, Computational Science and Engineering Program

Preprint of a Paper Submitted for Presentation at the *2001 ANS International Meeting on Mathematical Methods for Nuclear Applications*
Salt Lake City, Utah, U.S.A., September 9-13, 2001

ADAPTIVE MESH REFINEMENT FOR THE TIME-DEPENDENT NODAL INTEGRAL METHOD

Allen J. Toreja[1] **and Rizwan-uddin**[1]
Department of Nuclear, Plasma, and Radiological Engineering
University of Illinois at Urbana-Champaign
214 Nuclear Engineering Laboratory MC 234
103 South Goodwin Avenue
Urbana, Illinois 61801 U.S.A.
atoreja@uiuc.edu rizwan@uiuc.edu

Keywords: nodal integral method, adaptive mesh refinement, convection-diffusion equation

ABSTRACT

Adaptive mesh refinement capability has been developed and implemented for the time-dependent nodal integral method. The combination of *adaptive mesh refinement* (AMR) with the *nodal integral method* (NIM) maintains the coarse mesh efficiency of the nodal method by allowing high resolution only in regions where it is needed. Furthermore, exploiting certain features of the nodal method, such as using transverse-integrated variables for efficient error estimation and using node interior reconstruction to develop accurate interpolation operators, can enhance the adaptive mesh refinement process. In this paper, the NIM-AMR is formally developed and applications of the NIM-AMR to convection-diffusion problems are presented. Results show that for a given accuracy, the NIM-AMR can be several times faster than the NIM alone.

1. INTRODUCTION

The nodal integral method (NIM) has been developed for a wide range of problems of interest to nuclear scientists and engineers, including the Navier-Stokes equations, the convection-diffusion equation, and the multi-group neutron diffusion equations (Fischer, 1981; Azmy, 1982; Michael, 1993; Wang, 2000). As an accurate coarse mesh method, the nodal integral method (NIM) has an advantage over conventional numerical schemes in that the NIM requires less CPU time for a given accuracy (Michael, 1995). However, this advantage is not fully realized in problems that are characterized by a large range of spatial scales. In these problems, localized regions may require a high degree of resolution, while the remainder of the problem may be adequately treated with a lower degree of resolution. Regions that necessitate high resolution, though they may constitute small portions of the problem domain, require that a high resolution mesh cover the entire problem domain. Like other numerical methods for structured grids, the application of the NIM to such problems thus requires an extension that allows adaptive mesh refinement (AMR) capability in specific regions of

[1] *Also, Computational Science and Engineering Program*

the problem domain. Such an extension would recover the coarse mesh efficiency of the NIM by allowing high degrees of resolution in specific localized areas *where* it is needed, and using a lower resolution everywhere else. Furthermore, in time-dependent problems, mesh refinement of selected regions may only be needed for a short duration during the simulation period. Therefore, in addition to providing mesh refinement *only where* needed, the AMR extension must also provide mesh refinement *only when* needed.

Mesh refinement capabilities for structured grids have been developed and successfully applied to a variety of fields, ranging from computational fluid dynamics to computational astrophysics (Berger, 1984; Howell, 1997; Bryan, 2000). Although developments of mesh refinement capabilities vary from one problem to another, there are several common underlying features. For instance, each development — whether it is a zooming grid based refinement or a tree based refinement — requires some type of hierarchy that defines and organizes the relationships between meshes and submeshes (regions of the domain where local refinement is needed). These hierarchies frequently require advanced data structures to efficiently store and access information. In addition, each development must have some type of criterion to identify regions where mesh refinement should take place. This selection criterion for mesh refinement is typically based on the location of large gradients or error estimates. Once the mesh points or nodes have been selected, a grid generation (or clustering) algorithm must then efficiently group these points or nodes into submeshes. Each application also requires communication procedures that transfer information between meshes and submeshes. Among these communication routines are extrapolation and interpolation of field values between meshes and submeshes.

The algorithms and data structures associated with adaptive mesh refinement have been efficiently consolidated into a general purpose adaptive mesh refinement algorithm by Berger and Oliger (1984). The Berger-Oliger method to adaptive mesh refinement has been successfully used by many researchers (Skamarock, 1989; Perng, 1992; Masso, 1995; Pember, 1995; Bryan, 2000). More recently, the *Hierarchical Adaptive Mesh Refinement* system (HAMR) has been developed by Neeman (1996) which extends the Berger method into a general purpose software architecture for building adaptive mesh refinement applications on grid hierarchies.

Adaptive mesh refinement capabilities have not been developed for the nodal integral method. The development of adaptive mesh refinement capabilities for the nodal integral method (NIM-AMR) can be carried out in the following manner. First, the development of the AMR for the NIM must incorporate the main elements of AMR, such as the hierarchy, selection criterion, clustering, and communication procedures. The complex data structures of the hierarchy will require a language, such as FORTRAN 90, which can handle advanced data structures. Furthermore, special considerations must be made in developing the selection criterion and communication procedures since the discrete variables of the NIM are the transverse-integrated variables along node surfaces rather than values at grid points. Finally, a governing algorithm will be needed which combines the different components of the AMR. This algorithm determines the order in which the solutions are calculated on different meshes and when mesh refinement should take place. The NIM-AMR is formally developed in the next section. It is then applied to the time-dependent convection-diffusion equation in Section 3, in which results for several test problems and comparison with the NIM without the AMR are presented and discussed.

2. FORMALISM

The main components of the NIM-AMR are: 1) the solver; 2) the level-grid hierarchy; 3) the selection algorithm; 4) the communication procedures; and 5) the governing algorithm. The first component, the solver, consists of the numerical scheme for the governing partial differential equations and the algorithm used to solve the resulting system of discrete algebraic equations. In the case of the NIM-AMR, the solver is the iterative approach to the solution of the set of discrete equations obtained by applying the NIM (Michael, 1993; 1995; Rizwan-uddin, 1997a).

The second component of the NIM-AMR is the level-grid hierarchy, which defines and organizes the relationships between meshes and submeshes (regions of the domain where local refinement is needed). Based on the hierarchy used in the *Hierarchical Adaptive Mesh Refinement* (HAMR) system developed by Neeman (1996), the hierarchy for the NIM-AMR is first divided into *levels*, with each level further subdivided into *grids*. All the grids of one specific level are characterized by one specific resolution, i.e., *grid spacing*. The starting point of the hierarchy is the root level, which consists of only one grid, the root grid. The root grid is the coarsest grid in the hierarchy and covers the entire computational domain. A constant refinement ratio of two is used in the hierarchy such that the grid spacing of all the grids at a particular level is half the grid spacing of the immediately coarser level. To maintain a consistent CFL number throughout the hierarchy, the time steps are also refined by a factor of two. Hence, each subsequent level in the hierarchy has a successively higher degree of spatial and temporal resolution than the root grid. The grids of these subsequent levels consequently cover a smaller, localized region of the original computational domain. Each grid, with the exception of the root grid, is a *child* grid spawned from a *parent* grid of the immediately coarser level. By definition, the parent grid must encompass all of its child grids. In this manner, the hierarchy allows finer grids to cover the coarser grids in regions where the higher resolution is required.

An example of the hierarchy used by the NIM-AMR is shown in Fig. 1 (Neeman, 1996). This hierarchy consists of three levels. Level 0 is the root level and has only one grid — grid 0 — which is the root grid that encompasses the entire computational domain (Fig. 1a). The next level is level 1 (Fig. 1b), which consists of three grids: grid 0, grid 1, and grid 2. The third and final level (Fig. 1c) consists of only two grids: grid 0 and grid 1. The grids of levels 1 and 2 have a higher spatial and temporal resolution than the root grid and cover a smaller, localized region of the computational domain. Furthermore, the grids of level 1 are children of the root grid, while grid 0 of level 2 is the child of grid 0 of level 1 and grid 1 of level 2 is the child of grid 2 of level 1. The final adaptively refined mesh is shown in Fig. 1d.

The adaptive nature of the NIM-AMR comes from the selection algorithm, which identifies regions of the domain that require mesh refinement. The criterion for selecting regions for mesh refinement can be based on values of the field variable or on the locations of large gradients. An alternative and effective criterion, which has been used in many AMR implementations (Berger, 1984; Perng, 1992; Neeman, 1996), is the Richardson truncation error estimate. In this estimate, the local truncation error is approximated at each grid point (or node in the case of the NIM). If the local truncation error exceeds a preset tolerance, the respective grid point is marked for refinement. For time-dependent calculations, the Richardson truncation error, ε, is given by (Neeman, 1996)

$$\varepsilon = \frac{u^h - u^{2h}}{2^{p+1} - 1} + O\left(h^{p+2}\right) \tag{1}$$

where u^h is the numerical solution for a mesh of grid spacing, h; u^{2h} is the numerical solution for a mesh of grid spacing, $2h$; and p is the order of the method (for both space and time).

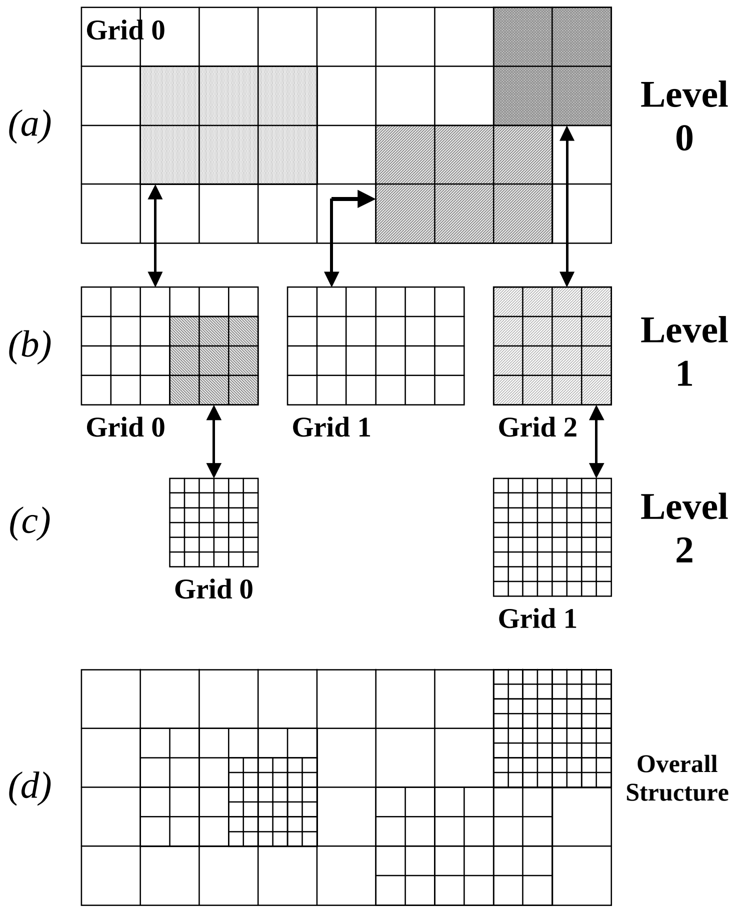


Fig. 1 Sample level-grid hierarchy for the NIM-AMR.

Although the Richardson error estimate is an excellent choice for the selection criterion, the implementation of the error estimate is not trivial. For example, a scheme must be devised to calculate the solution on the coarser “comparison” mesh, u^{2h}. For finite difference methods, comparison meshes are generated through *striding copies* in which every other locus of the current mesh with grid size, *h*, is copied to a mesh with twice the grid size, 2*h* (Neeman, 1996). Furthermore, to maintain a consistent CFL number, the comparison meshes must use a time step that is twice the time step of the current mesh. Once these copies are made, the solution on the comparison meshes is advanced one time step by using the solver. Since the time step of the comparison meshes is larger than the time step of the current mesh, the solution on the current mesh must be *temporarily* advanced one time step so that the solution on the current mesh and the comparison meshes are at the same point in time. The newly calculated solution on the current mesh are the u^h values in Eq. (1) and are erased once the calculation of the error estimate has been made. The newly calculated solution on the current mesh is used only for estimation of the error; hence, the values of the current mesh must be saved in auxiliary storage before temporarily advancing the solution.

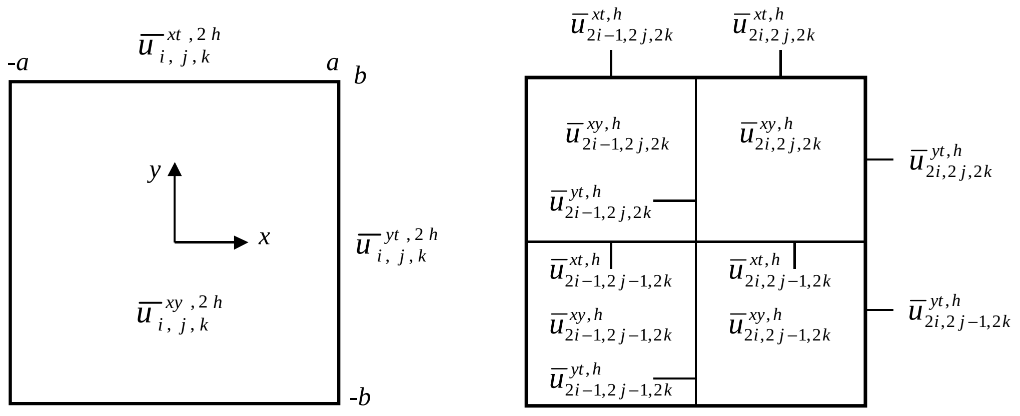
Although the use of striding copies is effective, it leads to significant computational overhead. For the NIM-AMR, a less expensive alternative is possible. One of the discrete variables in the NIM is the field variable spatially averaged over the node *(i, j)* at a specific time, $\bar{u}^{xy}_{i,j,k}$, where the subscripts *i* and *j* are the spatial indices and *k* is the time step index. Moreover, in the NIM, each node of a submesh is actually the child of a parent node, as shown in Fig. 2. Therefore, instead of creating comparison meshes, estimation of the truncation error in the NIM-AMR may be performed using values from the parent grid. For a mesh with node size, *h*, the solution is first *temporarily* advanced one (appropriate) time step. As defined in the hierarchy, the parent grid of the current grid is characterized by a node size of *2h* with each node of the parent grid associated with four nodes of the child (current) grid. In addition, the time step of the parent grid is, by definition, twice the time step of its child grid and thus the *temporary* solution of the current grid is at the same point in time as the parent grid solution. Since node-averaged values are used instead of values at grid points, a consistent comparison can be made by using the average of the four node-averaged variables, $\bar{u}^{xy,h}_{2i,2j,2k}$ $\bar{u}^{xy,h}_{2i-1,2j,2k}$, $\bar{u}^{xy,h}_{2i,2j-1,2k}$, $\bar{u}^{xy,h}_{2i-1,2j-1,2k}$, from the child grid (shown in Fig. 2) and the corresponding node-averaged variable, $\bar{u}^{xy,2h}_{i,j,k}$, from the parent grids, also shown in Fig. 2. Thus, each of the four nodes from the current grid has the same local truncation error, which is given by

$$\varepsilon = \frac{\frac{1}{4}\left(\bar{u}^{xy,h}_{2i,2j,2k} + \bar{u}^{xy,h}_{2i-1,2j,2k} + \bar{u}^{xy,h}_{2i,2j-1,2k} + \bar{u}^{xy,h}_{2i-1,2j-1,2k}\right) - \bar{u}^{xy,2h}_{i,j,k}}{2^{p+1} - 1} + O(h^{p+2}). \quad (2)$$

The calculation of the error estimate in the NIM-AMR is more efficient than the striding copies approach used with finite difference schemes since values that already exist in the hierarchy are used rather than calculating new values on temporary comparison meshes that need to be created. The one exception in the NIM-AMR is the root grid, which requires a temporary coarse mesh to evaluate the truncation error. However, this is unavoidable since, by definition, the root grid must be the coarsest grid in the hierarchy.

Once the selection routine has identified the nodes to be refined, a clustering algorithm is then used to efficiently group the selected nodes and generate the appropriate submeshes. For the NIM-AMR, the point clustering and grid generation algorithm developed by Berger and Rigoutsos (1991) is used.

The next component of the NIM-AMR is the set of communication procedures that transfer information between meshes and submeshes. For the NIM-AMR, the most significant routines in the set are extrapolation, which projects values from the finer grid onto the coarser parent grid, and interpolation, which calculates values on a finer grid based on values from the coarser parent grid. Specifically, extrapolation is the updating process in which values on a grid are corrected using values from its child grid. In the NIM-AMR, extrapolation is performed by overwriting the transverse-integrated discrete variables at each coarse grid node with the average of the transverse-integrated discrete variables from the corresponding nodes of the child grid.



$\overline{u}^{xt,2h}_{i,j,k}$
$-a$
a
b
$-b$
y
x
$\overline{u}^{yt,2h}_{i,j,k}$
$\overline{u}^{xy,2h}_{i,j,k}$
$\overline{u}^{xt,h}_{2i-1,2j,2k}$
$\overline{u}^{xt,h}_{2i,2j,2k}$
$\overline{u}^{xy,h}_{2i-1,2j,2k}$
$\overline{u}^{xy,h}_{2i,2j,2k}$
$\overline{u}^{yt,h}_{2i,2j,2k}$
$\overline{u}^{yt,h}_{2i-1,2j,2k}$
$\overline{u}^{xt,h}_{2i-1,2j-1,2k}$
$\overline{u}^{xt,h}_{2i,2j-1,2k}$
$\overline{u}^{xy,h}_{2i-1,2j-1,2k}$
$\overline{u}^{xy,h}_{2i,2j-1,2k}$
$\overline{u}^{yt,h}_{2i,2j-1,2k}$
$\overline{u}^{yt,h}_{2i-1,2j-1,2k}$

Fig. 2 Schematic diagram of a coarse grid node (node size, *2h*) and corresponding child grid nodes (node size, *h*). Local coordinates and discrete transverse-integrated variables associated with the nodes are also shown.

An advantage that the NIM provides in the interpolation process is that the node interior variation based on solutions of the transverse-integrated differential equations that result from the NIM can be exploited to derive efficient interpolation operators (Wescott, 1999; 2001). The use of these approximate solutions in the node-interior-reconstruction based operators, rather than standard interpolants — which are not dependent on the governing equations — is likely to lead to a more accurate AMR scheme. In the NIM, the transverse integrated variables, $\overline{u}^{yt}_{i,j,k}(x)$, $\overline{u}^{xt}_{i,j,k}(y)$, and $\overline{u}^{xy}_{i,j,k}(t)$ can be combined to reconstruct the node interior variation. For example, $u_{i,j,k}(x, y, t)$ over the node (i, j, k) can be approximated by

$$u_{i,j,k}(x, y, t) \cong \overline{u}^{yt}_{i,j,k}(x) + \overline{u}^{xt}_{i,j,k}(y) + \overline{u}^{xy}_{i,j,k}(t) - 2\overline{u}^{xyt}_{i,j,k} \qquad (3)$$

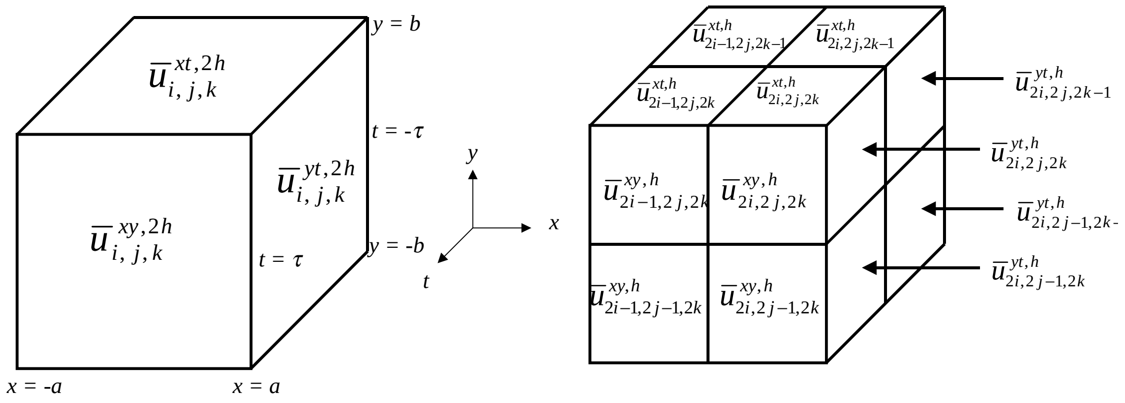
where $\overline{u}^{xyt}_{i,j,k}$ is the node-averaged value. Such analytical node interior reconstruction is likely to lead to more accurate discrete transverse-integrated and node-averaged values for the child nodes. By using the nodal reconstruction on the coarse space-time node of Fig. 3, the transverse-integrated variables associated with the eight smaller space-time nodes, also shown in Fig. 3, can be evaluated. For example, the transverse-integrated variables associated with the node *(2i, 2j)* from the fine mesh at time step *2k* are given by

$$\overline{u}^{yt,h}_{2i,2j,2k} = \frac{1}{\tau}\int_0^{\tau} \overline{u}^{xy,2h}_{i,j,k}(t)dt + \frac{1}{b}\int_0^{b} \overline{u}^{xt,2h}_{i,j,k}(y)dy + \overline{u}^{yt,2h}_{i,j,k}(x = a) - 2\overline{u}^{xyt,2h}_{i,j,k}, \quad (4)$$

$$\overline{u}^{xt,h}_{2i,2j,2k} = \frac{1}{\tau}\int_0^{\tau} \overline{u}^{xy,2h}_{i,j,k}(t)dt + \frac{1}{a}\int_0^{a} \overline{u}^{yt,2h}_{i,j,k}(x)dx + \overline{u}^{xt,2h}_{i,j,k}(y = b) - 2\overline{u}^{xyt,2h}_{i,j,k}, \quad (5)$$

$$\overline{u}^{xy,h}_{2i,2j,2k} = \frac{1}{a}\int_0^{a} \overline{u}^{yt,2h}_{i,j,k}(x)dx + \frac{1}{b}\int_0^{b} \overline{u}^{xt,2h}_{i,j,k}(y)dy + \overline{u}^{xy,2h}_{i,j,k}(t = \tau) - 2\overline{u}^{xyt,2h}_{i,j,k}. \quad (6)$$

Once a submesh is created, the initial values of the submesh along with the boundary values are evaluated using Eqs. (4-6). Furthermore, initial guesses for the interior values can also be made using Eqs. (4-6).



$\overline{u}^{xt,2h}_{i,j,k}$
$\overline{u}^{yt,2h}_{i,j,k}$
$\overline{u}^{xy,2h}_{i,j,k}$
y = b
t = -τ
t = τ
y = -b
x = -a
x = a
y
x
t
$\overline{u}^{xt,h}_{2i-1,2j,2k-1}$
$\overline{u}^{xt,h}_{2i,2j,2k-1}$
$\overline{u}^{xt,h}_{2i-1,2j,2k}$
$\overline{u}^{xt,h}_{2i,2j,2k}$
$\overline{u}^{xy,h}_{2i-1,2j,2k}$
$\overline{u}^{xy,h}_{2i,2j,2k}$
$\overline{u}^{xy,h}_{2i-1,2j-1,2k}$
$\overline{u}^{xy,h}_{2i,2j-1,2k}$
$\overline{u}^{yt,h}_{2i,2j,2k-1}$
$\overline{u}^{yt,h}_{2i,2j,2k}$
$\overline{u}^{yt,h}_{2i,2j-1,2k-}$
$\overline{u}^{yt,h}_{2i,2j-1,2k}$

Fig. 3 Schematic diagram of coarse *space-time* node (node size, *2h*) and the corresponding child *space-time* nodes (node size, *h*). Local coordinates and discrete transverse-integrated variable associated with the nodes are also shown.

The final component of the NIM-AMR is the governing algorithm, which is based on the algorithm used in the Berger-Oliger method (1984). The governing algorithm of the NIM-AMR (*integrate*) is shown in Fig. 4. The *integrate* algorithm recursively cycles through the levels of the hierarchy, advancing solutions on grids by the appropriate time step and refining grids *where and when* necessary. Specifically, the order in which grids are advanced in time follows a “ W cycle” as shown in Fig. 5. Furthermore, the *integrate* algorithm demonstrates how the level-grid hierarchy changes with time. In the first root level time step, the level-grid hierarchy is generated with the number of levels in the hierarchy, which is not to exceed a preset maximum, determined by the selection algorithm. All the grids of a particular level exist only for a preset *even* number of (that particular level's) time steps and the grids are deleted once this time limit is reached. (Since the NIM-AMR uses a refinement ratio of two in both space and time, the grids must exist for an *even* number of appropriate time steps in order for the level-grid hierarchy to be consistent.) When grids from this particular level are needed later, new grids on this level will be created (and again deleted) *where and when* necessary. In this manner, the grids of the hierarchy "move and conform" to areas where refinement is required, and are deleted when refinement is no longer required.

```
integrate(level)
begin
   for all grids at this level,
      {collect boundary values;}
   for all grids at this level,
      {evaluate constants;}
   for all grids at this level,
      {calculate solution;}
   if level is not finest allowed and if next level
   does not exist, then
      {for all grids at this level,
            {select nodes for refinement and
             create appropriate subgrids;}}
   if level is not finest existing, then
      {integrate(level + 1);
       integrate(level + 1);}
   for all grids at this level,
      {increment time;
       update values;}
   for all grids at this level except root level;
      {project values onto parent grids;}
   if level is not root and if time to delete, then
      {delete all grids at this level;}
end
```

Fig. 4   Governing algorithm for the NIM-AMR.

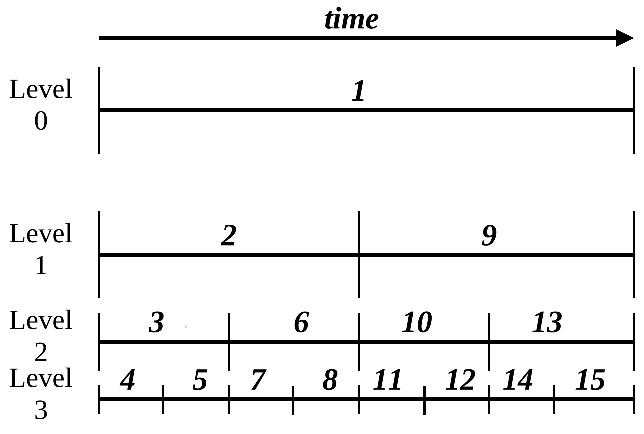


Fig. 5   Recursive "W cycle" of the NIM-AMR.

Because of the recursive nature of the governing algorithm and the complex data structures associated with the level-grid hierarchy, the implementation of the NIM-AMR requires a language with advanced capabilities such as FORTRAN 90, which provides all the necessary resources, including dynamic memory allocation, advanced data structures, and recursive subroutine capability. This concludes the formal development of the nodal integral method with adaptive mesh refinement.

3. RESULTS AND DISCUSSION

Time-dependent convection-diffusion problems are used to test the nodal integral method with adaptive mesh refinement. The time-dependent convection-diffusion equation (CDE) is

$$\frac{\partial C}{\partial t} = D\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2}\right) - v_x \frac{\partial C}{\partial x} - v_y \frac{\partial C}{\partial y} + Q \tag{7}$$

where C is the concentration, D is the diffusion coefficient, Q is the source term, and v_x and v_y are the velocities in the x and y directions respectively. Previously, the NIM has been applied to a number of convection-diffusion problems (Michael, 1995; Rizwan-uddin, 1997; Toreja, 1999; Toreja, 2000) and is found to be more efficient than standard numerical schemes, such as the LECUSSO method (Gunther, 1988; 1992) and the control volume scheme of Faghri (1984). Thus, the addition of the adaptive mesh refinement capability will add to the overall efficiency of the NIM. The NIM-AMR has been coded in FORTRAN 90, and is currently operational on a *Sun Ultra 5* Workstation.

3.1 Diffusion Dominated Problems

The first set of problems used to test the NIM-AMR consists of diffusion dominated (low Peclet number) problems. In these problems, the NIM-AMR did not show any clear advantage over the NIM without the adaptive mesh refinement capability. That is, the NIM-AMR did not yield lower CPU times for a given accuracy in comparison to the NIM alone. However, this is to be expected. Low Peclet number problems are typically characterized by smoothly varying field distributions in which the errors throughout the computational domain tend to be similar. Hence, application of the selection algorithms in the NIM-AMR would either choose the entire domain for mesh refinement or none at all, depending upon the preset tolerance used in the Richardson error estimate. If subregions which need refinement cannot be identified, such as in diffusion dominated problems, then adaptive mesh refinement cannot lead to improved efficiency.

3.2 Decaying Shock Problem

The next problem used to test the NIM-AMR is the decaying shock problem, which was previously used to test the NIM for the viscous, two-dimensional Burgers' equation (Wescott, 1999; 2001). In this problem, an exact solution for the concentration field is chosen to be

$$C(x, y, t) = \frac{1}{2}\left[1 - \tanh\left(\frac{x}{D}\right)\right] * \left[\frac{1}{2} - \frac{y}{4}\right] * \left[-1 + 2e^{-t}\right]. \quad (8)$$

Furthermore, a uniform diagonal flow field with $v_x = v_y = 1.0$ is used for this problem. Substituting Eq. (8) into Eq. (7) yields the source term,

$$Q(x, y, t) = -\frac{1}{8D}\left\{\begin{array}{l} e^{-t}\left(D\left(\left(-2 + e^{t}\right)v_y + 2(-2 + y)\right)\left(-1 + \tanh\left(\frac{x}{D}\right)\right)\right) \\ + e^{-t}\left(\left(-2 + e^{t}\right)(-2 + y)\left(\frac{1}{\cosh^2\left(\frac{x}{D}\right)}\right)\left(v_x + 2\tanh\left(\frac{x}{D}\right)\right)\right) \end{array}\right\}. \quad (9)$$

Hence, Eq. (8) is a solution to the CDE when the source term is given by Eq. (9). The concentration field, which is shown in Fig. 6a for t = 0.0, has a spatially stationary front in the x direction and has a linear variation in the y direction. Moreover, the steepness of the front depends on the argument in the hyberbolic tangent function, x/D. Thus, for low diffusion coefficients, the gradient is quite sharp. Although the front does not move spatially (the source term counteracts any convective or diffusive transport), the front initially decays in time, and then reappears approaching a steady-state solution, which is the reflection of the initial condition across the x-y plane, as shown in Fig. 6b.

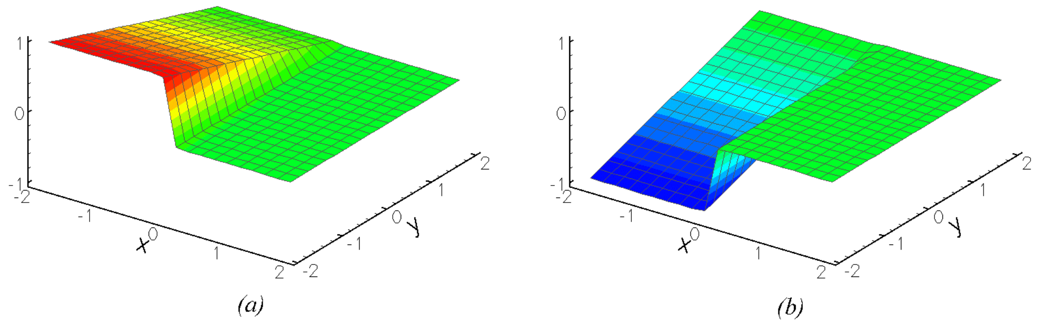


Fig. 6 (a) Initial condition and (b) steady-state solution for the decaying shock problem.

The decaying shock problem is solved using the NIM-AMR for a diffusion coefficient, D = 0.01, over a square computational domain, $-2 \le x \le 2$; $-2 \le y \le 2$, from an initial time, t = 0.0, to a final time, t = 2.0. Since an analytical solution is available for this problem, exact Dirichlet conditions are imposed along the boundary of the domain. For the NIM-AMR calculations, a uniform node size of $\Delta x = \Delta y$ = 0.2 and a uniform time step of Δt = 0.2 is used on the root grid. Hence, the node Peclet number based on the root grid resolution is $Pe = U^*\Delta x/D \cong 1.0^*0.2/0.01 = 20$. At this Peclet number, the concentration field is neither convection nor diffusion dominated. Furthermore, in the NIM-AMR calculations, the grids are tested for refinement every eight time steps using a preset error tolerance of $2.5(\Delta h)^2$ in which Δh is the node size of the current grid for which the error estimate is made.

To demonstrate the efficiency of the adaptive mesh refinement capability, results obtained using the NIM-AMR are compared with those obtained using the NIM. The iterations in both, the NIM-AMR and the NIM calculations, are slightly underrelaxed ($0.8 \leq \omega \leq 0.95$). Table 1a summarizes the NIM-AMR results for one and two levels of refinement. Similarly, Table 1b summarizes the NIM results for four different meshes: 1) Δx = 0.2, Δt = 0.2; 2) Δx = 0.1, Δt = 0.1; 3) Δx = 0.08, Δt = 0.08; and 4) Δx = 0.05, Δt = 0.05. To make a quantitative comparison between the two sets of results, CPU times for the NIM at a given RMS error are calculated via regressive analysis of the NIM results presented in Table 1b. Based on this regression analysis, a CPU time of approximately 35 seconds is needed to achieve an RMS error of 0.0237 using the NIM. In comparison, for the NIM-AMR, an RMS error of 0.0237 is obtained with one level of refinement in a CPU time of 20.3 seconds, which is a little more than half the CPU time required by the NIM. Thus, for this desired accuracy, the NIM-AMR is *almost twice as fast* as the NIM. Using two levels of refinement, the NIM-AMR yields an RMS error of 0.0178 in a CPU time of 76.6 seconds. Based on the regression analysis, the NIM achieves an RMS error of 0.0178 in a CPU time of 86.0 seconds. The lack of improvement of the NIM-AMR for two levels of refinement indicates that the overhead costs associated with the AMR capability will eventually diminish the advantage of the NIM-AMR if too many levels of refinement are used unnecessarily.

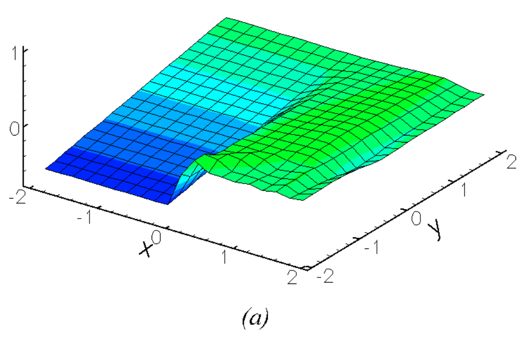
Table 1a NIM-AMR results for the decaying shock problem

Levels	*RMS Error*	*CPU Time*
1	0.0237	20.3
2	0.0178	76.7

Table 1b NIM results for the decaying shock problem

Mesh	Δx	Δt	*RMS Error*	*CPU Time*
1	0.2	0.2	0.0479	3.39
2	0.1	0.1	0.0276	25.5
3	0.08	0.08	0.0225	47.0
4	0.05	0.05	0.0136	174

One of the advantages of adaptive mesh refinement is the efficient reduction of errors in specific regions. The NIM calculations for the Δx = 0.2 mesh exhibits non-physical oscillations in the concentration distribution. An example of these oscillations can be seen in the surface plot of the node-averaged concentration, $\overline{C}^{xy}$, at t = 1.6, shown in Fig. 7a. In Fig. 7b, the corresponding node-averaged concentration distribution obtained with the NIM-AMR using one level of refinement is shown. Although the NIM-AMR results also exhibit slight non-physical oscillations, these oscillations are significantly smaller than those in the NIM calculations. Moreover, the NIM-AMR with one level of refinement consistently yields results with smaller non-physical oscillations than the NIM for the entire simulation period.



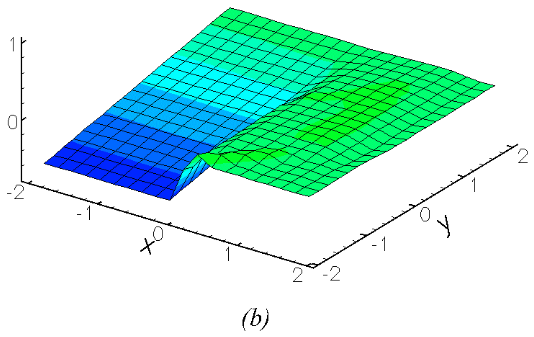


Fig. 7 Node-averaged concentration, $\overline{C}^{xy}$, at t = 1.6 for the (a) NIM and (b) NIM-AMR.

In the NIM-AMR calculations for the decaying shock problem, the communication procedures are performed using Eqs. (4-6) which are based on the node interior reconstruction. To evaluate the performance of these node-interior-reconstruction based communication procedures, the NIM-AMR calculations for the decaying shock problem are repeated using standard second and third order interpolation/extrapolation schemes (*Numerical Recipes*, Press, 1992). A summary of results for these modified NIM-AMR calculations using the standard interpolation schemes is given in Table 2. For one level of refinement, the NIM-AMR with the second and third order interpolation/extrapolation schemes has an RMS error of 0.0244 in a CPU time of 21.2 seconds. Both the RMS error and the CPU time are comparable to those for the NIM-AMR results obtained with the node-interior-reconstruction based interpolation schemes (Table 1a, Level 1). At two levels of refinement, the CPU time for both the NIM-AMR with the standard interpolation/extrapolation schemes and the NIM-AMR with the node-interior-reconstruction based schemes are again fairly close — 70.0 seconds with the standard schemes and 76.7 seconds with the nodal schemes. However, with an RMS error of 0.0219, the reduction in RMS error from one level of refinement to two levels of refinement for the NIM-AMR with the standard schemes is not as large as the reduction of the RMS error in the NIM-AMR with the nodal schemes, which has an RMS error of 0.0178. The superior performance of the NIM-AMR with the node-interior-reconstruction based routines is an example of how this particular characteristic of the NIM can be advantageously exploited in the adaptive mesh refinement process.

Table 2 NIM-AMR results using standard interpolation/extrapolation routines

Levels	*RMS Error*	*CPU Time*
1	0.0244	21.2
2	0.0219	70.0

3.3 Advection of a Gaussian Hump in a Rotating Flow Field

The final problem used to test the NIM-AMR is the advection of a Gaussian hump in a rotating flow field, which is a relatively difficult benchmark problem used to test the robustness of numerical schemes developed for the CDE (Cantenkin, 1990; Croucher 1998). A concentration distribution in the shape of a Gaussian hump with a standard deviation of σ = 0.15 is initially placed near the point *(x = 2.86, y = 2.86)*. The computational domain for this problem is the square region, $0 \le x \le 4; 0 \le y \le 4$. The rotating flow field is given by

$$\vec{V} = 2\pi \begin{pmatrix} 2 - y \\ x - 2 \end{pmatrix}. \qquad (10)$$

The flow field is configured such that the hump will rotate counterclockwise through one half of a revolution about the point *(x = 2.0, y = 2.0)* in 0.5 time units.

For the NIM-AMR calculations, a uniform node size of $\Delta x = \Delta y = 0.08$ and a uniform time step of $\Delta t = 0.01$ is used on the root grid. To simulate nearly pure advection, a diffusion coefficient of $D = 10^{-10}$, which correspond to a maximum node Peclet number of $Pe = U_{max} * \Delta x / D \cong 8.0 * 0.08 / 10^{-10} = 6.4 * 10^{9}$, is used. Thus, minimal diffusion and minimal spreading of the hump is expected over the simulation period. In this problem, the grids are tested for refinement every eight time steps and the preset error tolerance is $5.0(\Delta h)^2$. In this particular example, the node-interior-reconstruction based communication procedures are found to be less efficient than the standard second/third order interpolation/extrapolation procedures. Consequently, the results of the NIM-AMR presented below are based on the second/third order standard interpolation/extrapolation procedures.

Similar to the previous test problem, calculations are performed with the NIM-AMR and the NIM alone. As before, slight underrelaxation ($0.8 \le \omega \le 0.95$) is utilized in both the NIM-AMR and the NIM calculations. In this problem, the meshes used by the NIM are: 1) Δx = 0.1, Δt = 0.0125; 2) Δx = 0.05, Δt = 0.00625; and 3) Δx = 0.04, Δt = 0.003125. The NIM-AMR and NIM results are summarized in Tables 3a and 3b, with a regression analysis performed on the NIM results to facilitate comparison between the sets of results. Based on the regression analysis, a CPU time of 9480 seconds (~ 2.633 hours) is needed for an RMS error of 0.00198 with the NIM. The NIM-AMR with two levels of refinement yields an RMS error of 0.00198 in a CPU time of 2400 seconds (~ 0.667 hours). Thus, the NIM-AMR for this given accuracy is approximately *four times faster* than the NIM. As expected, the NIM-AMR yields even smaller RMS errors with three levels of refinement. Figure 8 displays contour plots of the node-averaged concentration, $\overline{C}^{xy}$, at times, t = 0.0 and t = 0.25, which are calculated using the NIM-AMR with three levels of refinement. In addition, the subgrid boundaries for the first level of refinement are outlined in the plot. As Fig. 8 demonstrates, the NIM-AMR performs quite well, producing symmetric and concentric contours for t = 0.25.

Table 3a NIM-AMR results for the Gaussian hump problem

Levels	*RMS Error*	*CPU Time*
1	0.00466	1080
2	0.00198	2400
3	0.00127	21700

Table 3b NIM results for the Gaussian hump problem

Mesh	Δx	Δt	*RMS Error*	*CPU Time*
1	0.1	0.0125	0.0218	193
2	0.05	0.00625	0.00680	1320
3	0.025	0.003125	0.00172	11800

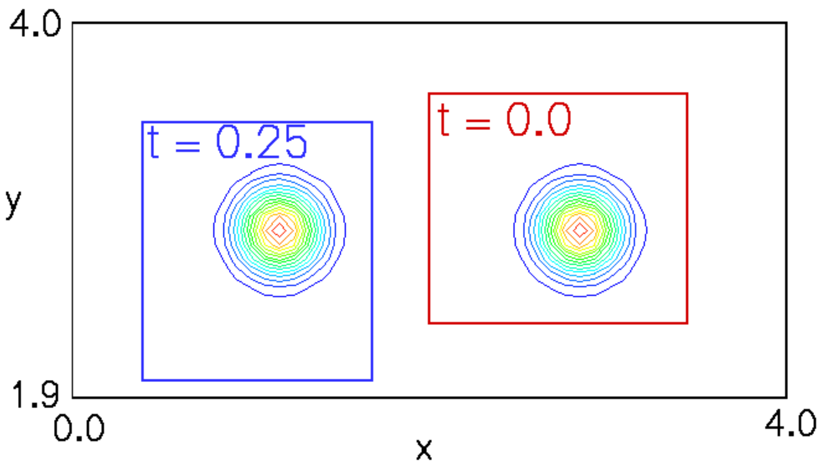


Fig. 8 NIM-AMR results for the node-averaged concentration, $\overline{C}^{xy}$, at times, t = 0.0 and t = 0.25. Subgrid boundaries for the first level of refinement are also shown.

4. CONCLUSIONS

Adaptive mesh refinement capabilities have been developed for the time-dependent nodal integral method. The nodal method with mesh refinement capability has been implemented in a computer code for the convection-diffusion equation and applied to several problems with a wide range of Peclet numbers. In diffusion dominated problems, the AMR capability does not lead to significant improvement of the NIM since there are no specific regions which need refinement. However, in intermediate Peclet number and pure advection problems with regions of sharp gradients, the NIM-AMR is *two to four times faster* than the NIM alone for a given accuracy. Furthermore, certain features of the NIM, such as the node interior reconstruction, when fruitfully exploited, have been shown to enhance the AMR process. These results clearly show that the combination of adaptive mesh refinement with the nodal integral method extends the coarse mesh efficiency of the nodal method to problems with a large range of spatial scales by utilizing mesh refinement *where* and *when* it is needed.

NOMENCLATURE

u	generic field variable
ε	Richardson truncation error
p	order of numerical method
v_x	velocity in x-direction
v_y	velocity in y-direction
C	concentration
Q	source terms
a	half node length in x-direction
b	half node length in y-direction
τ	half time step
Pe	Peclet number

Subscripts

i	x-direction index
j	y-direction index
k	time step index

Superscripts

h	grid size h
$2h$	grid size $2h$
—x	averaged over the node in x-direction
—y	averaged over the node in y-direction
—t	averaged over the time step

Acronyms and Abbreviations

AMR	adaptive mesh refinement
CDE	convection-diffusion equation
CFL	Courant Friedrichs Lewy
CPU	central processing unit
LECUSSO	locally exact consistent upwind scheme of second order
NIM	nodal integral method
NIM-AMR	nodal integral method with adaptive mesh refinement
RMS	root mean square

ACKNOWLEDGEMENTS

This research was supported in part by the Center for Simulation of Advanced Rockets, University of Illinois at Urbana-Champaign. The Center is supported by the US Department of Energy through the University of California under subcontract B341494.

REFERENCES

Y. Y. Azmy, 1982. A Nodal Integral Method For the Numerical Solution of Incompressible Flow Problems, M. S. Thesis, University of Illinois at Urbana-Champaign.

M. J. Berger, J. Oliger, 1984. Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations. *Journal of Computational Physics*, **53**, 484-512.

M. J. Berger, I. Rigoutsos, 1991. An Algorithm for Point Clustering and Grid Generation. *IEEE Transactions on Systems, Man, and Cybernetics*, **21**, 5, 1278-1286.

G. L. Bryan, M. L. Norman, 2000. A Hybrid AMR Application For Cosmology and Astrophysics. In: S. B. Baden, N. P. Chrisochoides, D. B. Gannon, M. L. Norman (Eds.), *Structured Adaptive Mesh Refinement (SAMR) Grid Methods*, Springer-Verlag, New York, pp. 165-170.

M. E. Cantenkin, J. J. Westerink, 1990. Non-diffusive N+2 Degree Petrov-Galerkain Methods for Two-Dimensional Transient Transport Compuations. *International Journal of Numerical Methods in Engineering*, **30**, 397-418.

A. E. Croucher, M. J. O'Sullivan, 1998. Numerical Methods for Contaminant Transport in Rivers and Estuaries. *Computers and Fluids*, **27**, 8, 861-878.

M. Faghri, E. M. Sparrow, A. T. Prata, 1984. Finite-Difference Solutions of Convection-Diffusion Problems in Irregular Domains, Using a Nonorthogonal Coordinate Transformation. *Numerical Heat Transfer*, **7**, 183-209.

H. D. Fisher, H. Finnemann, 1981. The Nodal Integration Method — A Diverse Solver for Neutron Diffusion Problems. *Atomkernenergie*, **39**, 229-236.

Cl. Gunther, 1988. A Consistent Upwind Method of Second Order for the Convection-Diffusion Equation. In: J. Noye, C. Fletcher (Eds.), *Comp. Tech. Appl.*, CTAC-87, Elsevier Science Publishers, S. V. (North Holland), pp. 447-463.

Cl. Gunther, 1992. Conservative Versions of the Locally Exact Consistent Upwind Scheme of Second Order Lecusso-Scheme. *International Journal for Numerical Methods in Engineering*, **34**, 793-804.

L. H. Howell, J. B. Bell, 1997. An Adaptive Mesh Projection Method for Viscous Incompressible Flow. *SIAM Journal of Scientific Computing*, **18**, 4, 996-1013.

J. Masso, E. Seidel, P. Walker, 1995. Adaptive Mesh Refinement in Numerical Relativity, In: R Ruffini, M. Keiser (Eds.), *Proceedings of the Seventh Marcel Grossman Meeting on General Relativity,* World Scientific.

E. P. E. Michael, J. J. Dorning, E. M. Gelbard, Rizwan-uddin, 1993. A Nodal Integral Method for the Convection-Diffusion Heat Equation. *Transactions of the American Nuclear Society,* **69**, 239-241.

E. P. E. Michael, 1995. *A Nodal Integral Method for the Linear Steady State Convection-Diffusion Equation*, M. S. Thesis, University of Virginia.

H. Neeman, 1996. *Autonomous Hierarchical Adaptive Mesh Refinement For Multiscale Simulations*, Ph. D. Thesis, University of Illinois at Urbana-Champaign.

R. B. Pember, J.B. Bell, P. Collela, W.Y. Crutchfield, M.L. Welcome, 1995. An Adaptive Cartesian Grid Method for Unsteady Compressible Flow in Irregular Regions. *Journal of Computational Physics*, **120**, 2, 278-304.

C. Y. Perng, R. L. Street, 1992. An Adaptive Grid Calculation For the Incompressible Unsteady Navier-Stokes Equations. In: I. Kececioglu, J. Murthy (Eds.), *Adaptive Computational Methods in Environmental Transport Processes*, ASME, New York, pp. 27-34.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, 1992. *Numerical Recipes in FORTRAN 77*, Cambridge University Press, New York, pp. 99-122.

Rizwan-uddin, 1997a. A Second Order Space and Time Nodal Method for the One-Dimensional Convection-Diffusion Equation. *Computers and Fluids,* **26**, 3, 233-247.

Rizwan-uddin, 1997b. An Improved Coarse-Mesh Nodal Integral Method for Partial Differential Equations. *Numerical Methods for Partial Differential Equations*, **13**, 113-145.

W. Skamarock, J. Oliger, R.L. Street, 1980. Adaptive Grid Refinement for Numerical Weather Prediction*. Journal of Computational Physics*, **80**, 27-60.

A. J. Toreja, Rizwan-uddin, 1999. Hybrid Numerical Methods for the Convection-Diffusion Equation in Arbitrary Geometries. In: J. M. Aragones, C. Ahnert, O. Cabellos (Eds.), *Proceedings of the International Conference on Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications*, Senda Editorial, Madrid, pp. 1705-1714.

A. J. Toreja, Rizwan-uddin, 2000. A Hybrid Numerical Method for Time-Dependent Convection-Diffusion Problems in Arbitrary Geometries. *Transactions of the American Nuclear Society*, **83**, 426-428.

F. Wang, Rizwan-uddin, 2000. A Nodal Scheme for the Time-Dependent Incompressible Navier-Stokes Equations. *Transactions of the American Nuclear Society*, **83**, 422-424.

B. L. Wescott, Rizwan-uddin, 1999. An Efficient Formulation of the Modified Nodal Integral Method and Application to 2-D Burger's Equation. In: *Proceedings of the Ninth International Conference on Nuclear Reactor Thermal Hydraulics (NURETH-9)*, American Nuclear Society, 1999.

B. L. Wescott, 2001. *Efficient Formulation of the Modified Nodal Integral Method and Development of Implicit, Explicit, and Multigrid Schemes Applied to the Two-Dimensional Burgers' Equation*, M. S. Thesis, University of Illinois at Urbana-Champaign.