461207

36p

VKI Lecture Series 1999-2000

Aero-Thermal Performance of Internal Cooling Systems in Turbomachines

NUMERICAL SIMULATION OF NON-ROTATING AND ROTATING COOLANT CHANNEL FLOW FIELDS

Part 1

David L. Rigby
Dynacs Engineering Co. Inc.
Cleveland, OH
email: rigby@grc.nasa.gov

This is a preprint or reprint of a paper intended for presentation at a conference.
Because changes may be made before formal publication, this is made available with the understanding that it will not be cited or reproduced without the permission of the author.

Contents

1. Introduction

Future generations of ultra high bypass-ratio jet engines will require far higher pressure ratios and operating temperatures than those of current engines. For the foreseeable future, engine materials will not be able to withstand the high temperatures without some form of cooling. In particular the turbine blades, which are under high thermal as well as mechanical loads, must be cooled (Taylor, 1980, Suo, 1978 and Snyder and Roelke, 1990). Cooling of turbine blades is achieved by bleeding air from the compressor stage of the engine through complicated internal passages in the turbine blades (internal cooling, including jet-impingement cooling) and by bleeding small amounts of air into the boundary layer of the external flow through small discrete holes on the surface of the blade (film cooling and transpiration cooling). The cooling must be done using a minimum amount of air or any increases in efficiency gained through higher operating temperature will be lost due to added load on the compressor stage.

Turbine cooling schemes have traditionally been based on extensive empirical data bases, quasi-one-dimensional computational fluid dynamics (CFD) analysis, and trial and error. With improved capabilities of CFD, these traditional methods can be augmented by full three-dimensional simulations of the coolant flow to predict in detail the heat transfer and metal temperatures. Several aspects of turbine coolant flows make such application of CFD difficult, thus a highly effective CFD methodology must be used. First, high resolution of the flow field is required to attain the needed accuracy for heat transfer predictions, making highly efficient flow solvers essential for such computations. Second, the geometries of the flow passages are complicated but must be modeled accurately in order to capture all important details of the flow. This makes grid generation and grid quality important issues. Finally, since coolant *flows are turbulent and separated the effects of turbulence must be modeled with a low* Reynolds number turbulence model to accurately predict details of heat transfer.

The overall objective of our ongoing research is to develop a CFD methodology that can be used effectively to design and evaluate turbine cooling schemes. Grid generation for complicated geometries such as coolant passages, is currently an active area of research. In general, grid systems for complicated geometries are classified as block-structured, unstructured or hybrid. Of those, unstructured grids offer the greatest flexibility for modeling of complex geometries and the generation of unstructured grids is largely automatic. In contrast, fully continuous block-structured grids, where all grid lines are at least C^1 continuous across block faces (here referred to as multi-block grids), are more difficult to generate but are the most suitable for simulations of viscous flows. In addition, flow solvers for structured grids typically require less memory than those for unstructured grids, and can take full advantage of various convergence acceleration schemes (e.g. multigrid) and fast solvers for implicit discretization (e.g., line Gauss-Seidel, approximate LU and ADI schemes). In this study, we use semi-automatically generated multiblock grids (i.e., shape of blocks is automatically determined but grid-topology or block-structure needs to be specified beforehand).

Turbulence models used in simulations of internal flows in complicated geometries must be able to model flows involving separation and adverse pressure gradients. One such model is the k-ω model of Wilcox(1994a and 1994b). This model has several desirable features. One important feature is that it does not require distance to a nearest wall as a parameter. Second, the low Reynolds number version of the model can be used to model transition (Wilcox, 1994b). Finally, as both k and ω are well behaved numerically, stiffness associated with low-Reynolds number k-ε turbulence models is eliminated. In addition work by Chima (1996) shows the model to be useful for predicting heat transfer over turbine blades.

The present code, Glenn-HT, has been used for many types of flowfields including turbine blades, tip clearance flows, film cooling holes, and coolant channels. For the present discussion, flow and heat transfer results for the following geometries will be addressed: (1) rectangular ducts with a 180 degree turn, Rigby et al. (1996a), (2) a straight channel with ribs and bleed holes, Rigby et al. (1997), and (3) a rotating channel with ribs and a 180 degree turn, Rigby (1998).

There is extensive literature available on the study of rib roughened surfaces. Some examples of experimental work include Boyle (1984), Chen et al. (1996), Hibbs et al. (1997), Rau et al. (1996), Shen et al. (1994), and Taslim et al. (1995). Some examples of numerical work in this area includes Stephens et al. (1995a, 1995b), Stephens et al. (1996), and Stephens and Shih (1997).

For rib roughened channels with rotation, the work of Park (1996), and Park et al. (1997) offer local mass transfer distributions. Regional measurements of heat transfer are available in Wagner et al. (1992).

2. Overview of Glenn-HT Code (excerpts from Glenn-HT Manual)

2.1. Introduction

This code is a general purpose flow solver, designed for simulations of flows in complicated geometries. The code is based on the TRAF3D code, an efficient computer code designed for simulations of flows in turbine cascades (Arnone et al. 1991). The Glenn-HT code employs the full compressible Navier-Stokes equations. It uses a multi-stage Runge-Kutta scheme to march in pseudo time. The code utilizes multi-grid and implicit residual smoothing to accelerate convergence to steady state. Convective and diffusive fluxes are computed using central differencing. Artificial dissipation is added to prevent odd-even decoupling. The discretization is formally second order accurate. To handle complex geometries, the code uses contiguous multiblock grid systems but has the added capability of handling grids with non-contiguous grid lines across branch cuts. For contiguous systems, all internal boundaries are conservative. The Glenn-HT code was described in detail by Steinthorsson et al. (1993). Note that the had been referred to as the TRAF3D.MB code until the name was recently changed. Some aspects of the formulation used in the code are the same as those described by Arnone et al. (1991). For the present computations the code was fitted with the low Reynolds number *k*-ω model of Wilcox (1994b).

2.2. Governing Equations

2.2.1. Navier-Stokes Equations

The compressible Navier-Stokes equations for a Newtonian fluid can be written as follows (see *e.g.*, Panton, 1984):

Continuity equation,

$$\frac{\partial \rho}{\partial t} + \nabla \bullet (\rho \boldsymbol{u}) = 0, \tag{2.1}$$

momentum equation,

$$\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \nabla \bullet (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \nabla \bullet \tau, \tag{2.2}$$

and total energy equation,

$$\frac{\partial}{\partial t}(\rho e) + \nabla \bullet (\rho \boldsymbol{u}(e + p/\rho)) = \nabla \bullet \boldsymbol{q} + \nabla \bullet (\tau \boldsymbol{u}), \tag{2.3}$$

where ρ is the density, $\boldsymbol{u}=(u,v,w)^T$ is the velocity, p is the thermodynamic (static) pressure, τ is the deviatoric viscous stress tensor, e is the total energy per unit mass, and $\boldsymbol{q}$ is heat flux due to diffusion. The last three of these quantities are given as

$$\tau = -\frac{2}{3}\mu(\nabla\bullet\boldsymbol{u})\boldsymbol{I} + \frac{1}{2}\mu(\nabla\otimes\boldsymbol{u} + (\nabla\otimes\boldsymbol{u})^T), \tag{2.4}$$

where $\boldsymbol{I}$ is the 3x3 identity matrix,

$$e = \hat{u} + \frac{1}{2}(\boldsymbol{u}\bullet\boldsymbol{u}), \tag{2.5}$$

and

$$\boldsymbol{q} = -k\nabla T, \tag{2.6}$$

where μ is the molecular (dynamic) viscosity, k is the thermal diffusivity, $\hat{u}$ is the internal energy of the gas and T is the (absolute) temperature.

In the above equations, the operator ⊗ is defined in terms of matrix multiples as

$$\boldsymbol{u}\otimes\boldsymbol{v} = (\boldsymbol{u}(\boldsymbol{v}^T))^T \tag{2.7}$$

where u and v are three-component column vectors. Correspondingly, the operator $\nabla\otimes$ is defined as

$$\nabla\otimes\boldsymbol{v} = (\nabla(\boldsymbol{v}^T))^T \tag{2.8}$$

where $\boldsymbol{v}$ is a three-component column vector. The divergence operator, $\nabla\bullet$, is defined as

$$\nabla\bullet\boldsymbol{v} = (\nabla^T)\boldsymbol{v} \tag{2.9a}$$

when $\boldsymbol{v}$ is a column vector and as

$$\nabla\bullet\boldsymbol{v} = ((\nabla^T)\boldsymbol{v})^T \tag{2.9b}$$

when $\boldsymbol{v}$ is a three-column matrix.

The gas is treated as ideal gas, *i.e.*, $d\hat{u} = c_v dT$, where c_v is the constant-volume specific heat. In addition, the gas is assumed to be calorically perfect so that c_v is constant, which is reasonably well valid for flows in turbomachinery–excluding, perhaps, flows in the combustor. Thus the total energy can be written as

$$e = c_v T + \frac{1}{2}(u \bullet u) \tag{2.10}$$

where the reference internal energy has been taken to be zero at a temperature of absolute zero.

For ideal gas, the equation of state can be written as

$$p = \rho \mathcal{R} T \tag{2.11}$$

and the speed of sound is

$$c = \sqrt{\gamma p / \rho} \tag{2.12}$$

where $\mathcal{R}$ is the gas constant for the particular gas mixture in use and γ is the ratio of specific heats.

The dynamic viscosity, μ, of the gas is assumed to vary with temperature as

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^{0.7} \tag{2.13}$$

where μ_0 is reference viscosity at temperature T_0. Thermal conductivity of the gas is then taken as

$$k = \frac{c_p \mu}{Pr} \tag{2.14}$$

where c_p is the constant–pressure specific heat and Pr is the *Prandtl number.*

2.2.2. Formulation for Rotating Reference Frame

To facilitate simulation of flows in rotating machinery, the governing equations are transformed to a coordinate system in a rotating reference frames. The advantage of this approach is that the grid system for the rotating geometry is stationary in the moving coordinate system, which leads to savings in computational effort and makes it possible to compute steady-state flows in the rotating frame of reference. The disadvantage is that source terms arise in the governing equations.

Several different formulations have been proposed for simulating flows in rotating reference frames. Here, the formulation proposed by Chima and Yokota (1988) is used. In this formulation the governing equations are mapped to coordinate system moving with the rotating reference frame. The velocity vectors used as dependent variables are, however, the absolute velocity. Unlike the formulation of Chima and Yokota, the total energy is the absolute total energy, i.e., based on absolute velocities, rather than rothalpy. Additional details can be found in Chima and Yokota (1988) and in Steinthorsson (1991) and Steinthorsson et al. (1991).

2.3. Turbulence Model

When working with complicated geometries it is advantageous to use a set of equations describing the turbulence that does not require the computation of the dimensionless distance to the wall y^+. The boundaries between adjacent blocks should be free to cut across boundary layers and regions of high shear. Having to carry information on solid walls and dealing with corners requires communication of much information that is quite cumbersome and time consuming both in terms of programming and CPU time.

The k-ω turbulence model developed by Wilcox (1994a,1994b) satisfies our requirements. Modifications by Menter (1992,1993) improved the robustness of the model. Chima (1996) incorporated some of the latter modifications to the turbulence model and presented some applications of this model in the context of a Navier-Stokes solver. In fact it is the three-dimensional variation to the formulation adapted by Chima that has been utilized in this paper. Chima has shown the model to possess very good convergence properties. He also showed that the model performs well in predicting the rate of heat transfer from a simulated flat plate and turbine blades under various conditions. Below we present the dimensionless equations describing the turbulence in tensor notation.

$$(\rho s_i)_{,t} + (\rho s_i u_j + q_{ij})_{,j} = (P_i - D_i) \tag{15}$$

$$q_{ij} = -Re^{-1}\left(\mu + \frac{\mu_t}{\sigma}\right)s_{i,j} \quad \text{j=1,3} \tag{16}$$

where $s_1=k$ and $s_2=\omega$ also $\mu_t=\alpha^* \frac{\rho k}{\omega}$.

The source terms, P_i, of equation (1) are defined as

$$P = \begin{bmatrix} Re^{-1} \cdot \mu_t 4\Omega^2 - \frac{2k\rho}{3}(\nabla \bullet V) \\ \rho\alpha\left[\alpha^* 4\Omega^2 - \frac{2}{3}\omega(\nabla \bullet V)\right] \end{bmatrix} \tag{17}$$

where Ω is the magnitude of the vorticity. The destruction terms, D_i, are given by

$$D = \begin{bmatrix} \beta^* \rho\omega k \\ \beta\rho\omega^2 \end{bmatrix} \tag{18}$$

The coefficients appearing in the model are

σ=2.0, β=3/40, $\beta^*=0.09F_\beta$, $\alpha=(5/9)(F_\alpha/F_\mu)$, and $\alpha^*=F_\mu$, where

$$F_\beta = \frac{\frac{5}{18}+\frac{Re_T{}^4}{R_\beta}}{1+\frac{Re_T{}^4}{R_\beta}}\ ,F_\alpha = \frac{\alpha_0+\frac{Re_T}{R_\omega}}{1+\frac{Re_T}{R_\omega}}\ ,F_\mu = \frac{\alpha_0^*+\frac{Re_T}{R_k}}{1+\frac{Re_T}{R_k}} \tag{19}$$

Above α_0=0.1, $\alpha_0{}^*$=0.025,R_β=8, R_ω=0.27, R_k=6 and $Re_T = \rho k/\mu\omega$.

2.4. Boundary Conditions

The types of boundary conditions encountered in solving the problem at hand are as follows:

1) Inlet: The inlet boundary condition for subsonic flows is treated by specifying the relative total temperature and relative total pressure as well as the inlet angle profiles. The outgoing Riemann invariant is extrapolated to the inlet from within. The total temperature and total pressure profiles are chosen to produce specified velocity and temperature profiles.

2) Exit: At the exit boundary, for subsonic flow, the pressure is specified while all other conditions are extrapolated from within.

3) Walls: At walls, the normal pressure gradient is set to zero, either the temperature or heat flux is specified, and the no-slip condition is enforced. The density and total energy are computed from the pressure and the temperature. The boundary conditions for the turbulence quantities are k=0 and

$$\omega = S_R\frac{\partial u}{\partial y}\bigg|_{wall} \quad \text{where, } S_R = \begin{bmatrix} \left(\frac{50}{K_R}\right)^2, K_R < 25 \\ \frac{100}{K_R}, K_R \ge 25 \end{bmatrix} \tag{20}$$

and K_R is the equivalent sand grain roughness height in turbulent wall units. This condition on ω is attributable to Wilcox. A discussion on the physical reasoning behind the condition can be found in Wilcox (1988).

An upper limit is imposed on the value of ω at the wall using the following boundary condition suggested by Menter(1993) and found effective by Chima(1996),

$$\omega_{max} = \frac{10}{Re}\cdot\frac{6}{\beta}\cdot\frac{\nu}{\Delta y^2} \tag{21}$$

(22)

(23)

2.5. Time Stepping Scheme

The discretized governing equations are marched in time to steady state using a Runge-Kutta time stepping scheme. Typically a four or five stage scheme is used. To accelerate convergence to steady state, the higher order time accuracy of the schemes is sacrificed in favor of near optimal high-frequency damping properties of the scheme, which improves effectiveness of multi-grid schemes without affecting the accuracy of the final steady state solution. The schemes used here were originally proposed for the Euler equations by Jameson, Schmidt and Turkel (1981) and have been widely used.

2.6. Convergence Acceleration

2.6.1. Local Time Stepping

For steady flows, Δt is viewed simply as a relaxation parameter that we can take advantage of to accelerate the progression of the computed solution to steady state. Thus, rather than use the same time step for every equation, time accuracy is sacrificed and a "local" time step is used. For non-dissipative, hyperbolic systems of equations, steady state is reached once all unsteady waves have been expelled out of the domain. Thus, the local time step is chosen based on the local stability limit which, for such systems, is proportional to the time it takes a characteristic wave to travel across a cell, *i.e.*, the local time step is computed from constant multiple of the local CFL number. For the Navier-Stokes equations, the presence of viscous terms lead to more complicated stability limits so that empirical formulas are used. (see Jameson, *et al.*, 1981, Arnone, *et al.*, 1990, and Swanson and Turkel, 1987)

For a given grid system, higher flow velocities lead to larger eigenvalues and, therefore, to smaller allowable time step size. Higher Reynolds number also reduces the allowable time step size. In boundary layers with highly clustered grids, the viscous time step limits the allowable time step size.

2.6.2. Implicit Residual Smoothing

To further improve stability characteristics and rate of convergence to steady state, a technique known as implicit residual smoothing (IRS) is applied in every stage of the Runge-Kutta time stepping scheme. Implicit Residual Smoothing was first introduced by Lerat (1979) for Lax-Wendrof-type of schemes and by Jameson (1983a,b) for multi-stage schemes. Modifications to the scheme by Jameson were introduced by Swanson and Turkel (1987) and by Martinelli and Jameson (1988) to improve the performance of the schemes on high aspect-ratio grids such as used in simulations of viscous flows. Here the scheme of Martinelli and

Jameson is used. For an in-depth discussion on implicit residual smoothing, see Swanson and Turkel (1997).

For multi-block grid systems, the Implicit Residual Smoothing is applied independently within each block. For relatively few blocks, this is certainly adequate and does not hamper performance (see *e.g.*, Steinthorsson, *et al.* 1993). For a large number of blocks, performance gain due to IRS may be affected, in particular if the blocks are small. However, the cost of coupling the blocks in the IRS will likely offset any gain in performance.

Finally, note that the *boundary conditions* on the smoothed residual are needed. Here the systems of equations are solved assuming that the smoothed residual in the ghost cells is zero. Several other approaches have been tried, including implicit treatment of boundary conditions. These approaches have been found to provide benefit in some cases but to be detrimental in others. Overall, the simple boundary treatment currently used has proven to be the most robust.

2.6.3. Multigrid

Multigrid convergence acceleration involves the use of a sequence of coarser and coarser grids to accelerate the evolution of the computed fine-grid solution to steady state. The multigrid concept can be traced at least back to the work of Fedorenko (1964). Multigrid was introduced to Computational Fluid Mechanics primarily through the work of Brandt (see Brandt, 1977), Jameson, Ni, (see Ni, 1981) and Turkel. The multigrid scheme used in the current code is based on the work of Jameson (1983) and was formulated by Arnone and Swanson (1988) in the original TRAF2D code.

Although the available theory behind multigrid methods will not be reviewed in detail, it is useful for readers unfamiliar with the method to review the underlying ideas. For an elementary but enlightening discussion, the interested reader is directed to a tutorial by Briggs (1987). For a more advanced discussion of multigrid method applied to Computational Fluid Dynamics, see *e.g.*, Hacbusch (1982).

3. Multi-Block Grid Generation Techniques

For the numerical solution of problems with complex geometry, grid generation becomes a very high priority. So important is the role of grid generation that it dictates the type of solver to be used. Currently, the two most popular types of solvers are structured multi-block and unstructured tetrahedron. Other techniques include overlapping structured grids, known as Chimera grids (see Steger); as well as a combination of structured and unstructured grids, i.e. DRAGON grids (see Kao and Liou). If the question of grid generation is set aside, the structured multi-block approach is the clear choice in terms of numerical efficiency and memory usage. Regardless of how fast or inexpensive computers become, this advantage will remain.

The generation of unstructured tetrahedron grids is currently regarded as requiring very few man hours relative to other types of grids. However, for grids which require anisotropic grid resolution, the complexity in grid generation remains. The generation of tetrahedrons with aspect ratio far from unity is a developing technology and not widely available. In addition, the accuracy of solvers on these types of grids is not well documented. In particular, the prediction of skin friction and heat transfer may require significantly more resolution than a hexahedron mesh. Ideas such as generating prismatic meshes near viscous boundaries may improve accuracy, but will further complicate the grid generation process.

One could assume that structured multi-block grid generation will remain too tedious and labor intensive. It is much more likely, however, that fully automatic structured multi-block grid generation will be realized in the near future. The most difficult task to automate is the topology generation, that is how the blocks should be arranged and connected to best resolve the features of the problem. If one imagines that each block is a cell in an unstructured hexahedron mesh, then it becomes apparent that unstructured hexahedron mesh generators such as CUBIT or HOUDINI, which are primarily used for finite element analysis in solids, may be exploited to generate the topology which could then be used in an existing structured multi-block grid generation programs such as GRIDGEN or GridPro.

Currently, the GridPro software from Program Development Corporation is used extensively at NASA Glenn for turbine related calculations. Each new geometry requires the development of an appropriate topology. Some general guidelines which help lead to high quality grids are as follows:

- Limit topological singularities to one away from optimal. For instance, in 2D four blocks is optimal so having three or five would be acceptable.
- Have grid lines follow the surface when ever possible. This is especially important for viscous grids, to keep small grid spacings from being introduced into the main flow path where they are not needed.
- Keep topological singularities off boundaries to maximize grid quality where numerical accuracy is most needed.

4. Block Merging with the Method of Weakest Descent

4.1. Introduction

A methodology for automatic block merging is developed to the point where a complicated multi-block grid system is supplied and a merged system with connectivity information is returned. The objective of the method is to produce the minimum number of blocks. The recently developed Method of Weakest Descent (MWD) is described and its application to representative test cases is presented, Rigby (1996) and Rigby et al (1997). The MWD is based on the premise that the more internal faces of a multi-block grid that can be removed, the lower the final number of blocks. With each internal face that is removed, other faces are disqualified from being removed as a result of the merging of two blocks. The MWD chooses which internal face to remove so as to minimize the number of disqualified faces, thus diminishing the set of available faces at the slowest rate (i.e. weakest descent). Reducing the available faces at the slowest rate allows more blocks to be merged before all valid internal faces are removed. When more than one candidate disqualifies the same number of other faces, the choice among those candidates is made randomly. Because of this randomness, each application of the MWD can produce different results. Many tests have been done, with the number of initial blocks ranging from 12 to 7936. The tests have shown that, for most cases, every application of the MWD produces a result near the expected minimum. Repeated application increases the likelihood of realizing the actual minimum.

4.2. Description of the Method of Weakest Descent

The MWD procedure is fully three dimensional and uses only the connectivity data so it requires very little memory. The method progresses by successively merging blocks which result in logically rectangular blocks. No invalid blocks are formed at any stage of the merging procedure. The patch which joins two blocks that can be merged is referred to as a candidate. The removal of a candidate implies the merging of the blocks on either side of it. In general, the objective is to deplete the set of candidates available for removal at the slowest rate possible, thus allowing for the most blocks to be merged. It is this notion of depleting the set of candidates available for removal at the slowest rate possible which led to the name of the algorithm. It is important to realize that when two blocks are merged not only is the candidate between the blocks lost, but also several other candidates may be disqualified. The key is to, at each stage, merge the blocks which disqualify the fewest candidates. The discussion of the simple circle in a box case, in the results section, covers this point in more detail.

A flow chart for the algorithm is shown in figure 1. The thick line with filled arrows shows the main loop which defines the algorithm. Each execution of this loop is referred to as a stage and represents the reduction in the number of blocks by one.

Each stage of the procedure consists of the following seven steps:

1. Join any group of patches which cover the entire face of both neighboring blocks.
2. Update patch array which relates patches to blocks, and block faces. With the connectivity data, given the patch number, one finds the blocks and associated indices. The patch array returns the patch number given the block and face information.
3. From the set of internal faces, determine those which are candidates for removal. Several straight forward tests are done to ensure that merging two blocks will result in a logically rectangular block. One of the less obvious tests is to make sure that a patch which has the same block on both sides, as in an O-type grid, is not taken to be a candidate.
4. If there are any candidates, continue.
5. Grade each candidate for removal based on how many other candidates would be disqualified by its removal. The grade given a candidate is simply the difference between the current number of candidates and the number that would exist if the two blocks were merged. It is important to include the production of candidates, which may result after groups of patches are joined, in the grade. Also, the loss of a candidate due to the formation of an O-type grid is not included in the grade to encourage the formation of O-type grids.
6. Remove a low grade face and merge the neighboring blocks. If there is more than one candidate with the same low grade, then one of them is chosen randomly.
7. Return to step 1 and repeat until no candidates for removal are found.

In the present work, grading of candidates was based solely on how many other candidates would be disqualified. Additional grading criterion could also be added such as: setting an upper limit on block size to meet a memory constraint, or setting a lower limit on the number of blocks based on number of available CPUs, or consider the size of the resulting block relative to other blocks to enhance load balancing. Then, depending on the reliability of the grading procedure, the candidate with the lowest score may be chosen or the random choice could be made from a range of scores. In fact, the random choice could be designed in such a way as to favor the candidates with lower grades. These ideas will be explored in future work.

4.3. Sample Applications of the MWD

4.3.1. Overview

First, a very simple example of a circle in a box is shown to explain the progression of an application of the Method of Weakest Descent. Then, several complex examples are shown.

For each of the three complex examples, the topology is shown to demonstrate the complexity. The effectiveness of the method is presented in terms of the probability versus the final number of blocks (where the probability refers to the probability of realizing a particular final number of blocks with each application of the MWD). This probability is calculated by performing 500 realizations of the MWD; then, the number of times each final number of blocks is realized is divided by 500.

To further demonstrate the effectiveness of the MWD, two other grading procedures are also presented. First, the MWD was run as previously described. The grading procedure was then disabled with the candidate for removal chosen randomly from the entire set of valid internal faces at each stage. Finally, instead of choosing the low grade, the high grade candidate was chosen.

A discussion on the CPU and memory requirements is also presented with regard to the third complex example.

4.3.2. Circle in a Box

Figure 2 shows a topology for a circle in a box. This case is useful to demonstrate the step by step progression of the MWD. The solid lines in figure 2 show the outline of blocks that will produce a high quality grid for this problem. Near the flat walls the grid will be nearly cartesian, while near the circle the grid will be nearly polar.

Figure 3 shows the state of the blocks at each stage of the MWD as the grid is merged from the initial twelve blocks to the minimum of five blocks. The number in the gray arrow to the left of each configuration refers to the stage. The initial configuration is shown in the upper left of figure 3. Initially, each patch is a candidate and receives a grade of either 3 or 5. One of the candidates with grade 3 is chosen randomly and is removed, as shown at stage 1 in figure 3. In stage 1, the circled candidate from the initial configuration has been removed, and two other patches which were previously candidates were disqualified as indicated by the x's. The grades of patches in neighboring blocks has also been updated to reflect the change in the topology. The progression to stage 2 is accomplished by randomly choosing one of the patches with a grade of 2. Removal of candidates continues until stage 7, when no more candidates exist. The final number of blocks is the initial number of blocks minus the number of stages, that is 5.

Figure 4 shows the number of candidates available at each stage drawn in figure 3. At each stage, the slope of this curve is equal to the negative of the grade given the candidate removed, except for stage 6 where a candidate of grade 2 was removed, but a loss of 3 was incurred. The reason for this is because, as mentioned earlier, the loss of a candidate due to the formation of an O-type grid is not included in the grade. The purpose of figure 4 is to clearly demonstrate the need to minimize the rate at which candidates are depleted so as to allow the maximum number of stages. Recall that each stage marks the reduction in the number of blocks by one.

If at each stage the highest grade candidate were chosen for removal, one would find that the final number of blocks would be eight, since only four stages would be possible.

4.3.3. Internal Passage with Ribs and Bleed

The final example is a complex internal coolant passage with ribs and bleed holes. Figure 9a shows the surface grid constructed using the GridPro software[1]. Notice that, by taking full advantage of the multi-block capability, high quality grids can be formed around the ribs and holes. Figure 9b shows the block boundaries for the initial 282 block grid from GridPro. Figure 9c shows the block boundaries of the merged grid system, which has only 22 blocks.

Figure 10 shows the probability results for 500 realizations. The results are very encouraging. The likelihood of producing a result near the observed minimum is very high with every application of the MWD.

Figure 11 shows the number of candidates available at each stage versus the number of blocks at each stage. Ten realizations are shown for each grading procedure. Clearly, the MWD produces the slowest loss of candidates, thus allowing more blocks to be merged before the pool of candidates is depleted.

To investigate the CPU time requirements as the number of blocks increases to much higher levels, the topology for a case similar to this one was divided into additional blocks. The original topology had 124 blocks. The divided topologies had 992, 1776, and 7936 blocks. The original topology had 238 internal patches, while the divided topologies had

2440, 4700, and 21664, respectively. Figure 12 shows the CPU time required for a single application of the MWD for each of the divided topologies on three different machines. The tests were run on a Silicon Graphics Indigo 2 with an R4000 chip, a Silicon Graphics Power Challenge with an R10000 chip, and on the NAS CRAY C90 (eagle). From figure 11, it can be seen that on the order of 5 seconds is required to merge 1000 blocks with the R4000 chip or on the C90. The R10000 chip consistently performs five to six times faster than the other two machines. The CPU time is seen to increase like the square of the number of blocks. It is expected that improvements in the implementation of the algorithm should allow the CPU time to increase linearly with the number of blocks.

The reason that the C90 does not show an advantage over the Silicon Graphics machines is because the code is dominated by operations which are not vectorized. The case with 7936 blocks was only run on the C90 because of the memory requirements. In its current state, the memory requirement for the code increases like the number of blocks raised to the 1.6 power. For 1776 block case, 3.2 MW were required, while for the 7936 block case, 35 MW were required. It is expected that improvements in the implementation of the algorithm should allow the memory to increase linearly with the number of blocks.

4.3.4. Final Remarks

A methodology for automatically merging complicated grid systems has been described. The method is currently developed to the point where 9 grid system is input, and a merged system with the associated connectivity information is produced; no additional user intervention is required. The Method of Weakest Descent has been demonstrated as an effective automatic block merging procedure. A single application has been shown to very quickly produce a result near the expected optimum. With additional applications, results even nearer the optimum can result. It is recommended that a specified number of realizations be done for each case, with the specified number perhaps being dependent upon the initial number of internal patches.

Without the automatic block merging capability, topologies as complicated as those presented would probably not even be attempted. Calculations on topologies with hundreds of blocks would be tedious to set up and post process, and be numerically inefficient to run. The ability to automatically merge grid systems allows the person generating the grids to take full advantage of the multi-block methodology. The Method of Weakest Descent has been incorporated into the GridPro software and is routinely used for turbine related calculations done at the NASA Glenn Research Center.

References

Arnone, A., and Swanson, R. C. (1988), "A Navier-Stokes Solver for Cascade Flows," NASA CR No., 181682.

Arnone, A., Liou, M.-S., and Povinelli, L.A. (1990), "Transonic Cascade Flow Calculations Using Non-Periodic C-Type Grids," Proceedings, Computational Fluid Dynamics Symposium on Aeropropulsion, NASA Lewis Research Center, Cleveland, April, 1990.

Arnone, A., Liou, M.-S., and Povinelli, L.A. (1991), "Multigrid Calculation of Three-Dimensional Viscous Cascade Flows," AIAA-91-3238.

Boyle, R. J., "Heat Transfer in Serpentine Passages With Turbulence Promoters," 1984, NASA TM 83614.

Brandt, A. (1977) "Multi-Level Adaptive Solutions to Boundary Value Problems," Mathem. of Computation, Vol. 31., pp. 333-390.

Briggs, W.L. (1987), A Multigrid Tutorial, SIAM, Philadelphia, Pennsylvania, 1987.

Chen, Y., Nikitopoulos, D.E., Hibbs, R., Acharya, S. and Myrum, T.A., "Detailed Mass Transfer Distribution in a ribbed Coolant Passage," ASME paper 96-WA/HT-11.

Chima, R.V., and Yokota, J.W. (1988), "Numerical Analysis of Three-Dimensional Viscous Internal Flows," NASA TM 100878.

Chima, R. V. (1996), "A k-w Turbulence Model for Quasi-Three-Dimensional Turbomachinery Flows," AIAA-96-0248.

CUBIT Mesh Generation Research Project, Sandia National Laboratories, "http://sass577.endo.sandia.gov/SEACAS/CUBIT/Overview.html".

Dannenhoffer, J. F., "A Technique for Optimizing Grid Blocks," NASA CP-3291, Proceedings of the Workshop on Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamic (CFD) Solutions, held at NASA Lewis Research Center, (1995).

Dannenhoffer, J. F., "Automatic Blocking for Complex 3-D Configurations," NASA CP-3291, May 1995.

Gridgen, User's Manual for-," 1995, Pointwise, Inc.

GridPro^{TM},az3000 Users Guide and Reference Manual" Program Development Corporation, White Plains, NY, 1993.

Hackbusch W. (1982) Multigrid Methods and Applications, Springer Verlag, Berlin.

Hibbs, R.G., Acharya, S., Chen, Y., and Nikitopoulos, D.E., 1997, "Mass/Heat Transfer in a Ribbed Blade Coolant Passage with Cylindrical vortex Generators: The Effect of Generator-Rib Spacing," ASME paper 97-GT-161.

Houdini Mesh Generator, Algor Inc., "http://algor.com/houdini/"

Jameson, A., Schmidt, W. and Turkel, E. (1981), "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes,"

Jameson, A. (1983a), "Transonic Flow Calculations," MAE Report 1651, MAE Department, Princeton University, July 1983.

Jameson, A. (1983b), "Evolution of Computational Methods in Aerodynamics," J. Appl. Mech., Vol. 50.

Kao, K-H, Liou, M-S, "Advance in overset grid schemes: From Chimera to DRAGRON grids," AIAA Journal, Vol. 33, pp. 1809-1815.

Lerat, A. (1979), "Une Classe de Schemas aux Differences Implicites Pour Les Systemes Hyperboliques de Lois de Conservation," Comptes Rendus Acad. Sciences Paris, Vol 288A.

Martinelli, L. and Jameson, A. (1988), "Validation of a Multigrid Method for the Reynolds Averaged Equations," AIAA-88-0414.

Menter, Florian R., 1992, "Improved Two-Equation k-w Turbulence Models for Aerodynamic Flows," NASA-TM 103975.

Menter, F. R. (1993), "Zonal Two-Equation k-w Turbulence Model for Aerodynamic Flows," AIAA-93-2906.

Ni, R. H. (1981), "A Multiple Grid Scheme for Solving the Euler Equations," AIAA J., Vol. 20, pp. 1565-1571.

Panton, R.L. (1984), Incompressible Flows, John Wiley & Sons, New York.

Park, C.W., "Local Heat/Mass Transfer Distributions in Rotating Two-Pass Square Channels," Ph.D. Dissertation, Department of Mechanical Engineering, Texas A&M University, College Station, TX, December, 1996.

Park, C.W., Lau S.C., and Kukreja, R.T., 1997, "Heat/Mass Transfer in a Rotating Two-Pass Square Channel with Transverse Ribs," Journal of Thermophysics and Heat Transfer, Vol. 11.

Rau, G., Cakan, M., Moeller, D. and Arts, T., 1996, "The Effect of Periodic Ribs on the Local Aerodynamic and Heat Transfer Performance of a Straight Cooling Channel," ASME paper 96-GT-541.

Rigby, D.L., Ameri, A.A., and Steinthorsson, E., "Internal Passage Heat Transfer Prediction using Multiblock Grids and k-w Turbulence Model,"ASME 96-GT-188 (also, NASA CR 198540 and ICOMP-96-8).

Rigby, D. L., Ameri, A. A., and Steinthorsson, E., 1996a, "Internal Passage Heat Transfer Prediction Using Multiblock Grids and a k-w Turbulence Model," ASME paper 96-GT-188.

Rigby, D. L., 1996b, "Method of Weakest Descent for Automatic Block Merging," 15th Int. Conf. on Num. Methods Fluid Dynamics, Monterey, CA.

Rigby, D. L., Steinthorsson, E., and Coirier, W. J., 1997, "Automatic Block Merging Using the Method of Weakest Descent," AIAA paper no. 97-0197.

Rigby, D.L., Steinthorsson, E., and Ameri, A.A., "Numerical Prediction of Heat Transfer in a Channel with Ribs and Bleed,"ASME 97-GT-431.

Rigby, D.L., "Prediction of Heat and Mass Transfer in a Rotating Ribbed Coolant Passage with a 180 Degree Turn,"ASME 98-GT-329.

Shaw, J. A., and Weatherill, N. P., "Automatic Topology Generation for Multiblock Grids," Applied Math. and Comp., vol. 2, pp 355-388, 1992.

Shen, J.R., Wang, Z., Ireland, P.T., Jones, T.V., Byerley, A.R., 1994, "Heat Transfer Enhancement Within a Turbine Blade Cooling Passage Using Ribs and Combinations of Ribs with Film Cooling Holes," Journal of Turbomachinery, Vol.118, pp. 428-434.

Snyder and Roelke, R.J., 1990, "Design of an Air-Cooled Metallic High Temperature Radial Turbine," Journal of Propulsion and Power, Vol. 6, pp. 283-288.

Steger, J.L., "Thoughts on the Chimera method of simulation of three-dimensional viscous flow," NASA Lewis Research Center, Computational Fluid Dynamics Symposium on Aeropropulsion, pp. 1-10.

Steinthorsson, E., Liou, M.-S., and Povinelli, L.A. (1993), "Development of an Explicit Multiblock/Multigrid Flow Solver for Viscous Flows in Complex Geometries," AIAA-93-2380.

Steinthorsson, E. (1991), "Numerical Simulations of Viscous, Compressible Flows in Complicated Geometries," Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Steinthorsson, E. Shih, T. I-P. and Roelke R. J. (1991), "Computation of the Three-Dimensional Flow and Heat Transfer with a Coolant Passage of a Radial Turbine," AIAA-91-2238.

Stephens, M.A., Shih T. I-P., Civinskas, K.C., 1995a, "Effects of Inclined Rounded Ribs on Flow and Heat Transfer in a Square Duct," AIAA paper no. 95-2115.

Stephens, M.A., Shih, T. I-P., Civinskas, K.C., 1995b, "Computation of Flow and Heat Transfer in a Rectangular Channel with Ribs," AIAA paper no. 95-0180.

Stephens, M.A., Chyu, M.K. , Shih, T. I-P., Civinskas, K.C., 1996, "Calculation and Measurements of Heat Transfer in a Square Duct with Inclined Ribs," AIAA paper no. 96-3163.

Stephens, M.A., Shih, T.I-P., 1997, "Computation of Compressible Flow and Heat Transfer in a Rotating Duct with Inclined Ribs and a 180-Degree Bend," ASME paper 97-GT-192.

Suo, M., 1978, "Turbine Cooling in the Aerothermodynamics of Aircraft Gas Turbines," AFAPL TR 78-52.

Swanson, R.C. and Turkel, E. (1987), "Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations," AIAA-87-1107.

Swanson, R.C., and Turkel, E. (1997), Introduction to Computational Fluid Dynamics, Part I., "Multistage Central Differencing Schemes for the Euler and Navier-Stokes Equations," Lecture Notes, Von Karman Institute.

Taslim, M.E., Li, T., Spring, S.D., 1995, "Experimental Study of the Effects of Bleed Holes on Heat Transfer and Pressure Drop in Trapezoidal Passages With Tapered Turbulators," Journal of Turbomachinery, Vol.117, pp. 281-289.

Taylor, J.R., 1980, "Heat Transfer Phenomena in Gas Turbines," ASME 80-GT-172.

Wagner, J.H., Johnson, B.V., Graziani, R.A., Yeh, F.C., "Heat Transfer in Rotating Serpentine Passages with Trips Normal to the Flow," Journal of Turbomachinery, Vol. 114, pp.847-857.

Wilcox, D.C. and Rubesin, M.W. (1988), "Progress in Turbulence Modeling for Complex FLow Fields Including the Effect of Compressibility," AIAA Journal, Vol. 26, pp. 1299-1310.

Wilcox, D.C. (1994a), Turbulence Modeling for CFD, DCW Industries, Inc., La Canada, CA 1994.

Wilcox, D.C. (1994b), "Simulation of Transition with a Two-Equation Turbulence Model," AIAA Journal, Vol. 32, No. 2., pp.247-255.

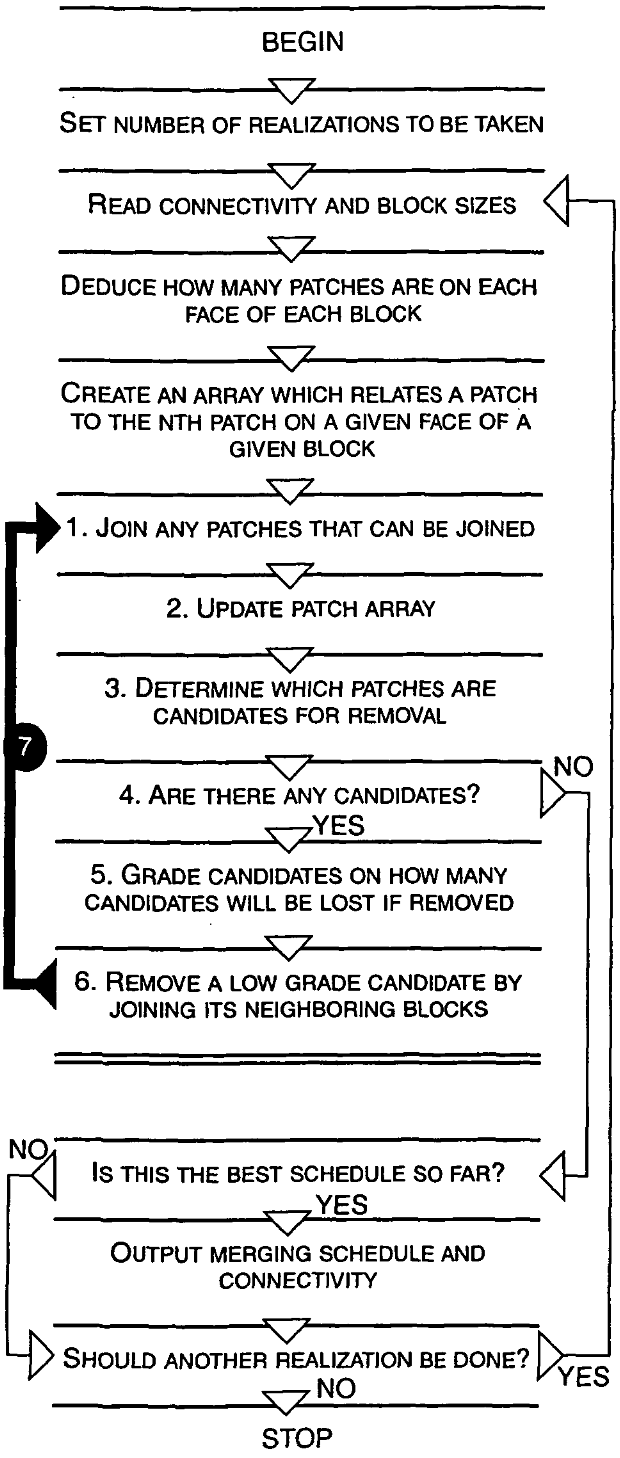
BEGIN
SET NUMBER OF REALIZATIONS TO BE TAKEN
READ CONNECTIVITY AND BLOCK SIZES
DEDUCE HOW MANY PATCHES ARE ON EACH FACE OF EACH BLOCK
CREATE AN ARRAY WHICH RELATES A PATCH TO THE NTH PATCH ON A GIVEN FACE OF A GIVEN BLOCK
1. JOIN ANY PATCHES THAT CAN BE JOINED
2. UPDATE PATCH ARRAY
3. DETERMINE WHICH PATCHES ARE CANDIDATES FOR REMOVAL
7
4. ARE THERE ANY CANDIDATES?
NO
YES
5. GRADE CANDIDATES ON HOW MANY CANDIDATES WILL BE LOST IF REMOVED
6. REMOVE A LOW GRADE CANDIDATE BY JOINING ITS NEIGHBORING BLOCKS
NO
IS THIS THE BEST SCHEDULE SO FAR?
YES
OUTPUT MERGING SCHEDULE AND CONNECTIVITY
SHOULD ANOTHER REALIZATION BE DONE?
YES
NO
STOP

Figure 1. Flow chart for the Method of Weakest Descent. Thick line shows main loop.

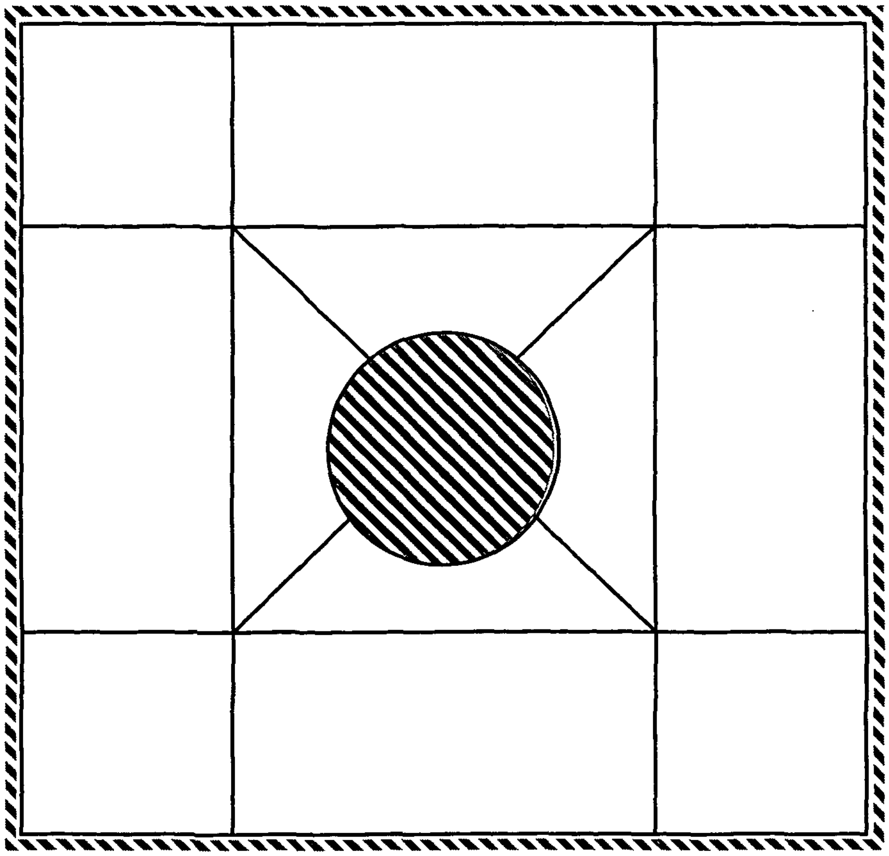

Figure 2. Topology for twelve block circle in a box case.

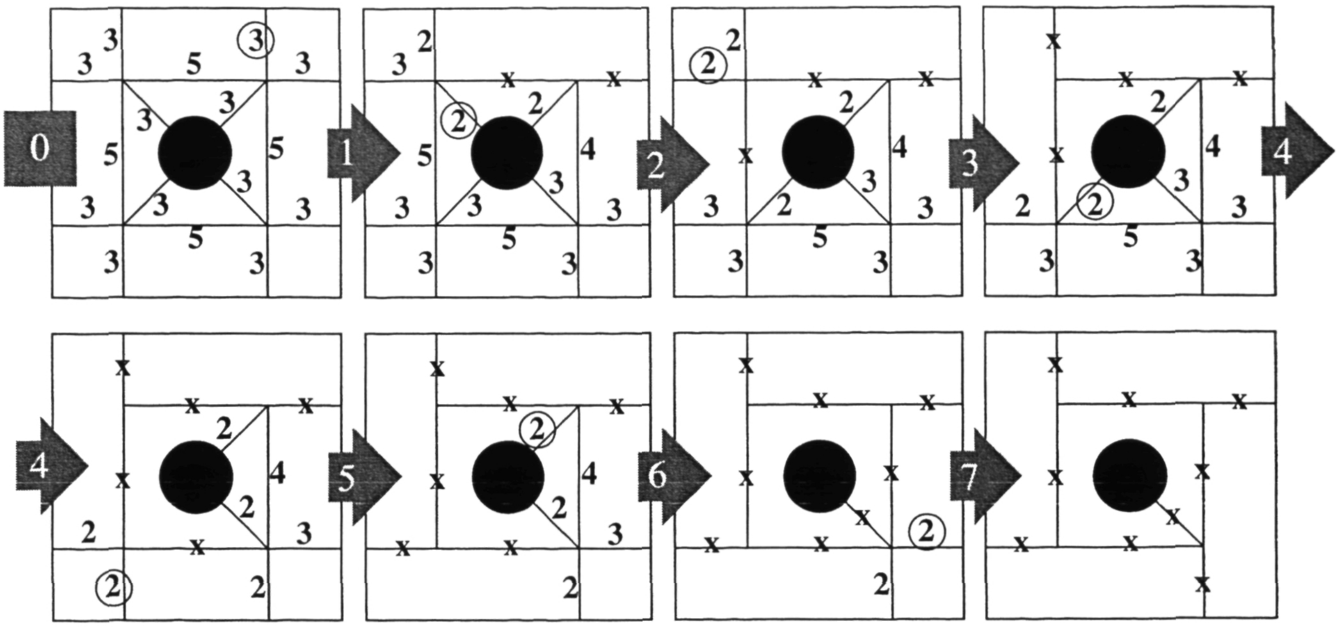

Figure 3. Stages for a circle in a box case. Grade shown near candidate. Invalid patches have x. Circle shows candidate which is about to be removed.

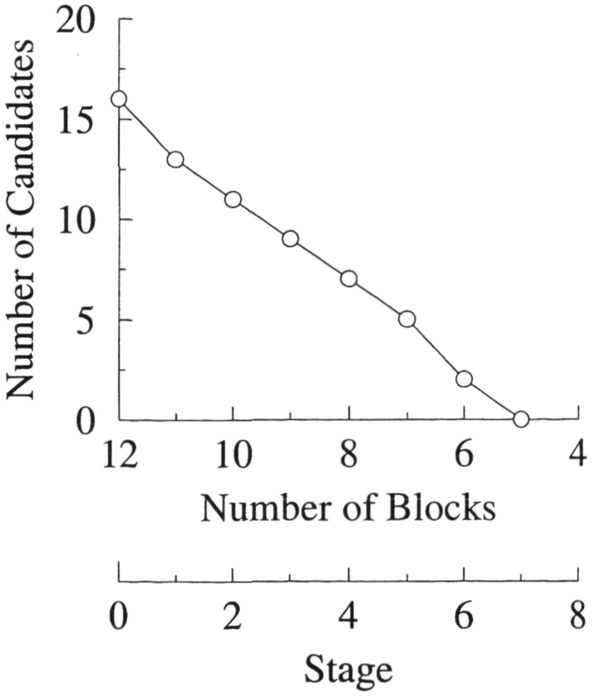

Figure 4. Number of available candidates at each stage for the circle in a box example.

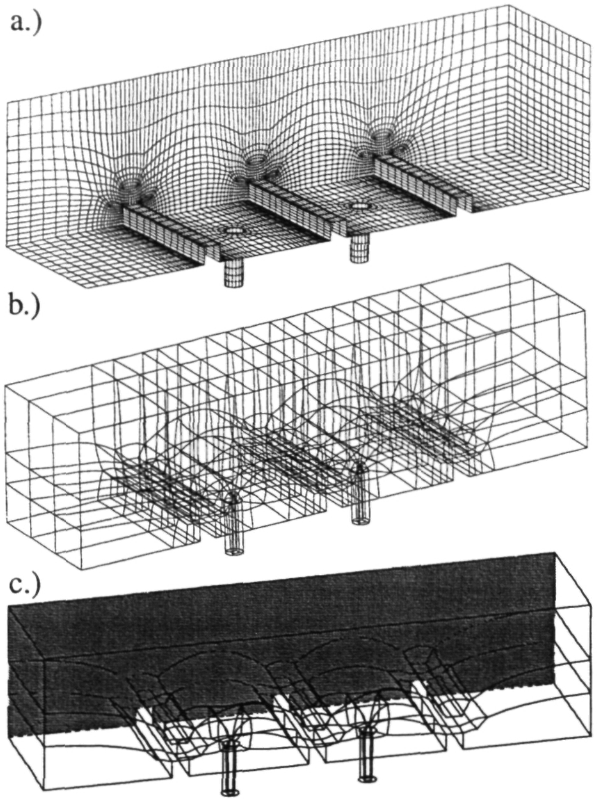

Figure 9.Internal passage with ribs and bleed topology (282 blocks merged to 22).

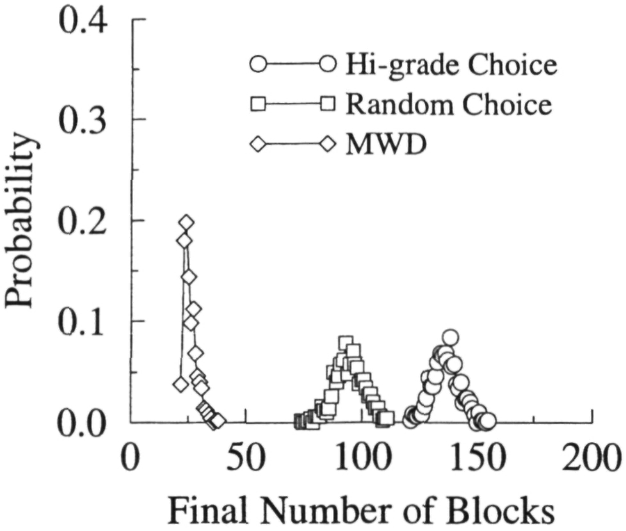

Figure 10. Merging results for internal passage with ribs and bleed.

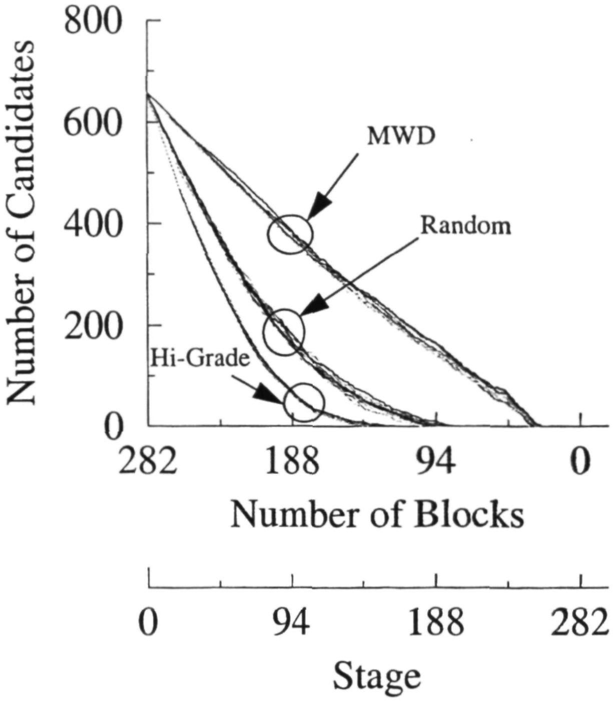

Figure 11. Number of available candidates at each stage for internal passage with ribs and bleed.

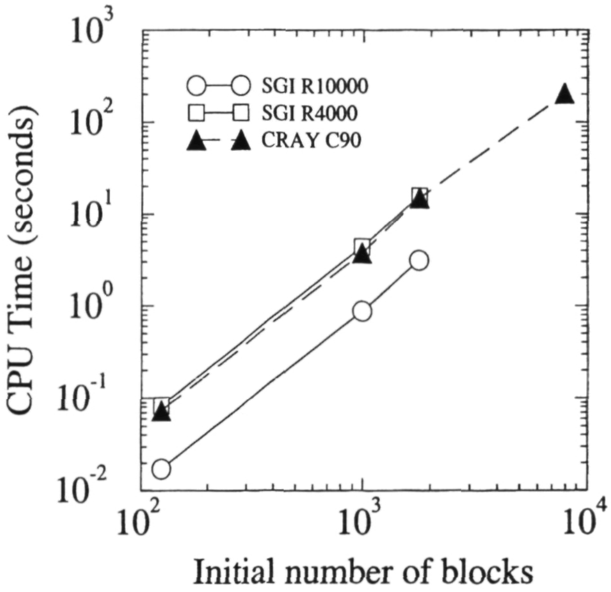

Figure 12. CPU time for a single application of the MWD versus the initial number of blocks.