INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 25, 225–243 (1997)

SOLUTION ALGORITHM FOR LOW-PECLET-NUMBER REACTING FLOW

CHRISTOPHER D. MOEN*

Thermal and Plasma Processes Department, Sandia National Laboratories, Livermore, CA 94550, U.S.A.

SUMMARY

An enhanced solution strategy based on the SIMPLER algorithm is presented for low-Peclet-number mass transport calculations with applications in low-pressure material processing. The accurate solution of highly diffusive flows requires boundary conditions that preserve specified chemical species mass fluxes. The implementation of such boundary conditions in the standard SIMPLER solution procedure leads to degraded convergence that scales with the Peclet number. Modifications to both the non-linear and linear parts of the solution algorithm remove the slow convergence problem. In particular, the linearized species transport equations must be implicitly coupled to the boundary condition equations and the combined system must be solved exactly at each non-linear iteration. The pressure correction boundary conditions are reformulated to ensure that continuity is preserved in each finite volume at each iteration. The boundary condition scaling problem is demonstrated with a simple linear model problem. The enhanced solution strategy is implemented in a baseline computer code that is used to solve the multicomponent Navier–Stokes equations on a generalized, multiple-block grid system. Accelerated convergence rates are demonstrated for several material-processing example problems. © 1997 by John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Fluids, **25**: 225–243 (1997).

No. of Figures: 17. No. of Tables: 3. No. of References: 24.

KEY WORDS: Navier–Stokes; low Peclet number; chemical vapour deposition; low Mach number; finite volume

1. INTRODUCTION

Numerical simulation is a useful tool for studying heat and mass transfer in gas phase manufacturing of thin film materials. Numerical models predict the uniformity and deposition rate of material coatings and help design reactors and processes. Many applications occur at low pressures, such as etching and chemical vapour deposition.[1,2] Chemical diffusivities become large at low pressure and the Peclet number for mass diffusion becomes small. Segregated solution algorithms based on the SIMPLER algorithm[3] are commonly used for chemical vapour deposition modelling.[4–6] Though they can be slow, they are desirable for their simplicity and economic use of computer resources. Unfortunately, as the Peclet number decreases, such methods show markedly poor convergence rates.

Increased coupling between equations improves convergence for some flow regimes, as discussed in the review by Patankar.[7] Fully coupled methods[1,8] are used for reacting flow, but at the cost of storing large matrices. For many applications the coupled species transport equations are split from the heat and momentum equations.[9,10] For highly diffusive problems, equation coupling is not as important as spatial coupling. Convergence is degraded at low Peclet number by poor numerical

* Correspondence to: C. D. Moen, Thermal and Plasma Processes Department, Sandia National Laboratories, Livermore, CA 94550, U.S.A.

CCC 0271–2091/97/020225–19 $17.50

Received 17 August 1996
Revised 5 August 1996

propagation of information between physical boundaries. Difficulties arise at the inflow boundary where convection must balance diffusion to preserve specified species mass fluxes. Information propagates very slowly from such a boundary for a particular class of segregated solution algorithms with explicit updating of boundary conditions. Boundary condition equations should be implicitly coupled to interior equations with complete solution of the resulting linearized system.

The focus of this article is to both explain the physical processes that adversely affect the convergence rate of the SIMPLER algorithm at low Peclet number and modify the solution algorithm accordingly. The modifications to the solution algorithm are demonstrated by enhancing the convergence of a baseline computational fluid dynamics (CFD) code called CURRENT. The CURRENT code is an extensive reformulation of the TEACH[11] code by Evans.[4,12] He recast the governing equations in terms of a generalized body-fitted co-ordinate system with multiple-block grids and implemented multicomponent transport with gas phase and surface phase chemical reactions.

It will be shown that strengthening the coupling between physical boundaries and across internal grid block boundaries in the baseline code greatly enhances convergence. Furthermore, convergence rates are improved with only modifications to the species and pressure correction algorithms. The solution algorithm enhancements affect both the non-linear and linear parts of the SIMPLER algorithm. The non-linear part is modified by including the boundary condition equations in the linearization of the transport equations. Implicit coupling of boundary conditions allows the boundaries to directly communicate at each iteration, but only if the linearized system is fully solved. It is during the solution of the linear problem that information is numerically propagated. The convergence of the linear problem is accelerated by adding a matrix-free[13] gradient algorithm to the line relaxation scheme. The benefit of the matrix-free method is that implementation requires only minor modifications to the existing CURRENT code. The solution of the continuity equation must also be modified to keep up with the improved species algorithm.

The following sections discuss the transport equations, the baseline code and the developments which lead to the enhanced solution strategy. First the transport equations are presented and the baseline solution algorithm is discussed. Then low-Peclet-number convergence problems are demonstrated with a simple model problem. Lessons from the model problem lead to modification of the baseline solution algorithm and boundary condition treatment. Improved performance is presented for three example chemical reactor problems.

2. TRANSPORT EQUATIONS

The governing equations used in this work describe the conservation of mass and the transport of momentum, energy and chemical species and are suitable for chemically reacting flows with multicomponent transport. The low-Mach-number approximation is used where the pressure is split into dynamic and thermodynamic parts and the viscous dissipation term is dropped from the thermal energy equation. Only laminar flows are considered.

The differential form of the governing transport equations is given for the purpose of presentation. The axisymmetric governing equations are written in a cylindrical co-ordinate system where the axial co-ordinate is x, the radial co-ordinate is r and the circumferential co-ordinate is θ. The procedure for deriving the finite volume discretization in generalized co-ordinates from the differential equations is found in Reference 14.

Continuity

$$\frac{\partial}{\partial x}(\rho u r) + \frac{\partial}{\partial r}(\rho v r) = 0. \tag{1}$$

The axial and radial velocity components are u and v. The circumferential velocity is w.

Axial momentum

$$\frac{\partial}{\partial x}(\rho u^2 r) + \frac{\partial}{\partial r}(\rho uvr) + r\frac{\partial P}{\partial x} = \frac{\partial}{\partial x}(r\tau_{xx}) + \frac{\partial}{\partial r}(r\tau_{xr}). \tag{2}$$

Radial momentum

$$\frac{\partial}{\partial x}(\rho uvr) + \frac{\partial}{\partial r}(\rho v^2 r) + r\frac{\partial P}{\partial r} - \rho w^2 = \frac{\partial}{\partial x}(r\tau_{rx}) + \frac{\partial}{\partial r}(r\tau_{rr}) - \tau_{\theta\theta}. \tag{3}$$

Circumferential momentum

$$\frac{\partial}{\partial x}(\rho uwr) + \frac{\partial}{\partial r}(\rho vwr) + \rho vw = \frac{\partial}{\partial x}(r\tau_{\theta x}) + \frac{1}{r}\frac{\partial}{\partial r}(r^2\tau_{\theta r}). \tag{4}$$

The circumferential velocity can be rewritten in terms of the angular velocity so that the circumferential momentum equation more closely resembles a general SIMPLER transport equation.

Temperature

$$\begin{aligned}\frac{\partial}{\partial x}(\rho uTr) + \frac{\partial}{\partial r}(\rho vTr) = &\frac{\partial}{\partial x}\left(r\frac{\kappa}{C_\mathrm{p}}\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial r}\left(r\frac{\kappa}{C_\mathrm{p}}\frac{\partial T}{\partial r}\right) + \frac{\kappa r}{C_\mathrm{p}^2}\left(\frac{\partial T}{\partial x}\frac{\partial C_\mathrm{p}}{\partial x} + \frac{\partial T}{\partial r}\frac{\partial C_\mathrm{p}}{\partial r}\right)\\ &- \frac{r}{C_\mathrm{p}}\sum_{g=1}^{NS} C_{\mathrm{p},g}\left(j_{x,g}\frac{\partial T}{\partial x} + j_{r,g}\frac{\partial T}{\partial r}\right) - \frac{r}{C_\mathrm{p}}\sum_{g=1}^{NS} h_g\dot{\omega}_g.\end{aligned} \tag{5}$$

The temperature equation is derived from the energy equation by subtracting out the kinetic energy and chemical heats of formation. The mixture-averaged thermal conductivity is κ, species enthalpies are h_g and chemical kinetic mass source terms are $\dot{\omega}_g$. The mixture-averaged specific heat C_p has been moved inside the spatial derivative for the heat conduction term to make the equation appear as a general SIMPLER transport equation. The gas phase chemical species are referenced by the subscript g and summations occur over the total number of species, NS.

Species

$$\frac{\partial}{\partial x}(\rho uY_g r) + \frac{\partial}{\partial r}(\rho vY_g r) = -\frac{\partial}{\partial x}(rj_{x,g}) - \frac{\partial}{\partial r}(rj_{r,g}) + r\dot{\omega}_g. \tag{6}$$

The governing equations are closed by defining the diffusive transport terms[15] $\boldsymbol{\tau}$ and $\mathbf{j}_g$ and the chemical kinetic and thermophysical properties.[16–18]

3. BASELINE SOLUTION ALGORITHM

The baseline solution algorithm described in this section is the standard implementation of the SIMPLER algorithm in the CURRENT code without enhancements. The transport equations are discretized as finite volumes on a body-fitted, multiple-block, staggered grid. The equations are linearized and solved in a segregated manner using line relaxation. Boundary conditions are updated

explicitly. In the following discussions a basic understanding of the use of the SIMPLER algorithm to solve transport equations is assumed.

3.1. Linearization and solution

The transport equations are written as convection and diffusion operators with a source term:

$$\nabla \cdot (\rho \mathbf{u}\phi - \Gamma_\phi \nabla \phi) = S_\phi, \tag{7}$$

where ϕ is any solution variable, Γ_ϕ is a diffusion coefficient and S_ϕ is the source term containing the remaining physical terms. The collection of convection and diffusion terms forms the standard SIMPLER transport operator for the transport variable ϕ. The equations are discretized on a nine-point stencil in two dimensions on a staggered grid system. The linearization is performed by freezing coefficients of the transport operator and the source term. The matrix elements for one finite volume of the linearization of equation (7) are written as

$$A_\mathrm{p}\phi_\mathrm{p} = A_\mathrm{n}\phi_\mathrm{n} + A_\mathrm{s}\phi_\mathrm{s} + A_\mathrm{w}\phi_\mathrm{w} + A_\mathrm{e}\phi_\mathrm{e} + S, \tag{8}$$

where the subscripts 'p', 'n', 's', 'e' and 'w' denote positions in the implicit five-point stencil. The stencil points are shown in Figure 1 (see Section 4.2). The corner points used in cross-derivative terms are moved to the source term, so that the source term S contains the physical terms S_ϕ and numerical difference terms from the transport operator.

The steady state solution strategy for the governing equations consists of outer iterations on the non-linear problem. At each outer iteration of the non-linear problem a set of linear problems is solved with inner line relaxation iterations. The line relaxation scheme is localized on each grid block, but new information is passed immediately across block boundaries during the sweeping procedure. The procedure consists of three sequences of alternating northward and eastward sweeps.

In the segregated solution strategy for the multicomponent Navier–Stokes equations each equation is solved sequentially. First the pressure equation is solved, followed by the scalar transport equations: circumferential velocity, temperature and then chemical species. The number of species equations solved is one less than the number of gas phase species. The mass fraction of the last species is found from the constraint that the sum of the mass fractions must equal one. After the scalar equations are solved, the thermophysical properties are updated. Next the axial and radial momentum equations are solved to find the axial and radial velocities. The velocities from the momentum equation are corrected to satisfy continuity by solving the pressure correction equation.

Usually, continuity cannot be satisfied until all equations converge. The problem with the continuity equation is due to the use of Neumann boundary conditions on all boundaries in the pressure correction equation. During the intermediate non-linear iterations it is useful to at least enforce global conservation by scaling the outflow velocity.

The implementation of the species diffusion fluxes merits special attention. The species mass diffusion fluxes $\mathbf{j}_g$ are functions of both mass and thermal diffusion and are defined by the multicomponent diffusion equation.[15] The diffusion equation can be manipulated into a form that is more readily applied in the SIMPLER algorithm:[6]

$$\mathbf{j}_g = -\rho \bar{D}_g \nabla Y_g - \left[\frac{\rho Y_g \bar{D}_g}{W} \nabla W + D_g^\mathrm{T} \frac{\nabla T}{T} - Y_g \bar{D}_g W \sum_{i=1, i\neq g}^{NS} \left(\frac{\mathbf{j}_i}{W_i \mathscr{D}_{ig}} + \frac{D_i^\mathrm{T}}{W_i \mathscr{D}_{ig}} \frac{\nabla T}{T} \right) \right]. \tag{9}$$

The multicomponent diffusion coefficients are $\bar{D}_g$, the binary diffusion coefficients are $\mathscr{D}_{ij}$, the thermal diffusion coefficients are D_i^T and the molecular weight is W. The $\mathbf{j}_g$-terms that occur in equations (5) and (6) are replaced with the right-hand side of equation (9), where the first term is used

in the SIMPLER transport operator and the terms in square brackets go into the source term. Equation (9) must first be solved for the $\mathbf{j}_g$ as a closure equation.

3.2. Boundary conditions

There are two types of boundary conditions: conditions at physical boundaries and conditions at grid block interfaces. The interface conditions are the result of the block-by-block solution algorithm for the linearized equations. The numerical implementation of boundary conditions is aided by ghost cells which are used to calculate fluxes and averages at boundaries and interfaces.

Most boundary conditions are updated explicitly in the baseline CFD code. Only in the pressure and pressure correction equations are the boundary conditions implicitly coupled to the interior equations. The ghost cells used to couple grid blocks along block interfaces are locally explicit but are updated during line relaxation sweeps so that information propagates across the blocks.

Dirichlet conditions are traditionally applied at an inflow boundary, but they may not preserve the correct species mass flow rates when the flow is highly diffusive. If there is a gradient in chemical composition near the inflow boundary, then a more stringent condition for preserving species mass fluxes is applied:

$$\frac{\dot{\mathbf{m}}_g}{A} = \rho \mathbf{u} Y_g + \mathbf{j}_g. \tag{10}$$

The correct composition of chemical species entering the reactor is artificially imposed instead of moving the grid boundaries far away from the source of the composition gradient. The ghost cell values for the mass fractions at the inflow boundary are set such that the sum of the species convective and diffusive fluxes satisfies the imposed species mass flow rates $\dot{\mathbf{m}}_g$. In addition, the temperature in the inflow boundary ghost cell is fixed at the upstream value, while the velocities are adjusted such that they satisfy the specified mass flow rate:

$$\frac{\dot{\mathbf{m}}}{A} = \rho \mathbf{u}. \tag{11}$$

At an outflow boundary, variables are extrapolated from the interior to the ghost cells. The boundary velocity is also extrapolated from the interior. Since the extrapolation for the boundary velocity does not necessarily satisfy continuity, the outflow velocities are scaled to satisfy global mass conservation.

There are two types of solid wall boundary conditions: reacting wall and non- reacting wall. For both wall boundary condition types the temperature is prescribed. When there are no chemical reactions on the surface, the normal velocity is zero and species gradients are zero. When the wall is chemically reacting, surface reactions convert gas phase chemical species to surface phase, bulk phase and other chemical species. A non-zero Stefan velocity can be generated at the wall and is found by summing the production rates $\dot{\sigma}_g$ for gas phase species on the surface:

$$\mathbf{u} \cdot \mathbf{n} = \frac{1}{\rho} \sum_{g=1}^{NS} \dot{\sigma}_g. \tag{12}$$

The surface chemical production rates are calculated[19,20] from mass fractions which are extrapolated from the interior to the wall. The production of surface species is assumed to be in a quasi-equilibrium state about the gas phase species composition.

4. LOW-PECLET-NUMBER CONVERGENCE PROBLEM

The SIMPLER algorithm, described briefly in the previous section, converges slowly at low pressures where the Peclet number for mass diffusion is low. The root of the low-Peclet-number convergence problem is the interaction between the solution procedure and the unfavourable scaling within the chemical species mass-flux-preserving inflow boundary condition. A simple linear model problem is used to demonstrate how convergence degrades with the species cell Peclet number. Chemical reaction rates, which usually cause convergence problems in reacting flow computations, play a lesser role in the convergence rate degradation. The reactions are relatively slow at low pressure and not very energetic, so there is not a strong effect on the temperature. Surface chemistry is more problematic than gas phase chemistry since it provides highly non-linear sources and sinks for gas phase species at physical boundaries.

4.1. Inflow boundary condition scaling

For low Peclet numbers, diffusive transport dominates convective transport, yet, for the limiting case of uniform inflow composition and no chemical reactions, the convective transport term determines the species distribution within the flow domain. At low Peclet numbers it is difficult to enforce the convective part of the boundary condition. The difference, or scaling, between the strength of convection and diffusion makes it difficult to propagate information in the solution procedure when the boundary condition is enforced explicitly.

The scaling problem is demonstrated using the inflow boundary condition, equation (10), where the diffusion flux is simplified by the assumption of Fickian diffusion:

$$\frac{\dot{m}_g}{A} = \rho u Y_g - \rho D \frac{\partial Y_g}{\partial x}. \tag{13}$$

Dividing by the total mass flux, equation (11), gives

$$\bar{Y}_g = Y_g - \frac{D}{u}\frac{\partial Y_g}{\partial x}, \tag{14}$$

where $\bar{Y}_g = \dot{m}_g/\dot{m}$ represents the reference inflow mass fraction for species g. The cell Peclet number $Pe_{\Delta x}$ is defined by the length scale of the finite volume, Δx, the convective velocity and a mass diffusivity D:

$$Pe_{\Delta x} = \frac{u\Delta x}{D}. \tag{15}$$

The boundary condition is discretized about the inflow cell face using centred differences and the explicit formula for updating the boundary point is

$$Y_{g,1} = \frac{1 - \frac{1}{2}Pe_{\Delta x}}{1 + \frac{1}{2}Pe_{\Delta x}} Y_{g,2} + \frac{Pe_{\Delta x}}{1 + \frac{1}{2}Pe_{\Delta x}} \bar{Y}_g. \tag{16}$$

The ghost cell mass fraction is $Y_{g,1}$ and the first interior cell value is $Y_{g,2}$. For small limiting values of the cell Peclet number the ghost cell mass fraction is more dependent on the interior point than the reference value $\bar{Y}_g$ during the explicit update, yet the mass fractions must approach the reference value when there are no chemical reactions or sources:

$$\lim_{Pe_{\Delta x} \to 0} Y_{g,1} = Y_{g,2} + Pe_{\Delta x}\bar{Y}_g. \tag{17}$$

Bad initial guesses for the mass fractions result in a very slow convergence to the actual solution. The interior points are strongly dependent on the boundary points because of the elliptic nature of the partial differential equations.

4.2. Linear model problem

The Peclet number scaling problem is demonstrated using a simple linear model problem. The model approximates the multicomponent transport processes with simple diffusion, described by the Laplace equation

$$\nabla^2 \phi = 0. \tag{18}$$

The 'mass-flux-preserving' inflow boundary condition is set over part of the boundary and the rest of the boundary has a zero-gradient condition:

$$\phi - \frac{1}{\epsilon} \nabla \phi \cdot \mathbf{n} = 1, \tag{19}$$

$$\nabla \phi \cdot \mathbf{n} = 0, \tag{20}$$

where $\mathbf{n}$ is the unit normal vector to the boundary. The Peclet number scaling is introduced as the parameter ϵ, even though there is no convection in the model. The solution to these equations is $\phi = 1$ and the initial guess is $\phi = 0$. For the purposes of discussion the scaling parameter ϵ will be referred to as the Peclet number *Pe*.

Equations (18)–(20) are discretized using centered differences on a uniform multiple-block grid. The scalar variable ϕ is located at the center of volumes formed by grid points. The grid consists of three grid blocks, each of size 31×11 points. The blocks stack on top of each other to form a square grid, shown in Figure 1, with cells of unit length. The 'inflow boundary' is the west face of the bottom block 1.

The discrete linear equations are solved using a line relaxation scheme, analogous to the baseline solution algorithm for the Navier–Stokes equations. Each iteration consists of four alternating sweeps, followed by an explicit updating of boundary conditions. The first forward sweep solves coupled tridiagonal equations in the *j*-direction for lines of constant *i* in each grid block. The sweeping proceeds through the grid blocks, propagating new information across interior block boundaries. The second sweep solves coupled tridiagonal equations in the *i*-direction for lines of constant *j* in each grid block. The final two sweeps are similar to the first two sweeps, but the sweep direction is reversed.

A large number of line relaxation iterations are required to solve the equations when the scaling is poor (small Peclet number) and the boundary conditions are evaluated explicitly. The convergence is

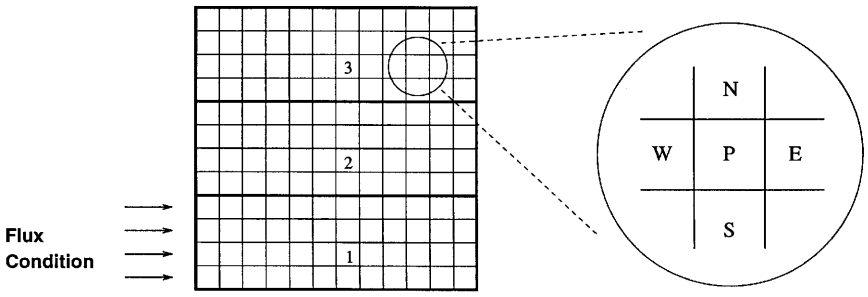

Figure 1. Three-block grid for model problem, with implicit matrix stencil

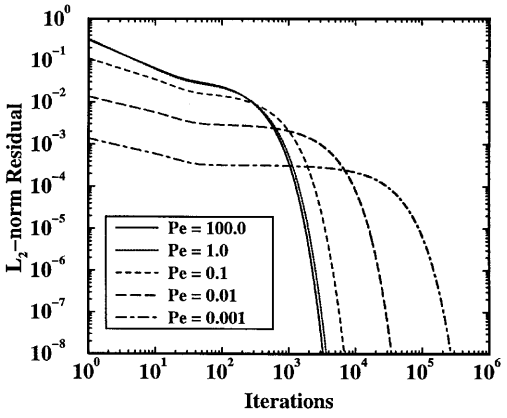

Figure 2. Line relaxation convergence scales inversely with Peclet number

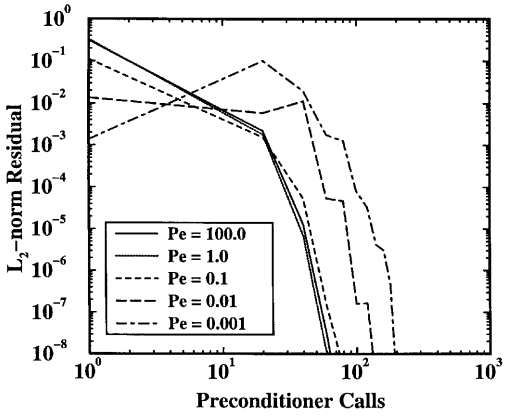

Figure 3. Line relaxation convergence is accelerated by GMRES

plotted as a function of the L_2-norm of the linear system residual in Figure 2. The number of iterations required to converge the problem scales with $\frac{1}{2} + 1/Pe$. As the Peclet number increases, the number of iterations required to converge the problem with the flux inflow boundary condition approaches that for a fixed Dirichlet boundary condition $\phi = 1$.

The solution procedure is improved by implicitly coupling the physical boundary conditions in the line relaxation scheme. Convergence improves by 25 per cent, but the work required is still excessive. The boundary information is propagated implicitly in the direction of the line solve, but more 'explicit-like' in the sweep direction. The full set of physical boundaries which drive the interior equations is not implicitly coupled. The explicit propagation of information is degraded by the presence of internal block boundaries. Each physical boundary point should simultaneously see every other boundary point, which implies that the linear problem must be solved exactly.

Matrix-free, preconditioned gradient algorithms provide an efficient solution to the boundary communication problem. The method is similar to direct inversion, but there is no need to store inverse matrix fill-in. The storage space requirements for gradient methods scale with the number of unknowns. The generalized minimal residual (GMRES) gradient algorithm[21] is used to invert the linear system with the block-by-block line relaxation implicit scheme acting as a preconditioner. The GMRES scheme enhances the implictness of the line relaxation scheme.

The convergence history for the enhanced implicit scheme is shown in Figure 3. The gradient algorithm is not restarted for the model problem and the number of search directions is limited to 20 so that outer iterations on the problem are required. The residual norm is plotted as a function of the number of preconditioning calls. Each preconditioning call is comparable in computational work to one iteration of the line relaxation scheme alone. The overhead work of the gradient algorithm effectively increases the amount of CPU time required for each preconditioning call by 30 per cent. However, the addition of the GMRES scheme accelerates the convergence of the line relaxation scheme by anywhere from a factor of four to 1000, depending on the cell Peclet number.

The amount of work required to converge on a finer mesh follows theoretical scaling laws for relaxation-type schemes. When the number of mesh points is increased by a factor of 16 to three 121×41 grids, the line relaxation scheme takes a greater number of iterations to converge, demonstrated in Figure 4. The number of iterations required to reach the convergence tolerance increases by roughly an order of magnitude. The convergence rate for the GMRES-accelerated scheme on the denser grid behaves similarly to the line relaxation itself, shown in Figure 5.

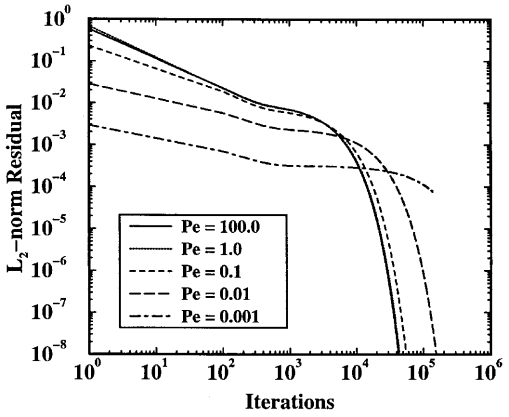

Figure 4. Line relaxation convergence degrades with increasing mesh density

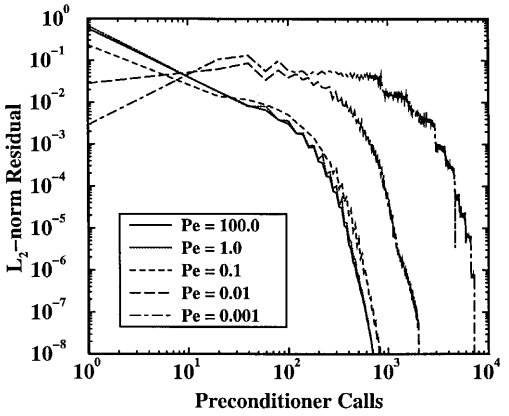

Figure 5. Line relaxation convergence accelerated by GMRES on denser grid

5. ENHANCEMENTS TO THE NAVIER–STOKES CODE

The solution algorithm enhancements to the SIMPLER algorithm involve only the species equations and the pressure correction equation. The modifications to the species equations follow those described for the model problem where the boundary condition equations are treated implicitly and the linear problem is solved to completion using a matrix-free, preconditioned gradient algorithm. Though any gradient algorithm can be used, or even multigrid, the GMRES algorithm is selected for this work. The GMRES algorithm is robust and only requires one application of the preconditioning step per Krylov vector.

The underrelaxation used in the SIMPLER algorithm must be reformulated for use with the enhanced solution strategy. The underrelaxation adds diagonal dominance to the linearized equations, but it sharply degrades performance when there are Peclet number scaling problems. The underrelaxation acts as an artificial time term, slowing down the propagation of information even more. A more favourable implementation of underrelaxation is to apply the damping parameter to the solution increment which results from the solution of the linearized equations.

A new problem arises when the linearized species equations are fully converged at each step: the non-linear solution process becomes unstable. In the baseline algorithm, neither continuity nor the linearized species equations are satisfied exactly at each non-linear iteration. The errors tend to offset each other and there are no large instabilities. Conversely, the enhanced solution algorithm does such a good job of satisfying the transport equations that they become very sensitive to mass errors. Errors in continuity cause artificial sources and sinks in the species equations. Stability is increased by satisfying the continuity equation more rigorously.

The solution to the problem of converging the continuity equation is twofold. First, it is recognized that continuity errors during iteration are due to the use of zero-gradient boundary conditions for the pressure correction on all boundaries and incomplete convergence of the linear pressure correction equation. The zero-gradient boundary condition on the pressure correction term does not allow outflow boundary velocities to change. The outflow boundary condition is reformulated so that the outflow velocity is corrected in a manner consistent with continuity. Secondly, the matrix-free gradient algorithm is added to the solution algorithm for the pressure correction equation to accelerate convergence.

The pressure correction procedure is reformulated in a manner consistent with other projection methods for low-Mach-number flows.[22–24] The gradient of the pressure correction corresponds to a velocity increment and the pressure correction variable itself is related to a pressure increment. In the enhanced algorithm the pressure correction is set to zero at the outflow boundary ghost points. A non-zero velocity correction is then calculated at the outflow boundary and the velocity is updated in a manner consistent with continuity. Continuity is satisfied in each finite volume at each non-linear step. The zero condition also corresponds to fixing the outflow pressure, since the pressure increment is zero. This does not mean the outflow pressure is fixed for the entire calculation, because the pressure itself is still updated according to the SIMPLER algorithm.

The actual implementation requires matrix coefficients at the outflow face. If the outflow is on an east face, then a value for A_e would be required in equation (8) for the interior volume adjacent to the outflow boundary. In the baseline algorithm the value of A_e is zero. In the enhanced algorithm the matrix coefficient is extrapolated from the interior.

6. EXAMPLE PROBLEMS

The effectiveness of the enhanced solution strategy is demonstrated for three manufacturing process examples. All suffer from poor convergence due to a low Peclet number. Comparisons are drawn between different solution algorithms in terms of convergence performance. In the baseline algorithm the transport equations are solved according to the SIMPLER procedure as outlined in Section 3. In the enhanced algorithm the modifications of Section 5 are applied to the species equations and the pressure correction equation. Additional results are included for an intermediate algorithm where the modified pressure correction procedure is used, but the gradient algorithm is applied with explicit boundary conditions. In all cases, improved convergence is realized only with the full set of modifications: implicitly coupled boundary conditions, full convergence of the linear algebra problem and modification of the pressure correction equation. Performance enhancement requires more than just improving the efficiency of the linear algebra solver with a preconditioned gradient algorithm.

Convergence is measured in terms of the L_1-norm of the residual of the species equation. The L_1-norm is constructed from flux balances over each control volume for each species equation, scaled by the maximum species convective flux. The residual norm can be misleading, since physical quantities of interest often converge to within engineering accuracy long before the residual norm reaches the limits of computer precision. A second measure of convergence is considered which is based on the physical quantities extracted from the simulation. The number of iterations required to converge to engineering accuracy is found by studying surface heat and mass flux profiles. The convergence of surface mass flux is plotted as a sequence of mass flux profiles at various iteration counts. All example problems are run on a Hewlett Packard 735/125 workstation. The discrete governing equations in the code are solved using dimensional quantities in terms of the metric CGS system.

6.1. SiO_2 *deposition*

The growth of silicon dioxide dielectric in a rotating disc chemical vapour deposition (CVD) reactor is a problem that involves gas phase and surface phase chemistry in a diffusive environment. The baseline algorithm does not converge extremely slowly for this problem, but the enhanced algorithm improves the average convergence time by over a factor of two.

In the CVD reactor, tetraethoxysilane (TEOS) gas is injected through a shower head arrangement onto a heated substrate, supplying the silicon and oxygen for the deposition. Heat is conducted from the substrate into the fluid and dissociates the TEOS gas into chemical precursors. The chemical precursors react at the substrate with intermediate surface species to form solid silicon dioxide.

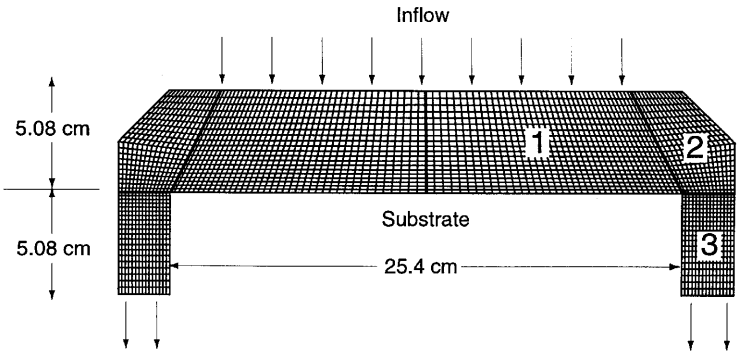

Figure 6. Three-block grid for TEOS reactor

6.1.1. Problem definition. The reactor grid is constructed from three mesh blocks, shown in Figure 6, with block sizes of 21×41, 21×16 and 21×16 grid points. The reactor is axisymmetric about the vertical centreline. Radial position is measured relative to the vertical centreline and axial position is measured relative to the base of the outflow face.

The axial temperature distribution along the outer vertical wall is represented by a piecewise linear curve, given in Table I. The growth substrate is maintained at 1000 K and rotates at 30 rpm. The thermodynamic pressure in the reactor is 1 Torr. The process gas, a mixture of nitrogen and TEOS as shown in Table II, flows into the reactor at 300 K. The species Peclet numbers are based on the inflow conditions and are given for a unit reference length of 1 cm: $Pe_{N_2} = 0{\cdot}27$–$0{\cdot}57$ and $Pe_{TEOS} = 2{\cdot}0$–$4{\cdot}2$. These numbers are multiplied by the wall-normal grid spacing of 0·35 cm to form the cell Peclet numbers.

Table I. Wall temperature profile as a function of axial position

Axial position (cm)	Temperature (K)
0·0	300·0
2·54	320·0
5·08	400·0
7·62	380·0
10·16	340·0

Table II. TEOS inflow conditions as a function of radial position

	Inner 5·08 cm	Outer 5·08 cm
Mole fraction X_{TEOS}	0·5	0·75
Mole fraction X_{N_2}	0·5	0·25
Velocity V_{in} (cm s^{-1})	20	30

The gas phase species considered in the model are N_2, $Si(OC_2H_5)_4$, $Si(OH)(OC_2H_5)_3$, C_2H_5OH, C_2H_4 and H_2O. A single gas phase reaction describes the thermal decomposition of TEOS into triethoxysilanol and ethene,

$$Si(OC_2H_5)_4 \rightleftharpoons Si(OH)(OC_2H_5)_3 + C_2H_4,$$

and the Arrhenius coefficients for describing the reaction rate $k = AT^b \exp(-E/RT)$ are $A = 4{\cdot}9 \times 10^{13}$ s^{-1}, $b = 0{\cdot}0$ and $E = 61{,}460$ cal mol^{-1}. The water and ethanol species are by-products of the surface phase reaction mechanism, given in Table III. A shorthand notation is used involving the symbol G for describing the intermediary glass-like surface species required to form bulk dielectric SiO_2(D). The notation (D) indicates the solid material in this mechanism. The computational solution indicates that the chemical precursors are fairly well distributed across the substrate surface. One of the primary chemical radical precursors to the silicon dioxide deposition is $Si(OH)(OC_2H_5)_3$, shown in Figure 7.

The underrelaxation parameters are used differently between the baseline algorithm and the enhanced algorithm, so they also take on different values to maintain stability. The baseline code uses an underrelaxation value of 0·5 for the temperature equation and 0·9 for the species equations. With the enhanced code the temperature equation is damped at a value of 0·9 while the species equations are damped at a value of 0·5.

Table III. TEOS surface phase reaction mechanism

Reaction	A	b	E
1. $Si(OC_2H_5)_4 + SiG3(OH) \rightleftharpoons SiO_2(D) + SiGE3 + C_2H_5OH$	$2{\cdot}5 \times 10^{4}$‡	0·0	44600
2. $SiG3E \rightleftharpoons SiG3(OH) + C_2H_4$	$1{\cdot}2 \times 10^{12}$	0·0	47000
3. $SiG(OH)E2 \rightleftharpoons SiG(OH)2E + C_2H_4$	$2{\cdot}4 \times 10^{12}$	0·0	47000
4. $SiGE3 \rightleftharpoons SiG(OH)E2 + C_2H_4$	$3{\cdot}6 \times 10^{12}$	0·0	47000
5. $SiG(OH)2E \rightleftharpoons SiG3(OH) + C_2H_5OH$	$1{\cdot}4 \times 10^{12}$	0·0	44000
6. $SiG(OH)E2 \rightleftharpoons SiG3E + C_2H_5OH$	$1{\cdot}4 \times 10^{12}$	0·0	44000
7. $SiG(OH)2E \rightleftharpoons SiG3E + H_2O$	$1{\cdot}4 \times 10^{12}$	0·0	45000
8. $Si(OH)(OC_2H_5)_3 + SiG3(OH) \rightleftharpoons SiO_2(D) + SiGE3 + H_2O$	$7{\cdot}0 \times 10^{0}$†	0·0	12000

Surface site density $7{\cdot}5 \times 10^{-10}$ mol cm^{-2}, bulk density 2·19 g cm^{-3}.
A in CGS units, b dimensionless, E in cal mol^{-1}; see SURFACE CHEMKIN manual.[19]
† Reaction probability.

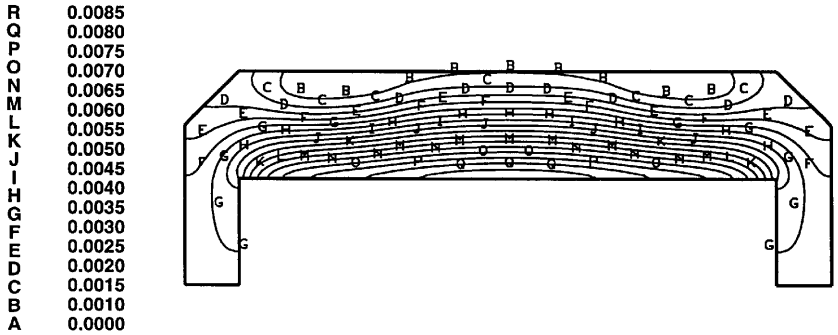

Figure 7. Triethoxysilanol mole fraction contours (A–R) in TEOS reactor

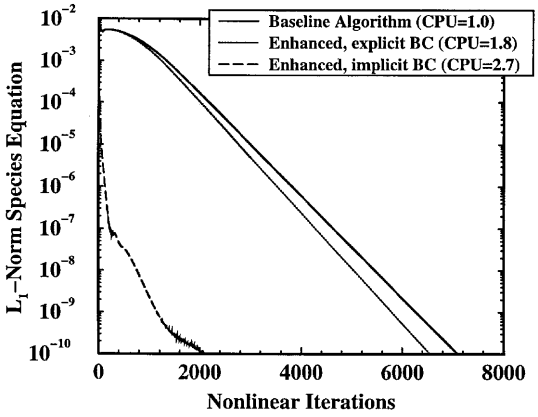

Figure 8. Convergence history for TEOS problem

6.1.2. Convergence performance. The convergence rate of the nitrogen species equation is shown in Figure 8, with relative CPU costs per iteration given in the legend. The baseline code runs at a rate of 1·28 s per iteration, while the enhanced code with the gradient algorithm runs at a rate of 3·45 s per iteration. Even though the enhanced code runs 2·7 times slower per iteration than the baseline code, the enhanced code converges to an engineering solution over twice as fast as the baseline code. Surface mass velocity profiles for the baseline code are plotted in Figure 9, showing convergence after 1250 iterations. The convergence of the surface mass velocity for the enhanced code is shown in Figure 10, with convergence occurring after 200 iterations.

In order to demonstrate that all the features of the enhanced solution algorithm are required for improved performance, results for some algorithm variations are presented. If the enhanced algorithm is changed such that all species boundary conditions are treated explicitly, the code converges at the rate of the baseline code, but at twice the CPU cost per iteration, shown as the middle curve of Figure 8. Performance improvements are coming from the direct coupling of boundaries and not just from the use of a more efficient linear algebra solver.

Although not plotted, two other intermediate solution combinations lead to similar poor performance. The use of the baseline pressure correction strategy with the modified species equation

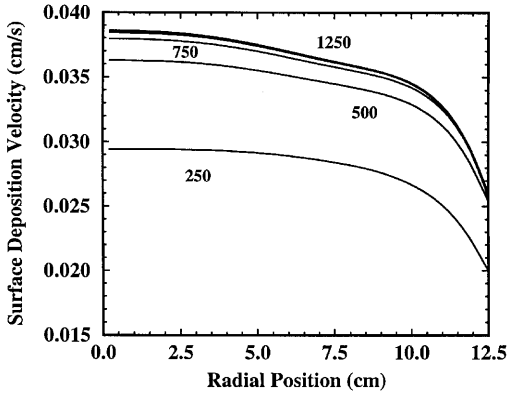

Figure 9. Surface deposition velocity profile as a function of solution iteration for baseline code

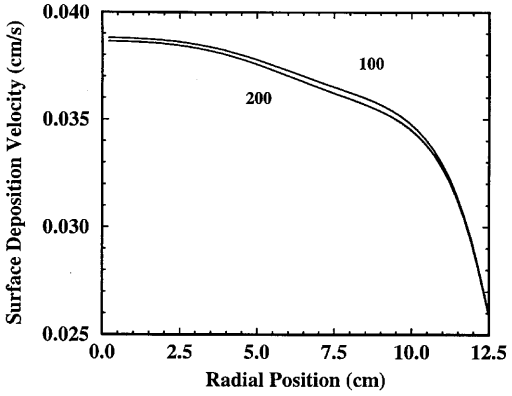

Figure 10. Surface deposition velocity profile as a function of solution iteration for enhanced code

solver leads to baseline convergence rates. The continuity equation must be rigorously satisfied. The use of implicit species equation boundary conditions, but with line relaxation and the modified pressure correction solver, also leads to baseline convergence rates. The linear algebra problem must be solved exactly so boundaries can communicate. The outflow boundary must know much species comes in through the inflow boundary and how much species are generated at reacting surfaces in order to satisfy conservation.

6.2. *Chemical downstream etch*

The chemical downstream etch (CDE) process uses a plasma to create chemical radical species for the surface etching of material layers in microelectronic device manufacturing. The reactor consists of a plasma generator, a plasma transport tube and an etch chamber. Plasma species are generated in a plasma generator far upstream of the etch process. The plasma species recombine during the transport phase so only neutral radical species reach the surface. The chemical radical species are transported downstream to a substrate where they selectively etch certain materials. For this calculation, only the transport of neutral species in the etching chamber is modelled.

6.2.1. Problem definition. A reactive gas mixture of F and NF_3 flows down over a substrate of solid silicon and the etch products are exhausted out of the bottom of the reactor. The reactor chamber, shown in Figure 11, is axisymmetric about the vertical centreline. The grid consists of three blocks of size 31×11, 31×21 and 31×11 grid points.

The gas phase species considered in the model are F, SiF_4 and NF_3. The inflow mole fraction composition is $X_F = 0{\cdot}4$, $X_{SiF_4} = 0{\cdot}2$ and $X_{NF_3} = 0{\cdot}4$ and the temperature is 300 K. The thermodynamic pressure is 10 mTorr and the inflow velocity is 940 cm s^{-1}. The species Peclet numbers are based on the inflow conditions and are given a unit reference length of 1 cm: $Pe_F = 0{\cdot}049$, $Pe_{SiF_4} = 0{\cdot}185$ and $Pe_{NF_3} = 0{\cdot}145$. These numbers are multiplied by the wall-normal grid spacing of 0·5 cm to find the cell Peclet numbers.

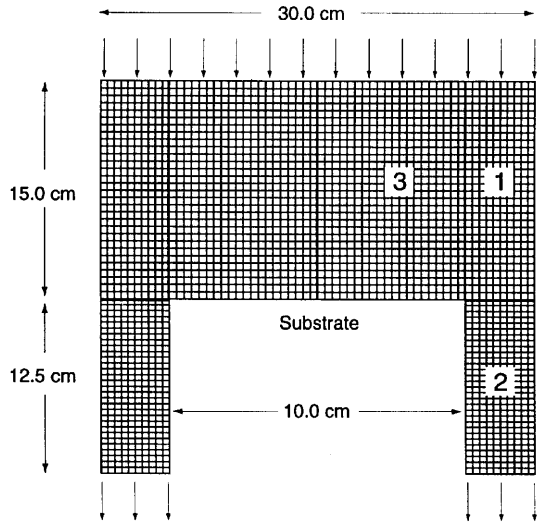

Figure 11. Three-block grid for chemical downstream etch reactor

There are no gas phase reactions. In the single surface phase reaction the etch rate of bulk silicon by atomic fluorine,

$$F + 0{\cdot}25Si(B) \rightleftharpoons 0{\cdot}25SiF_4,$$

is described by a simple sticking coefficient of 0·01. The surface site density is $4{\cdot}17 \times 10^{-9}$ mol cm^{-2} and the bulk density is 2·33 g cm^{-3}. The resulting contours of SiF_4 mole fraction are shown in Figure 12.

The underrelaxation parameters are used differently between the baseline algorithm and the enhanced algorithm, so they also take on different values to maintain stability. The underrelaxation parameters used with the baseline code are 0·5 for velocity and 0·25 for species. The surface reaction rates are fast relative to the diffusion time scales and a damping coefficient of 0·1 is required for the species equations with the enhanced algorithm.

6.2.2. Convergence performance. The convergence rates for the baseline code and the enhanced code are shown as the first and fourth curves in Figure 13. The baseline code runs at a rate of 0·475 s per iteration, while the enhanced code runs at a rate of 1·2 per iteration. Even though the enhanced code runs three times slower per iteration than the baseline code, it reaches the engineering accuracy norm of 3×10^{-6} twenty times faster, shown in Figure 14. The baseline code requires at least 2500 iterations to converge to the correct surface mass flux. The enhanced code converges in about 40 iterations.

It is important to combine both the modified pressure correction and species equation strategies in the enhanced algorithm. If explicit boundary conditions are used with the gradient solver for the species equations, very poor performance is achieved, shown as the second curve in Figure 13. The convergence rate is about an order of magnitude slower than the baseline algorithm. The difference between using the old and modified pressure correction strategies, along with the modified species equations strategy, is shown in the third curve of Figure 13. The modified pressure correction solver is worth a doubling in performance.

6.3. Methyltrichlorosilane injection

Cold methyltrichlorosilane (MTS) gas, CH_3SiCl_3, is injected and mixed into hot helium gas for a silicon carbide CVD flow tube experiment. The flow tube is designed such that the gases are

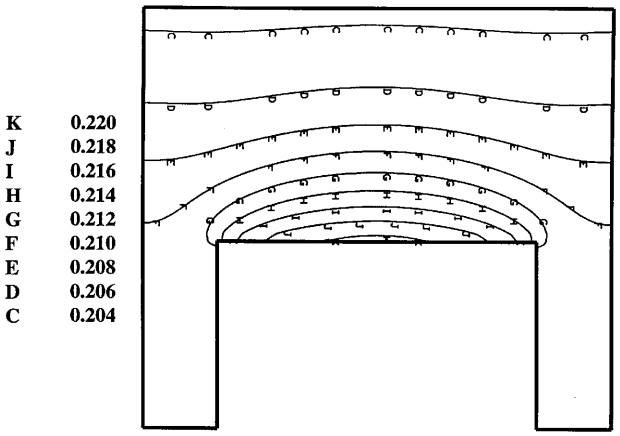

Figure 12. SiF_4 mole fractions (C–K) for CDE at 10 mTorr

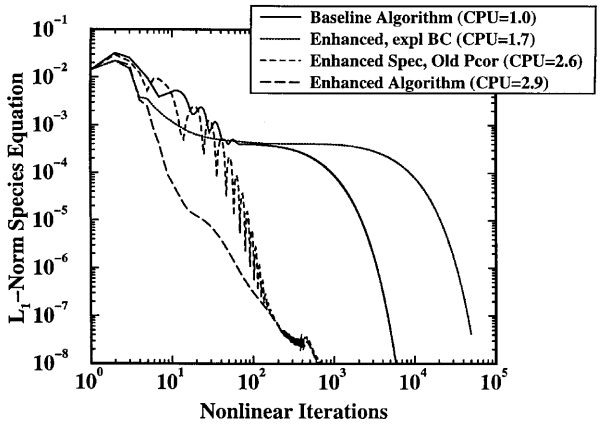

Figure 13. Convergence performance for CDE process

completely mixed at the exhaust. The flow tube solution is very problematic for the baseline algorithm.

6.3.1. Problem definition. The geometry consists of a cylindrical center plug with normal injectors, inside a cylindrical mixing chamber, shown in Figure 15. The spherically capped center plug is 2·5 cm in diameter and the mixing chamber is 5·0 cm in diameter. The injectors are 1·0 mm in diameter and are modelled as a continuous ring. The grid system consists of five blocks of size 26×26, 7×26, 11×26, 36×21 and 36×26 grid points. The grid is symmetric about the axial centerline.

The flow tube model is not a low-pressure problem, but it is very diffusive. The thermodynamic pressure is 100 Torr. The MTS flows in a rate of 500 sccm and a temperature of 300 K. The helium flows in at a rate of 4500 sccm and a temperature of 1400 K. The species Peclet numbers are based on the inflow conditions and are given for a unit reference length: $Pe_{\mathrm{He}} = 0{\cdot}037$ and $Pe_{\mathrm{MTS}} = 300{\cdot}0$. These numbers should be multiplied by the wall-normal grid spacing of 0·367 cm for helium and 0·0346 cm for MTS to find the cell Peclet numbers. The MTS rapidly mixes with the hot helium and contours of the MTS mole fraction are shown for the near-injector region in Figure 16. The mole fraction of MTS for complete mixing is 0·1.

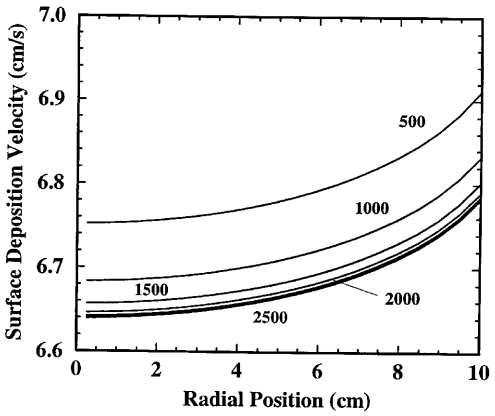

Figure 14. Surface mass flux profiles as a function of iteration for CDE process

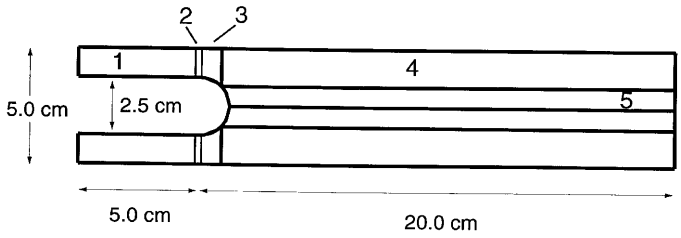

Figure 15. Five-block MTS grid system

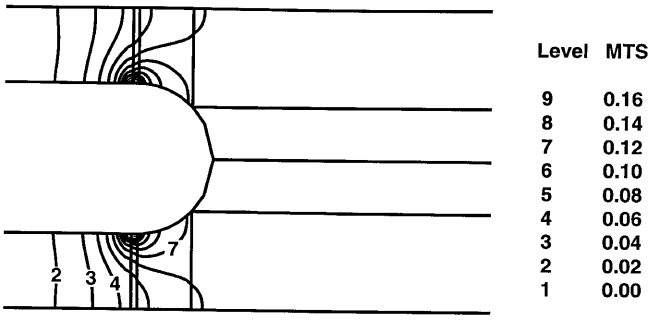

Figure 16. MTS mole fractions near injectors

6.3.2. Convergence performance. The convergence history for the species equations, in terms of the scaled L_1-norm of the conservation laws, is plotted in Figure 17. The converged mixing levels are approached after about 200 iterations. The enhanced code runs at about 6 s per iteration for this problem. The convergence history for the baseline code is not shown because it is prohibitively slow for this case. Well over 100,000 iterations are required with the baseline code. The enhanced algorithm provides an acceleration of over a factor of 100 relative to the baseline algorithm.

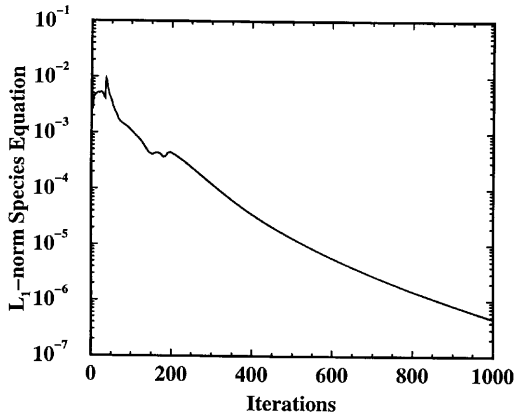

Figure 17. MTS convergence performance for enhanced solution strategy

7. CONCLUSIONS

An enhanced solution strategy based on the SIMPLER algorithm is presented for low-Peclet-number mass transport calculations with applications in material processing. When the flow is highly diffusive, a chemical species flux- preserving inflow boundary condition is imposed. The boundary condition contains a scaling problem that causes the implicit line relaxation solution algorithm for the species equations to converge very slowly. The convergence rate degrades as the species Peclet number decreases.

The adverse interaction between the inflow boundary condition and the implicit line relaxation scheme is first demonstrated using a simple linear model problem. Explicit updating of boundary conditions causes information to propagate very slowly between physical boundaries. In order to improve boundary communication, the boundary conditions must be treated as implicit equations and the linear problem must be solved to completion. In this work, full solution of the linear problem is achieved by increasing the efficiency of the line relaxation scheme with a gradient algorithm.

Convergence for the non-linear Navier–Stokes equations is enhanced by modifying both the non-linear and linear parts of the baseline SIMPLER algorithm. The modifications are applied only to the species equations and the pressure correction equation. The boundary condition equations are implicitly coupled to the interior equations during the linearization, so the linear matrix problem contains equations for the boundary points. The linear problem is solved completely at each non-linear iteration using a gradient method to accelerate the existing line relaxation method. A new problem arises when the linearized species equations are fully converged at each step; the non-linear solution process becomes unstable because of the lack of mass continuity. In the baseline algorithm, continuity is not satisfied exactly at each non-linear iteration. Boundary conditions for the pressure correction equation are reformulated to ensure continuity is preserved in each finite volume at each iteration.

The resulting code modifications are relatively non-intrusive, requiring few changes to validated sections of code. The modifications to the solution strategy are consistent with the segregated strategy of the SIMPLER algorithm. The approach is much more attractive than using a fully coupled Newton method, because it is easier to increase the chemical complexity without running out of computing resources.

REFERENCES

1. K. F. Jensen, D. I. Fotiadis and T. J. Mountziaris, 'Detailed model of the MOVPE process', *J. Cryst. Growth*, **107**, 1–11 (1991).
2. C. R. Kleijn, 'On the modelling of transport phenomena in chemical vapour deposition and its use in reactor design and process optimization', *Thin Solid Films*, **206**, 47–53 (1991).
3. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.
4. G. Evans and R. Greif, 'A two-dimensional model of the chemical vapor deposition of silicon nitride in a low-pressure hot-wall reactor including multicomponent diffusion', *Int. J. Heat Mass Transfer*, **37**, 1535–1543 (1994).
5. C. R. Kleijn and C. J. Hoogendoorn, 'A study of 2- and 3-D transport phenomena in horizontal chemical vapour deposition reactors', *Chem. Eng. Sci.*, **46**, 321–334 (1991).
6. C. R. Kleijn, Th. H. van der Meer and C. J. Hoogendoorn, 'A mathematical model for LPCVD in a single wafer reactor', *J. Electrochem. Soc.*, **136**, 3423–4333 (1989).
7. S. V. Patankar, 'Recent developments in computational heat transfer', *J. Heat Transfer*, **110**, 1037–1045 (1988).
8. D. A. Knoll, P. McHugh and D. Keyes, 'Newton–Krylov methods for low Mach number combustion', *AIAA Paper 95-1672*, 1995.
9. H. Moffat and K. F. Jensen, 'Complex flow phenomena in MOCVD reactors: I. Horizontal reactors', *J. Cryst. Growth*, **77**, 108–119 (1986).
10. H. Moffat and K. F. Jensen, 'Three-dimensional flow effects in silicon CVD in horizontal reactors', *J. Electrochem. Soc.*, **135**, 459–471 (1988).
11. A. D. Gosman and W. M. Pun, 'Calculation of recirculating flow', unpublished lecture notes, Imperial College of Science and Technology, 1973.

12. W. S. Winters, G. H. Evans and C. D. Moen, 'CURRENT—a computer code for modeling two-dimensional, chemically reacting, low Mach number flows', *Tech. Rep. SAND96-XXXX*, Sandia National Laboratory, Livermore, CA, 1996.
13. L. B. Wigton, N. J. Yu and D. P. Young, 'GMRES acceleration of computational fluid dynamics codes', *AIAA Paper 85-1494*, 1985.
14. W. Shyy, S. S. Tong and S. M. Correa, 'Numerical recirculating flow calculation using a body-fitted coordinate system', *Numer. Heat Transfer*, **8**, 99–113 (1985).
15. K. K. Kuo, *Principles of Combustion*, Wiley, New York, 1986.
16. R. J. Kee, F. M. Rupley and J. A. Miller, 'CHEMKIN-II: a Fortran chemical kinetics package for the analysis of gas-phase chemical kinetics', *Tech. Rep. SAND89-8009B*, Sandia National Laboratory, Livermore, CA, 1991.
17. R. J. Kee, F. M. Rupley and J. A. Miller, 'The Chemkin thermodynamic data base', *Tech. Rep. SAND87-8215B*, Sandia National Laboratory, Livermore, CA, 1992.
18. R. J. Kee, G. Dixon-Lewis, J. Warnatz, M. E. Coltrin and J. A. Miller, 'A Fortran computer code package for the evaluation of gas-phase multicomponent transport properties', *Tech. Rep. SAND86-8246*, Sandia National Laboratory, Livermore, CA, 1991.
19. M. E. Coltrin, R. J. Kee and F. M. Rupley, 'SURFACE CHEMKIN (version 4.0): a Fortran package for analyzing heterogeneous chemical kinetics at a solid-surface—gas-phase interface', *Tech. Rep. SAND90-8003B*, Sandia National Laboratory, Livermore, CA, 1991.
20. M. E. Coltrin, R. J. Kee and F. M. Rupley, 'Surface Chemkin: a general formalism and software for analyzing heterogeneous chemical kinetics at a gas–surface interaction', *Int. J. Chem. Kinet.*, **23**, 1111–1128 (1991).
21. Y. Saad and M. H. Schultz, 'GMRES: a generalized minimal residual algorithm for solving nonsymmetrical linear systems', *SIAM J. Sci. Stat. Comput.*, **7**, 856–869 (1986).
22. A. J. Chorin, 'Numerical solution of the Navier–Stokes equations', *Math. Comput.*, **22**, 745–762 (1968).
23. H. Dwyer, 'Calculation of low Mach number reacting flows', *AIAA J.*, **28**, 98–105 (1991).
24. H. Dwyer, 'Navier–Stokes calculations of multi-dimensional flows with complex chemical kinetics', *Comput. Syst. Eng.*, **5**, 105– 116 (1994).

