(NASA-CR-199613) A MASSIVELY PARALLEL COMPUTATIONAL APPROACH TO COUPLED THERMOELASTIC/POROUS GAS FLOW PROBLEMS (MIT) 11 p

N96-12582

Unclas

G3/34 0072088

1994 JANNAF ROCKET NOZZLE TECHNOLOGY SUBCOMMITTEE MEETING
Seattle, Washington
November 15-17, 1994

A MASSIVELY PARALLEL COMPUTATIONAL APPROACH TO COUPLED THERMOELASTIC/POROUS GAS FLOW PROBLEMS

David Shia, Research Assistant, and Hugh L. McManus, Class of 1943 Assistant Professor
Massachusetts Institute of Technology, Cambridge, MA 02139

ABSTRACT

A new computational scheme for coupled thermoelastic/porous gas flow problems is presented. Heat transfer, gas flow, and dynamic thermoelastic governing equations are expressed in fully explicit form, and solved on a massively parallel computer. The transpiration cooling problem is used as an example problem. The numerical solutions have been verified by comparison to available analytical solutions. Transient temperature, pressure, and stress distributions have been obtained. Small spatial oscillations in pressure and stress have been observed, which would be impractical to predict with previously available schemes. Comparisons between serial and massively parallel versions of the scheme have also been made. The results indicate that for small scale problems the serial and parallel version use practically the same amount of CPU time. However, as the problem sizes increases the parallel version becomes more efficient than the serial version.

INTRODUCTION

Ablatives fabricated from polymeric composite materials have been used as insulation liners for solid rocket motor nozzles and as heat shields for reentry vehicles. In practice, these parts may become damaged prematurely due to stresses and internal pressures created when the material is heated very rapidly. Therefore, to make full use of these ablatives, their responses must be understood.

The responses of composite ablatives can be studied by performing experiments. However, these experiments must be done on a large scale to give accurate results, and can be prohibitively expensive. Moreover, experiments do not always reveal the details of the underlying physical processes. Analytical models are therefore required. The governing equations of composite ablatives are typically very complex, and closed-form solutions are impossible to obtain in most cases. This implies that the governing equations need to be solved numerically using computers.

In the previous numerical modeling works done by McManus[1], Sullivan[2], and Kuhlmann[3], the numerical schemes adopted are either finite element or implicit finite difference methods. In both methods, some simplifications must be made in the governing equations in order to keep the computation tractable. For instance, chemical reactions for absorbed volatiles sometimes are assumed to be only temperature dependent. In reality, these chemical reactions are both temperature and pressure dependent. It is important that sufficient complexities are included in the computation such that the predicted results can be used with confidence.

Explicit finite difference methods have not been used often in this field. Henderson[4] used the method to solve the heat transfer equation. This method was later abandoned by Henderson due to its demand on computational resources. There are two major advantages in using the explicit scheme. The first advantage is that it leads to a simple algorithm which is easy to program. The second advantage of using the explicit scheme is that complex nonlinear physics can be easily incorporated into the algorithm. However, there are two major drawbacks in the explicit scheme. The first one is the stability consideration. In the explicit scheme, for a given spatial discretization, the time step needs to be small enough so that the solution will not diverge. This means that when the solution after a long period of time is desired, a large number of time steps are required. The second drawback applies especially to serial machines. When higher accuracy is desired, smaller spatial discretization will be needed. This leads to a large number of iterations in space on serial computers. As the spatial discretization is increased, time steps need to be reduced correspondingly in order to maintain stability. For serial machines, the total number of operations required increases very rapidly as finer and finer spatial discretizations are used. These difficulties can be overcome when the algorithm is implemented on parallel computers. With parallel computers, the spatial finite difference equations can be solved at all of the nodes simultaneously. This reduces the necessary number of *sequential* operations. Thus, even when the solution of the problem requires small time steps for stability, it still can be performed in a reasonable amount of time.

This paper will present the results of a study on the feasibility of using the explicit scheme and the power of parallel computers. A sample problem is solved on the massively parallel computer, a Thinking Machines CM-5, at MIT. The transpiration cooling problem is used as the sample problem. In this problem, a plate made of porous material heated on one side is cooled by sending a gas flow from the cool side to the hot side. This problem resembles the composite ablative problem in that the solution requires solving simultaneously the continuity, energy, and momentum equations.

In order to clarify subsequent discussions, a brief section on the principle of parallel computing on the CM-5 is presented below. Then, the transpiration cooling problem is defined. Following the problem definition, the governing equations for the transpiration problem are given. Then, the parallel computer implementation is shown. Finally, the solution of the transpiration cooling problem is discussed.

Supported by NASA Marshal grant NAG8-295.

PARALLEL COMPUTING ON THE CM-5

The Connection Machines CM-5 is a massively parallel, SIMD (shared memory) computer. Machines of this type consist of a very large number of processing elements. Each processing element has physically connected memory for easy communication. On the other hand, intense communication needs to take place between the large number of processors. Efficient algorithms will minimize this communication. For large numerical codes, the regularity of the data structure and the absence of sequential operations on it are important factors[5].

The schematic drawing of CM architecture is shown in Figure 1. The serial control processor directs the actions of the set of parallel processors. The serial controller also performs all operations that are programmed to be performed strictly sequentially. In this case, the parallel processors do nothing. The parallel processors act on data elements stored in their local memories. Parallel processors are most efficient when acting on large data sets each element of which can be acted on independently. The entire set can then be acted on simultaneously, one processor working on each element.

CM Fortran is the language used here to implement the parallel algorithm on the CM-5. It allows data parallel programming in a language familiar to most researchers. CM Fortran incorporates into the Fortran language the array extensions of Fortran 90. A full description of CM Fortran and may be found in the Connection Machine documentation[6]. CM Fortran does not require the programmer to be concerned about the details of parallelization. The CM Fortran compiler will perform the parallelization after the code is written. However, the programmer needs to arrange the data structure so that the compiler can parallelize the code in the most optimal fashion. The most optimal parallelization can be achieved by the compiler when data structure is arranged into different sets of *conformable arrays* (arrays that are all of the same size and shape), and that all operations are performed with conformable arrays from the same set. The reason is that the compiler will assign conformable arrays to the same parallel processor set. When this is done, operations are performed in parallel without communications between different sets of processors.

A specific example is given below for calculating the first spatial derivative of A, a function defined in one dimension x at n evenly spaced nodes separated by a distance Δx. On one boundary, at $x = 0$, $A = 0$. On the second boundary, $x = n\,\Delta x$, the derivative of A is set to 0. The central difference formula

$$\frac{dA_i}{dx} = \left(\frac{A_{i+1} - A_{i-1}}{2\Delta x} \right)$$

is used to approximate the derivative. The procedure written in CM Fortran follows. A, A_p, A_m, D_x and δA are array objects of length n. The derivative of A can be calculated by the code:

A_p = EOSHIFT (A, 1, 1, $A(n\text{-}1)$)
A_m = EOSHIFT (A, 1, -1, 0)
$D_x = 2 * \Delta x$
$\delta A = (A_p - A_m) / D_x$

The EOSHIFT function is a utility routine that allows the locations of array elements to be shifted by a specified amount. It inserts specified values in the appropriate end of an array. The first line shifts A such that $A_p(1) = A(2)$, $A_p(2) = A(3)$ etc. and the last value $A_p(n)$ is set to $A(n\text{-}1)$. The second line shifts A the other way and sets the first element of A_m to 0. The third loads packs all elements of D_x with the value $2 * \Delta x$. Then the finite difference formula

$$\frac{dA_i}{dx} = \left(\frac{A_{i+1} - A_{i-1}}{2\Delta x} \right)$$

is calculated in parallel and the results are stored in δA. Suppose array A contains five element such that $A(1) = 1$, $A(2) = 2$ etc. The shifting of array A, and the filling of arrays A_p and A_m, are illustrated in Figure 2. This step requires the processors to communicate only with near neighbors. Next D_x is filled; for the purpose of this example, Δx is set equal to 1. The final calculation is then done purely in parallel, with no communication required with other processors.

PROBLEM STATEMENT

To study the feasibility of the massively parallel explicit scheme, a sample problem is solved. The sample problem is the transpiration cooling problem. This problem resembles the ablative problem except that chemical reactions are absent. In this problem, a plate made of isotropic porous material heated on one side is cooled by sending a gas flow from the cool side to the hot side. This sort of cooling is under consideration for the cooling of turbine blades, combustors, and, the skins and leading edges of hypersonic vehicles. The problem is illustrated in Figure 3.

It is assumed that the pressure, temperature, and tractions are applied uniformly over the surface of the plate, so that a gradient is established only in the thickness direction. On one side ($x = h$) of the porous material, the pressure, temperature and traction are specified. On the other side ($x = 0$), the pressure, temperature and displacements are

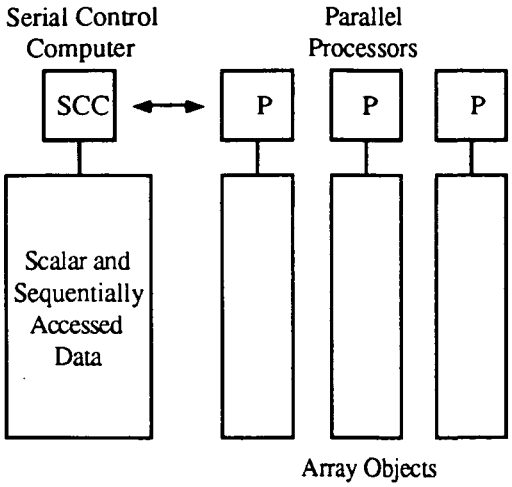

Figure 1. Connection Machine Architecture[6]

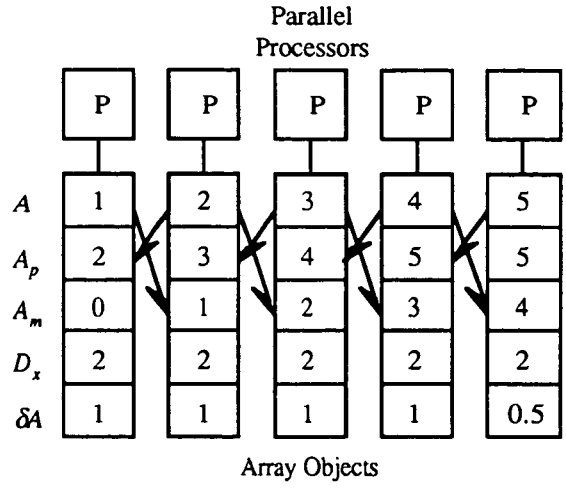

Figure 2 Example computation using parallel processors

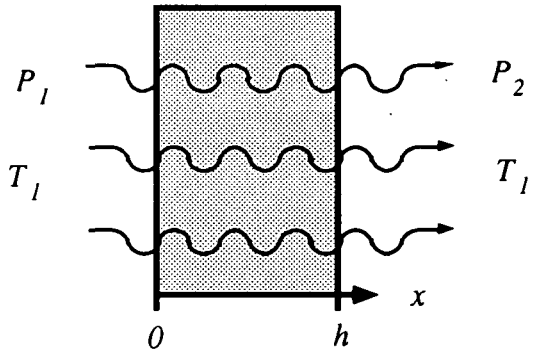

Figure 3. Transpiration cooling of a plate of thickness h

specified. It is desired in this problem to find the pressure, temperature, and stress distributions throughout the thickness of the porous plate as a function of time and space.

GOVERNING EQUATIONS

The 1-D continuity equation and Darcy's Law are combined to model the gas flow inside the porous plate. The continuity equation and Darcy's Law are shown in Eqs. 1 and 2, respectively,

$$\frac{\partial m_g}{\partial x} = -\frac{\partial \hat{m}_g}{\partial x} \tag{1}$$

and

$$\hat{m}_g = -\frac{\rho_g \gamma}{\mu_g}\frac{\partial p}{\partial x} \tag{2}$$

where m_g is the mass of gas per unit volume of the porous plate, $\hat{m}_g$ is the gas mass flux, ρ_g is the gas density, γ is the permeability of the porous plate, μ_g is the viscosity of the gas, and p is the gas pressure.

The gas inside the porous material is assumed to behave ideally, so that the pressure can be given by the ideal gas law given in Eq. 3,

$$p = \frac{RTm_g}{M\phi} \tag{3}$$

where R is the universal gas constant, M is the molecular weight of the gas, ϕ is the porosity of the plate, and T is the temperature. Note that $m_g = \phi\rho_g$.

The temperature inside the porous plate is governed by the energy equation

$$(c_{ps}m_s + c_{pg}m_g)\frac{\partial T}{\partial t} = k\frac{\partial^2 T}{\partial x^2} - c_{pg}\hat{m}_g\frac{\partial T}{\partial x} \tag{4}$$

where c_{ps} and c_{pg} are specific heat capacities for the solid and gas, m_s is the mass of the solid per unit volume of the porous plate, and k is the heat conductivity of the porous material.

To derive the governing equations for the stress inside the porous plate, one starts with the equation,

$$\rho\ddot{u}_i + c\dot{u}_i = \sigma_{ij,j} \tag{5}$$

where dot indicates derivative with respect to time, and comma indicates derivative with respect to space, ρ is the density of the solid, u_i is the displacement vector, c is the damping coefficient, and σ_{ij} is the total stress tensor. It is necessary to introduce a damping term into the governing equation so that the steady state solution can be obtained. The total stress tensor is defined in reference 1 as

$$\sigma_{ij} = \sigma_{ij}^m - p\delta_{ij} \tag{6}$$

where σ_{ij}^m is the mechanical stress tensor and δ_{ij} is the Kronecker delta. In order to apply the explicit scheme, the total stress tensor needs to be expressed in terms of displacement in Eq. 5. To accomplish this, the definition of the total strain tensor is needed[1]. The total strain tensor is defined as

$$\varepsilon_{ij} = S_{ijkl}\sigma_{kl}^m + \Lambda_{ij}\Delta p + \alpha_{ij}\Delta T \tag{7}$$

where S_{ijkl} is the compliance tensor of virgin material, Λ_{ij} is the pressure compliance tensor given by Reference 8, α_{ij} is the thermal expansion tensor, Δp is the change in internal pressure, and ΔT is the change in temperature.

The next step is to multiply Eq. 7 by C_{mnij} (the stiffness tensor of the virgin material) and solve for the mechanical stress tensor. The result is shown below,

$$\sigma_{mn}^m = C_{mnij}\varepsilon_{ij} - C_{mnij}\Lambda_{ij}\Delta p - C_{mnij}\alpha_{ij}\Delta T \tag{8}$$

Substituting Eq. 8 into Eq. 6 and substituting the result into Eq. 5, and applying the strain displacement relations

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \tag{9}$$

and assuming that pressure, temperature, and traction are uniform over the plate surface and hence spatial derivatives in the in-plane directions are zero[1], the following three equations for the displacements are obtained,

$$\rho\ddot{u}_1 + c\dot{u}_1 = (Du_{1,3})_{,3} \tag{10}$$

$$\rho\ddot{u}_2 + c\dot{u}_2 = (Du_{2,3})_{,3} \tag{11}$$

$$\rho\ddot{u}_3 + c\dot{u}_3 = (\frac{1}{A}u_{3,3} - \frac{(A+B)}{A}\Delta p - \frac{C}{A}\Delta T)_{,3} \tag{12}$$

where

$$A = \frac{2\upsilon^2 + \upsilon - 1}{E(\upsilon - 1)} \tag{13}$$

$$B = -\frac{\Lambda(\upsilon + 1)}{(\upsilon - 1)} \tag{14}$$

$$C = -\frac{\alpha(\upsilon + 1)}{(\upsilon - 1)} \tag{15}$$

$$D = \frac{E}{2(1 + \upsilon)} \tag{16}$$

$$\Lambda = -\frac{1-2\upsilon}{E} \tag{17}$$

E is the Young's modulus, υ is the Poisson's ratio, and the pressure compliance constant Λ is taken from reference 8 for the isotropic and dilute porosity case.

IMPLEMENTATION

In order to apply the explicit scheme, Eqs. 1 and 4 need to be cast in terms of ρ_g, T, and their temporal and spatial derivatives. After some algebraic manipulations, Eqs. 1 and 4 take on the following form

$$\frac{\partial \rho_g}{\partial t} = \frac{\gamma R}{\phi \mu M}\left[T\left(\frac{\partial \rho_g}{\partial x}\right)^2 + 3\rho_g\left(\frac{\partial T}{\partial x}\right)\left(\frac{\partial \rho_g}{\partial x}\right) + \rho_g T\frac{\partial^2 \rho_g}{\partial x^2} + \rho_g^2\frac{\partial^2 T}{\partial x^2}\right] \tag{18}$$

$$\frac{\partial T}{\partial t} = \frac{1}{(c_{ps}(1-\phi)\rho + c_{pg}\phi\rho_g)}\left[k\frac{\partial^2 T}{\partial x^2} + \frac{c_{pg}\gamma R}{\mu M}\left(\rho_g T\frac{\partial \rho_g}{\partial x}\frac{\partial T}{\partial x} + \rho_g^2\left(\frac{\partial T}{\partial x}\right)^2\right)\right] \tag{19}$$

This is achieved by using Eqs. 2 and 3 and assuming that the material properties are constant in space and time. Equations 18 and 19 are coupled nonlinear differential equations. These two equations can be solved by finite difference or FEM schemes. In both implicit and FEM schemes, the governing equations are discretized, and the resulting equations are nonlinear algebraic equations. These must be linearized and then solved iteratively, with each iteration requiring the inversion of a potentially large matrix. In the explicit finite difference scheme, the discretized governing equations are linear in the unknown quantities, and these can be solved for explicitly from the values which are known from previous times steps.

Equations 10-12 and 18-19 are all cast in explicit finite difference form and are solved simultaneously for displacements, temperature, and pressure distributions through the thickness of the porous plate. Stress distributions are then calculated by using Eqs. 8 and 9.

In Eqs. 10-12, central difference is applied to both temporal and spatial derivatives. In Eqs. 18 and 19, forward difference is applied to the temporal derivative, backward difference is applied to all first order spatial derivatives, and central difference is applied to all second order spatial order derivatives. These approximation schemes are chosen to ensure stability of the explicit scheme.

Equations 20- 24 are the finite difference equations used in the parallel algorithm.

$$\rho_{gi}^{j+1} = \frac{\Delta t\gamma R}{\phi\mu_g M}\left[T_i^j\left(\frac{\rho_{gi}^j - \rho_{gi-1}^j}{\Delta x}\right)^2 + 3\rho_{gi}^j\left(\frac{T_{i+1}^j - T_{i-1}^j}{2\Delta x}\right) + \rho_{gi}^j T_i^j\left(\frac{\rho_{gi+1}^j - 2\rho_{gi}^j + \rho_{gi-1}^j}{\Delta x^2}\right) + {\rho_{gi}^j}^2\left(\frac{T_{i+1}^j - 2T_i^j + T_{i-1}^j}{\Delta x^2}\right)\right] + \rho_{gi}^j \tag{20}$$

$$T_i^{j+1} = \frac{\Delta t}{\left(c_{ps}(1-\phi)\rho + c_{pg}\phi\rho_{gi}^j\right)}\left[k\left(\frac{T_{i+1}^j - 2T_i^j + T_{i-1}^j}{\Delta x^2}\right) + \frac{c_{pg}\gamma R}{\mu_g M}\left(\rho_{gi}^j T_i^j\left(\frac{\rho_{gi}^j - \rho_{gi-1}^j}{\Delta x}\right) + \left(\frac{T_{i+1}^j - T_{i-1}^j}{2\Delta x}\right) + {\rho_{gi}^j}^2\left(\frac{T_{i+1}^j - T_{i-1}^j}{2\Delta x}\right)^2\right)\right] + T_i^j \tag{21}$$

$$u_{1i}^{j+1} = \frac{D}{(1 + c\Delta t/2\rho)}\left[\frac{u_{1i+1}^j - 2u_{1i}^j + u_{1i-1}^j}{\Delta x^2}\right] + 2u_{1i}^j + \left(\frac{c\Delta t}{2\rho} - 1\right)u_{1i}^{j-1} \tag{22}$$

$$u_{2i}^{j+1} = \frac{D}{(1 + c\Delta t/2\rho)}\left[\frac{u_{2i+1}^j - 2u_{2i}^j + u_{2i-1}^j}{\Delta x^2}\right] + 2u_{2i}^j + \left(\frac{c\Delta t}{2\rho} - 1\right)u_{2i}^{j-1} \tag{23}$$

$$u_{3i}^{j+1} = \frac{1}{(1+c\Delta t/2\rho)A}\left[\left(\frac{u_{3i+1}^{j} - 2u_{3i}^{j} + u_{3i-1}^{j}}{\Delta x^2}\right) - (A+B)\left(\frac{p_{i+1}^{j} - p_{i-1}^{j}}{2\Delta x}\right)\right.$$
$$\left. - C\left(\frac{T_{i+1}^{j} - T_{i-1}^{j}}{2\Delta x}\right)\right] + 2u_{3i}^{j} + \left(\frac{c\Delta t}{2\rho} - 1\right)u_{3i}^{j-1} \qquad (24)$$

In the parallel algorithm, the spatial derivatives are calculated simultaneously. This is achieved by declaring multiple conformable arrays that contain appropriate data elements. By adding and/or subtracting the arrays, the finite difference approximation for the spatial derivatives can be obtained in unison for all nodes. A simple example of how this is done has been given in the parallel computing section.

EXAMPLE PROBLEM

The transpiration cooling problem is solved for a porous aluminum plate cooled by water vapor. The problem is illustrated in Figure 3. The properties for the porous plate and the cooling gas are listed in Table I.

Table I. Properties for Porous Plate and Cooling Gas

c_{ps}	c_{pg}	ϕ	k	γ	μ_g	M	E	υ	c	ρ	h	α
$\frac{J}{kg \cdot K}$	$\frac{J}{kg \cdot K}$		$\frac{W}{m \cdot K}$	m^2	$\frac{N \cdot s}{m^2}$	$\frac{kg}{kmole}$	Pa		$\frac{N \cdot s}{m^4}$	$\frac{kg}{m^3}$	mm	$\frac{1}{K}$
167	2000	0.05	150	1.0×10^{17}	1.0×10^{-5}	18	1.0×10^{9}	0.3	1.0×10^{8}	2700	30	1.2×10^{-5}

The temperature and pressure values are fixed on both sides of the porous plate. On one side of the plate ($x = 0$), the displacements are fixed. On the other side of the plate ($x = 30mm$), the surface tractions are prescribed. The values used for the temperature, pressure, and surface tractions on the boundaries are listed in Table II.

Table II. Boundary Conditions

$T_1\ (K)$	$T_2\ (K)$	$P_1\ (Pa)$	$P_2\ (Pa)$	$\tau_{1b}\ (Pa)$	$\tau_{2b}\ (Pa)$	$\tau_{3b}\ (Pa)$
273	373	2×10^5 or 2×10^6	1×10^5	0	0	0

Initially, the temperature and pressure distributions in the porous plate are uniform. The porous plate also has zero displacement and velocity initially. The values used for the initial conditions are given in Table III.

Table III. Initial Conditions

$T_{initial}$	$p_{initial}$	$u_{1initial}$	$u_{2initial}$	$u_{3initial}$	$\dot{u}_{1initial}$	$\dot{u}_{2initial}$	$\dot{u}_{3initial}$
273 K	0.1 MPa	0 m	0 m	0 m	0 m/s	0 m/s	0 m/s

Unless otherwise specified, all results shown below are generated with the values listed in Table I - III.

RESULTS

STEADY-STATE DISTRIBUTIONS

Steady-state distributions for temperature, pressure, and stress are obtained to check against known solutions. The explicit method cannot calculate steady state results directly, so the transient problem is solved by marching forward in time. The steady state solution is assumed to have been reached when the temperature, pressure, and displacement distributions at two time steps, t and $t + \Delta t$ differ by less than a specified tolerance value.

An analytical expression for steady-state temperature distribution can be derived from Eq. 4 by setting the left-

hand-side to zero and assuming that all coefficients on the right-hand-side are constant. The solution is[9]

$$T(x) = \frac{(T_2 - T_1)}{(1 - \exp(h/\delta))}(1 - \exp(x/\delta)) + T_1 \qquad (25)$$

where

$$\delta = \frac{k}{\dot{m}_g c_{pg}}$$

In Figure 4, both Eq. 25 and the numerical results are plotted for two different mass fluxes, corresponding to the two different values of P_1 listed in Table II. As can be seen, the numerical results agree well with the analytical solutions. As the mass flux increases, the transpiration cooling effect becomes more apparent.

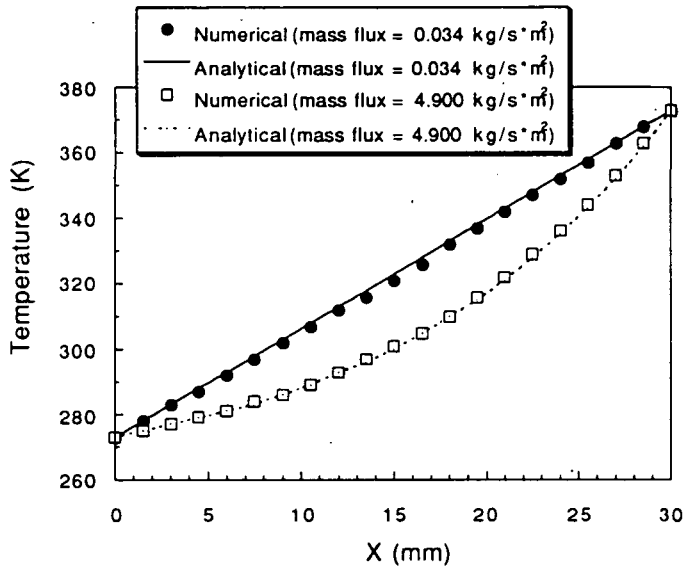

Figure 4. Steady-State Temperature Distributions

An analytical expression for the steady-state pressure distribution can be derived from Eq. 2 if the temperature distribution, $\dot{m}_g$, γ, and μ_g are constant. The derived analytical expression is[10],

$$P(x) = \sqrt{(P_{max}^2 - (P_{max}^2 - P_2^2)\frac{x}{h}} \qquad (25)$$

where

$$P_{max} = \sqrt{\frac{2h\dot{m}_g RT\mu_g}{M\gamma} - P_2^2}$$

In Figure 5, both Eq. 25 and the numerical results are plotted. As can be seen from Figure 5, the numerical results agree well with the analytical solutions.

To verify the stress distribution predicted by the code, a constant surface traction is applied on the surface of the plate while temperature and pressure distributions are uniform. The stress distribution due to this condition should also be uniform and has a value of the applied surface traction. As expected, the calculated steady state stress distribution is uniform and the magnitudes are equal to the applied surface tractions.

TRANSIENT DISTRIBUTIONS

Results from the transient analysis are shown for the first case studied, with P_1 = 0.5 MPa. The transient temperature distributions are shown in Figure 6. The temperature distribution rises from the initial condition to the steady-state linear distribution in approximately one second. Note that an approximate analytical solution for the transient temperature distribution can be obtained for the case of small mass flux, $\dot{m}_g$, by ignoring the second term on the left and right hand sides of Eq. 4. This analytical solution can be found in Reference 11. It is in good

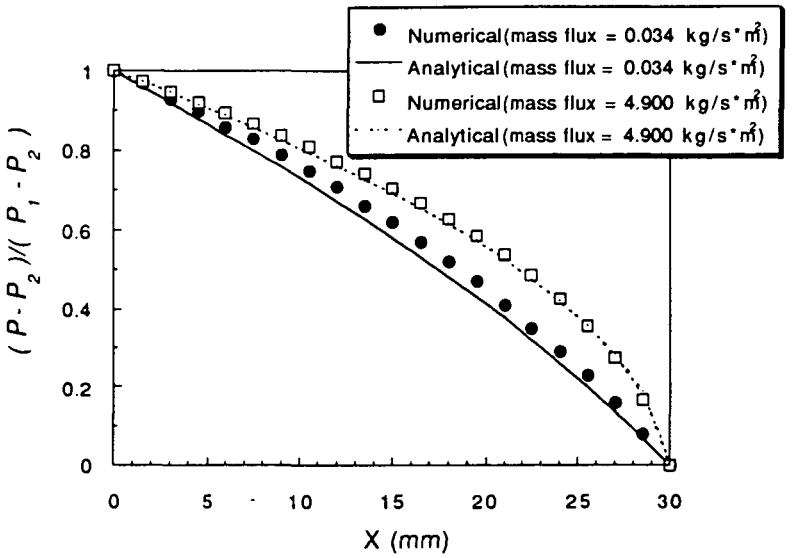

Figure 5. Steady-State Pressure Distribution

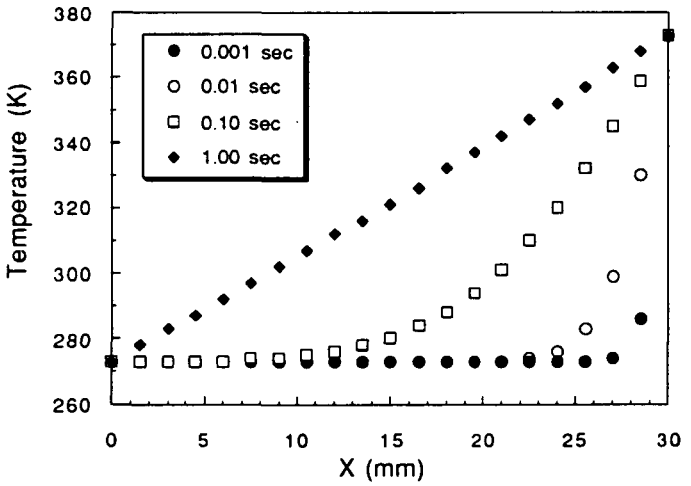

Figure 6. Transient Temperature Distributions

agreement with the results shown in Figure 6. The fast response time is due to the high thermal conductivity assumed in this example.

The transient pressure distributions are shown in Figure 7. There are two things worth noting from the figure. The first thing is the small spatial oscillation in the pressure distribution early in the analysis. This small oscillation in pressure is due to a very sharp temperature rise near the boundary (x=30 mm). This sharp rise in temperature can be seen from Figure 7 at 0.001 second. This sharp rise in temperature near the boundary increases the gas pressure before the flow of gas from the other boundary (x=0 mm) has reached there. The second thing worth noting is that the pressure distribution reaches steady state about 100 times faster than the temperature distribution.

The explicit solution of the stress equations (Eqs. 22-24) requires damping or a steady state solution will never be reached. In the cases studied here, an unrealistically high damping term is used to suppress dynamic behavior (i.e. stress waves) which are excited by the sudden application of the boundary conditions at the start of the calculation. The transient stress distributions are shown in Figure 8. Notice that the stress distributions have the same shape as the pressure distributions. Without surface tractions, internal pressure and temperature gradients are the only sources of stress. However, in this case the contribution due to pressure ($O(10^6)$ Pa) is much greater than the contribution due to temperature gradient ($O(10^2)$ Pa). Therefore, the stress distributions take on the same shape as the pressure distributions. If the value of the damping coefficient c is low enough, stress waves are observed in the stress distribution. Such oscillation is shown in Figure 9.

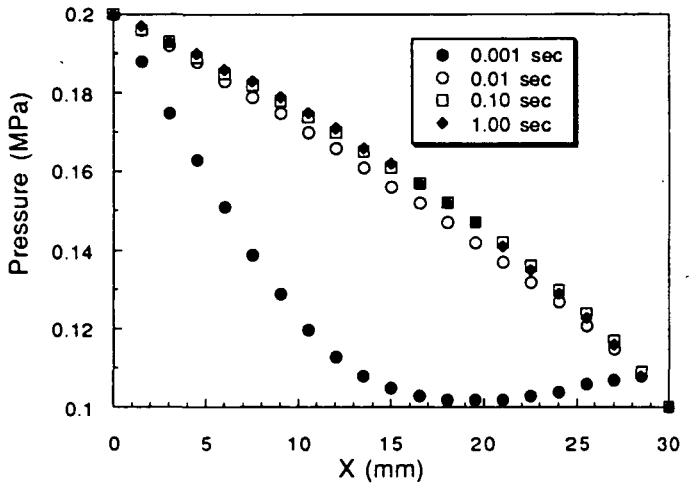

Figure 7. Transient Pressure Distributions

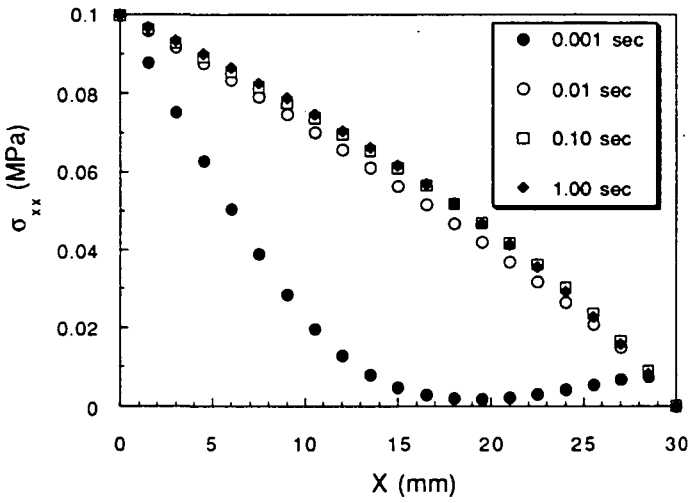

Figure 8. Transient Stress Distributions

Comparing the temperature, pressure, and stress solutions, it is important to note the widely varying time scales. Temperature reaches steady state in about one second. This is a very low time for a thermal problem, and is due to the high conductivity of the material used in the sample problem. Pressure reaches steady state in much less time–about 0.01 seconds. Stress does not tend to a steady state at all without damping. The time scale of the stress problem is governed by the transit of stress waves across the plate, which in this example takes on the order of milliseconds.

Two observations follow directly from the differences of time scales. If a single time step value Δt is used for all of the calculations, it must be small enough to assure stability of the fastest processes modeled (in this case, the stresses), and hence the other calculation are done many times more often than is necessary for stability. This would indicate that increased efficiency would be gained by solving the pressure , temperature, and stress problems in as independent a manner as possible. It also tends to validate the previously used analytical assumptions that the stress state is essentially static on the time scale of the temperature and pressure problems, and that the pressure state is essentially steady state on the time scale of the thermal problem.

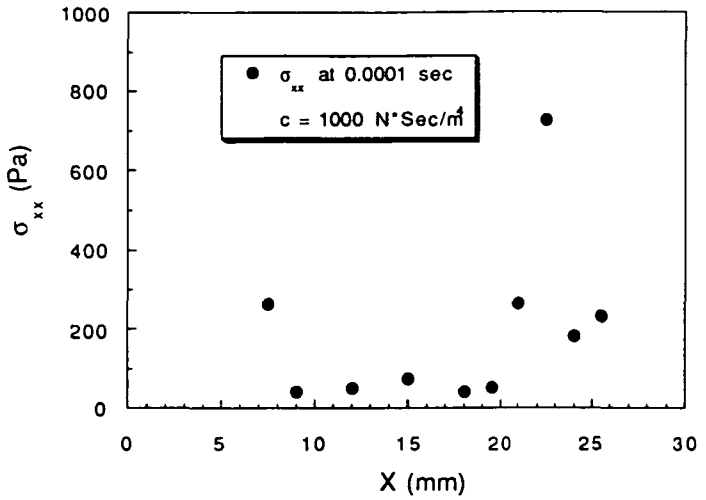

Figure 9. Oscillatory Stress Distribution

SERIAL VS PARALLEL SCHEMES

In Table IV, the performance data between the serial and parallel schemes of the code are compared. Three different levels of spatial discretizations are used: 11 nodes, 21 nodes, and 51 nodes. For low level of discretization, the averaged CPU time for both serial and parallel schemes are practically the same. As the level of discretization becomes finer, the increase in CPU time for the serial scheme is considerably greater than the parallel scheme.

Table IV. Serial Scheme vs. Parallel Scheme

Number of Nodes	Parallel Version CPU Time (s)	Serial Version CPU Time (s)
11 Nodes	13.4	15.0
21 Nodes	13.7	17.0
51 Nodes	15.4	36.6

This implies that the parallel version is more advantageous than the serial version as the problem becomes more complex.

CONCLUSIONS

The transpiration cooling problem is solved numerically with the explicit finite difference scheme. The numerical results compare favorably with available analytical solutions. The numerical solution demonstrates the feasibility of using the explicit finite difference scheme on a massively parallel computer to solve this class of problems.

By using the explicit scheme, some interesting transient results are obtained. For example, oscillations in pressure and stress due to the sharp temperature rise at the boundary at the beginning of the problem are obtained. This type of result cannot be easily obtained by other schemes such as implicit and finite element schemes. On the other hand, pressure distributions have been shown to reach a steady state much faster than temperature distributions. This result implies that for preliminary analyses of this type of problem, the simplifying assumptions that the stresses and pressure are steady state on the time scale of the temperature problem are reasonable.

The performance of parallel and serial versions is compared. For small scale problems, the parallel and serial versions used about the same amount of CPU time on the CM-5. However, as the problem size increases (in this case, through finer spatial discretization) the parallel version becomes more efficient than the serial version.

The explicit scheme holds promise as a numerical scheme for the ablative problem. This problem is more complex than the transpiration cooling problem in that there are chemical reactions. Currently, work on the ablative problem is in progress.

REFERENCES

1. McManus, H.L., and Springer, G.S., "High Temperature Thermomechanical Behavior of Carbon-Phenolic and Carbon-Carbon Composites, I. Analysis", *Journal of Composite Materials,* Vol. 26, No. 2, 1992, pp 206-229.
2. Sullivan, R.M. and Salamon, N.J., "A Theoretical Formulation of a Finite Element to Model the Thermomechanical Behavior of Charring Carbon-Phenolic Insulators", *1989 JANNAF Rocket Nozzle Technology Subcommittee Meeting,* Naval Surface Warfare Center, Silver Spring, MD, 1989.
3. Kuhlman, T.L., "Thermo-Chemical-Structural Analysis of Carbon-Phenolic Composites with Pore Pressure and Pyrolysis Effects", PDA-89-4097-00, Prototype Development Associates, Costa Mesa, CA, 1989.
4. Henderson, J.B. and Wiecek, T.E., "A Mathematical Model to Predict the Thermal Response of Decomposing, Expanding Polymer Composites", *Journal of Composite Materials,* Vol. 21, (1987), pp 373-393.
5. Agarwal, R.K., "Parallel Computers and Large Problems in Industry", *Computation Methods in Applied Sciences*, the First European Conference on Numerical Methods in Engineering, September 7-11 1992, Brussels, Belgium, pp 1-11.
6. "CM Fortran Manual", Thinking Machines Corporation, Cambridge, Massachusetts, Jan., 1993.
7. Walker, D.W., "Particle-Cell Plasma Simulation Codes on the Connection Machine", *Parallel Methods on Large-Scale Structural Analysis and Physics Applications,* Symposium on Parallel Methods and Large-Scale Structural Analysis and Physics Applications, February 5-6 1991 at NASA Langley Research Center, Hampton, VA, pp 307-318.
8. McManus, H.L., "Application of Poroelasticity Theory of Decomposing Composites", 33rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Dallas, TX, April 1992, pp. 3172-3178.
9. Glass, D., Personal Communication, 1993.
10. McManus, H.L.N., "Exact Solutions to Quasi-Steady State Internal Gas Flow Problems", *1992 JANNAF Rocket Nozzle Technology Subcommittee Meeting*, Lockheed Missiles and Space Company, Sunnyvale, California, December 8-10, 1992.
11. Boas, M.L., "Mathematical Methods in the Physical Sciences", John Wiley & Sons, 1966, pp. 550-552.