CONVECTION EQUATION MODELING: A NON-ITERATIVE DIRECT MATRIX SOLUTION ALGORITHM FOR USE WITH SINDA

N94-23651

Dean S. Schrage
Sverdrup Technology Inc.
Lewis Research Center Group
Brook Park, Ohio 44142

ABSTRACT

The determination of the boundary conditions for a component-level analysis, applying discrete finite element and finite difference modeling techniques often requires an analysis of complex coupled phenomenon that cannot be described algebraically. For example, an analysis of the temperature field of a coldplate surface with an integral fluid loop requires a solution to the parabolic heat equation and also requires the boundary conditions that describe the local fluid temperature. However, the local fluid temperature is described by a convection equation that can only be solved with the knowledge of the locally-coupled coldplate temperatures. Generally speaking, it is not computationally efficient, and sometimes, not even possible to perform a direct, coupled phenomenon analysis of the component-level and boundary condition models within a single analysis code. An alternative is to perform a disjoint analysis, but transmit the necessary information between models during the simulation to provide an indirect coupling. For this approach to be effective, the component-level model retains full detail while the boundary condition model is simplified to provide a fast, first-order prediction of the phenomenon in question. Specifically for the present study, the coldplate structure is analyzed with a discrete, numerical model (SINDA) while the fluid loop convection equation is analyzed with a discrete, analytical model (direct matrix solution). This indirect coupling allows a satisfactory prediction of the boundary condition, while not subjugating the overall computational efficiency of the component-level analysis. In the present study a discussion of the complete analysis of the derivation and direct matrix solution algorithm of the convection equation is presented. Discretization is analyzed and discussed to extend of solution accuracy, stability and computation speed. Case studies considering a pulsed and harmonic inlet disturbance to the fluid loop are analyzed to assist in the discussion of numerical dissipation and accuracy. In addition, the issues of code melding or integration with standard class solvers such as SINDA are discussed to advise the user of the potential problems to be encountered.

NOMENCLATURE

C	=	Courant Number (U Δt / Δx)	
C_p	=	specific heat	(W s / kg °C)
e	=	total specific energy	(W s / kg)

PRECEDING PAGE BLANK NOT FILMED

$E_{..}$	=	global error	
G	=	thermal conductance	(W / °C)
h	=	enthalpy	(W s / kg)
$h^{\cdot}$	=	effective heat transfer coefficient	(W / m^2)
H	=	enthalpy flux	(W)
k	=	thermal conductivity	(W / m °C)
L	=	total tube length	(m)
$\dot{m}$	=	mass flowrate	(kg / s)
P	=	pressure	(N / m^2)
$\dot{Q}$	=	heat load	(W)
t	=	time	(s)
T	=	temperature	(°C)
$\tilde{T}$	=	temperature, actual equation solved	(°C)
u	=	specific internal energy	(W s / kg)
U	=	flow velocity	(m / s)
x	=	space	(m)

Greek

$\alpha_{\cdot}$	=	thermal diffusivity ($k / \rho C_p$)	(m^2 / s)
$\alpha^{\cdot}$	=	effective thermal diffusivity flux	(s^{-1})
Δ	=	denotes difference	
ρ	=	density	(kg / m^3)
τ	=	fluid transit time	(s)

Superscripts and Subscripts

i	=	space increment
j	=	time increment

INTRODUCTION

The process of convection (or advection) involves the transport of a scalar property within the confines of a motive flow, traveling a finite velocity. If the convected quantity represents the fluid enthalpy (or temperature as will be shown), then energy is transported by the convection of the fluid at a particular enthalpy and flow velocity. In this case, a disturbance in the inlet temperature (enthalpy) would be convected along the length of a conduit in space and time. By constructing a differential control volume and performing a transient energy balance, the first-order wave or convection equation can be derived by balancing heat addition with transient heating and convection enthalpy. Figure 1 shows the control volume with an inlet and outlet enthalpy flux (H) and differential heating (dQ) from a target sink surface (coldplate). The transient energy balance can be written according to:

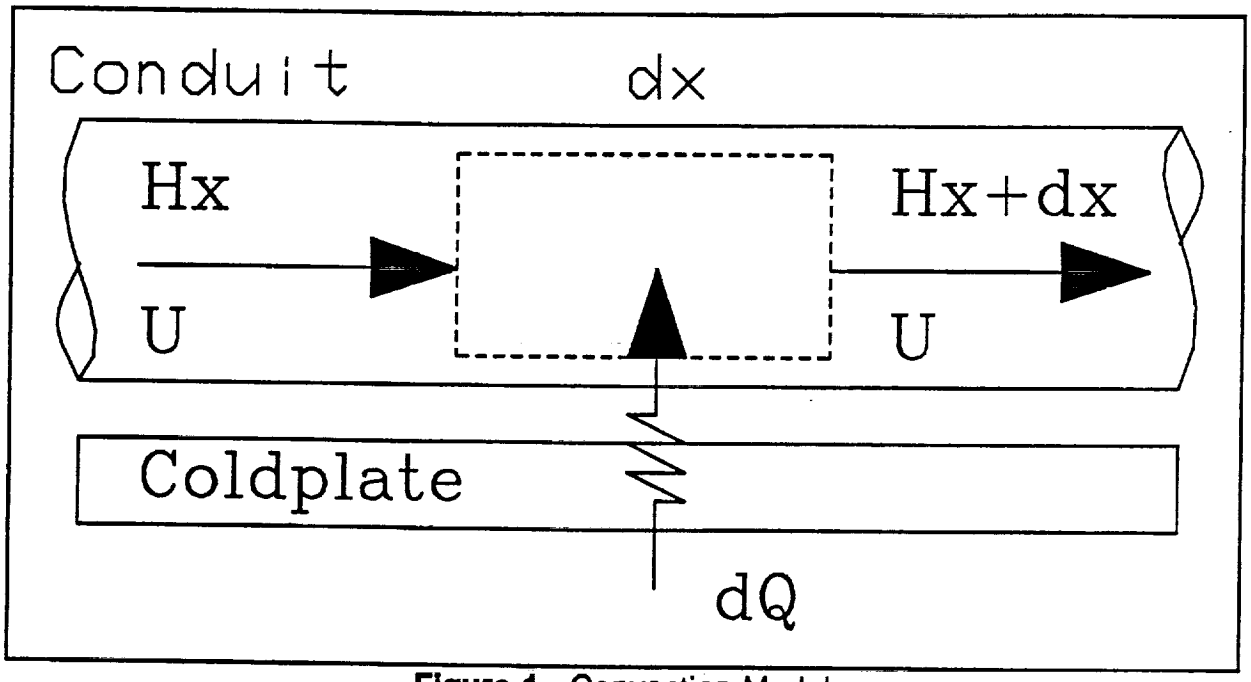

Figure 1. Convection Model

$$d\dot{Q} = \frac{\partial}{\partial t}\int_{dM} e\, dM + H_{x+dx} - H_x$$

$$d\dot{Q} = d\dot{Q}_T + d\dot{Q}_A \qquad (1)$$

Terms e, H, dQ are the total specific energy, enthalpy flux and total heat conduction respectively. The total specific energy is the sum of the internal, kinetic and potential energies while the enthalpy flux is a product of the local mass flowrate and local flow enthalpy. Terms dQ_T and dQ_A are transverse and axial heat conduction terms, respectively. The transverse heat conduction term represents the conduction into the control volume from the local coldplate, while the axial heat conduction term represents the net heat conduction into the control volume from the upstream and downstream fluid layers. Generally, the axial heat conduction term is assumed to be small in comparison to the convection process and for the present study, axial conduction is neglected[1]. The above terms can be expanded according to:

[1] *By normalizing the equations, assuming the convection term is of the same magnitude as the capacitance term, it can be shown that the axial conduction term will be negligible provided that:*

$$\frac{\tau k}{L^2 \rho C} = \frac{\tau \alpha}{L^2} \ll 1$$

For the present study this ratio is approximately 0.1.

$$H_{x+dx} - H_x = \dot{m}h_x - \left(\dot{m}h_x + \frac{\partial \dot{m}h}{\partial x}dx\right)$$

$$e = u + KE + PE$$

$$u = h - \frac{P}{\rho} \qquad (2)$$

$$d\dot{Q} = dG\,(T_\infty - T)$$

At this point it is appropriate to make several simplifying assumptions. First, we can generally neglect changes in kinetic and potential energy. Next, if we assume an incompressible fluid, then the pressure dependence can be eliminated and we can formulate a simple state equation to relate enthalpy to temperature. With this, Eqns. 1 and 2 can be combined to yield:

$$\frac{\partial T}{\partial t} + U\frac{\partial T}{\partial x} = \alpha^{\bullet}(T_\infty - T)$$

$$U = \frac{L}{\tau} \qquad (3)$$

Equation 3 is the first-order convection equation that describes the local fluid temperature (T) as a function of both time and one-dimensional space along the length of the conduit. Parameters U and $\alpha^{\bullet}$ are the fluid flow velocity and effective thermal diffusivity flux, respectively, and is a compact notation to describing the relevant features of the convection process (geometry, flow velocity and coldplate coupling, thermal-physical properties). The effective thermal diffusivity flux is written:

$$\alpha^{\bullet} = \frac{h^{\bullet} D_H L}{\tau \dot{m} C} \qquad (4)$$

This parameter describes the relative strength of the thermal coupling between the tube and the local sink surface. For Example, when $\tau \rightarrow 0$, the energy transport is dominated by local conduction to the sink; when $h^{\bullet} \rightarrow 0$, the energy transport is dominated by fluid convection.

In its appearance, Eqn. 3 is linear, in-homogeneous[2] with constant coefficients and should yield to an analytical solution.[3] However, as addressed in the abstract, the present

[2] *The equation is in-homogeneous because of the presence of the source term, T_∞ which varies in time and space.*

[3] *An inhomogeneous ordinary differential equation can be described along a characteristic line ($\Psi = x - Ut$) and solved specifically for those regions where either the initial conditions or the boundary conditions effect the solution.*

study deals with the feature of a coupled boundary condition. The source term in Eqn. 3 is determined from the coupled boundary condition of the local coldplate temperature.[4] Thus the source term cannot be decoupled from the dependent variable - at least not analytically. This interdependence between boundary conditions is the impetus for the present study - to develop a suitable means to coordinating a solution of Eqn. 3 with the boundary conditions (coldplate temperatures) put forth from a discrete finite difference model.

ANALYSIS

If, because of the coldplate coupling, that analytical solutions to Eqn. 3 are intractable, we must resort to a numerical solution, preferably a finite difference technique. Consider then a discrete numerical approach. Equation 3 appears innocuous. The partial derivatives are first order, not mixed and the equation is linear. It would be desirable to develop an explicit discrete equation, or discretization, that would not require iteration at each time step - this in order to generate a rapid disjoint solution discussed in the abstract. To satisfy this, we could use forward differencing in time and central differencing in space:

$$\frac{\partial T}{\partial t} \approx \frac{T_i^{j+1} - T_i^j}{\Delta t}$$
$$\frac{\partial T}{\partial x} \approx \frac{T_{i+1}^j - T_{i-1}^j}{2\Delta x} \qquad (5)$$

Superscripts (j) indicate time and subscripts (i) indicate space. Substituting Eqns. 5 into Eqn. 3, and assuming for the sake of convenience that the fluid loop is thermally decoupled from the coldplate ($\alpha^* = 0$), we can write the discrete difference equation (kernel):

$$T_i^{j+1} = T_i^j + \frac{C}{2}(T_{i+1}^j - T_{i-1}^j)$$
$$C = \frac{U\Delta t}{\Delta x} \qquad (6)$$

This kernel is referred to as the Euler forward, centered difference (EFCD). Equation 6 is explicit such that we solve for the future time (j+1) temperatures based on old (j) values, so the numerical solution would consist of a simple marching scheme. Without the need for iteration, the EFCD kernel is computationally efficient. However, this is not a sufficient criteria to select this kernel - above all, the scheme must also be numerically stable. We can evaluate stability in a macroscopic manner by computing the

[4] *The coldplate temperature is given from the solution of the parabolic heat equation with the boundary conditions determined from the local fluid temperature.*

actual equation solved by the discrete representation of Eqn. 6. We do this by letting the discrete values interpolate the continuous values of an effective temperature ($\tilde{T}$), where $T_i^j \rightarrow \tilde{T}(t_i, x_i)$. If we expand, in a Taylor series, those discrete values away from i,j, then the actual equation solved is simply the original convection equation with a remainder term:

$$\frac{\partial \tilde{T}}{\partial t} + U \frac{\partial \tilde{T}}{\partial x} = -\frac{U^2 \Delta t}{2} \frac{\partial^2 \tilde{T}}{\partial x^2} + Order(\Delta t^2, \Delta x^2) \qquad (7)$$

The remainder term is composed of a second partial of T with respect to x (and other higher order mixed partials). This term is exactly representative of axial diffusion or conduction, but it is numerically generated and not physical. It is true that the magnitude of the axial conduction is small for a small value of Δt, but the fact that the leading coefficient is negative results in a solution that will always grow without bound after a finite number of recursive solutions of Eqn. 6. Thus, the Euler forward, centered difference kernel (Eqn. 6) is unstable and should not be used.

While the use of Taylor series to determine the actual equation solved is a suitable approach to eliminate those kernels that create deleterious artificial phenomenon, there are more sophisticated approaches to determining the subtleties of numerical stability of a differencing kernels (Strauss, 1992). These techniques are used to analyze some 12 difference kernels applied to the one-dimensional convection equation in Anderson et al. (1984); some general conclusions from the Anderson et al. study are:

- all differencing kernel possess inaccuracy as a result of truncation error
- truncation error creates numerical dissipation and dispersion phenomenon[5]

Crank-Nicolson Kernel

Selection of a kernel requires a careful trade of the computational requirements (explicit versus implicit solutions) and the effects of the spurious numerical phenomenon weighed against the assumptions applied to the system at hand[6]. In the present study, we desire a kernel that will produce a negligible dissipation and will require a minimum of computation. The former criteria tends to associate with implicit, centered point kernels, while the latter criteria, with non-iterative explicit kernels.

The Crank-Nicolson centered-difference (CNCD) kernel is implicit, can be

---

[5] *Dissipation results from even order partial derivative terms and has the effect of introducing a spurious axial conduction. Dispersion results from odd order partial derivative terms and has the effect of changing the phase relationships between waves. The combination of dissipation and dispersion is terms diffusion (Anderson et al., 1984).*

[6] *To elaborate, one should weigh the assumptions made in the analysis with the noted effects created by the kernel. For example, if the convection equation is derived by neglecting an axial conduction term, then a comparable amount of spurious dissipation would be considered acceptable. However, it is difficult to determining the magnitude of the injected numerical dissipation.*

described in a matrix form suitable for a direct matrix inversion and possess second order accuracy in time and space. The kernel is formed analogous to the EFCD kernel above, but includes a present-time space difference for stability:

$$\frac{\partial T}{\partial t} \approx \frac{T_l^{j+1} - T_l^j}{\Delta t}$$

$$\frac{\partial T}{\partial x} \approx \frac{\dfrac{T_{l+1}^{j+1} - T_{l-1}^{j+1}}{2\Delta x} + \dfrac{T_{l+1}^{j} - T_{l-1}^{j}}{2\Delta x}}{2} \qquad (8)$$

The actual equation solved by the CNCD kernel creates numerical dispersion, (a remaining odd-ordered partial derivative term), but is unconditionally stable for any time step.

At this point, we have discussed the development of the kernel with a constant space incremental (Δx). We can generalize the CNCD kernel to apply a variable spatial step size[7]. We begin by creating a differencing scheme to approximate the space derivative with varying step size according to:

$$\frac{df}{dx} \approx a\, f_{l-1} + b\, f_l + c\, f_{l+1} \qquad (9)$$

[7] *This is necessary in order to simulation fluid networks that have variable sized nodes - a feature in the present study.*

The leading coefficients, a, b, c are determined by applying a Taylor series expansion to approximate the function away for the discrete local i values[8]:

$$a = \frac{-1}{\Delta x_1(1+R)}$$

$$b = \frac{1 - R^2}{\Delta x_1(1+R)}$$

$$c = \frac{R^2}{\Delta x_1(1+R)} \qquad (10)$$

$$R = \frac{\Delta x_1}{\Delta x_2}$$

$$\Delta x_1 = x_i - x_{i-1}$$

$$\Delta x_2 = x_{i+1} - x_i$$

By combining Eqns. 9, 10 with Eqn. 3, the extended CNCD kernel can be written:

$$A\,T_{i-1}^{j+1} + B\,T_i^{j+1} + C\,T_{i+1}^{j+1} = D\,T_{i-1}^{j} + E\,T_i^{j} + F\,T_{i+1}^{j} + \alpha^* T_{\infty,j}$$

$$A = \frac{a\,U}{2}$$

$$B = \frac{1}{\Delta t} + \frac{b\,U}{2} + \frac{\alpha^*}{2}$$

$$C = \frac{c\,U}{2} \qquad (11)$$

$$D = -A$$

$$E = \frac{1}{\Delta t} - \frac{b\,U}{2} - \frac{\alpha^*}{2}$$

$$F = -C$$

Equation 11 describes a system of discrete equations for each node (i = 1 to N). In order to apply the kernel over the discrete domain from i = 1 to N we need to perform a conditioning of the first and last equations. At the first node (i = 1), a boundary value of $T_{i-1=0}$ is required. This value is the specified inlet fluid temperature to the fluid loop (as a function of time). At the last node, the kernel requires a value of T_{N+1}, but this value does not exist. This is circumvented by simply adjusting the local values of the derivative

[8] *The details of this are excluded, but can be found in most treatise on numerical analysis techniques. Briefly, the method involves substituting the Taylor series expansions for each discrete value and observing the coefficients of each derivative terms. Three independent equations are created to solve for a, b, c.*

coefficients, a, b, c to apply a backwards-differencing in space at this last node[9]:

$$\text{for } i=N$$

$$\bar{a} = \frac{-1}{\Delta x_1}$$

$$\bar{b} = \frac{1}{\Delta x_1} \qquad (12)$$

$$\bar{c} = 0$$

We can now generate a tri-diagonal matrix equation according to:

$$\begin{bmatrix} B & C & & & & & \\ A & B & C & & & & \\ & A & B & C & & & \\ & & \bullet & \bullet & \bullet & & \\ & & & \bullet & \bullet & \bullet & \\ & & & & A & B & C \\ & & & & & \bar{A} & \bar{B} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \bullet \\ \bullet \\ T_{N-2} \\ T_{N-1} \\ T_N \end{bmatrix}^{j+1} = \begin{bmatrix} E & F & & & & & \\ D & E & F & & & & \\ & D & E & F & & & \\ & & \bullet & \bullet & \bullet & & \\ & & & \bullet & \bullet & \bullet & \\ & & & & D & E & F \\ & & & & & \bar{E} & \bar{F} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \bullet \\ \bullet \\ T_{N-2} \\ T_{N-1} \\ T_N \end{bmatrix}^{j} + \begin{bmatrix} \alpha^* T_{\infty,1} + D\,T_0^j - A\,T_0^{j+1} \\ \alpha^* T_{\infty,2} \\ \alpha^* T_{\infty,3} \\ \bullet \\ \bullet \\ \alpha^* T_{\infty,N-2} \\ \alpha^* T_{\infty,N-1} \\ \alpha^* T_{\infty,N} \end{bmatrix} \qquad (13)$$

In compressed form, Eqn. 13 can be simplified and we can solve for the future time (j+1) values of temperature by multiplying both sides by the inverse of the leading matrix:

$$[M][T]^{j+1} = [M][T]^j + [\bar{T}_\infty]$$

$$[T]^{j+1} = [M]^{-1}[M][T]^j + [M]^{-1}[\bar{T}_\infty] \qquad (14)$$

At this stage, Eqn. 14 is usually solved numerically, typically with a Gauss-Siedel or an over-relaxation method. Under the best of conditions, either of these methods converge to a solution with a finite number of iterations with sucessive substitutions of pervious solutions. This process can be computationally expensive especially if the boundary conditions need to be updated frequently. Alternatively, because matrix M is sparse, square, tri-diagonal, we can apply a direct solution technique to determine M^{-1}. A well-established approach to finding the solution of Eqn. 14 is to use LU decomposition. Briefly, this technique involves creating lower and upper diagonal matrices that are manipulated to yield a direct solution with minimal storage requirements. Development

[9] *Because the backwards difference is only first order accurate in space at the last node, the overall order of the entire solution is somewhat less than second order - unfortunate indeed but unavoidable.*

of this method is quite standard and is excluded for the sake of brevity. The interested reader is referred to Gerald and Wheatly (1984).

The necessary conditions for the LU decomposition is that M must be diagonally dominant. This in effect, does pose a limit on the selection of time step Δt for a given Δx because these parameters describe the respective diagonals of M. Diagonal dominance will be observed if:

$$\left|\frac{1}{\Delta t} + \frac{b\,U}{2} + \frac{\alpha^*}{2}\right| \geq \left|\frac{a\,U}{2}\right| + \left|\frac{c\,U}{2}\right| \qquad (15)$$

In the special case of b = 0 (uniform node size) and $\alpha^* = 0$ (coldplate decoupled), Eqn. 15 can be more appropriately expressed as a function of the Courant number (C):

$$C = \frac{\Delta t\,U}{\Delta x} \leq 2 \qquad (16)$$

This restriction is mitigated by the fact that diagonally dominance will be enhance by coldplate plate coupling. Therefore, Eqn. 16 should be considered a rough criteria in selecting the time step. Regardless, the user should be aware that accuracy decreases with increasing time step, and so in general, C should be made as small as possible.

CASE STUDIES

In this section, the direct solution to Eqn. 14 is presented in both time and space for the cases of the inlet fluid temperature described by: a pulse function; a harmonic function. The pulse function will be used to demonstrate the features of spurious dispersion. The harmonic function, for which an exact solution exists, is used to determine the global error and method order. In both cases, the fluid is decoupled from the baseplate in which case we are solving the homogeneous convection equation. The space domain is divided into 100 equal sized nodes with C = 1.

Pulse Inlet Temperature

A pulse boundary condition is analogous to a sudden burst change in inlet fluid temperature for a finite length of time, then reverting back to the original pre-disturbance value. The pulse or square wave, in the absence of any dissipation, should retain its original shape and form along the characteristics. That is, the temperature disturbance should be convected at the velocity U and should retain the stepped nature of the original disturbance along lines where x - Ut is a constant. In this test, the magnitude and time duration of the pulse are both unity. Figure 2 presents a three dimensional plot of temperature versus time and space. The general trends indicate that the pulse does not dissipate forward into the flow as indicated by the sharp ridge that defines the leading characteristic. However, in regions behind or downstream of the pulse, the surface is

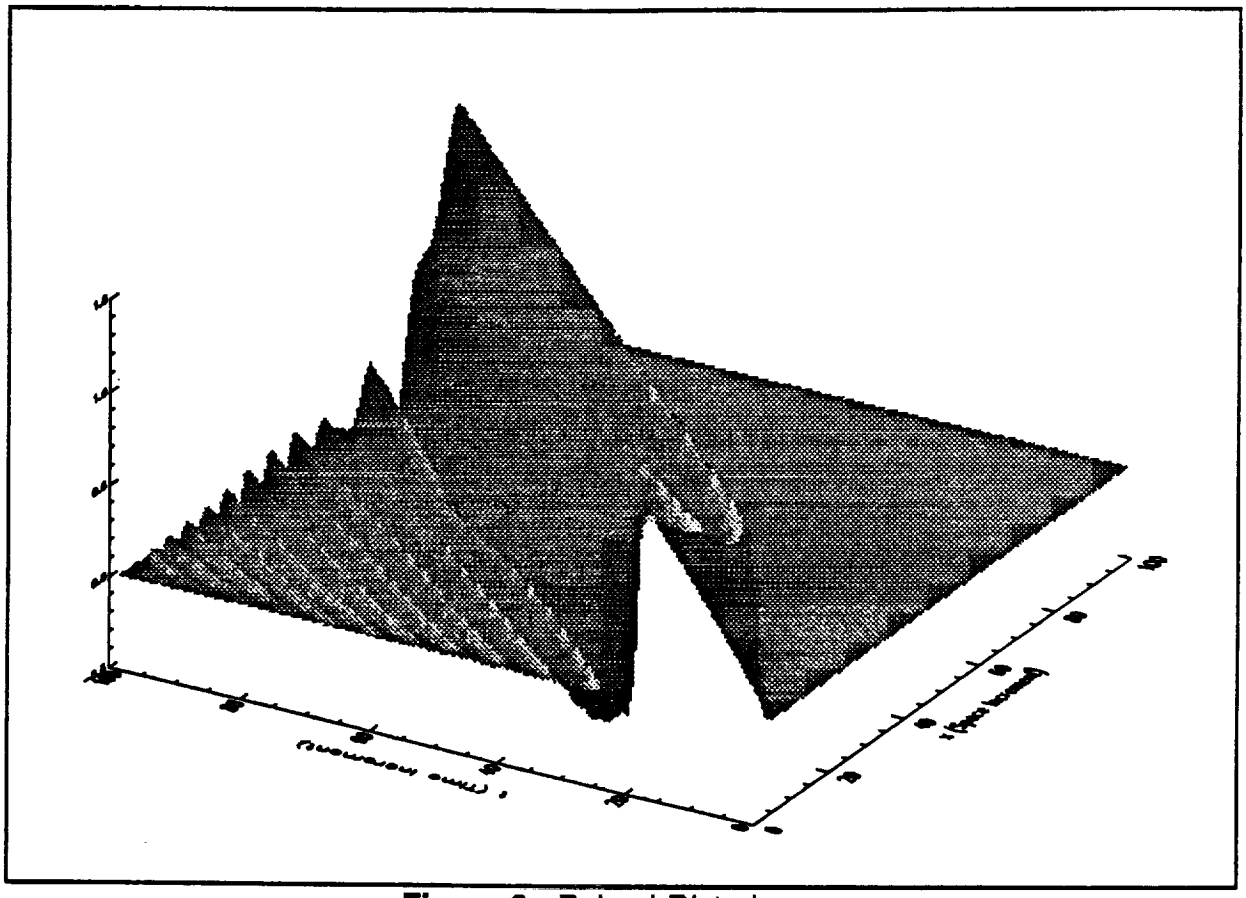

Figure 2. Pulsed Disturbance

wavy which means that there are residual effects of the disturbance that has since left the region. This is the result of dispersion, and is exacerbated by the fact that we have used a discontinuous step function for the disturbance.

Harmonic Inlet Temperature

A harmonic disturbance is useful to study from the standpoint of determining the global error and the overall method order. In this case we define the inlet temperature as $T_0 = \sin(2\,\pi\,t)$; the exact solution is given:

$$T(t, x) = \sin(2\,\pi(t - \frac{x}{U})) \qquad (17)$$

Figure 3 presents the time-space plot of temperature. We immediately see that dispersion of the waves is diminished in comparison to the pulse disturbance.

The global error (E_∞) of a method is the maximum absolute difference between the numerical and exact solutions. We can compute E_∞ for several values of Courant number to establish the dependence between discretization on accuracy. In general, the global error increases with Courant number as depicted in Figure 4. We can determine an approximate method order if we assume that the global error is a function of the time and

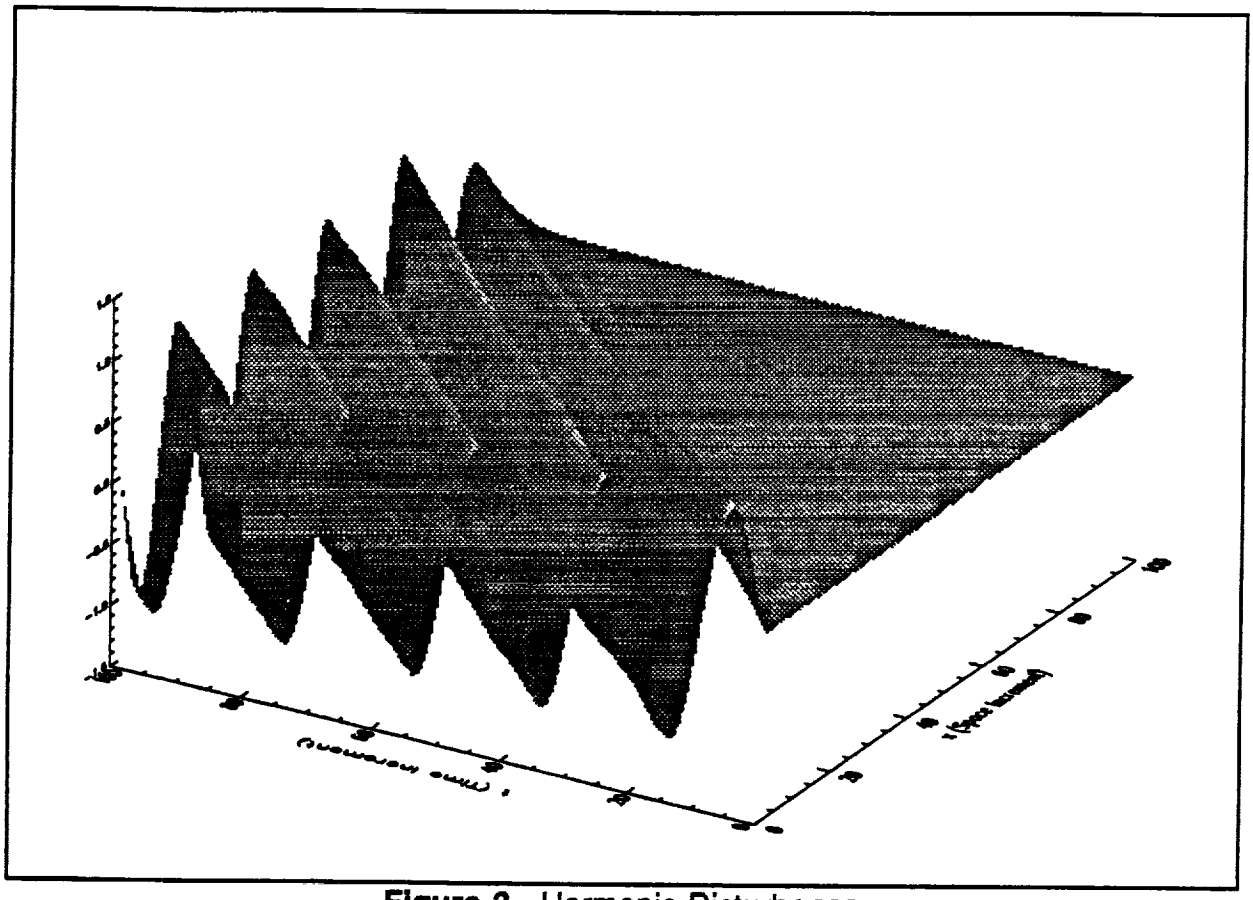

Figure 3. Harmonic Disturbance

space increments according to:

$$E_\infty = a\,\Delta t^P + b\,\Delta x^Q \tag{18}$$

For a constant space increment, we can rewrite Eqn. 19 as a function of the Courant number:

$$E_\infty = \bar{a}\,C^P + \bar{b} \tag{19}$$

Now if we have two successive values of E_∞ for corresponding values of C, and if we assume the coefficients $\bar{a}$, $\bar{b}$ do not change for the small change in C, then we can compute the method order P according to:

$$P = \frac{\log(E_{\infty 1}) - \log(E_{\infty 2})}{\log(C_1) - \log(C_2)} \tag{20}$$

Equation 20 predicts a nominal value P = 1.8. This means that the overall order of the method is slightly less than second order. This is expected because a backwards

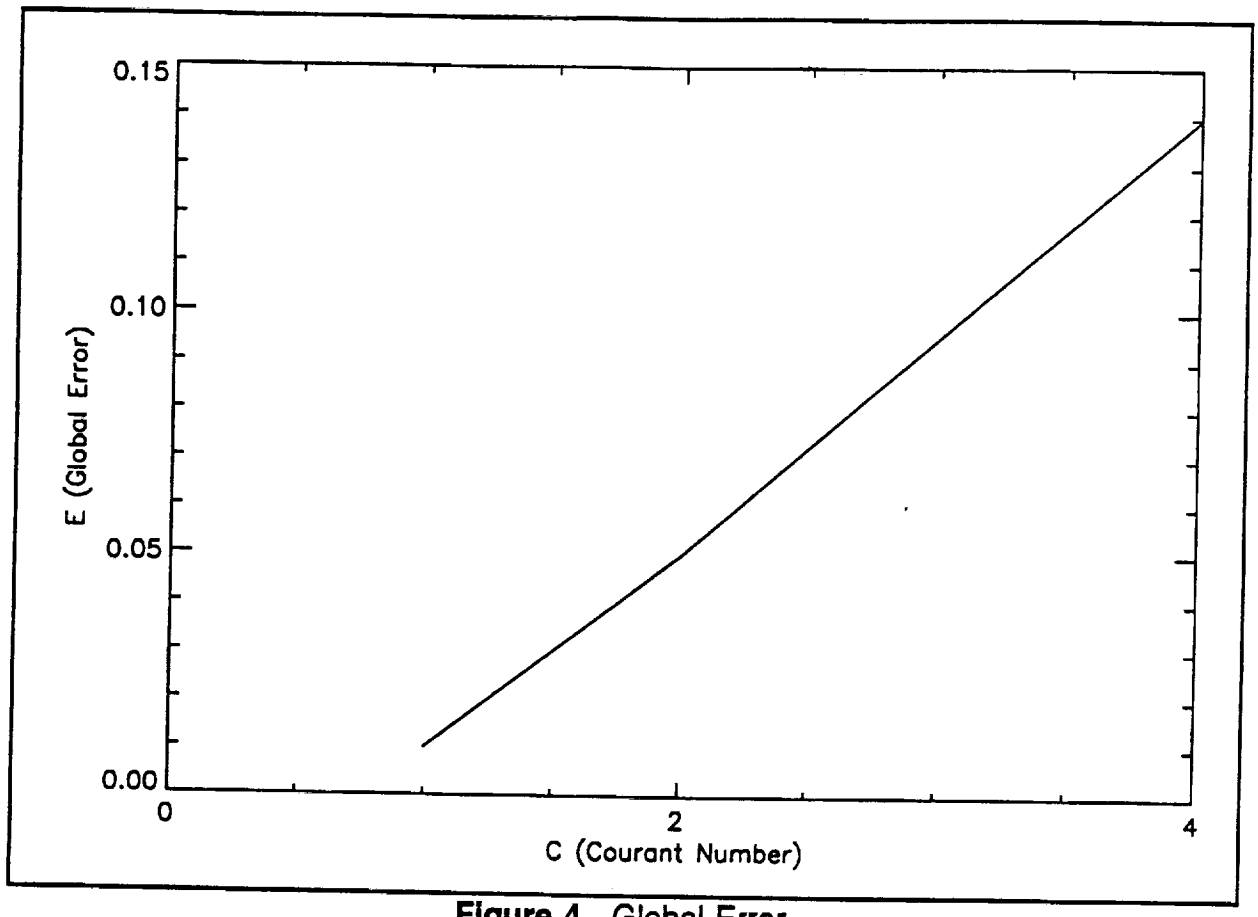

Figure 4. Global Error

difference kernel, of first order accuracy, is applied at the node N. First order inaccuracies generated at the last node tend to propagate into the solution domain, reducing the overall solution order. Overall, a specified method order should be considered approximate and will yield to the practical considerations of discretization and the application of boundary conditions.

MELDING WITH STANDARD CLASS SOLVERS

The CNCD solution algorithm with LU matrix decomposition routine was programmed in FORTRAN in a structured subroutine form. The inputs to the code are the time, time step (Courant number), appropriate geometry and thermal physical properties and boundary and initial conditions (coldplate and inlet fluid temperatures respectively). The code can be executed independently to return a temperature array describing the local time and space values of the fluid temperature. This would be an acceptable output as a utility routine if the code could be melded and referenced (called) within a standard class solver such as SINDA. Of this melding or integration, there are several important considerations that the user must be aware of:

- clashing of variable names between the convection code and the general purpose code (SINDA)

- creating common blocks required by the convection routine and defining global parameters

- conversion of absolute and relative node locations to the temperature values in array format to be referenced[10]

Without elaboration, these above problems with code melding are considered rudimentary and can be managed with careful programming. However, the most important consideration is that of time-step synchronization.

Standard solvers that analyze transient phenomenon, have internal discretization structures, similar to those of the present study, but are typically removed from the user. Removed in the sense that the local variables are not known, these discretization structures select the appropriate time step during the simulation to ensure a specified accuracy. Because the present algorithm is not an integral accessory to the standard solver, the time and time-step of the standard solver must be sampled during a melded simulation between solver and convection equation algorithm. Ideally, we desire to make a single subroutine call to the convection code to update the boundary conditions. However, as observed from the present study, the time-step used in the standard solver is typically larger than that limited by the Courant criteria (Eqn. 16). Thus we cannot make a single call to the convection routine to update the boundary conditions, rather, a sub-integration must be performed.

This sub-integration is simply a moving time window. If the standard solver is sampled to indicate a local time of 1 and time step if 0.1, and if the Courant criteria requires a time step of 0.01 (say for C = 1), then we must perform 10 sub-integrations of the convection equation to integrate from time t = 1 to t = 1.1. Upon completion of the sub-integration, the user must whether decide to update the outer integration.[11] If the outer integration step is large (the step internal to SINDA), then it may be necessary to reenter the outer-integration upon completion of the sub-integration and iterate the solution. If a relative small outer-loop time step is applied (this can be done by limiting the time step selection mechanism in SINDA) then this iteration step can be eliminated.

CONCLUSIONS

A numerical study of the solution of the one-dimensional convection equation is

---

[10] This is a SINDA convention specific to the SINDA code. The user names nodes with absolute numbers, i.e. nodes 1 to 100. While, SINDA applies a relative numbering convention internally. To reference the temperatures of those nodes that are boundary conditions, the user must convert the numbering convention with a utility routine.

[11] An update or iteration is required because the convection solution forms a coupled boundary conditions with the standard solver.

presented. A solution kernel has been presented that will yield satisfactory accuracy with minimal computation complexities. The features of stability, accuracy and spurious numerical phenomenon have been addressed in a general fashion to provide the user with a cursory insight into problematic features of numerical convection equation solutions. The key feature of time-step synchronization, as it applies to standard solver code melding, has been discussed in considerable detail.

REFERENCES

Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., Computational Fluid Mechanics and Heat Transfer, McGraw Hill, New York, 1984, pp. 88-107.

Gerald, C. F., and Wheatley, P. O., Applied Numerical Analysis, Addison Wesley, Reading, Massachusetts, 1984, pp. 139-143.

Strauss, W. A., Partial Differential Equations - an Introduction, John Wiley & Sons, Inc., New York, 1992, pp. 194-197.

Session Four

Aeronautics

PRECEDING PAGE BLANK NOT FILMED