PROCEEDINGS : CAMME-1996
©NML JAMSHEDPUR; pp.48-68

# Performance analysis of an efficient finite element solver for solidification simulation of continuous casting of steel billets

T. S. PRASANNA KUMAR

*Automation Division, Tata Iron & Steel Co. Ltd., Jamshedpur - 831 001.*

## ABSTRACT

*The Gauss-Seidel iterative technique of solving a set of linear algebraic equations was applied for solving the system matrices in the Finite Element Analysis of 2-D dynamic heat diffusion problems encountered in the solidification analysis of continuous casting of steel billets. An efficient algorithm for storing and manipulating only the non-zero terms of the system matrices was developed. The CPU time per iteration for solving the system matrix was independent of bandwidth. The oscillatory characteristics of the algorithm with respect to different one-step recurrence schemes, the number of iterations for solution convergency and error propagation with respect to overrelaxation factor and convergence limit were studied for a standard problem and compared with analytical solution. The accuracy of the iterative solution was compared with the standard method of direct reduction based on Gauss elimination (active column reduction method). The iterative technique performed better than the direct method with respect to memory requirement and CPU time, achieving acceptable accuary limits. The solver was applied for the solidification simulation of continuously cast billets at Tata Steel. The 1-D heat flux formulation of the type $q = A - B\sqrt{t}$ applicable for the C.C. mould region was modified to account for the lower heat flux at the billet corners. The midface shell thickness obtained by simulation at mould exit was compared with the measured thicknesses obtained from a breakout strand. The heat transfer coefficient in the spray cooling zone was adjusted to get an acceptable match between the measured and simulated shell thicknesses in the secondary cooling zone. The program was run on IBM PC AT computer with Intel 80486 CPU (33 MHz). The present implementation of the iterative technique for solving the system equations reduces the matrix solution time to*

*(1/18) and the overall time for each time step to 1/8 the times required under direct methods, for the parameters considered. There was no appreciable error in the estimated shell thickness within the CC mould. However, the error in the secondary cooling and the radiation zone ranged from 2 to -5%. The second norm of temperature distribution across the billet cross section varied from 0.001 to 0.005 for the entire strand.*

## INTRODUCTION

The Finite Element Method (FEM) is one of the computationally intensive numerical methods. Ideally, it requires very large memories and fast processing speeds to hold and manipulate a large amount of data. The hardware requirements of any Finite Element Analysis (FEA) has therefore been quite stringent restricting its implementation to either the mini or supermini computers. The improved processing speed and the superior graphics available on today's PC have, however, changed the scene. But the most important restriction under MS-DOS, namely a maximum RAM of 640 kB still remains. Another limitation is a maximum of 64 kB for predeclared variables. Though higher RAMs are available, they can be used only as Extended or Expanded memory. Storage and retrieval of data from the hard disk will make the program extremely slow requiring better algorithms to be developed. In the FEA of any problem, the step which takes maximum time is the solution of the system matrices; especially, in dynamic problems, where the matrices have to be solved for each time step. The memory requirement is governed by the problem size in number of degrees of freedom. The symmetry and sparseness of the system matrices are invariably made use of for minimising the memory requirement. Several techniques are available for solving the matrices which try to achieve minimisation of both memory requirement and CPU time; most of them applicable to methods based on the direct Gauss elimination technique, like handling terms within the skyline, frontal solution technique, substructuring[2] etc. The other important solution technique viz., the Gauss-Seidel technique has not been in use now, though it was initially tried by Wilson *et.al.* This latter technique, though known to converge unconditionally, has been largely discarded in FE implementations because of the uncertainty about the number of iterations it may require[3].

This work reports the implementation of an efficient method of solving the 2-D transient heat condution equation by the Gauss-Seidel

technique on a PC. The program was written in Turbo PASCAL under MS-DOS. The PC had a RAM of 640 kB and a 20 MB hard disk and a co-processor. Variables were assigned dynamic allocation through doubly linked list structure. An efficient method of storing and retrieval of matrix elements was designed. The solution time was independent of band-width. The influence of different recurrence schemes, overrelaxation factors and convergence limits on solution accuracy and iterations were studied. The solution by Gauss-Seidel method was also compared with that obtained by the Gauss elimination technique. The program was successfully applied for solidification simulation of continuously cast steel billets. The 1-D heat flux formulation applicable for the mould region was modified to account for the lower heat flux at the corners. The solidified shell thickness at mould exit predicted by the model was compared with the actual thickness measured from a strand which had suffered a shell breakout.

## GENERAL 2-D CONDUCTION HEAT TRANSFER MODEL

The transient conduction heat transfer in 2-D Cartesian space is defined by the equation

$$\frac{\partial}{\partial x}\left(k\,\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\,\frac{\partial T}{\partial y}\right)+Q=\rho c\,\frac{\partial T}{\partial t}$$

with the initial condition

$$T = T_i\ (x, y) \text{ at } t = 0$$

and the boundary conditions

$$T = T_0\ (x, y) \text{ at } t > 0 \text{ on } S_1,$$

$$-kn_x\frac{\partial T}{\partial x}-kn_y\frac{\partial T}{\partial y}=h_c\,(T-T_\infty) \text{ on } S_2 \text{ and}$$

$$-kn_x\frac{\partial T}{\partial x}-kn_y\frac{\partial T}{\partial y}=q\,(x, y) \text{ on } S_3 \qquad \ldots\ 1$$

A general program was developed[4-6] to solve eqn. (1) wherein any irregular solution domain could be defined and any of the boundary conditions could be imposed (Fig. 1). The solution domain was divided into linear triangular elements. The temperature distribution within the

element was written as[4]

$$T(e)\,(x, y, t) = \sum_{i=1}^{3} N_i\,(x, y)\,T_i\,(t) \qquad \ldots\ 2$$

where $N_i$ are the shape functions. Applying Galerkin criteria to eq. 1, we get,

$$\int_{D(e)} N_i \left[ \frac{\partial}{\partial x}\left( k \frac{\partial T^{(e)}}{\partial x} \right) + \frac{\partial}{\partial y}\left( k \frac{\partial T^{(e)}}{\partial y} \right) + Q - \rho c \left( \frac{\partial T^{(e)}}{\partial t} \right) \right] dx\, dy = 0 \qquad \ldots\ 3$$

Integration of eq. 3 term by term and application of natural boundary conditions yields, in matrix notation.

$$[K]\,\{T\} + [C]\,\{T\} + \{F\} = 0 \qquad \ldots\ 4$$

The matrices were evaluated for each element and the global matrix assembled.

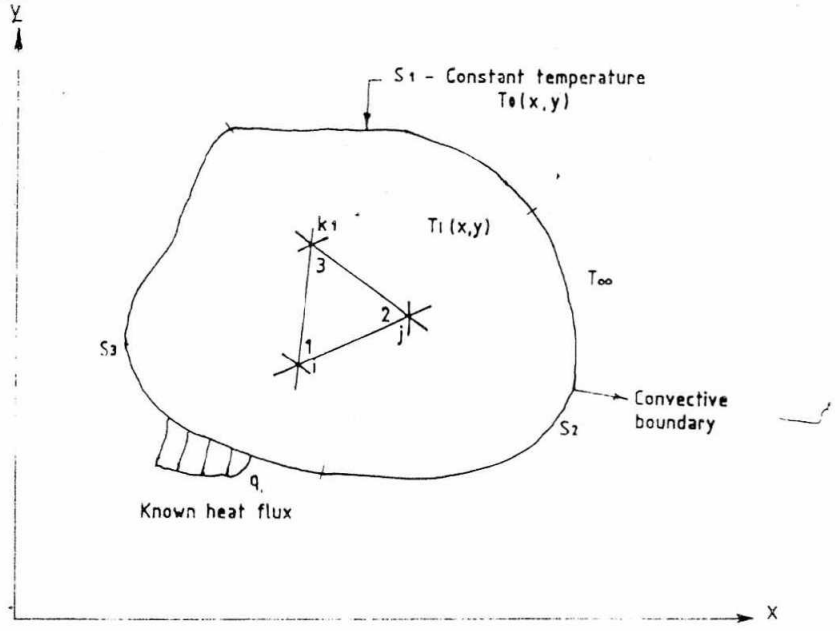


*Fig. 1 : Solution domain with the associated boundary and initial condistions*

The one-step recurrence relationship suggested by Zienkiewicz was applied for solving the first order equations developed in eq. 4. With

time as independent variable, assuming linear time elements, the variation in temperature within a time element for the entire domain was written as[7]

$$\{T\} = \sum_{n=1}^{2} N_n \{T\}_n \qquad \ldots 5$$

where $N_1 = 1 - \xi$; $N_2 = \xi$; $\xi = t/\Delta t$, and $\{T\}_n$ is a modal set of values of $\{T\}$ at a time t. By the weighted residual method, we get, for any time step n,

$$\int_0^1 W/[K]\ (N_1\{T\}_n + N_2\{T\}_{n+1}) + [C](N_1\{T\}_n + N_2\{T\}_{n+1}) + \{F\})d\xi = 0 \ldots 6$$

which upon regrouping becomes

$$\left([K]\theta + \frac{(C)}{\Delta t}\right)\{T\}_{n+1} + \left([K]\ (1-\theta) - \frac{[C]}{\Delta t}\right)\{T\}_n + \theta\{F\}_n + (1-\theta)\{F\}_{n+1} = 0 \qquad \ldots 7$$

where

$$q = \frac{\int_0^1 W\xi\, d\xi}{\int_0^1 \xi\, d\xi}$$

Zienkiewicz[7] considers the influence of W on the stability of the solution of eq. 7. Of the various forms W can take, two forms were tried here; W = 1 and W = N = $\xi$, for studying the solution characteristics obtained by the iterative technique. The first formulation with $\omega=1$, corresponding to $\theta = 0.5$ is identical with the well known mid-difference Crank-Nicholson scheme while the latter with W = $\xi$, corresponding to $\theta = 2/3$ was first suggested by Zienkiewicz, eq. 7 was thus reduced to the form

$$[A]\ \{T\}_{n+1} + \{B\} = 0 \qquad \ldots 8$$

where the forms of [A] and {B} are obvious from eq. 7.

The unknown $\{T\}_{n+1}$ was then evaluated successively for each time step using the iterative equation

$$\{T\}_{n+1}^{1+1} = \{T\}_{n+1}^{1} + \beta [A]_D^{-1} \left( \{B\} - [A]_L \{T\}_{n+1}^{1+1} - [A]_D \{T\}_{n+1}^{1} - [A]\{T\}_{n+1}^{1} \right) \quad \ldots 9$$

where $[A]_D$ is a diagonal matrix and $[A]_L$ is lower triangular matrix such that

$$[A] = [A]_L + [A]_D + [A]^T_L$$

1 indicates the iteration step and ß is the overrelaxation factor. The iteration was continued until the change in the current estimate of the temperature vector was smaller than a convergence limit ε, defined by

$$\varepsilon = \frac{(\{T\}n+1 - \{T\}n)2}{(\{T\}n+1)2} \quad \ldots 10$$

The method used for solving the system matrices given in eq. 8 through the iterative relationship in eq. 9 has the advantage that the zeroes of [A] matrix, whether outside or within the band does not contribute to the solution and hence need not be stored. In this program, only the nonzero terms of the upper half of [A] was stored as shown in Fig. 2c. Retrieval of the matrix elements corresponding to any row and column was done through an indexing system which kept track of row-column information vis-a-vis any term of [A] stored in the compressed form. Matrix assembly also was based on this indexing system. This method of handling the matrices required minimum memory and was independent of band-width.

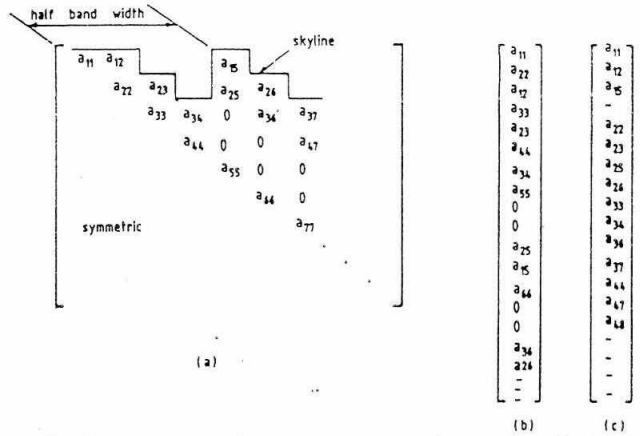


*Fig. 2 : (a) A typical system matrix and storage schemes needed for*
*(b) Gaussian elimination (standard)*
*(c) Storage scheme implemented in the present algorithm (Gauss Seidel)*

## RESULTS OF NUMERICAL EXPERIMENT

The solution procedure adopted in this work was iterative in nature involving a overrelaxation factor and a convergence limit which would influence solution accuracy with time. A comprehensive study of the behaviour of the solution algorithm was therefore carried out under different conditions. A simple problem for which an analytical solution existed, viz., transient heating of a steel slab was considered for comparing solution accuracies[8]. A rectangular domain of 0.075m x 0.05m was discretized into 64 linear triangular elements and 45 nodes (Fig.3). eq. 1 was solved over this domain for an initial condition of $\{T_i(x, y)\} = 35$ at $t = 0$ and the boundary conditions $T_o(0, y) = 250$ at $t>0$. The other three boundaries were considered adiabatic making the heat transfer essentially one-dimensional;. The internal heat generation term Q in Eqn. 1 was set to zero.

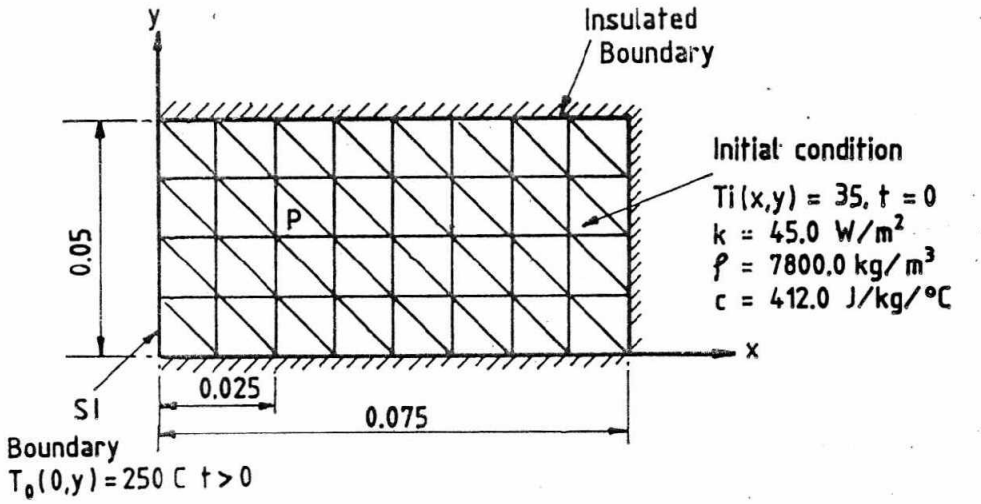


*Fig. 3 : Finite element model of heating of a steel slab*

The test problem as defined above could be treated as one-dimensional transient heating of a semi infinite slab for which the analytical solution is given by

$$\frac{T(x, t) - T_O}{T_i - T_O} = \text{erf}\frac{x}{2\sqrt{\alpha t}} \qquad \ldots 11$$

The problem was solved for a representative set of variables which would affect the solution accuracy and time. The analysis was carried out under the following heads.

**Influence of Different Weight Functions on the Oscillatory Characteristics of the Solution**

The analytical and FEM solutions for the temperature at P(0.025, 0.025) during the first 30 seconds of heating are given in Fig. 4 which shows the oscillatory characteristics of the solution due to the different weight functions, θ. Of the many values θ can take, only two were considered here viz., θ = 1/2 and θ = 2/3. The first value of θ = 1/2 is identical to the mid-difference Crank-Nicholson scheme which is used extensively in Finite Difference formulations while θ = 2/3 was first suggested by Zienkiewicz[7] and is considered here because of its superior properties. Referring to Fig. 4, the Galerkin process with θ = 2/3 gave better results both in terms of oscillations and accuracy. It was observed that with higher time steps of Δt=5s, the initial oscillation was high but died down rapidly approaching the same accuracy levels at smaller time steps of Δt = 2s.

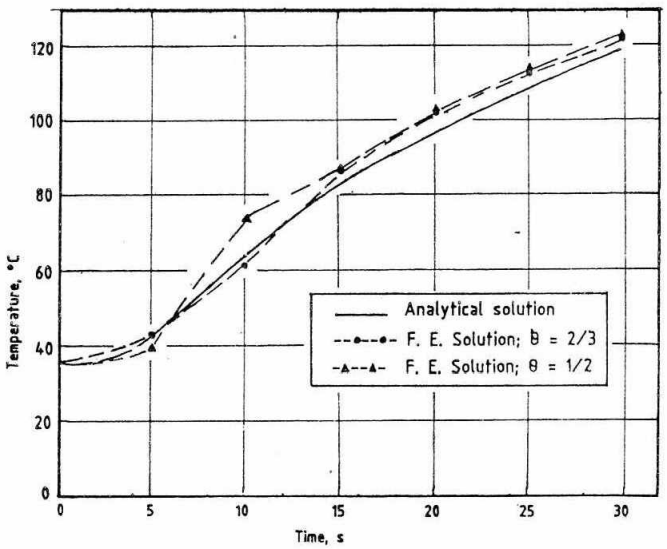


*Fig. 4 :Comparison of analytical and finite element solutions for different time integration schemes.*
*β = 1.5; ε = 0.01; Δt = 5s*

**Influence of Mesh Size of Accuracy**

From an analysis of the influence of weight functions on the oscillatory characteristics of the solution, it was clear that the Galerkin process of time integration with θ = 2/3 gave better results compared with the C-N scheme. The influence of mesh size, convergence limit and the overrelaxation factor on the solution accuracies was therefore

confined only to a weight function of 2/3. The time step was taken as 2s.

Improvement in solution accuracy with mesh size is shown in Fig. 5. The test problem was solved for two grades of mesh at 45 nodes and 288 nodes. Fig. 5 shows the temperature history at the same sample point P for these two finite element models as compared with the analytical solution. With 288 nodes, the finite element solution was almost identical with the analytical solution for the given problem.

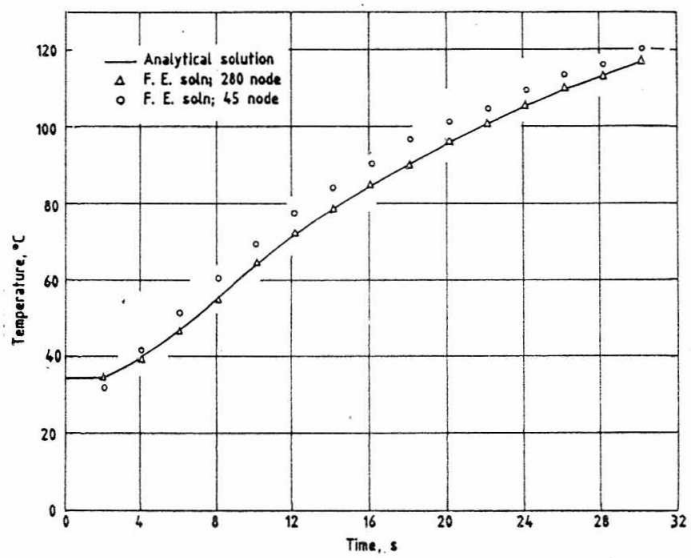


*Fig. 5 :Comparison of finite element solutions with the analytical solutions*

**Number of Iterations Per Time Step**

An important feature of the algorithm adopted in this work for the solution of the system matrices was the user controlled facility of specifying both the convergence limit ε and the relaxation fact ß depending upon the nature of the problem and to optimise the solution accuracy and CPU time. Both ε and β affect accuracy and number of iterations for convergence. In dynamic problems therefore, ε and ß must be chosen judicially to keep the number of iterations per time step within reasonable limits at the same time achieving acceptable accuracies. The behaviour of the algorithm for a range of values of ε and β was therefore studied with reference to the test problem already described. For purposes of comparing the accuracies obtainable with different ε and β, and to establish their influence on number of iterations per step, (eqn. 8) was also solved by the direct $LDL^T$ decomposition method (Fig. 6). Bathe and Wilson[2] have dealt with this algorithm in detail and their algorithm was adopted here also. The $LDL^T$

decomposition is based on the direct Gauss elimination scheme which requires all the terms within the skyline of the system matrix to be stored. Hence, it is more memory intensive than the present iterative technique, but no iteration is needed within the time step. This scheme is generally preferred to the iterative technique because the uncertainties about number of iterations required for convergence are removed. However, in initial value problems such as the one considered here, the starting vector for solution is the initial condition itself and hence the iterative method has some advantages. In this study, therefore, the solution accuracy obtained with the iterative technique was compared with that obtained by the direct method. The number of iterations required for convergence was also studied.

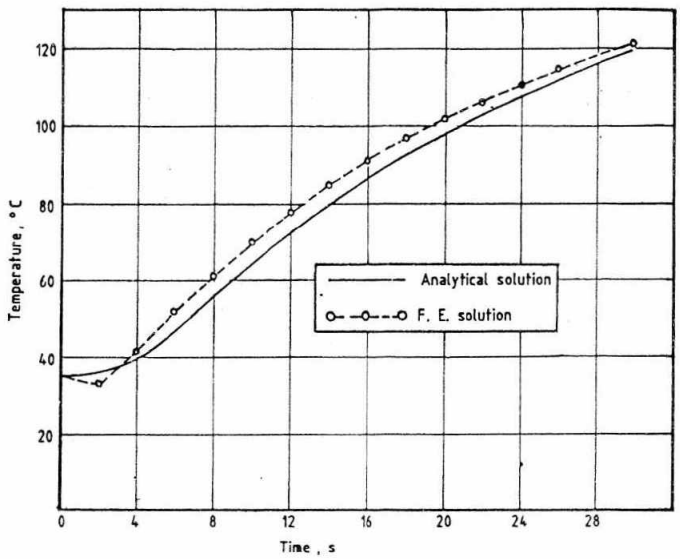


*Fig. 6 : Comparison of analytical and finite element solution by Gauss Elimination Scheme (Standard - $LDL^T$ decomposition; $\Delta t$ = 2 sec.*

**Influence of Convergence Limit (ε) on Number of Iterations and Solution Accuracy**

Fig. 7 shows the propagation of error in solution at point P for different convergence limits ε, ranging from 0.1 to 0.001. The relaxation factor ß was set at 1.5 and time step $\Delta t$ = 2s. It is clear from the figure that with ε = 0.001, the solution approaches that obtained by the direct technique. The error in solution was computed from the formula

$$\% \text{ Error} = \frac{T_{dir} - T_{iter}}{T_{dir}} \times 100 \qquad \ldots 12$$

where $T_{dir}$ is the temperature at P(x, y) obtained by the direct Gauss elimination technique and $T_{iter}$ is the temperature obtained by the iterative technique. The value in the parenthesis in Fig. 7 is the number of iterations required for convergence at that time step. It is observed that for the problem under consideration, number of iterations is high at the beginning and after about 6-8 time steps, it drops to about half of the initial value. Also, with smaller ε, the number of iterations would be large, but at the same time, higher accuracies are obtained. For example, with ε = 0.01, accuracies within about 9% error could be obtained with just 1 iteration per time step while the error could be reduced to 0.045% with an ε = 0.001, which would need 5 iterations per time step.

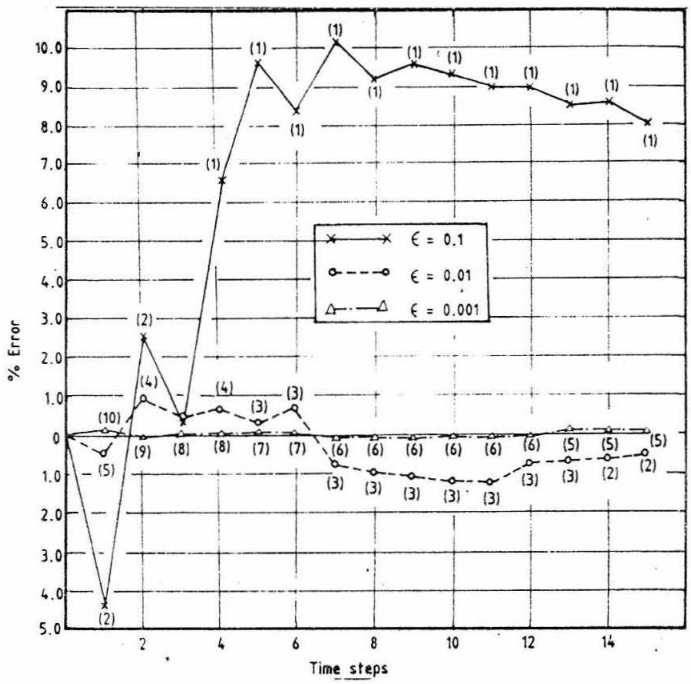


*Fig. 7 : Improvement in solution accuracy with convergence limit (ε) for the Present iterative technique (Compared with the Standard Direct LDL$^T$ decomposition Method). Value in paranthesis indicates No. of= iterations, β = 1.5; θ = 2/3; Δt = 2 sec.*

**Influence of Relaxation Factor on Number of Iterations and Solution Accuracy**

The influence of relaxation factor ß for a range of values from β= 1 to β = 1.8 (β = 1 corresponds to no relaxation) on the number of iterations required for convergence and its effect on solution accuracy

is shown in Fig. 8. The convergence limit ε was set at 0.001 for all the cases. It was observed that for the problem under consideration, the number of iterations required for convergence increased with higher ß, with an accompanying improvement in solution accuracy. But the increase in number of iterations was not quite proportionate with the decrease in error. When β was changed from 1 to 1.8, the error decreased from 0.079% to 0.037%, but the number of iterations increased from 3 to 15.

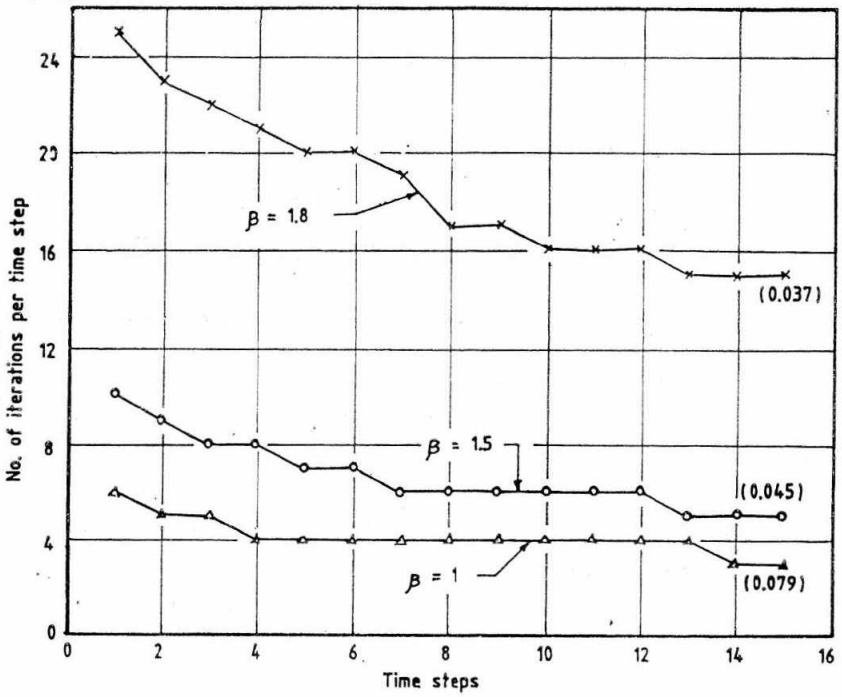


*Fig. 8 : Influence of relaxation factor on No. of iterations per time step and solution accuracy (Compared with Direct LDL$^T$ decomposition method). Value in paranthesis indicates percentage error at the end of 15th time step. ε = 0.001; θ = 2/3; Δt = 2s*

## CPU Time and Memory Requirement of the Algorithm

The terms in the system matrix [A] of eq. 8 and hence the CPU time for solution in this program are independent of band width. To bring out this feature, the test problem was solved for two different node numbering schemes thus altering the band width. The domain was discretized into 400 linear triangular elements and 231 nodes. The finite element meshes and the node numbering schemes are shown in Fig. 9 along with the boundary conditions. The essential features of the two different finite element models are given in Table 1. The two

finite element models were solved by both the direct Gauss elimination scheme and the iterative technique.

*Table 1 : Number of iterations required for each time step for different convergence limits; dt = 0.5 sec.*

| Time step number | Convergence limit(ε) $10^{-2}$ | $10^{-4}$ | $10^{-8}$ |
|---|---|---|---|
| 1 | 1 | 22 | 89 |
| 2 | 1 | 9 | 29 |
| 3 | 2 | 9 | 29 |
| 4 | 2 | 8 | 27 |
| 5 | 1 | 8 | 27 |
| 10 | 1 | 7 | 27 |
| 20 | 1 | 7 | 27 |
| 50 | 1 | 6 | 24 |
| 100 | 1 | 5 | 24 |
| 200 | 1 | 4 | 24 |
| 400 | 1 | 4 | 24 |

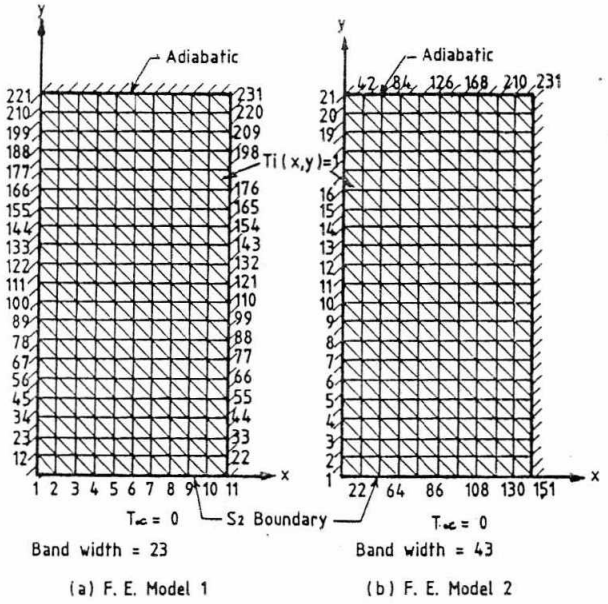


*Fig. 9 : Finite element model of test problem*
*(a) Node numbering scheme giving a band width of 23*
*(b) Node number scheme giving a band width of 43, k = 1, ρ = 1, c = 1, h =1*

In the direct $LDL^T$ decomposition method, even the zero terms within the skyline of the system matrix will have to be handled (Fig. 2b). The number of terms in the system matrix for F.E. Model 2, with a band width of 23 was 2661 while for F.E. Model 2, with a band width of 43, the number of terms was 4661. In the present implementation of the iterative technique, however, the zeroes even within the band width have been eliminated (Fig. 2c) and for both the F.E. Models considered, only 924 terms need be handled. (Fig. 10). The influence of the number of terms in the matrix on the CPU time for different solution steps is shown in Table 2. Except for matrix assembly, the other steps involved in the solution procedure viz., application of boundary condition (step b), time integration (step c) and solution of matrices (step d) took less time under the present iterative algorithm compared to the direct method. The times remained constant for the iterative technique even with increase in the band width; while it increased by times for the higher band width model. Both the models were solved entirely in the core memory of the PC without making use of the hard disk.

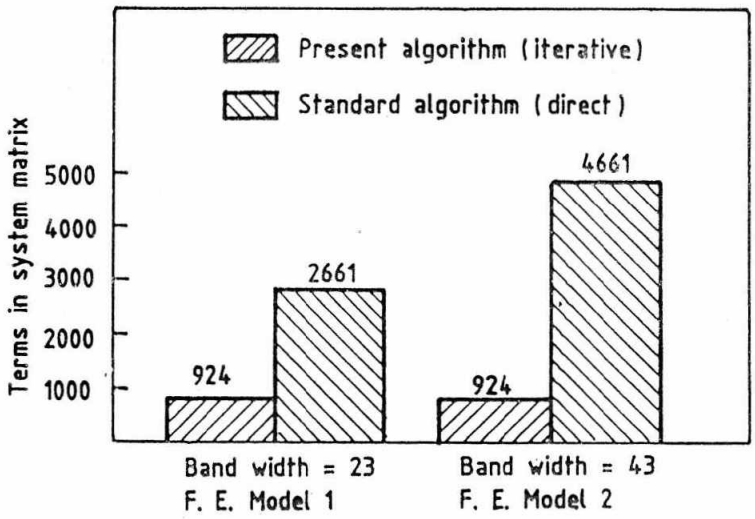


*Fig. 10 : Bar Chart Showing the System Matrix Terms in the two algorithms for different band widths but the same number of Nodes (Problems 2)*

***Table 2** : CPU times (sec) for solution steps*

| Solution step | Direct method | Iterative method |
|---|---|---|
| i) Matrix assembly | 0.33 | 0.27 |
| ii) Time integration | 0.17 | 0.17 |
| iii) Matrix Solution | 5.21 | 0.29* |

*for 5 iterations

A direct solution to Eqn. 4 was also obtained by the Gauss elimination technique using the $LDL^T$ decomposition method. Bathe and Wilson[24] have dealt with this algorithm in detail. The direct method requires all the terms within the skyline of the system matrix to be stored. Hence, it is more memory and CPU intensive than the iterative technique, but no iteration is needed within the time step. The solution accuracies obtained by the direct and iterative techniques were studied vis-a-vis the CPU time.

## APPLICATION TO CONTINUOUS CASTING OF BILLETS

The model was applied for the solidification simulation of a 125mm square section billet at Tata Steel. The temperature dependent material properties, boundary conditions and the source term used in the formulation of this problem make the system of matrices in eq. 2, non linear. A simple extrapolation was used here, taking the value of the matrices previously calculated for the next step also; making

$$[K]_{n+1} = [K]_n,\ [C]_{n+1} = [C]_n \text{ and } \{F\}_{n+1} = \{F\}_n \qquad \ldots 13$$

The following equations were used for estimating the liquidus and solidus temperatures[5]:

$$T_{liq} = 1537\text{-}88C\%\text{-}25S\%\text{-}8Si\%\text{-}5Mn\%\text{-}30P\%$$

$$T_{sol} = 1537\text{-}200C\%\text{-}12.3Si\%\text{-}124.5P\%\text{-}183.9S\%\text{-}6.8Mn\% \qquad \ldots 14$$

The specific heat and the thermal conductivity were defined by the following functions :

$$\begin{aligned}
c &= 787 \text{ J/kg K} && T >= T_{liq} \\
c &= 268 + (0.334)T \text{ J/kg K} && 1137 < T < T_{sol} \\
c &= 648 \text{ J/kg K} && 850 < T < 1137 \\
c &= 3849 - (3.716)T \text{ J/kg K} && T <= 850 \\
k &= 43 \text{ W/m K} && T >= T_{liq} \\
k &= 18.28 + (0.0039)\ T \text{ W/m K} && 800 < T < T_{sol} \\
k &= 59.4 - (0.0418)T \text{ W/m K} && T <= 800 \qquad \ldots 15
\end{aligned}$$

The values of c and k over the solidification range were obtained by

linear interpolation. The thermal conductivity in the superheat zone was increased by a factor of 7 to take into account, the effect of fluid flow. The latent heat of solidification was taken as 272 kJ/kg released uniformly over the solidification range. Density was taken as constant at 7400 kg/m$^3$.

The heat flux in the mould region was modelled as :

$$q = 2.68 - 0.226 \sqrt{t'} \text{ Mw/m}^2$$

where $t' = 0$ for $T_{surf} > = T_{sol}$

$t' = t - t_0$ for $T_{surf} < = T_{sol}$ ... 16

$t_0$ = local solidification time

to take care of the early solidification of the billet at the corners. The secondary cooling in the castor under consideration was provided with two zones with the first of length 3m under a pressure of 4-6 bar and the second of length 5m under a pressure of 1.6 - 2 bar. The h.t.c. in these two zones were therefore assumed to be in the same ratio as the square root of the nozzle pressure and the values were so chosen as to give satisfactory match between the simulated and measured shell thicknesses.

## RESULTS AND DISCUSSION

The solution domain (1/4 of a 125mm square billet) was discretized into 1152 linear triangular elements with 625 nodes. The material chosen for simulation was C1008 grade steel with a nominal composition of 0.08%C, 0.1%Si, 0.45%Mn, 0.04%S and 0.04%P. The solidified shell thickness predicted by the model was later compared with the thickness measured from a breakout strand upto a strand length of about 4-5m. The analysis was done using both the direct and iterative solution schemes. The influence of the covergence limit on the solution accuracy obtained with the iterative technique was first studied. Two parameters were considered important for comparing the accuracies viz. shell thickness and billet midface temperatures. Number of iterations required for convergence and the CPU time were also studied. Finally, the results of both direct and iterative calculations were compared for the entire strand length in order to quantify the computational errors that might be expected and the CPU time saved by adopting a fast algorithm implemented in this work.

**Solution Accuracy and CPU Time**

The primary advantage of using an iterative algorithm implemented in this work is in the speed of computation. However, the boundary conditions become nonlinear in the secondary cooling zone and the radiation zone, with the heat transfer becoming a function of surface temperatures. Hence, the solution accuracy with reference to shell thickness and temperature distribution across the billet cross section were studied for the secondary cooling zone. An analysis of the order of errors arising due to the iterative algorithm (with $\varepsilon = 0.0001$) indicated that while there was no significant error in the prediction of shell thickness within the CC mould, the error ranged from 2 to -5% in the secondary cooling zone (Fig. 11). The error in the temperature distribution across the billet cross section, calculated from Eqn. similar to 10, ranged from 0.002 to 0.005 (Fig. 12). The detailed analysis is given elsewhere[8].

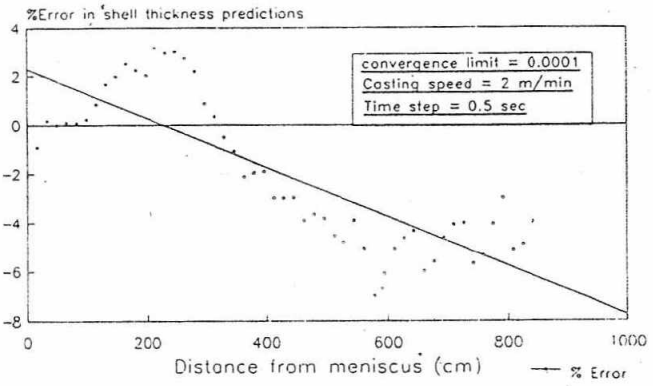


*Fig. 11 : Error trend in shell thickness predictions due to iterative solution procedur*

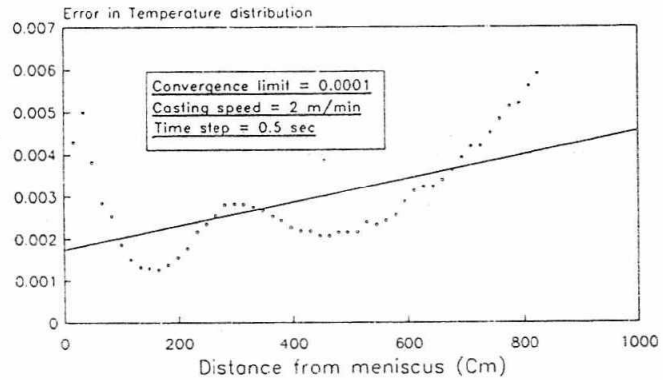


*Fig. 12 : Error trend in the temperature distribution across the billet cross section due to the iterative solution*

The program was run on an IBM PC AT computer with Intel 80486 running at 33 MHz. The computer had 64 kB cache memory, 640 kB based memory and 3 MB extended memory. Identical system resources were used for both the direct and iterative solution methods and the matrices were handled entirely in the main memory. The number of iterations required for each time step for the two values of $\varepsilon$ are given in Table 1. It was observed that the number of iterations dropped rapidly which was a characteristic of this problem. The CPU times for both the direct and iterative methods for the three important stages in the solution procedure are tabulated in Table 2. The present implementation of the Gauss Seidel iterative technique reduces the matrix solution times to 1/18 and the overall time for each time step to 1/8 the times required under direct methods.

## CONCLUSIONS

A memory efficient and band width independent iteration solution scheme for solving the system matrices in the FEA of general 2-D heat conduction problems was implemented on a PC. This iterative technique showed excellent properties summarised below. The model was later applied to the solidification simulation of steel billets at Tata Steel with considerable advantage with respect to CPU time.

1. The Galerkin process in time integration with $\theta = 2/3$ showed better properties than the C-N algorithm with $\theta = 1/2$ for the problem considered. The oscillations were less for $\theta = 2/3$ than for $\theta = 1/2$. The solution accuracy also was better for $\theta = 2/3$ than for $\theta = 1/2$.

2. The user defined convergence limit $\varepsilon$ had a very significant effect in obtaining acceptable accuracies within reasonable times. While smaller $\varepsilon$ always improved accuracies, it also increased the number of iterations required for convergence within the time step. However, the number of iterations per time step dropped to half its starting value at later times for all $\varepsilon$ considered. Solutions very near to the direct method using the $LDL^T$ decomposition algorithm could be obtained for the test problem for an $\varepsilon = 0.001$ which need 5 iterations per time step. For an $\varepsilon = 0.1$, the solution converged for a single iteration but the error was about 8.0%.

3. The relaxation factor ß which also was defined by the user had a similar influence on the number of iterations and accuracy as the convergence limit $\varepsilon$. But the improvement in accuracy was not commensurate with the increase in number of iterations required for

convergence. When the ß factor was set at 1.8, it required 15 iterations resulting in an error of 0.037% while for ß = 1.5, the solution converged in 5 iterations giving an error of 0.045%, a marginal increase in error but a three time savings in computer time.

4. The terms handled in the system matrices was independent of band width and hence, the CPU time. Though the solution algorithm requires iteration till convergence occurs, the time per iteration being a small fraction of the time required for solution of the same matrix by direct method, the total time per time step will be much less compared to the latter. Further, problems with upto 1000 degrees of freedom could be handled within the RAM under this algorithm which would otherwise require data handling through hard disk making the CPU time many folds more for problems of that size.

5. Though the number of iterations was initially high, it rapidly dropped off with time steps which was an advantage with the algorithm.

6. The CPU time for solving the system equations in the solidification simulation of steel billets was reduced to 1/18 and the overall time for each time step to 1/8 the time required under direct methods, for the parameters considered. There was no appreciable error in the estimated shell thickness within the CC mould. However, the error in the secondary cooling zone ranged from 2 to -5%. The second norm of temperature distribution across the billet cross section varied form 0.001 to 0.005.

## ACKNOWLEDGMENT

The author wishes to thank the management of Tata Iron and Steel Co., Ltd., Jamshedpur, for granting permission to publish this paper.

## NOMENCLATURE

a,b,c : Shape function parameters

c : Specific heat, J/kg/K

e : Element

$h_c$ : Convective heat transfer coefficient, $W/m^2/K$

$h_r$ : Radiative heat transfer coefficient, $W/m^2/K^4$

k : Thermal conductivity, W/m/K

l : Iteration step

$n_x$, $n_y$ : Unit normals in the x and y directions

| | | |
|---|---|---|
| $q$ | : | Boundary heat flux, $W/m^2$ |
| $t$ | : | Time, sec |
| $u$ | : | Casting speed, m/sec |
| $x, y$ | : | Cartesion coordinates |
| $z_1$ | : | Length of primary cooling zone, m |
| $z_2$ | : | Length of secondary cooling zone, m |
| $z_3$ | : | Length of radiation zone, m |
| $[A]$ | : | System matrix |
| $[A]_L$ | : | Lower triangular matrix |
| $[A]^T_L$ | : | Transpose of lower triangular matrix |
| $[A]_D$ | : | Diagonal matrix |
| $\{B\}$ | : | Force vector |
| $[C]$ | : | Capacitance matrix |
| $\{F\}$ | : | Force vector |
| $[K]$ | : | Stiffness matrix |
| $N_{i,j,k,1,2}$ | : | Shape functions |
| $Q$ | : | Heat source/sink, $W/m^3$ |
| $S_{1,2,3}$ | : | Domain boundaries |
| $T$ | : | Temperature, K |
| $T_i$ | : | Initial temperature, K |
| $T_o$ | : | Boundary temperature, K |
| $T_\infty$ | : | Ambient temperature, K |
| $\{T\}_n$ | : | Temperature vector at nth time step |
| $W$ | : | Weight function |
| $\alpha$ | : | Thermal diffusivity, $m^2/sec$ |
| $ß$ | : | Relaxation factor |
| $\varepsilon$ | : | Convergence limit |
| $\theta$ | : | Parameter |
| $\rho$ | : | Density, $kg/m^3$ |
| $\xi$ | : | Local coordinate |
| $\Delta$ | : | Element area, $m^2$ |
| $\Delta t$ | : | Time step, sec |

## REFERENCES

[1] Martin Heller, "The PC/Workstation Convergence: An explosion of opportunity", *Metal Progress,* 1987.

[2] K.J. Bathe and E. Wilson, "Numerical Methods in Finite Element Analysis", *Prentice Hall India,* 1987.

[3] E. Wilson, Private Communications.

[4] T.S. Prasanna Kumar, "Assessment of Soundness in Aluminium Alloy Castings during Solidification: Through Process Modelling and Acoustic Emission Monitoring", *Ph.D. Thesis,* IIT Madras, 1986.

[5] T.S. Prasanna Kumar, O. Prabhakar, "Finite Element Formulations for Estimating Feeding Efficiency Factors", *Trans. A.F.S.,* v.79, 1985, pp789-800.

[6] T.S. Prasanna Kumar, O. Prabhakar, "Interactive Computer Graphics Pre-Processing and Two-Dimensional Finite Element Solidification Modelling of Castings", *Trans. A.F.S.,* V.80, 1986, PP715-718

[7] O.C. Zienkiewics, "The Finite Element Method, *Tata McGraw, Hill.*

[8] T.S. Prasanna Kumar, "Application of an Efficient Finite Element Solver for Solidification Simulation of Continuous Casting of Steel Billets", Accepted for Publication in *Tata Search,* 1996.