OPTIMIZATION OF MOLD TAPER DESIGN USING MATHEMATICAL MODELS OF CONTINUOUS STEEL SLAB CASTING

BY

WILLIAM ROBERT STORKMAN

B.S., University of Illinois, 1986

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1990

Urbana, Illinois

MAY 1990

WILLIAM ROBERT STORKMAN

OPTIMIZATION OF MOLD TAPER DESIGN USING

MATHEMATICAL MODELS OF CONTINUOUS STEEL SLAB CASTING

MASTER OF SCIENCE

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

THE GRADUATE COLLEGE

MAY 1990

WE HEREBY RECOMMEND THAT THE THESIS BY

WILLIAM ROBERT STORKMAN

ENTITLED OPTIMIZATION OF MOLD TAPER DESIGN USING MATHEMATICAL MODELS OF CONTINUOUS STEEL SLAB CASTING

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

Director of Thesis Research

Head of Department

Committee on Final Examination†

Chairperson

† Required for doctor's degree but not for master's.

OPTIMIZATION OF MOLD TAPER DESIGN USING MATHEMATICAL MODELS OF CONTINUOUS STEEL SLAB CASTING

William Robert Storkman, M.S.
Department of Mechanical and Industrial Engineering
University of Illinois at Urbana-Champaign, 1989
Dr. Brian G. Thomas, Advisor

The behavior of the solidifying shell in the mold is very important to the ultimate quality of continuously cast steel. To further understand this behavior, a two-dimensional, stepwise-coupled, transient finite element model has been developed to simulate heat flow, solidification, shrinkage, mold contact, and stress generation in the shell forming within the continuous slab casting mold. The model is applied to explore the influence of varying mold taper on heat flow, shell growth, and the formation of off-corner surface depressions. The calculations predict that these depressions potentially may form whenever a linear taper is employed on the narrow face. Generally, too little taper in the upper portion of the mold fails to follow the rapidly solidifying and shrinking shell allowing the narrow face shell to bulge and rotate around the corner. Lower in the mold, high tapers force the wide face shell into compression enlarging the depression through a buckling mechanism. Below the mold, lack of support between rollers augments the rotation and bulging of thin shells. The model has been applied to design a three-fold narrow face taper that more closely follows the natural shrinkage of the shell. The small gap between the mold and the shell produced by the proper taper promotes more consistent heat transfer to the mold, growing a more uniform, thicker, bulge resistant shell, while preventing the mold from applying large compressive loads through the corner to buckle the wide face of the shell. When combined with adequate narrow face support and sufficient spray cooling below the mold, the optimal design should reduce or even eliminate the corner depression problem.

ACKNOWLEDGEMENTS

I would like to express thanks to my thesis advisor, Dr. Brian G. Thomas, for his support during my graduate studies and for his patience and encouragement during the period after my studies were completed. I also appreciated the friendship and assistance provided by Brian's other graduate students, especially Glen Haegele, upon whose research much of this project depends.

Acknowledgements are extended to the staff of the National Center for Supercomputer Applications for the support of the CRAY-XMP/48 on which some of these mathematical simulations were performed.

Funding for this project was provided by ARMCO Advanced Materials Corporation. This funding and the opportunity to apply fundamental finite element techniques to the continuous casting process was greatly appreciated and enlightening. I would also like to thank my employer, Cummins Engine Company, and my supervisors who did not demand completion of this work, allowing a more complete study of this casting process. The experience I've gained into the finite element method both from the extra time spent on this project and from my supervisors in my work has been invaluable.

I would also like to thank my wife and family whose love and support made completion of this work possible and rewarding.

TABLE OF CONTENTS

1.0 INTRODUCTION

The continuous steel casting process produces higher quality castings with higher productivity than previously used ingot casting. To take advantage of these attributes and remain competitive, the steel industry has been replacing existing ingot casting equipment with continuous casting machines for the past few decades. Optimization of the continuous slab casting process could further increase its profitability by eliminating waste, both in low quality castings and in extra processing requirements, and increasing production flows. This study furthers the understanding of the solidification process in the mold allowing better control of the casting environment and reducing defective castings.

One example where mold solidification understanding and optimization can be effective is stainless steel slab casting. Continuously cast austenitic stainless steel slabs are plagued with defects affecting the corners of the slab. These defects manifest themselves as 2 to 7 millimeter (0.08 to 0.3 inch) deep, longitudinal surface depressions or "gutters" that form roughly 75 to 150 millimeters (3 to 6 inches) from the corner on the wide face as portrayed in Figure 1. These gutters can be found in all four off-corner regions on the wide faces of the slab and are accompanied by a slight bulging of the narrow face. Experience on the caster is limited but tends to suggest that the gutters are worse for smaller narrow face tapers [1]. These gutters must be ground away completely to remove any oxides which could initiate cracking in later processing, making them extremely costly both in wasted steel and in extra process time. Plain carbon steel castings crack below the surface in this off-corner region presumably under the same mechanisms as the off-corner gutter formation in stainless steel slabs. The continuous casting process and the interaction between the thermal and mechanical loads forming these off-corner defects are discussed below.

The solidification process occurring within a continuous casting machine, shown in Figure 2, is initiated in the mold and completed in the spray zone. The process begins in the ladle, where the steel is melted and treated. The ladle empties this molten steel into the tundish from where it flows through the nozzle's regulating valves into the mold. The molten steel contacts the water-cooled walls of the mold and solidification begins, forming a thin shell of solid metal. The position in the mold where this initial solidification occurs is referred to as the meniscus.

The shell forms at the meniscus and grows as heat continues to convect from the molten steel, through the solidifying shell, and out to the mold wall cooling. The shell transforms

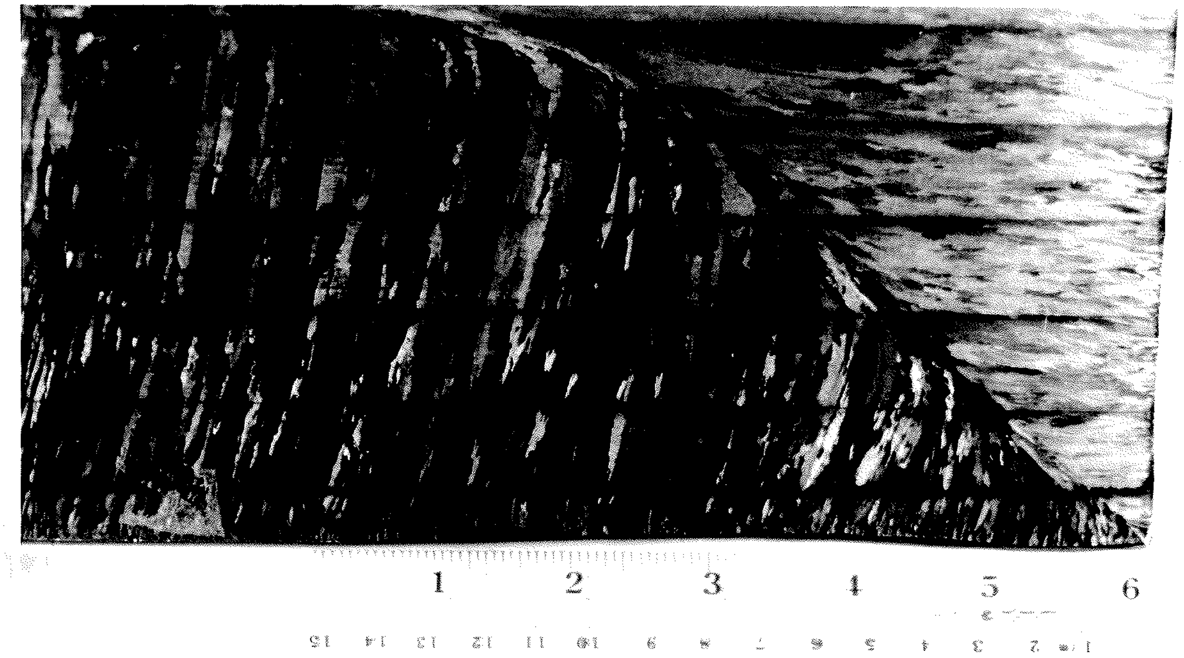


Figure 1 Photograph of a section of the AISI 304 stainless steel slab exhibiting gutter problems.

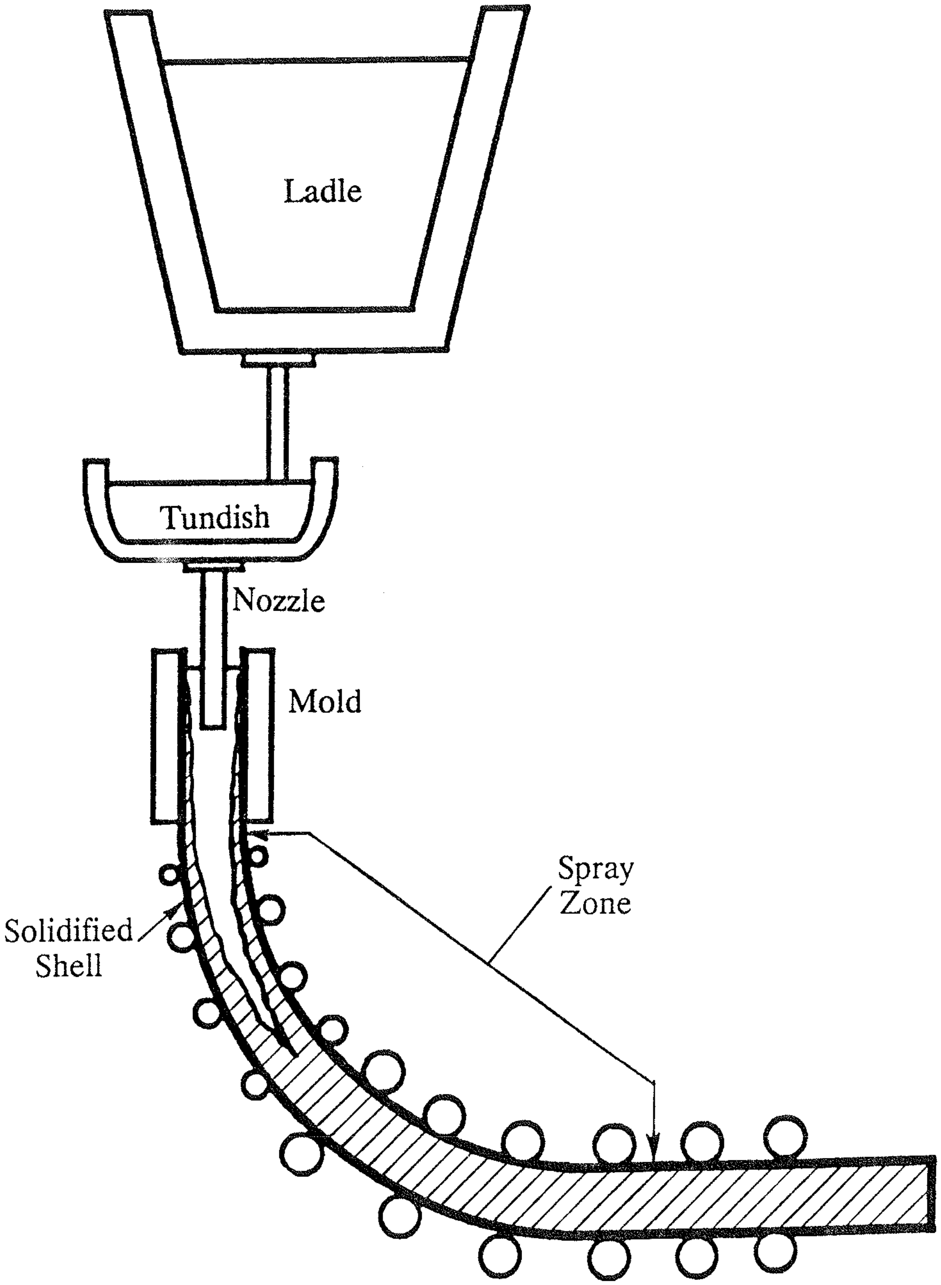


Figure 2 Illustration of a conventional continuous casting machine.

from liquid to solid and cools continuously causing the shell to shrink due to the phase transformation and thermal contraction. This shrinkage forms a gap between the shell and the mold, insulating the growing shell. The gap, fluid flow, and forming shell which control the heat transfer from the steel are shown in Figure 3. Planar mold walls or end plates, which are tapered to offset the shrinkage and minimize this gap, are used to maintain high heat transfer from the shell. Although the shrinkage of the shell tends to pull the shell from the mold walls, the pressure of the molten steel pushes the shell back into contact with the mold, bulging the shell and closing the gap. The opposing shrinkage and bulging produce gaps in the thicker and stronger areas of the shell mold wall while support is provided to the thinner. An intricate heat flow balance is formed as the shrinkage rate and the mold wall position, defined by the taper and mold distortion, determine the size of the gap which influences the heat removal and the shrinkage rate.

As the shell thickens down the mold, it requires less support to resist the pressure, allowing larger gaps to grow. The larger gaps increase the resistance to heat flow to the mold and slow solidification and shrinkage. Near the end of the mold the slower shrinkage is exceeded by the taper and gaps reduce in size. At high linear tapers, the gaps can be completely closed allowing the mold to push on the thicker, yet still fragile shell. Thus, early in the mold, high mold wall taper is required to eliminate bulging of the thin shell under the ferrostatic loads, while later in the mold, high tapers can apply detrimental compressive loads. Therefore, the mold wall position defined by the taper influences the intertwined mechanical and thermal behavior of the shell and can have a dramatic impact on the ultimate quality of the continuously cast steel slab.

At the end of the mold, the shell is strong enough to hold the inner pool of molten steel inside the cup-shaped solid shell. Rollers and the thicker shell provide support against the ferrostatic pressure below the mold. Without the mold walls, water or a mixture of water and air can be sprayed directly on the shell. This spray, as well as the direct contact with the cool rollers, improves the efficiency of the removal of heat from the strand by eliminating the insulating gap. Thus, the rollers just below the mold support the bulging shell and provide a mechanism for more efficient heat transfer than the mold provides. The rollers increase in stiffness further into the spray zone and begin applying mechanical loads to the nearly solidified strand. These loads impart a moment on the strand transforming the flat vertical strand into a flat horizontal slab reducing the height necessary to enclose a continuous caster.

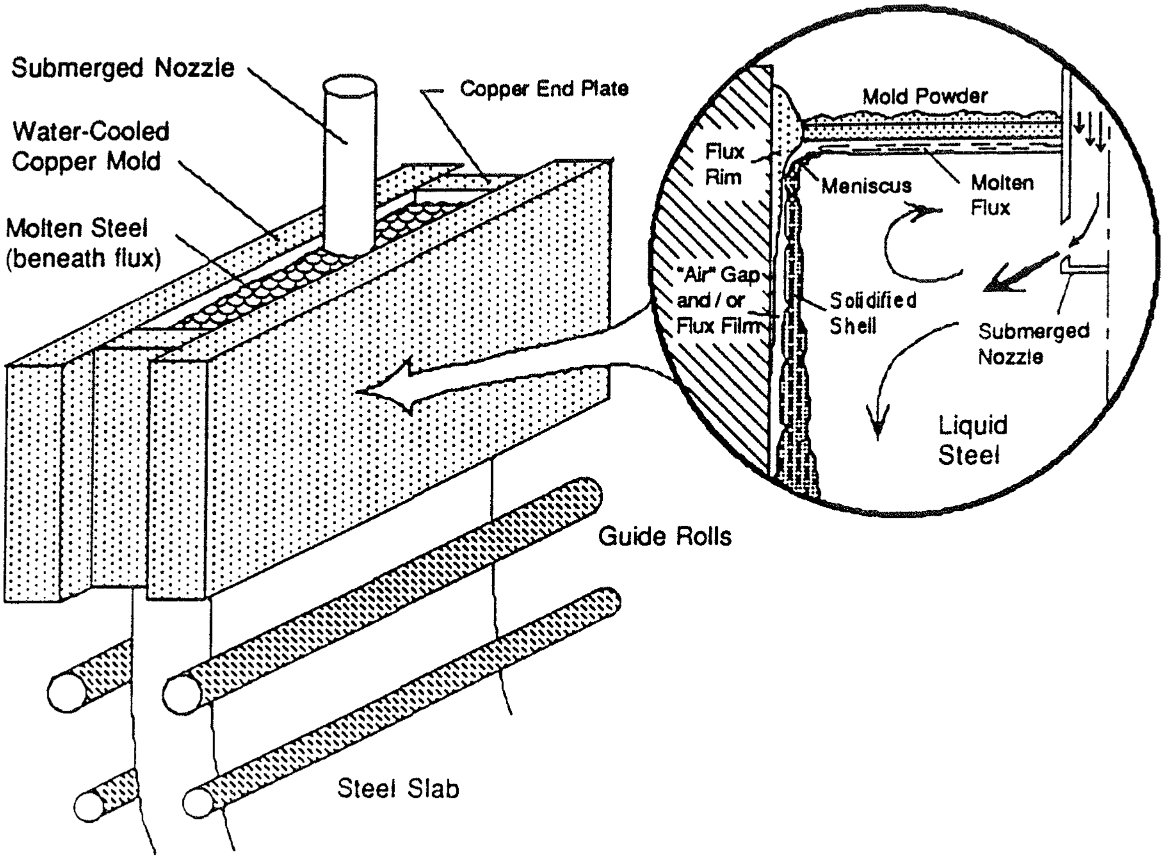


Figure 3 Illustration of the heat transfer phenomena occurring within the mold. This shows the interaction of the powder flux film, air gap, molten steel pool and forming shell.

The behavior of the shell under the intertwined thermal and mechanical loads imposed by the continuous casting process is believed to initiate the defect in the off-corner regions of the slab. The balance of the thermal and mechanical loads is controlled by the shrinkage of the shell, its strength to withstand the ferrostatic pressure pushing it toward the mold, the casting speed, the mold flux heat transfer characteristics, mold distortion, the manner of superheat dissipation, and the amount of mold taper present to offset the shell shrinkage. With improper taper, the combination of bulging under too little taper and compression under too much taper may distort the shell while it is weak and possibly fragile. The distortion can lead to defects such as surface shape problems, cracks, and even breakouts. Previous mathematical thermal stress models investigating the solidification in the mold have demonstrated this modelling is helpful to understand defects formed or initiated in the mold [2,3,4]. Combining results of concurrent work involving turbulent fluid flow modelling in the molten steel pool [5], and the results of recent mold distortion investigations [6] into a thermal stress model also capable of modelling the shell and mold interactions allows the optimization of mold taper design based on the control of the thermal and mechanical behavior.

A two-dimensional, transient, thermal stress model was developed to simulate the solidification and thermal stress phenomena in the mold as well as the mold wall and shell interactions that determine the effects of the mold taper design. This model tracks the thermal and mechanical behavior of a transverse slice through a continuously cast section as it moves down through the caster. Because of the two-dimensional nature of this modelling procedure, it is ideally suited for defects of a longitudinal nature such as the off-corner gutters. The model was applied to investigate the effects of mold taper on heat flow, shell growth, shrinkage, and stress development in the solidifying shell. The results of this investigation have suggested improvements to the continuous casting mold design that should reduce these problems. Application of this model into the thermal processing effects through the spray zone by Haegele [7] has also demonstrated this model to be an effective tool for solving problems associated with the latter stages of solidification.

2.0 MODEL DEVELOPMENT AND VERIFICATION

The step-wise coupled model developed for this research grew from two individual finite element models developed and described by Thomas [8,9,10]. This code incorporates many changes to allow a better simulation of the continuous caster. Due to the extent of these changes and for completeness, the entire model derivation and development is presented here.

2.1 THERMAL FINITE ELEMENT MODEL DEVELOPMENT

The thermal finite element model is formulated for a two-dimensional domain solving for temperatures within a transverse section of the caster. Boundary conditions are provided to simulate this transverse section as it steps along the axis of the caster, as shown in Figure 4, providing a full three-dimensional temperature distribution after a complete simulation. This procedure neglects the heat conduction along the axis of the caster, while it includes the heat convection that results from the casting velocity. Thus, the heat conduction along the axis must be small compared to the heat convection for this procedure to be valid. A study of the ratio of heat convection to heat conduction, which is indicated by the Peclet number [11], reveals that for typical operating conditions of continuous ferrous castings, this assumption is reasonable. However, continuous aluminum casting has a significant portion of its heat conducted in the casting direction due to its high thermal conductivity relative to its casting velocity. Therefore, the model developed for this work should only be applied to the continuous casting of ferrous materials.

The heat flow in continuous slab casting allows the use of two-fold symmetry on the finite element domain to simplify the problem and reduce the required solution time. The finite element domain employed for the study of the continuous casting process, as well as the mesh of constant gradient triangular elements used for these studies, is shown in Figure 4. The model is formulated to solve the transient heat conduction equation

$$\nabla(k\nabla T) = \rho c_p \frac{\partial T}{\partial t} \qquad 2.1.A$$

and the boundary heat flux equation

$$q'' = -k \frac{\partial T}{\partial n} \qquad 2.1.B$$

for the temperatures across the domain. After applying the Galerkin method described by Zienkiewicz [12], the finite element formulation of this equation is:

$$[K]\{T\} + [C]\{\dot{T}\} = \{Q\}. \qquad 2.1.C$$

The stiffness or conductivity matrix **[K]** represents the conduction through the material, the capacitance matrix **[C]** represents heat storage including the latent heat of solidification, and

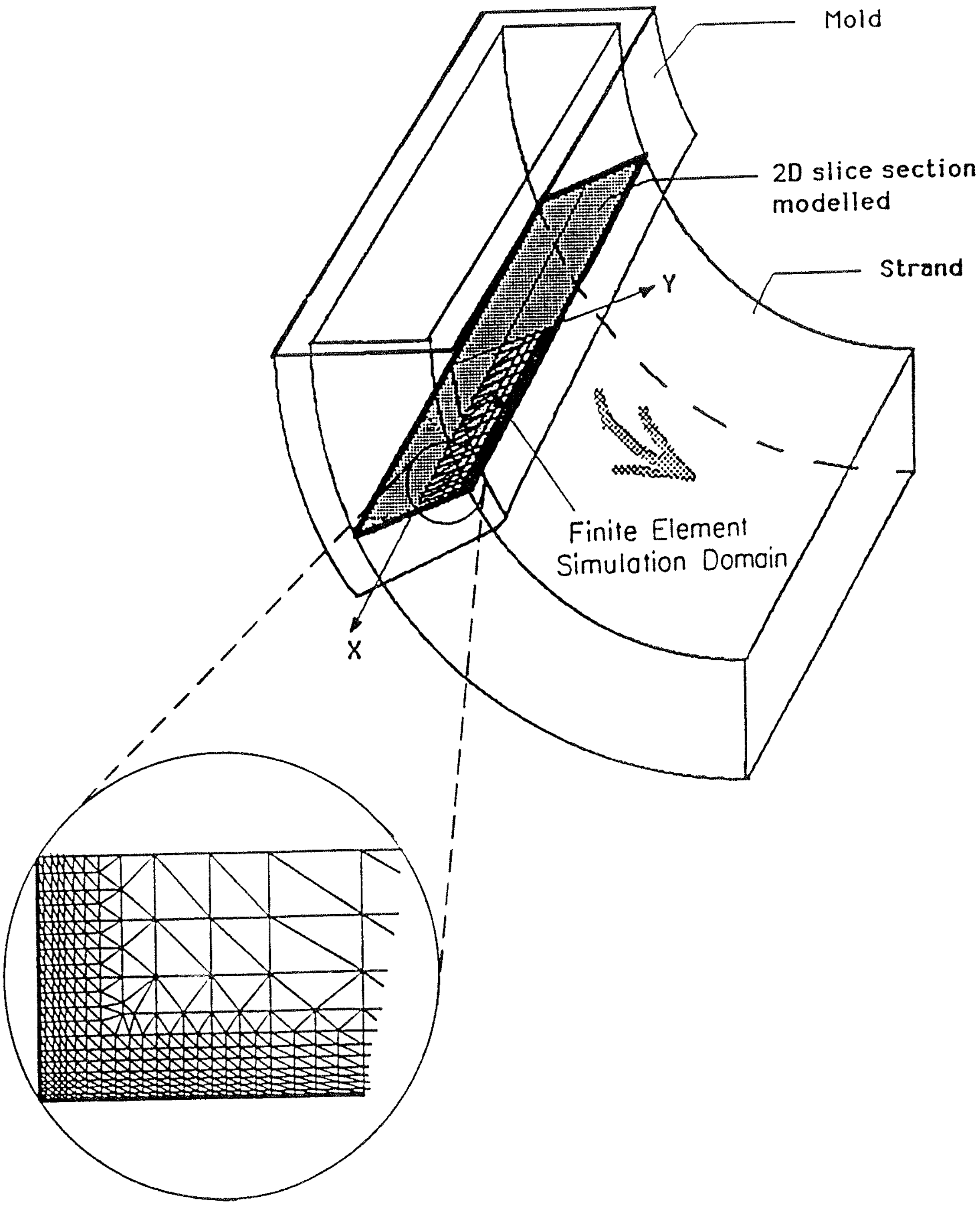


Figure 4 The meshed, finite element domain is shown at a particular time step. The two-dimensional solution produces results for the entire three dimensional mold by stepping along the axial direction of the caster.

the heat flow vector {**Q**} contains the effects of the net heat flow across the boundary and into the system. The gradient vector $\{\dot{T}\}$ contains the time derivatives of the temperatures {T}. This equation does not allow for the superheat dissipation due to convection in the molten pool, latent heat of solidification directly, or the process of evaluating the partial derivative of temperature with respect to time. A variety of techniques can be applied to incorporate these phenomena into the finite element method [8]. The techniques considered for this application are discussed in the following sections. The model is formulated for constant gradient triangular elements only, and the matrix formulation in the following sections are all based on this choice of element.

2.1.1 Conductivity Matrix

The conductivity matrix is derived directly from the Galerkin method. This method dictates that the elemental conductivity matrix takes the form:

$$\mathbf{K_{ij} = \frac{t \cdot k(T)}{4A} (b_i b_j + c_i c_j)} \qquad 2.1.1.A$$

where the gradient matrix elements are given by

$$\mathbf{b_i = y_j - y_k} \qquad 2.1.1.B$$

$$\mathbf{c_i = x_k - x_j} \qquad 2.1.1.C$$

and where **t** is the unit thickness which drops out of two-dimensional analyses, **k(T)** is the temperature dependent thermal conductivity of the material, and **A** is the area of the triangular element. The elemental conductivity matrix is assembled according to the element connectivity to form the global conductivity matrix [**K**] given in Equation 2.1.C.

2.1.2 Capacitance Matrix

Three techniques can be used to formulate the thermal capacitance matrix: the consistent capacitance matrix, the lumped capacitance matrix, or the area coordinate (Ohnaka) lumped capacitance matrix. The formulations associated with each technique are discussed by Thomas [8]. The consistent capacitance matrix formulation was employed for these simulations, and it is the only formulation discussed below.

The consistent capacitance matrix formulation is consistent with the Galerkin method of deriving the finite element matrix equations. This formulation becomes inaccurate with both large and small time step sizes. The other formulations are modifications derived from the consistent capacitance matrix that produce solutions which increase in accuracy as the time step is reduced in size. However in the next section, these lumping methods are shown to be inaccurate when the temperature gradient degrades from two-dimensional to nearly one-dimensional. The continuously cast slab is dominated by nearly one-dimensional heat flow due to its aspect ratio. The inaccuracy with small time step size, which can be controlled with proper input parameters, is more desirable than the inaccuracy of the one-dimensional heat transfer simulation with the lumping techniques for this simulation of slab casting.

The elemental consistent capacitance matrix that results from the application of the Galerkin method is described by the following equation.

$$\mathbf{C}_{ij} = \frac{\mathbf{t \cdot A}}{\mathbf{12}} \frac{\partial \mathbf{H(T)}}{\partial \mathbf{T}} (\mathbf{1}+\delta_{ij}) \qquad 2.1.2.A$$

where

$$\delta_{ij} = \mathbf{1 \text{ if } i=j \quad or} \quad \delta_{ij} = \mathbf{0 \text{ if } i \neq j} \qquad 2.1.2.B$$

and where **H(T)** is the temperature dependent enthalpy function. This elemental matrix is assembled into the global capacitance matrix **[C]** in Equation 2.1.C.

2.1.3 Enthalpy Function

The enthalpy function allows for step changes in the product of density and specific heat over a small temperature range and is employed in solidification modelling to account for the energy associated with a phase change [13]. Enthalpy, a thermodynamic property related to a material's internal energy state, is defined in this technique as the integral over temperature of the product of density and specific heat. The latent heat of solidification is added into the enthalpy function over the solidification temperature range producing the desired result incorporating the phase change energy into the material's energy state.

The enthalpy function is defined such that the partial derivative of this function with respect to time is equivalent to the product of density and specific heat. Therefore, this derivative can be replaced for this product in the transient heat conduction equation, Equation 2.1.A, and this derivative is incorporated into the thermal capacitance matrix according to Equation 2.1.2.A. The algorithm used to approximate the partial derivative of enthalpy with temperature is suggested by Lemmon:

$$\left(\frac{\partial H}{\partial T}\right)^2 = \frac{({}^{\partial H}/_{\partial x})^2 + ({}^{\partial H}/_{\partial y})^2}{({}^{\partial T}/_{\partial x})^2 + ({}^{\partial T}/_{\partial y})^2} \qquad 2.1.3.A$$

The enthalpy function and Lemmon's method of defining its derivative with temperature is employed in the model to insure all the heat associated with the phase change is accounted for in the heat diffusion model.

2.1.4 Time Stepping Technique

The transient heat conduction equation involves the partial derivative of temperature with time which is not explicitly accounted for by the finite element model; this derivative is approximated by monitoring the change in nodal temperature between time steps. Forward and backward Euler schemes, as well as modifications of these algorithms, have been used to approximate this partial derivative, but these two-step approaches can cause instabilities. A variety of three-level techniques have been developed to stabilize the computations. This model employs the Du Pont II, three-level time stepping technique which utilizes the results from two previous time steps to approximate the temperature gradient in time. This technique approximates the temperature gradient vector as

$$\{\dot{T}\} = \{ T_3 - T_2 \} / \Delta t. \qquad 2.1.4.A$$

Further, the technique extrapolates the temperature vector from the old temperatures by

$$\{T\} = {}^1\!/_4 \{ 3T_3 + T_1 \} \qquad 2.1.4.B$$

where $\mathbf{\Delta t}$ is the time step size, $\mathbf{T_3}$ is the desired temperature for the current iteration, $\mathbf{T_2}$ is the current temperature, and $\mathbf{T_1}$ is the temperature from the previous iteration. Substituting these equations into Equation 2.1.C gives the following equation

$$[A]\{T_3\} = \{F\} \qquad 2.1.4.C$$

where

$$[A] = [K] + {}^{[C]}/_{\Delta t} \qquad 2.1.4.D$$

and

$$\{F\} = \{Q\} - {}^1\!/_4 [K]\{T_1\} + {}^{[C]\{T_2\}}/_{\Delta t}. \qquad 2.1.4.E$$

The finite element code actually performs the assembly process concurrently with the formation of matrix **[A]** and vector **{F}**. Equation 2.1.4.C is supplied to the Cholesky decomposition solution routines and the new temperatures, $\mathbf{T_3}$, are returned into vector **{F}**.

Another time stepping technique, formulated by Lees [8], was considered during the model development phase of this work. This formulation calculates the temperature gradient with time and the extrapolation to the new temperature in a different manner than the Du Pont method. This formulation was not used to model the continuous caster because test problems proved the Du Pont technique more accurate for this formulation on a problem resembling the continuous caster.

2.1.5 Heat Flow Vector

Before Equation 2.1.4.C can be solved, the heat flow vector {Q} must be evaluated and added into the appropriate term of the force vector {F}. The heat flow vector contains the contributions of the boundary conditions which are not temperature dependent, including heat convection, heat flux, heat generation, and fixed temperature boundary conditions. Currently, the model is not set up to allow heat generation as a boundary condition. Fixed temperature boundary conditions are not directly added into the force vector. Instead, matrix operations are performed to insure that the resulting temperature at the specified node is fixed to the given value while maintaining the appropriate equations for this degree of freedom at all other nodes.

True finite element theory requires that heat convection terms effect both the force vector and the stiffness matrix because the heat flux is given by

$$\mathbf{q'' = h \cdot (T - T_{amb}).} \qquad 2.1.5.A$$

Therefore the heat convection coefficient, **h**, should be assembled into the stiffness matrix while the ambient term, $\mathbf{hT_{amb}}$, is included in the force vector. The transient nature of this finite element model allows the heat convection calculation to be converted to a surface heat flux based on the nodal temperature from the previous step [8]. This introduces little error provided the time steps are small and allows the conductivity matrix to remain triangularized throughout the entire iteration.

The conversion of the heat convection boundary condition to a heat flux condition produces only one type of thermal boundary condition that effects the heat flow vector, that being the heat flux condition. Deriving this condition from the Galerkin method reveals that

$$\mathbf{Q_i = Q_j = \tfrac{1}{2}Lq''t} \qquad 2.1.5.B$$

where $\mathbf{Q_i}$ is the heat flow into node i and **q"** is the specified heat flux at the boundary. **L** is length of the boundary line between node i and node j.

2.1.6 Interfacial Gap Heat Transfer

The interfacial gap heat transfer boundary condition is a subset of the specified heat flux boundary condition that has been developed specifically to aid the modeling of the heat transfer from the continuously cast shell to the mold. The algorithm used to calculate this boundary condition uses many parameters available from the coupled simulation to calculate the heat flux from every element edge on the surface of the shell. This heat flux is loaded into the thermal force vector as described by Equation 2.1.5.B.

Heat transfer from the surface of the shell to the mold occurs predominantly by two mechanisms: radiation and conduction across the interfacial gap. A third mechanism, radiation-conduction, occurs in semi-transparent material such as molten continuous casting powder films, but is important only at very high temperatures for thick powder films. This mode of heat transfer is negligible with common casting powders and practice and can be neglected without neglecting an influential part of the caster's heat flow. Radiative heat transfer becomes significant across large gaps, which have extremely high resistances to conductive heat flow because of the path length. With small gaps, which are prevalent near the meniscus and along most of the wide face throughout the length of the mold, heat conduction across the shell-mold interface is the dominant mode of heat transfer. Therefore, both radiation and conduction heat transfer modes are critical to the understanding of the shell-mold interactions and the heat transfer across this interface.

Conduction across the interface is particularly sensitive to the thickness of the gap that forms as the shell solidifies and shrinks away from the mold wall. The step-wise coupling process, described in a following section, allows the thermal loads to be calculated from an accurate prediction of the gap size producing an accurate thermal load for this mode of heat transfer.

The heat transfer characteristics of the powder and the gas that fills the void also play a key role in determining the amount of heat conducted to the mold. The parameters that influence this heat transfer and the resulting heat flow path resistance model are shown in Figure 5. The path resistance model used in the finite element code employs a radiation resistance in parallel with a series of 4 heat conduction resistances. These 4 conduction resistances represent:

1) the contact resistance between the mold and flux film,

2) the conduction through the flux film,

3) the conduction through the gas vapor gap, and

4) the contact resistance between the flux and the steel shell.

The flux film is assumed to grow from a minimum at the meniscus to a maximum thickness based on the casting speed and the casting powder's cooling rate, temperature dependent viscosity, and melting rate. The thickness of the gas vapor gap is the remainder of the full gap which is not filled with powder.

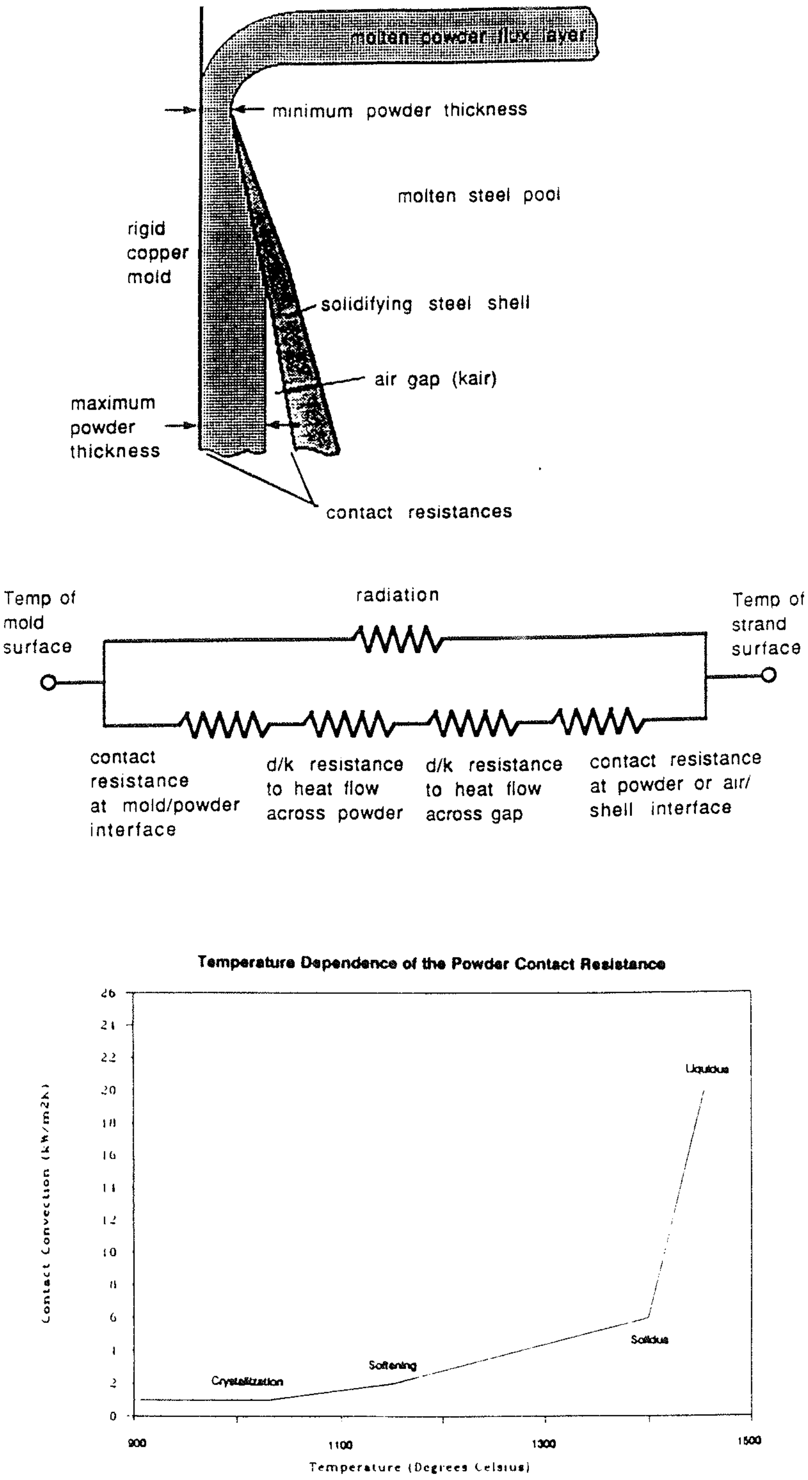


Figure 5 The electrical analog is used to represent the heat flow resistance model employed to calculate the heat flow across the interfacial gap, using the temperature dependent properties and input parameters.

The contact resistance between the mold and the powder flux film is given by

$$\mathbf{R}_{mp} = \frac{\mathbf{1}}{\mathbf{h}_o} \qquad 2.1.6.A$$

where $\mathbf{h}_o$ is the contact convection coefficient at the mold surface, which is set to 3000 W/m^{2}K (3.67 BTU/hrF·in^2). The resistance to conduction through the flux film is calculated by

$$\mathbf{R}_{cp} = \frac{\mathbf{d}}{\mathbf{k}_p} \qquad 2.1.6.B$$

where $\mathbf{k}_p$ is the thermal conductivity of the powder and **d** is the thickness of the gap that is filled with powder. The conductivity of the powder flux used is 0.70 W/mK (0.0337 BTU/hrF·in). The thickness of the powder gap is the minimum powder thickness, 0.1 μm ($3.94*10^{-9}$ in), if the gap size is less than or equal to this thickness; the gap thickness is the true gap size if the gap is greater than the minimum powder thickness but less than the maximum powder thickness; or the gap thickness is the maximum powder thickness, 50 μm ($1.97*10^{-6}$ in), if the gap size is larger than this thickness.

The resistance to conduction through the air gap, $\mathbf{R}_{ag}$, follows the relation expressed in Equation 2.1.6.B. The conductivity of the air is 0.06 W/mK ($2.89*10^{-3}$ BTU/hrF·in) and the gap size is equal to the difference of the predicted gap from the previous mechanical iteration and the maximum powder gap size. The contact resistance between the powder flux and the steel shell, $\mathbf{R}_{fs}$, follows the relation in Equation 2.1.6.A except the contact convection coefficient must be temperature dependent due to the large change in viscosities in both the steel and the powder over the surface temperature range. This temperature dependence is illustrated in Figure 5 and is shown in the table below.

Temperature Description	Temperature		Contact Convection	
Powder Crystalline Temperature	1030°C	1885°F	1000W/m^{2}K	48.1 BTU/hrF·in^2
Powder Softening Temperature	1150°C	2100°F	2000W/m^{2}K	96.2 BTU/hrF·in^2
Metal Solidus Temperature	1399°C	2550°F	6000W/m^{2}K	289 BTU/hrF·in^2
Metal Liquidus Temperature	1454°C	2650°F	20000W/m^{2}K	962 BTU/hrF·in^2

The resistance to the radiative heat flow is linearized with the temperature difference between the two transferring surfaces by

$$\mathbf{R}_{rad} = \frac{\mathbf{1}}{\mathbf{h}_{rad}} \qquad 2.1.6.C$$

where

$$\mathbf{h}_{rad} = \sigma\varepsilon*(\mathbf{T}_s+\mathbf{T}_m)*(\mathbf{T}_s^{\,2}+\mathbf{T}_m^{\,2}) \qquad 2.1.6.D$$

where σ is the Stefan-Boltzmann constant, ε is the emissivity of the two surfaces (0.8), $\mathbf{T}_s$ is the temperature of the shell surface at the previous iteration, and $\mathbf{T}_m$ is the surface temperature of the mold wall [14]. These heat flow resistances combine to produce the following relations governing the heat flow across the element boundary:

$$q'' = \frac{(T_s - T_m)}{R_{tot}} \qquad 2.1.6.E$$

where

$$R_{tot} = \frac{R_{rad} * (R_{mp} + R_{cp} + R_{ag} + R_{fs})}{R_{rad} + R_{mp} + R_{cp} + R_{ag} + R_{fs}} \qquad 2.1.6.F$$

This heat flux contributes to the heat flow vector {Q} according the Equation 2.1.5.B. The procedure described above accounts for the influence of the interfacial gap heat transfer on the temperature distribution and the heat flow from the shell.

2.2 THERMAL FINITE ELEMENT MODEL VERIFICATION

The verification of the thermal finite element model compares finite element results of two-dimensional corner solidification to those analytically derived by Rathjen and Jiji [14]. The material properties used are representative of steel with the constant material property and unique solidification temperature restrictions imposed by the analytical solution. The table below summarizes the properties used. This analytical solution was also used to determine which formulation discussed by Thomas [8] would most closely approximate the continuous casting process.

Material Property	SI units	English units
Density	7200 kg/m^3	449.5 lbm/in^3
Specific Heat	750 J/kg·K	0.179 BTU/lbm·F
Thermal Conductivity	30 W/m·K	1.444 BTU/hr·F·in
Latent Heat of Solidification	262.5 kJ/kg	0.113 BTU/lbm
Solidus Temperature	1499°C	2730°F
Liquidus Temperature	1501°C	2734°F

The initial and boundary conditions for this problem are shown on a portion of the uniform mesh in Figure 6. The temperatures of Point 1 and Point 2, located in this figure, were compared against the analytical values. Point 1 is located near the corner where temperature gradients and heat flow would be two-dimensional. Point 2 is located on the adiabatic boundary on the short side where the heat flow would be highly one-dimensional. The thermal formulation chosen to model the slab casting must be able to model both regions well because the defect formation occurs in the two-dimensional corner area while the majority of the slab along the wide face exhibits more one-dimensional thermal behavior.

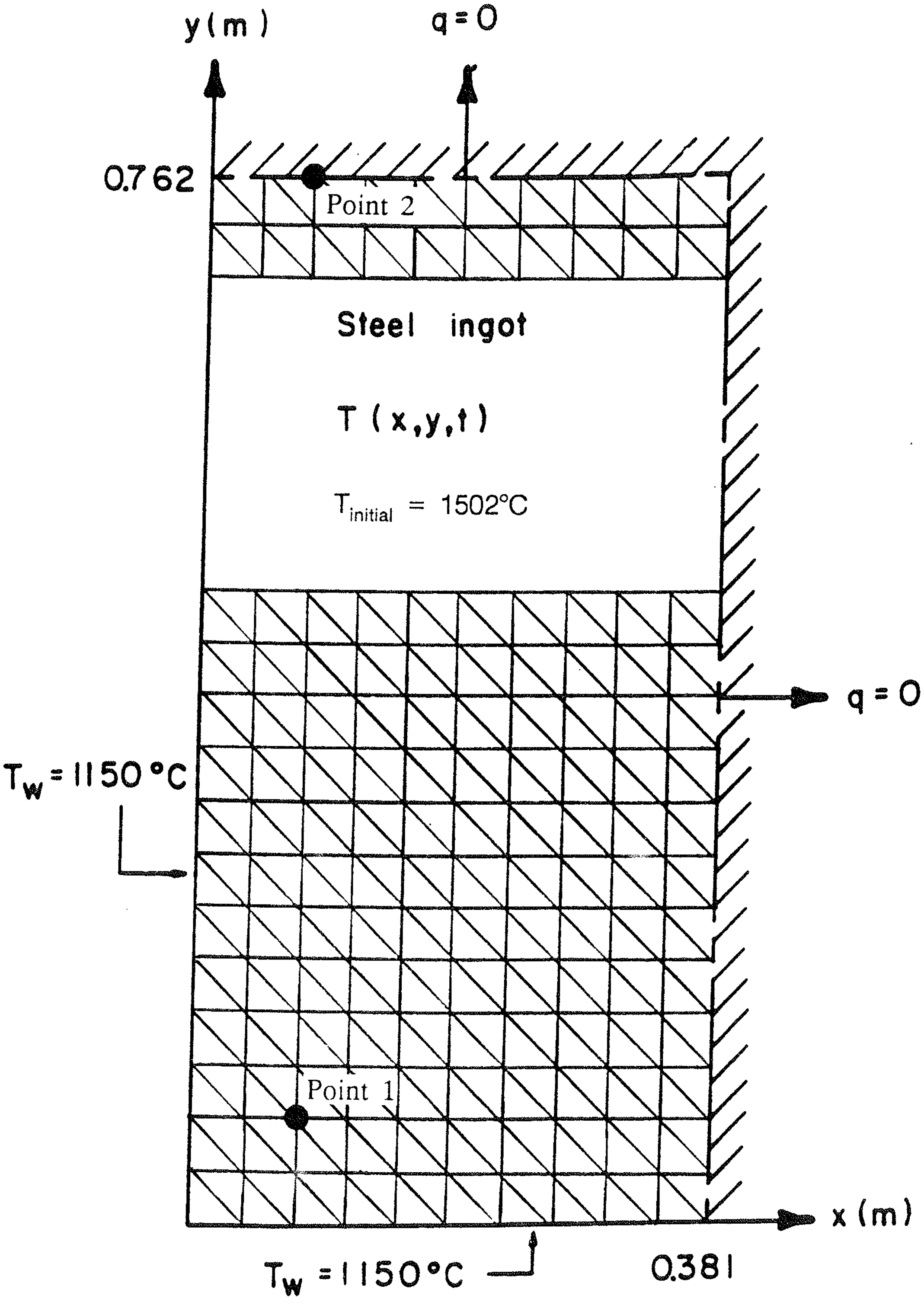


Figure 6 The two-dimensional corner solidification problem was analyzed using this mesh and boundary conditions.

The temperature histories for Point 1 and Point 2 are provided in Figure 7 for a variety of the formulations discussed by Thomas [8]. The formulations compared in this figure include the Ohnaka or area coordinate lumped capacitance matrix with the Du Pont II three-level time-stepping technique (Ohnaka1), the Ohnaka lumped capacitance matrix with the Lees three-level time-stepping technique (Ohnaka2), and the consistent capacitance matrix with the Du Pont II three-level time-stepping technique (Consistent). This figure illustrates that these formulations provide adequate results when the temperature gradients are two-dimensional. All formulations lose some accuracy when the temperature gradient is rather one-dimensional, near Point 2, however the consistent capacitance formulation produces more accurate results for this one-dimensional condition than the Ohnaka lumped formulation with either time stepping technique. This suggests that although the mesh is probably too coarse to accurately resolve the true temperature gradients, the consistent capacitance formulation with the Du Pont II three-level time stepping technique is more acceptable than the other formulations. Therefore, this combination was applied for all further modelling.

The consistent capacitance matrix formulation loses accuracy for both large and small time step sizes; therefore the time step used in modelling these solidification problems must be carefully chosen. The effect of time step size was investigated with this analytical solution and these results are summarized in Figure 8. These results indicate that a time step between 60 and 120 seconds produces the most accurate results for this mesh with these material properties. They also indicate that accuracy is acceptable with a time step size differing from the optimum by nearly a factor of three. This test problem has indicated that the optimum time step size can be calculated by

$$\Delta t = 0.3 \cdot (\Delta x)^2 / \alpha \qquad 2.2.A$$

where Δt is the optimum time step size, Δx is the size of the element in the direction of the heat flow, and α is the thermal diffusivity of the material just below the solidus temperature. This result has been confirmed with many one dimensional solidification problems.

2.3 MECHANICAL FINITE ELEMENT MODEL DEVELOPMENT

The mechanical finite element model utilizes the same mesh employed to solve the thermal solidification problem. This two-dimensional mesh of constant strain triangles requires a two-dimensional stress-state assumption. Traditional two-dimensional stress-state assumptions include plane strain under which the out of plane strain is restricted to zero everywhere in the domain, plane stress under which the out of plane stress is restricted to zero, and generalized

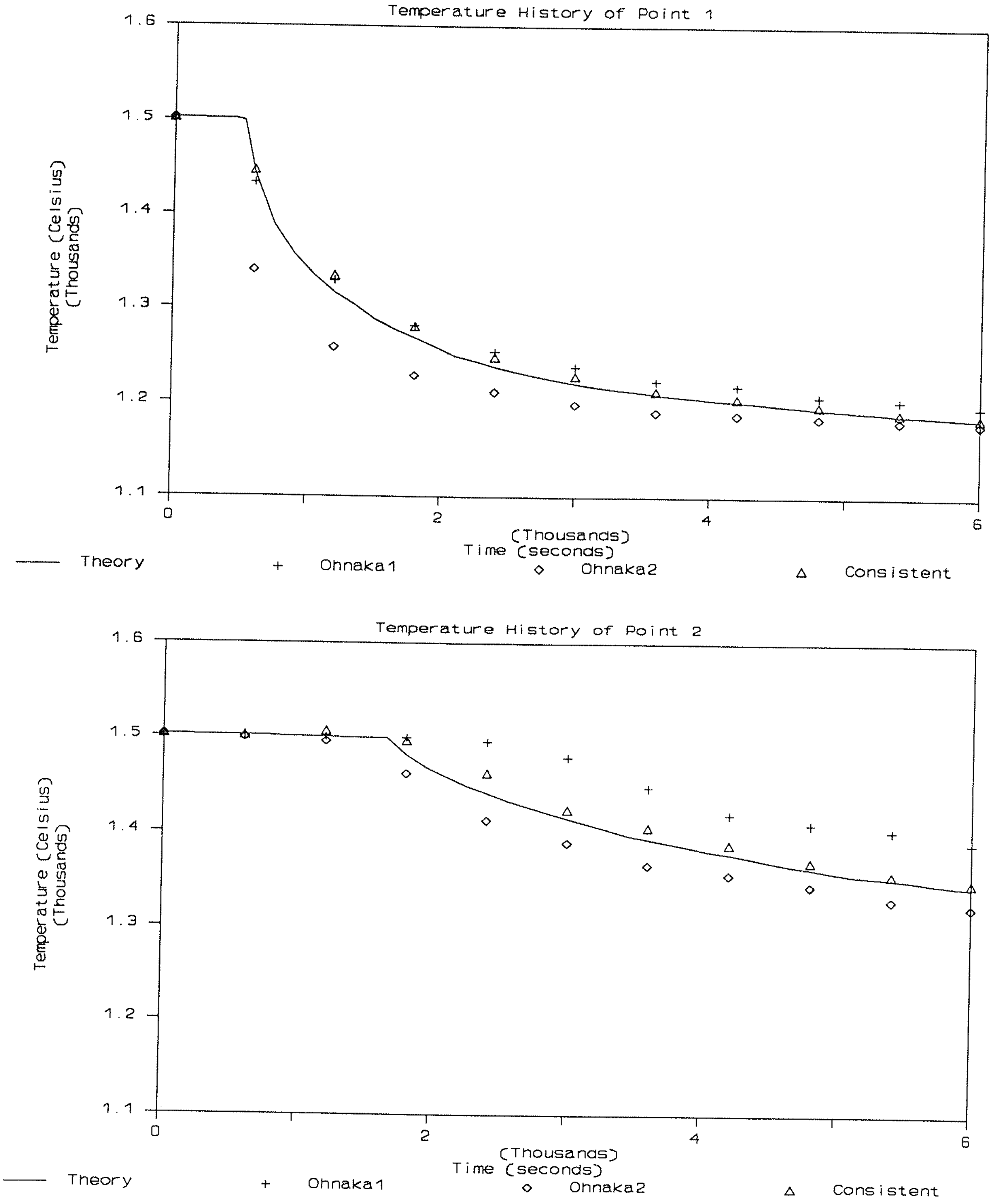


Figure 7 The consistent capacitance matrix with the DuPont II time stepping technique provides the most accurate solution in both one and two-dimensional temperature gradients.

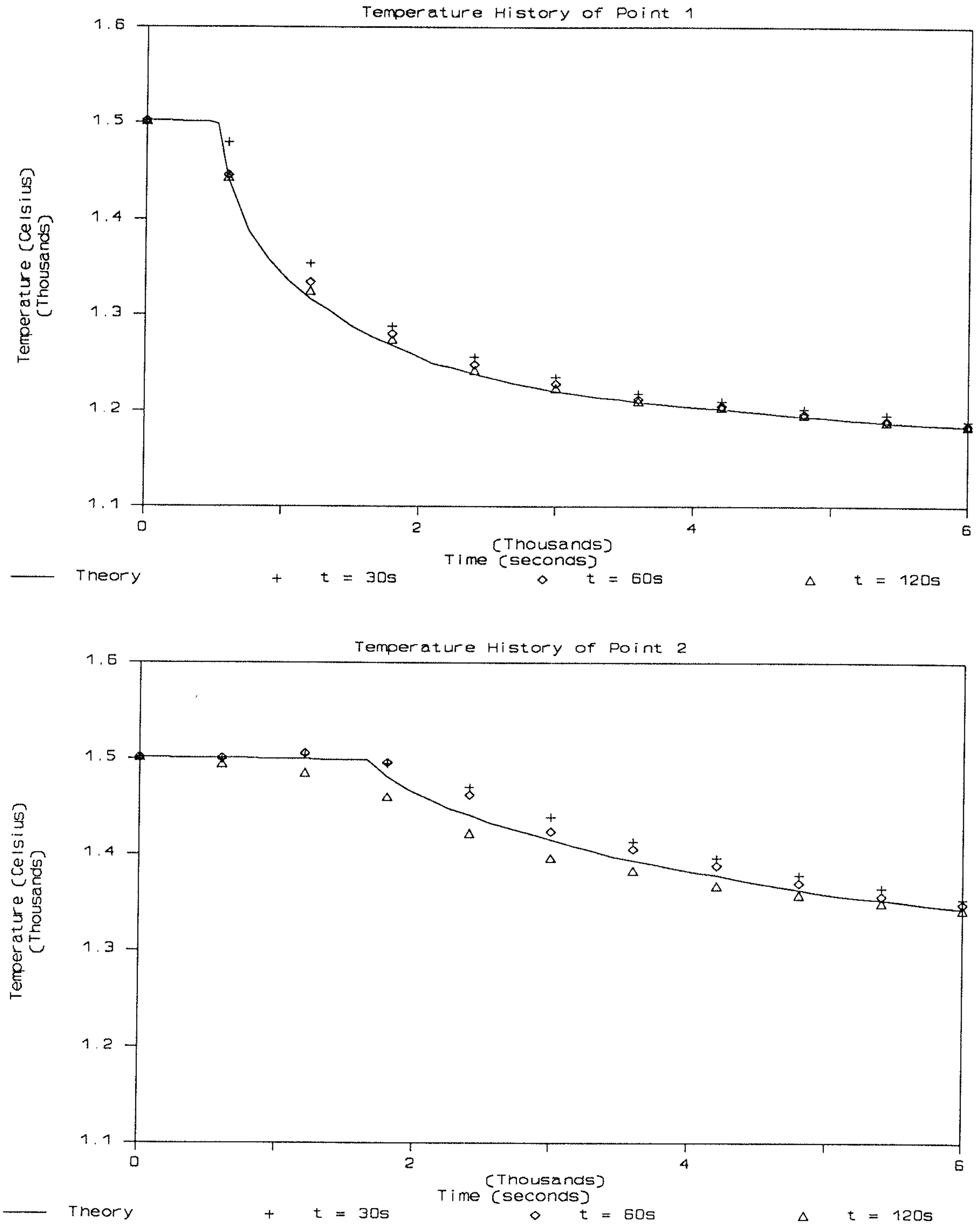


Figure 8 The wide range of time steps producing acceptable solutions and avoiding the inherent instabilities of the consistent matrix formulation is illustrated in these results.

plane strain under which the out of plane strain is restricted to a constant which minimizes the average out of plane stress over the entire domain. This model was developed to allow these three stress-state assumptions. The two-dimensional restriction requires that only the x direction displacement, u, and the y direction displacement, v, can be obtained from the mechanical solution. From these displacements and the governing two-dimensional stress-state assumption, the x, y, and z normal strains and the xy shear strain can be determined. The structural stiffness matrix and force vector formulations vary under the governing stress-state assumption; these formulations are discussed below.

2.3.1 Stiffness matrix

The elemental structural stiffness matrix is derived by applying Galerkin's method of minimizing the approximation error for the displacements at every node in the domain. The two displacements at each of the three nodes of the triangular element produces an elemental displacement vector of length six requiring a six by six elemental stiffness matrix. Assuming the elemental displacement matrix is arranged as { $\mathbf{u}_i$, $\mathbf{v}_i$, $\mathbf{u}_j$, $\mathbf{v}_j$, $\mathbf{u}_k$, $\mathbf{v}_k$ }, to reduce the global stiffness matrix storage requirements, the six by six elemental stiffness matrix can be expressed as

$$\mathbf{K}_{(2i-1)(2j-1)} = \frac{\mathbf{t*E}}{\mathbf{4A\lambda}} \ (\mathbf{b}_i\mathbf{b}_j\mathbf{e}_1 + \mathbf{c}_i\mathbf{c}_j\mathbf{e}_4) \qquad 2.3.1.A$$

$$\mathbf{K}_{(2i-1)(2j)} = \frac{\mathbf{t*E}}{\mathbf{4A\lambda}} \ (\mathbf{b}_i\mathbf{c}_j\mathbf{e}_2 + \mathbf{b}_j\mathbf{c}_i\mathbf{e}_4) \qquad 2.3.1.B$$

$$\mathbf{K}_{(2i)(2j-1)} = \frac{\mathbf{t*E}}{\mathbf{4A\lambda}} \ (\mathbf{b}_j\mathbf{c}_i\mathbf{e}_2 + \mathbf{b}_i\mathbf{c}_j\mathbf{e}_4) \qquad 2.3.1.C$$

$$\mathbf{K}_{(2i)(2j)} = \frac{\mathbf{t*E}}{\mathbf{4A\lambda}} \ (\mathbf{c}_i\mathbf{c}_j\mathbf{e}_1 + \mathbf{b}_i\mathbf{b}_j\mathbf{e}_4) \qquad 2.3.1.D$$

With i and j ranging from 1 to 3. The λ parameter is defined as

$$\lambda = \mathbf{1} - \nu^2 \qquad 2.3.1.E$$

for plane stress analyses and

$$\lambda = (\ \mathbf{1+\nu}\ )(\ \mathbf{1-2\nu}\ ) \qquad 2.3.1.F$$

for plane strain and generalized plane strain analyses. The constants $\mathbf{b}_i$, $\mathbf{c}_j$, $\mathbf{A}$, and $\mathbf{t}$, are the geometric parameters described in Section 2.1.1, $\mathbf{E}$ is the average elastic modulus for the material of the element at the three nodal temperatures, and ν is the Poisson ratio. The constants multiplied by these geometric parameters are also stress-state dependent as described in the equations below.

Plane Stress	(Generalized) Plane Strain	
$e_1 = \mathbf{1}$	$e_1 = \mathbf{1} - \nu$	2.3.1.G
$e_2 = \nu$	$e_2 = \nu$	2.3.1.H
$e_3 = \mathbf{1} - \nu$	$e_3 = \mathbf{1} - 2\nu$	2.3.1.I
$e_4 = {}^1/_4\, e_3$	$e_4 = {}^1/_4\, e_3$	2.3.1.J

The elemental stiffness matrix is assembled into the global stiffness matrix, **[K]**. The incremental global displacement vector, **{u}**, is the result of solving

$$[K]\ \{u\} = \{F\} \qquad 2.3.1.K$$

by the Cholesky decomposition solution routines also used for the heat transfer matrix solution. The incremental mechanical force vector **{F}** is discussed in the following section.

2.3.2 Mechanical Force Vector

The incremental mechanical force vector **{F}** accounts for the surface traction forces, body forces, pressure loads, thermal loads, inelastic loads, initial stresses, and initial strains within a load step. Because of the two-dimensional stress-state assumptions, this vector must also account for some terms associated with the structure's behavior in the third dimension.

An incremental method of solution, as described by Mendelson [16], is employed to account for the time dependent inelastic or creep phenomena and to ease the formulation with transient thermal loads. This method requires that the force vector components contain only the increment in force between the current load step and the previous. Therefore, the solution of Equation 2.3.1.K results in incremental displacements for each load step. The incremental displacements are added to the previous total displacements at the end of the load step to yield the current total displacements. Thus, the initial stress and strain effects on the force vector need not be considered because these are stored in the total stress and strain vectors before the solution begins.

Surface traction loads vary linearly across the surface according to the shape function employed in the derivation of the constant strain triangular element. These loads are assembled into the global mechanical force vector at the two nodes of the element boundary on which the surface traction acts. Galerkin's method [10] requires that linearly varying surface traction loads assemble into the global force vector according to

$$\mathbf{F}_{xi} = {}^{1}\!/_{6}\ \mathbf{Lt}\ (\ 2\mathbf{S}_{xi} + \mathbf{S}_{xj}\ ) \qquad 2.3.2.A$$

$$\mathbf{F}_{yi} = {}^{1}\!/_{6}\ \mathbf{Lt}\ (\ 2\mathbf{S}_{yi} + \mathbf{S}_{yj}\ ) \qquad 2.3.2.B$$

where $\mathbf{F}_{xi}$ is the force vector component for the x direction of the i node on the boundary and $\mathbf{S}_{xi}$ is the surface traction force component in the x direction evaluated at node i. **L** is the length of the boundary and again **t** is the unit thickness. Equation 2.3.2.B represents the same quantities except for y component of the traction force. Pressure loading on the surface is a specific type of surface traction loading. For pressure loading, the load is constant over the surface and the resultant component direction must by perpendicular to the boundary surface. Thus, pressure loading is a class of surface loads and do not have to be handled differently.

Body forces act throughout the domain of the finite element mesh. The formulation used for this model requires the body force must be constant across an element. If the body force varies over the element, the average at the three nodal positions are used. With this restriction, the elemental body force vector is

$$\mathbf{F}_{xi} = {}^{1}\!/_{6}\ \mathbf{t}\ \mathbf{f}_{x} \qquad 2.3.2.C$$

where $\mathbf{F}_{xi}$ is the x-component of the body force on the i node of the element and $\mathbf{f}_x$ is the average body force in the x direction. Similar equations could be written for the y direction at the j and k nodes of the element. Body forces in the third direction cannot be handled in this model due to the two-dimensional restriction.

Thermal loads act throughout the entire structure including the third direction not modelled by the two-dimensional finite element domain. Therefore, the formulation of this component of the load vector is dependent upon the choice of stress-state assumptions. Under either assumption the thermal load vector component is

$$\mathbf{F}_{xi} = \frac{\mathbf{tE{\cdot}\Delta TLE}}{\mathbf{2\phi}}\ \mathbf{b}_i \qquad 2.3.2.D$$

for the x component and

$$\mathbf{F}_{yi} = \frac{\mathbf{tE{\cdot}\Delta TLE}}{\mathbf{2\phi}}\ \mathbf{c}_i \qquad 2.3.2.E$$

for the y component. The constant, ϕ for plane stress is given by

$$\phi = \mathbf{1} - \nu \qquad 2.3.2.F$$

or by

$$\phi = \mathbf{1} - \mathbf{2}\nu \qquad 2.3.2.G$$

under the plane strain or generalized plane strain assumptions. Again, **E** is the average elastic modulus at the three nodal temperatures, **t** is the unit thickness, and $\mathbf{b}_i$ and $\mathbf{c}_i$ are the geometric parameters described earlier. The **ΔTLE** variable represents the difference in the thermal linear expansion function over the temperature range of the incremental load step. The thermal linear expansion function is the integral of the thermal expansion coefficient over temperature and describes both the change in thermal expansion coefficient with temperature as well as the expansion that can be associated with the solid to liquid phase change as well as solid to solid phase changes [10].

Inelastic material behavior effects another component of the mechanical force vector. This formulation of this inelastic component is derived from the Prandtl-Reuss equations for plastic flow and the elastic-plastic stress-strain relations from Mendelson [16]. The Prandtl-Reuss equation for the incremental plastic strain in the x direction follows.

$$\Delta\varepsilon_x{}^p = \frac{\Delta\varepsilon_e{}^p}{\sigma_e}\ (\ \sigma_x - {}^1/_2\sigma_y - {}^1/_2\sigma_z\ ) \qquad 2.3.2.H$$

where $\Delta\varepsilon_e{}^p$ is the incremental effective plastic strain within the load step which is calculated from constitutive equations for total plastic strain or plastic strain rate. The total effective stress, σ_e is calculated using the Von Mises criterion from the component total stresses: σ_x, σ_y, σ_z, and τ_{xy}. Similar equations for the incremental plastic strain in the other normal directions can be derived by interchanging the subscripts. The incremental plastic shear strain is given by

$$\Delta\gamma_{xy}{}^p = \frac{3\Delta\varepsilon_e{}^p}{2\sigma_e}\ \tau_{xy} \qquad 2.3.2.I$$

These above equations describe how the effective plastic strain, given from the material properties and the stress-state, can be separated into the individual components of incremental plastic strain. Mendelson's plastic stress-strain equations combined with the Galerkin method produce the following equations describing the influence of these inelastic loads on the mechanical load vector in the x and y directions respectively.

$$\mathbf{F}_{xi} = \frac{\mathbf{t{*}E}}{\mathbf{2{*}\lambda}}\ (\mathbf{b}_i{*}(\mathbf{e}_1\Delta\varepsilon_x{}^p+\mathbf{e}_2\Delta\varepsilon_y{}^p)\ +\ {}^1/_2\mathbf{c}_i\mathbf{e}_3\Delta\gamma_{xy}{}^p) \qquad 2.3.2.J$$

$$\mathbf{F}_{yi} = \frac{\mathbf{t{*}E}}{\mathbf{2{*}\lambda}}\ (\mathbf{c}_i{*}(\mathbf{e}_2\Delta\varepsilon_x{}^p+\mathbf{e}_1\Delta\varepsilon_y{}^p)\ +\ {}^1/_2\mathbf{b}_i\mathbf{e}_3\Delta\gamma_{xy}{}^p) \qquad 2.3.2.K$$

The constants $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$, and λ are defined in Equations 2.3.1.E through 2.3.1.J.

The final effects to be added into the mechanical load vector result from the two-dimensional stress-state assumption. In a three-dimensional analysis, the derivation of the stiffness matrix accounts for these loads. However, the elimination of the z-direction displacement from the global displacement vector requires the z-displacement terms in the equations be accompanied into the force vector. These loads are given by

$$F_{xi} = \frac{tE\eta}{2\cdot\lambda} \left(\Delta\varepsilon_z^p - \Delta\varepsilon_z \right) \mathbf{b}_i \tag{2.3.1.L}$$

and

$$F_{yi} = \frac{tE\eta}{2\cdot\lambda} \left(\Delta\varepsilon_z^p - \Delta\varepsilon_z \right) \mathbf{c}_i \tag{2.3.1.M}$$

where η is zero under the plane stress-state assumption and η is equal to the Poisson ratio of the material under the plane strain or generalized plane strain assumptions. Because η is zero for plane stress and $\Delta\varepsilon_z$ is zero for plane strain, this component of the incremental mechanical force vector only has an effect when plastic loads are included for the plane strain condition or when the generalized plane strain assumption is used. These loads are often lumped into the plastic loading term; separating them emphasizes they originate merely because of the two-dimensional assumption and are not part of the derivation for plastic loading.

Surface traction, body forces, thermal expansion loads, inelastic flow loads, and z-dimension loads are components of the incremental mechanical load vector that supply the boundary conditions and loads to the mechanical finite element problem. These components are assembled in the global mechanical force vector. The mechanical simulation of the continuous caster provides various methods of calculating some of these loads, particularly the surface contact loads, which vary throughout the caster. These modifications necessary for the continuous casting simulations are described in the following sections.

2.3.3 Ferrostatic Pressure Application

The liquid pool in the vertical continuous caster builds up ferrostatic pressure inside the solidifying shell which increases lower in the mold under the weight of the liquid. The pressure bulges the shell toward the mold decreasing the interfacial heat transfer gap. An analysis investigating the interaction between the shell and the mold must account for this load. The pressure load cannot be applied merely as a boundary condition because the surface it acts upon moves as the solidification front progresses into the liquid pool. The algorithm used to apply the ferrostatic pressure searches all elements to determine where the solidification front is located and applies the pressure to the nodes on this surface.

The magnitude of the ferrostatic pressure loading is calculated assuming the fluid is a static pool [17]. At the meniscus of the caster, the free surface of the liquid pool has a pressure equal to atmospheric pressure, thus no load is applied, however further down the mold pressure is applied. The magnitude of the total pressure is equal to the mass density of the molten material times the acceleration of gravity times the vertical distance below the meniscus. The incremental formulation of this finite element model requires that only the portion of the load that did not exist at the previous step be loaded into the incremental load vector for the current load step. Thus, only the change in vertical position from the current load step to the previous load step enters the calculation of the magnitude of the ferrostatic pressure force.

The incremental pressure load is applied continuously around the perimeter at the solid-liquid interface. To determine the nodes on the element edges where this pressure is applied, the model defines a temperature, $\mathbf{T}^*$, slightly larger than the solidus temperature. For elements that have exactly two nodes with temperatures greater than $\mathbf{T}^*$, surface traction components are determined to require the pressure acts perpendicular to the surface between the two hot nodes in the direction of the cold node and with the magnitude of the incremental pressure. These surface traction components are computed and assembled into the force vector according to Equations 2.3.2.A and 2.3.2.B.

2.3.4 Mechanical Interface Constraints

The interaction between the shell and the mold not only influences the heat transfer significantly, but also effects the loading on the exterior portion of the shell. As the shell shrinks away from the mold, the exterior surface of the shell is a free face and carries no load. However, where the shell contacts the mold wall, load is transferred between the two structures causing some deflection in both depending on their relative stiffnesses. The details of modelling the mechanical interface to simulate its proper response provides problems for the finite element method. Several techniques have been developed to account for the loading at mechanical interfaces.

Ohnaka [4] has employed one technique accounting for contact loading for another application to continuous casting. This method estimates the load required to narrowly avoid penetration of the mold by the shell and applies this load to the shell on the first iteration. The displacement solution is obtained and reaction forces are calculated. This procedure iterates until the penetration is negligible and reaction forces exist only where the mold contacts the shell.

One advantage with this method is that the application of the load effects only the force vector, requiring changes only to the load vector and allowing the triangularized stiffness matrix to be reused for solution with a new estimated load. Another advantage is that the stiffness of the system can be estimated after only one iteration which can lead to a very close estimation for subsequent iterations.

The disadvantages with this method is that small errors in the load produce surface displacements which cause large stresses that the plasticity parameters will iterate on until the stresses relax. This procedure will eventually lead to a proper solution, but the iteration on the plasticity with the large load will slow solution time. The trade-offs between the triangularization time savings and the extra plasticity iteration time requirement can only be evaluated by comparable application of the various techniques. However, the scheme employed by Ohnaka is worthy of merit, and investigation and evaluation of this technique may improve this finite element model.

This model tackles the mold wall contact problem by incorporating a stiff spring into the stiffness matrix to force the shell to the mold wall position on a subsequent iteration if contact exists after the first iteration. This neglects the mold deflections under the reaction load which are small due to the high mold stiffness. This method is assured of applying a reasonable load to the structure because the shell is pushed no further than is necessary to avoid interference with the mold. It does, however, modify the stiffness matrix so retriangularization time during solution is an added cost on each iteration.

The major disadvantage with this method of evaluating intermittent contact is that as the center along a face of the shell is pushed to the mold wall, the nearby surface is carried with it. Therefore, not all nodes that interfere would require the stiff spring loads to avoid interference with the mold wall. These nodes that do not require the spring would instead be pulled back by the spring to the mold wall position, simulating completely unrealistic contact conditions for the continuous caster. The procedure eventually employed to handle the contact between the mold and shell surfaces require additional assumptions about the distorted shape of the shell in the mold. These assumptions restrict the use of this code to the modelling of continuous castings, but they increase the model's capability to realistically simulate the mold-shell interactions.

The presumed distorted shape exhibits characteristics of the final slab shape, shown in Figure 1. The shell is presumed to have its maximum mold interference at the center of the face, which is reasonable considering the bulging due to the internal pressure. From this presumed shape, the following criteria were developed to decide which nodes that interfered with the mold wall were to be fixed to the mold wall position.

a) If the center of the face does not interfere with the mold, but others on the face do interfere, these nodes violate the assumption and stiff springs are applied to force the shell to the mold wall position.

b) If the center of the face interferes with the mold and it has the maximum interference across that face, only the nodes half way between the center and the point of no interference have the stiff springs applied. This assumes that those not forced to the mold will follow those that are forced because of the stiffness of the shell.

c) If the center of the face interferes with the mold but a place other than the center has an interference larger than the center, the nodes between the center and that with the largest interference are forced to the mold wall position by the spring elements. Again, this assumes that do not have the attached springs will follow the others due to shell stiffness.

Figure 9 illustrates each of these three criteria. Note that the algorithm employed to enforce these criteria does not force any node to the mold position that does not interfere with the mold. Also note that these criteria do not influence the resulting shape of the shell other than to insure that the mold limitations are maintained.

Using the physical phenomena of the shell distortions and the resulting presumed shape allows the use of at most one iteration following the initial solution that applies no mold forces to the shell. Although this method does not insure that all mold interference is eliminated, it allows little interference without overly influencing the shell's displacements. It also corrects its mistakes on the next iteration. For example, if the deformed shape at the current iteration is similar to the likely deformed shape shown in Figure 9b, only the center part of this face would be forced to the mold position. If the shell is not stiff enough to retain the remainder of the shell within the mold walls, the deformed shell on the next iteration will resemble either Figure 9a or Figure 9c depending on the amount of shrinkage occurring in that iteration. The spring loads imposed on either of these shapes will force the violating area back into the mold. Therefore, this procedure allows some error on each iteration but insures that this error is not large and that the errors will correct themselves in very few iterations; it also reduces the number of iterations and the computation time required to resolve this mechanical interface boundary condition.

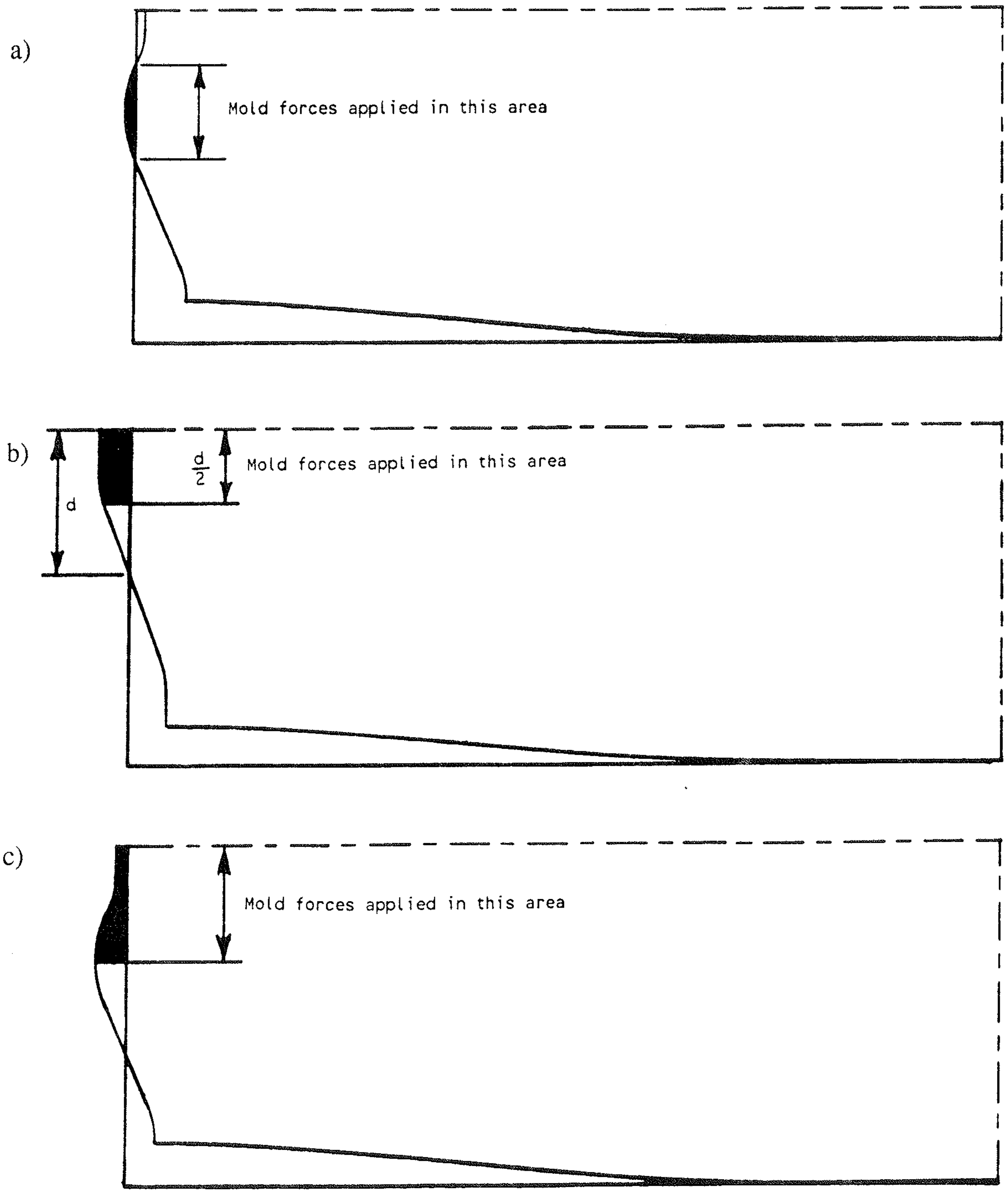

a)
Mold forces applied in this area
b)
d
$\frac{d}{2}$
Mold forces applied in this area
c)
Mold forces applied in this area

Figure 9 The mechanical interactions between the shell and the mold wall are illustrated using a overly distorted presumed shell shape model.

2.4 MECHANICAL FINITE ELEMENT MODEL VERIFICATION

The problem chosen to verify the formulation of the mechanical portion of the finite element code tests the formulation of the stiffness matrix and the thermal and plastic force vector components. Pressure, surface traction, z dependent loads, and ferrostatic pressure formulations were extensively tested on elastic problems not shown. The mechanical interface modelling section was left to be tested in modelling the continuous caster.

The thermal, plastic problem chosen for the verification of the finite element formulation was derived from an elastic solution of a thermally loaded beam presented by Boley and Weiner [18]. Along with testing the formulation, this test problem helped establish some of the convergence criteria currently used and the load step size reduction. The thermal field applied was not chosen to simulate a portion of the continuous caster, nor did the plastic strain rate function simulate the high temperature steel behavior. Thus, this problem did test the formulation, but it could have been more helpful on the convergence criteria with a more suitable problem setup.

The verification problem involved a long, thin beam, 20mm by 800mm, (0.8 inch by 3.2 inch) subjected to an instantaneous cubic temperature increase:

$$\Delta\mathbf{T} = \mathbf{100 + 3000\ y - 3.{\cdot}10^{6}\ y^{2} - 50.{\cdot}10^{6}\ y^{3}} \qquad 2.4.A$$

with y=0 at the midpoint of the beam and the temperature rise is in degree Celsius. Figure 10 illustrates a portion of the mesh used, the mechanical constraints, and the thermal loading function. The beam has an elastic modulus, **E**, of 200 GPa (29 Mpsi), a Poisson ratio, ν, of 0.25, and a thermal expansion coefficient, α, of $12.{\cdot}10^{-6}$ per degree Celsius ($5.6{\cdot}10^{-6}$ per degree Fahrenheit). The plastic strain rate was assumed to be a linear function with the Von Mises equivalent stress. The linear plastic strain rate constant, **A**, was chosen to be $1.{\cdot}10^{-13}$ $Pa^{-1}s^{-1}$. The details of this solution have been described by Haegele [7]. The stress and plastic strain history are given by the following equations.

$$\sigma_y = \sigma_z = \tau_{xy} = \mathbf{0} \qquad 2.4.B$$

$$\sigma_x = \sigma_{xo}\ \mathbf{exp[-EA(t\text{-}t_o)]} \qquad 2.4.C$$

$$\varepsilon_x^{p} = \frac{\sigma_{xo}}{E}\,\mathbf{(1\text{-}exp[-EA(t\text{-}t_o)])} \qquad 2.4.D$$

$$\varepsilon_y^{p} = \mathbf{-0.5} \cdot \varepsilon_x^{p} \qquad 2.4.E$$

The finite element code solved this transient, plastic creep problem with a 505 node mesh and 800 constant strain triangles. A constant time step of 1.0 second was used, but the plasticity

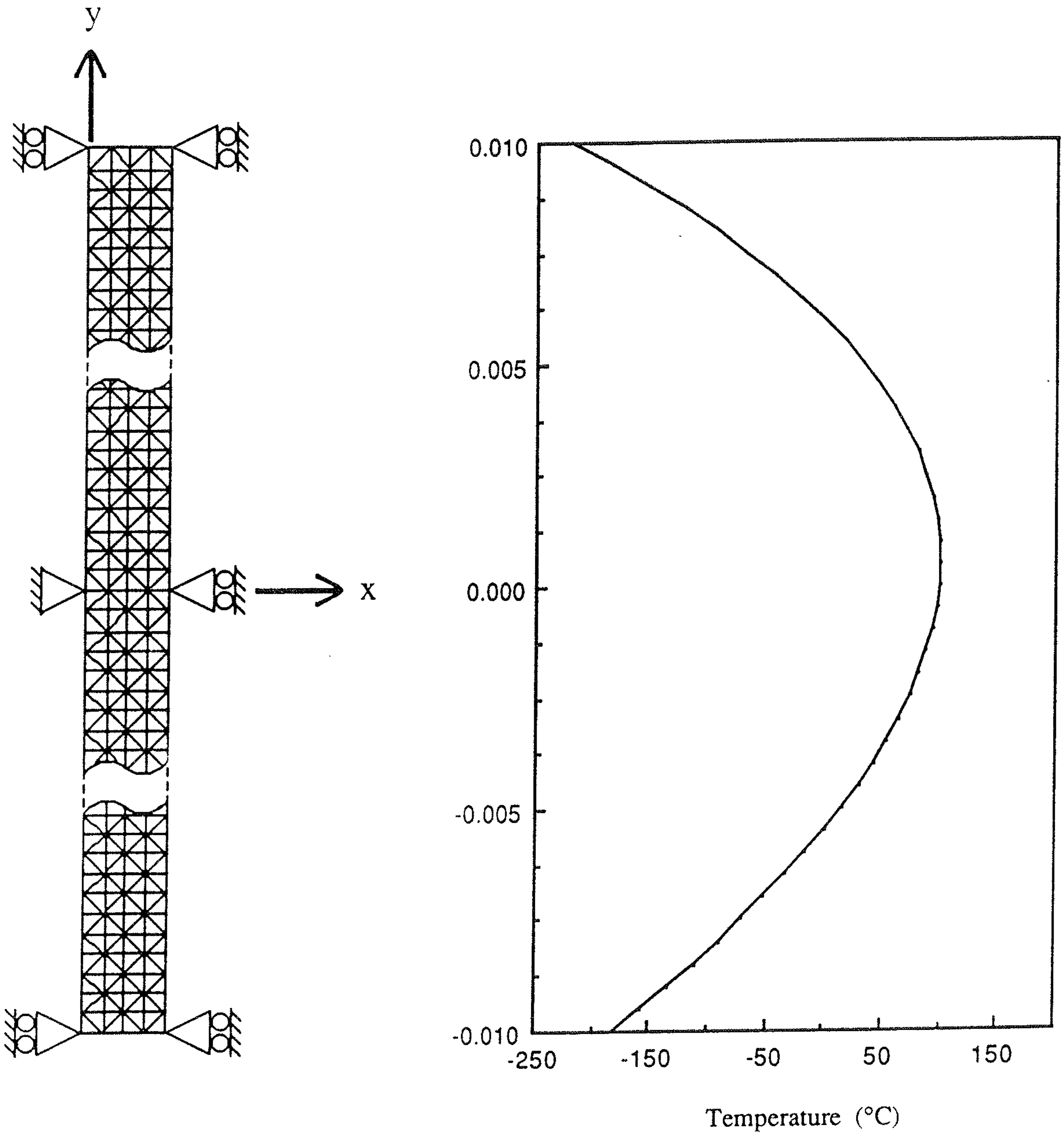


Figure 10 The boundary conditions and the thermal loading function used in the mechanical model verification problem are shown above.

sub-iteration convergence criteria cut this back to about 0.050 second. Figure 11 compares the finite element results to the analytical solution at the top, center, and bottom locations on this beam for the normal stress and normal plastic strain in the x direction. This solution illustrates that the model will accurately solve a problem dominated by thermal and plastic loads. It also illustrates that the sub-iteration convergence will force the reduction in load step size to allow the plastic behavior to be modelled accurately. However, when the loading and highly non-linear material properties of the continuous casting problem are added into the analysis, the convergence scheme fails as far too many plasticity sub-iterations are required to accurately follow the inelastic behavior of the shell. Therefore, the results to date do not account for plasticity to keep solution times reasonable.

2.5 STEP-WISE COUPLED ANALYSIS

The finite element codes, on which this model is based, were used to study the casting and thermal sequence problems occurring within cast ingots [18,19] requiring the simulation to track the material temperature and stresses over many hours and through many processes. Solidification in the mold was only a minor portion of this work, and the interfacial gap heat transfer phenomena was only a small part of the solidification. Thus, approximations on gap formation sufficed for the thermal analysis allowing the thermal and mechanical analyses to be run as two separate entities. For similar reasons, the thermal sequencing below the mold and through the processes following casting can also justify simplifying many of the details of the interfacial heat transfer [7].

However, the thermal and mechanical interactions at the mold-shell interface must be modelled carefully to understand the behavior of the shell in early mold solidification. At this interface, the heat transfer analysis requires displacements to calculate the extent of the gap, which determines the heat flux across it. The displacements are determined from the mechanical analysis, which requires the temperatures from the thermal analysis to calculate thermal loads. Three methods are available to allow both models to perform the necessary calculations based on information from the other model.

The first solution method assumes a displacement field for the entire domain of the problem. A thermal analysis can be performed on this displacement field, resulting in a temperature field which allows the mechanical model to predict displacements. These displacements can be used to update the thermal solution. Iteration proceeds between the two models after each complete solution. The initial guess is critical for this solution technique.

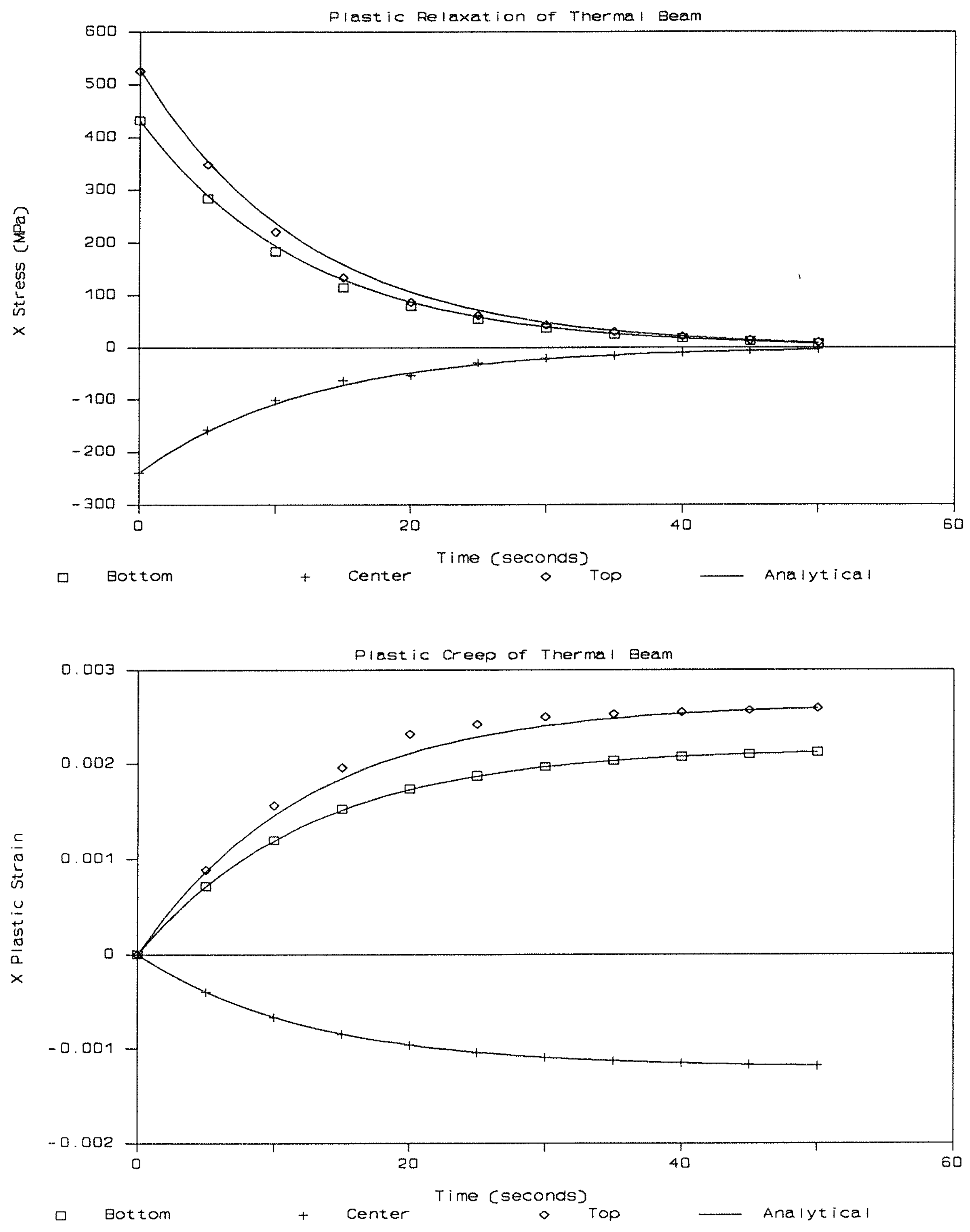


Figure 11 The iterative procedure employed in the finite element formulation follows the non-linear inelastic behavior of the thermally loaded beam.

A poor initial displacement field results in a poor thermal solution which only leads to another poor displacement field. Many iterations were required to produce a reasonable solution, or more often the solution would diverge.

The second solution technique incorporates a finite element derived for three field variables: temperature, x displacement, and y displacement. Constraint equations, added as part of the boundary conditions, relate the thermal heat flow dependence on the gap size which is determined from the displacements. No iteration is necessary to insure the heat flow is calculated with the proper gap size. Unfortunately, this solution technique increases the bandwidth and the number of degrees of freedom by 50% requiring significantly more matrix solution time. The addition of constraint equations to relate the heat flow to the displacements would further increase solution time. Rough approximations of solution time reveal that this method is beneficial provided an iterative solution cannot be solved in four iterations or fewer for every step.

The third solution technique, which is employed for this model, assumes a displacement field for the first cross-section through the mold. These displacements result in temperatures from the thermal model and new displacements from the mechanical model. The new displacements update the thermal model before the current step is completed, allowing iteration within a time step rather than after a complete solution. This final method is referred to as the step-wise coupled solution technique.

Applying the step-wise coupled solution technique to the continuous casting problem, eliminates the necessity of guessing an initial displacement condition. This first step in the caster is above the meniscus where excellent contact exists between the molten steel, powder, and mold. Therefore, the initial gap size is the minimum powder thickness. This procedures is quite stable for the solution of continuous cast slab problems. The ferrostatic pressure across the entire wide face insures most of this face will remain in close contact with the mold wall, stabilizing the majority of the heat transfer from all boundaries. Therefore, the step-wise coupled solution technique has been found to be very effective for the solution of continuously cast slab problems.

2.6 CONVERGENCE TECHNIQUES AND PROBLEMS

The highly non-linear, dominant physical phenomena in the continuous casting mold produce numerical solution problems for the mathematical model. These problems have been addressed with convergence schemes resulting in moderately acceptable computation costs associated with the mathematical solution. A more robust scheme will be needed to produce acceptable convergence on the inelastic behavior of the solid steel at casting temperatures. The scheme described below is acceptable for the verification problem discussed in Section 2.4, but produces exceedingly small step sizes under the conditions of the continuous caster. The unstable physical phenomena that require iterative techniques are the highly non-linear, inelastic behavior of the solid steel at high temperatures, the mechanical loading between two contact surfaces both moving in time, the coupling between the thermal and mechanical phenomena existing due to the shell-mold interface, and the iteration necessary to determine the z strain which minimizes the z stress under generalized plane strain conditions.

2.6.1 Convergence on Z Strain

For the generalized plane strain stress-state assumption, the normal z strain must be known when the load vector is assembled. However, the stresses calculated after solving for displacements are used to predict this strain to minimize the z stress over the entire domain. The iteration required for z strain converges very rapidly when tested under elastic loading. In test cases, the procedure usually needed the first step to arrive at a result that was within 1% of the final, correct z strain.

The minimization and convergence procedures use the z strain from the previous load step in this incremental method of solution. After solution of the displacements, stresses and strains in the x-y plane are determined, and the z stress is calculated from the elasto-plastic stress strain relations using the assumed z strain. The product of the z stress at the node and the nodal area is numerically integrated to find the average z stress. This integral is minimized by calculating the value of the z strain which would have produced an average z stress of zero. The calculated z strain is compared to the assumed z strain. If these values are within a given tolerance, recommended to be 1%, the calculated z strain is stored for the next load step and no iteration is necessary. If these values are outside the allowable tolerance, the calculated z strain replaces the assumed z strain in the recalculation of the force vector. The triangularized matrix is used again for the solution since no change is needed in the stiffness matrix. This procedure efficiently converges on the z strain satisfying the generalized plane strain stress-state assumption. However, due to the necessity of converging on other quantities, this stress-state is not used for these solutions to avoid this iteration.

2.6.2 Convergence on the Mechanical Interface Constraints

The procedure to simulate the mechanical interface, as discussed in Section 2.3.4, assumes the shell surface is free initially, evaluates this assumption, and corrects it if necessary. Correcting the surface loading for those nodes penetrating the mold wall involves applying a spring element at the penetrating node such that it is pushed inside the mold wall by an amount slightly larger than the mold wall will move by the next load step. The stiffness of the spring element is the product of an input stiffness multiplier, recommended to be 500,000, and the corresponding stiffness on the diagonal of the stiffness matrix for the interfering node. One end of the stiff spring is attached to the interfering node and the other is placed at the spring's equilibrium position inside the mold wall which is defined by

$$\Delta \mathbf{u}_n = \boldsymbol{wall} - \mathbf{u}_{n-1} + \boldsymbol{offset} \qquad 2.6.2.A$$

where ***wall*** is the mold wall position accounting for taper and mold distortion, $\mathbf{u}_{n-1}$ is the total displacement of the node through the previous iteration to insure that only the incremental displacement of the shell is forced by the spring, and ***offset*** is defined so that the equilibrium position of the spring is slightly inside the mold wall. This is determined by assuming the next time step size will be the same as the current, calculation the new wall position at the next time step, and setting ***offset*** to be twice as large as the difference between these two wall positions. This technique guarantees that convergence will not be met because a load greater than necessary to avoid penetration pushes the shell.

The finite element code violates convergence at this interface, but simulates the conditions in the continuous casting mold well. This proper simulation is guaranteed by the incremental method of the process which actually violates the convergence. The force applied to impose the additional small displacement on the shell is small compared to the thermal loads at any given step, provided the time step is small compared to the time required to travel the entire mold length. Therefore, the incremental stresses produced by this load are negligible. This procedure guarantees that if this area of the shell is prone to any shrinking it will not be forced against mold on the next load step. Therefore, the small force is not compounded by continually applying the small incremental force step after step. However, if the shell is prone to bulge under thermal or ferrostatic loading, the shell will again fall into the penetration area. Therefore, the small incremental load will again be applied. After continuing along this sequence for many loads steps, the total load that is applied to the shell surface is significant and effects the stress-state as a consistent load from the mold wall.

Therefore although the convergence within a given load step is sacrificed by this procedure, convergence over many steps is guaranteed because the mold cannot restrain the shell that is prone to shrink faster than the mold taper, and it will force the shell to never penetrate the mold wall. Because convergence within a load step is sacrificed, only one iteration is required to fix those nodes that are originally found to penetrate the mold.

2.6.3 Convergence on Interfacial Heat Transfer

The step-wise coupled solution technique, described in Section 2.5, allows for iteration between the thermal and mechanical portions of this model during every time step but does not mandate the convergence scheme used. The convergence scheme currently used for this mathematical model assumes the gap size for the thermal step is equal to the gap size from the previous thermal step. The thermal solution is derived with this gap size and the mechanical solution follows. The displacements from the mechanical solution along with the mold wall position supplied by the user are used to calculate the true gap size. If the true and assumed gaps do not agree within a given tolerance, the model calculates a new gap size and solves both analyses again replacing the assumed gap with the new gap size. The new gap size is determined from

$$(\text{gap})_{\text{new}} = \alpha(\text{gap})_{\text{true}} + (1-\alpha)(\text{gap})_{\text{assumed}} \qquad 2.6.3.\text{A}$$

where α is an acceleration constant, set to 0.4. This iteration continues until the true and assumed gap sizes agree within the tolerance limit.

2.6.4 Plasticity Convergence

The finite element formulation of simulating inelastic behavior requires that some stress-state properties be known during the formulation of the inelastic force vector. This stress-state effects the incremental, equivalent plastic strain, $\Delta\varepsilon_e^p$, in Equation 2.3.2.H. The code calculates $\Delta\varepsilon_e^p$ from the plastic strain rate function described by Azzi [6]. Plastic strain rate is dependent upon the equivalent stress, temperature, and the total plastic strain. Of these, only temperature is known when the force vector is assembled. Therefore, the values for the equivalent stress and the total plastic strain must be assumed when the force vector is assembled. The stress-state is calculated with these assumed values and the assumptions are checked against the resulting equivalent stress and total plastic strain at the end of the step. Iteration is employed to insure the assumptions used to calculate the true stress-state agree with the true stress-state within a given tolerance.

The code refers to iteration on equivalent stress and total plastic strain, which does not involve matrix retriangularization, as a convergence iteration because it is used to converge on the inelastic properties. Another iteration step for incremental effective inelastic strain, termed the sub-iteration, is employed as suggested by Mendelson [16]. The sub-iteration criterion states that the incremental, effective plastic strain should not exceed three to five percent of the total strain accumulated through the current iteration. Sub-iterations are used to reduce the size of the load step, and thus the incremental effective plastic strain, if this criterion is exceeded for any node in the finite element domain. Sub-iterations divide the thermal load step size, while a convergence iteration assumes the step size is sufficiently small for the inelastic, mechanical analysis, and the full increment in thermal load and time step is applied to the mechanical model at that step.

The sub-iteration may be triggered in two ways while the convergence iteration can only be triggered in one. First the model calculates a percent difference between the assumed effective stress and the calculated effective stress. If this percent difference is smaller than either the convergence iteration criterion, 2.5%, or the sub-iteration criterion, 10%, neither iteration is flagged. If the percent difference is larger than the convergence iteration criterion a convergence iteration is flagged, while if it is larger than the sub-iteration criterion a sub-iteration is flagged. The other way to flag the sub-iteration is by the Mendelson criterion. This compares the incremental effective plastic strain to the total accumulated strain and if the incremental effective plastic strain is larger than the specified fraction, 3%, of the total strain than the sub-iteration is flagged.

A positive check of the sub-iteration flag requires the time and load step to be cut in half, reducing the size of all incremental values. Linear interpolation of the previous and current temperatures is employed to avoid requiring the thermal solution for the sub-iteration. A full mechanical solution is required for every sub-iteration. The iterations proceed and the convergence criteria are evaluated which could again require a reduced time step size. If the sub-iteration is not required, the convergence iteration flag is checked. If it is positive, convergence iterations proceed until all criteria are met.

As the verification problem in Section 2.4 shows, this iteration technique allows material inelastic behavior to be followed accurately. Unfortunately, on the current continuous casting problem, this method attempts to divide each thermal load step into approximately 100 mechanical load steps forcing the given 600 steps to become 60,000 iterations on inelastic behavior coupled with mechanical and thermal interface iterations. The total iteration procedure is depicted in a flow chart shown in Figure 12. A more robust scheme must be developed to eliminate many of these iterations, reducing computation times, while adequately simulating the inelastic behavior of the steel.

The large run times of the code because of the iterations shown in Figure 12 prohibits the use of the entire capabilities of the code until more robust iterative schemes are developed for inelastic behavior. The results of simulations of the continuous casting process in this report eliminate the need for z strain convergence by utilizing the plane stress assumption for stress-state. The simulations also eliminate the inelasticity convergence problems by assuming the material behaves elastically. The elastic assumptions are not valid and do not allow the stresses to relax inelastically resulting in completely unreasonable stresses. However, inelastic phenomena have a much more significant effect on stresses than on strains and displacements [16]. Also, incorporating plasticity into the formulation would force the shell to follow the mold more closely due to creep deformation. This would cloud this investigation of the mold shape which would best follow the natural shrinkage of the shell, supporting the shell from large creep deformations. Therefore, the conclusions drawn from these analyses were based mostly on displacements, thermal solutions, and the change in the stresses from one simulation to another. Combining this use of elastic finite element simulation with sound engineering practice allows reasonable conclusions to be drawn on taper adjustments even though the entire process is not completely modelled.

Iteration Procedure Used in the Finite Element Model

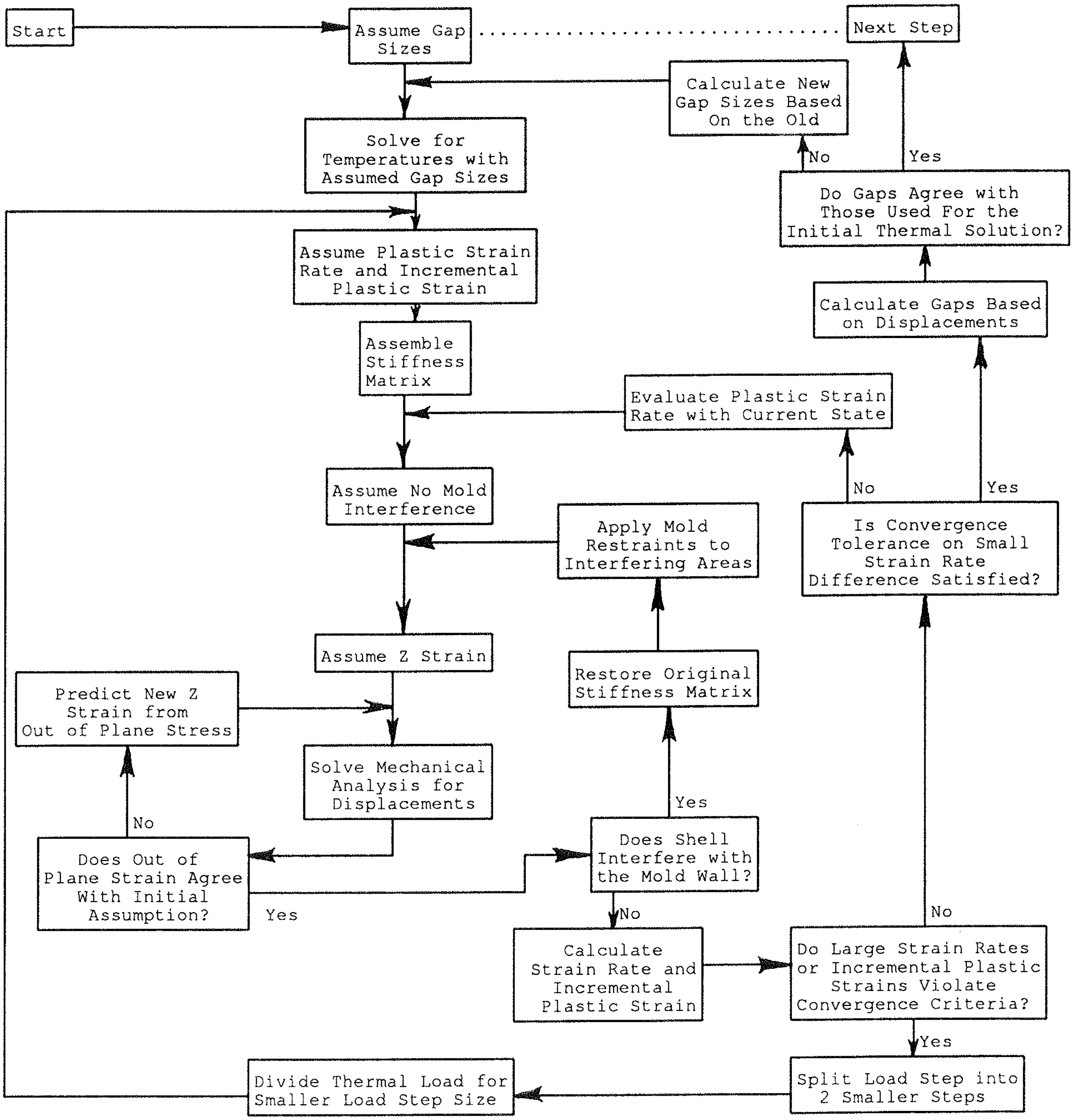

Start
Assume Gap Sizes
Next Step
Calculate New Gap Sizes Based On the Old
Solve for Temperatures with Assumed Gap Sizes
No
Yes
Do Gaps Agree with Those Used For the Initial Thermal Solution?
Assume Plastic Strain Rate and Incremental Plastic Strain
Calculate Gaps Based on Displacements
Assemble Stiffness Matrix
Evaluate Plastic Strain Rate with Current State
No
Yes
Assume No Mold Interference
Apply Mold Restraints to Interfering Areas
Is Convergence Tolerance on Small Strain Rate Difference Satisfied?
Assume Z Strain
Restore Original Stiffness Matrix
Predict New Z Strain from Out of Plane Stress
Solve Mechanical Analysis for Displacements
Yes
No
Does Out of Plane Strain Agree With Initial Assumption?
Does Shell Interfere with the Mold Wall?
Yes
No
No
Calculate Strain Rate and Incremental Plastic Strain
Do Large Strain Rates or Incremental Plastic Strains Violate Convergence Criteria?
Yes
Divide Thermal Load for Smaller Load Step Size
Split Load Step into 2 Smaller Steps

Figure 12 The iteration schemes employed by the finite element code are shown above. These schemes are required to achieve convergence on the highly non-linear phenomena of the continuous caster.

3.0 DISCUSSION OF LINEAR TAPER RESULTS

Optimization of the narrow face taper profile must begin with the study of this taper's impact on temperatures, heat flow, stress generation, and shell shape. This study must define process variabilities with linear taper changes, develop a mechanism which explains the formation of the surface shape defects, and design the mold taper to prevent or reduce this defect. For the initial study, linear tapers were varied independently on both of the faces to further the understanding of the influence of mold taper. The narrow face taper [NFT] was then modified in bi-linear and tri-linear tapers to reduce the surface depressions. The wide face taper [WFT] was also modified but restricted to remain linear. This linear restriction allows easier taper machining and production use.

The results presented in this report are based on the simulations of the casting conditions given in Tables 3.1 and 3.2 on the next page. These runs simulate a slab measuring 0.203 meter (8 inches) thick, the dimension along the narrow face, and 0.914 meter (36 inches) wide, the dimension along the wide face. The mold length is 0.730 meters (28.7 inches) along an arc with a radius of curvature of 10 meters (394 inches) but the molten steel level is 80 millimeters (3.1 inches) below the top of the mold, so the total distance travelled through the mold is 0.650 meters (25.6 inches). The pouring temperature of 1485°C (2700°F) and the casting velocity of 0.015 meter per second (35.4 inches per minute) were chosen to match typical operating conditions for austenitic stainless steel casting. The tapers used during a typical casting process are 0.7% per mold on the narrow face and 0.5% per mold on the wide face; these values were used to match the heat flow parameters which were measured with these tapers by Samarasekera [22]. The resultant heat flow as a function of distance along the mold is shown in Figure 13. This figure compares the predicted heat flow from the perimeter of the mold to that measured from experiments. The agreement between the experiment and the mathematical model indicates that the interfacial gap heat transfer algorithm provides enough flexibility to simulate the heat transfer phenomena at this interface.

As shown in Table 3.1, the elastic plane stress assumptions are employed for these simulations. The plane stress assumptions eliminate the need to iterate on the out of plane strain to minimize the average out of plane stress. The need to iterate on the non-linear inelastic behavior of the material is eliminated through the use of the elastic assumption. Eliminating these iteration procedures produces reasonable solution time for each simulation. More importantly, however, allowing the plastic flow of the material would allow the shell to

TABLE 3.1 - SIMULATION CONDITIONS

Mold Dimensions:		
Slab width	0.914 m	36.0 inch
Slab thickness	0.203 m	8.0 inch
Mold Length	0.700 m	27.6 inch
Initial Linear Taper	0.7% per mold on Narrow Face	
	0.5% per mold on Wide Face	
Material Properties:		
Grade of Steel	AISI-304 Stainless Steel	
Composition	18% Cr, 8% Ni,	0.06% C
Phase	100% Austenite	
Liquidus Temperature	1454°C	2650°F
Solidus Temperature	1399°C	2550°F
Casting Conditions:		
Initial Temperature	1485°C	2705°F
Casting Speed	0.015 m/sec	0.59 inch/sec
Meniscus Level Below Top of Mold	0.050 m	2.0 inch
Stress State:	Elastic Plane Stress	

TABLE 3.2 - TAPERS SIMULATED

Run #	Narrow Face Taper (%/mold)	Wide Face Taper (%/mold)
0	Variable (Match Shrinkage)	0.5%
1	0.7%	0.5%
2	0.5%	0.5%
3	0.3%	0.5%
4	0.9%	0.5%
5	1.1%	0.5%
6	1.3%	0.5%
7	0.7%	0.3%
8	0.7%	0.1%
9	0.7%	0.7%
10	1.3%	0.7%
11	0.3%	0.1%
12	1.3%(.25);0.30%(.65)	0.7%
13	1.4%(.16);0.75%(.35);0.40%(.65)	0.6%
14	1.5%(.18);0.75%(.40);0.40%(.65)	0.7%
15	1.5%(.17);0.75%(.40);0.40%(.65)*	0.8%

* Notation indicates that a three-fold taper was used. The first linear portion had a taper of 1.5%/mold and occurred from the meniscus to 0.17m below the meniscus. The second portion began at this point, had a taper of 0.75%/mold and extended to 0.40m below the meniscus. This final portion of this taper began where the second finished, had a taper of 0.40%/mold and extended through the end of the mold, 0.65m below the meniscus.

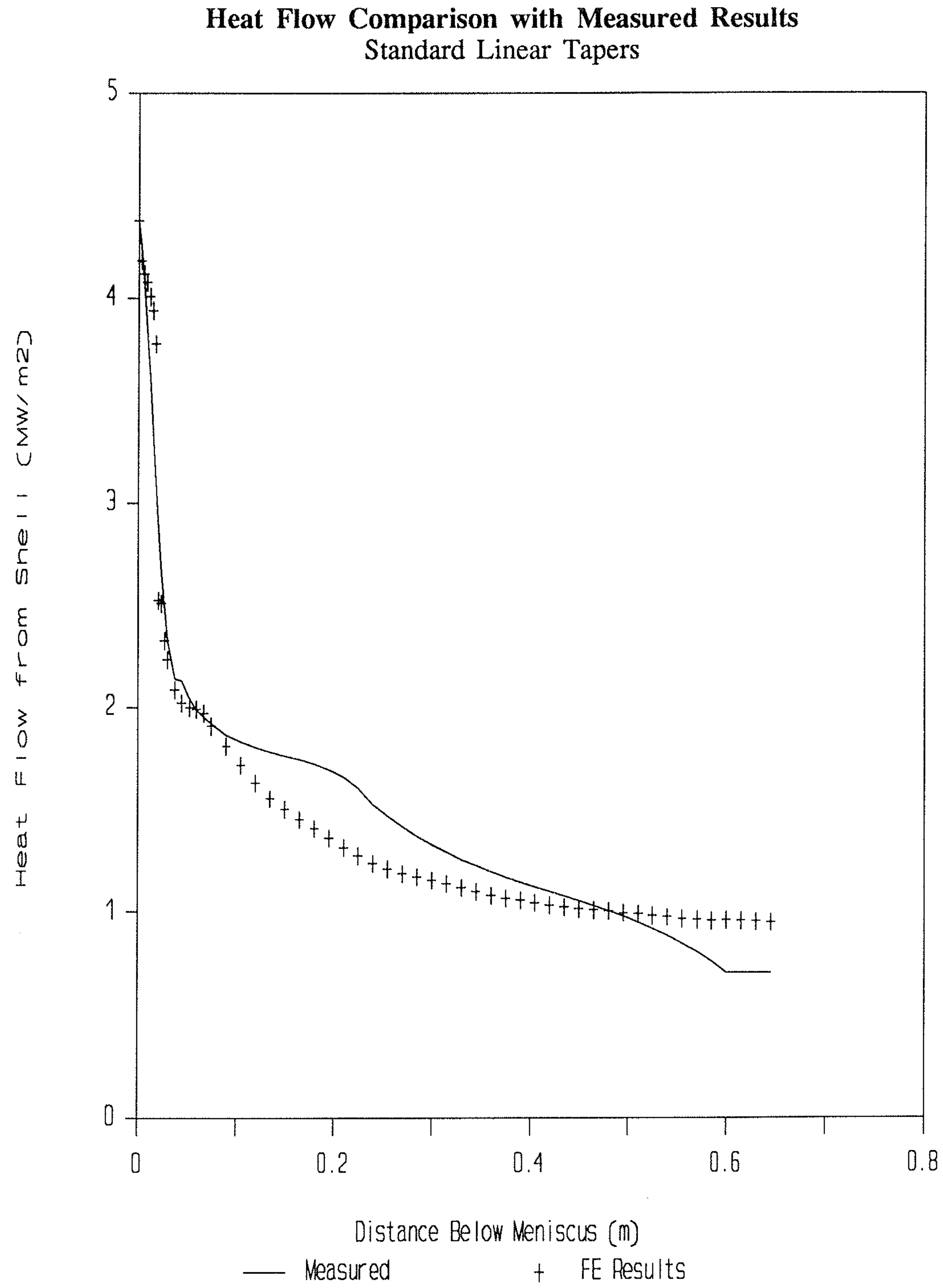


Figure 13 The agreement between the mathematical results and the experimental measurements [22] indicates that a suitable model has been developed for the interfacial gap heat transfer.

more closely conform to the shape of the mold. However, this investigation intends to design the mold taper so that it would closely follow the shrinkage of the shell without the need for the plastic deformations, producing a better quality casting because of the lower residual plastic strains accumulated in the raw casting. Therefore, the inclusion of the high temperature plastic behavior of steel is required to fully verify the formulation and operation of this model. However, the elastic stress state produces more reasonable run times and better suits the needs of this investigation, and therefore it is applied for these simulations.

The elastic assumption employed in the present investigation has several important implications on the results. The first, and probably the most obvious effect of the elastic restriction, is that without creep relaxation the stresses calculated by the finite element model are considerably higher than actual operating stresses. Stresses exceeding 800MPa (115ksi) have been calculated when large tapers compress the shell's wide face. Stainless steel cannot withstand these stresses at temperatures encountered during casting indicating some inelastic deformation would occur. Secondly, deflections, gap dimensions, heat flow rates, and temperatures will also likely be effected by the inclusion of the steel's inelastic behavior.

The effect of plasticity on gap formation, displacements, and temperatures is considerably smaller than the effect on stress, since shell behavior is dominated by thermal strains. Mechanical loads are significant only when the mold wall imparts a force to the shell, but at these points the displacements and gaps and therefore temperatures are determined from the known mold wall position. Therefore, the gaps and temperatures should be reasonably accurate throughout the entire simulation dominated by both thermal and mold restraint loads. The general trends predicted by the model should all remain valid with both elastic and inelastic material behavior. Furthermore, since mold optimization is based on displacement calculations, and *minimizing* the stresses, the improved taper designs should be reasonable when inelastic effects can be modelled more easily. Finally, including inelastic effects in future runs can validate these results as well as the procedure of analyzing the mold taper elastically. If the elastic assumptions lead the design toward the proper optimum, it would then allow quicker, easier simulations to be run, with only verification of the results including the more complex plasticity equations.

3.1 TAPER EFFECTS ON THE HEAT TRANSFER FROM THE SHELL

The total average heat flux extracted from the perimeter of the shell is not greatly affected by large changes in taper, as shown in Figure 14. This figure compares the global heat transfer along the axis of the caster for the standard taper [WFT=0.5%, NFT=0.7%] with that from a low taper run [WFT=0.1%, NFT=0.3%] and that from a high taper run [WFT=0.7%, NFT=1.3%]. Only a slight increase in total average heat flux is observed with these dramatic increases in taper. Therefore, the total average heat flux is relatively insensitive to changes in taper.

The insensitivity to taper indicates that most of the perimeter of the shell behaves the same for all tapers. The results of all the simulations consistently indicate that the majority of the wide face maintains close contact with the mold wall throughout the mold. This face is exposed to high ferrostatic pressure loading due to its high relative surface area, compared to the shell thickness and resulting bending strength. This pressure forces the majority of the wide face to conform with the mold wall, so the shell transfers heat uniformly, regardless of the taper. Because of its large surface area and the high heat transfer rates, the portion of the wide face in constant contact with the wall dominates the heat transfer around the mold. Thus, the total average heat flow was relatively constant down the mold for the complete range of tapers investigated.

The ferrostatic pressure maintains close contact along the majority of the wide face, therefore those remaining areas must be responsible for the slight differences in heat flow with different tapers. Therefore, the gap sizes in the off-corner and corner regions of the wide face as well as across the entire narrow face must change with the various tapers. These results also indicate that the gaps change as intuitively anticipated. For the higher tapers, the gaps along the faces would be smaller and therefore more heat would flow to the mold coolant passages. This also suggests that the behavior of these regions that create and manifest the surface shape defects can be modified by changes in the mold taper design.

The majority of the wide face maintains as close a contact with the shell as the mold flux viscosity and surface roughness will allow. However, the corner region and the narrow face can shrink away from the mold walls, creating a gap. The width of this gap significantly impacts the heat transfer across it, as indicated in Figure 15. This figure tracks both the temperature of a node in the off-corner region of the narrow face and the corresponding gap size for a run with WFT = 0.5% and NFT = 1.3%. The results clearly illustrate the heat

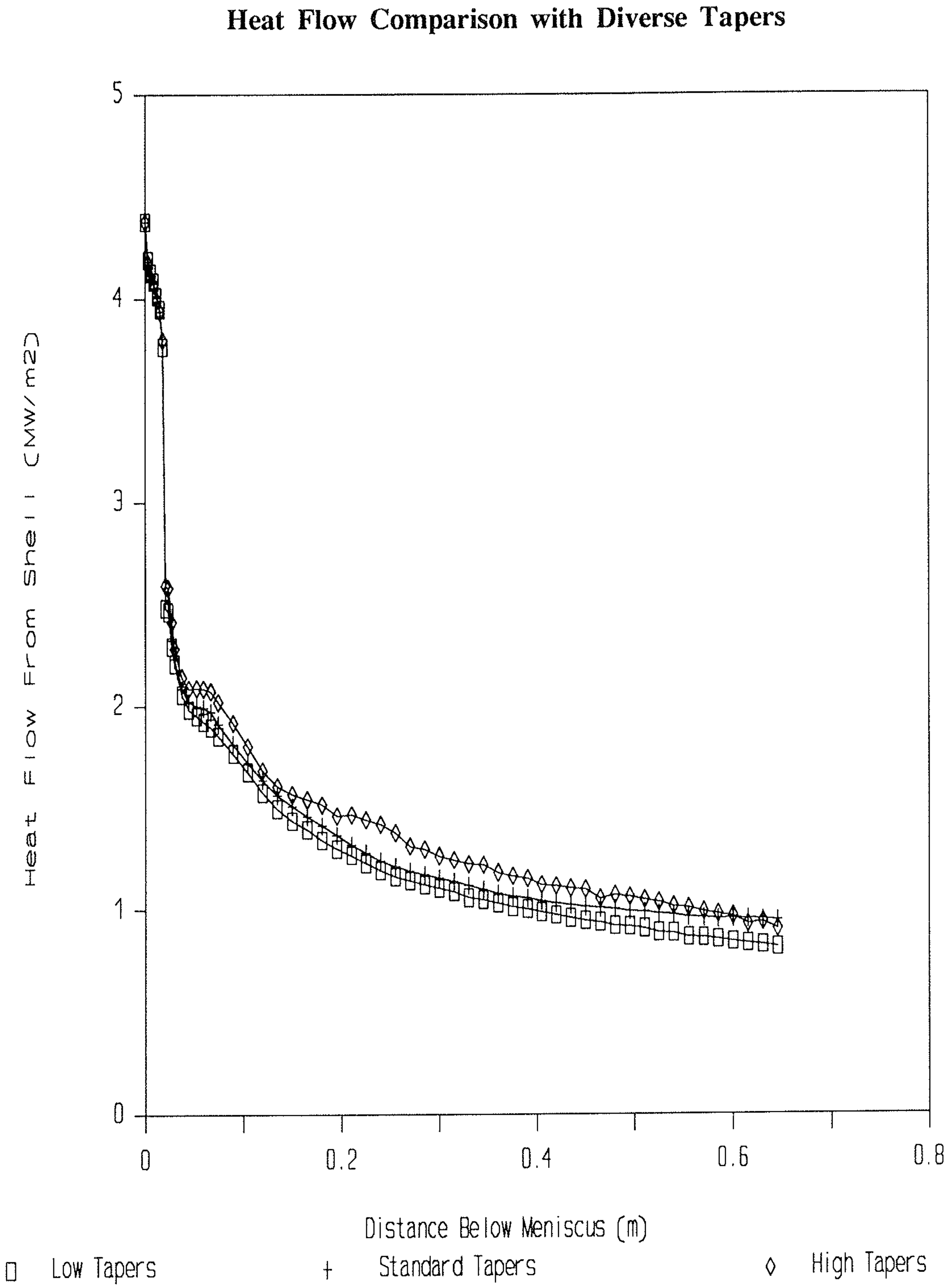
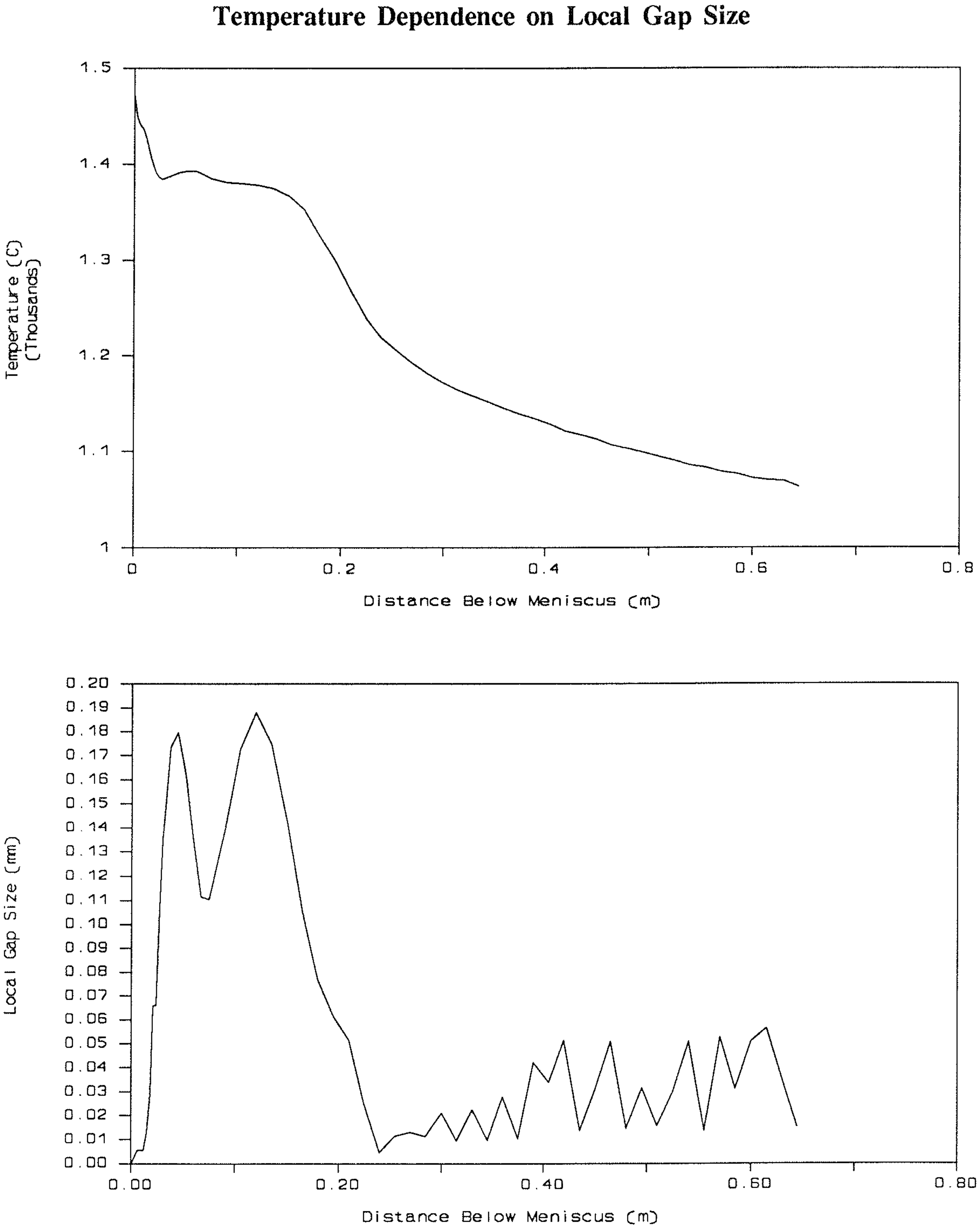


Figure 14 The small change in heat flow with extreme tapers indicates that the predominant heat flow is across the wide face which remains in contact with even the smallest linear taper.



Temperature Dependence on Local Gap Size

Figure 15 The direct correspondence between local gap size and shell surface temperature illustrates the important influence of the gap size on heat transfer.

transfer and surface temperature dependence on gap size. Initially, the gap is very small, and a rapid decrease in temperature of the off-corner node is observed. The decrease continues until the shell is just below the solidus temperature (1399°C, 2550°F). As the shell solidifies from 0 to 50mm (0 to 2 inches) below the meniscus, the gap grows from 0.02 to 0.2mm (0.0008 to 0.008 inch). At a gap size of roughly 0.2mm, the heat that conducts into the shell approximately equals the heat that leaves the shell. This holds the surface of the shell at a constant temperature as seen in Figure 15. The off-corner region remains at a constant temperature while the temperature of the corner and the middle of the narrow face continue to fall resulting in a local hot spot in the off-corner region. Eventually, the rate of mold wall movement due to the taper exceeds the rate of shrinkage of the shell which closes the gap between 150 and 250mm (6 to 10 inches) below the meniscus. Although somewhat erratic, the gap size below 250mm (9.8 inches) below the meniscus is nearly five times smaller than its peak value higher in the caster. The reduction in gap width causes the off-corner temperature to drop from 1380°C to 1200°C (2540°F to 2190°F) over this 100mm (4 inch) period, which is equivalent to six seconds at this casting velocity. The cooling rate diminishes slowly throughout the rest of the mold, due to both the thickening of the shell and a smaller temperature difference driving the heat transfer.

The above account, which is very dramatic due to the high taper but noticed on most other runs as well, illustrates the strong correspondence between the temperature history and gap thickness. The narrow face taper significantly controls the gap sizes on this face, as shown in Figure 16. These results indicate that the taper design can be used to reduce locally low heat transfer zones and local hot spot formation. Examination of the simulation results for other tapers indicates that these phenomena are common to all taper designs. Diminished local heat transfer rates, increasing gap sizes, higher local temperatures or hot spots, and a locally thinner shell are all expected whenever and wherever the taper design allows gap sizes to exceed approximately 0.1mm (0.005 inch).

3.2 INFLUENCE OF NARROW FACE TAPER

The first set of simulations performed with the model investigated the effects of holding the wide face taper constant at the standard value of 0.5% per mold while the narrow face taper varied from 0.3% to 1.3% per mold. The shell behavior was monitored throughout the mold and differences between the different runs provide insight into the effect of narrow face taper. The following observations are direct results of these simulations.

Narrow Face Profile Comparison with Increasing Narrow Face Taper

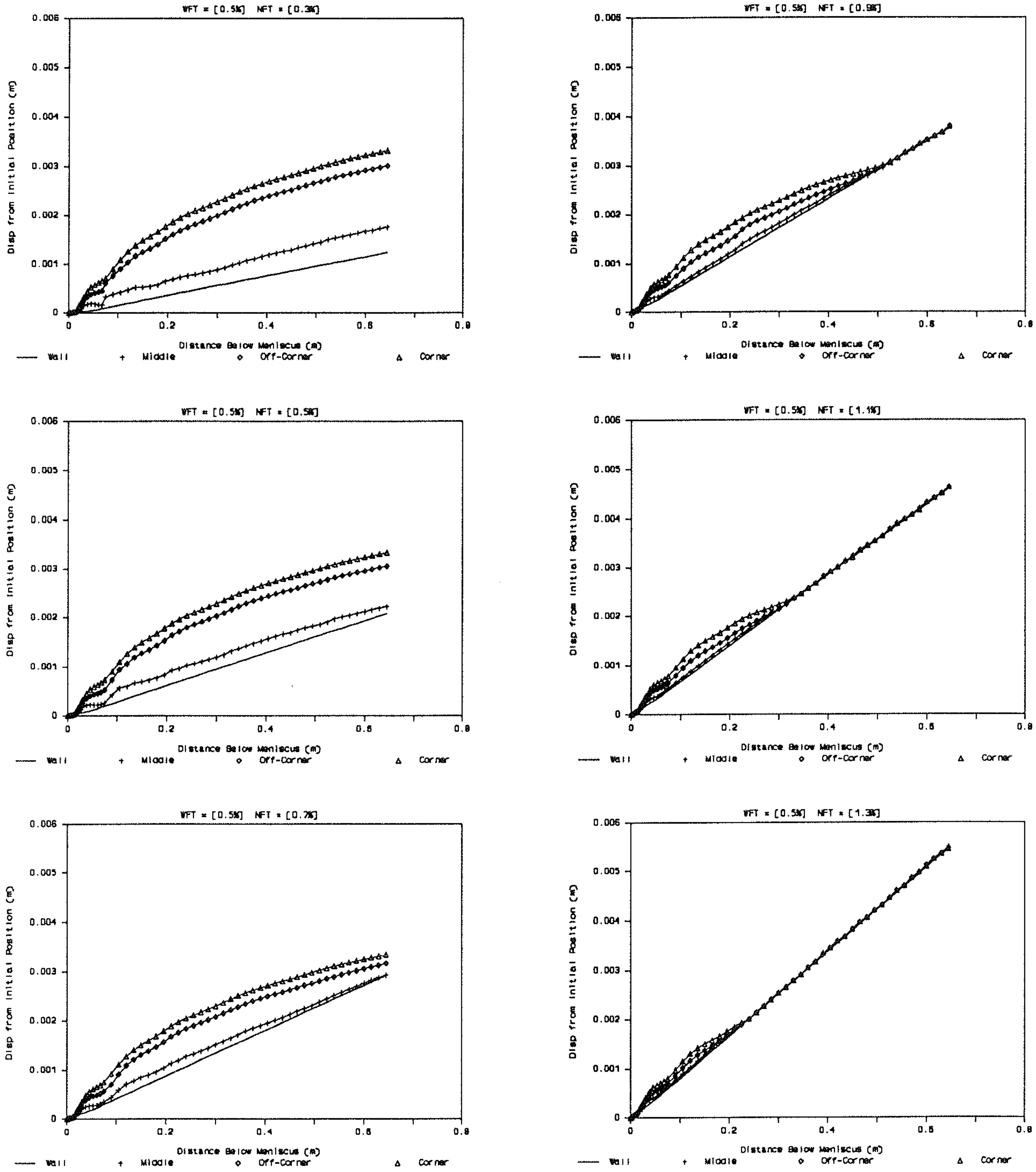


Figure 16 The gaps along the narrow face shell of the mold close earlier with increasing taper increasing the heat flow from the narrow face, supporting the bulging, and applying crushing loads to the wide face.

3.2.1 The Effect on Gap Size

The most striking results from the narrow face taper runs are shown in Figure 16. This figure reveals how the shrinkage of the narrow face compares to the wall position as the casting solidifies. The difference between the wall position and the node position corresponds to the gap size. This figure shows that for tapers below 0.7%, gaps exist across the entire narrow face and down the entire length of the mold, indicating that the narrow face is completely unsupported within the mold. These gaps increase dramatically with decreasing taper. Even with the 0.7% taper, the gap at the middle of the narrow face is just barely closed by mold exit.

For tapers of 0.9% per mold or more, no gaps exist along any portion of the narrow face at mold exit. However, for the upper portion of a mold with a 0.9% per mold taper, significant gaps do form across the entire width of the narrow face. This allows significant thinning of the shell and indicates insufficient taper in this part of the mold. These gaps are closed completely lower in the mold, indicating compression of the shell by the narrow face mold wall due to excessive taper in the bottom quarter of the mold. For tapers above 0.9%, the gaps still form high in the mold but disappear very quickly, as the shell and narrow face wall remain in complete contact for the majority of the time in the mold. With tapers of 1.1% or more, the shell in the middle of the narrow face contacts the mold wall down its entire length. These results show that use of a linear taper cannot match the natural shrinkage of the shell in both the upper and lower parts of the continuous casting mold. The linear tapers result in a combination of insufficient taper and excessive taper with shell compression at some point in the mold.

3.2.2 The Effect on Temperature

Figure 17 presents the effect of narrow face taper on the temperature histories on the surface of the shell. Not surprisingly, the smaller taper runs that fail to close up the gap exhibit very high surface temperatures. In fact, many nodes on the narrow face are calculated to be just below the solidus temperature by the time of mold exit. This indicates a serious problem removing heat from the narrow face. Even for the 0.7% taper, the gaps in the off-corner region are not closed by mold exit, and temperatures here are very high. The high narrow face temperatures correspond to thinner shells. This makes the shell more susceptible to bulging under ferrostatic pressure, both in and below the mold. This is perhaps the most serious cause of gutter formation.

Surface Temperature Comparison with Increasing Narrow Face Taper

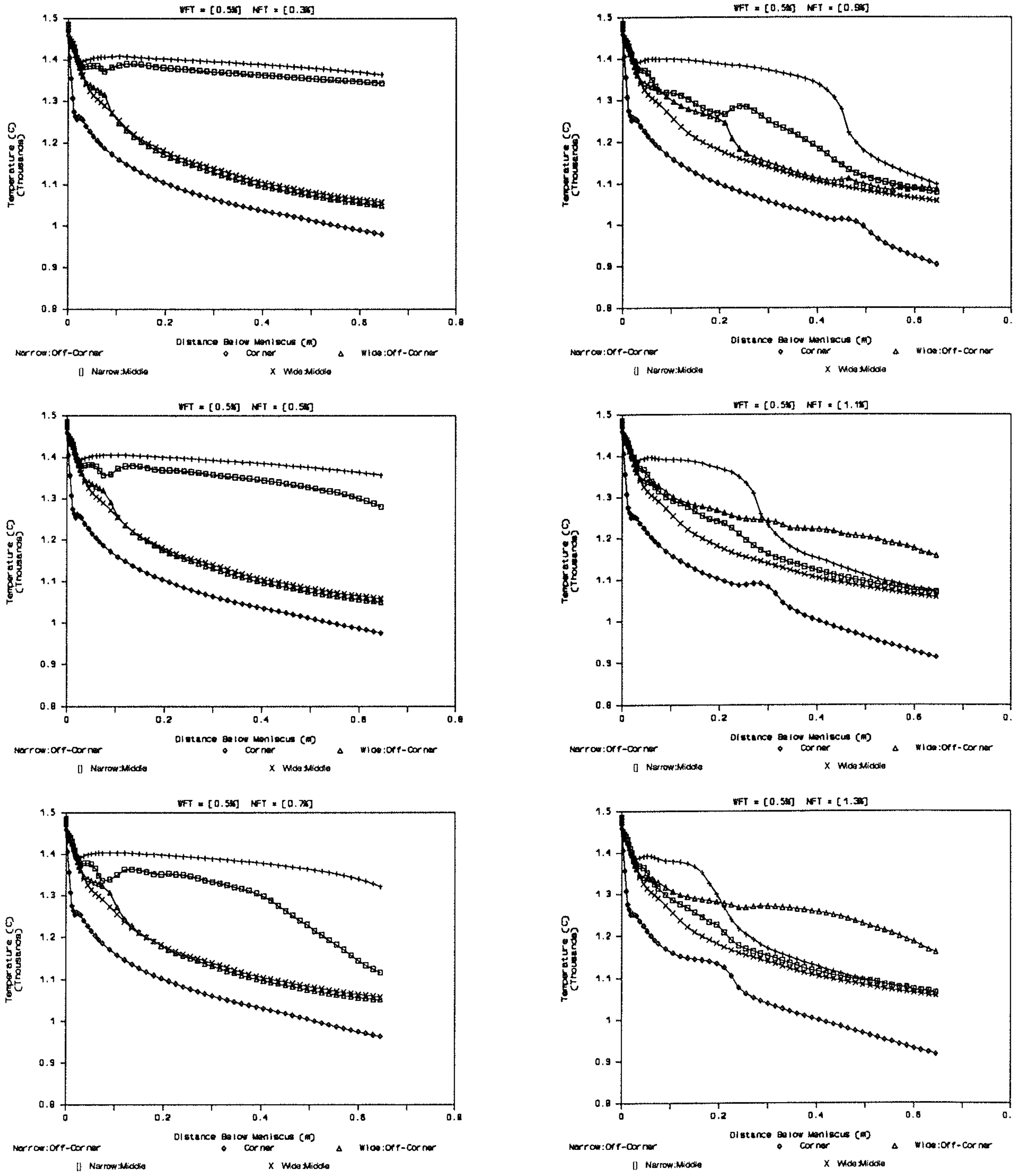


Figure 17 The temperature histories of the shell surface indicate that narrow face tapers greater than 0.7% cause a dramatic reduction in temperature along the entire narrow face.

The temperatures predicted at mold exit are likely higher than what would be seen in the actual caster. This is predominantly due to the exclusion of plasticity in these simulations. Creep is significant at the temperature and stress levels observed, so the bulging that occurs with all the above tapers would tend to be much greater. This bulging would decrease the gap thickness increasing the heat transfer. The effect would be most dramatic in the runs with the smallest tapers where the gaps and temperatures are largest and greater bulging would have the most significant effect on reducing gaps and temperatures and increasing shell thickness.

The temperature histories for the runs with large wide face taper exhibit similar trends as those discussed above with the exception that the gaps close earlier in the mold, the rapid decrease in the temperature starts sooner and continues longer through the mold. This allows the temperatures to reach much lower values by mold exit. A large taper can be beneficial, particularly high in the mold because:

1) it closes the gaps along the narrow face,
2) it promotes a more uniform temperature distribution across and along the narrow face reducing local off-corner hot spots,
3) it increases heat removal, producing a thicker narrow face by mold exit reducing the chance of breakout due to overly thin shell walls, and
4) it reduces the extent of narrow face bulging due to ferrostatic pressure, corner rotation, and gutter formation occurring both in and below the mold.

A larger taper can be detrimental lower in the mold because the rate of change of the mold wall position due to the taper exceeds the rate of change of the shell position due to shrinkage. The relative motion of the wall toward the shell closes the gap. After the taper has completely closed the gaps, the mold actually pushes against the solidifying shell causing compressive stresses to build up in the shell along the wide face and eventually causing buckling of this face.

3.2.3 The Effect on Stress

The stresses along the wide face for the various linear narrow face tapers are compared in Figure 18. This allows the assessment of possible problems associated with too much taper that pushes on the shell. The runs with small tapers (0.7% or less) do not close the gaps at all so do not apply compressive mechanical loads due to contact between the narrow face mold wall and the shell. The stresses produced along the wide face are due to changing the thermal gradients in the shell and the pressure loading from the molten core.

Wide Face Compression Due to Increasing Narrow Face Taper

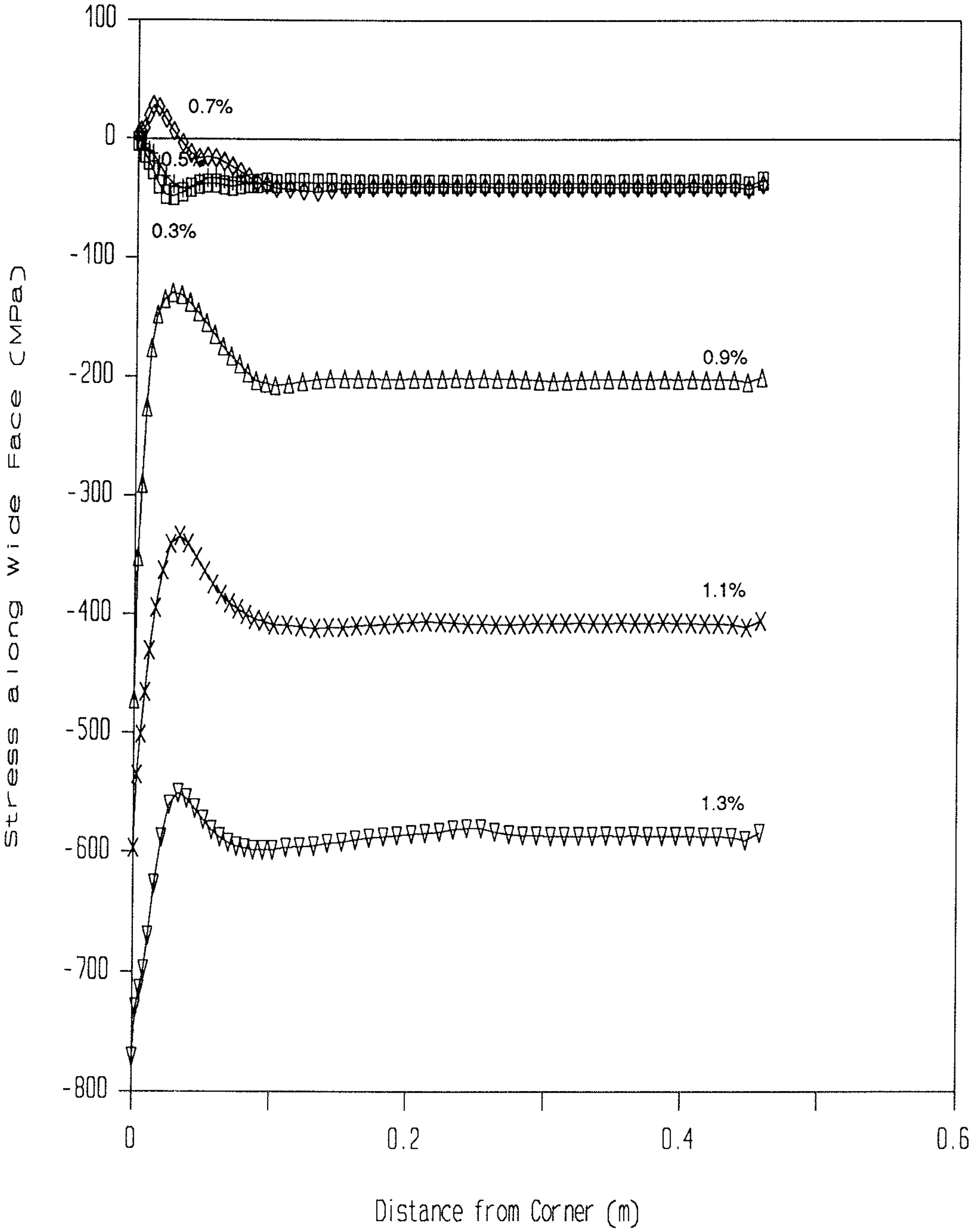


Figure 18 The larger linear tapers compress the wide face shell as the rate of change of the taper exceeds the shrinkage rate. This produces increasing compression in this shell with increasing taper.

Figure 18 shows that the stresses developed in these cases are slightly compressive, on the order of 40 MPa (6 ksi).

The compressive surface stresses at mold exit can be explained by differential thermal expansion. Pressure loads from the molten pool force the shell out toward the mold wall which would produce slightly tensile hoop stresses at the shell surface. This indicates that the compressive stresses are produced by the changing thermal gradients and must be larger than the tensile stresses due to the pressure. At the beginning of solidification high in the mold, the thin shell is relatively free to contract around the liquid core, which offers little resistance. Only slight tensile stresses are generated at the surface. As the shell thickens while moving down through the mold, the surface cools more slowly than the interior portion of the shell, thus causing the interior of the shell to shrink more than the surface. By the time of exit from the mold, the differential shrinking forces the surface layer into compression setting up complimentary tensile stresses just beneath the surface. Thus, the thermal effects on the shell produce a compressive stress in the surface of the shell. This behavior is essentially independent of narrow face taper provided mold wall restraints are minimal.

Compressive stresses along the wide face exceeding 40 MPa (6 ksi) indicates that the narrow face taper at mold exit is too large and that the end walls are actually compressing the shell to force it to conform. Figure 18 clearly indicates that this occurs at tapers exceeding 0.7% and a drastic increase in compressive stresses is observed with increasing taper.

It is interesting to note that the stress field in the off-corner region with the 0.7% taper is slightly tensile, contrasting with both the smaller and larger tapers. This can be explained by the closing of the gap at the middle of the narrow face, which just barely occurs by mold exit for the 0.7% taper case, as seen in Figure 16. The slight pushing of the mold against the middle of the narrow face bends the shell about the corner generating tensile stresses in the off-corner surface of the wide face due to the moment about the corner. This moment is made possible by the much thicker shell compared with that of the smaller tapers. It does not supply enough load through the shell to produce any compression along the wide face shell, however. It can also be noticed that this mechanism produces more tensile stress in the off-corner region of all tapers that contact the narrow face by mold exit. However, higher tapers apply significant compression along the wide face shell superimposing compressive stress throughout the length of the shell on the tensile off-corner region. This large compression tends to mask the bending mechanism which produces a more tensile stress in the off-corner region.

Tapers above 0.7% generate increasingly larger compressive stresses within the wide face shell. This compressive stress increases to 800MPa for the 1.3% per mold taper. These stresses are excessively large due to the absence of plastic relaxation. However, they clearly indicate that at high narrow face tapers, the shell along the wide face is subjected to a significant compressive load. This load is large enough to force the shell to yield, flow plastically, and even buckle.

The results discussed above indicate that problems could occur in the shell by nonuniform shell thickness (local thin spots) or through compressive loads due to excessive taper. Figure 19 puts these problems into perspective by comparing each taper design to the ultimate problem to be solved, the gutter formation. This plot shows the wide face shape at mold exit for the mold designs with variable narrow face taper.

3.2.4 The Effect on Slab Shape

The compression of the wide face, produced by the mold wall contact, has an important effect on enhancement of the existing gutter observed even with the elastic model. Figure 19 illustrates this by comparing the displaced shape of the wide face of the shell at mold exit produced with different narrow face tapers. This figure shows that for large tapers, exceeding 0.9%, the narrow face compresses the wide face shell so much that it buckles that shell inward in the off-corner region. Figure 19 shows that for a narrow face taper of 1.3%, (and to a lesser extent for a 1.1% taper as well), this produces a shallow off-corner depression between 50 and 250 millimeters (2 to 10 inches) from the corner. It is remarkable to note that this gutter has the same general location and shape as the gutter observed in the final slabs. However, its maximum depth of only 0.05mm (0.002 inch) is much smaller than that observed in the final slab.

The model's underestimation of the actual gap depth could be explained by the following. First, the elastic assumptions imposed on the material behavior limits deflections produced by a given load because creep is not accounted for. Second, the model does not model the buckling behavior of the shell. The displaced shape shown in Figure 19 is not used to calculate the direction of the load in the shell; instead, the original shape is used therefore requiring all loads at the surface of the shell to act directly along the wide face. If the buckled shape and rotation loads associated with this shape were used to calculate the direction

Wide Face Shape Due to Increasing Narrow Face Taper

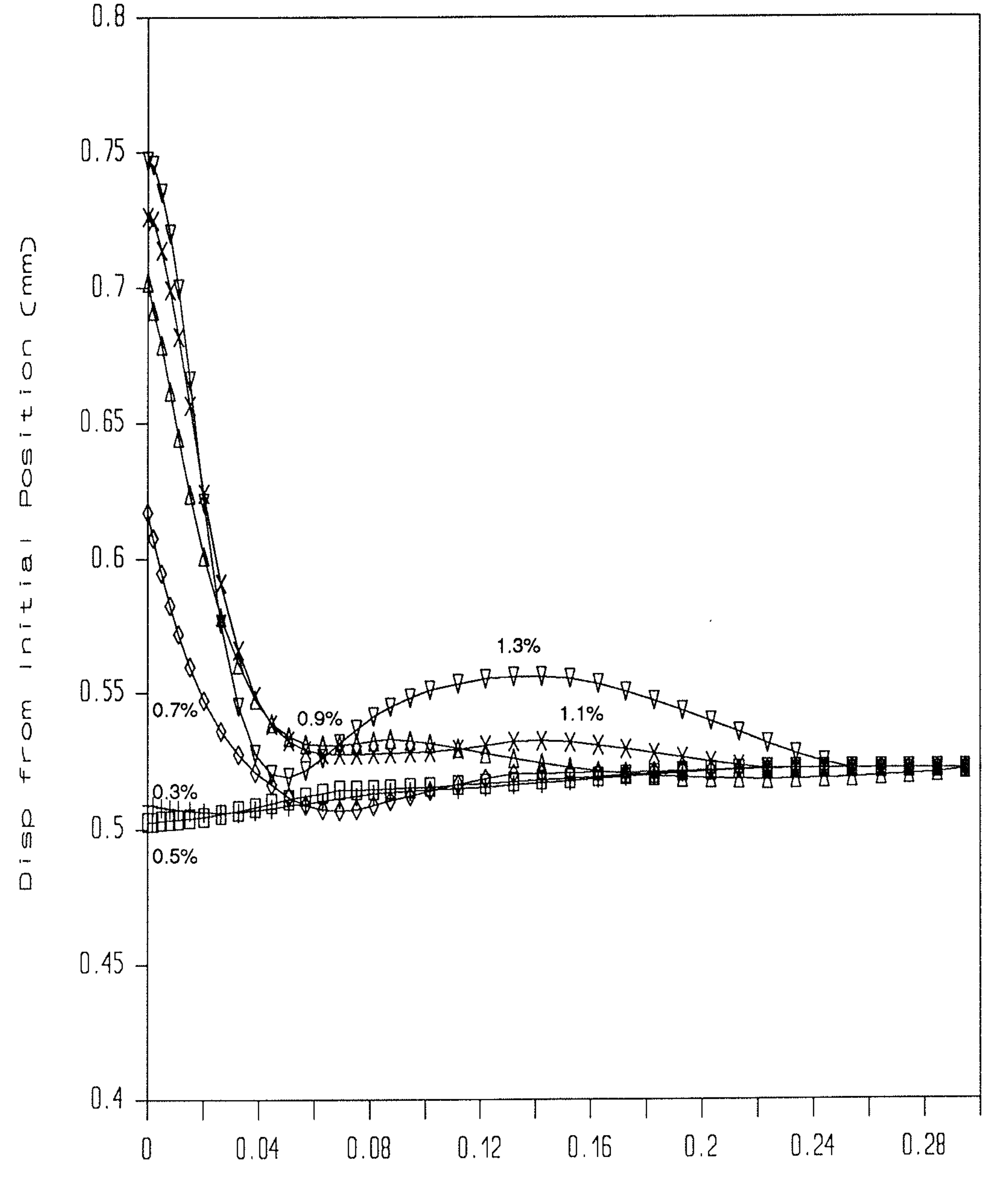


Figure 19 The increasing linear narrow face tapers increase the tendency to produce off-corner surface depressions.

of the next load, the buckled shape would be much more dramatic. Third, the depth of the gutter increases just below the mold. The thinner weaker shell produced in the off-corner region would make it more susceptible to further corner rotation due to bulging of the unsupported narrow face below the mold. Finally, further growth of the gutter is possible as the slab moves down through the spray zone. Cooling effects such as the collection of spray cooling water running down the slight gutter that has already formed and further solidification and shrinkage of the shell could play a role. All of these effects will tend to increase the depth of the gutter through the spray zone region after it has initiated in the mold.

Some indication of the tendency to form an off-corner gutter can be obtained by monitoring the off-corner position relative to the corner on a wide face profile plot. Examination of the results in Figure 20 in this manner reveals that all the linear narrow face tapers do have the tendency to produce an off-corner gutter on the wide face at some stage in the mold. This is due to the bulging and hinging phenomena which occurs early in the solidification of the shell, originally proposed by Brimacombe et al [23] and discussed more in the mechanism section.

It is significant that the mathematical model is able to predict and calculate these phenomena, including gap formation, shell thinning, corner rotation, shell compression, and buckling. This provides a useful tool to use in studying shell behavior and mold taper design. The investigation of changes in the linear narrow face taper has revealed some design guidelines for mold taper optimization:

1) The gaps on the narrow face must remain smaller than 0.1 millimeter (0.004 inch) to avoid shell reheating and local hot spots.
2) The off-corner region of the wide face must not shrink away from the wall quicker than the corner to avoid off-corner gutter initiation.
3) Compression along either shell must be avoided.

3.3 INFLUENCE OF WIDE FACE TAPER

The importance of wide face taper was investigated in several simulations in a similar manner to the narrow face investigation. The wide face taper was varied from 0.1% to 0.7% per mold, while keeping the narrow face constant at 0.7% per mold. These results indicate that the shell is much less sensitive to wide face taper than to the narrow face taper.

As discussed previously, the reduced sensitivity occurs because ferrostatic pressure always forces the majority of the long, flexible wide face of the shell to contact the mold. Thus, the

Wide Face Profile Comparison with Increasing Narrow Face Taper

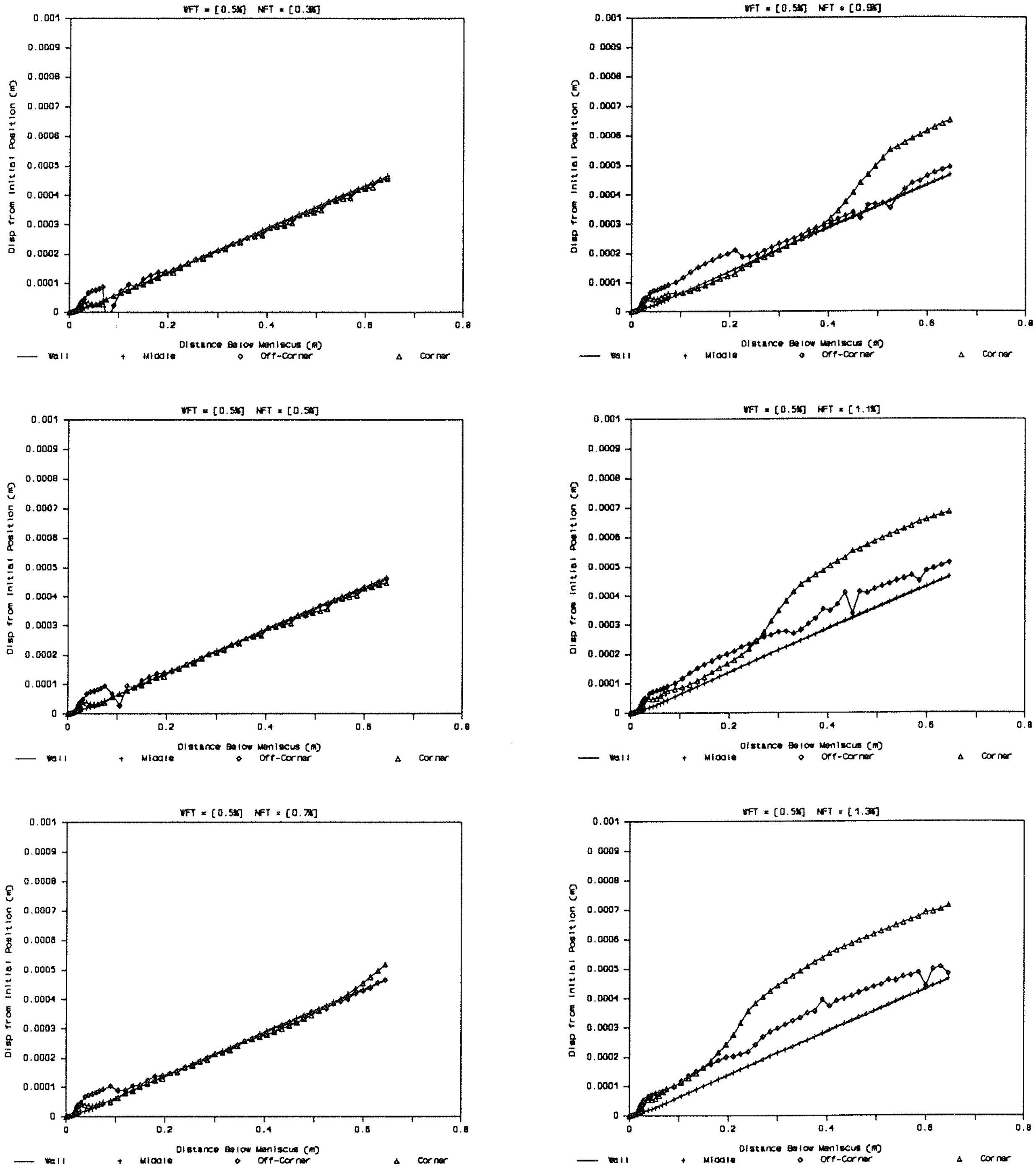


Figure 20 The tendency to form off-corner surface depressions is exhibited when the displacement of the off-corner region relative to the wall is greater than this relative displacement for the corner.

gaps, temperatures, and shell thicknesses remain relatively constant over the wide face, with the possible exception of the corner. Here, trends similar to the narrow face are observed. Increasing wide face tapers reduce formation of the off-corner gap on the wide face, with a corresponding decrease in the off-corner hot spot and shell thinning. However, very large tapers can generate compressive stresses in the narrow face.

Wide face taper also has a slight influence on narrow face shell behavior. This is illustrated in Figure 21, which compares the narrow face profiles obtained using 0.1% and 0.5% wide face taper. Increasing wide face taper improves contact slightly over the wide face, resulting in lower average temperatures, and correspondingly more shrinkage of the wide face. This results in slightly larger gaps forming on the narrow face, particularly at the corner. This effect can be seen in Figure 21, but its effect is not substantial.

In optimizing the mold taper design to better follow shell shrinkage, it was deemed adequate to maintain a linear taper down the wide face to ease the use of the mold in the production environment. This wide face taper must be adequate to support the wide face and influences the optimization of the narrow face taper, but careful attention to the narrow face taper is much more important.

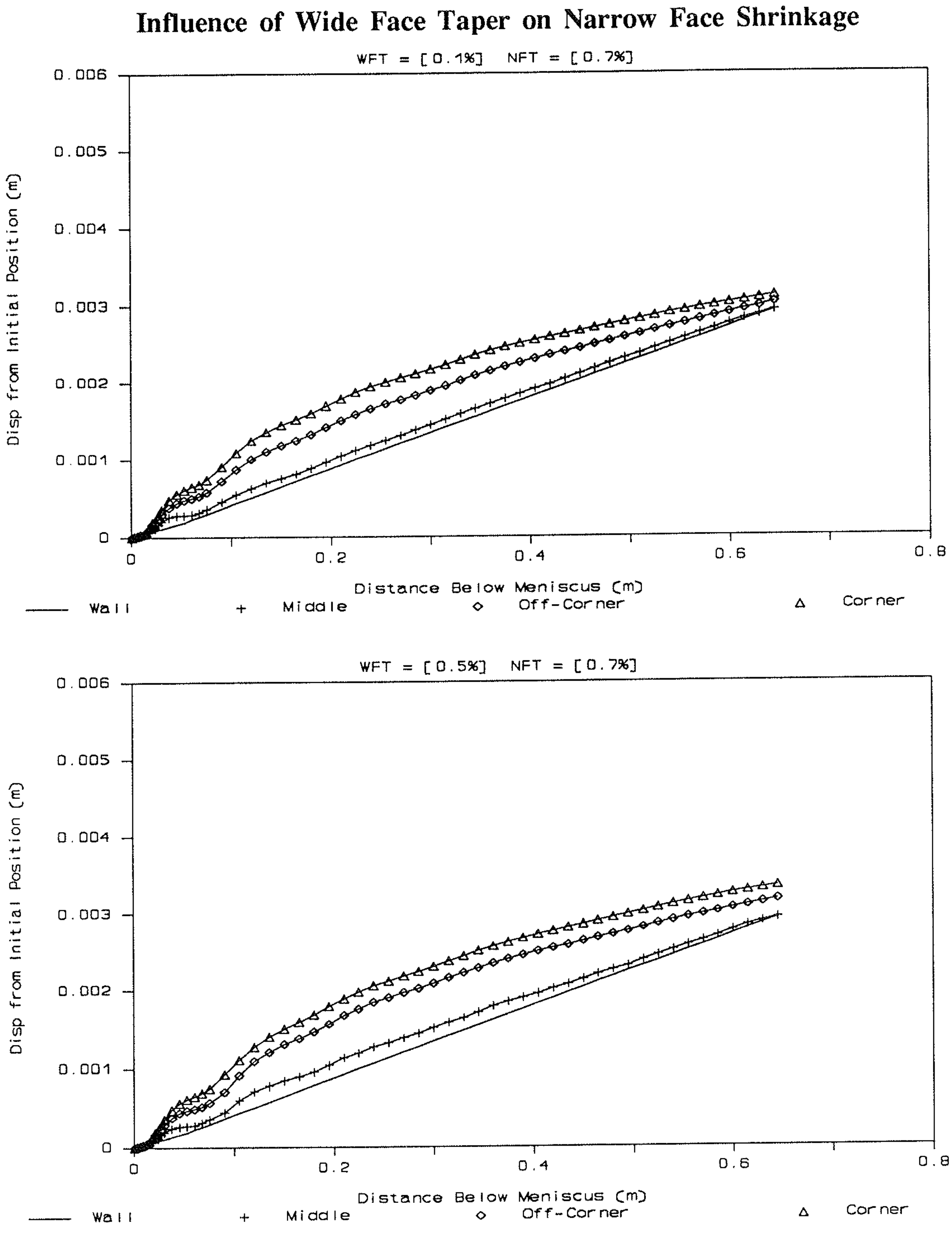


Figure 21 The shrinkage of the narrow face is slightly influenced by the taper of the wide face due to the moderately increased heat transfer and shrinkage rate caused by the higher taper.

4.0 MECHANISMS OF OFF-CORNER GUTTER FORMATION

The mathematical model results present a clear description of the thermal and mechanical behavior of the shell as it solidifies, cools, and travels down through the mold. This can be described and applied to understanding the mechanisms of gutter formation, as follows.

Near the meniscus, steel solidifies against the mold across both the wide and narrow faces. Once a continuous shell has formed around the mold, continued cooling makes the shell start to shrink away from the mold walls. This occurs first in the corner, where a gap forms. Initially, however, the corner has two-dimensional heat flow, so it grows relatively cold. At the same time, contact between the shell and mold far away from the corner is maintained by ferrostatic pressure, and the middle of each face grows cold also. Thus, heat flow is diminished the most in the off-corner regions of both the wide and narrow faces, once significant gaps have formed. This produces the corresponding highest surface temperatures or "hot spots" which are accompanied by local thinning and weakening of the shell.

Ferrostatic pressure continues to maintain good contact between the long, weak shell along most of the wide face, as it moves down through the mold. This contact is maintained on the wide face regardless of the taper, because the shell is too weak to withstand the pressure over the large area and adjusts continuously to any contour of the wide face. The gaps between the wide face mold and shell is determined mainly by the viscosity of the mold lubricant and surface roughness of the shell. This good contact causes the temperature of the wide face to drop rapidly resulting in the shell shrinking away from the narrow face. In the upper part of the mold, narrow face taper is insufficient to compensate for this shrinkage for all but the most excessive linear tapers, because cooling and shrinking of the wide face is much less in the lower part of the mold where shells are thick and the surface temperature difference reduced. Therefore, a linear taper that matches shrinkage in the upper part of the mold would crush the shell by mold exit.

Thus, the narrow face shell starts to pull away from the narrow face in the upper part of the mold. This results in a large gap, particularly in the off-corner region of the narrow face, with consequent decreased cooling and higher temperatures found there. The hot spot on the narrow face near the corner is therefore thinner and relatively weak. This allows ferrostatic pressure against the narrow face, which is unsupported due to insufficient taper, to bulge the narrow face shell outward. This causes the shell to rotate and bend around the corner, lifting up the off-corner region of the wide face [23]. This creates a slight gap in the off-corner region of the wide face, enhances the hot-spot there, creates a thinner shell, and weakens the wide face off-corner region.

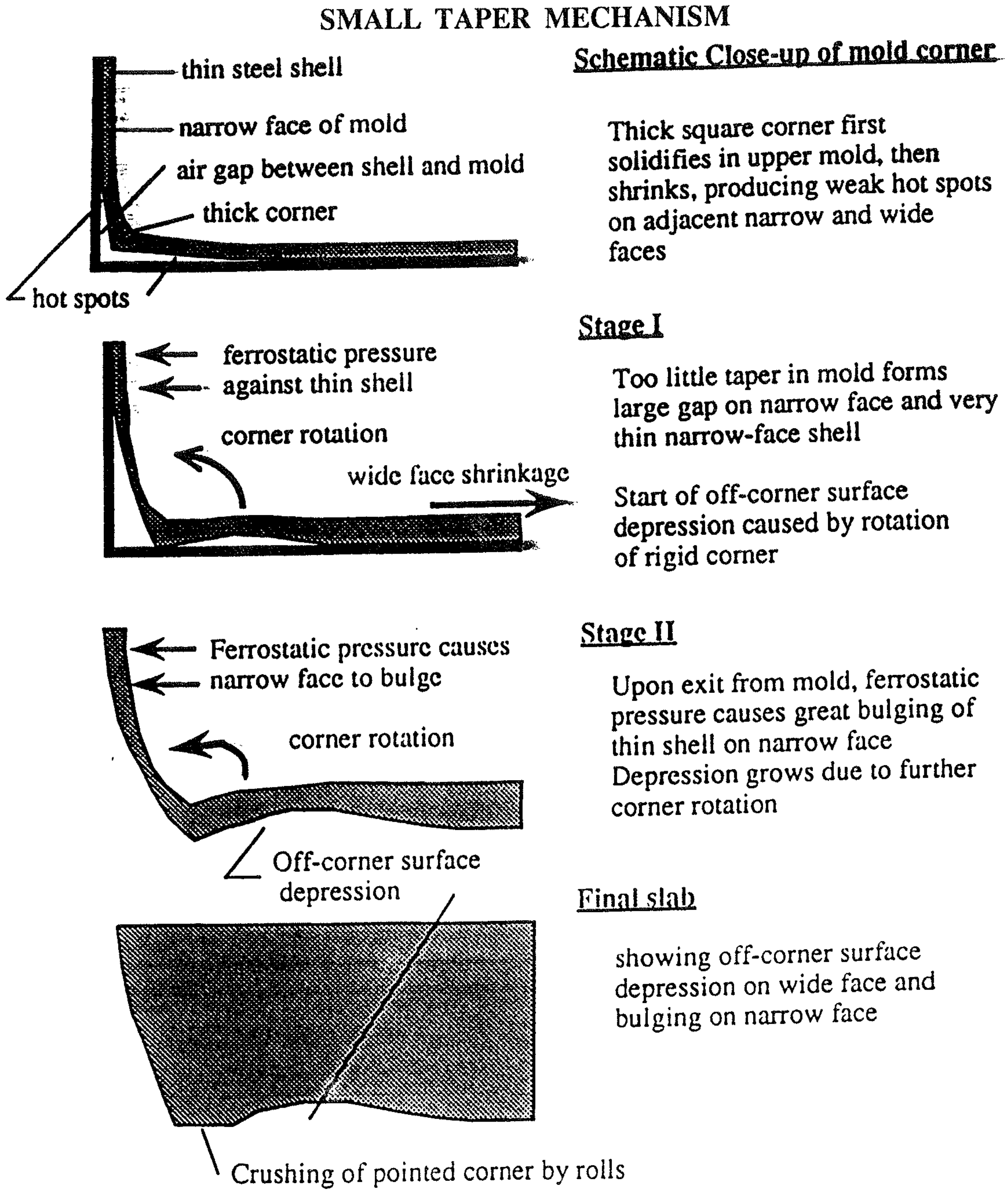
The succeeding events that enhance the wide face off-corner gutters depend on the amount of narrow face taper employed. These mechanisms are similar in that they initiate with the corner rotation phenomena and get worse at extremely large or small tapers. Thus, the mechanism is broken into the small and large taper mechanisms for clarity and to emphasize the role of the mechanical loading from the mold wall.

4.1 SMALL TAPER MECHANISM

For tapers of 0.7% or less, the gaps that form high in the mold persist to mold exit. The shell in the off-corner regions has a high surface temperature exiting the mold and is very thin and weak, due to the large gaps between the shell and mold wall created by insufficient taper. After the slab exits the mold, ferrostatic pressure against the narrow face will continue to induce it to bulge. Being hot and weak, the narrow face shell is susceptible to creep and excessive bulging of the narrow face will result. Without a mold wall to contain it, bulging of the narrow face shell could become quite severe. The wide face is supported by rollers below the mold and is thicker than the narrow face and more able to resist the bulging. The hot spot in the off-corner region of the wide face is the weakest area on this face and hinges under the moment from the narrow face corner rotation. This allows further rotation of the corner of the shell, enhancing the off-corner depression down the surface of the wide face. Travelling beneath the rolls may compress the fragile shell later on, resulting in a slight flattening of the pointed corner. Combined with shrinkage and further thermal distortion through the spray zones, the final gutter observed down the off-corner of the slab can be created.

This mechanism, believed responsible for gutter formation when small tapers (0.7% or less) are employed, is illustrated in Figure 22. This mechanism predicts that the worse gutters should be encountered with decreasing narrow face taper, since the shell would be thinner and more susceptible to bulging under ferrostatic pressure. However, since the simulation did not extend to below the mold, this mechanism is still untested, but very likely with the temperatures and shell thicknesses resulting from the small taper simulations.

The mechanism implies that the location and alignment of the first support rolls, particularly along the narrow face, should be important to the extent of off-corner gutter formation, since they affect bulging. Spray cooling just below the mold is also important,



Schematic Close-up of mold corner

Thick square corner first solidifies in upper mold, then shrinks, producing weak hot spots on adjacent narrow and wide faces

Stage I

Too little taper in mold forms large gap on narrow face and very thin narrow-face shell

Start of off-corner surface depression caused by rotation of rigid corner

Stage II

Upon exit from mold, ferrostatic pressure causes great bulging of thin shell on narrow face
Depression grows due to further corner rotation

Final slab

showing off-corner surface depression on wide face and bulging on narrow face

Figure 22 Off-corner surface depressions form for ***small*** narrow face tapers under this mechanism. This mechanism applies where the shell is inadequately supported by the mold wall and through the first supporting rollers after mold exit.

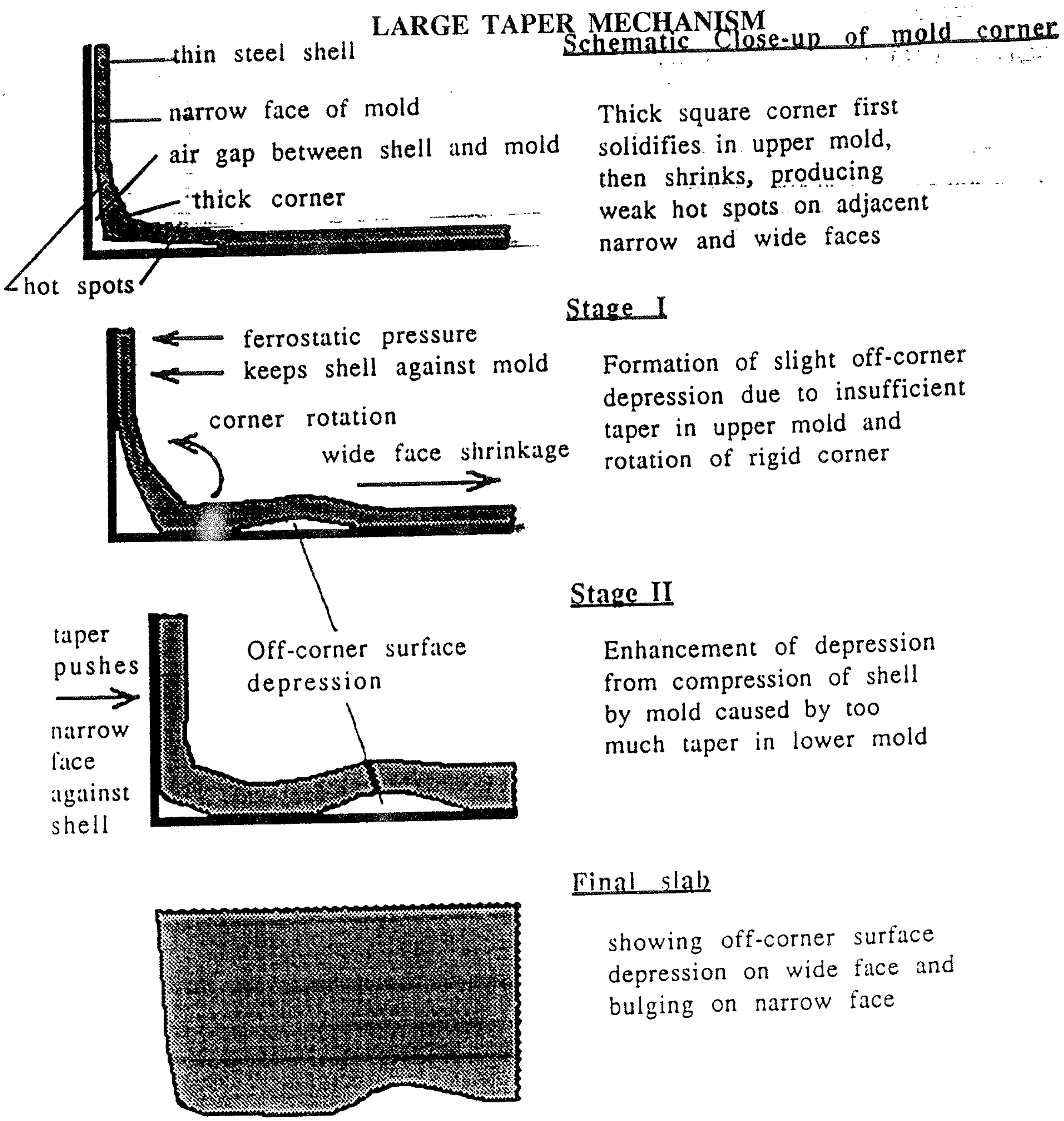
since it affects shell temperature and thickness. However, without proper taper and cooling of the narrow face in the mold, elimination of the gutter is impossible because of the thin shells and inconsistent support of the rollers.

4.2 LARGE TAPER MECHANISM

With large tapers on the narrow face, off-corner gutter is proposed to form via the mechanism illustrated in Figure 23. Lower in the mold, shrinkage of the wide face is less because the wide face temperature stabilizes and cools much more slowly. For narrow face tapers exceeding 0.7%, contact between the narrow face and shell is eventually restored. This reverses the internal pressure along the narrow face that was forcing the corner to rotate, and the off-corner gutter that formed earlier on the wide face is diminished.

When the taper down the narrow face is very large, the narrow face mold wall completely closes the gaps and thickens the narrow face shell due to reduced gap sizes. High tapers push against the narrow face shell on the corner, compressing the wide face. The combination of narrow face thickening and wide face compression can become so great that it causes the wide face shell to buckle inward, reforming the shallow off-corner depression that formed earlier which likely plays a role as an initiation site for this buckling. This causes a dramatic reduction in the heat flow from the region. The thickening of the narrow face reduces its flexibility. With the reduced flexibility, the narrow face transfers more of the moment caused by the ferrostatic pressure to the wide face. This moment rotates the corner producing a larger gap in the off-corner wide face position. This phenomena is particularly noticeable in the high taper temperature results in Figure 17. As the narrow face off-corner and middle positions dramatically reduce in temperature because of the impinging wall, the wide face off-corner temperature remains essentially constant. By mold exit, this effect causes the off-corner position on the wide face to exceed all other temperatures.

Below the mold, the thin shell produced in the off-corner region due to buckling is weak and susceptible to more corner rotation as the support to the increasing ferrostatic pressure is removed from the narrow face. This likely aggravates the off-corner gutter through corner rotation in the same manner as the mechanism proposed for small tapers.



<u>Schematic Close-up of mold corner</u>

Thick square corner first solidifies in upper mold, then shrinks, producing weak hot spots on adjacent narrow and wide faces

<u>Stage I</u>

Formation of slight off-corner depression due to insufficient taper in upper mold and rotation of rigid corner

<u>Stage II</u>

Enhancement of depression from compression of shell by mold caused by too much taper in lower mold

<u>Final slab</u>

showing off-corner surface depression on wide face and bulging on narrow face

Figure 23 Off-corner surface depressions form for ***large*** narrow face tapers under this mechanism. This mechanism applies where the shell is compressed after bulging and rotation have occurred.

5.0 MOLD TAPER OPTIMIZATION

The previous analyses provided useful data on the operation of the caster with linear tapers, the effects of changes to the taper, problems that arise due to taper changes, and a reasonable understanding of the mechanisms of wide face, off-corner gutter formation. The knowledge gained from this effort allows the optimization of the mold taper to reduce or eliminate the gutter problem, although both mechanisms also establish the criticality of the rollers just below the mold. This area will likely need optimization as well to fully remove the off-corner gutter problem.. The objective of the mold taper optimization is not only a taper that produces a slab with a minimum gutter, but also a shell with a constant thickness and uniform surface temperature for better casting control and higher strength to withstand the bulging in the supporting rolls.

The design criteria for this optimization will investigate the important characteristics of a mold design that does not produce a slab with a gutter. The desire to keep the temperatures uniform across the faces suggests that a constant size gap be maintained. The desire to maintain high productivity suggests that the gaps should be minimal to allow efficient heat transfer and higher cooling rates which together would allow high casting speeds. A taper design that satisfies the above constraints is one in which linear, excessive tapers are applied to both faces maintaining minimal gaps at all areas on the shell. As a demonstration, a run of this sort was attempted. This run actually did not have enough taper to satisfy the above conditions, but even at tapers lower than what was sought, compressive stress builds up in both faces and considerable buckling would result.

In addition to the criteria mentioned above, the compression of the shell due to excessive taper must be avoided. These constraints on mold design bracket an optimum. The dual-mechanism of defect formation would suggest that the compression of the shell is as important to the taper design as increased heat removal. It would also suggest that a taper that would support the bulging of the mid-face areas, especially in early stages of shell development, would also tend to produce a better slab. Therefore, the dual-mechanism of defect formation is quite useful in qualitatively investigating the methods of mold design optimization. The mechanisms suggests that uniform cooling along a face which produces a constant shell thickness should be sought. Also the compression of the shell and unrestrained bulging of the faces of the casting should be avoided. The mechanism and previous linear taper results are also useful in providing direction of achieving the optimal design. The compressive loads along the shell are predominantly driven by compression along the corner not in the middle of the face. This would suggest that a taper design restraining the middle of the face from bulging no further than the corner shrinkage would prevent the corner rotation formed by bulging. Finally, gaps are typically largest at the corner provided other factors do not initiate off-corner depressions. Therefore, keeping the gaps at the corner small minimizes the gaps across the entire face, produces efficient uniform cooling, supports bulging, eliminates corner rotation, and avoids shell buckling. Therefore, it can be concluded that the optimal mold taper design would follow the natural shrinkage of the corner.

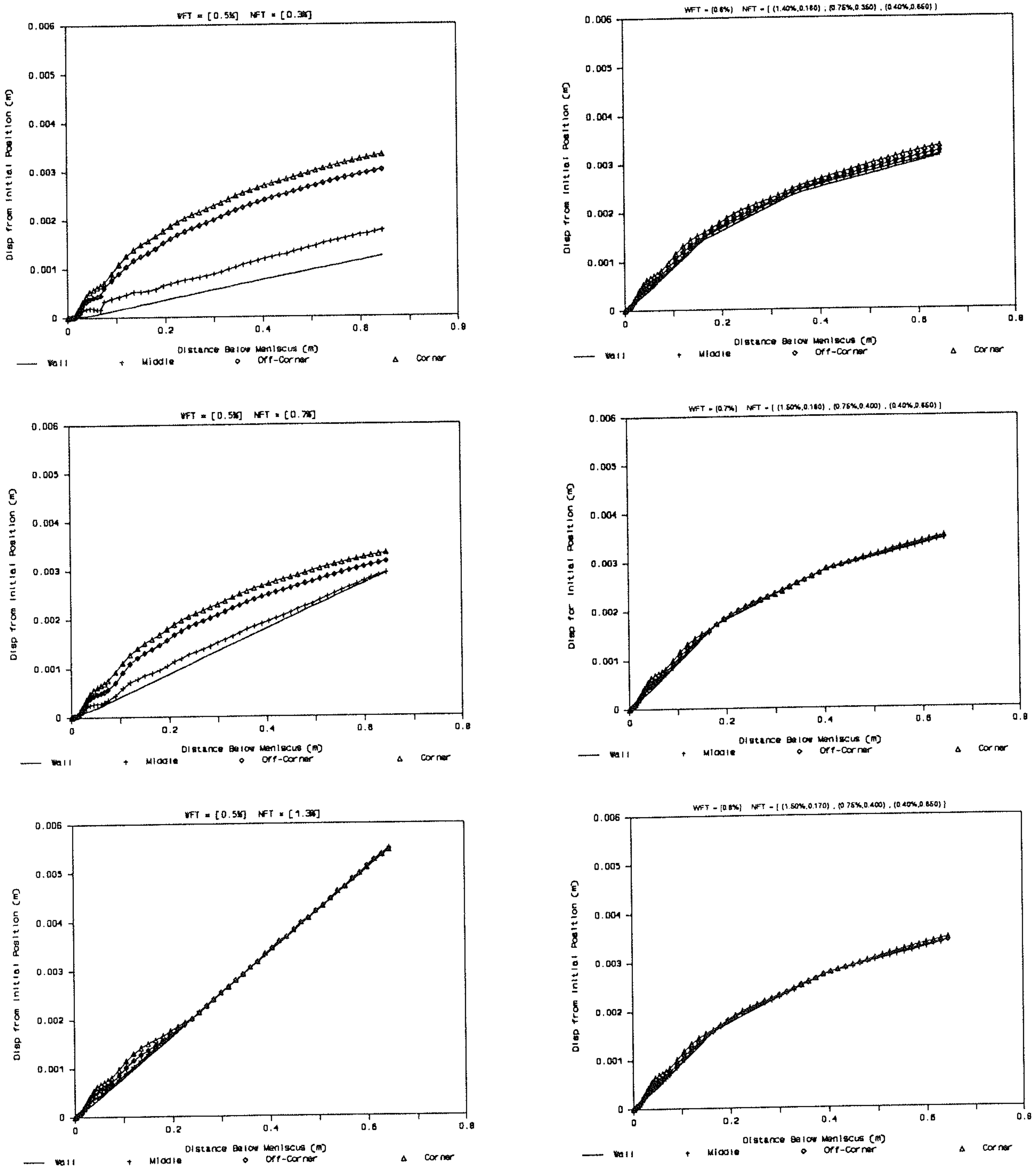
Many simulations were performed in an attempt to reach this optimum taper. First a taper consisting of two planar sections (bi-linear cross-section) was attempted. Air gaps and wide face compression still remained, suggesting that more flexibility would be needed to accurately follow the corner shrinkage. Subsequent simulations employed tri-linear taper designs, which seem well capable of matching shrinkage. One mold design more closely matches the mid-face position than the corner; another taper design reasonably matched the narrow face shrinkage but the wide face taper allowed excessive gaps. In addition to being interesting in their own rights, these runs indicate that the method of optimization is indeed reasonable, and that maintaining tight control of the taper is very important.

The optimal mold design most closely matches both the wide and narrow face shrinkage. This design includes a wide face taper of 0.8% per mold and a narrow face taper of 1.5% per mold from the meniscus to 170mm (6.69 inches) below the meniscus. The narrow face taper decreases to 0.75% per mold from 170mm (6.69 inches) below the meniscus to 400mm (15.75 inches) below the meniscus, and finally the taper changes to 0.40% per mold from 400mm (15.75 inches) below the meniscus to mold exit, which is at 650mm (25.59 inches) below the meniscus. The 0.8% taper on the wide face was adjusted, as well as possible within the linear constraint, to minimize the gaps in the corner without compressing the narrow face shell. The nomenclature used to describe the complete taper design reads:

WFT=[0.8%], NFT=[(1.50%,0.170) ; (0.75%,0.400) ; (0.40%,0.650)].

Figure 24 compares the results of the three tri-linear taper runs mentioned above with extremes in linear narrow face tapers, calculated earlier. These previous results are included to help indicate the improvements gained with the taper optimization. Notice that the large gaps present with linear tapers of 0.7% or less have been virtually eliminated in the optimized

Narrow Face Profile Comparison with Optimized Taper Designs



WFT = [0.5%] NFT = [0.3%]

WFT = (0.6%) NFT = [(1.40%,0.180) ; (0.75%,0.360) , (0.40%,0.660)]

WFT = [0.5%] NFT = [0.7%]

WFT = (0.7%) NFT = [(1.50%,0.180) , (0.75%,0.400) , (0.40%,0.650)]

WFT = [0.5%] NFT = [1.3%]

WFT = [0.6%] NFT = [(1.50%,0.170) , (0.75%,0.400) , (0.40%,0.650)]

Figure 24 The optimized taper simulations allow minimal gaps compared to the small taper simulations and produce the near uniform gap size throughout the mold produced by the very high linear taper.

design. This should allow higher productivity due to more efficient cooling, as well as more uniform temperatures and shell thicknesses to reduce the bulging below the mold.

Figure 25, which compares the corresponding wide face profiles, indicates that the wide face in the optimized design reduces the gaps as much as a linear taper will allow, without compressing the narrow shell. Figure 26 compares the temperature histories of some nodes on the surface of the shell. Notice that the effective cooling of the optimized design brings all temperatures below 1250°C (2280°F) only 200mm (8 inches) below the meniscus. Even the extreme linear taper required 500mm (20 inches) to cool to this extent, while the small tapers did not achieve this low temperature. Although the temperatures reveal some reheating, these reheating times are short and the temperature rise is small. Figure 26 also indicates that all the temperatures at mold exit are extremely uniform excepting the colder corner. Therefore, the goal to produce uniform and low temperatures by mold exit could not be met as well by any of the linear tapers.

Compression down the wide face for the optimized design is shown relative to the other taper designs in Figure 27. This indicates that the compressive stresses are a little higher than desired in a truly optimized design. However, in comparison with the high taper linear run, considerable improvement was made. Minor adjustments might improve the optimized taper design slightly, but these adjustments would probably be within the variation in taper experienced during casting. The implications of the past two figures together is greater than either of their parts. Together these figures indicate that the optimal design can maintain the small gaps and reduce mold exit temperature and temperature uniformity better than the high linear tapers. These figures also indicate that the optimal design induces only slightly more compression in the wide face than the small linear taper designs. Therefore, the optimal design improves on the best features of the extreme linear tapers and avoids the drawbacks of each.

The previous figures indicate that the optimized mold design meets the established design criteria, based on the knowledge of the linear taper designs and the defect formation mechanisms. Figure 28 confirms that these design criteria are reasonable since it shows that a mold design based on these criteria reduces the wide face gutter at mold exit. Notice that deflection just off the corner of the wide face for the optimized taper shows a gutter of less than 0.02 millimeter (0.0008 inch). This is an order of magnitude less than the nearly 0.23 millimeter (0.009 inch) gutter produced with the high linear tapers. The optimized

Wide Face Profile Comparison with Optimized Taper Designs

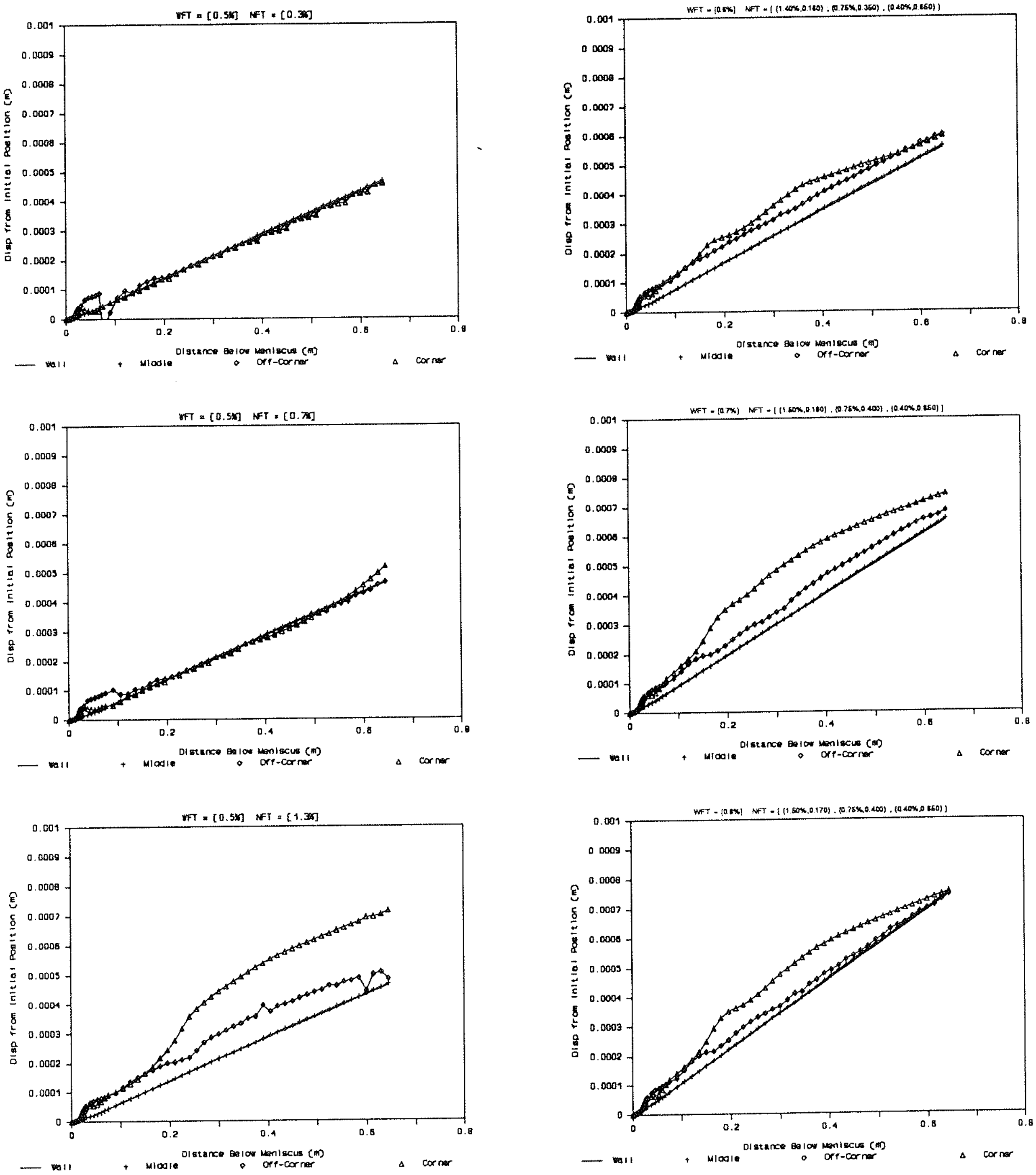


Figure 25 The optimized narrow face maintains the smallest gap size possible without compressing the narrow face shell. This optimized design also eliminates all areas on the caster where the off-corner region shrinks more than the corner.

Surface Temperature Comparison with Optimized Taper Designs

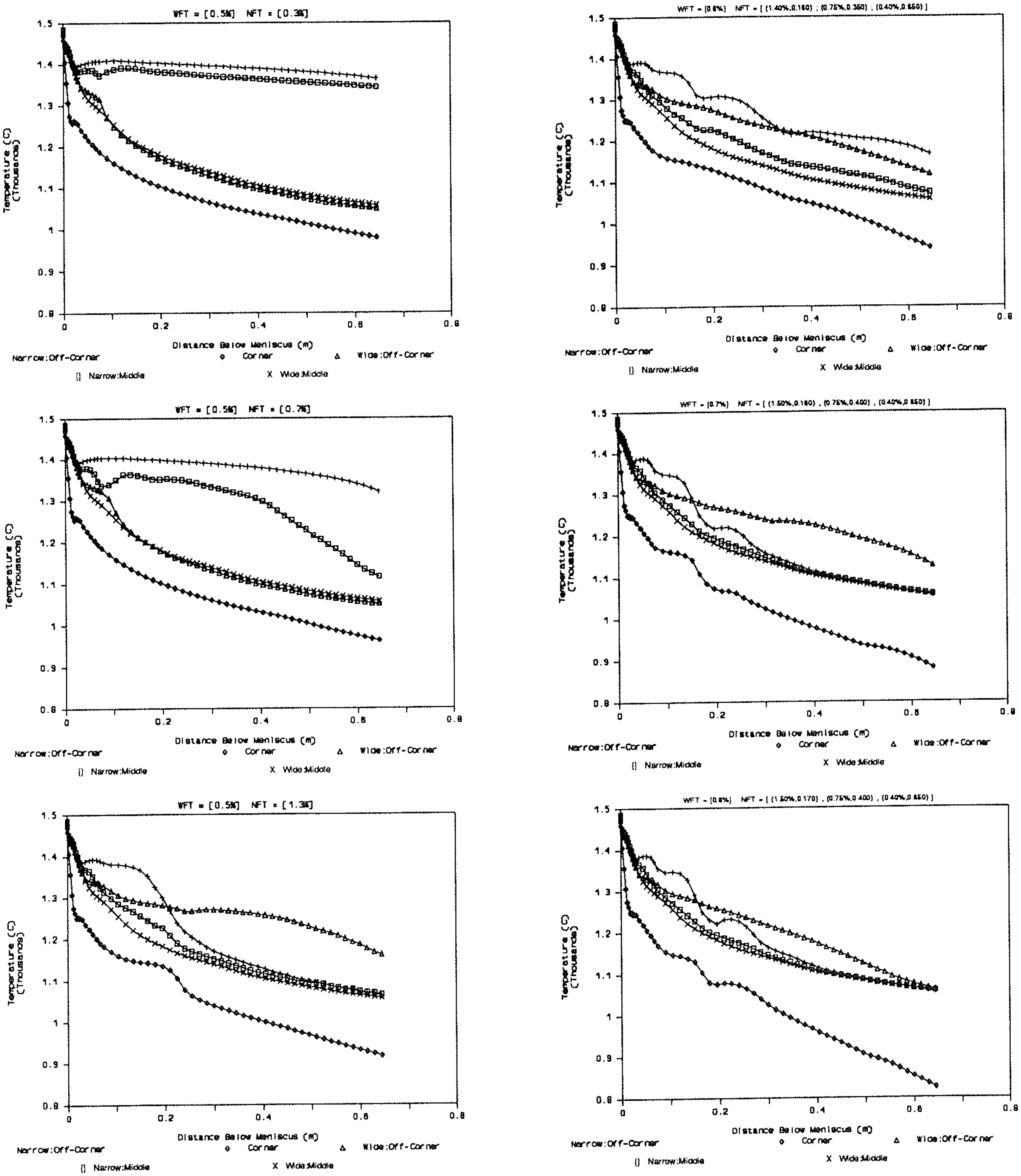


Figure 26 The optimized taper design produces lower mold exit temperatures, which are more uniform around the casting's perimeter, and less reheat during casting.

Wide Face Stress Comparison with Optimized Taper Designs

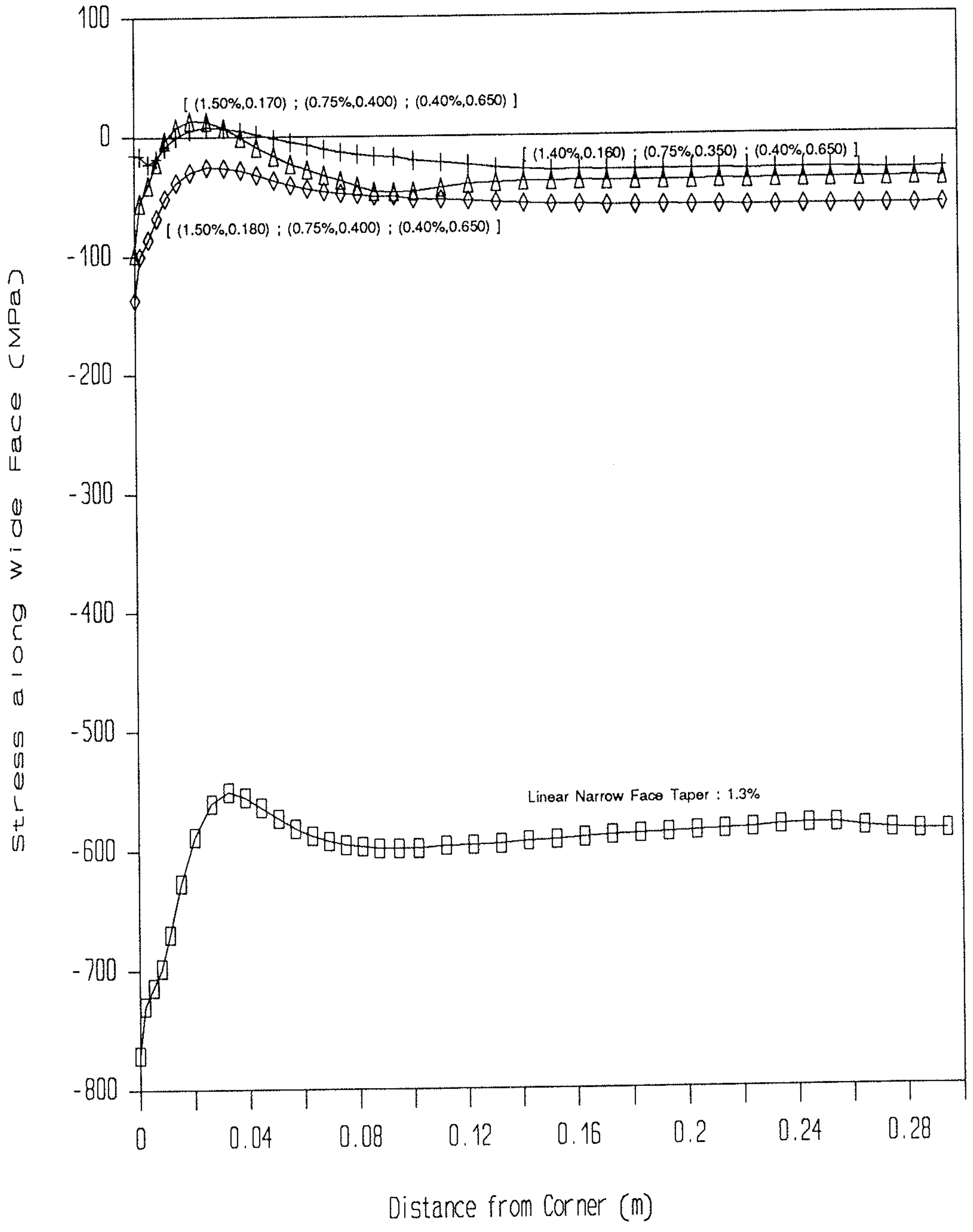


Figure 27 The optimized taper designs maintain the small gap sizes and efficient heat transfer observed with the large linear taper without introducing large compressive stresses along the wide face.

Wide Face Stress Comparison with Optimized Taper Designs

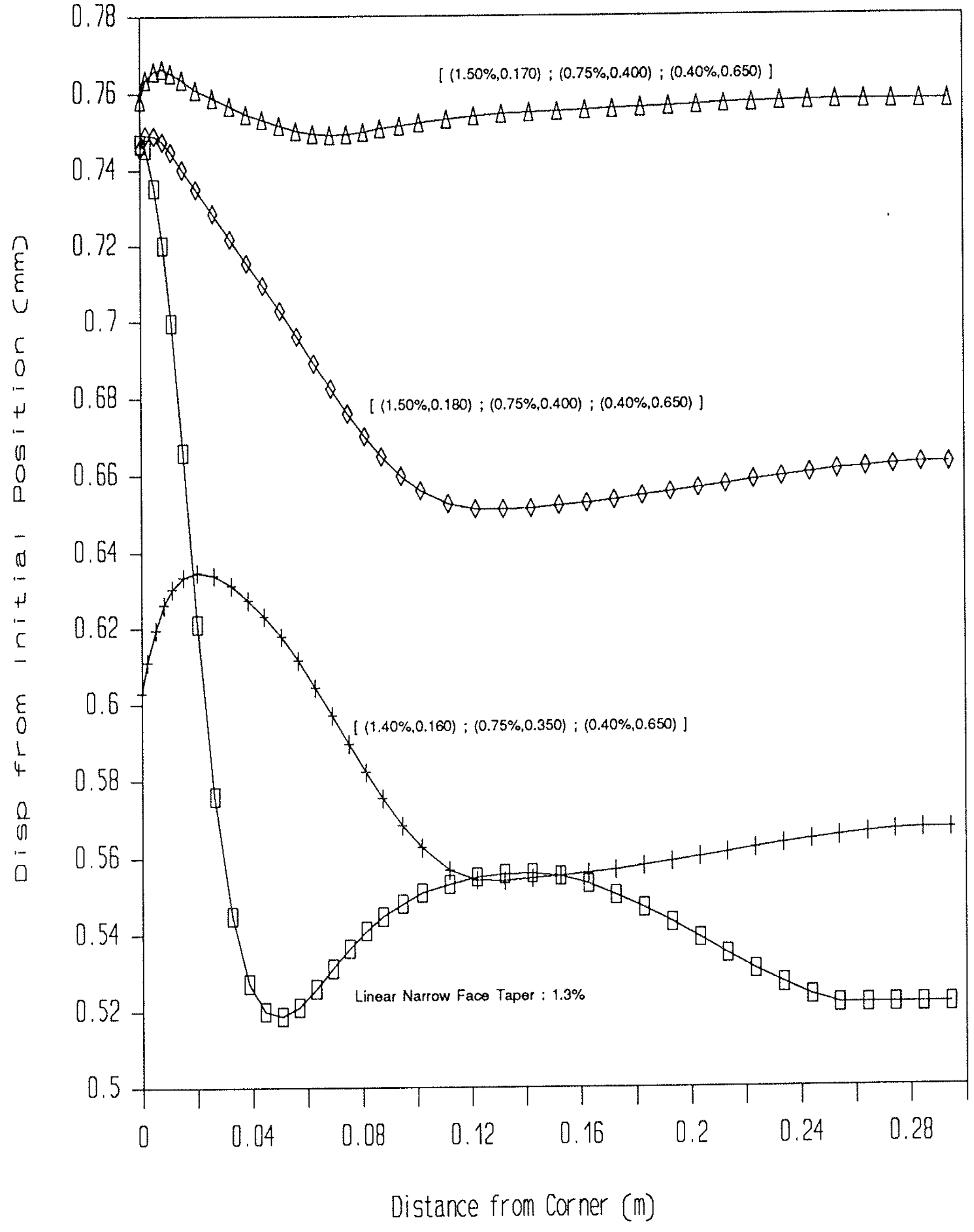


Figure 28 The optimized taper design eliminates all tendencies to produce an off-corner surface depression at the mold exit.

design also exhibits a thicker shell and lower temperatures resulting in dramatically increased strength at mold exit than the linear taper designs. This thicker, stronger, more uniform shell prevents this gutter from growing as much in the spray zone below the mold as described in the small-taper defect formation mechanism. Also notice the tri-linear design that more closely followed the mid-narrow face position. While this design moved the gutter closer to the corner, a significant gutter still appears to form, indicating that these results are highly dependent on keeping the taper following the corner. This also indicates that a taper must be maintained within about 0.2 millimeter (0.008 inch) of the desired position to realize the benefits from the improved taper design. Therefore, additional casting controls will be required for full experimental verification of these results and full implementation of this optimized design in the production of continuously cast stainless steel slabs.

6.0 CONCLUSIONS

A finite element model has been developed specifically to simulate the thermal and mechanical behavior of the solidifying strand in the continuous steel slab casting process. This model has successfully tracked the temperature, solidification, gap formation, distortion, and stress generation in the solidifying shell as it moves down through the mold under a variety of different mold taper conditions. It has also lead to a mold taper design which should dramatically improve the quality of the casting.

Elastic simulations using the mathematical model have reproduced several phenomena believed to occur to the shell solidifying in the mold. These simulations have produced the following findings.

- A model for heat flow across the interfacial gap has been developed that includes radiation in parallel with conduction contact resistances, conduction through the lubricating mold flux layer defined by minimum and maximum dimensions, and conduction across the remaining air gap. It provides enough flexibility to achieve good agreement with experimentally measured total heat flux in the mold and enough correlation to the actual casting process to investigate effects of flux powder changes.
- The size of the gap that forms between the shell and mold controls heat transfer, local shell surface temperature, and local shell thickness. Locally increased surface temperatures, or hot spots, produce locally weak and thin shells and a the result of poorly designed mold tapers.
- The total heat removed in the mold is relatively insensitive to changes in mold taper design.
- The majority of the wide face shell remains in contact with the mold wall throughout the entire time in the mold.
- The shell shrinks most in the upper part of the mold, forming large gaps on parts of both the narrow and wide faces.
- The size of the gaps, and consequently the thermal and mechanical behavior of the shell, are greatly affected by the narrow face taper.
- A linear narrow face taper always allows bulging, formation of off-corner hot spots, and a slight off-corner depression of the wide face, due to corner rotation high in the mold.
- Compression and buckling stresses in the wide face shell were calculated by the model for linear narrow face tapers exceeding 1.0% and resulted in a distorted slab shape

qualitatively similar to the off-corner gutter defect observed on the wide face of final cast slabs.

- Gap thicknesses exceeding 0.1 millimeter (0.004 inch) result in local hot spots on the solidifying shell.
- High temperatures, and an extremely thin narrow face shell, and a severe hot spot and off-corner gutter on the wide face shell were calculated by the model for very small tapers.
- Wide face taper also affects narrow face shrinkage but is less critical to gutter formation than narrow face taper.
- The behavior of the shell below the mold and its interactions with the spray cooling and supporting rollers plays an important roll in propagating the off-corner gutters initiated in the mold.

Severe off-corner gutters form via two separate mechanisms, each at the extremes in narrow face taper.

- At small tapers, the thin shell on the narrow face bulges beneath the mold, causing corner rotation. This mechanism also explains the off-corner gutter growth below the mold.
- At large tapers, the narrow face thickening with smaller gaps enhances corner rotation. The mold wall compresses the wide face shell, causing it to buckle.

Finally, the model has been applied to create a mold taper design that more closely matches the shrinkage of the corner of the shell and avoids the formation of off-corner gutter. This work has culminated in finding a mold taper design that appears to meet this objective for a specific set of casting conditions. The proposed mold taper design has been based on the following conclusions.

- A tri-linear taper along the narrow face can sufficiently approximate the continuously varying shell shrinkage, while a bi-linear taper cannot.
- A tri-linear taper has been proposed for the narrow face, which predicts considerable reduction in the depth and extent of off-corner hot spots and depressions in the mold. The optimal taper avoids gaps exceeding 0.05 millimeter (0.002 inch), resulting in relatively uniform heat transfer and shell growth around the shell's perimeter. The taper

design also avoids compressing the shell more than 0.02 millimeter (0.001 inch) past the natural shrinkage of the corner.

- An optimal taper design is achieved by following the natural shrinkage of the corner of the shell providing support for the bulging, thinner central portions of the face and eliminating compression along the opposite face.
- The wide face is less critical to off-corner gutter formation than the narrow face taper, but it must be matched to the optimal narrow face taper to produce the shrinkage as it was designed. In the proposed design, the linear wide face taper has been increased to 0.8% per mold for higher productivity and higher quality product.

In conclusion, a tool has been developed that can investigate many continuous casting phenomena and determine their effect on the thermal and mechanical behavior of the shell. It has immediate practical use in optimizing mold taper designs as demonstrated by this project.

7.0 RECOMMENDATIONS

The finite element simulations of the continuous casting process have provided considerable insight into the physical phenomena controlling shell behavior in the mold. Caster design and controlling parameters can be modified to help prevent off-corner gutter formation in stainless steel slabs. These modifications are described below.

- **Controlling taper** down the narrow face, as well as other casting process variables, is crucial. Without proper taper and cooling of the narrow face in the mold, prevention of gutter formation is impossible. It is recommended that the taper be monitored continuously during casting to ensure the desired taper is maintained. It is most important to prevent loss of taper, since very small tapers appear to generate the most severe gutter problems due to the effects in the rollers below the mold. However, it is also important not to have excessive taper, since this will also create gutter problems, and many other potential problems. A linear taper of between 0.7% and 0.9% per mold appears to be the best compromise of linear tapers for casting 0.9 meter (36 inch) wide, AISI 304 stainless steel slabs.
- **Develop and employ an improved mold taper design** that more closely follows the shrinkage of the shell. The present linear tapers cannot achieve this, particularly on the narrow face. For 304 stainless steel cast at 0.015 meters/sec (35.4 inches/min), a 0.9 meter (36 inch) wide slab, and an 80 millimeter (3.1 inch) deep meniscus, the following taper design is suggested.

Narrow Face Taper	Wide Face Taper	Distance Below Meniscus
1.50% per mold	0.80% per mold	0.00 through 0.17 meter (4.9 inch)
0.75% per mold	0.80% per mold	0.17 through 0.40 meter (8.0 inch)
0.40% per mold	0.80% per mold	0.40 through 0.65 meter (17.0 inch)

Note that this solution requires reasonable knowledge and control of the meniscus level relative to the end of the mold to be effective.

- **Instrument the mold** to measure slab shape, shell surface temperatures at mold exit, and mold temperatures. This information is needed to correlate the mathematical model and improve the quality of its predictions. Measuring these quantities under a variety of casting conditions will also help refine and understand the mechanism of off-corner gutter formation as well as many other mold related defects.

- **Investigate cooling and support conditions below the mold.** Adequate support of the narrow face shell is important to prevent bulging and corner rotation that likely aggravates gutter formation below the mold. In addition, correct alignment of the mold with the upper support rolls is important to prevent distortion of the thin shell, which would further enlarge gutters disproportionately on different faces. Finally, spray cooling just below the mold is important to shell strength and should be investigated, particularly on the narrow face.

8.0 FUTURE WORK

The results of this research have demonstrated an effective tool has been developed to aid continuous casting mold design. Further work in both the code development and the model application would improve the usefulness of this tool. This work could proceed with the following objectives.

1) **Incorporate concurrent work on fluid flow and mold distortion into this model.** Mika's continuous casting turbulent fluid flow investigation [5] can provide information on the heat flow to the solid-liquid interface from the molten steel pool. This would eliminate assumptions in the current model and reduce run times by only requiring the temperatures and stresses in the solid shell to be modelled. Continuous slab casting mold distortion has been investigated by Azzi [6]. Incorporation of these results into the facility already in this finite element model will refine predictions or this model.
2) **Continue the casting simulation through the first few support rollers.** The investigation of the spray zone and further processing by Haegele [7] did not incorporate the required details to study the off-corner gutter growth below the mold. An investigation incorporating this detail will refine the presented dual-mechanism of gutter formation.
3) **Nest the iteration procedure.** The iteration schemes used in this finite element model can be nested to proceed concurrently rather than consecutively. This may reduce the computational needs of this code and allow the application of its plastic flow formulation.
4) **Develop and apply the plastic flow formulation to the continuous casting mold.** The high temperature inelastic behavior of solidifying steel will be required to develop a true understanding of the continuous casting process. Results using this formulation can confirm the usefulness of results obtained from elastic assumptions.
5) **Formulate large displacement and post-buckling simulation into the finite element model.** The gutter formation mechanism for large tapers requires the explicit calculation of buckling strains to realistically calculate the shape and extent of the forming gutter. These calculations have not been formulated into this finite element model and are not required to detect the formation of the gutter nor to design the process to avoid them. Including the buckling strains would allow better correlations of this model's results to the actual performance of the caster.

REFERENCES

1. Evaluation of Casting Variables and Their Effect on Slab Shape, ARMCO Report on Stainless Steel - Continuous Casting - Butler Works, 1986.

2. A. Grill, K. Sorimachi, and J. K. Brimacombe, "Heat Flow, Gap Formation and Break-Outs in the Continuous Casting of Steel Slabs", Metallurgical Trans., Vol. 7B, 1976, pp. 177-189.

3. J. O. Kristiansson, "Thermal Stresses in the Early Stage of Solidification of Steel", Journal of Thermal Stresses, Vol. 5, 1982, pp. 315-330.

4. I. Ohnaka and Y. Yashima, "Stress Analysis of Steel Shell Solidifying in Continuous Casting Mold", presented at Modeling of Casting and Welding Processes, Engineering Foundation Conference, Palm Coast, FL, April 17-22, 1988.

5. B. G. Thomas and L. M. Mika, "Simulation of Fluid Flow and Heat Transfer Inside a Continuous Slab Casting Machine", Second FIDAP Users Conference, Evanston, IL, Oct. 2-4, 1988.

6. J. A. Azzi, "Preliminary Steps in the Development of a 3-D Thermal-Stress Model of Continuous Casting", M. S. Thesis, University of Illinois, Urbana, IL, 1988.

7. G. T. Haegele, "Mathematical Modeling of Defect Formation During the Hot Charging of Continuously Cast Steel Slabs", M. S. Thesis, University of Illinois, Urbana, IL, 1988.

8. B. G. Thomas, I. V. Samarasekera, and J. K. Brimacombe, "Comparison of Numerical Modeling Techniques for Complex, Two-Dimensional, Transient Heat Conduction Problems", Metallurgical Trans., Vol. 15B, No. 2, 1984, pp. 307-318.

9. B. G. Thomas, I. V. Samarasekera, and J. K. Brimacombe, "Mathematical Model of the Thermal Processing of Steel Ingots, Part I: Heat Flow Model", Metallurgical Trans., Vol. 18B, No. 1, 1987, pp. 119-130.

10. B. G. Thomas, I. V. Samarasekera, and J. K. Brimacombe, "Mathematical Model of the Thermal Processing of Steel Ingots, Part II: Stress Model", Metallurgical Trans., Vol. 18B, No. 1, 1987, pp. 131-147.

11. W. Kurz, and D. J. Fisher, Fundamentals of Solidification, Aedermannsdorf, Switzerland, Trans Tech Publications, 1986.

12. O. C. Zienkiewicz, The Finite Element Method, McGraw-Hill Company, London, Third Ed., 1977.

13. G. Comini, S. Del Guidice, R. W. Lewis, O. C. Zienkiewicz, "Finite Element Solution of Non-Linear Heat Conduction Problems with Special Reference to Phase Change", International Journal for Numerical Methods in Engineering, Vol. 8, 1974, pp. 613-624.

14. F. P. Incropera and D. P. DeWitt, Fundamentals of Heat Transfer, John Wiley and Sons, New York, 1981.

15. K. Rathjen and L. Jiji, Journal of Heat Transfer, Trans. ASME, Vol. 93, 1971, pp. 101-109.

16. A. Mendelson, Plasticity: Theory and Application, R. E. Krieger Publishing, New York, 1983.

17. R. W. Fox and A. T. McDonald, Introduction to Fluid Mechanics, John Wiley and Sons, New York, Second Ed., 1978.

18. B. A. Boley and J. H. Weiner, Theory of Thermal Stresses, R. E. Krieger Publishing, Malabar, Florida, 1985.

19. B. G. Thomas, I. V. Samarasekera, and J. K. Brimacombe, "Investigation of Panel Crack Formation in Steel Ingots: Part I, Mathematical Analysis and Mid-Face Panel Cracks", Metallurgical Trans., Vol. 19B, No. 2, 1988, pp. 277-287.

20. B. G. Thomas, I. V. Samarasekera, and J. K. Brimacombe, "Investigation of Panel Crack Formation in Steel Ingots: Part II, Off-Corner Panel Cracks", Metallurgical Trans., Vol. 19B, No. 2, 1988, pp. 289-301.

21. R. D. Cook, Concepts and Applications of Finite Element Analysis, John Wiley and Sons, New York, Second Ed., 1984.

22. I. V. Samarasekera and J. K. Brimacombe, Canadian Metallurgical Quarterly, Vol. 18, 1979, pp. 251-268.

23. J. K. Brimacombe, I. V. Samarasekera, and R. Bommaraju, Optimum Design and Operation of Moulds for the Continuous Casting of Steel Billets, TMS-AIME Conference, Washington D. C., 1986.

APPENDIX 1
USE AND STRUCTURE OF THE FINITE ELEMENT MODEL

The finite element code, written in FORTRAN-77 with some non-standard features, is structured into subroutines by function. The structure of this code as well as its input is explained to help aid future use and development. The finite element results produced from this code are difficult to interpret without the accompanying post processing routines. These routines as well as other useful routines were developed and discussed by Haegele [7]. The following section describes the source code in detail; it is useful to the user interested in determining how the finite element theory is implemented, understanding how convergence tolerances effect the calculations, and modifying the code structure for better convergence or additional features. Only parts of this section are necessary for the initial user of this code.

A1.1 FINITE ELEMENT SOURCE CODE

This version of the finite element code was developed under the UNIX operating system and does include some non-standard FORTRAN features allowed by this system, particularly the include statement. These features rarely offer much difficulty in transporting this code to different machines. The source code is organized into 10 separate UNIX files each containing a variety of subroutines and function subprograms. These files group together parts of the codes that perform similar functions to facilitate the organization of the code. These files are:

inclcom Contains the common blocks and parameter statements to be included in the FORTRAN subroutines

ccmain.f Main program for CONCAST - couples the heat and stress models, controls input and output routines, and tracks load steps through time

hsask.f Prompts the user for information needed to run both the heat and stress models

hsrun.f Contains the subroutines responsible for controlling the individual heat and stress models separately or during each portion of the coupled analysis

dvbsol.f Contains the Cholesky decomposition solution routines

inital.f Contains the routines needed for initial problem set up and geometry and constant definition

props.f Contains the subroutines defining material properties

htsub.f Contains the subroutines to set up of the heat transfer matrices

htbnd.f Contains the subroutines to apply the boundary conditions to the heat transfer force vector strsub.f Contains the subroutines to set up of the stress analysis matrices

strbnd.f Contains the subroutines to apply the boundary conditions to the stress analysis force vector

The subroutines within these files are discussed in detail in the following sections with the exception of dvbsol.f. Dvbsol.f contains code for the Cholesky decomposition and back substitution for a variable bandwidth symmetric matrix. This procedure is discussed in many numerical analysis references. The finite element code does not take advantage of the routines' use of the variable bandwidth solution procedures to allow a more understandable addressing system between the stiffness matrix and the degree of freedom in the displacement vector.

A1.1.1 INCLCOM

The file inclcom contains the parameter statements and common blocks included in every routine in the finite element code structure. This is accomplished on the UNIX machines with the FORTRAN INCLUDE statement. However, the CTSS operating system on the CRAY-XMP/48 does not permit the use of this feature. Instead, a cliche is defined for all statements to be included, and the precompiler on this system translates this cliche into FORTRAN code performing the same function as the include statement. The variable names are explained here and have the same meaning everywhere in the code. Important variables within the other files will be explained in the discussion of each file, but the included variable definitions will not be repeated. The parameters used in this file determine how much space is reserved for every FORTRAN array. To avoid violating double precision data type mismatch on some machines, these parameters must be set to even numbers.

<u>Parameters</u>:

nendp Parameter for the number of elements in the stiffness matrix array

ndfp Parameter for the number of degrees of freedom in the mechanical force vector

nnph Parameter for the number of nodes used in setting up arrays for the heat transfer and coupled analyses

nnps Parameter for the number of nodes used in setting up arrays for the stress and coupled analyses neph Parameter for the number of elements used in setting up arrays for the heat transfer and coupled analyses

neps Parameter for the number of elements used in setting up arrays for the stress and coupled analyses

nbeph Parameter for the number of boundary edges used in setting up arrays for the heat transfer and coupled analyses

nbeps Parameter for the number of boundary edges used in setting up arrays for the stress and coupled analyses

Character Strings:

rtitle 80 character string containing the title describing the run conditions

mtitle 80 character string containing the title describing the mesh used for the analysis

jtitle 80 character string allowing room to read in another title that is discarded

decsize OBSOLETE VARIABLE: 1 character (y or n) to indicate the decision of whether a time step size file will be supplied to the code

dectemp 1 character (y or n) to indicate the decision of whether a temperature file is supplied to a mechanical analysis

sizename OBSOLETE VARIABLE: 25 character string containing the name of the file containing the time step size control flags

tempname 25 character string containing the name of the file containing the temperatures from a previous thermal solution that will be used for a separate mechanical analysis

str0dec 1 character (y or n) to indicate if a file exists describing the initial stress state of the material

str0nam 25 character string containing the name of the file describing the initial stress state of the material

powflo OBSOLETE VARIABLE

These character strings are common to all routines because they are stored in the common block named chars.

Named Common Blocks:

Common block - size - This common block contains variables determining the actual size of the arrays used in the finite element code.

nends Actual length of the stress analysis stiffness vector

nendh Actual length of the heat transfer analysis stiffness vector

ndf Actual number of degrees of freedom in the mechanical force vector

nn Actual number of nodes

ne Actual number of elements

nbe Actual number of boundary edges

nhbw Largest number of nodes contained in the half bandwidth for heat transfer solutions

nbw Largest number of degree of freedom contained in the half bandwidth for stress analysis

nns Number of nodes needed for stress analysis - This equals zero for heat transfer analysis and nn for stress or coupled analyses

nbes Number of boundary edges needed for stress analysis - This equals zero for heat transfer analysis and nbe for stress or coupled analyses

Common block - vbsolve - Contains the vectors needed during the solution process

a Contains both the heat transfer and stress analysis stiffness matrices

a0 Contains a copy of the stiffness matrix, a, before it is adjusted for boundary conditions

f Contains the mechanical force vector before solution and the nodal incremental displacements after the mechanical solution

fstore Contains a copy of the mechanical force vector before inelastic loads are applied

fht Contains the heat transfer force vector before solution and the new temperatures after the thermal solution

stiffgap Contains a large number for the stiffness of the gap if the mold and shell are calculated to be interfering

irh Array used to instruct the Cholesky decomposition routines where each row of the heat transfer stiffness matrix begins in the a vector irs Array used to instruct the Cholesky decomposition routines where each row of the mechanical stiffness matrix begins in the a vector

Common block - geomtry - Contains arrays describing the geometry of the finite element mesh as well as the geometric constants calculated from the mesh

xcoord Contains the x coordinate of all the nodes

ycoord Contains the y coordinate of all the nodes

bb Contains the geometric constant $\mathbf{b}_i$ in Equation 2.1.1.B

cc Contains the geometric constant $\mathbf{c}_i$ in Equation 2.1.1.C

areann Contains the total nodal area for each node

arean Contains the nodal area for each of the 3 nodes in every element

area Contains the area of each element

areat Contains the total area covered by the finite element mesh

node Assigns each node number to the i,j, or k position within each element

letype Contains each element's type which can be used to distinguish material property differences

Common block - bound - Contains the variables associated with the boundary conditions

bl Length of each of the boundary edge

bnxny Boundary normal vectors

bndgap Current gap at each of the two nodes of the boundary edge

bndgap0 Old gap at each of the two nodes of the boundary edge

avehw1 OBSOLETE VARIABLE: Average of the v displacements along middle portion of the wide face

nbnc Node numbers of the two nodes on the boundary edge

lbhtyp Heat transfer boundary condition type

nposh Flag for the position of the heat transfer boundary condition

lbstype Stress analysis boundary condition type

nposs Flag for the position of the stress analysis boundary condition

ichngdof Flag, used at the start of a mechanical load step, indicating which degrees of freedom are to be fixed to the mold wall to avoid interference based on previous load steps

jchngdof Flag indicating which degrees of freedom are fixed to be just inside the mold position when the mold interference criterion is checked

Common block - temps - Contains the arrays of temperatures at different iterations as well as the direct results of calculations from these temperatures

t Temperatures for the previous iteration for the heat transfer analysis

told Temperatures for the previous iteration of the heat transfer analysis

tpast Temperatures for the previous iteration of the stress analysis which is different from that of the heat transfer analysis only when the load step size is cut to facilitate inelastic convergence

tavg Average temperature for every element

tavgt Average temperature for the entire finite element domain

temp0 Initial temperature for the analysis if it is not read from a file

tmpinit Time at which the initial temperature should be read if initial temperatures are read in from a file

qtotp Total heat flow across the perimeter of the shell while in the mold

qpred Experimentally determined heat flow across the perimeter of the mold for the ARMCO Butler Caster with a casting speed of 15 millimeters per second (0.6 inch per second)

qtotf Total heat flow across the perimeter of the shell while in the spray zones

q1..q6 Heat flow across specific points on the surface of the shell while in the spray zone

hc1..hc6 Heat convection coefficient corresponding to q1..q6 locations

Common block - props - Contains material property vectors

e Elastic modulus at each node

fa Fraction of ferritic material at each node

at Difference between the thermal linear expansion value at the current load step and the thermal linear expansion at the previous load step

tle Thermal linear expansion for the current load step

epdot Plastic strain rate for the current load step

epdt0 Plastic strain rate for the previous load step

var Vector used to store property values while calculations on these properties are occurring

tsol Solidus temperature of the material

tliq Liquidus temperature of the material

pnu	Poisson ratio for the material
dens	Density of the material
gaccel	Acceleration due to gravity in consistent units with material properties
ar1	Pearlite to ferrite + austenite transition temperature upon reheating
a1	Pearlite to ferrite + austenite transition temperature in equilibrium
ac1	Ferrite + austenite to pearlite transition temperature upon cooling
ar3	Ferrite + austenite to austenite transition temperature upon reheating
a3	Ferrite + austenite to austenite transition temperature in equilibrium
ac3	Austenite to ferrite + austenite transition temperature upon cooling
pc	Percent carbon in the steel
pmn	Percent manganese in the steel
stifmult	Multiplier for stiffness of gap condition

Common block - strout - Contains the displacements, stresses, and strains, resulting from solution of the mechanical analysis

ut	Total u displacement at each node
vt	Total v displacement at each node
sigmat	Total stress at each node for four components - x,y,z, and xy
sigma	Incremental stress at each node for four components
epslnt	Total strain at each node for four components
epsln	Incremental strain at each node for four components
evp	Incremental plastic strain at each node for four components
evpt	Total plastic strain at each node for four components
e0	Total thermal strain at each node

Common block - effecs - Contains the effective stress and strain values

seff	Effective or equivalent total stress based on Von Mises criteria
eeff	Effective total strain
evpeff	Effective incremental inelastic strain
epefft	Effective total inelastic strain

Common block - spcpts - Contains nodal coordinates or node numbers that are used for output of special quantities

xmin	Minimum value for the x coordinate over the entire mesh
xmax	Maximum value for the x coordinate over the entire mesh

ymin	Minimum value for the y coordinate over the entire mesh
ymax	Maximum value for the y coordinate over the entire mesh
isp	Array of node numbers where temperatures are written to the summary file
nncen	Node number in the center of the narrow face
nwcen	Node number in the center of the wide face
nnoffc	Node number in the off-corner region on the narrow face
nwoffc	Node number in the off-corner region on the wide face
ngout5..16	Node numbers determined by the code useful in generating taper plots
intjunk	Unused variable keeping an even number of integer variables in the common block to help double precision floating point variable addressing

Common block - ctrl - Contains variables which control the time stepping, iteration, or loading conditions

ctime	Current time
zdist	Current distance along axis of caster
dth	Time step size for thermal analysis
dts	Time step size for mechanical analysis
casvel	Casting speed or velocity
radius	Radius of curvature on the casting mold
rmolng	Length of mold below the meniscus
tinit	Initial time
tmax	Maximum time
freq	OBSOLETE VARIABLE
timeload	Specific time at which a specified surface load should be applied
timepres	Time after which ferrostatic pressure should be applied to the solidifying surface
iter	Current iteration number
imax	Maximum number of iterations
iz	Flag indicating which stress-state assumption is being used to resolve the z direction stresses and strains
ianalyz	Flag indicating which sort of analysis is to be performed
isimul	Flag indicating if which stage of the continuous casting process is being simulated

idie — Flag indicating termination of the run for any reason

Common block - flags - Contains specific flags indicating how the analysis will be performed, what formulation is used, or what loading is applied

ifcap — Flag indicating which formulation is used in assembling the thermal capacitance matrix

ifenth — Flag indicating which formulation is used in evaluating the enthalpy gradient

ifstep — Flag indicating which formulation determines the time stepping procedure

ifeval — Flag indicating at which temperature the material properties are evaluated

ipsfunc — Flag indicating the form of the plastic strain function

imoldf — Flag determining if mold interference constraints should be checked

ipress — Flag indicating if ferrostatic pressure should be applied to the solidifying surface

itriang — Flag indicating the triangularization state of the stiffness matrix

icurve — Flag indicating if the mold is straight or curved

isloads — Flag indicating if specified surface traction loads should be applied

iunit — Input unit number

ioutput — Flag indicating the brevity of the prompts

iounit — Output unit number

ifsubs — Flag indicating a need to sub-iterate on inelastic behavior

Common block - tols - Contains the values for the tolerances and other parameters for the convergence criteria

strrat — The ratio of incremental effective plastic strain to total effective strain which causes a sub-iteration when exceeded

strmin — The smallest total strain on which the strrat ratio should be based

difcnv — The percent difference between initial and resultant plastic strain rates which causes a convergence iteration when exceeded

difsub — The percent difference between initial and resultant plastic strain rates which causes a sub-iteration when exceeded

epmin — The minimum strain rate on which the above plastic strain rate comparisons should be based

gpscnv — The percent difference in generalized plane strain z strain which causes an iteration on this z strain when exceeded gpsez The converged upon value for z strain under the generalized plane strain assumptions

strmin2 The smallest total strain on which the trunrat ratio should be based

trunrat The ratio of incremental effective plastic strain to total effective strain which causes truncation to the maximum incremental effective plastic strain

epfact Convergence acceleration factor for convergence on plastic strain rate

Common block - chars - Contains the character string data discussed above

Common block - caster - Contains information used in simulating the continuous caster below the mold. These variables have been discussed by Haegele [7].

Common block - gaps - Contains information on the material in the gap between the shell and the mold and information regulating the convergence on the gap condition.

powmax Maximum thickness of the casting powder between the shell and the mold

powmin Minimum thickness of the casting powder between the shell and the mold

rkair Thermal conductivity of the gas between the shell and the mold when this area is not filled with powder

rkpow Thermal conductivity of the casting powder

h0mold Contact convection coefficient at the mold wall surface

h0shell Contact convection coefficient at the shell surface

hradgap Radiation convection coefficient across the gap

lastgfor Flag used to insure that only one gap iteration is employed

imfoul Unused variable keeping an even number of integer variables in the common block to help double precision floating point variable addressing

The variables in these common blocks are common to practically all subroutines. A few subroutines require a specific element in the array rather than the whole array. In these cases, the element is passed into the routine in the FORTRAN call statement.

A1.1.2 CCMAIN.F

The main program is the only routine contained in ccmain.f. It is responsible for directing control to the other routines, controlling the stepping through time and load steps, controlling output intervals, and iterating on the heat transfer boundary conditions. The comments at the beginning track the major changes in the code as well as discuss the major features.

The first task accomplished by ccmain is setting up of data into the common blocks and declaring variables needed only for the main program. It sets up arrays to control the time stepping procedure without the need of an external file and also sets up similar arrays to allow the user flexibility in the frequency that data is written to the output files. The print intervals control when the data is written to the binary output file for later post-processing. The plot intervals control when data is written to the special output files set up for specific desired plots.

After displaying a banner for users, the program sets up the output files and establishes the output file, killfile. The output in killfile is the digit zero as long as the program is running. The digit in this file can be changed one. At specified locations in the finite element code, this file is reviewed. If the digit one is found the program will terminate after writing out its final data set to the binary output file, binout, and after writing out a start up file, called startup, so that a successive run can initiate where the current run was killed. A list of files used in this code, their use, unit number, and contents are summarized in the table below.

Table Summarizing Files Used by Finite Element Model

Filename	Use	Unit	Contents
file2	input	2	OBSOLETE - File of codes indicating a change in time step size
file2	input	2	Initial stress-state file
file3	input	3	Input temperature file for mechanical analysis
file4	input	4	Mesh of nodes, elements, and boundary specifications
stdin	input	5	Standard input file
stdout	output	6	Standard output file
summary	output	7	Summary of temperatures at special points
runlog	output	8	Contains a log of all errors, warnings, and comments during the run
temps	output	9	Complete temperature history for every node at all time steps
prtinfo	output	10	List of iterations printed in the binary output file
taper	output	11	Narrow face mold position with 3 narrow face shell positions
wideface	output	12	Wide face mold position with 3 wide face shell positions
heatout	output	14	Comparison of average heat flow from model with experimental results
binout	output	15	Binary output file for post-processing
file16	output	16	Scratch file for output when ioutput = 1
fluxout	output	19	Heat flow file for spray cooling simulation
startup	output	20	File containing the stress-state information at the final time step
hcvout	output	21	Convection coefficient file for spray cooling simulation
narowend	output	22	Shape of shell along the narrow face at the end of the analysis
wideend	output	23	Shape of shell along the wide face at the end of the analysis
killfile	in/out	34	File used to prematurely terminate run
echoin	output	39	File which stores all user responses for input file creation
file40	input	40	Command input stream when interactive use is suppressed

After opening the output files, the main program prompts the user to issue an input file name or to execute the program in interactive mode. If the program is run in interactive mode, the file echoin records all user responses so that this file can be used as the input file on subsequent runs. The main program goes through part of the interactive process in order to establish the mesh file; it then determines if enough data space is provided for this mesh and terminates the program if space is insufficient. Control is then switched to subroutine htask.

Subroutine htask is only called if a heat transfer or coupled analysis is performed. This subroutine prompts the user for information needed during the solution of the heat transfer analysis. When control returns to the main program, subroutine strask is called if a stress or coupled analysis desired. Similar to htask, strask prompts the user for information that will be needed for the mechanical analysis. Finally control switches back to the main program where the user is prompted for information needed only for the coupled analysis or information common to any analysis. After determining the plot intervals, all the information prompted from the user is completed.

The next phase ccmain enters is setting up the arrays with the supplied information and writing initial records to selected output files. With arrays properly set, ccmain calls subroutine initial to read in the mesh and set up the needed geometric and material constants. The main program enters a loop to step through the time and loading history, once all the preliminary data for the arrays and the mesh are in place. As the time stepping procedure begins, killfile is read in to see if a request has been made to prematurely terminate the process. If so, the program skips to the steps needed to wrap up the analysis.

The time stepping technique involves incrementing the current time variable, ctime, by the time step size. This program supports a variable time step size; thus the size must be determined before ctime is incremented which is the first step of the next time step. Current time, distance down the caster, and current and old temperatures are updated according to the new time step and the time is checked against the supplied maximum time to see if the run is completed. At this point in the process, the gap between the mold and the shell is updated for the next load step.

Control transfers from the main program to the subroutine htrun which runs the heat transfer portion of the solution. Upon return, control switches to the subroutine strrun which runs the mechanical analysis. With the stress run completed and control switched back to the main program, displacements are available and control switches to the subroutine gapcalc to calculate the gap sizes. Upon return, the convergence on gap size is checked if a coupled analysis is in progress and the convergence on the displacements is less than or equal to one. This provides a method to override the gap checking procedures, which should be done in simulations where the heat transfer is not gap dependent, by supplying a number greater than one for this convergence parameter. The number of boundary nodes that fail this gap heat transfer convergence is tabulated and if this exceeds the tolerance, a message is recorded in runlog and the control reverts back to the calculation of the gaps so that another attempt is made at a converged solution between the heat transfer and stress analysis portions of the code. However, if the solution passes the convergence tests, the solution is written to binout if the print interval is satisfied. The print interval is then updated to record when the next iteration to be stored in binout and file prtinfo receives a record indicating the iteration number, current time, and current distance down the caster of the results written to binout. If the plot interval is satisfied, subroutines tdout and taperout are called to write out specific information to output files. The plot interval is updated and the control proceeds to the next time and load step.

When all load steps are complete or the maximum time is exceeded, the program writes the results for the final step to all the files. This insures that the final step is recorded even if the print and plot intervals are not satisfied. The startup file is written so that subsequent analyses can begin where this analysis has ended and file final slab shape files are generated. Finally, all open files are closed, and the program is terminated.

A1.1.3 HSASK.F

The file, hsask.f, contains subroutines strask and htask. These subroutines prompt the user for information required for the stress analysis or heat transfer analysis portions of the code, respectively. The responses can be read from the input file by these subroutines if previous responses are provided in a file. Only minimal checking for consistency of the data is performed during the interactive process increasing response time to the user. The stress analysis portion of this analysis is discussed first.

The stress analysis interactive subroutine, strask, first prompts the user for the two-dimensional stress-state assumptions to be employed in the analysis. Responses to this prompt should indicate that the plane strain, plane stress, or generalized plane strain assumption is employed. Axisymmetry has been formulated into the model but has not been tested and is likely to have errors. Therefore, this stress-state assumption should not be used at this time. If the generalized plane strain assumption is used, the user must supply a tolerance for the iterative convergence of z strain. Temperature data must be supplied in one form or another. If running a coupled analysis, the stress analysis receives temperatures from the heat transfer part of the run. However, when running a separate stress analysis run, a file must be provided describing the transient temperatures for all nodes.

The user supplies information on the area of the continuous caster that is to be simulated. The interactive procedure for the thermal processing after the mold is not covered here because it is discussed in a previous report [7]. The user is next prompted for information on the mold geometry, processing conditions, and simulation parameters. These prompts are also requested for thermal runs, so they are avoided if the analysis couples heat transfer and mechanical analyses. The file name containing an initial stress-strain distribution is stored for later retrieval, if the user indicates this file is available and necessary.

Following the prompts on the caster simulation method and initial conditions, the user is prompted for information concerning the boundary conditions. The first of these prompts asks the user how the specified surface loads are to be handled for the simulation. Surface loads can be avoided entirely, applied only at the first iteration, only at a specified time, or at all iterations. Due to the incremental nature of this model, a surface load that acts constantly through the simulation should be only applied in full at the first iteration and discarded for further iteration. A load that is applied constantly after a specific time should only be applied after a specific time. Finally, loads that vary with time must be applied at every iteration. However, only the incremental and not the total load for each step should be applied.

After instructing the code how surface loads are to be handled, the pressure from the molten pool is activated. This involves the procedure for locating the solid-liquid front and applying a pressure to the solid as described in Section 2.3.4. If this option is activated, the user is prompted for a time after which the pressure loads should be applied. This option should be set less than or equal to the initial time. Mold-shell contact loading is activated by the prompt following the ferrostatic pressure loading prompts. If the contact loading is used, the mesh must be set up so that a contact loading boundary condition is activated at those boundaries that potentially contact the wall. The user must also modify the taper subroutine, which will be discussed later, that defines the position of the wall. The wall position is taken to be at x=0 and y=0 by default. Therefore, although during the interactive portion of running the code the application of contact boundaries is quite easy, mesh and subroutine preparation must precede using this in the analysis.

The final prompts from the stress routine involve the inelastic behavior of the material. The user must first indicate which of three plasticity options are to be used: no plastic strain incorporated, the plastic strain rate function, or the total plastic strain function. The plasticity functions should describe the effective plastic strain or effective plastic strain rate as a function of variables describing the material's stress-state. Incremental strains are calculated from these functions and are supplied to Equations 2.3.2.H through 2.3.2.K. The convergence parameters for plasticity, discussed in Section 2.6.3, can also be controlled during the interactive portion of the stress analysis.

The heat transfer portion of the interactive phase of running the model is set up similarly to the stress analysis portion in that simulation parameters are determined first, followed by initial condition determination, and ending in solution technique resolution. Determining the simulation controlling parameters is identical to that used for the stress analysis portion and replaces it for coupled analyses. The user is prompted to provide an initial temperature file if desired. This temperature file will influence the old or previous temperatures, $\mathbf{T}^1$, as described in the formulation discussions. If an initial stress-state is provided for a coupled analysis, the initial temperatures on the stress-state file over ride those read in for heat transfer. Therefore, two files with identical temperature information are not required. Finally the techniques employed for the thermal simulation can be modified. These include the capacitance matrix formulation technique, the enthalpy evaluation technique, and the time stepping technique. A final option determines at which temperature the material properties are evaluated for the current iteration. The recommended techniques can be accepted by accepting the default procedures.

The completion of these two subroutines returns the control to the main program where additional interactive prompts are issued. Once these final prompts are answered, the code performs the calculations as specified by these prompts. These subroutines, contained in the hsask.f file are not accessed at any other point in the finite element program.

A1.1.4 HSRUN.F

The hsrun.f file contains the subroutines responsible for running the heat transfer and stress analysis portions of the coupled analysis. These subroutines are referred to as htrun and strrun.

The heat transfer program structure is much simpler than the stress analysis portion due to the lack of iteration and convergence required in the thermal analysis. The control from htrun is immediately transferred to a subroutine called stepdo which assembles the stiffness and capacitance matrix elements into the **[A]** matrix. This subroutine also forms the portion of the thermal force vector, **{F}**, that does not include boundary conditions. With this subroutine reverting control back to the running subroutine, another subroutine, htbnd, is called to assemble the boundary conditions into the thermal force vector. At the completion of the htbnd subroutine, the thermal matrices are completely set up and ready for solution; the subroutine vband is called upon to solve this linear system of equations using the Cholesky decomposition techniques. With temperatures available from this time step, control from the htrun routine transfers to the tavgc routine to calculate average temperatures, and finally control returns to the main program. The main program calls the strrun routine if the analysis is coupled, or increments the time step and returns control to htrun for this next time step.

The strrun subroutine has the same general procedure as the htrun routine but is appears significantly more complicated due to its iteration control. The routine begins by calling tempr, a routine to read in temperatures from a separate heat transfer run, if needed. Successive calls to subroutines subfa, varprop, and stiff, evaluate material structure, material properties, and assemble the global stiffness matrix. After the stiffness matrix is calculated, the code sets a point to where convergence iterations that only adjust the force vectors can return. This process also increments the number of convergence iterations so a running total is kept. The mechanical force vector must be zero before this vector is evaluated because the individual force vector components are added into this vector. The plastic strain rate is also determined if this portion of the code is activated. Calls to routines forcet, forcee, and forcez evaluate and sum the thermal, inelastic, and z dependent loads respectively. This force vector is stored before the load and fixed displacement boundary conditions are applied allowing the code to return to this point when gap condition iterations are required.

The strrun routine issues control to the subroutine strbnd to accumulate the effects of all the boundary conditions. When this routine returns control to strrun, the mechanical problem is completely set up. Again the vband routine is called to solve this system of equations returning incremental displacements for this load step. The gapcheck subroutine is called to evaluate the nodes on the shell that interfere with the mold. If interference is detected, the stiffness matrix and force vector are reset to the values prior to addition of the boundary conditions. The code contains two separate calls to vband; one utilizing a triangularized stiffness matrix and the other requiring triangularization.

The iteration on the shell to mold contact occurs only once if needed. The incremental displacements are the result of the solution that satisfies the shell contact tolerance. The subroutine stresp is called to evaluate nodal thermal, inelastic, and total strains for this increment of the load step. From the incremental strains, nodal stresses are calculated. Control is directed to subroutine supdate which adds the incremental solution to the overall total component stresses, strains, and displacements. The total component stresses and strains are used in routines eff, effe, and effp to calculate effective total stress, effective total strain, and effective total plastic strain respectively. These stress, strain, and displacement values completely define the stress-state of the shell for this iteration. If an inelastic solution is in progress, subroutine plastic is called to predict the plastic strain and plastic strain rate that would result from this stress-state. Subroutine check is then called upon to evaluate these properties against the properties assumed at the beginning of the mechanical load step. This subroutine also compares the incremental plastic strain to the overall total strain to evaluate this against prescribed convergence criteria. If convergence has not occurred to the satisfaction of the criteria, a sub-iteration or a convergence iteration can be performed to achieve convergence. If either convergence scheme is necessary, subroutine sddate is called to remove the incremental values from the totals for this load step which has not converged. However, if the convergence criteria are fulfilled, the load step is finished and control reverts back to the main program to evaluate the heat transfer convergence criteria.

A1.1.5 INITAL.F

The inital.f file contains subroutines necessary to initially define and set up the finite element problem to be solved; these routines are used only once. The first subroutine in this file is the subroutine initial. This subroutine calls input to read in the mesh, and next it calls conprop to define material constants that do not vary with temperature or stress-state. Setup is the next subroutine to be accessed. This subroutine determines constants associated with the element formulation, areas of each triangle, and vectors normal to the boundary edges. For mechanical or coupled analyses, the material structure, temperature and stress-state dependent properties, and effective stress and strains are all assigned with calls to the following subroutines: subfa, varprop, supdate, eff, effe, effp, and plastic. Finally, the initial state is written to the binary output file, the temperature history file, and the temperature summary file with calls to boutpt and tdout.

The subroutine input is the next subroutine appearing in the inital.f file. This subroutine reads in the nodes, elements, and boundary conditions from the mesh file. When reading in the nodes, it reads in the node number and the x and y coordinates of the node. The format on this read statement is (i5,2f10.4). The elements are read in as element number, node1, node2, node3, and element type flag in the format of (5i5). The element type flag can be used to separate material differences throughout the finite element domain. Input of boundary conditions consists of reading in the boundary condition number, two nodes on the boundary line, heat transfer boundary condition type, heat transfer boundary condition position, mechanical boundary condition type, and mechanical boundary condition position. The flags for boundary condition type and positions are set to different integer values to indicate which boundary condition is to be applied. This is explained more in the section of this report discussing the input. These boundary flags are read in under the format of (7i5). The input file also computes the irh and irs vectors once the mesh has been read. These vectors indicate the position of elements of vector **[A]** relative to their position in the full stiffness matrix which is not stored to reduce FORTRAN storage requirements. These vectors are used by the solution procedure.

The setup subroutine calculates geometric constants needed later in the analysis. The procedure begins by setting the total area to zero so that it can accumulate all element areas. Maximum and minimum coordinate values are determined. The $\mathbf{b}_i$ and $\mathbf{c}_i$ values from Equations 2.1.1.B and 2.1.1.C are computed as is the area of the triangular element. The area of the element is checked to insure that it is positive, indicating the element specification is correct. If it is not, the code corrects the numbering sequence and recalculates the geometric constants. This prevents the element ordering specification from influencing the obtained results. After the element passes this test, the nodal areas are calculated. Constants are then calculated to allow the program to locate the off-corner region and the narrow and wide face centers on the mesh. These calculations require that the corner be located at the coordinates (xmin,ymin) while the narrow face center is located at (xmin,ymax) and the wide face center is located at (xmax,ymin). The contact surface procedure discussed in Section 2.5 depends on this mesh orientation. Additional constants are defined for the output files associated with the contact surfaces. Finally the boundary lengths and normal vectors are calculated for boundary conditions that do not specify fixed displacements. Upon completion of these boundary constant calculations, the control returns to the initial subroutine which calls the setup subroutine.

A1.1.6 PROPS.F

The props.f file contains subroutines and function subprograms that define the material properties. The material properties used for these analyses are discussed in the next main section. The first function provided calculates the thermal conductivity based on the incoming nodal temperature and current element type. The variable rk must contain the specified conductivity when exiting the subprogram. Another function subprogram provided is to calculate the enthalpy of the material. The variable h must contain the enthalpy when the function returns control to the calling routine. Similar functions are provided for the elastic modulus and thermal linear expansion. The elastic modulus returns the variable emod and the thermal linear expansion returns the value of ctle. A conprop subroutine is also provided in this model. This routine is accessed only once from the subroutine initial and provides the values for the Poisson ratio, density, liquidus temperature, and solidus temperature which are not dependent on the temperature of the material or the state of stress.

The function subprogram called eprate evaluates the effective plastic strain rate based on the current amount of total effective plastic strain, effective stress at the node, temperature, material structure, and the current time from the start of the analysis. This function contains the constitutive relations developed by Azzi [6]. The variable eprate must contain the effective plastic strain rate when control is returned to the calling routine. Another function subprogram is called epconst. This function defines the total effective plastic strain and can be used instead of the plastic strain rate function providing the user with the flexibility to use either formulation to input the plastic strain behavior of the material. The available parameters on which to base this function are current plastic strain rate, effective stress, temperature, material structure, time, and time step size. The value of epconst contains the total effective plastic strain under the stress-state provided by the incoming variables. The model references this function under both the current and previous conditions to obtain the incremental effective plastic strain.

The function uts is no longer used in this model. It calculates the ultimate tensile strength of the material under the thermal and mechanical conditions. This allows stresses to be represented as a portion of the available strength. The subroutine subfa is used to determine the material state. For analyses with plain carbon steel [7,10,18,19], this function would calculate the fraction of ferrite and austenite current making up the material matrix structure. This would then influence the material behavior. This analysis models the casting of AISI-304, an austenitic stainless steel. This material solidifies into an austentic structure and remains austentic at all temperatures; thus this routine is not used for the current model. However, the facility to adjust material behavior and properties based on the microstructure has already been built into the code.

The final subroutine in the props.f file is used only for mold simulations of the continuous caster. This routine, called gapprops, evaluates the thermal conductivity properties of the materials in the gap between the shell and the mold. With these properties the equivalent heat resistance across the gap is calculated, see Figure 5, upon which heat flow calculations are based. Information used for these calculations can include all the common block information, and incoming variables for the temperature of the shell, temperature of the mold, x coordinate, y coordinate, heat transfer boundary condition flag, node number, node number on the boundary, and boundary edge number. The variable rkair determines the thermal conductivity of the gaseous material that fills in voids between the powder and the mold when large gaps exist. The variable rkpow contains the thermal conductivity of the powder material, and h0mold is the contact convection coefficient at the mold interface. Temperatures for the state of the casting powder are defined: tcrys defines the crystallization temperature, and tsoft defines the softening temperature. Corresponding to these temperatures are the contact convection coefficient at the powder interface with the shell of hcrys and hsoft. Contact convection coefficients are also defined for the powder at the metal liquidus and solidus temperatures, hliq and hsol respectively. Minimum and maximum powder thicknesses are specified as well as the radiation emissivity of the shell and the mold surfaces. From these values, an effective convective coefficient for radiation, hradgap, is calculated. This calculation assumes base units are Celsius for temperature, Watts for power, and meters for length. The Stefan-Boltzmann constant should be modified for other sets of consistent units. The contact convection coefficient at the shell interface is calculated based on the current temperature and the temperature dependent contact convection coefficient of the powder.

The props.f file contains all the material property information and constants associated with the simulated material and the gap material. The user must provide functions describing the variability of these properties in this section. The code must then be compiled with the correct set of material properties to be simulated. The compilation of FORTRAN source code is system dependent and is not discussed here.

A1.1.7 HTSUB.F

The file htsub.f contains subroutines associated with the heat transfer analysis. Most of these files are exclusive to a heat transfer analysis, but some that influence both the thermal and mechanical analyses are located in htsub.f. The first subroutine is tavgc which calculates the average temperature of each element and the average temperature over the entire finite element domain. The subroutine tdout is the next subroutine appearing in the htsub.f file. The tdout routine write special temperatures to the summary file and writes all temperatures to the temps file.

The subroutine stepdo assembles the stiffness and capacitance matrices into the [A] and {F} vectors. This subroutine contains all the formulation options requested in the interactive procedure. Initially, this file zeroes out the [A] and {F} vectors so that the thermal load components can accumulate in these vectors, element by element. The first formulation option encountered is that which specifies at which temperature the material properties should be evaluated for the current iteration. These temperatures are computed in the desired manner and the thermal conductivity and enthalpy properties are evaluated. Gradients of enthalpy and temperature are computed using the geometric constants, $\mathbf{b}_i$ and $\mathbf{c}_i$. The enthalpy of the element is then calculated by the selected manner. Using the formulation for the capacitance matrix selected, the elements for the stiffness and capacitance matrices for each node of the element is computed and loaded into the solution vectors according to the selected time stepping technique. The result of the subroutine is the completely assembled stiffness and capacitance matrices, in [A], and the thermal load vector, {F}, without the boundary condition effects.

The subroutine following stepdo is fixdof which arranges the solution vectors to insure that a degree of freedom is set to a specific value. This subroutine is accessed by both the heat transfer portion of the code to fix temperatures and the stress analysis portion of the code to fix displacements. This subroutine is passed the two solution vectors, the value a degree of freedom will be fixed to, the degree of freedom fixed, the half bandwidth, number of degrees of freedom in the {F} vector, and the number of elements in the [A] vector. The code determines the location of the degree of freedom in the {F} vector and performs row operations on the stiffness matrix in the [A] vector to insure the coefficient of this degree of freedom is unity and no other terms exist for this equation. With the [A] vector arranged in this manner, the {F} vector is set value desired for that degree of freedom. This procedure is described by Cook [21].

The subroutine taperout follows fixdof and is used to print out files specifically devoted to the shell-mold interference problem and is only activated for coupled analyses. This subroutine calls the subroutine gapcalc to update the current boundary gap calculations. It successively calls the taper subroutine to determine the mold wall position, calculates the gaps with a call to gapcalc, and writes them to a file. The subroutine gapcalc to update the gap predictions is also included in the htsub.f file. This routine loops through each of the two nodes on all boundary edges. By calling taper at each of these nodes to determine the position of mold wall opposite the shell, the gap at each node of every boundary edge is calculated using the total displacements of the node. With the gaps calculated and stored in the array bndgap, the code returns with updated gaps at the boundary elements.

A1.1.8 HTBND.F

The file htbnd.f contains subroutines associated with determining the heat transfer boundary conditions and assembling them into their respective locations in the {F} vector. The file begins with subroutine htbnd. The subroutine sets the variables qtotp and qtotf to zero so that they may total the heat flow out of the finite element domain at every step. Zone calculations are performed for spray zone simulations but ignored for other types of simulations. A loop evaluates the boundary conditions at every boundary edge specified in the mesh file. Based on boundary edge type and position which are explained in Section A1.2, the code determines what part of the boundary condition file applies to each boundary condition. For the specified heat flux boundary condition, subroutine heatflx is called to determine the heat flux, qflux, at the boundary. With this value it determines the component of the {F} vector as described in Section 2.1.5, and adds this component into the thermal force vector. The specified heat convection coefficient boundary condition calls the subroutine htconv to obtain values for the heat convection coefficient and ambient temperature. These values are returned to htbnd from which a heat flux is calculated based on Equation 2.1.5.A.

The interfacial gap heat transfer boundary conditions are handled identically to the specified heat flux case except additional calculations are provided to determine the overall heat convection coefficient and the ambient temperature; these calculations are summarized below. First the position flag determines if the boundary condition on the wide face, narrow face, or one of the two cut boundary conditions. If the boundary is determined to be on the cut face, this routine insures this face is adiabatic. If the node is determined to be on the wide face, the ambient temperature is determined by referencing the function subprogram temold. This function provides the spatial variation of temperature for the entire mold surface, and will eventually be supplied by Azzi's heat transfer analysis of the mold [6]. Subroutine gapprops is called to determine the properties of the material filling the gap between the shell and the mold and the contact convection coefficients associated with these materials. From these properties, an equivalent heat flow resistance is determined by the analogy shown in Figure 5. The temperature difference across the gap is calculated using the mold temperature available from the temold function and the temperature at the node from the previous iteration.

This information calculates the heat flux from the current boundary edge on the wide face. Calculations for the narrow face proceed identically to those of the wide face. In this manner a heat flux is determined for every boundary edge on the shell, and the heat flow is accumulated into the thermal force vector. The heat flow is also accumulated into the qtotp variable which tracks the total heat from the perimeter.

The final boundary condition type handles radiant heat transfer. This procedure calls a subroutine htrad which calculates an equivalent heat transfer coefficient and ambient temperature given the temperature of the node at the boundary. This equivalent heat transfer coefficient linearizes the radiant heat transfer by using the past load step's value for temperature at the current node. This thermal load is assembled into the thermal load vector.

If a fixed temperature boundary condition is encountered as the code cycles through every boundary condition, it is skipped over to apply the other boundary conditions first. After all the heat flow boundary conditions are applied, the fixed temperatures are applied by performing the matrix manipulations within the fixdof subroutine. These specifications on the degrees of freedom must be performed after all other boundary condition types to insure the matrix manipulations are not altered before entering into solution phase.

Finally, after fixing the temperature specified boundary conditions, the code calculates the experimentally determined heat flows from the perimeter from curve fit results from Samaresekera [22]. These numbers, as well as the equivalent calculation determined from the finite element results, are written to the special output file heatout. This file can be used to compare the experimental and model predictions for the heat flow from the caster.

The other subroutines included in file htbnd.f are routines that define the magnitude of the heat transfer for the various boundary condition types: heatflx returning qflux, htconv returning hconv and tinf, heatgen returning qgen, htrad returning hrad and tinf, and fixtemp returning temp. The taper subroutine is used to define the mold wall position, stored in the variable wall, to account for both the taper and the mold distortion, with variable distor. Subroutine temold is also included in the htbnd.f file. This routine defines the mold temperature on an absolute scale.

A1.1.9 STRSUB.F

Like the htsub.f file, the strsub.f file contains the subroutines specifically associated with the stress analysis portion of the code, except for the application of the boundary conditions. The first subroutine in the file is called boutpt which is responsible for writing all results to the binary output file for post-processing. This subroutine does not receive data from the common block; instead information is passed into this routine. The data written to the binary file are current time, iteration number, temperature, total u-displacement, total v-displacement, four components of total stress, four components of total strain, four components of total plastic strain, effective total stress, effective total plastic strain, effective total strain, effective strain rate, ultimate tensile strength at the current stress-state, total effective stress divided by the ultimate tensile strength, total thermal strain, fraction ferrite, and thermal linear expansion.

The subroutines eff, effe, effp follow subroutine boutpt. These routines use the Von Mises criteria to evaluate an effective stress, effective strain, and effective plastic strain, respectively [16]. The difference between the effective strain and the effective plastic strain routines is that total strain in one direction effects the orthogonal strains according to the Poisson ratio. Plastic strain in one direction, on the other hand, effects the orthogonal strains in a manner to maintain the total volume of the loaded body, which requires a plastic flow Poisson ratio of 0.5 [16].

The effective subroutines are followed by the function subprogram itrans. This function is used to relate any degree of freedom to its respective location in the **[A]** vector. Function dasinhx is a function used to calculate the arc hyperbolic sine in double precision which is no longer used. This subroutine check is one of two that evaluates the inelastic convergence criteria. The check routine evaluates the difference between the previous inelastic strain rate with the current inelastic strain rate, finds the maximum anywhere in the finite element domain, and stores this value in the variable cedmax. The limits on the convergence criteria are provided by the user and stored in variables difsub and difcnv. If cedmax exceeds difsub, a sub-iteration will be performed to converge on the material inelastic behavior in two smaller load steps. However, if cedmax exceeds difcnv but is within the limits of difsub, a convergence limit is performed to converge on plastic properties. If cedmax is within the both of these limits, subroutine check does not force an iteration for plasticity convergence.

The subroutine zero follows subroutine check in the strsub.f file. This subroutine is a special application of the subroutine fixdof in the htsub.f file. Because structural analyses often incorporate many degrees of freedom that are fixed to zero, this subroutine applies a quicker method of applying a zero value fixed degree of freedom to the solution vectors. This technique is also discussed by Cook [21].

The subroutines appearing after zero in the file contain the formulation of the mechanical analysis assembling various component load vectors into the total mechanical load vector, {F}. The first of these subroutines, forcet, assembles the thermal load vector. This routine defines stress-state dependent constants, calculates thermal load vector components by Equations 2.3.2.D through 2.3.2.G, and assembles these components into the full mechanical load vector.

Subroutine forcee separates out the component incremental plastic strains from the Prandtl-Reuss equations and determines the plastic load vector components from Equations 2.3.2.H through 2.3.2.K. To do so, it first sets up constants based on the two-dimensional stress-state assumptions used. Next, it loops through the nodes determining the component incremental plastic strains. From these it loops over the elements to find the average plastic strains in each component direction for the element, provided the element is not molten. These average elemental strains are distributed between the nodes according to the geometric constants, and loaded into the total mechanical load vector.

Subroutine forcez computes the components needed in the force vector resulting from eliminating the z displacements from the field variables. This subroutine defines constants depending on the two-dimensional stress-state assumption, and loops through all the elements finding the temperatures of all the nodes. Using the z component of the incremental plastic strain computed from forcee and the z strain satisfying the generalized plane strain assumptions if used, the component of the mechanical force vector is computed using Equations 2.3.2.L and 2.3.2.M. These components are assembled into the total mechanical force vector preparing this vector for boundary conditions.

Following these subroutines that assemble the force vector, subroutine tempr reads in temperatures if a coupled analysis is not performed. When this subroutine is used, it expects a temps file from a previous thermal solution. This file cannot be called temps, however, because the temps file must be available to be written to before the analysis type is selected. The tempr subroutine begins by assigning the current temperatures to the past temperatures. The analysis time at which the thermal solution was written to the temperature history file is read and compared to the current time for the stress analysis. If the time on the temperature file is smaller than the current time, the mechanical analysis will skip this set of temperatures, read the next time and temperatures and compare these to the mechanical analysis time. If the time on the temperature file is greater than or equal to the time in the analysis, the mechanical time step size is altered to reflect the true step size and the temperatures at this time step are used. In this way, the mechanical analysis cannot have a time step smaller than the thermal time step, but is allowed to skip many thermal steps. For the highly coupled continuous casting mold simulations it is not advisable to skip thermal steps to maintain a high level of confidence in the results.

The subroutines supdate and sddate follow the tempr routine. The supdate routine takes the results from the incremental step just completed and adds them into the total results through the previous step. After the supdate routine is completed, convergence criteria are checked using the totals that have been updated and the effective stress and strain values from these updated totals. If convergence fails and the solution must be attempted again, the incremental results must be subtracted from the totals to avoid adding in a load step which has not converged. The sddate routine performs this task of subtracting the increments from the totals for an unconverged load step. These routines add in the results in the mechanical force vector into the total displacements. This must be done because these routines are run only after solution at which time the force vector contains the resultant displacements, not the mechanical load components. The sddate routine also calls the subroutines eff, effe, effp to recalculate and update the effective stress and strains.

The subroutine stiff follows the supdate and sddate routines; it is responsible for assembling the mechanical stiffness matrix. Its first step is to define constants according to the two-dimensional stress-state assumptions. Following this, a loop over all the elements finds the average elastic modulus for the elements and calculates the $\mathbf{b}_i\mathbf{c}_j$ terms for the three nodes of the element. Utilizing the itrans function dictating where the ij element in the stiffness matrix belongs in the **[A]** vector, the stiffness matrix is assembled into the **[A]** vector for solution. When this process is complete for all the elements, the flags indicating the change in status of the boundary conditions are all reset to zero. A copy of the **[A]** vector is also stored so it can be recovered in its original state after fixed displacement boundary conditions have altered it. The triangularization flag is set to indicate that the stiffness matrix has not yet been triangularized, and the stiffness matrix formulation is complete.

The stresp subroutine, next in the strsub.f file, numerically integrates the incremental displacements to yield incremental strains. First the routine sets all incremental stress and strain values to zero so they may accumulate the values for the iteration which has just completed solution. Looping through each element individually, the liquid or solid state of the material is evaluated. If the element is found to be liquid, no stress or strain is accumulated into this element. Material properties are evaluated for all solid elements, and the displacements are found in the force vector. The strains are produced from the approximate differentiation of the displacements with respect to x and y. This differentiation utilizes the differential geometric constants, $\mathbf{b}_i$ and $\mathbf{c}_i$. Stress-strain relations under the two-dimensional stress-state assumptions are applied to achieve stresses from the derived strains [16]. If the generalized plane strain assumption is employed, the z stress is numerically integrated of the area and the z strain is calculated to minimize the z stress. Convergence criteria are evaluated on the z strain. Once this strain converges, the elemental stresses computed on each element is separated into the nodal components based on nodal area. Therefore when stresp is completed, the incremental stresses and strains are available from the current iteration.

Following stresp in file strsub.f is subroutine plastic. This subroutine, along with subroutine check, control the iteration on the inelastic behavior. The plastic subroutine defines the plastic properties for the displacements, stresses, and strains, for the current load step. When this routine is entered in elastic analyses, the plastic strain rate and incremental effective plastic strain are set to zero and the control returns to the calling routine. However, if the analysis involves inelastic behavior, the choice of the supplied plasticity function is evaluated. If the plastic strain rate function is supplied by the user, the function is evaluated for the current conditions to yield a plastic strain rate. This plastic strain rated is multiplied by the time step size to yield the incremental effective plastic strain. If the total plastic strain function is provided, this function is evaluated at the current conditions to yield the total plastic strain. The previous value for this function is subtracted from the current value to yield the incremental plastic strain. This is divided by the time step size to give the plastic strain rate. In this way, both the plastic strain rate and the incremental effective plastic strain are calculated from this subroutine. These properties can be compared against the convergence criteria to flag the need to iterate if convergence is not satisfied for all criteria.

The next subroutine is called varprop. This subroutine is called at the start of each new iteration or sub-iteration. This subroutine evaluates the mechanical material properties which are temperature but not stress-state dependent including elastic modulus and thermal linear expansion. Every time varprop updates these properties, the stiffness matrix must be reassembled.

The gapcheck subroutine is the final subroutine in the strsub.f file. This subroutine contains the contact loading convergence criteria calculations described in Section 2.3.4. The position of the mold wall at the current iteration and again and the next load step are calculated, assuming the load step size will not change. The difference between these two values indicates how much the mold wall moves within the load step. One quarter of this difference is used as the tolerance on the interferences as well as the tolerance on determining the maximum interference. This tolerance is computed for both the narrow face and the wide face and are stored in variables gaptol1 and gaptol2 respectively. Flags are initialized for counting the number of boundary edges interfering with the mold wall, storing the minimum gap (or the maximum interference), and saving the position of the minimum gap.

With the initialization of these flags, the boundary contact is evaluated. If the boundary condition type indicates the boundary edge is on the narrow face surface, the subroutine taper is called on both nodes on the boundary edge to determine the mold wall position. This position and the total displacement vectors allow calculation of the gaps at each node. If the surface interferes with the mold wall, the interference is counted and the gap size and the position along the narrow face is compared to the minimum. The procedure continues for all contact surfaces along the narrow face. At its completion, the minimum gap, gapminn; the location associated with this minimum gap, yofgmin; and the minimum location associated with any interference, ygapmin, are all known. These values are used to determine the areas in which interference occurs and which nodes need mold constraints as depicted in Figure 9.

The gap in the center of the narrow face is compared against gaptol1. If interference is not detected, as in Figure 9a, all nodes that interfere along the narrow face should be fixed to be within the mold wall position. Each boundary node's displacement is compared against the mold wall position from subroutine taper and gaps are calculated. If either node is determined to interfere with the mold wall by more than gaptol1, the jchngdof flag is changed to 1 to indicate that this node will be fixed to be within mold wall on the next iteration.

As in Figure 9c, if the gap in the center of the narrow face does indicate interference, but it is not the maximum interference, a variable yoffix is set equal to the y coordinate of the node with the maximum interference. Similarly, if the gap at the center of the narrow face indicates interference exists and that the center has the maximum interference, as in Figure 9b, the variable yoffix is determined to be half way between the corner and the first node which interferes with the mold. Boundary edges on the narrow face are flagged if the boundary interferences and has a y-coordinate greater than yoffix.

Similar calculations insure the wide face boundary edges do not interfere with the mold wall. The interfering nodes are flagged the same way as the narrow face nodes, by changing the value of jchngdof for that node from zero to one. Finally, before the gapcheck subroutine returns control to the main program, the jchngdof array is checked to determine if an iteration on the contact condition is needed. If the iteration is required, the mechanical force vector is restored to its value before the contact boundary conditions are added, the mechanical stiffness vector is restored to its values before any fixed displacement boundary conditions were attached, itriang is set to zero to indicate the stiffness vector has not been triangularized, and a message is written to the log file indicating how many boundary edges on each face require the iteration. These last steps set the matrices for the iteration on the mechanical contact conditions which is initiated when control returns to the calling subroutine.

A1.1.10 STRBND.F

The final file in the source code of this finite element model is strbnd.f. This file applies the boundary conditions for the mechanical model by accumulating them into the mechanical force vector. The length, mechanical boundary condition type, the mechanical position flag, the two nodes and their coordinates, and their respective locations in the mechanical force vector is attributed to each boundary condition. If the boundary condition type indicating that a surface load is to be applied, the prompt from the user indicating how surface loads should be handled is checked. If the check against the user response and the iteration and time of load step are consistent, the program calls subroutine fsurf to determine the appropriate load to apply to the surface. The components of the mechanical load vector are accumulated into this vector according to Equations 2.3.2.A and 2.3.2.B.

Ferrostatic pressure loads are applied next if the user has indicated that these loads are to be used and if the time provided by the user has been exceeded. The T^* variable, explained in Section 2.3.3, is defined for a particular fraction solid and the magnitude of the ferrostatic pressure is determined through a simple application of static fluid equations [16]. As discussed in Section 2.3.3, all elements are checked individually to determine those that have exactly two nodal temperatures greater than T^* and one less. When one such element is found, the two hotter nodes are stored as nbd1 and nbd2, while the third colder node is stored as nbad. The pressure load is applied between nodes nbd1 and nbd2, so the length of this element edge is determined as well as their respective locations in the mechanical force vector. Geometric calculations are used to determine the direction and components of the pressure load insuring the load is in the direction of the cold third node. The magnitude is calculated from the element edge length and the magnitude of the pressure. These components are accumulated into the force vector, and the next element satisfying these criteria is found until all elements have been checked.

Adding fixed displacement constraints including the symmetry boundary conditions and the mold wall interference restraints start of the completion of the ferrostatic pressure loading. The mold restraints are applied first. Currently, these calculations are performed each time the boundary conditions are applied, however they are not used for the first iteration because the jchngdof variable indicates that no mold restraints are to be applied on the first iteration. To apply the mold wall restraints, the file determines the appropriate boundary length type and position and stiffness vector locations. The procedure then goes through all the boundaries on the wide and narrow faces, determines the mold wall position for the current time and the mold wall position for the next load step. An offset is then defined to insure the node will be fixed inside the mold wall position on the next step. Checking each node individually, the procedure checks the jchngdof variable and that the stiffness vector has not been triangularized. If both of these indicate that the restraint needs to be applied, the incremental displacement, fixdisp, is calculated so that the total displacement at the end of this step will be just inside the mold wall position. The subroutine fixdof is called to perform the matrix operations necessary to fix this degree of freedom. This procedure continues for all boundaries on the wide and narrow faces. The fixed displacement conditions to eliminate rigid body motion or to apply symmetry constraints are applied next. This procedure involves calling subroutine zero in the strsub.f file.

The final subroutine of the strbnd.f file is subroutine fsurf. This routine allows the user to supply equations calculating the incremental surface forces which are specified for the surface loading boundary condition type. This routine does not have the common block information included for access but this could be added if needed for the calculations. The variables sx and sy that are passed out of this routine should contain the x and y components of the surface traction load respectively.

The completion of the file strbnd.f completes this section describing the finite element source code and how it is used to perform the needed calculations. This description and the

FORTRAN source code should be all user's need to understand the calculations involved, the different options available to choose, and the implementation of the finite element theory used for this application to the continuous casting solidification. It should also guide additional user modifications to help this tool grow to be more useful to the design of the continuous casting machine and the optimized process of continuous casting.

A1.2 FINITE ELEMENT USER INPUT

The input for this finite element model come from three different sources. First, the interactive prompts or the command file containing the replies to the prompts to initiate and guide the solution in the desired manner. Secondly, the material property and boundary condition files can be changed by the user to modify the material simulated and the loads applied during the simulation. Finally, the program accepts information in the form of files necessary to its operation. The interactive procedure is relatively self-explanatory so will not be discussed here. If additional information is needed about a particular prompt, the previous sections describes in detail, the resulting directions the finite element model takes with each prompt.

A1.2.1 Input From External Files

The input coming in from external files has been reduced considerably from previous versions of this mathematical model. Most of the files that bring in information are used to communicate results from previous runs which will be helpful for the current analysis. These include the initial stress-state file, the initial temperature file, and the temperature history file. The initial stress-state file is produced as the file called startup, however this file is only written for the final load step. If an initial stress-state file is needed for any step in a previous analysis, the post-processor can produce this file. The initial temperature file, for the thermal analysis, and the temperature history file, for the mechanical analysis, are produced by the model as the file temps.

The most important information that comes in from an external file is the mesh. This file is assigned to FORTRAN unit 4 when the user issues the name of the mesh file during the interactive session. The user must have a mesh file prepared before running CONCAST. The first line of the mesh file must contain a character string of no more than 80 characters that should describe the mesh. The second line, read under the (4i5) format, should contain the number of nodes in the mesh, the number of elements in the mesh, the number in the half bandwidth with the elements numbered as provided, and the number of boundary conditions specified. The third line of the mesh file, read under the (11i5) format, should contain the node number in the corner and the 10 other node numbers for the temperatures desired on the summary file. These lines are read during the initial phase of the main program.

The actual mesh data is read during the operation of subroutine input. This begins by reading the fourth line of the mesh file. This line as well as lines for all other nodes, are read under the (i5,2f10.4) format. These lines contain the node number and the x and y coordinates of the node, respectively. This procedure continues for all nodes. The corner of the shell must be located at (xmin,ymin) and that the narrow face runs along x=xmin while the wide face runs along y=ymin. The narrow face center must be located at (xmin,ymax) and the wide face center is located at (xmax,ymin). These restrictions on the mesh layout is required by the interfacial contact convergence scheme. If the contact procedure is not used, the mesh does not have to follow the above restrictions. However, all meshes must not skip any node numbers, but these numbers do not have to be in order in the mesh file.

Once all the node information is input, the element information is read. The format for this process is (5i5). The element information line should contain the element number, node1, node2, and node3 for the element, and the element type. The element type is passed to all files evaluating or using material properties so it can be used to separate portions of the finite element domain consisting of materials with different properties. There is no limit on the number of element types used in the finite element analysis.

The boundary condition information is read following the element data. The boundary information consists of the boundary number, node1, node2, thermal boundary condition type, thermal boundary condition position, mechanical boundary condition type, and mechanical boundary condition position. These are read under the format of (7i5); the last two fields do not need to be filled for meshes used only for thermal simulations. The flags to differentiate the thermal boundary conditions are explained below.

Type	**Thermal Boundary Condition**
1	Specified heat flux across the boundary length
2	Specified heat convection across the boundary length
3	Specified heat generation at the element number in the node 1 field
4	Specifies a fixed temperature for the node number in the node 1 field
5	Specifies an interfacial heat transfer gap across the boundary length
6	Specified radiant heat transfer across the boundary length

The position type can be used to help the user develop the subroutines to determine the magnitude of the load at these boundaries. For instance, the position field is used in this analysis to specify which face of the mold the boundary is on. Position 1 is associated with the narrow face, position 2 with the wide face, and positions 3 and 4 are the adiabatic boundaries along the cut planes. The position variable on the mechanical boundaries are used in a very similar fashion. The mechanical boundary types are summarized in the table below.

Type	**Mechanical Boundary Condition**
1	Displacement fixed to zero at the boundary
2	Specified surface load applied across the boundary
3	Interfacial mold contact along the narrow face
4	Interfacial mold contact along the wide face

Mechanical boundary condition type 1 has a specified set of position flags. If the boundary condition type is 1 and the position flag is 1, the u-displacements are set to zero. When the position flag is 2, the v-displacements are set zero, and with the position flag set to 3, the u and v-displacements are set to zero.

More specifications can be added when the need arises to perform calculations to determine the amount of heat flow or load is to be applied to a surface. The specifications of the boundary conditions for the mesh, completes the mesh file. This file is the only input file that must be produced by the user. A simple mesh generation program has been developed and explained by Haegele [7] to help users develop these files.

A1.2.2 Input Through The Finite Element Source Code

The material properties and the magnitude of the boundary conditions must be specified within the FORTRAN source code. The material properties that are constant do not need their own subroutines but are combined into one called conprop. The values used for this simulation are 0.30 for Poisson's ratio, 8000 kg/m^3 for the density of AISI-304 stainless steel, 1454°C and 1399°C for the liquidus and solidus temperatures respectively. The input of the material properties is discussed in Section A1.1.6. The material property functions used for this analysis for thermal conductivity, enthalpy, elastic modulus, and thermal linear expansion can be found in the FORTRAN source code. The plastic strain rate function [6] is also included although these results did not employ the inelastic effects.

The boundary conditions are defined in the finite element code similarly to the material properties. The file strbnd.f only contains one such subroutine, fsurf, which defines the incremental surface traction forces. These forces were not used in this analysis. The htbnd.f file contains functions for the heat flux, heat convection, and heat radiation across the boundary. These functions are described in more detail in Section A1.1.8. The input functions of interest to the analysis of the continuous caster include subroutine taper and function temold. The taper subroutine includes a function for the mold distortion and the undistorted mold wall position. The temold function defines the spatial variation of the mold surface temperature. The functions used for the continuous casting simulations are provided with the source code.

The information presented in this third chapter has described the FORTRAN code that implements the theories presented in the second chapter. The details of the variable names and the purpose of the structure of the code are presented to provide a guide to future modification of this mathematical procedure. The input to the code is discussed to ease the use of this tool. The combination of these data provide the user's guide for the CONCAST mathematical modelling procedure.

APPENDIX 2

STANDARD TAPERS

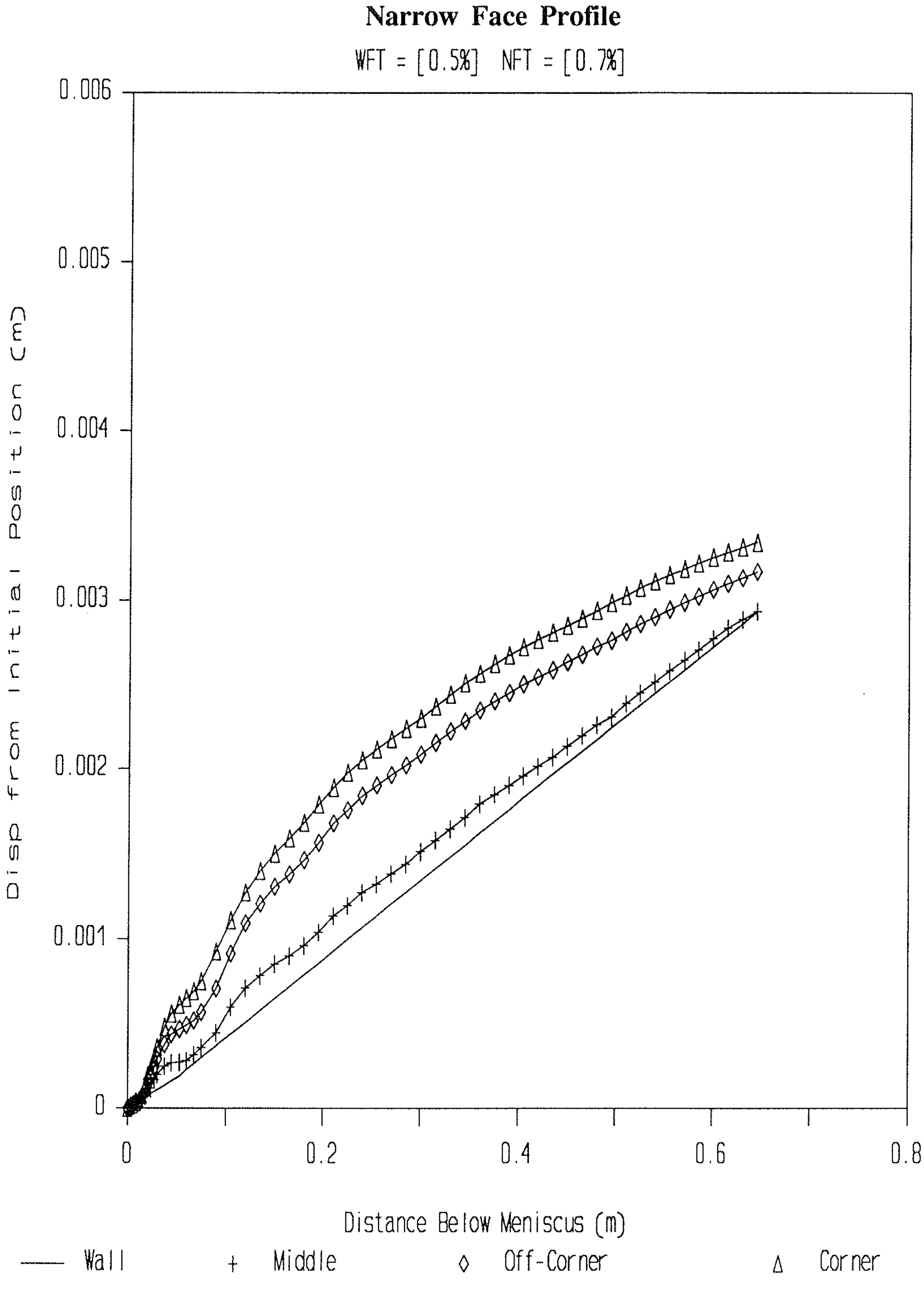


Figure 29 The shrinkage of the narrow face is displayed in relation to the mold wall position.

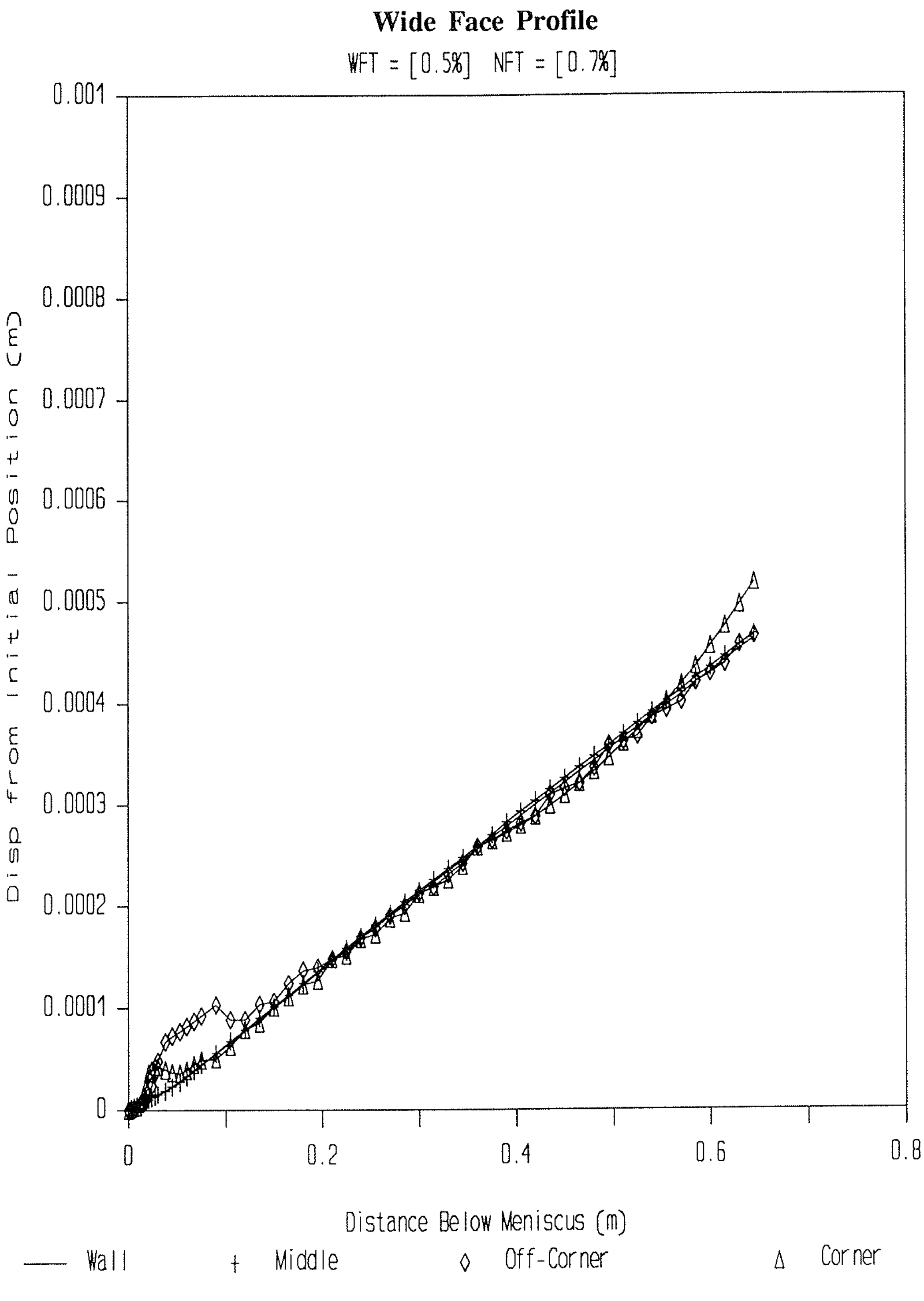


Figure 30 The shrinkage of the wide face is displayed in relation to the mold wall position.

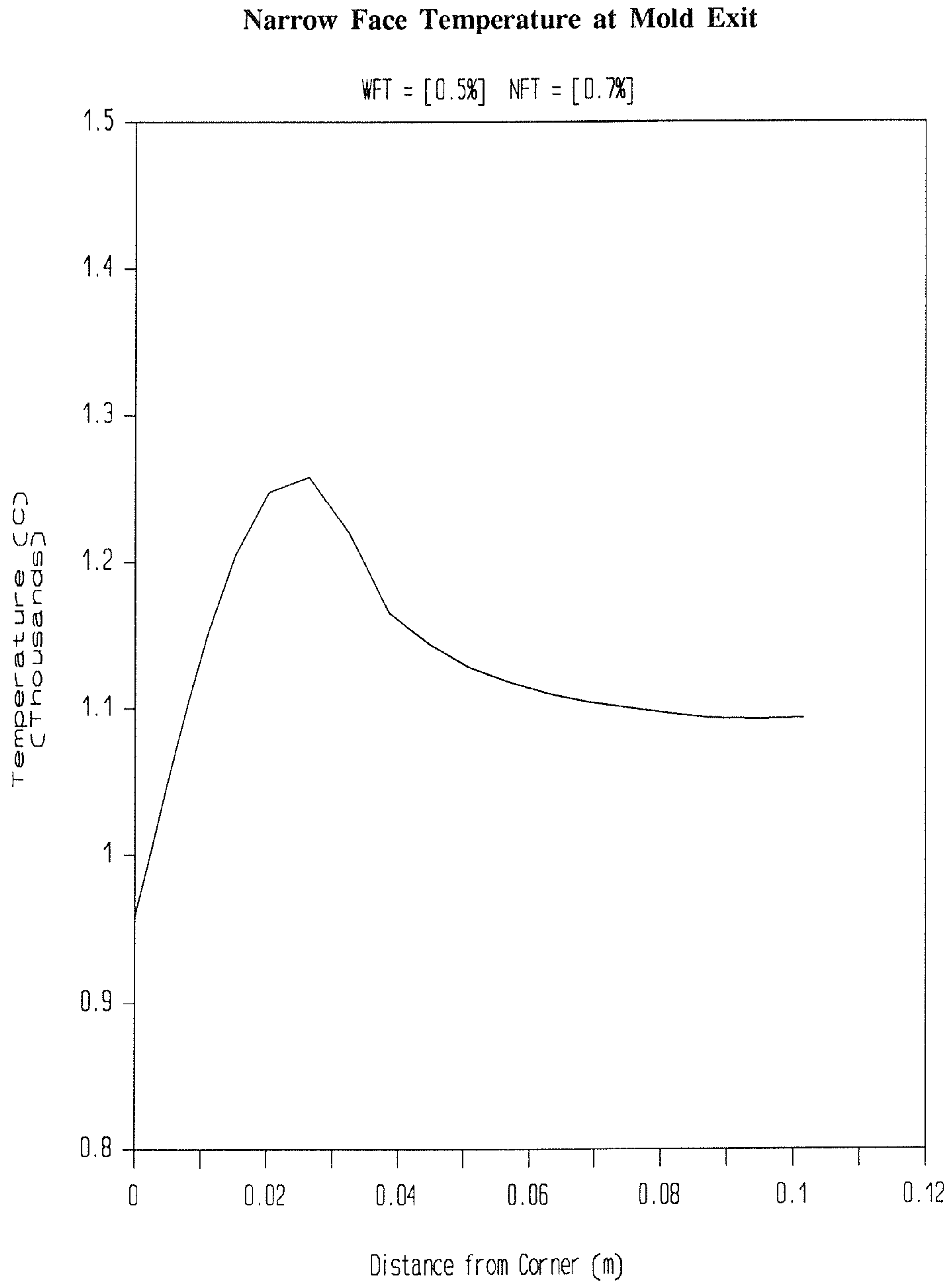


Figure 31 External surface temperatures at mold exit are shown along the narrow face.

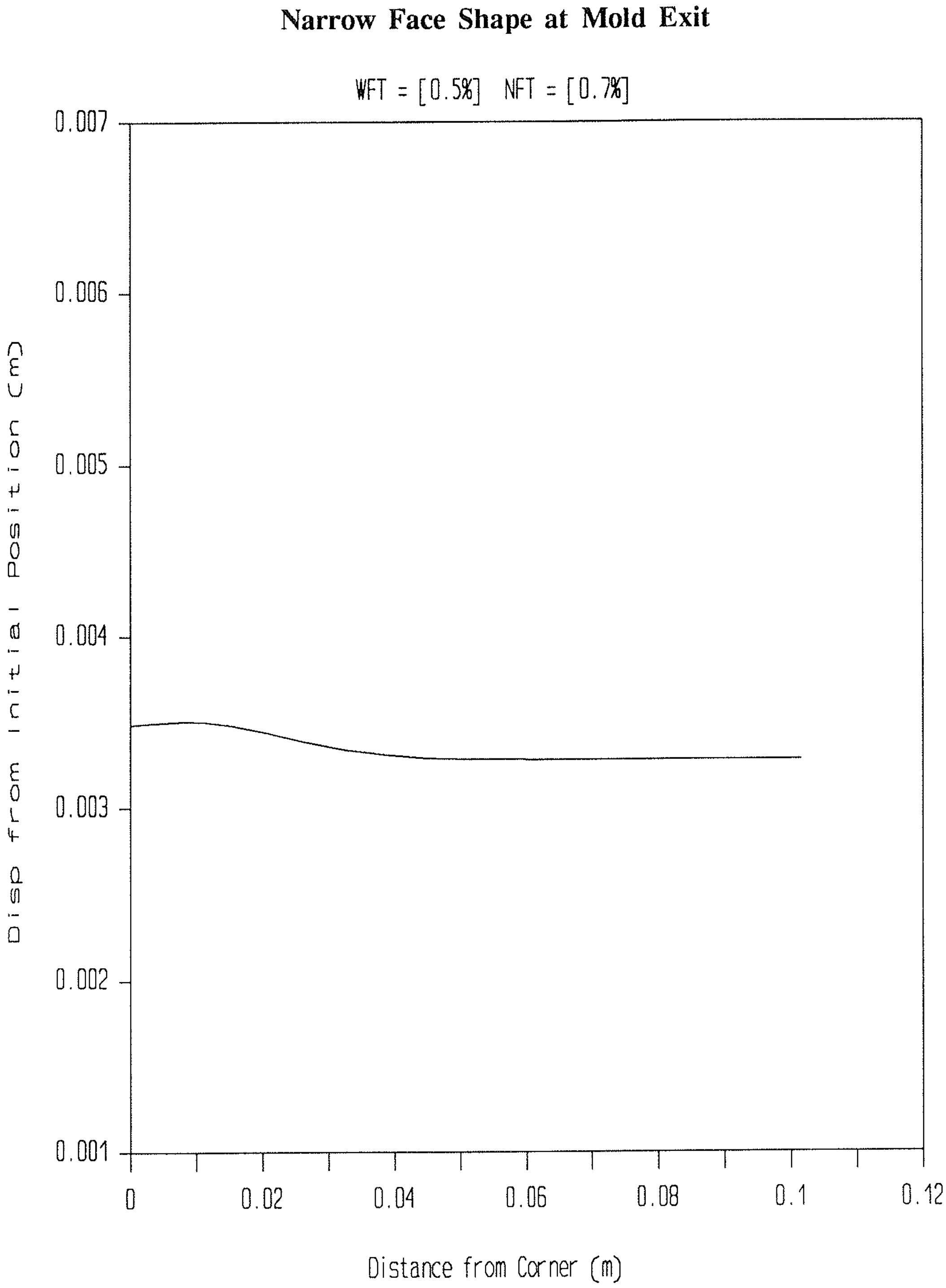


Figure 32 A cross-section through the forming shell at mold exit shows the shape of the narrow face.

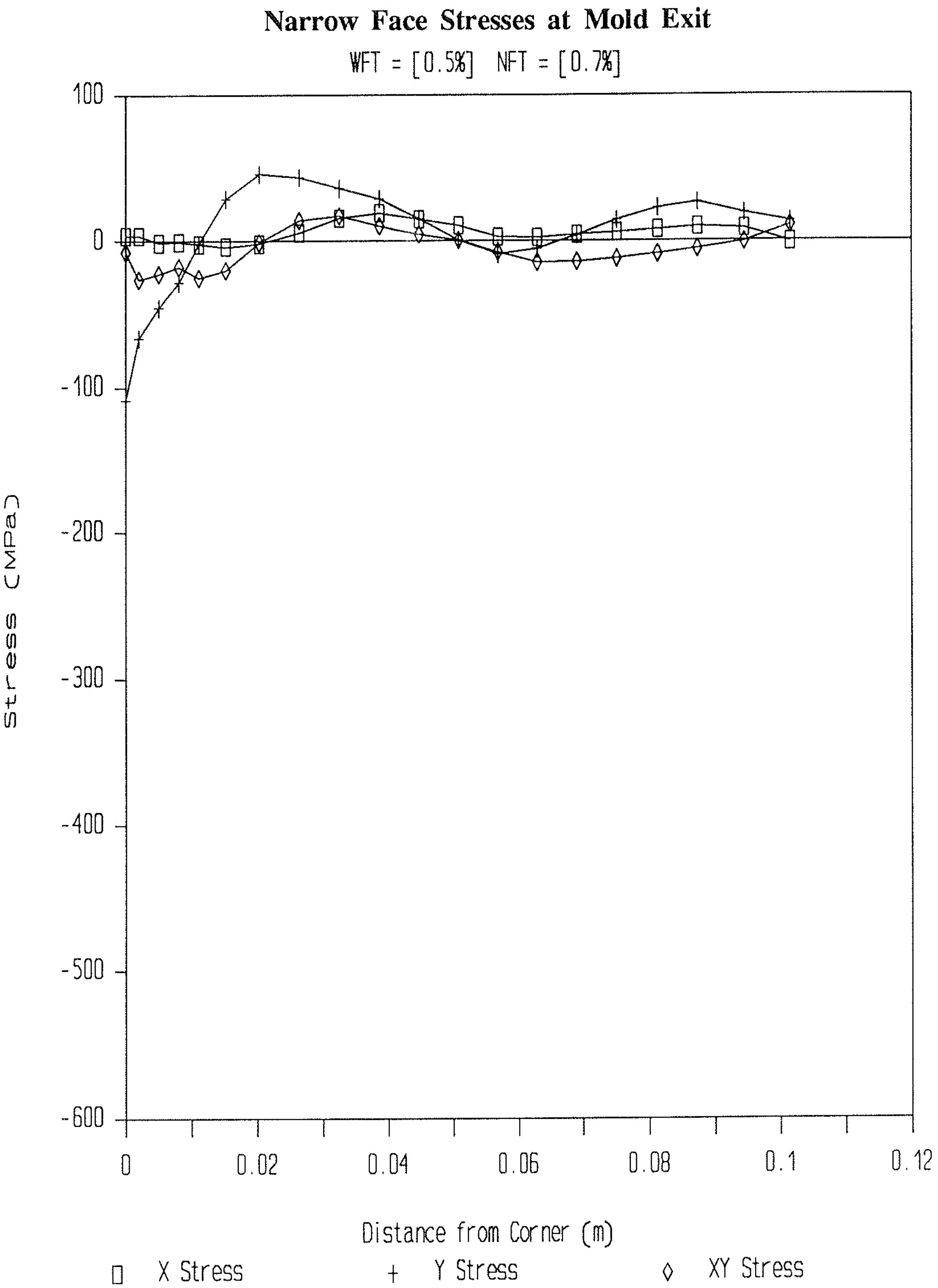


Figure 33 The three components of stress are shown for the external surface of the narrow face at mold exit.

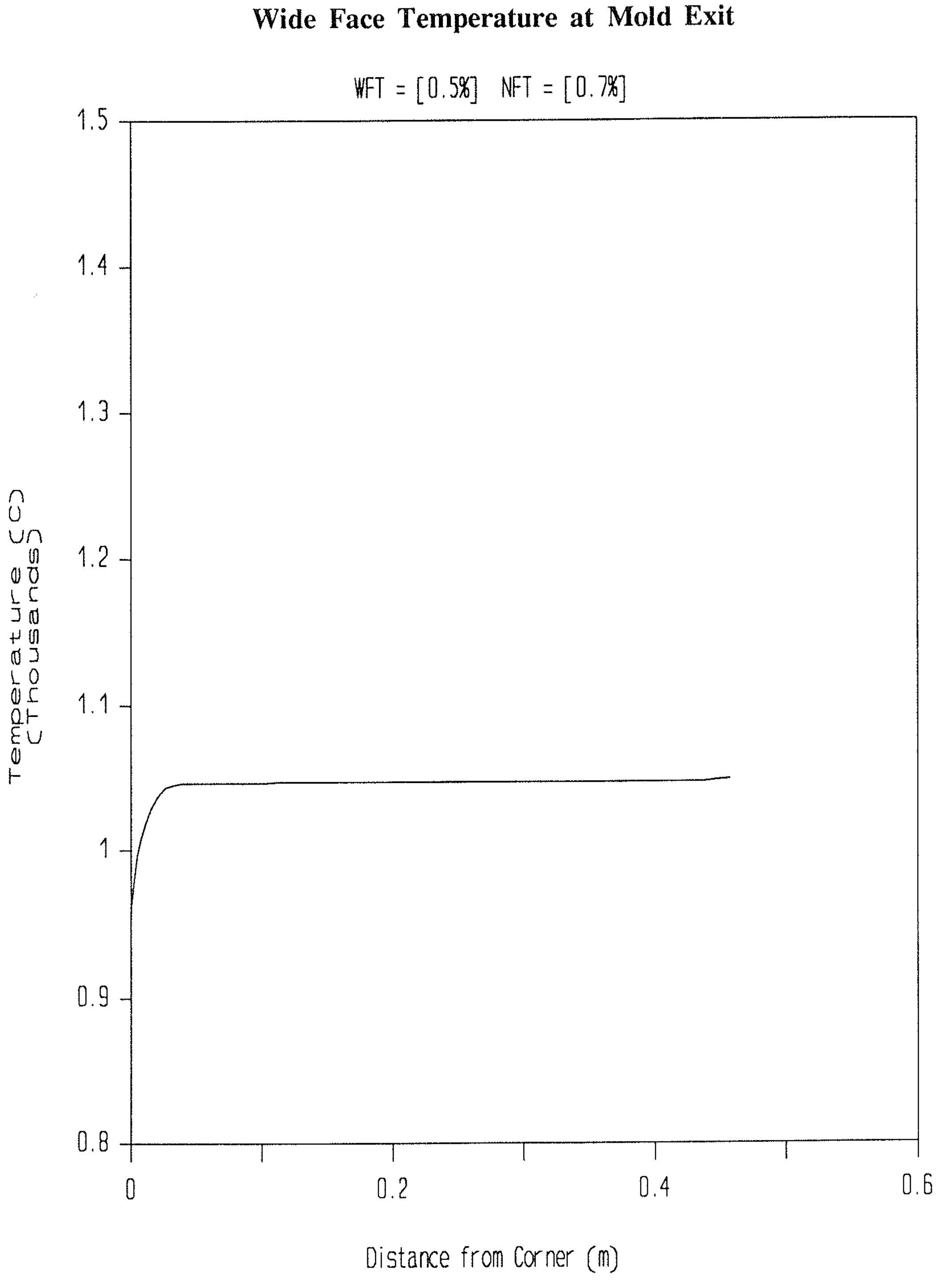


Figure 34 External surface temperatures at mold exit are shown along the wide face.

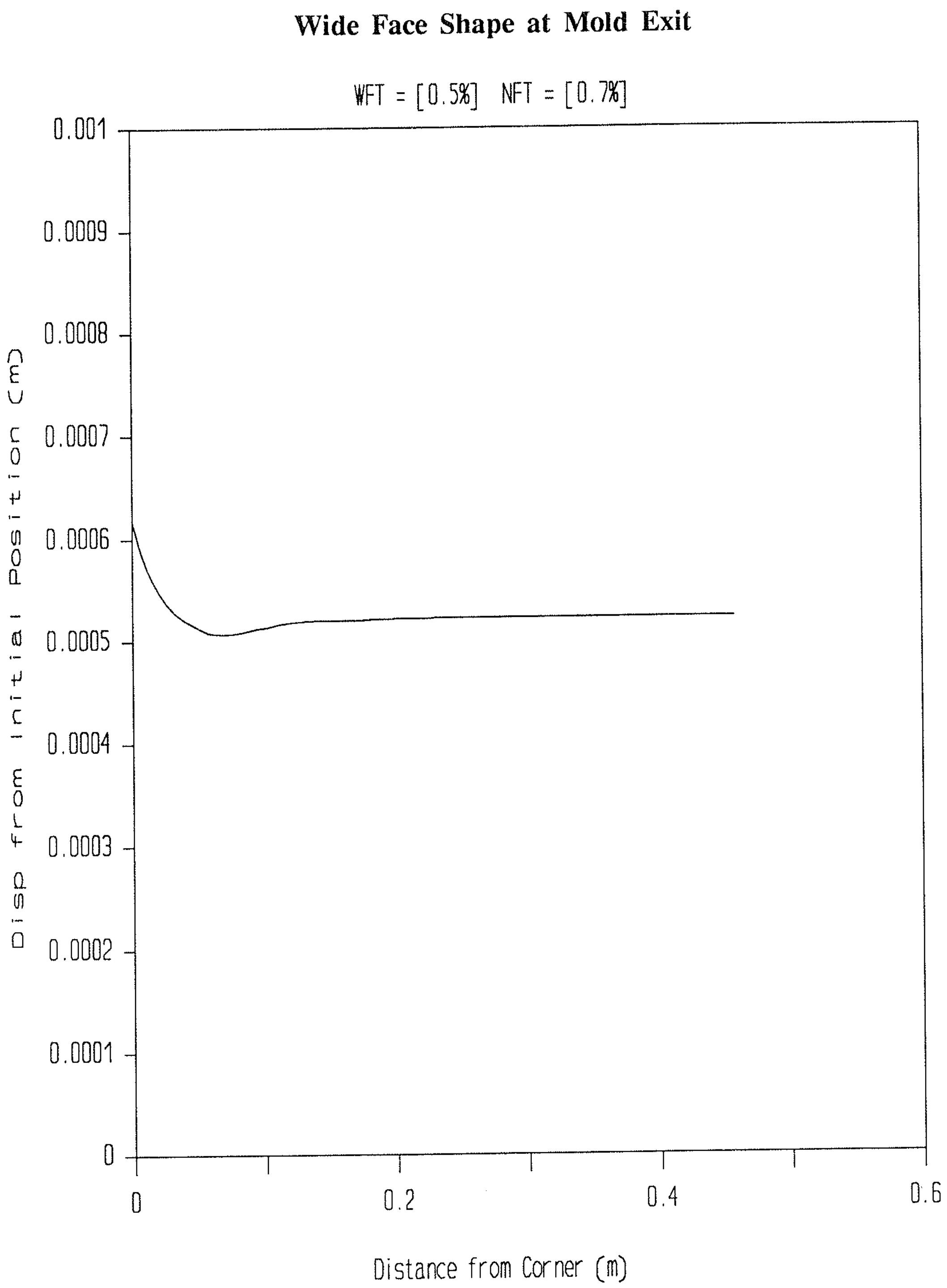


Figure 35 A cross-section through the forming shell at mold exit shows the shape of the wide face.

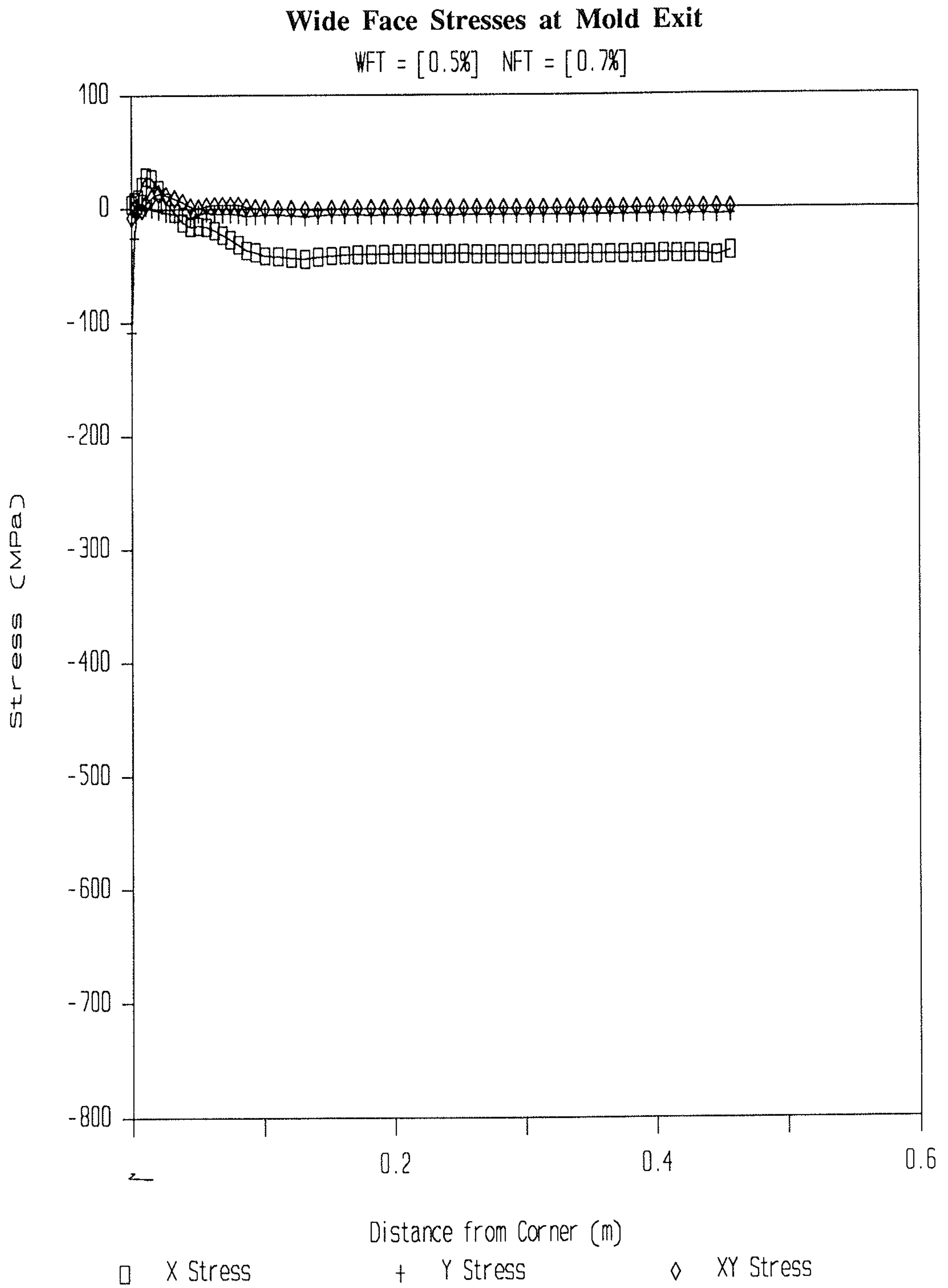


Figure 36 The three components of stress are shown for the external surface of the wide face at mold exit.

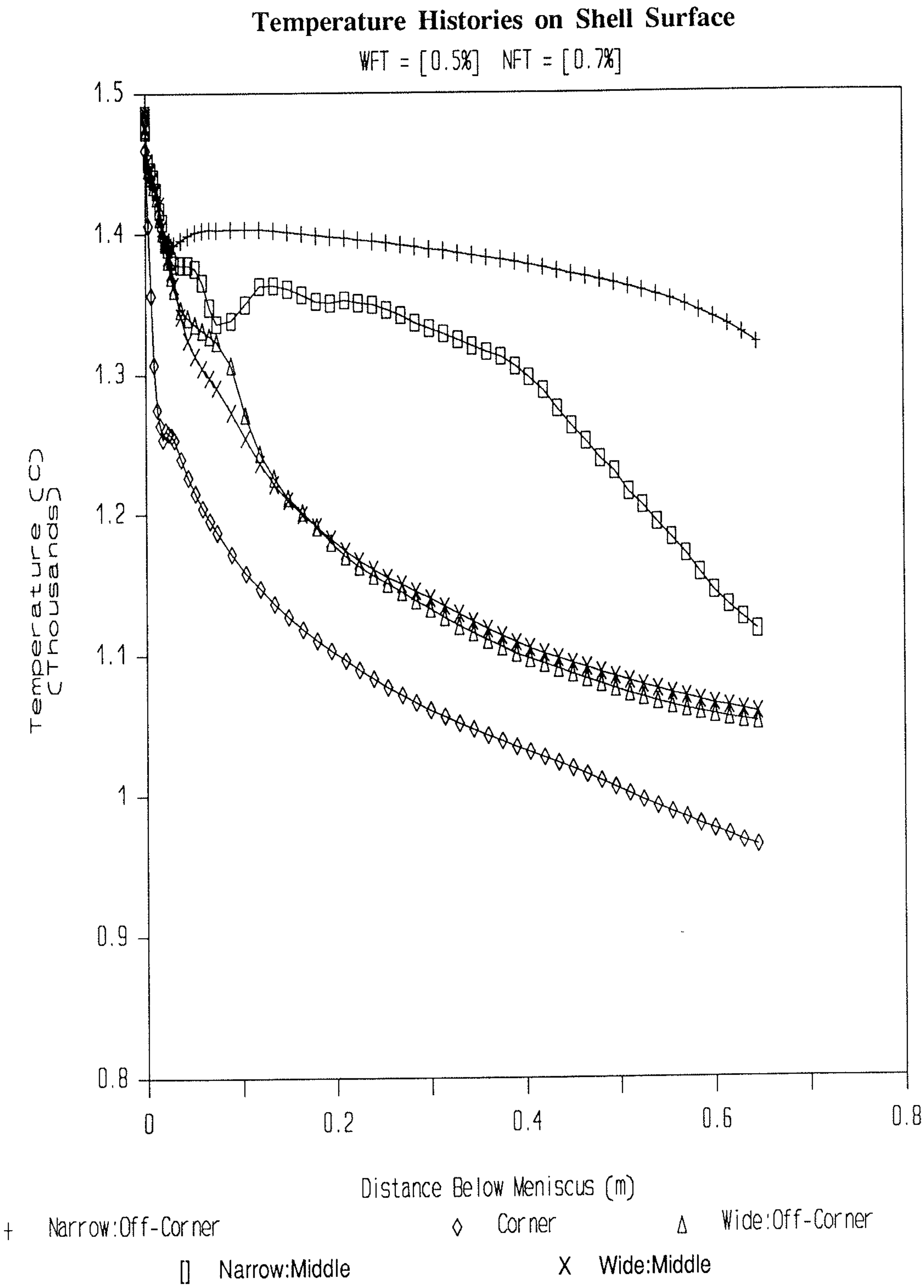


Figure 37 The temperature histories of some significant locations on the surface of the shell are shown over the mold solidification portion of the continuous casting process.

APPENDIX 3

LARGE NARROW FACE TAPER

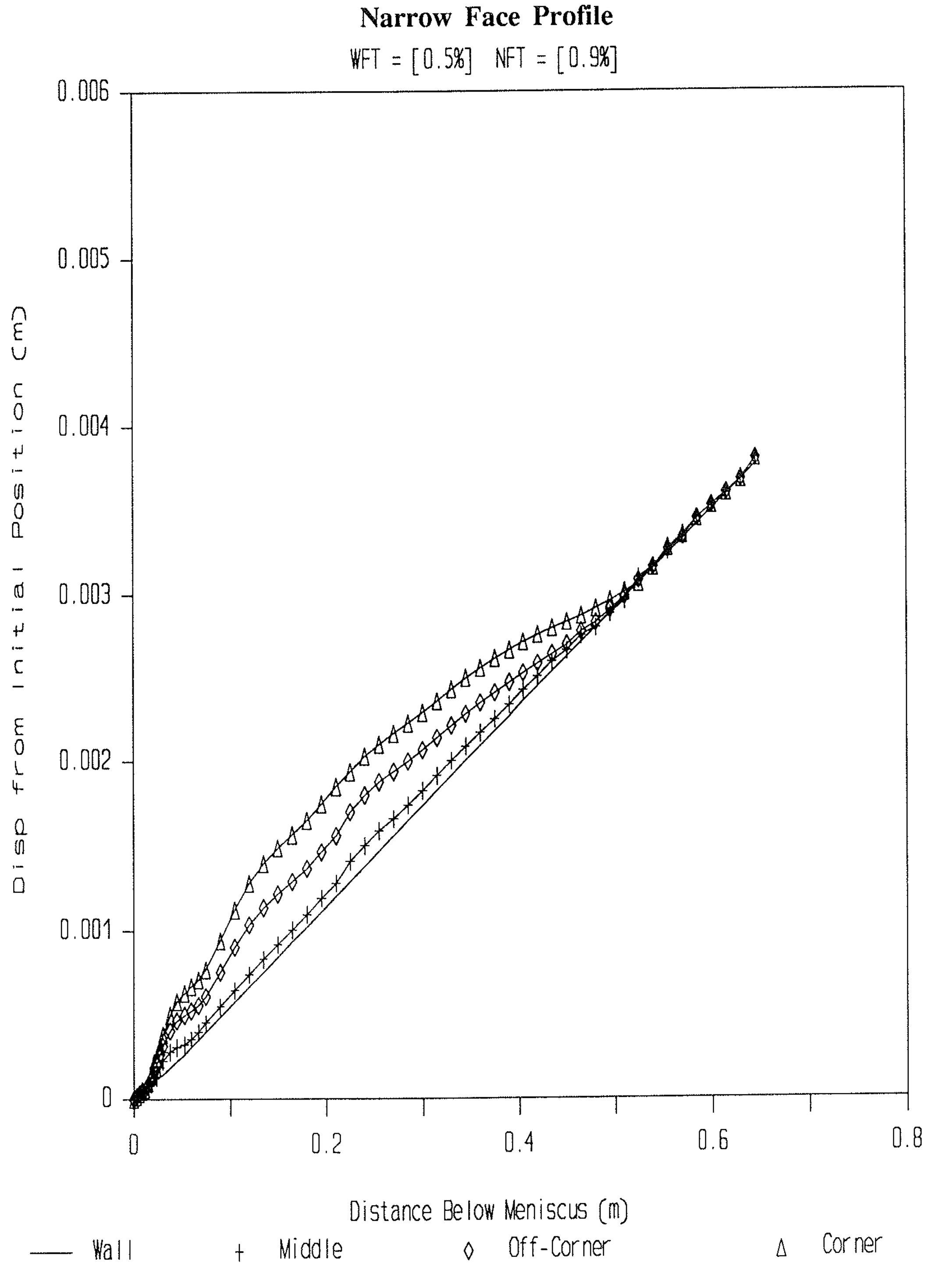


Figure 38 The shrinkage of the narrow face is displayed in relation to the mold wall position.

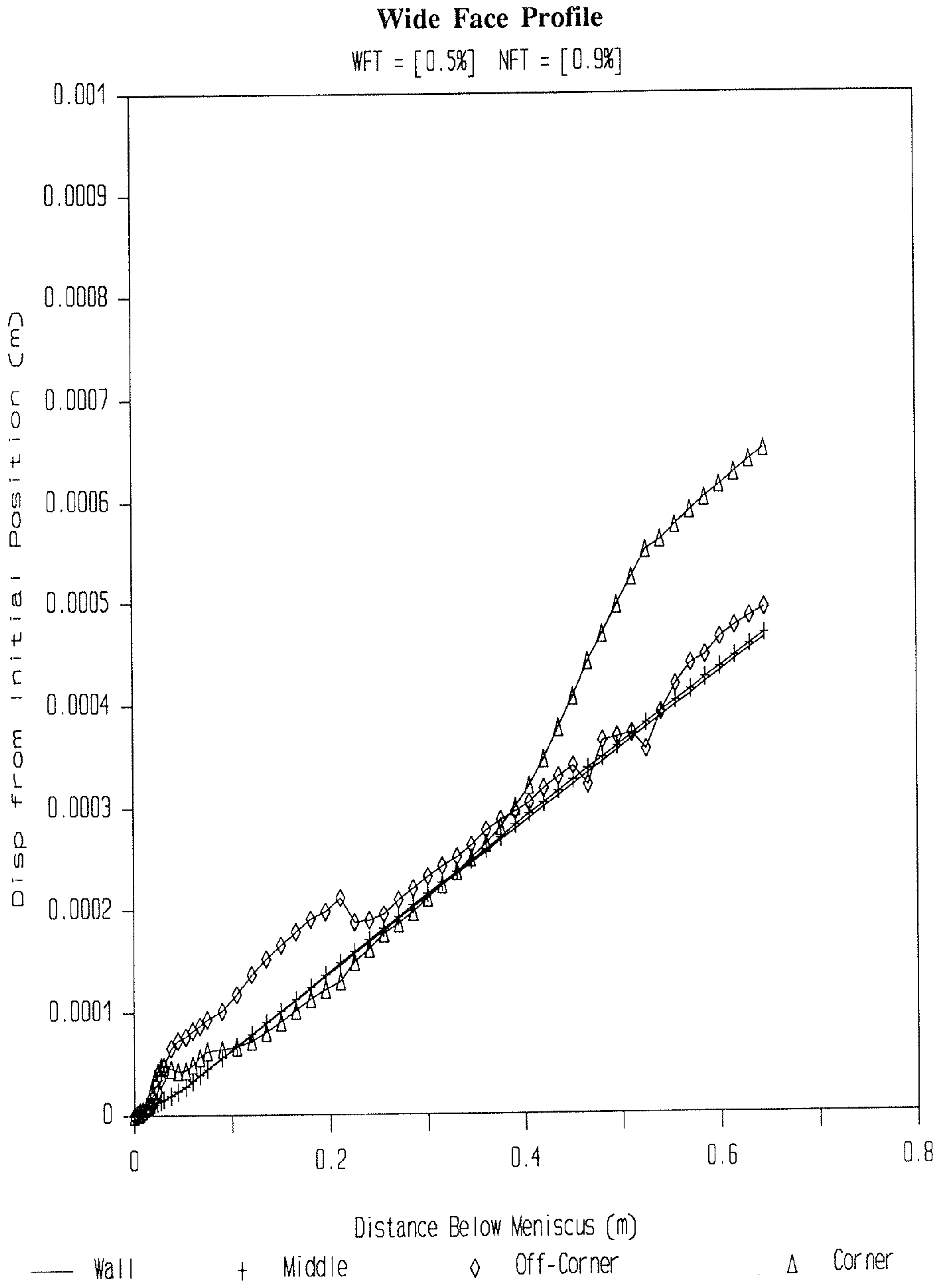


Figure 39 The shrinkage of the wide face is displayed in relation to the mold wall position.

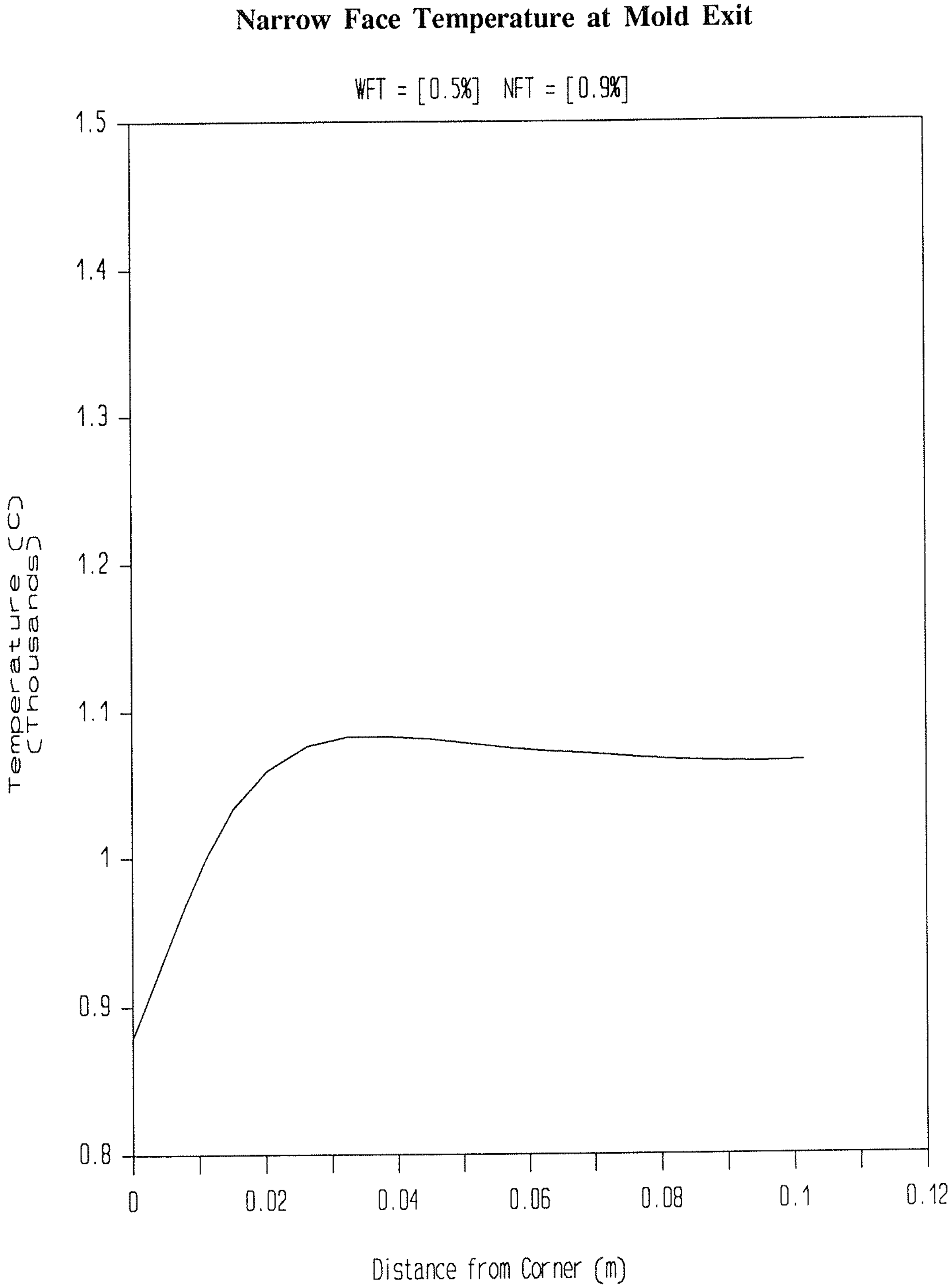


Figure 40 External surface temperatures at mold exit are shown along the narrow face.

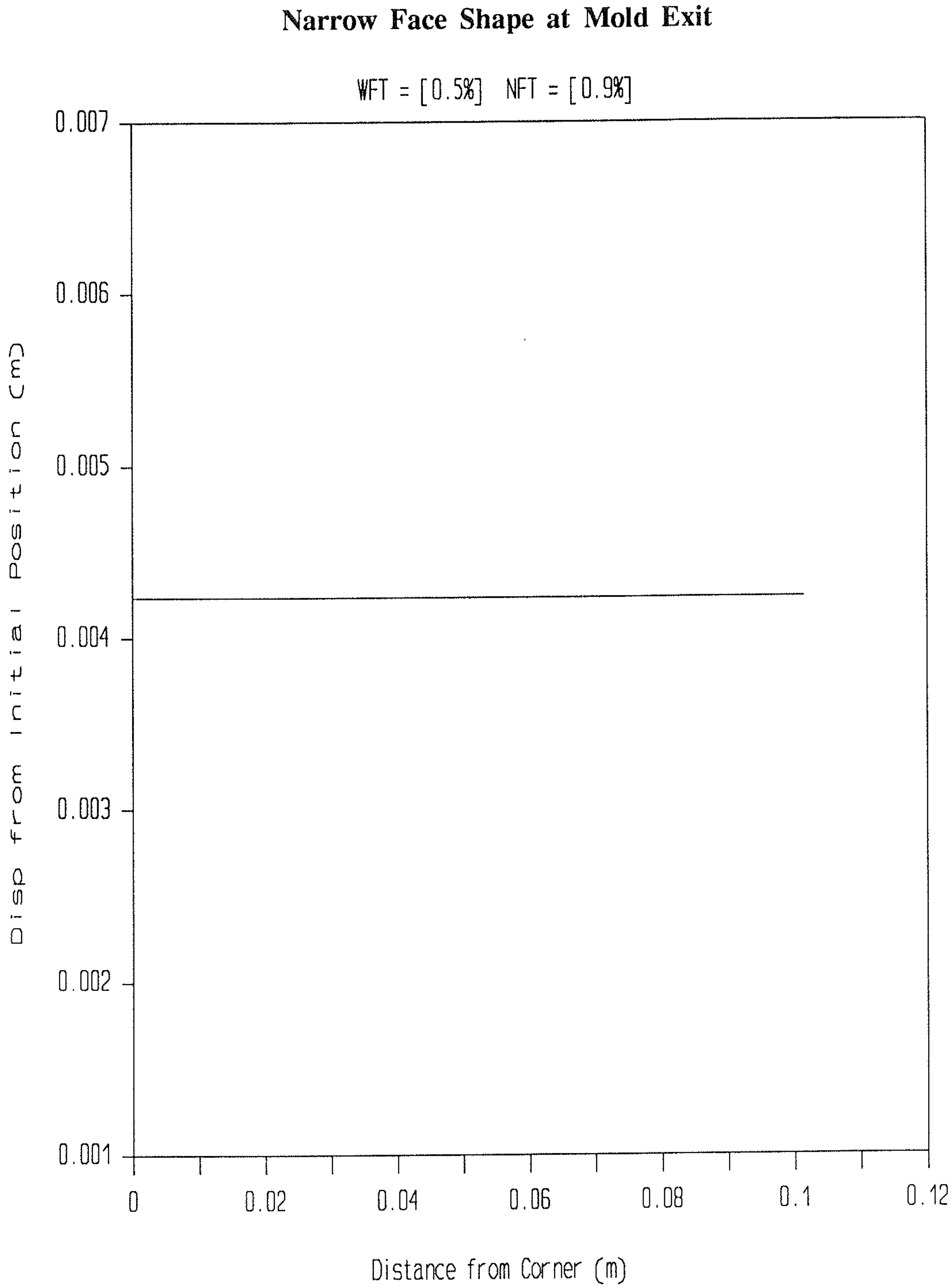


Figure 41 A cross-section through the forming shell at mold exit shows the shape of the narrow face.

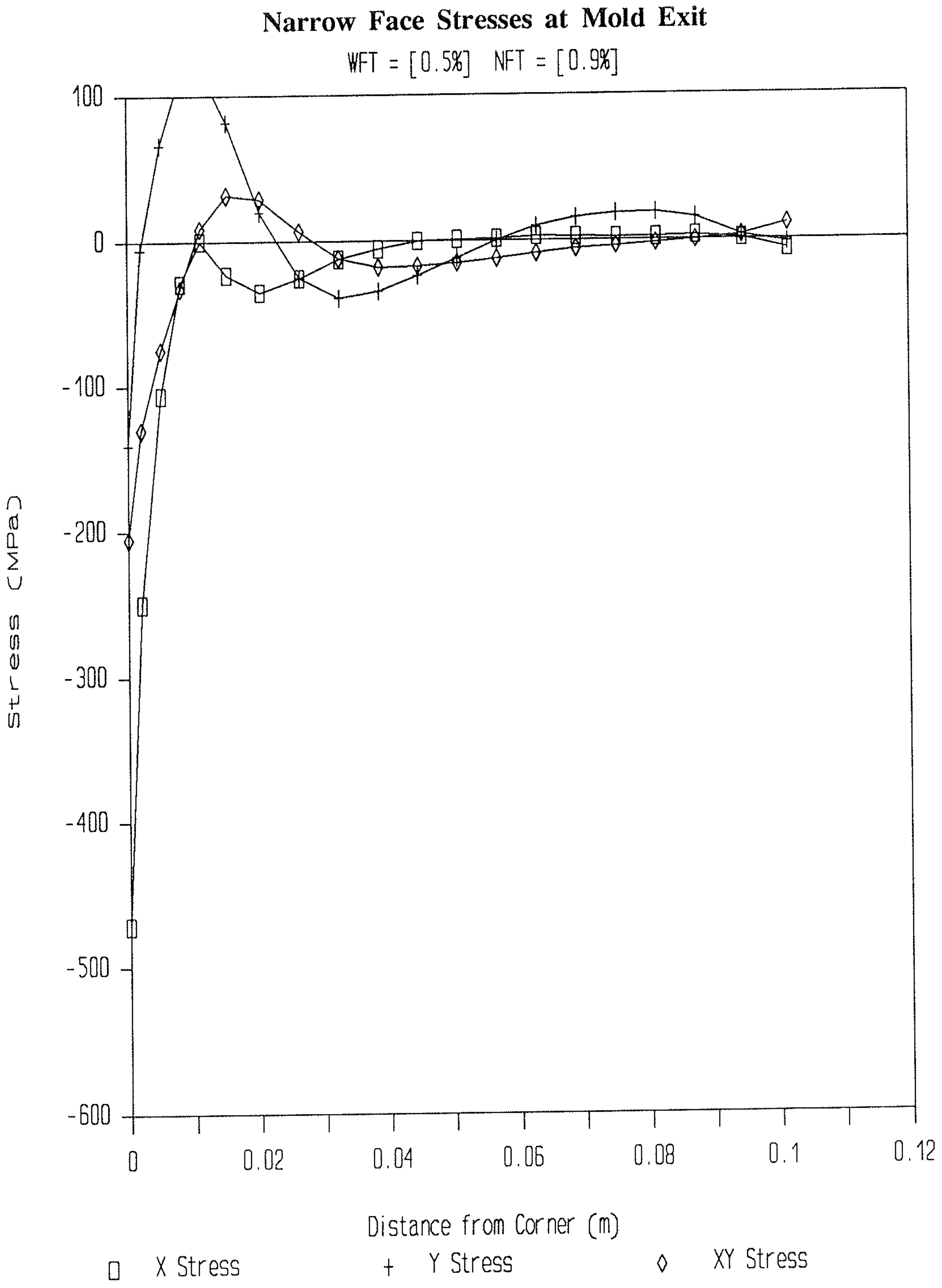


Figure 42 The three components of stress are shown for the external surface of the narrow face at mold exit.

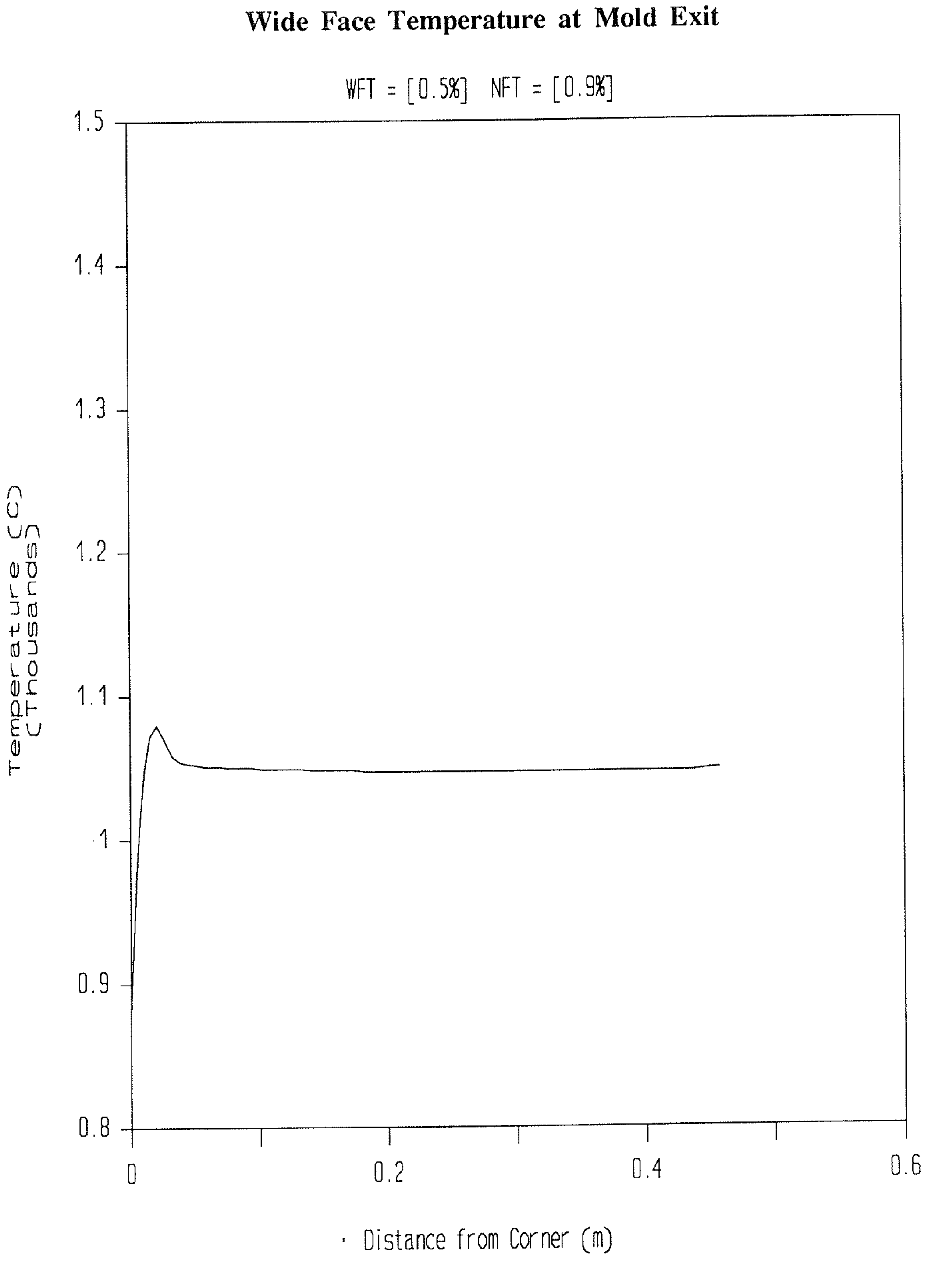


Figure 43 External surface temperatures at mold exit are shown along the wide face.

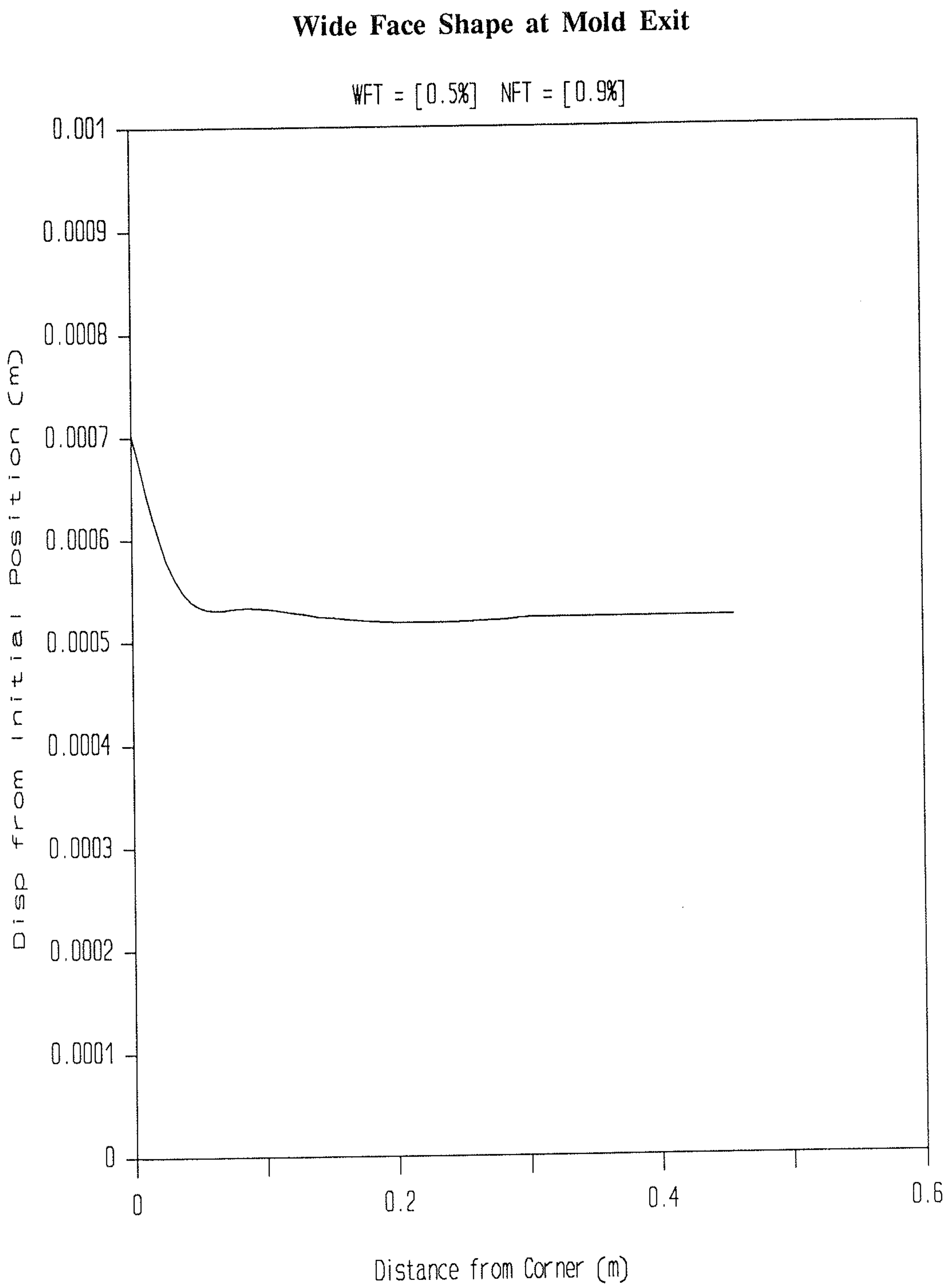


Figure 44 A cross-section through the forming shell at mold exit shows the shape of the wide face.

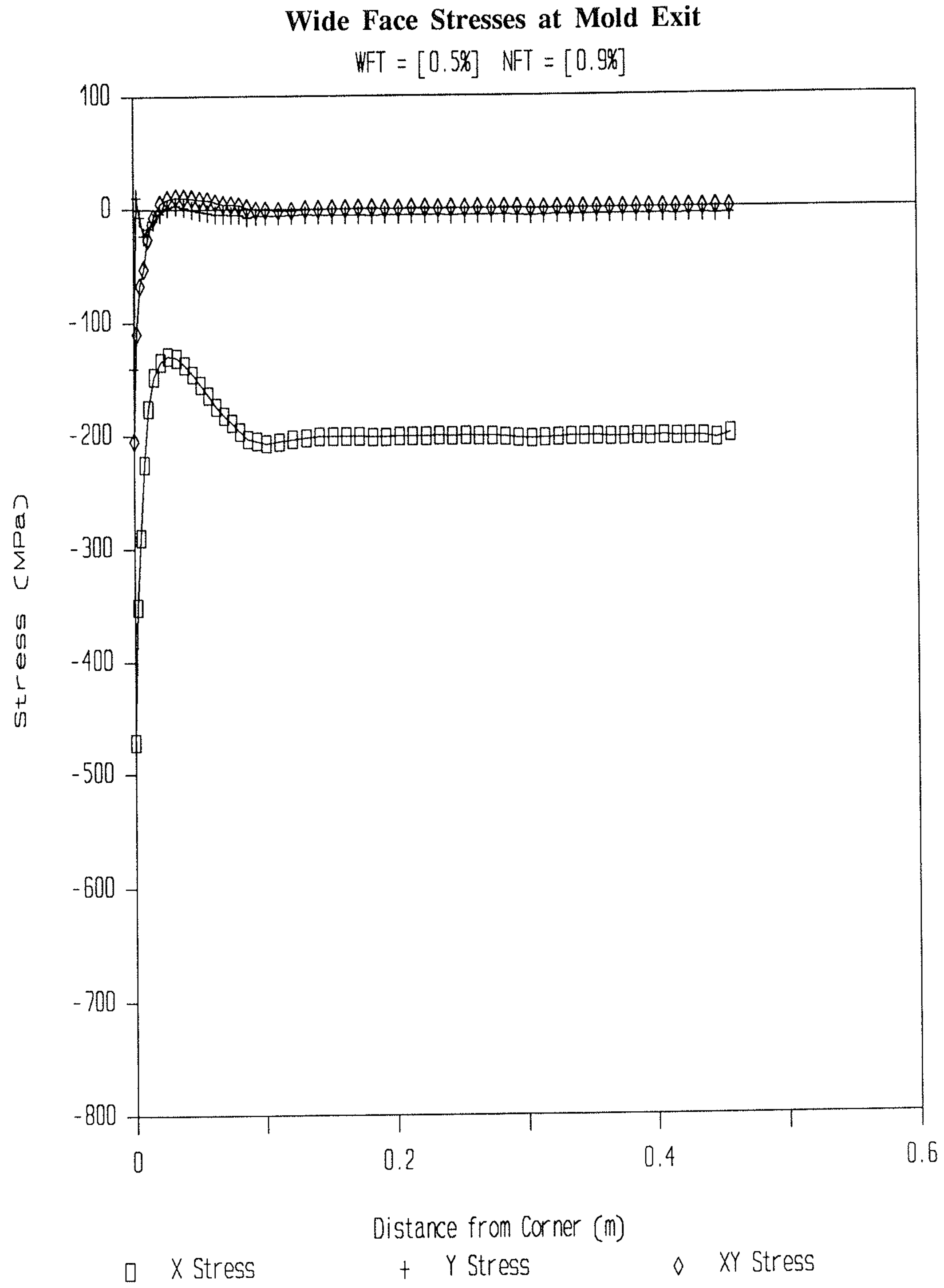


Figure 45 The three components of stress are shown for the external surface of the wide face at mold exit.

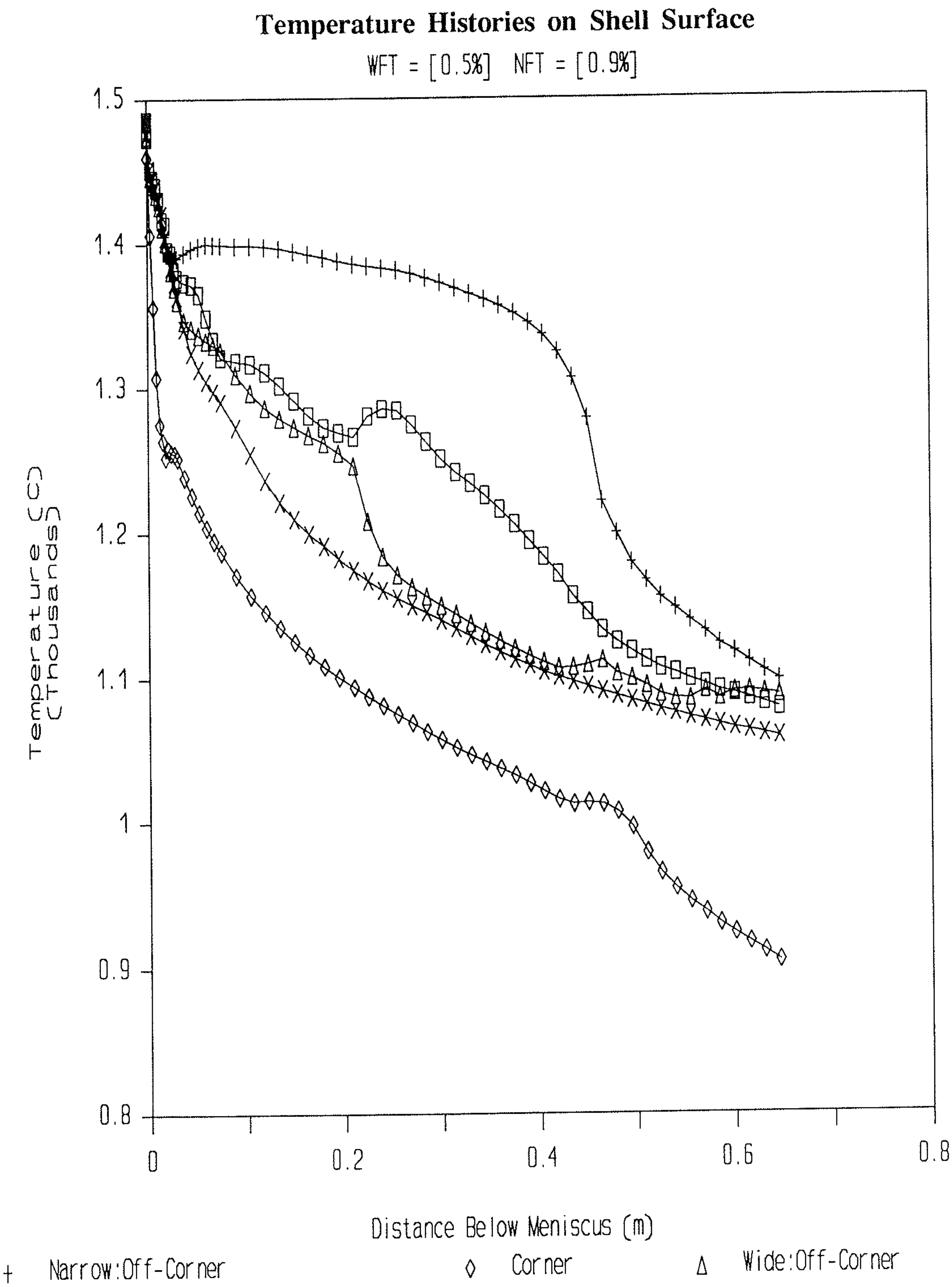


Figure 46 The temperature histories of some significant locations on the surface of the shell are shown over the mold solidification portion of the continuous casting process.

APPENDIX 4

EXTREMELY LARGE TAPERS

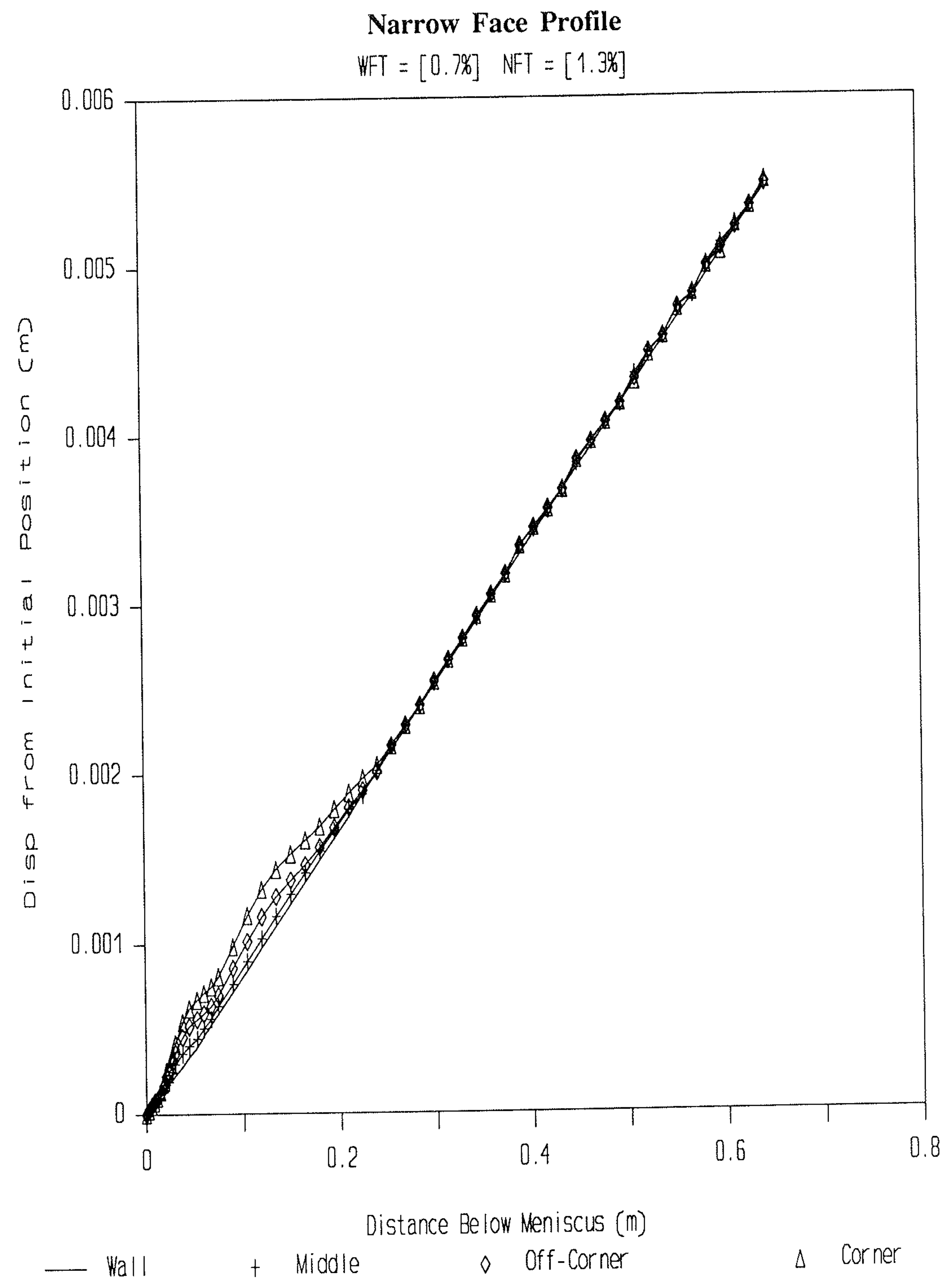


Figure 47 The shrinkage of the narrow face is displayed in relation to the mold wall position.

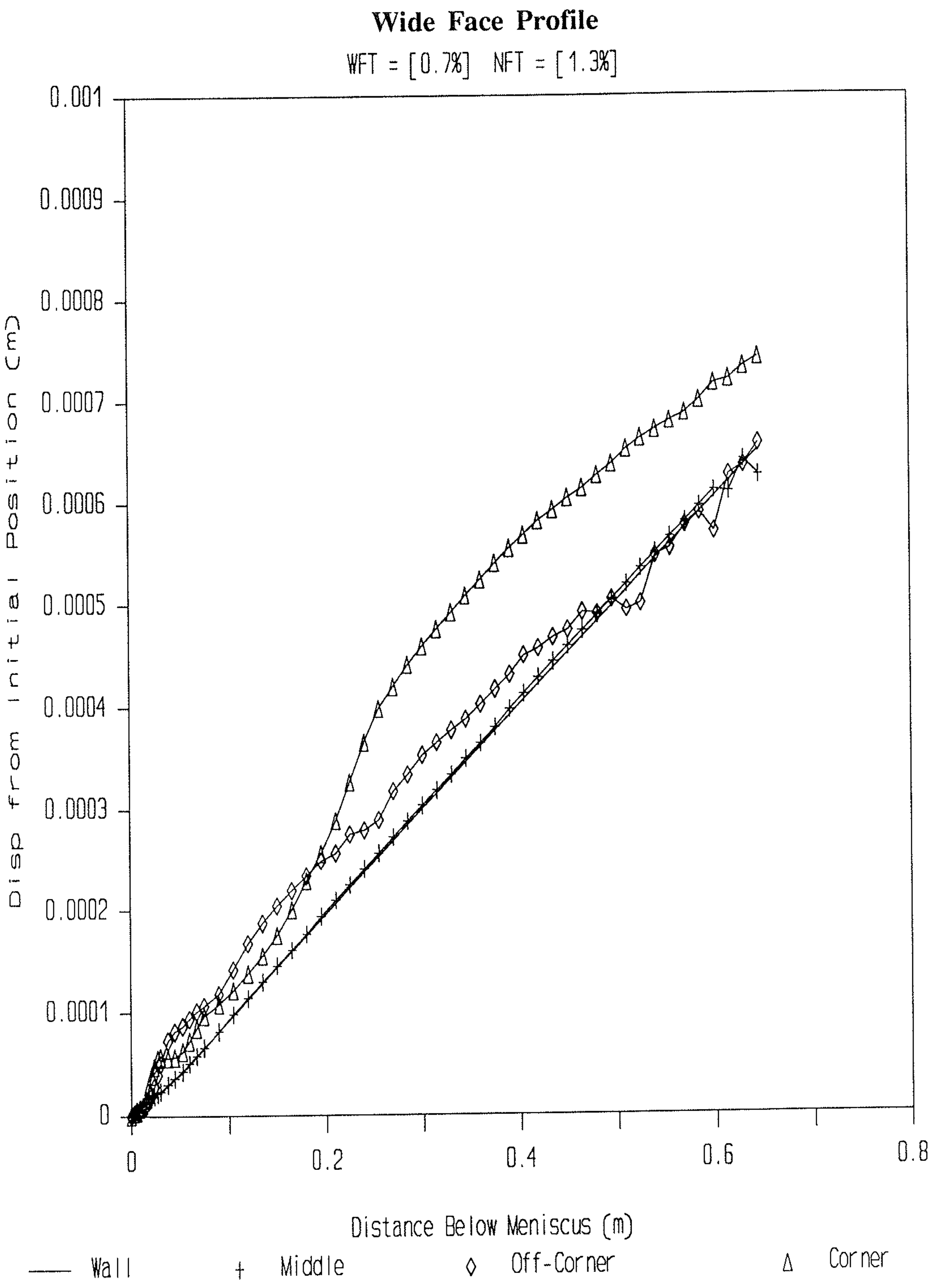


Figure 48 The shrinkage of the wide face is displayed in relation to the mold wall position.

Narrow Face Temperature at Mold Exit

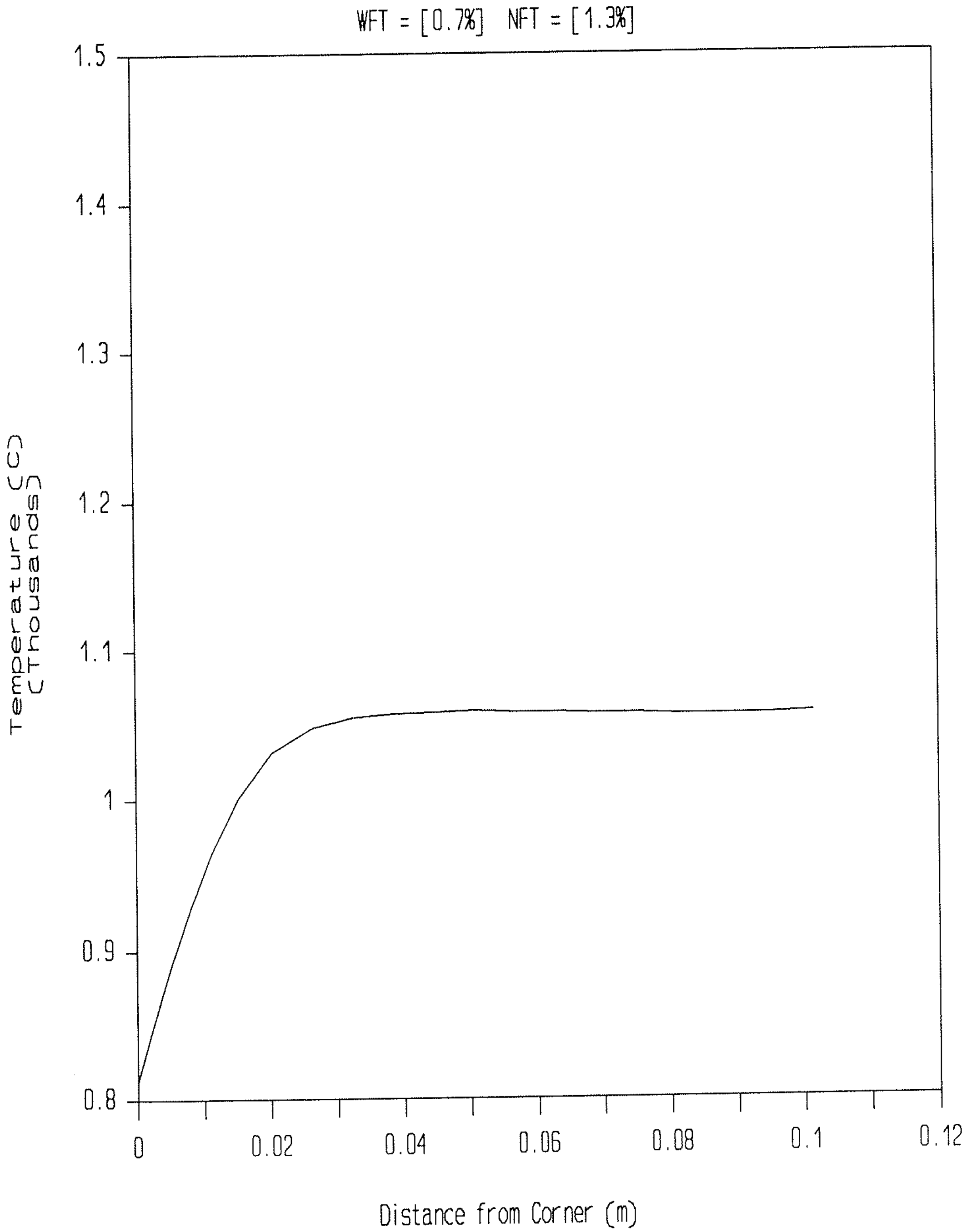


Figure 49 External surface temperatures at mold exit are shown along the narrow face.

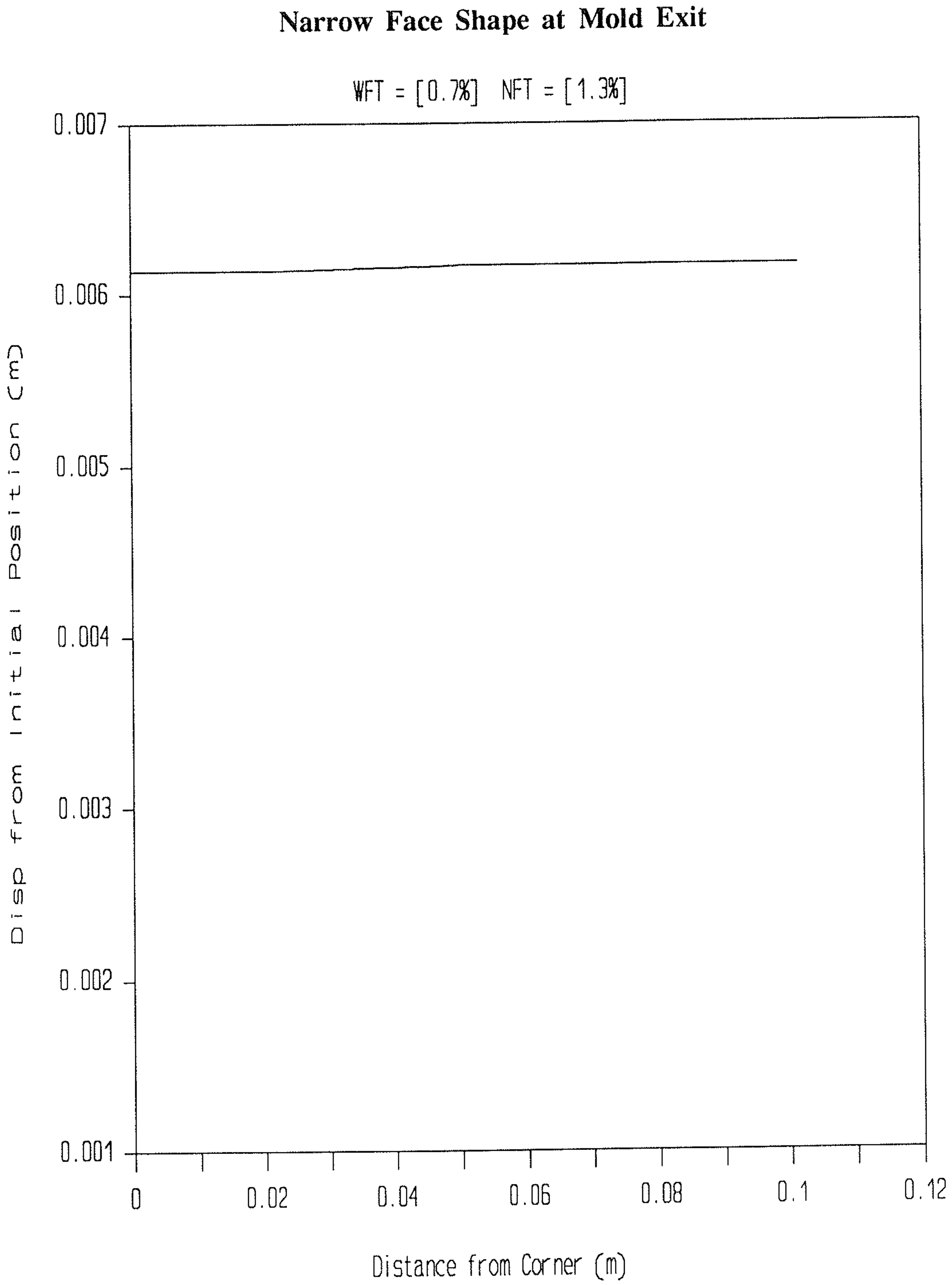


Figure 50 A cross-section through the forming shell at mold exit shows the shape of the narrow face.

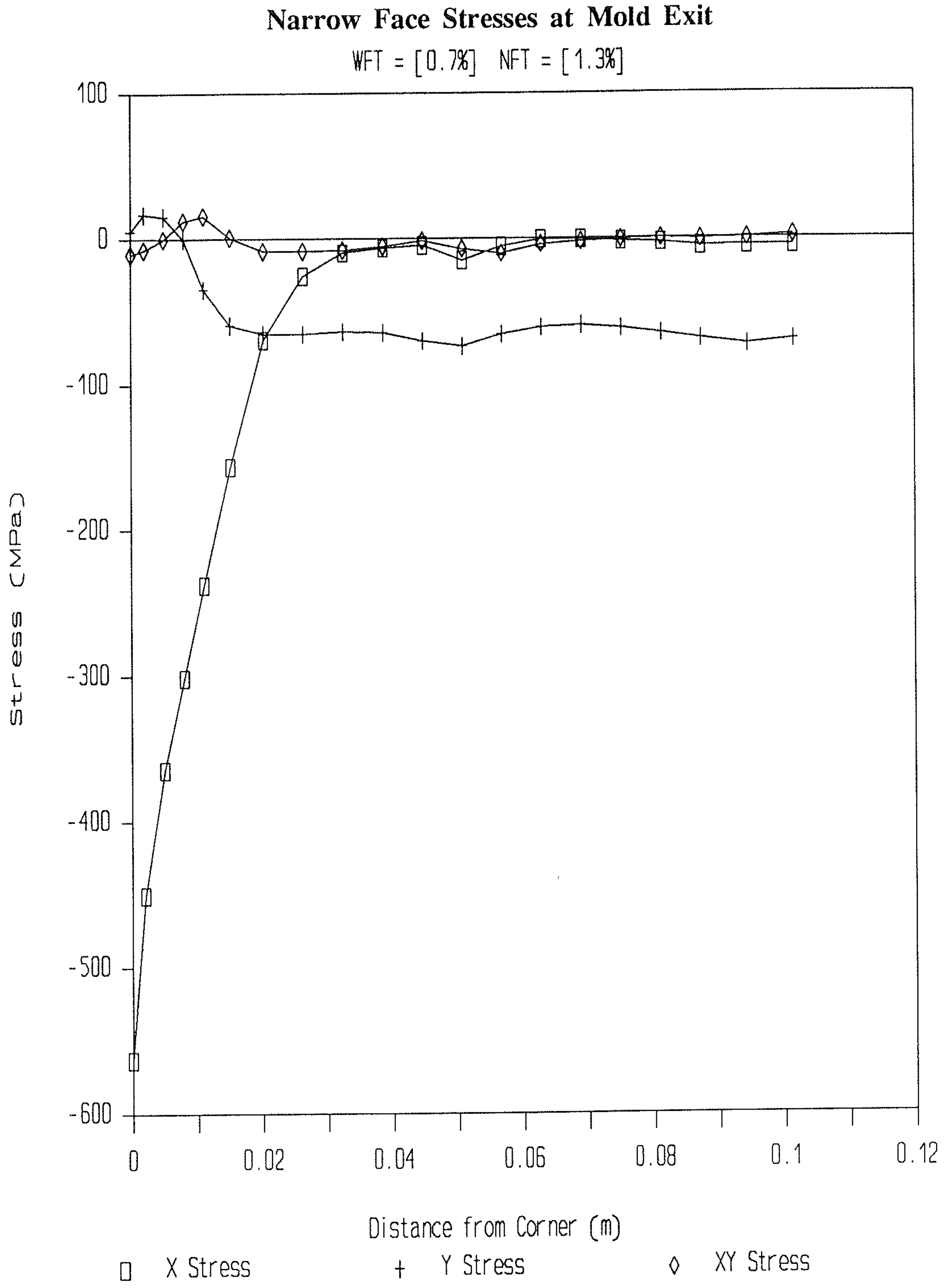


Figure 51 The three components of stress are shown for the external surface of the narrow face at mold exit.

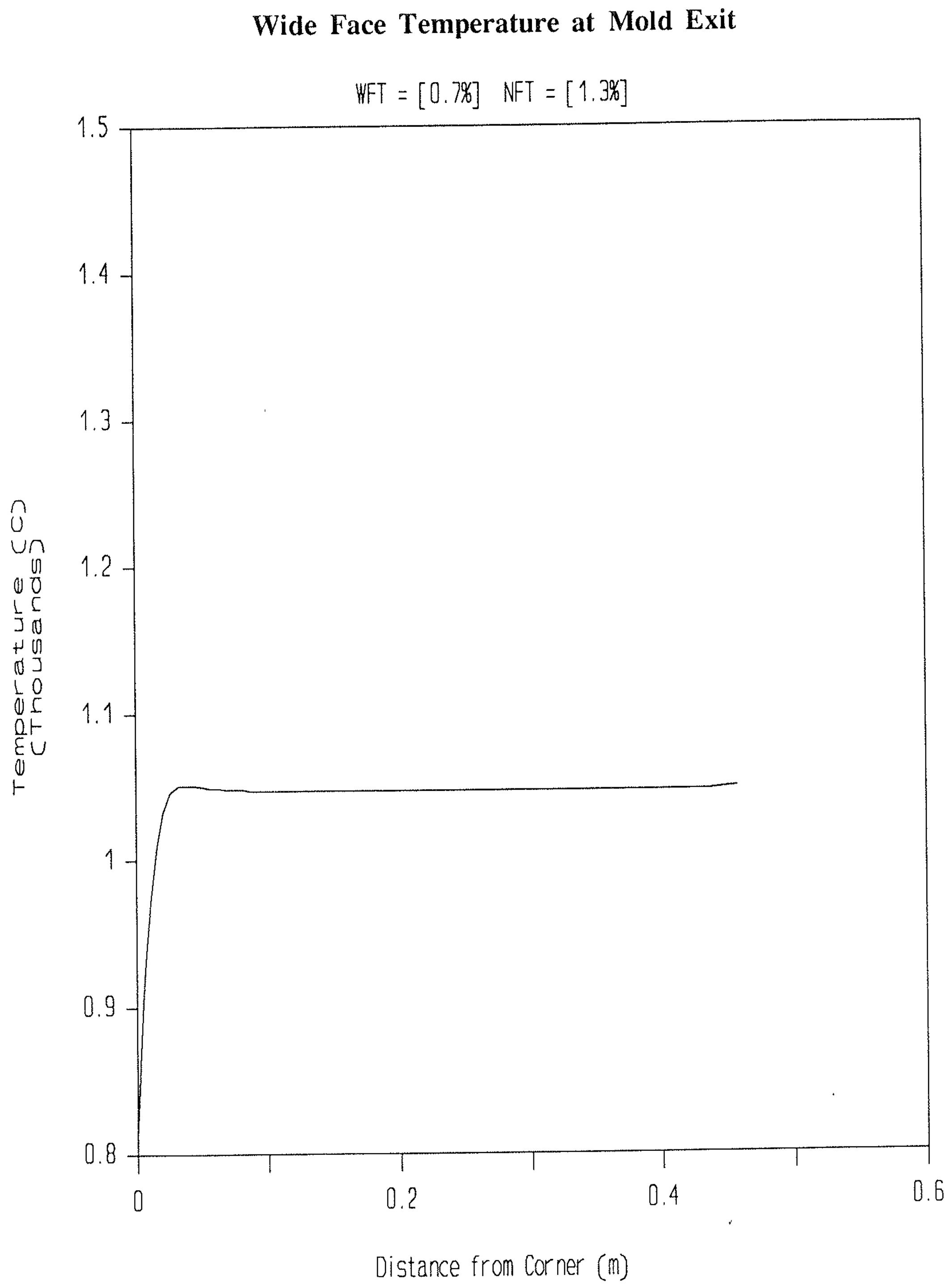


Figure 52 External surface temperatures at mold exit are shown along the wide face.

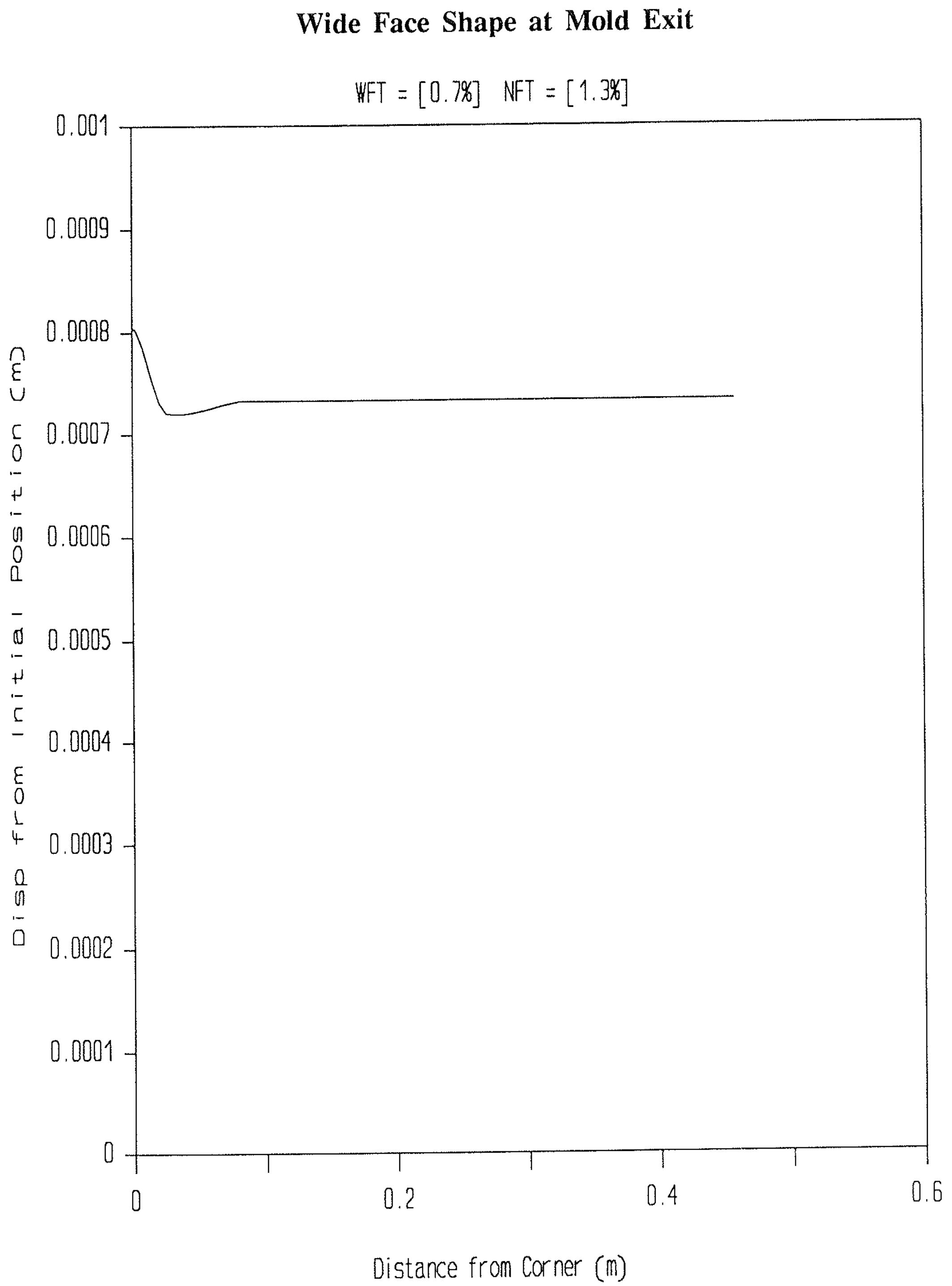


Figure 53 A cross-section through the forming shell at mold exit shows the shape of the wide face.

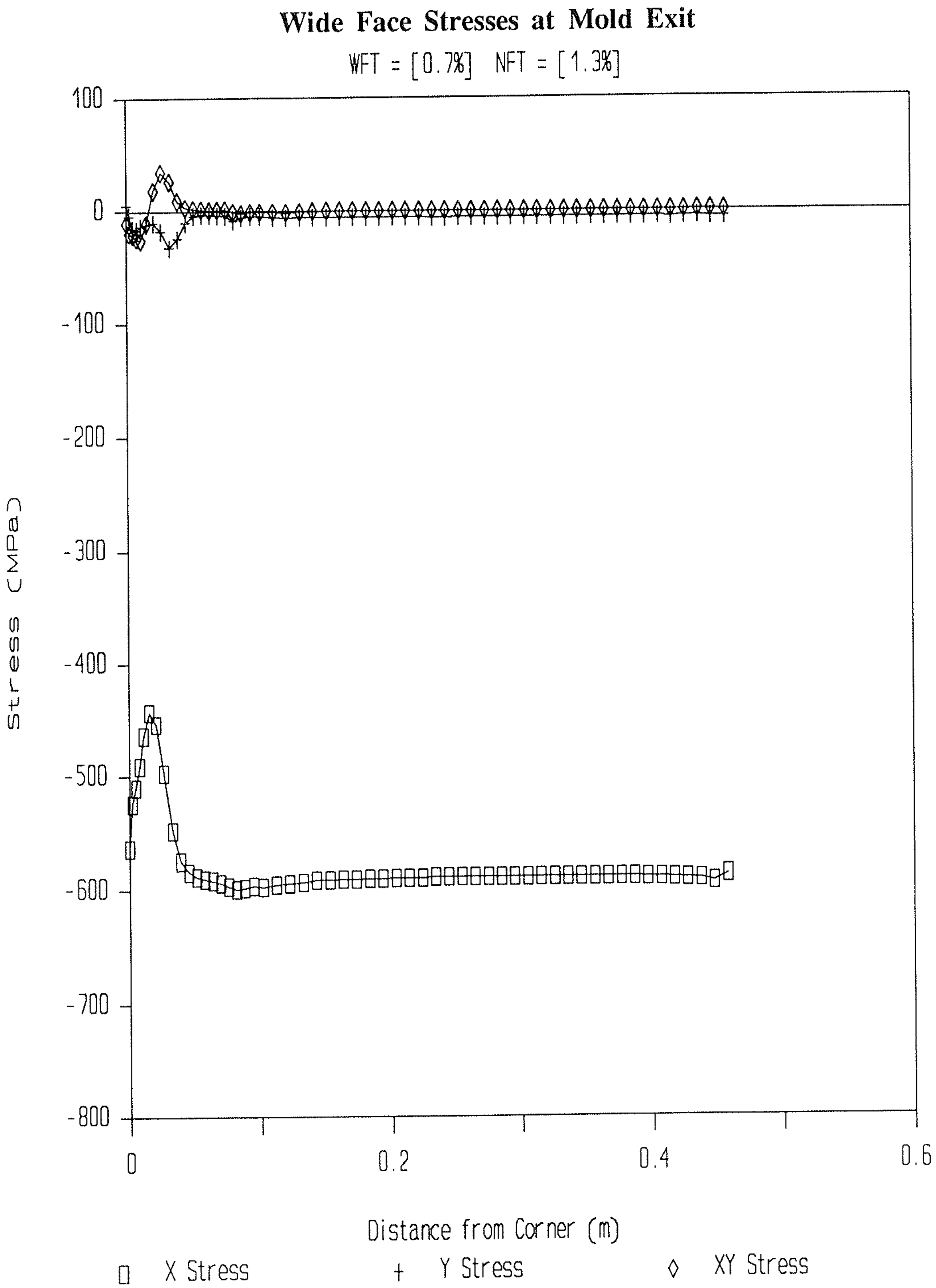


Figure 54 The three components of stress are shown for the external surface of the wide face at mold exit.

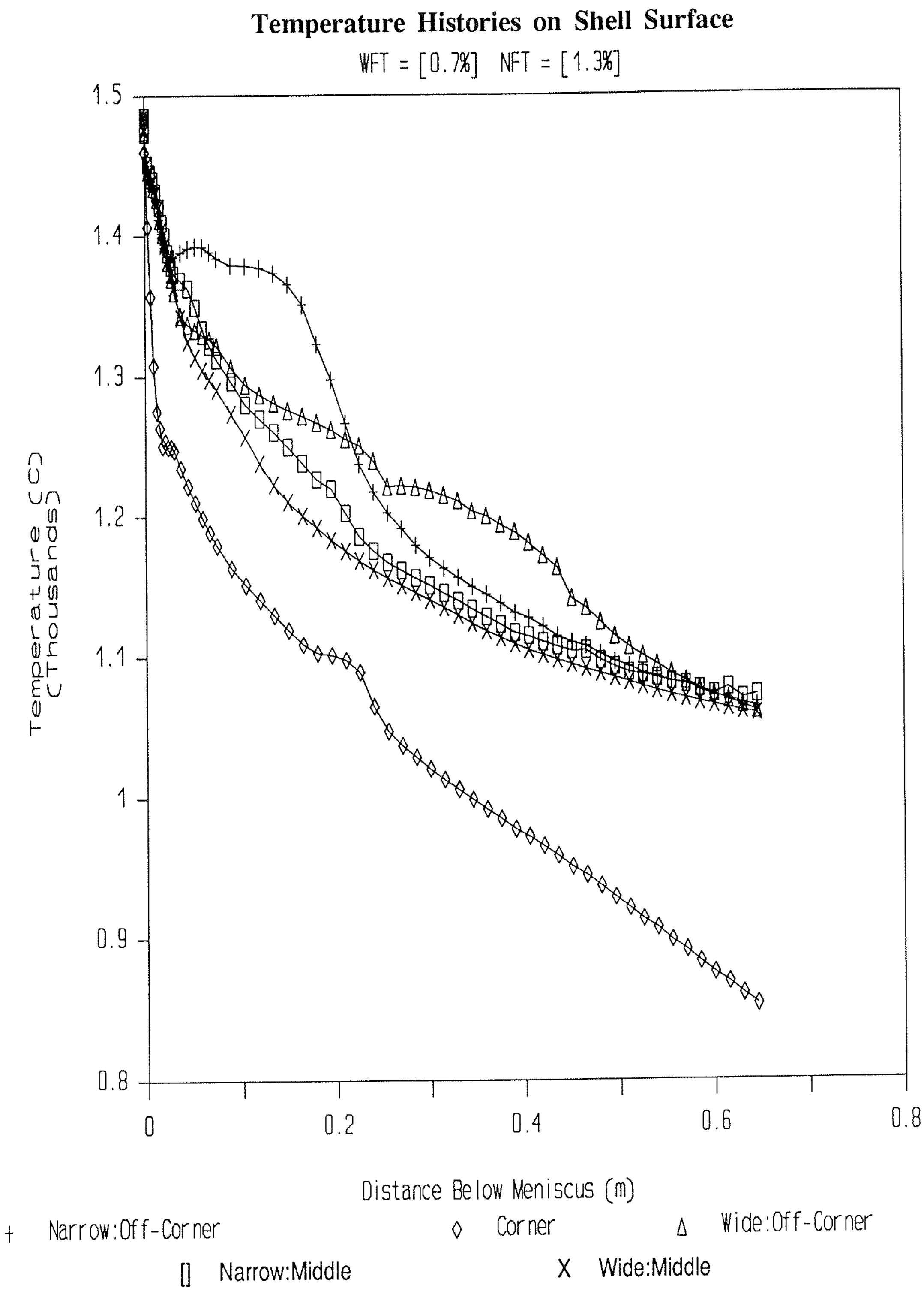


Figure 55 The temperature histories of some significant locations on the surface of the shell are shown over the mold solidification portion of the continuous casting process.

APPENDIX 5

EXTREMELY SMALL TAPERS

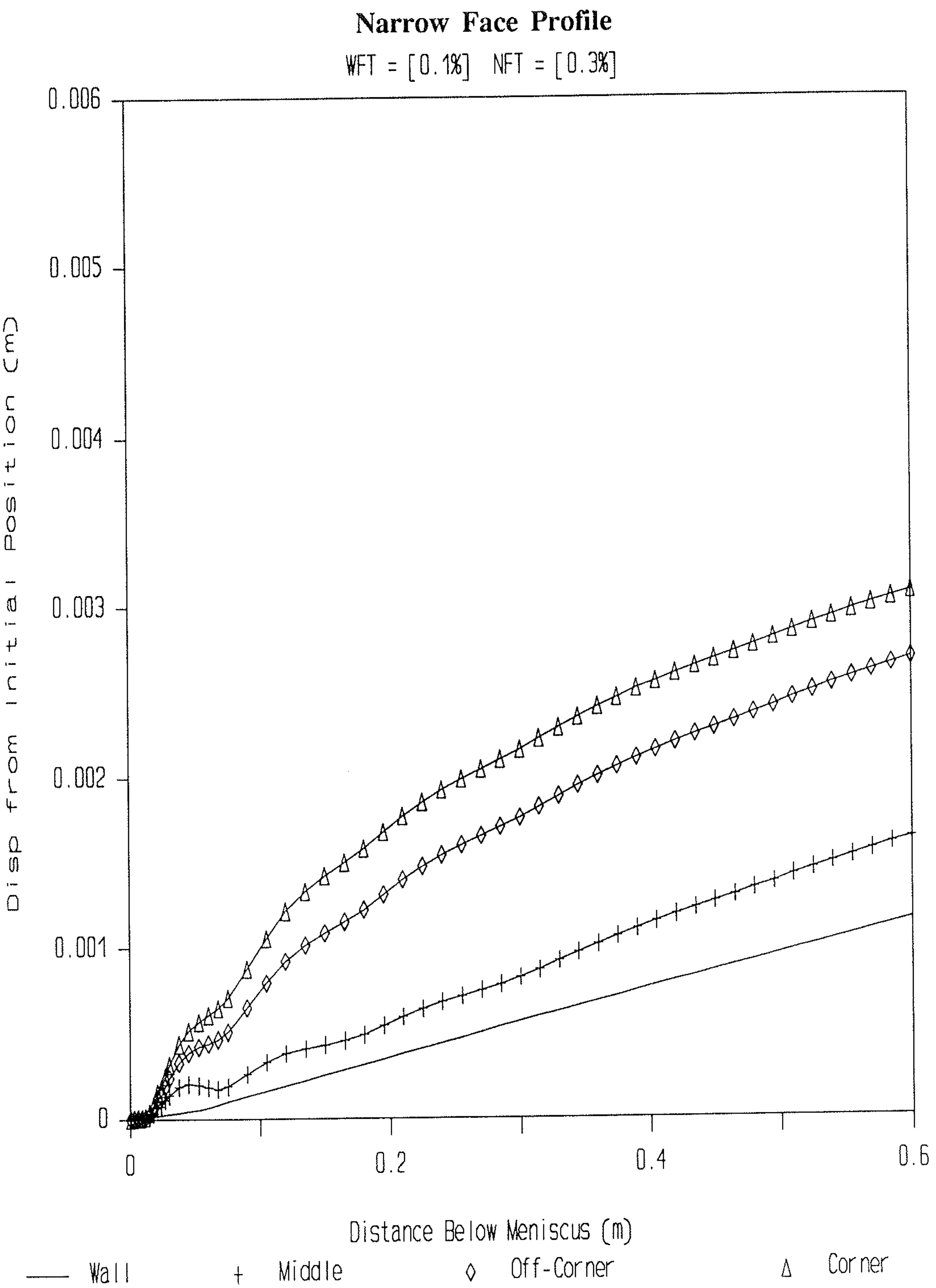


Figure 56 The shrinkage of the narrow face is displayed in relation to the mold wall position.

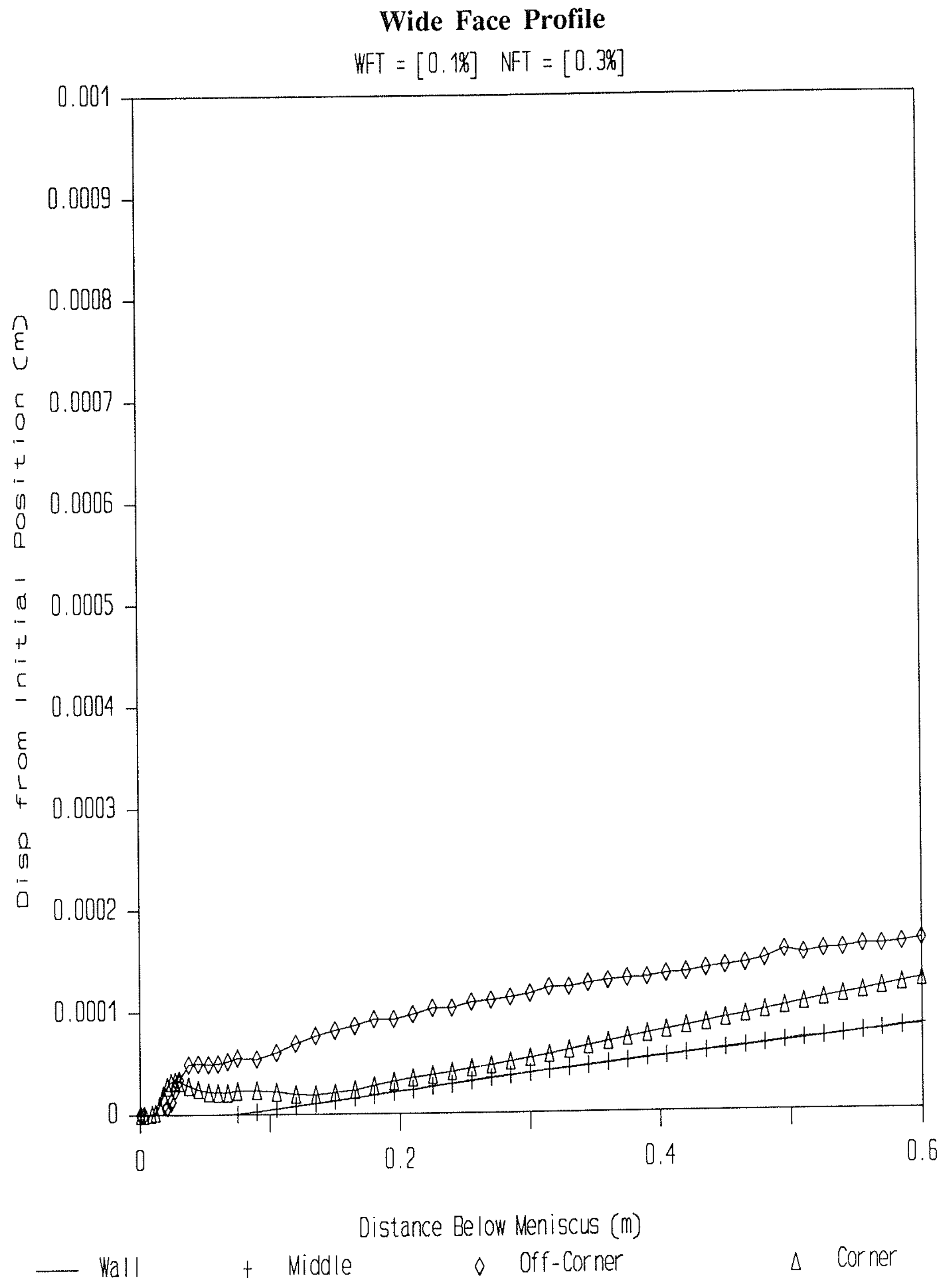


Figure 57 The shrinkage of the wide face is displayed in relation to the mold wall position.

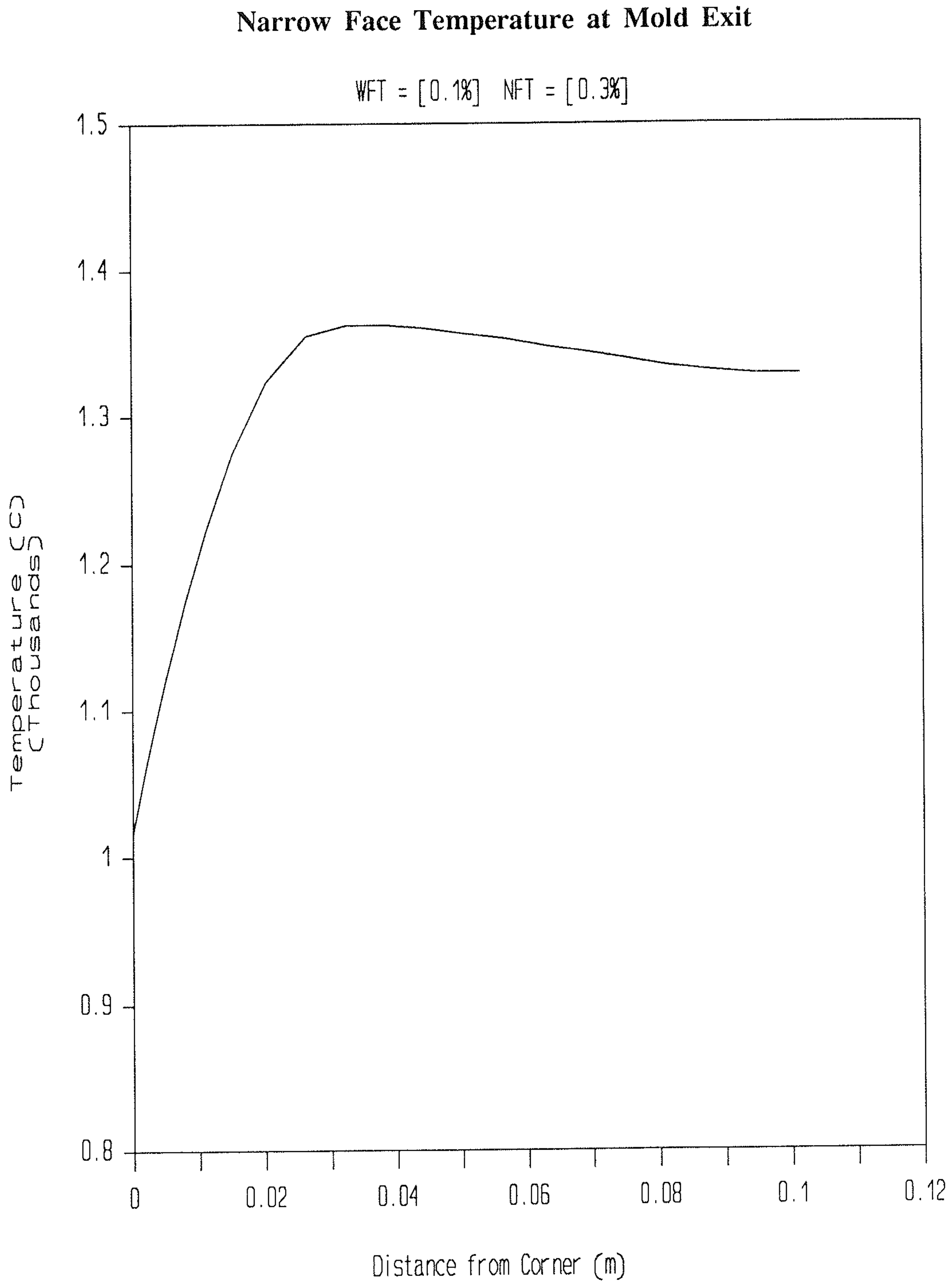


Figure 58 External surface temperatures at mold exit are shown along the narrow face.

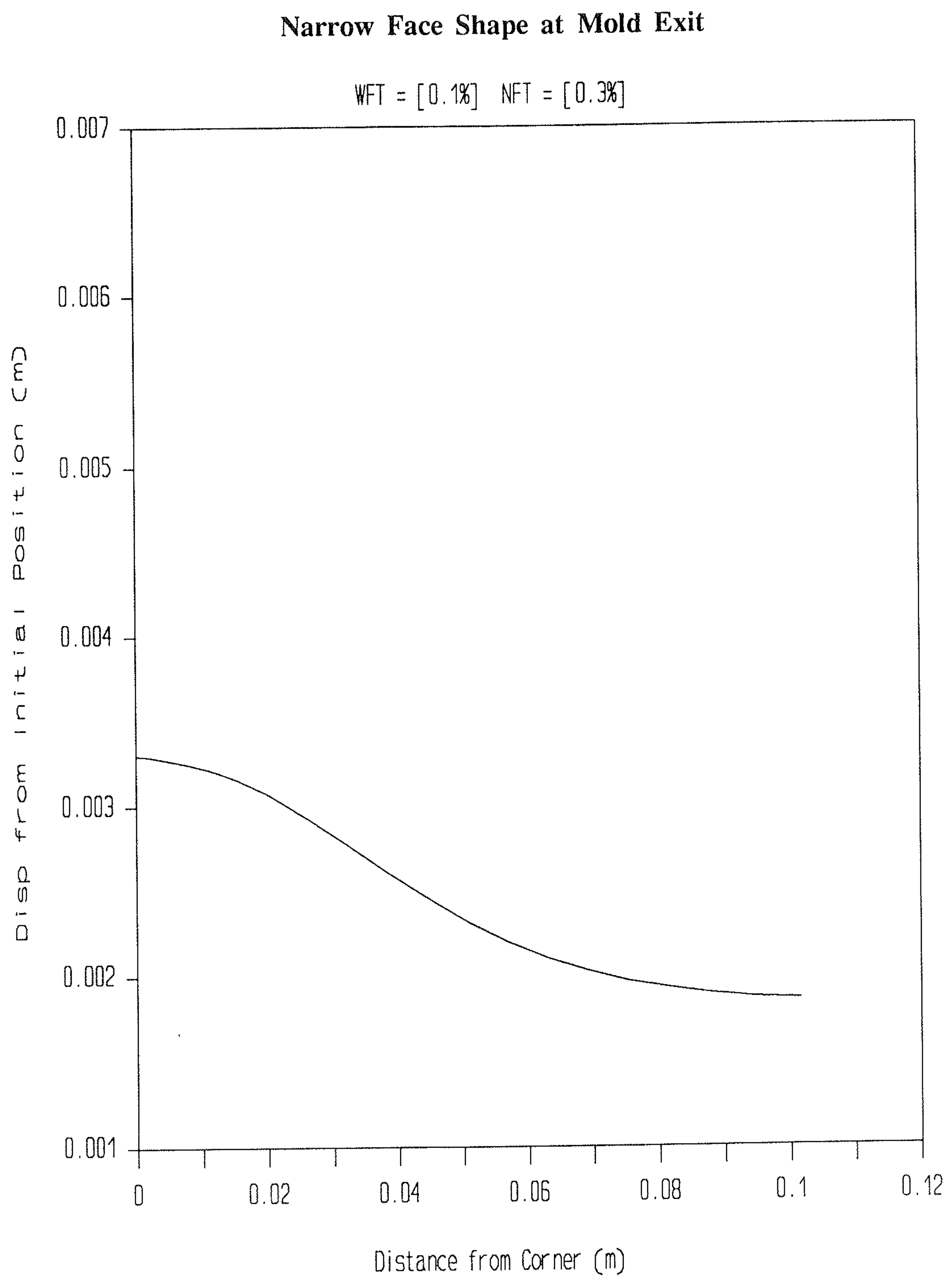


Figure 59 A cross-section through the forming shell at mold exit shows the shape of the narrow face.

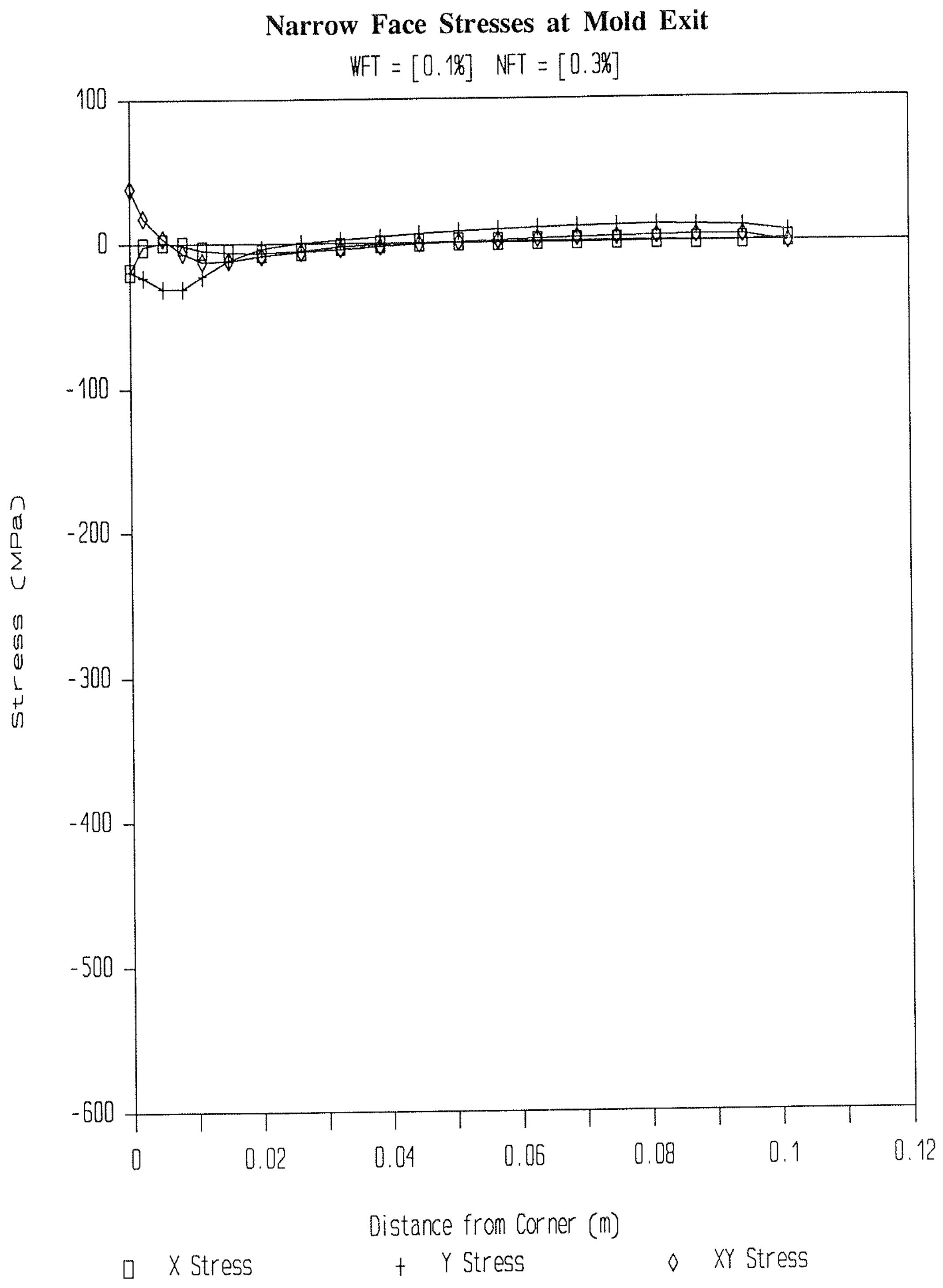


Figure 60 The three components of stress are shown for the external surface of the narrow face at mold exit.

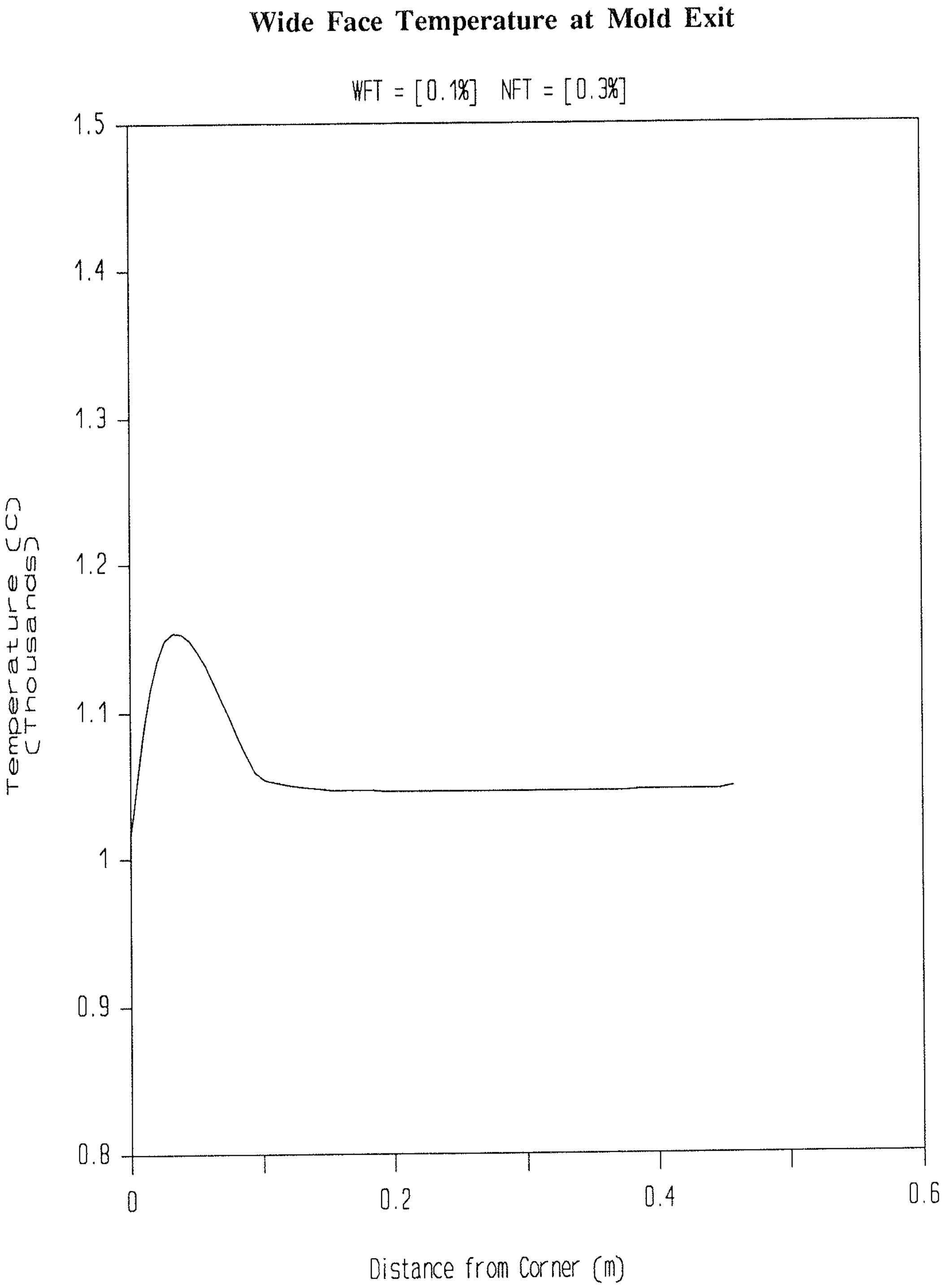


Figure 61 External surface temperatures at mold exit are shown along the wide face.

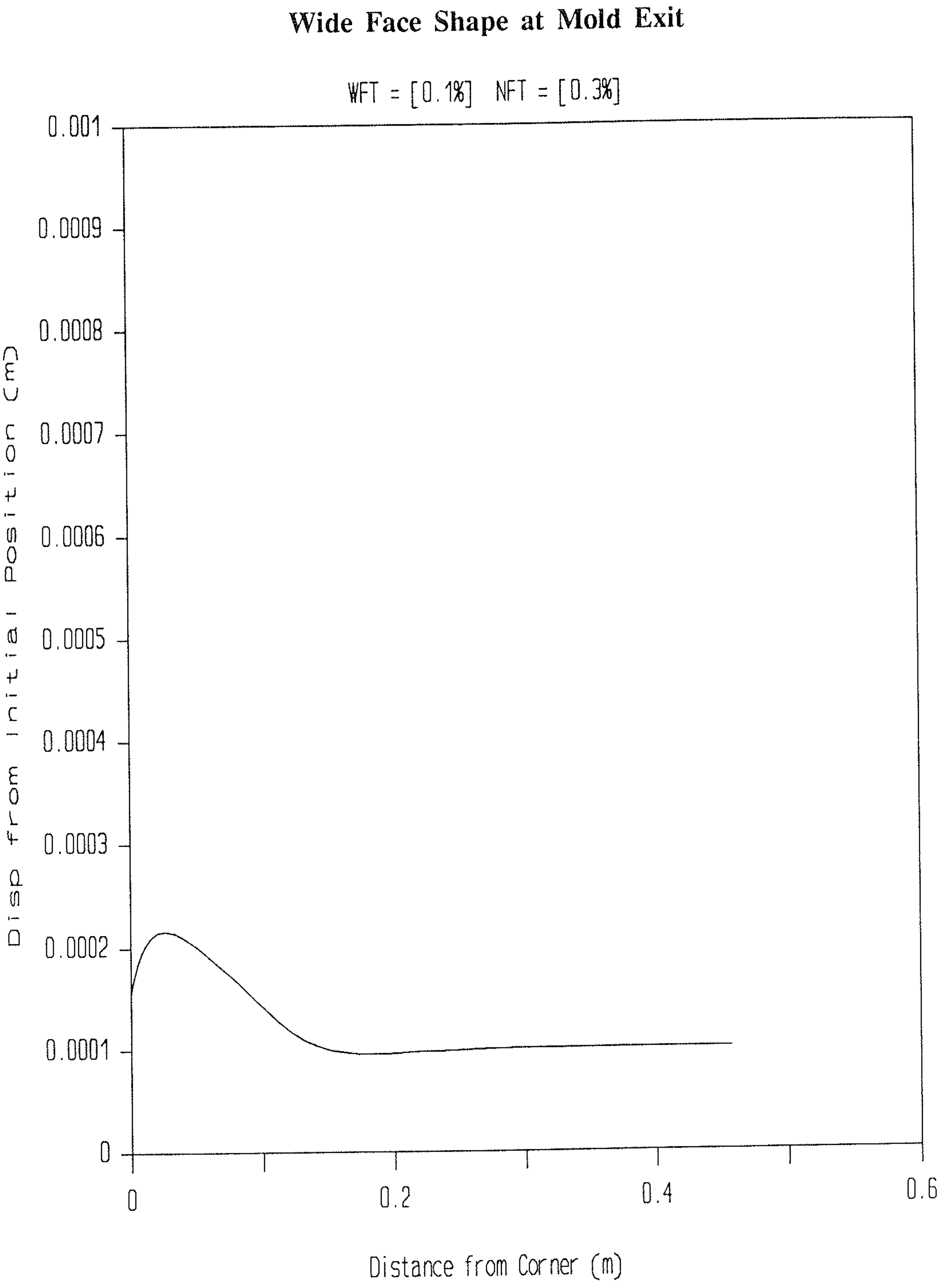


Figure 62 A cross-section through the forming shell at mold exit shows the shape of the wide face.

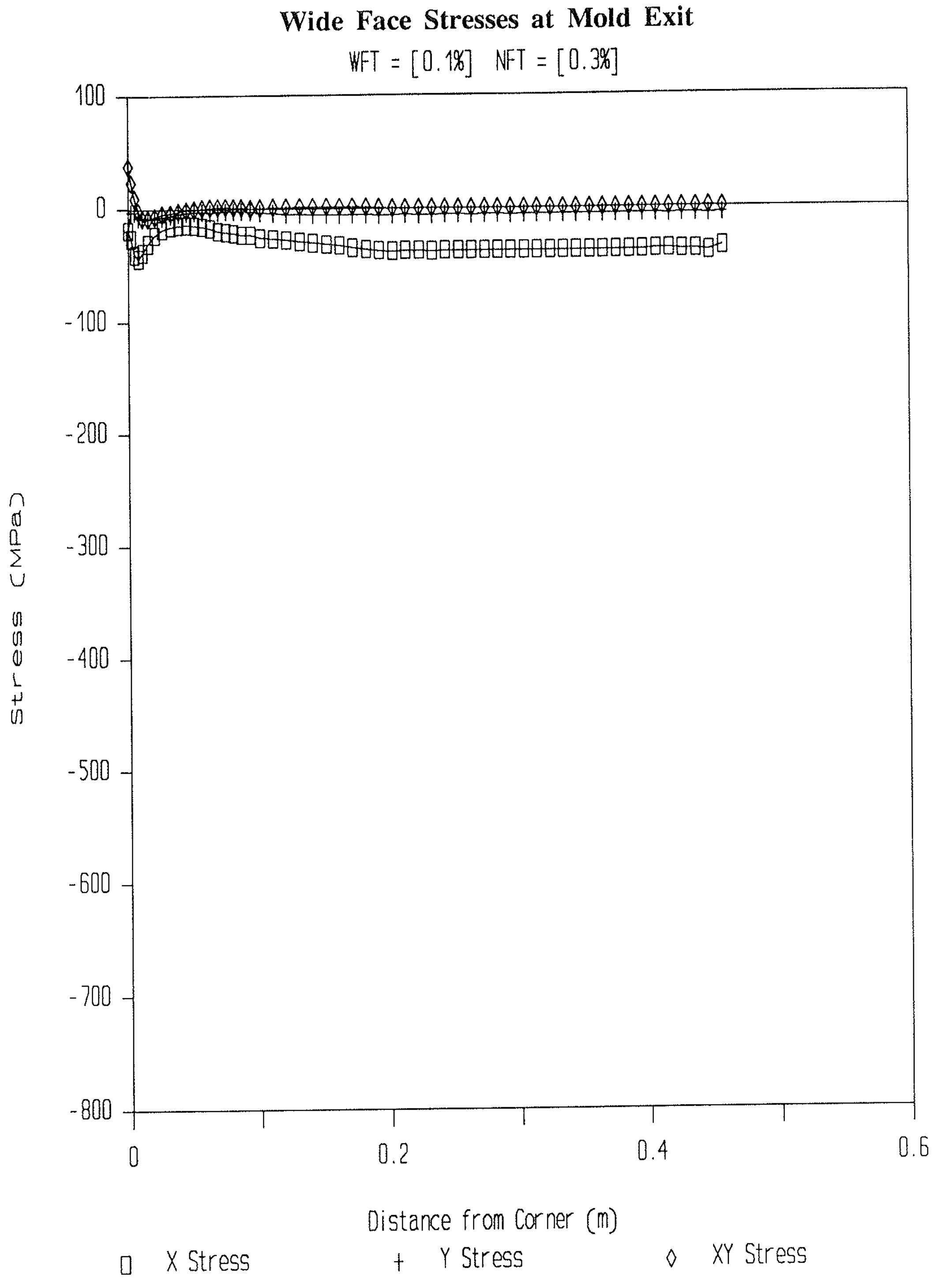


Figure 63 The three components of stress are shown for the external surface of the wide face at mold exit.

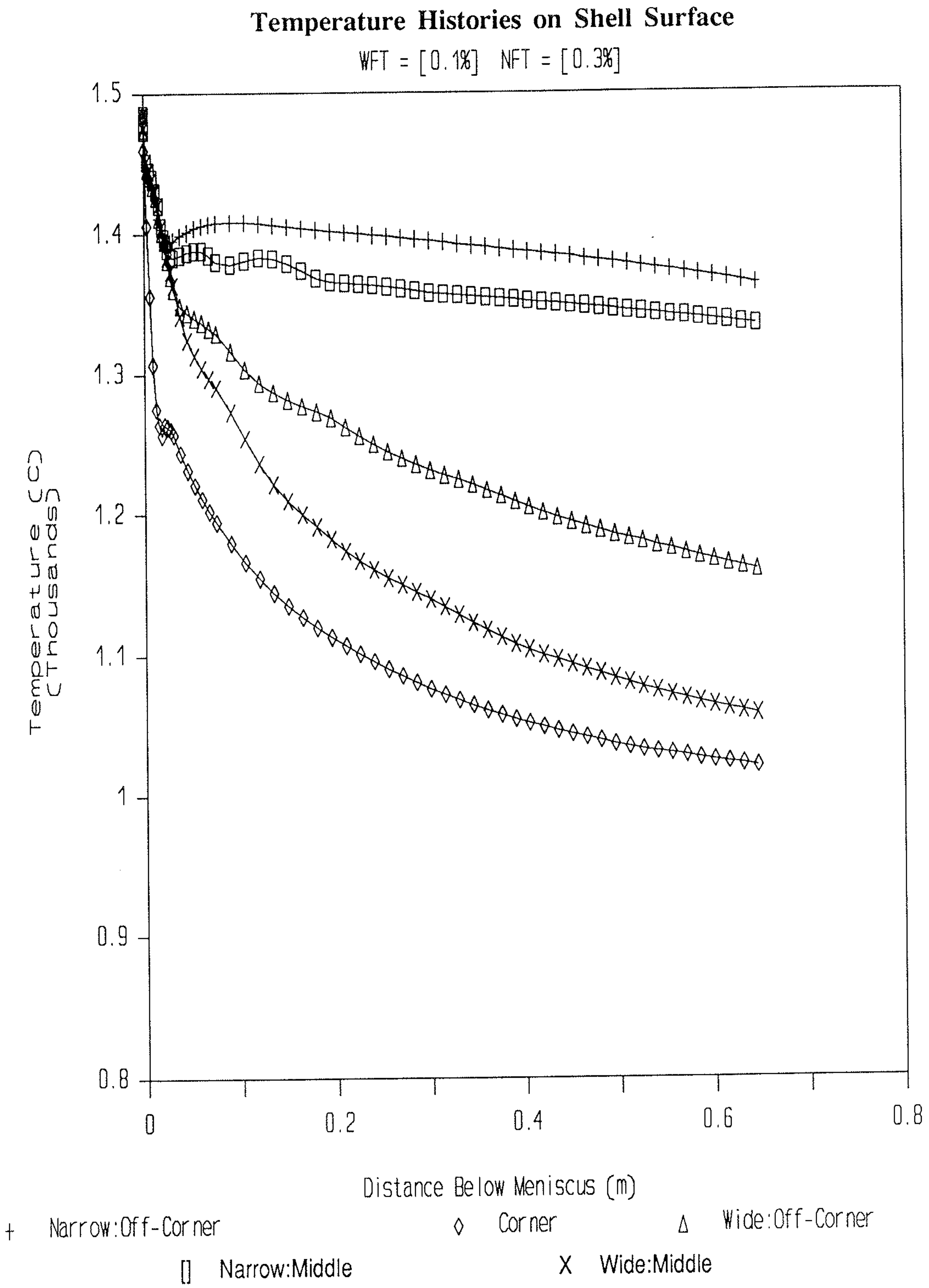


Figure 64 The temperature histories of some significant locations on the surface of the shell are shown over the mold solidification portion of the continuous casting process.

APPENDIX 6

OPTIMIZED TAPER DESIGN

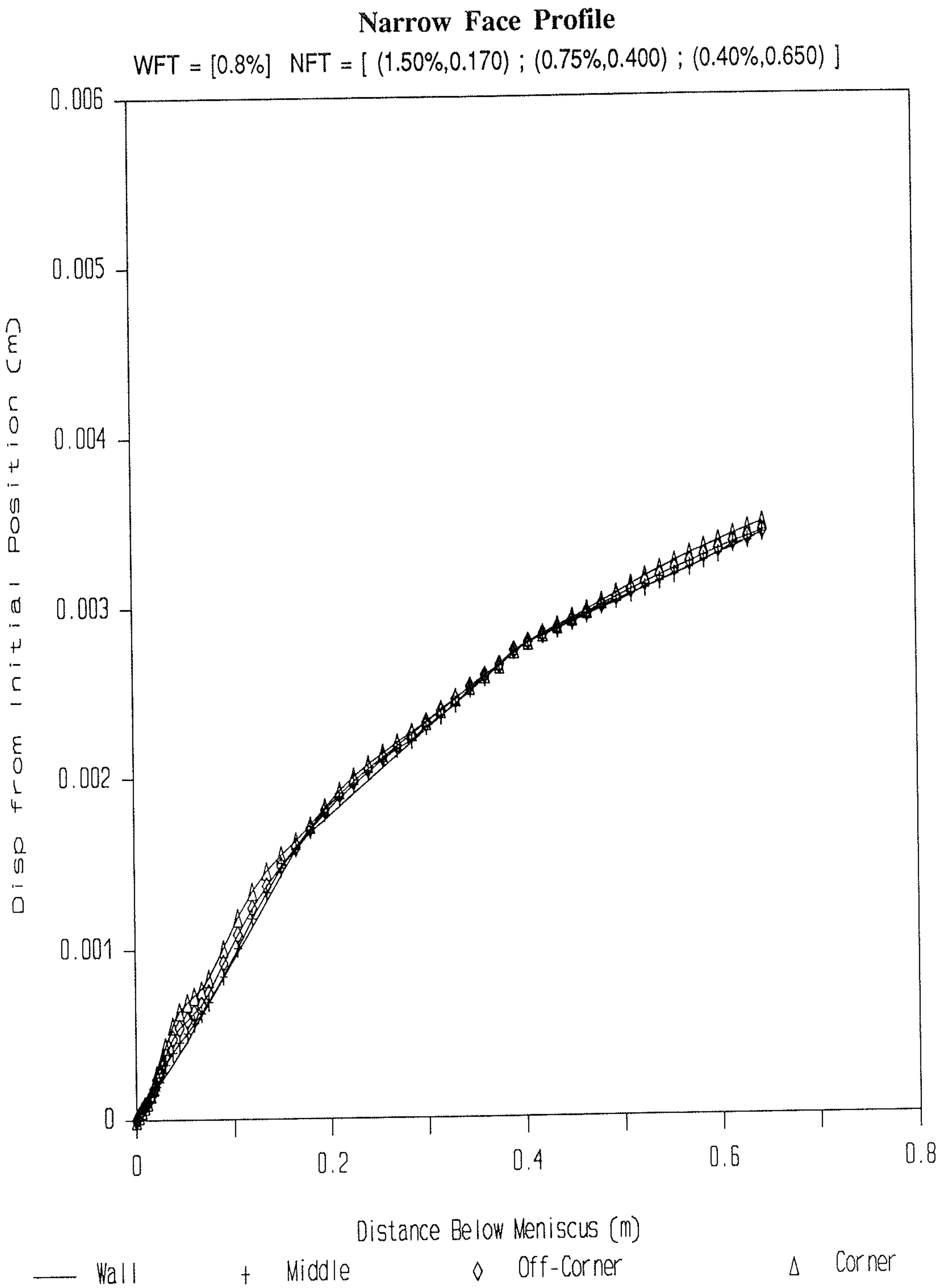


Figure 65 The shrinkage of the narrow face is displayed in relation to the mold wall position.

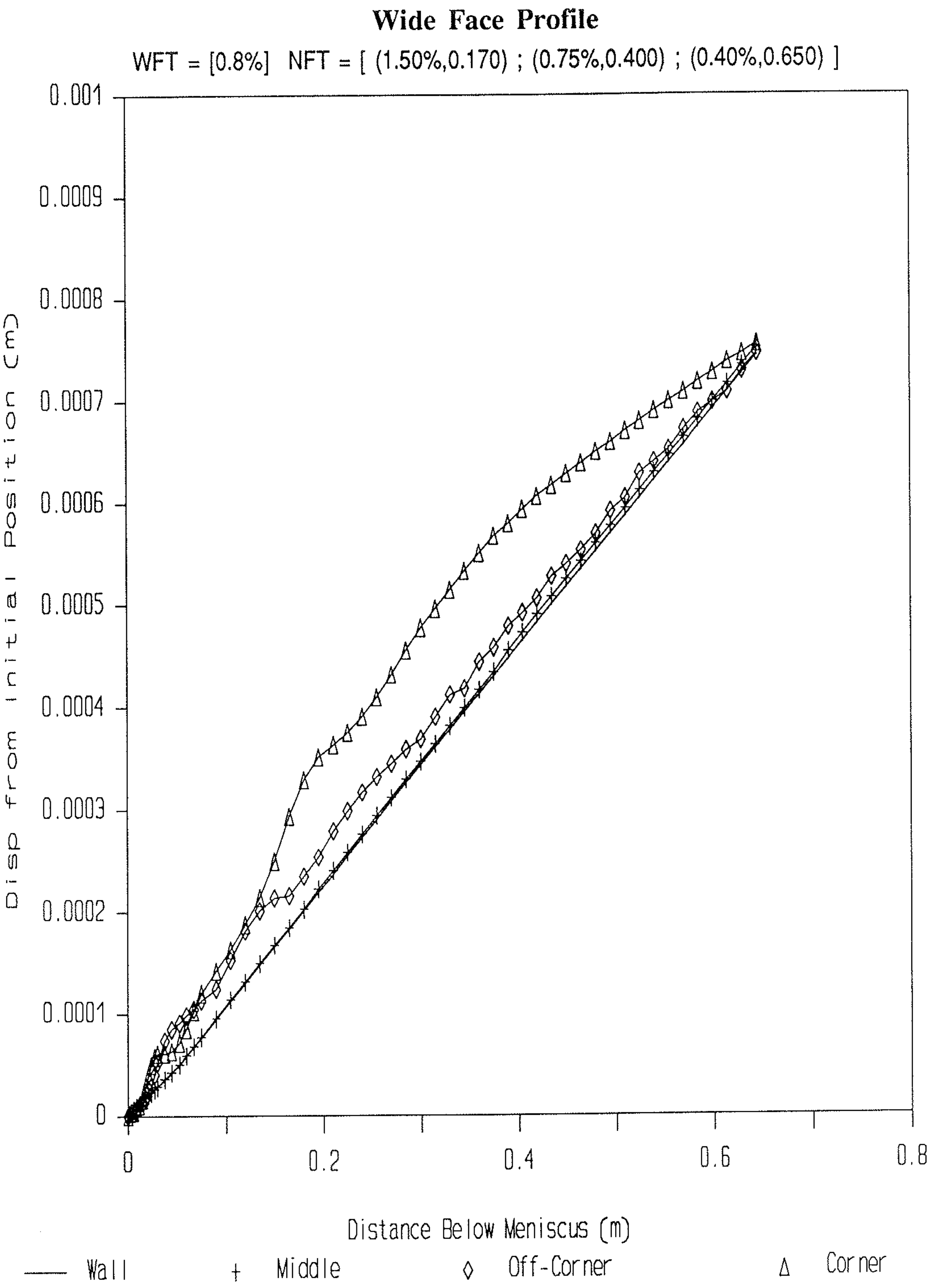


Figure 66 The shrinkage of the wide face is displayed in relation to the mold wall position.

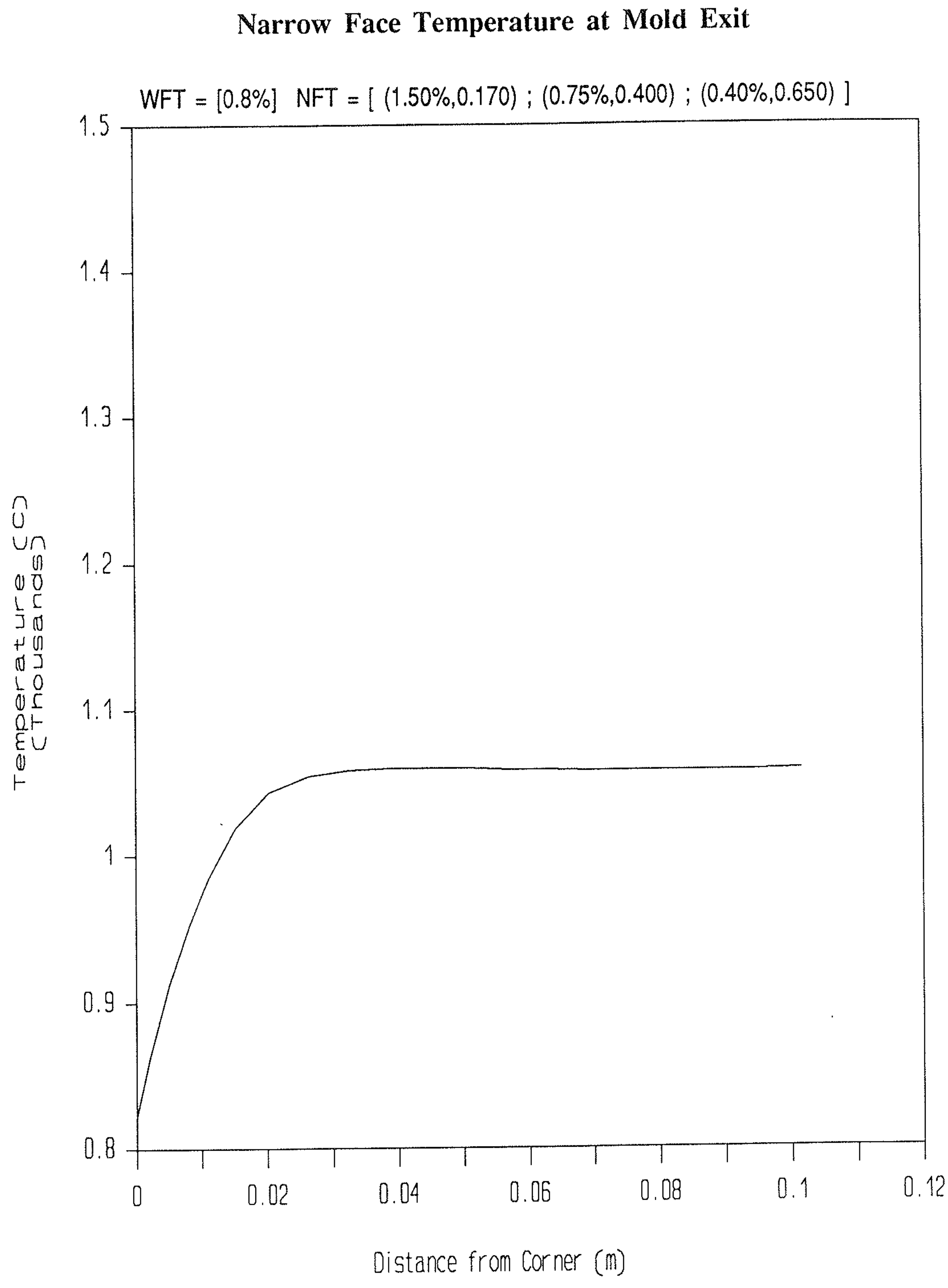


Figure 67 External surface temperatures at mold exit are shown along the narrow face.

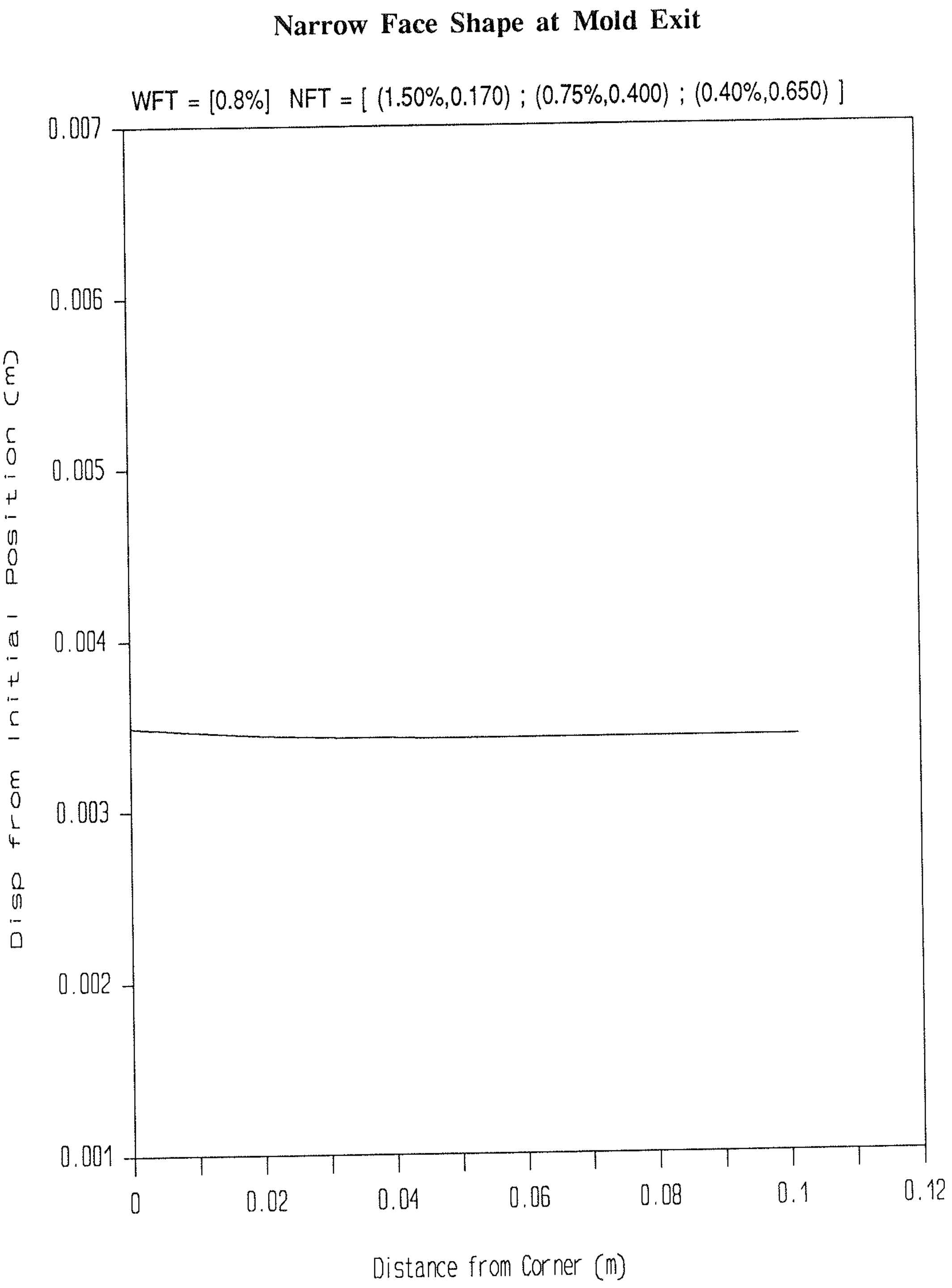


Figure 68 A cross-section through the forming shell at mold exit shows the shape of the narrow face.

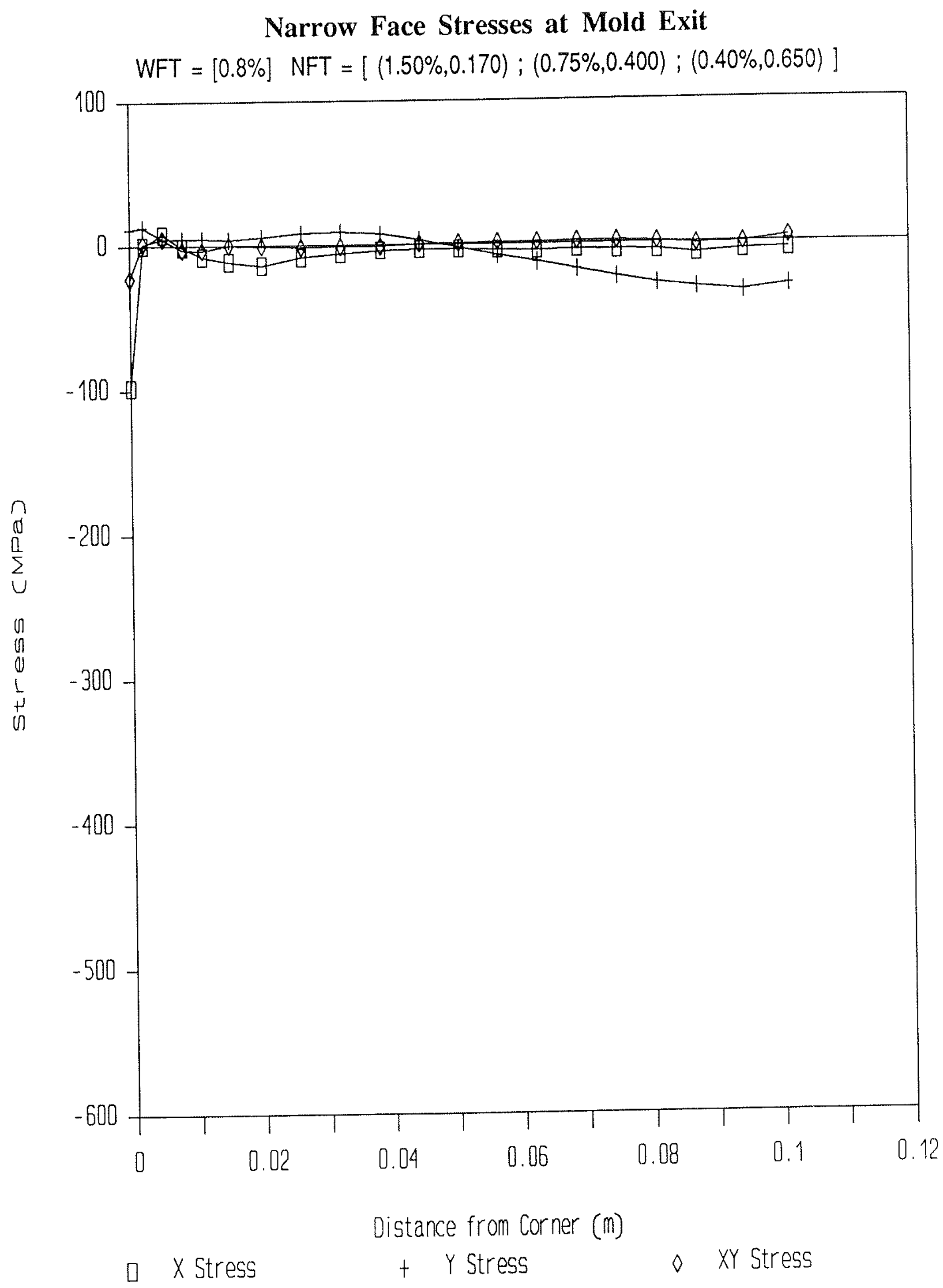


Figure 69 The three components of stress are shown for the external surface of the narrow face at mold exit.

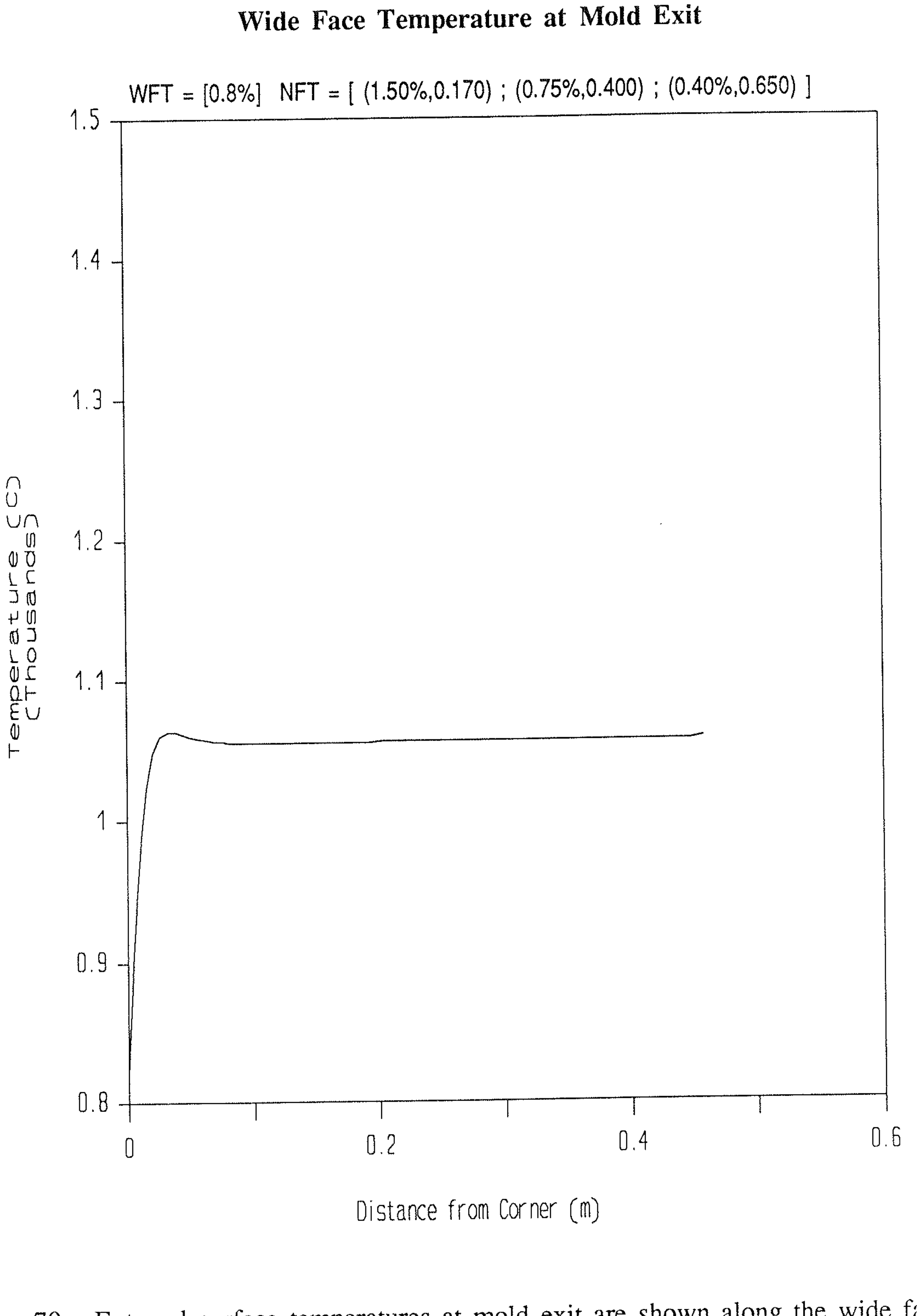


Figure 70 External surface temperatures at mold exit are shown along the wide face.

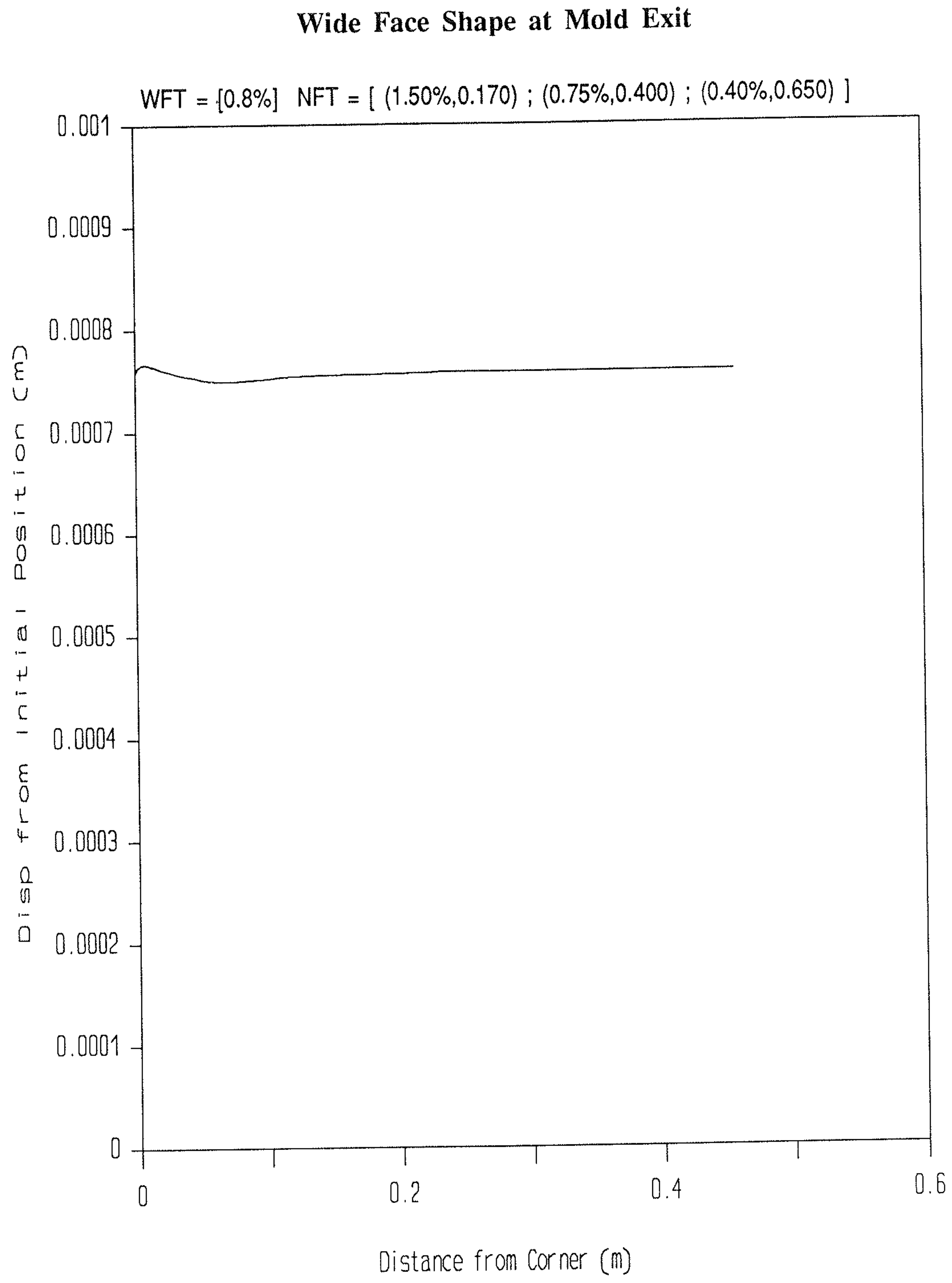


Figure 71 A cross-section through the forming shell at mold exit shows the shape of the wide face.

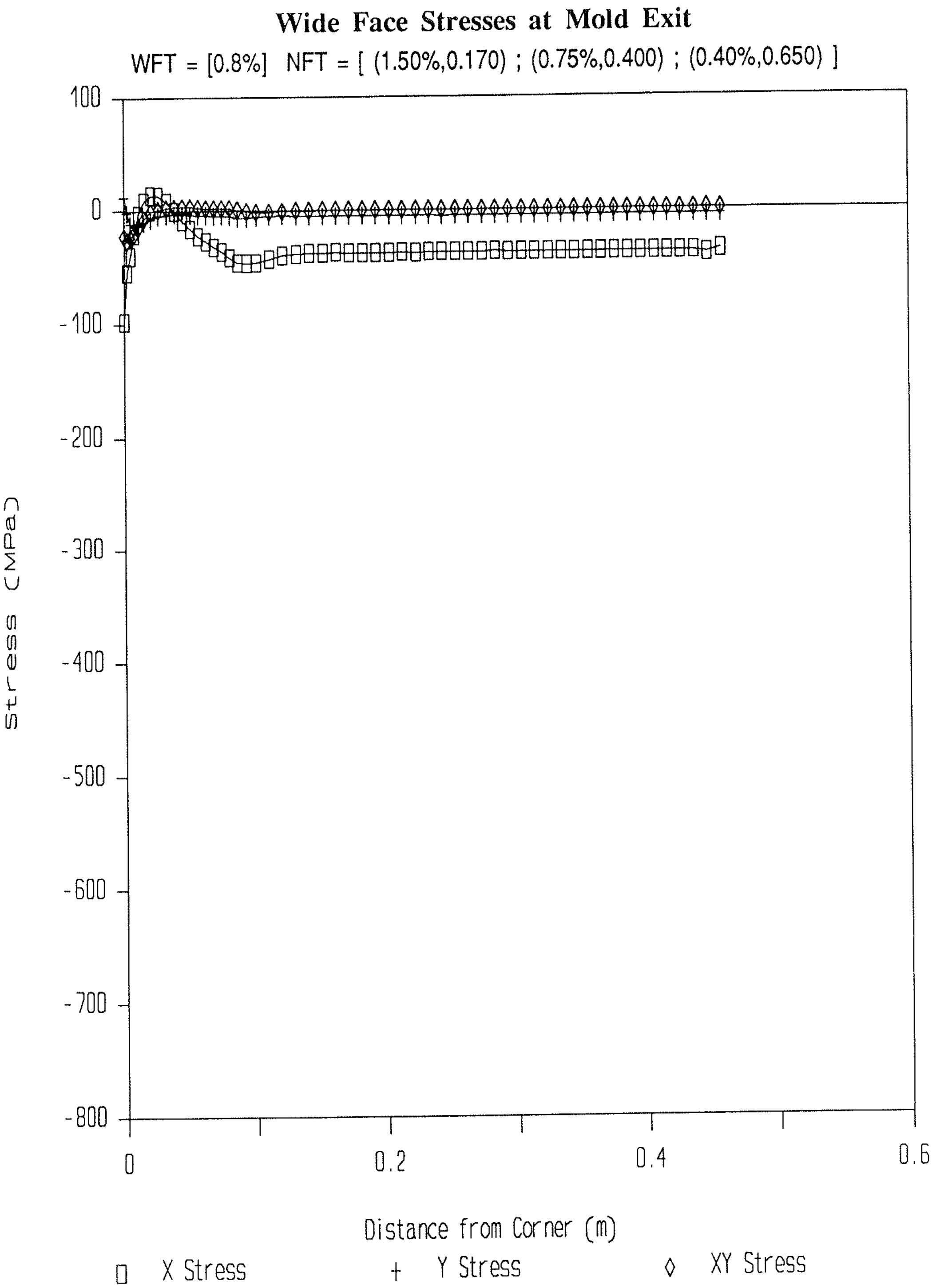


Figure 72 The three components of stress are shown for the external surface of the wide face at mold exit.

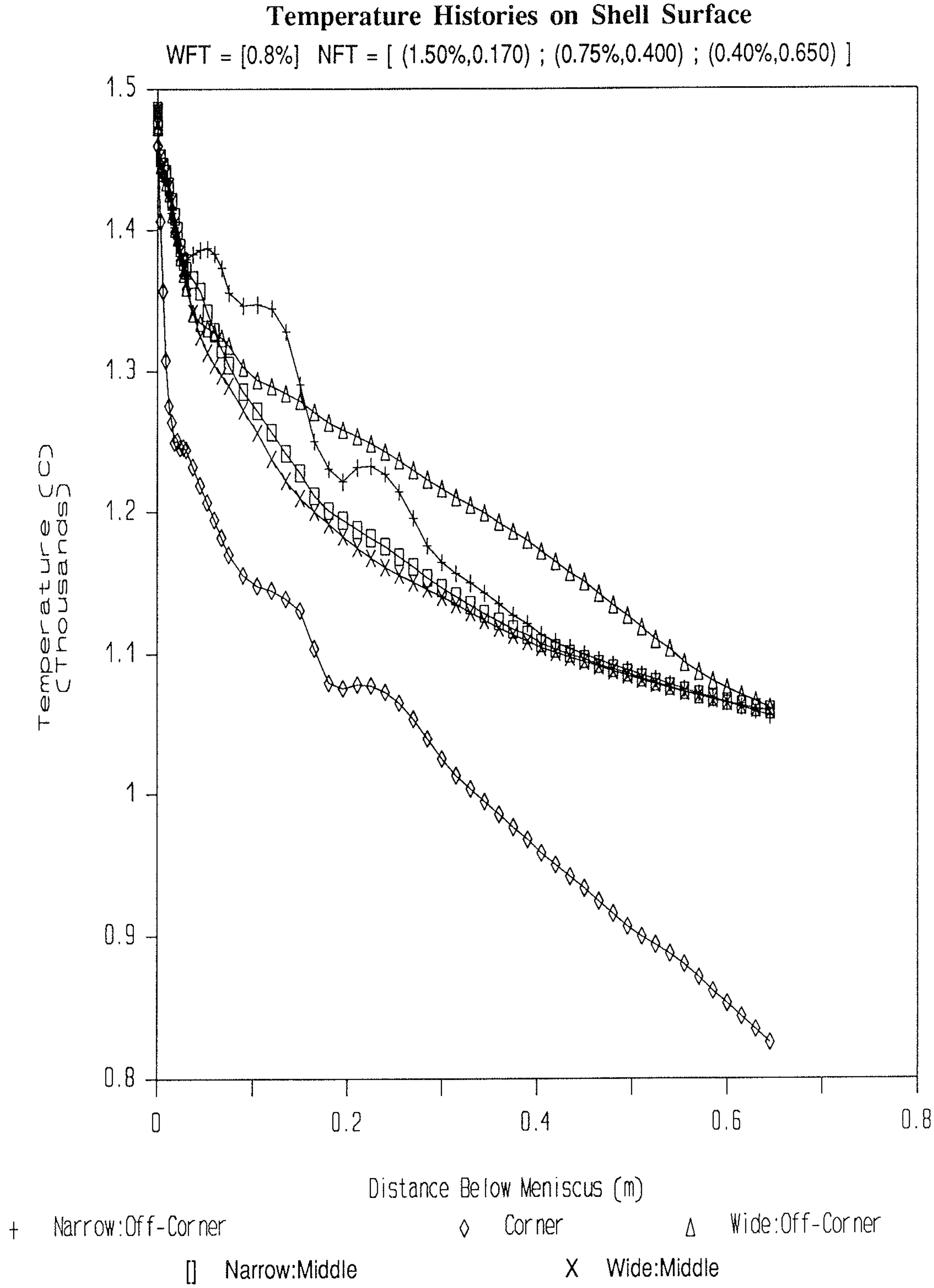


Figure 73 The temperature histories of some significant locations on the surface of the shell are shown over the mold solidification portion of the continuous casting process.