Pipe Circularity Reformation Via Line Heating

by

Rodrigo V. Andrade

B.S. in Mechanical Engineering (1995)
Naval Polytechnic Academy, Chile

Submitted to the Department of Ocean Engineering and the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degrees of

Master of Science in Naval Architecture and Marine Engineering

and

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2001

Author . . . .
Department of Ocean Engineering
August 10, 2001

Certified by . . . .
Nicholas M. Patrikalakis, Kawasaki Professor of Engineering,
Professor of Ocean and Mechanical Engineering
Thesis Co-Supervisor

Certified by . . . .
Takashi Maekawa
Lecturer and Principal Research Scientist
Thesis Co-Supervisor

Accepted by . . . .
Henrik Schmidt, Professor of Ocean Engineering
Chairman, Departmental Committee on Graduate Students
Department of Ocean Engineering

Accepted by . . . .
Ain A. Sonin, Professor of Mechanical Engineering
Chairman, Departmental Committee on Graduate Students
Department of Mechanical Engineering

Pipe Circularity Reformation Via Line Heating

by

Rodrigo V. Andrade

Submitted to the Department of Ocean Engineering
and the Department of Mechanical Engineering
on August 10, 2001, in partial fulfillment of the
requirements for the degrees of
Master of Science in Naval Architecture and Marine Engineering
and
Master of Science in Mechanical Engineering

Abstract

Fabrication of pipes requires the use of several manufacturing processes, such as bending, welding, drilling and wringing. However, in most cases the circular ends deviate from true circles and need reformation to be welded to flanges. Currently the reformation is conducted by hammering and relies on the intuition of skilled workers. This reforming process is not only expensive but also generates unhealthy loud noise.

The objective of this research is to develop an automatic system of circularizing the ends of a deformed pipe by laser line heating. The overall problem is defined as follows: Given the design of a metal pipe, measure the shape of the cross sections of both ends and a branch end of the manufactured pipe and determine the heating paths together with the heating conditions to reform the manufactured pipe to within acceptable tolerances with respect to the designed pipe using the line heating method.

The line heating conditions to be applied to the pipe have to be determined in real time to make the process efficient. A Neural Network is created for this purpose and the database used to run it is generated using a simplified thermo-mechanical model of the pipe validated by a Finite Element Model (FEM).

Thesis Co-Supervisor: Nicholas M. Patrikalakis, Kawasaki Professor of Engineering,
Title: Professor of Ocean and Mechanical Engineering

Thesis Co-Supervisor: Takashi Maekawa
Title: Lecturer and Principal Research Scientist

Acknowledgments

I would like to thank Dr. Takashi Maekawa for his continuous advice and help during this research and Professor Nicholas M. Patrikalakis for introducing me in this topic.

I thank my spouse and daughters for their love and care during my studies at MIT.

Funding for my studies at MIT was provided by the Chilean Navy. The MIT Fabrication Laboratory research in the area of this thesis was supported in part by Toshiba Corporation. Both are gratefully acknowledged.

To my wife Carola,
for her unconditional support
and infinite love

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and motivation

Fabrication of pipes used for switchgears requires the use of several manufacturing processes, such as bending, welding, drilling and wringing as shown in Figure 1-1. The complete process makes very difficult to preserve the exact circularity at the free ends of the pipe, which causes problems when it will be welded to a flange. The correction of circularity is currently conducted by hammering and is performed by skilled workers relying on their intuition and experience to obtain the desired result. Also, the process generates unhealthy loud noise and is also expensive; many years of on-the-job training are often necessary for

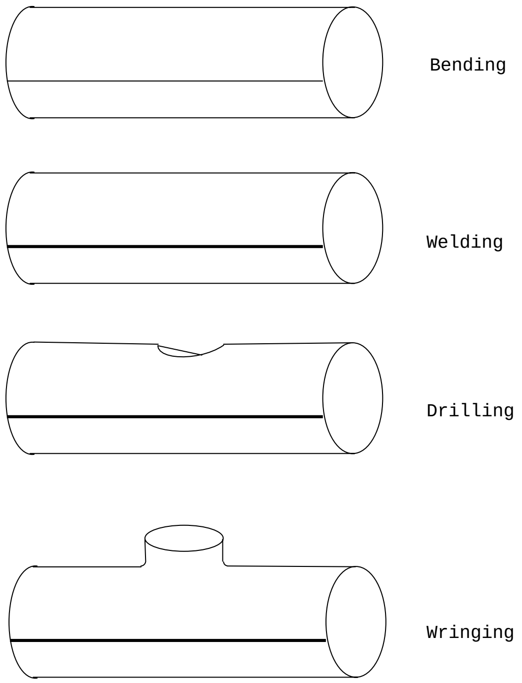


Figure 1-1: Processes involved in the manufacturing of pipes

a worker to master this skill through experience. Line heating appears at this point as an alternative to perform the reformation process to overcome the disadvantages of the reformation by hammering.

Line heating is a process of gradually adding plastic strain to a metal plate to generate the desired shape. When a plate is subjected to a thermo-mechanical process, plastic deformation is produced by the thermal stresses generated during the heating and subsequent cooling of the material. During this process, one side of the plate's surface is heated while the other side is kept at a lower temperature. The temperature gradient in the material across the thickness causes the metal to bend in one direction. In the mean time, the expanded metal is constrained by the surrounding cooler metal, and compressive stresses result. When the heat is removed, the material cools and the metal contracts (see Figure 1-2). The surface will then deform and assume an equilibrium state in the direction reverse to that when it was heated. The curvature generated is a function of temperature gradient between the top and bottom surfaces of the material. An ideal thermo-mechanical forming system would be able to heat a steel plate with desired temperature gradient at any point. The line-heating process is currently being used in a large number of shipyards to form hull plates. Three types of heat sources can be used in the line heating process: an oxyacetylene torch or a set of torches, induction heating and laser. Compared with mechanical pressing, thermo-mechanical forming using an oxyacetylene torch is more versatile and less expensive. However, line heating with an oxyacetylene torch has some inherent drawbacks. Forming by line heating is an art which requires many years of experience because complex mechanisms

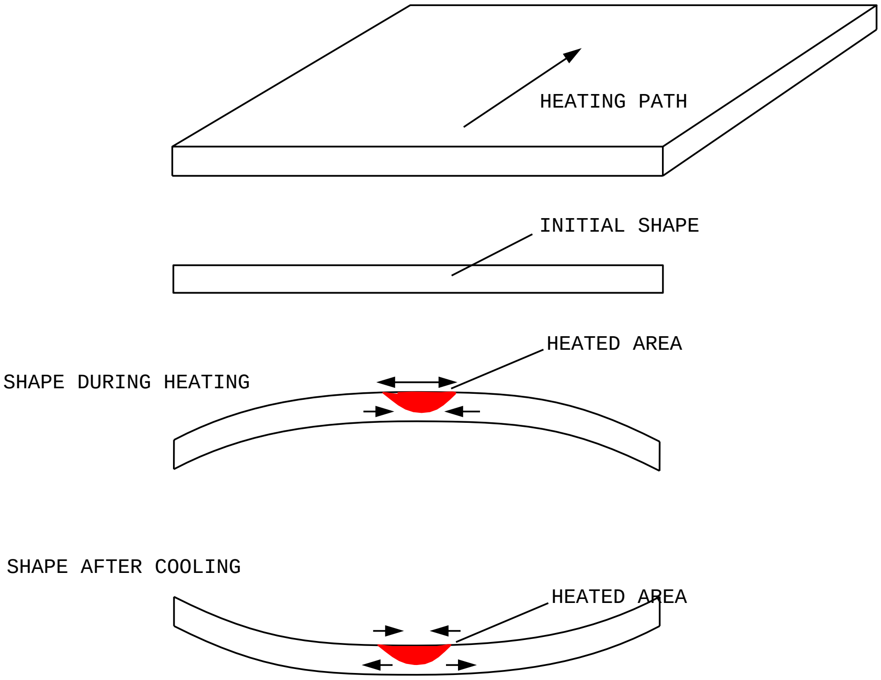


Figure 1-2: Line heating effect over a plate

are involved. In order to produce the exact desired shape, one must know how the plate should be heated. One must also have a means to control the heating and cooling processes. Compared with torch heating, induction heating is easier to control, and is used in some Japanese shipyards. However, the equipment is heavier and the heated area is larger, so induction heating is usually not performed manually. Instead, a robotic system is needed to control the heating process.

Laser forming is a thermo-mechanical method which uses a laser instead of an oxyacetylene torch as the heat source [17]. The basic metal forming mechanisms for laser forming are essentially the same as the forming technique using an oxyacetylene torch. Compared with the heat source of an oxyacetylene torch, a laser has the following advantages: (1) The power and its distribution are easier to control and reproduce. (2) The heated region is smaller so that material degradation (degradation of material properties due to line heating) is minimized. (3) A laser system can be integrated with a robotic system for automation of the line heating process. We will use this laser line heating technique for reformation of pipe ends so that they are nearly circular.

1.2 Research objective

This thesis research aims to develop an automatic system of circularizing the ends of a pipe which is used for switchgears by line heating. In other words, the thesis intends to solve the following problems:

(1) Modeling of the thermo-mechanical process over a pipe by using a non-linear three-dimensional finite element method.

(2) Modeling of the simplified thermo-mechanical process applied to a pipe which further reduces simulation time so that the model can be used in real time process planning.

(3) Designing of a Neural Network based on the information generated by using the models developed in (1) and (2) to efficiently predict the necessary heating conditions to reform the circularity on the pipe's free end.

1.3 Thesis outline

The remainder of the thesis is arranged as follows:

Chapter 2 presents a non-linear thermo-mechanical three-dimensional finite element model for the temperature field and the resulting deformation prediction of a pipe's free end due to line heating.

Chapter 3 presents a simplified thermo-mechanical model for the prediction of the temperature field and the resulting deformation of a pipe's free end due to line heating.

Chapter 4 presents a neural network application to determine in real time the heating parameters necessary to correct the deformation of a pipe.

Finally, Chapter 5 concludes the thesis, summarizes its contributions, and provides suggestions for future research.

Chapter 2

Coupled thermo-mechanical finite element analysis

2.1 Introduction

The process of correction of the circularity of a pipe by line heating is a coupled nonlinear thermo-mechanical process which makes its analytic simulation and prediction difficult to perform. Finite element analysis (FEM) is a suitable tool to achieve a good representation and prediction of the application and effects of the heating source. On the other hand, the finite element analysis takes a considerable computation time making it not suitable to obtain results in a short time. In this chapter a Finite Element Model based on ABAQUS software [1] is developed for thermo-mechanical analysis of the process of circularity correction of a pipe. The use of the FEM technique provides a way to verify, validate and compare the results obtained using the simplified thermo-mechanical model described in Chapter 3 of this thesis.

2.2 FEM model definition

2.2.1 Pipe geometry

The characteristics of the model used in this research are based on the pipes manufactured to be used on switchgears. The pipes are formed from plates using cold rolling, and the complete manufacturing process requires the use of bending, welding, drilling and wringing, which in most cases make the circular ends of the pipes deviate from true circles, thereby requiring reformation to be welded to flanges. The model consists of a pipe formed from a mild steel plate and its typical dimensions and characteristics are:

1. Length: 1.4 m

2. Internal diameter: 0.75 m

3. Thickness: 0.009 m
4. Material: Mild Steel

2.2.2 Finite element model mesh generation

Using ABAQUS we have developed a full 3-D FEM model for a pipe to perform a non-linear coupled thermo-mechanical FEM analysis and determine the relation between heating conditions and curvature distribution of the deformed cross section. The pipe can be treated as a thick pipe since it is the gradient of the temperature across the thickness that provides the mechanism to bend it. Therefore, a 3-D analysis is necessary and 3-D mesh needs to be generated. For our research the 20-node brick elements and 15-node triangular prism elements shown in Figure 2-1 are the type of elements used in the analysis. Mesh generation is carried out first on the upper or lower surface of the flattened rectangular plate of the pipe using quadrilateral and triangular elements as shown in Figure 2-2 . A dense 256 × 240 grid of points is generated on the flattened plate, which lie on the $x - y$ plane. The finest quadrilateral elements consist of a 3 × 3 grid, while the coarsest elements consist of a 17 × 17 grid of points. Then, using the reference system defined in Figure 2-3, a 3-D mesh can be generated by mapping onto a circular cylinder and offsetting across the pipe thickness. When the finest quadrilateral element is mapped onto the 3-D pipe, its dimension is 1.84 cm by 1.75 cm while the coarsest one is 14.7 cm by 14.0 cm. In order to accurately capture the characteristics of the laser forming process, we choose a mesh size which increases exponentially across the thickness of the pipe, being finer near the heated side of the pipe. The i^{th} layer thickness from the bottom layer is defined as

$$\delta z_i = \frac{t_k \cdot (1-r)}{1-r^{n_z}} r^{(n_z - i)} \quad i = 1, \ldots, n_z \ , \tag{2.1}$$

where δz_i is the i^{th} layer thickness starting from the bottom, t_k is the material thickness, n_z is the number of layers across the thickness, and r is defined by

$$r = ratio^{\frac{1.0}{n_z - 1}} \ , \tag{2.2}$$

where $ratio$ is the defined ratio between the bottom layer and the upper layer. For the simulation we used $n_z = 3$ and $ratio = 6$. The final generated mesh is shown in Figures 2-4 and 2-5. The first shows a general view of the pipe meshing and the second shows a closer view of the denser meshed area and the different layer heights across the thickness.

2.2.3 Thermal boundary conditions

Boundary heat transfer is modeled by natural heat convection and radiation. Convection follows Newton's law, according to which the rate of the loss of heat per unit area in Wm^{-2}

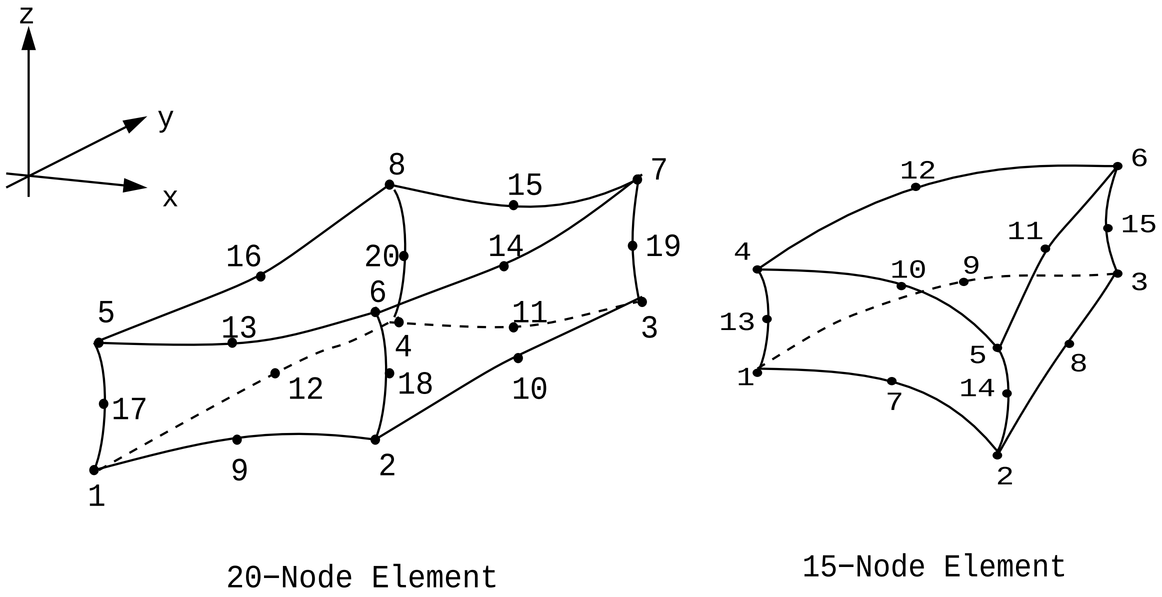


Figure 2-1: Elements used in FEM analysis

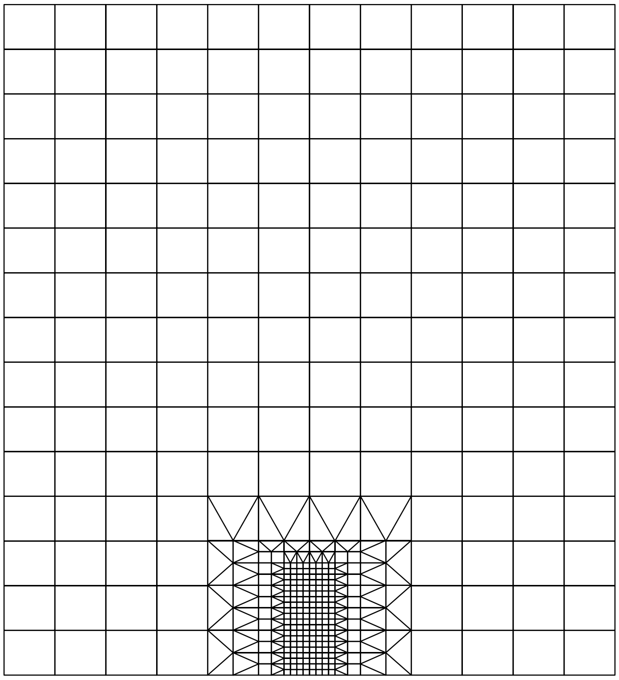


Figure 2-2: Initial mesh

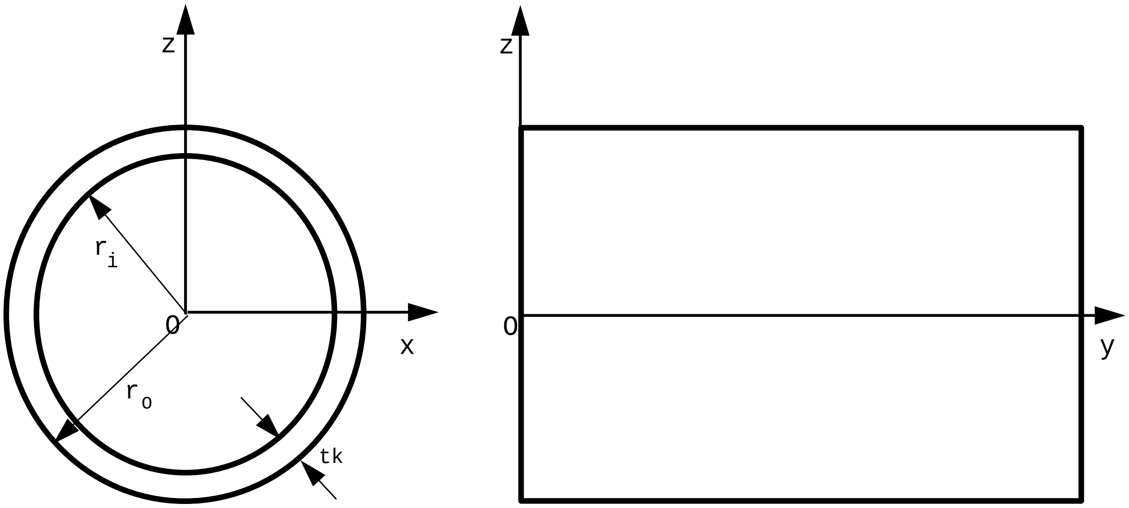


Figure 2-3: Coordinate system definition

due to convection is

$$q = h_c(T_s - T_a)\ , \tag{2.3}$$

where the coefficient of convective heat transfer h_c is a function of the difference between the wall temperature T_s and the environment temperature T_a and of the orientation of the boundary [19][22] given by:

$$h_c = \frac{k_s N_u}{L}\ , \tag{2.4}$$

where k_s is the thermal conductivity of the metal plate, N_u is the Nusselt number, and L is the characteristic length of the plate (or surface). Since the pipe diameter is relatively large compared with the diameter of the heating spot size we can treat the pipe as a horizontal plane surface. For horizontal plane surfaces with surface area A_s and perimeter p, the characteristic length is given by $L = A_s/p$. If we denote the Rayleigh number by Ra_L, the Nusselt number is defined by:

$$N_u = b(Ra_L)^m, \tag{2.5}$$

where for horizontal surfaces facing upwards,

$$\begin{aligned} b = 0.54, \quad m = \frac{1}{4}, \ \ when \ \ 10^4 < Ra_L \le 10^7\ , \\ b = 0.15, \quad m = \frac{1}{3}, \ \ when \ \ 10^7 < Ra_L < 10^{11}\ , \end{aligned} \tag{2.6}$$

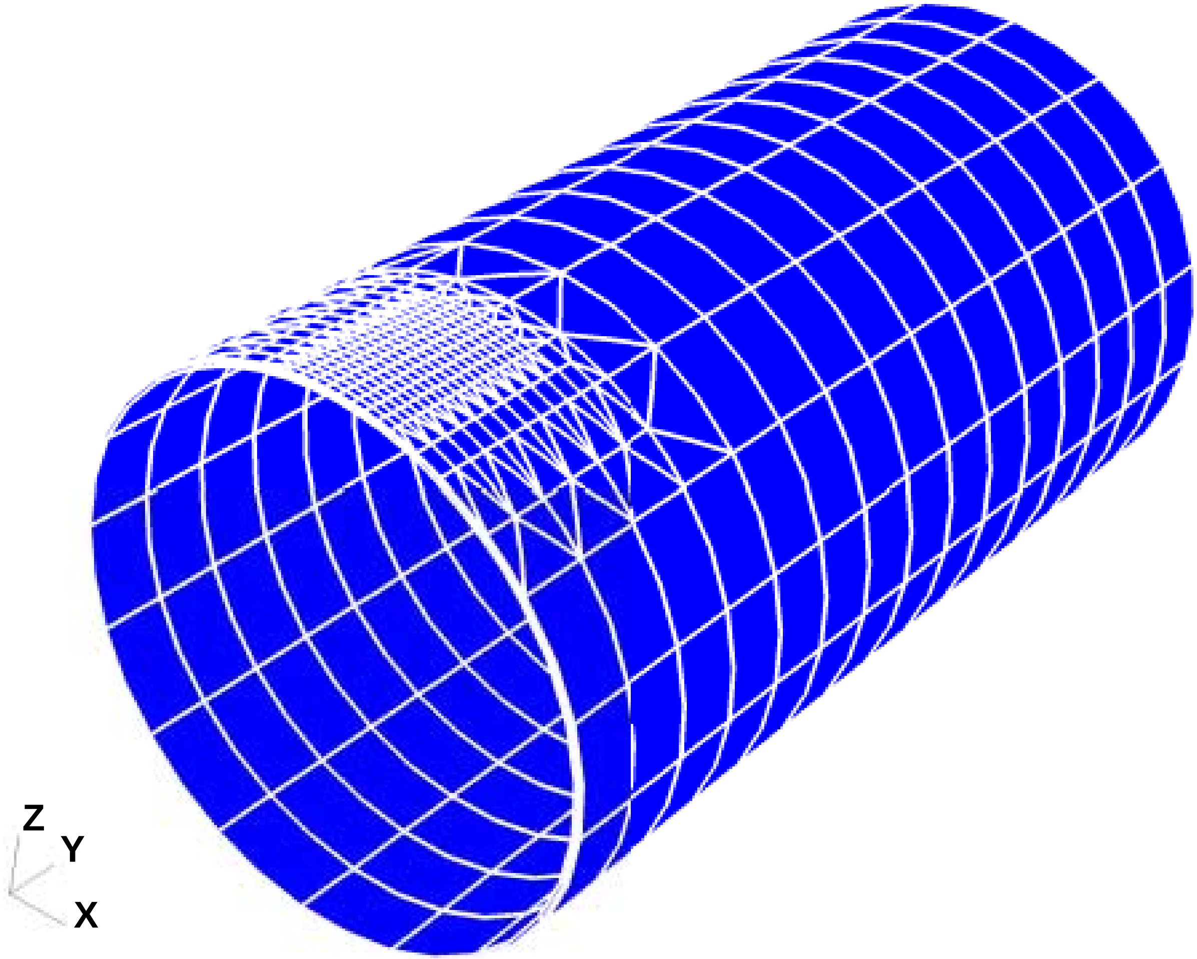


Figure 2-4: Pipe mesh general view

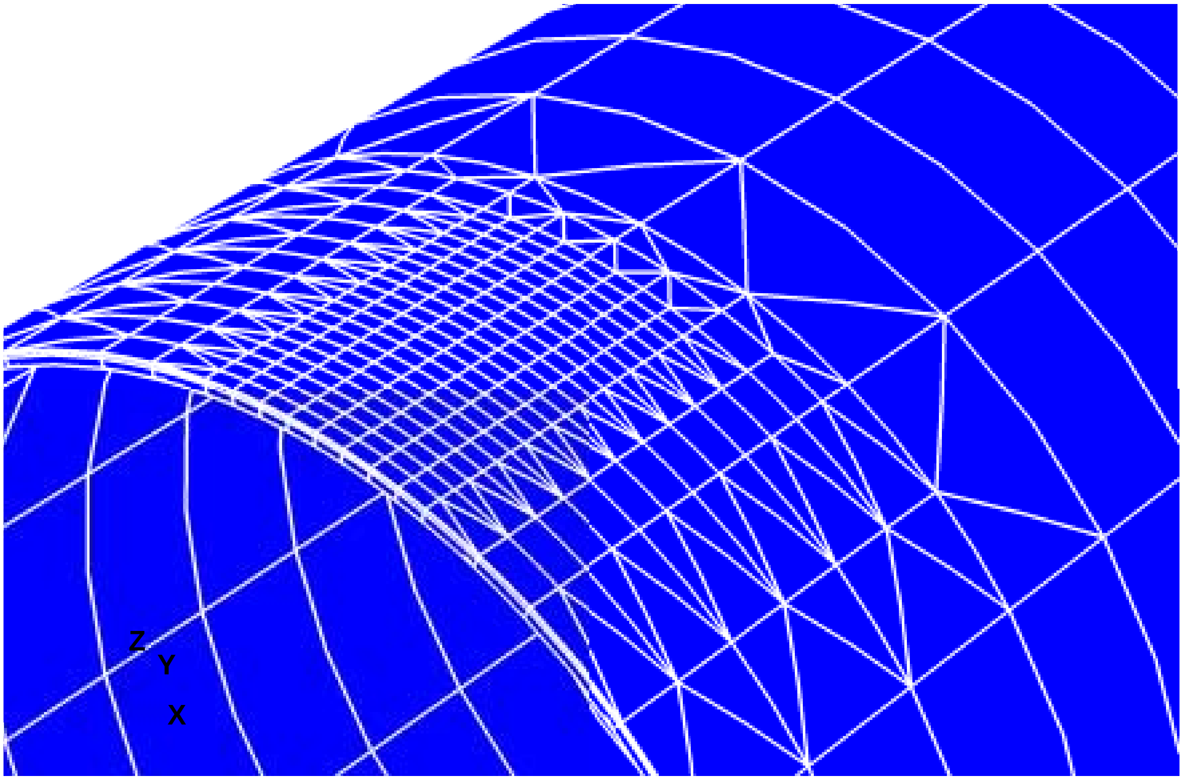


Figure 2-5: Pipe mesh thickness layers view

for horizontal surfaces facing downwards,

$$b = 0.27, \quad m = \frac{1}{4}, \;\; when \;\; 10^5 < Ra_L < 10^{11} \; . \tag{2.7}$$

The Rayleigh number is given by $Ra_L = Gr_L \cdot Pr$, where Gr_L is the Grashof number, and Pr is the Prandtl number. Both the Grashof number and the Prandtl number are functions of ambient air properties and temperature differences between the wall and the environment. The Grashof number is defined as

$$Gr_L = \frac{g\beta(T_s - T_a)L^3}{\nu^2} \; , \tag{2.8}$$

where g is the gravitational acceleration; β is the coefficient of thermal expansion of air; T_s and T_a are the temperatures (in degrees ^oC or K) of the metal plate and air, respectively; L is the characteristic length of the plate; ν is the kinematic viscosity of air. The Prandtl number Pr is defined as

$$Pr = \frac{\rho C_p}{k_a}\nu = \frac{\nu}{\alpha} \; , \tag{2.9}$$

where C_p is the specific heat of air, ρ the air density, k_a the thermal conductivity of air, and $\alpha = \frac{k_a}{C_p \rho}$ is the thermal diffusivity of air.

The rate of the loss of heat per unit area in Wm^{-2} due to radiation [19] is

$$q = 5.67 \times 10^{-8} \varepsilon (T_s^4 - T_a^4), \tag{2.10}$$

where ε is the surface emissivity (non-dimensional), whose value depends on the surface condition and the temperature of the metal plate. T_s and T_a are measured in degrees K.

Thermal properties of mild steel plates

The thermal conductivity k, specific heat C_p and convective heat transfer coefficients adapted from [23] for a mild steel pipe of the dimensions defined in Section 2.2.1 are shown in Table 2.1. In the table, "–" means either the data is not available (for thermal conductivity and specific heat) or was not calculated (for convective heat transfer coefficients).

Spatial distribution of the heat flux

Heat flux from an oxyacetylene torch or a laser beam is usually modeled as a Gaussian distribution [20]. Accurate measurements of energy distribution of the Nd:YAG laser system with fiber optic beam delivery and focus optics were performed using a charged coupled device (CCD) by researchers at the Applied Research Laboratory of Pennsylvania State University [21]. The Nd:YAG beam displays a Gaussian distribution with an annular lobe,

Temperature	Thermal conductivity k	Specific heat C_p	Convective heat transfer coefficient ($Wm^{-2}K^{-1}$)	
T (^oC)	$Wm^{-1}K^{-1}$	$Jkg^{-1}K^{-1}$	h_{up}	h_{down}
0	51.9	450	–	–
75	–	486	–	–
100	51.1	–	4.4468	1.8028
175	–	519	–	–
200	49.0	–	5.1405	2.1258
225	–	532	–	–
275	–	557	–	–
300	46.1	–	5.3252	2.2583
325	–	574	–	–
375	–	599	–	–
400	42.7	–	5.5800	2.3303
475	–	662	–	–
500	39.4	–	5.6701	2.4723
575	–	749	–	–
600	35.6	–	6.2027	2.6202
675	–	846	–	–
700	31.8	–	6.5913	2.8011
725	–	1432	–	–
775	–	950	–	–
800	26.0	–	6.7781	2.8881
900	–	–	7.0061	2.9515
1000	27.2	–	7.2161	3.0093
1100	–	–	7.4101	3.0626
1200	–	–	7.5911	3.1120
1500	29.7	400	–	–

Table 2.1: Thermal properties of mild steel

the amplitude of which is approximately 12% of the amplitude of the inner lobe. The outer lobe is believed to be a higher-order transverse mode caused by interaction of the beam and fiber. About 30% of the beam power is distributed in the outer lobe. The outer lobe has the shape of the sine (cosine) function. For the heating condition used for processing the Inconel plates, the inner lobe is 27.5 mm in diameter and the center of the outer lobe is 59.4 mm. Based on these data, the composite beam profile can be expressed as [22]:

$$q''(r) = \begin{cases} q_{max} e^{-cr^2} & r \leq r_2 \\ q_{max} \left[c_1 + c_2 \sin\left(\frac{r-r_2}{r_1-r_2}\frac{\pi}{2}\right)\right] & r > r_2 \end{cases} , \tag{2.11}$$

where q_{max}, r_2, c, c_1, c_2 are unknown variables, and $r_1 = \frac{59.4}{2} = 29.7$ mm. Denote Q the power of the laser, and p the absorption rate. The unknown variables satisfy the following conditions:

(1) At $r = r_0 = \frac{27.5}{2} = 13.75$ mm:

$$q_{max} e^{-cr_0^2} = 0.12 q_{max} , \tag{2.12}$$

(2) At $r = r_1 = 29.7$ mm:

$$q_{max} \left[c_1 + c_2 \sin\left(\frac{r-r_2}{r_1-r_2}\frac{\pi}{2}\right)\right] = 0.12 q_{max} , \tag{2.13}$$

(3) At $r = r_2$, compatibility between inner and outer regions:

$$q_{max} e^{-cr_2^2} = q_{max} c_1 , \tag{2.14}$$

(4) The inner region has heat flux $0.7Q \cdot p$:

$$2\pi \int_0^{r_2} q_{max} e^{-cr^2} r dr = 0.7Q \cdot p , \tag{2.15}$$

(5) The outer lobe has heat flux $0.3Q \cdot p$:

$$2\pi \int_{r_2}^{2r_1-r_2} \left[c_1 + c_2 \sin\left(\frac{r-r_2}{r_1-r_2}\frac{\pi}{2}\right)\right] r dr = 0.3Q \cdot p . \tag{2.16}$$

After solving the above 5 equations (2.12-2.16), we obtain

$$\begin{aligned} q_{max} &= p 6.4815 \times 10^6 \ W/m^2 , \\ c &= 1.1215 \times 10^4 / m^2 , \\ c_1 &= 6.80757 \times 10^{-4} , \\ c_2 &= 0.11932 , \end{aligned}$$

$$r_2 \quad = \quad 25.5mm \; .$$

The composite laser beam profile is shown in Figure 2-6. Energy distribution of the laser is

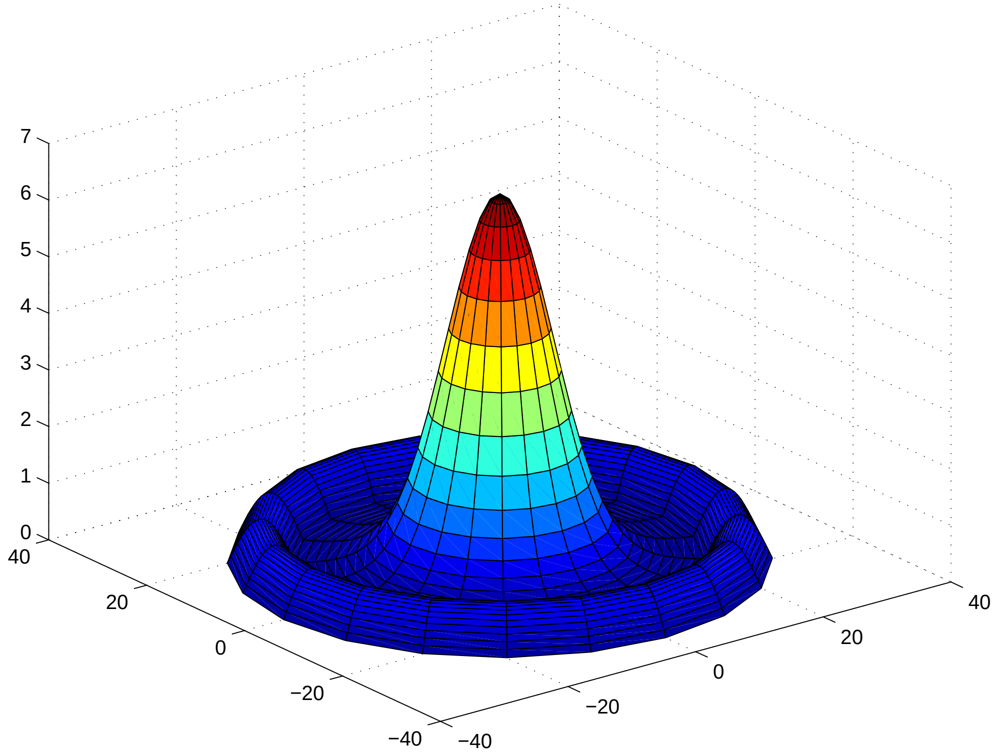


Figure 2-6: The composite laser profile (spatial heat distribution)

also characterized by the approximate beam diameter (spot size) as a function of distance from the focus optics to the work-piece (stand-off distance). Spot size was measured from burn patterns obtained from a Cotronix board, which is a fiber based low temperature refractory material, after a short period (2 seconds) of irradiation using various stand-off distances. The measured spot size for the above heat distribution is 22 mm, which corresponds to a stand-off distance of 18.5 cm. Researchers at the Applied Research Laboratory of Pennsylvania State University [21] suggest to use a Gaussian distribution within an equivalent diameter to simplify the heat flux distribution. The spot size diameter range measured by this institution for different stand-off distances and heating powers is used in our simulations as an aproximation of the dimension of the equivalent area heated with a Gaussian distribution. Twice the spot size will be defined as the area with a constant heating distribution.

When the spot size is enlarged by increasing the stand-off distance, the size of the inner region increases. Here we assume the size of the inner region increases proportionally to the spot size. For a spot size of 25.4 mm, the inner region has a diameter of 31.75 mm. To make the FEM simulation easier, the heat flux region is modeled as a truncated Gaussian distribution, along with a uniform distribution:

$$q''(r) = \begin{cases} q_{max}e^{-cr^2} & \text{if } r < 15.875\ mm \\ q_0 & \text{if } 15.875\ mm < r < 33.9\ mm \end{cases} \tag{2.17}$$

where $q_{max} = p \cdot 4.8740 \times 10^6 W/m^2$, $c = 8.4132 \times 10^3/m^2$, $q_0 = p \cdot 5.8488 \times 10^5 W/m^2$.

2.2.4 Mechanical properties of mild steel

Mechanical properties of the mild steel used for the simulation are given as follows [6] [5]:

1. Density: 7800 kg/m^3.
2. Mechanical properties are shown in Table 2.2. Young's modulus and yield stress are given small, finite values at high temperatures to avoid difficulties with numerical convergence [5].

Temperature	Yield	Young's	σ_y at strain	Thermal expansion
	stress	modulus	of 1.0	coefficient
T (^oC)	σ_y (MPa)	E (GPa)	(MPa)	α ($10^{-6} 1/^oC$)
0	290	200	314	10
100	260	200	349	11
300	200	200	440	12
450	150	150	460	13
550	120	110	410	14
600	110	88	330	14
720	9.8	20	58.8	14
800	9.8	20	58.8	14
1200	–	2	–	15
1550	0.98	0.2	1.0	

Table 2.2: Mechanical properties for mild steel

Mechanical boundary conditions

In mechanical analysis, necessary constraints are added to eliminate rigid body movement. The fixtures are defined in such way as to reduce the number of degrees of freedom and by making them similar to the fixtures used in real experiments. Figure 2-7 shows the boundary conditions used in the finite element model. To reduce the number of degrees of freedom we used a symmetry condition along the top and bottom centerlines nodes of the pipe, constraining them in the x direction (blue dots in Figure 2-7). Total fixture in the three directions was used at point B which is located at the bottom far end of the pipe, and partial constraints in the x and y direction were used at point T which is located at the pipe's top far end.

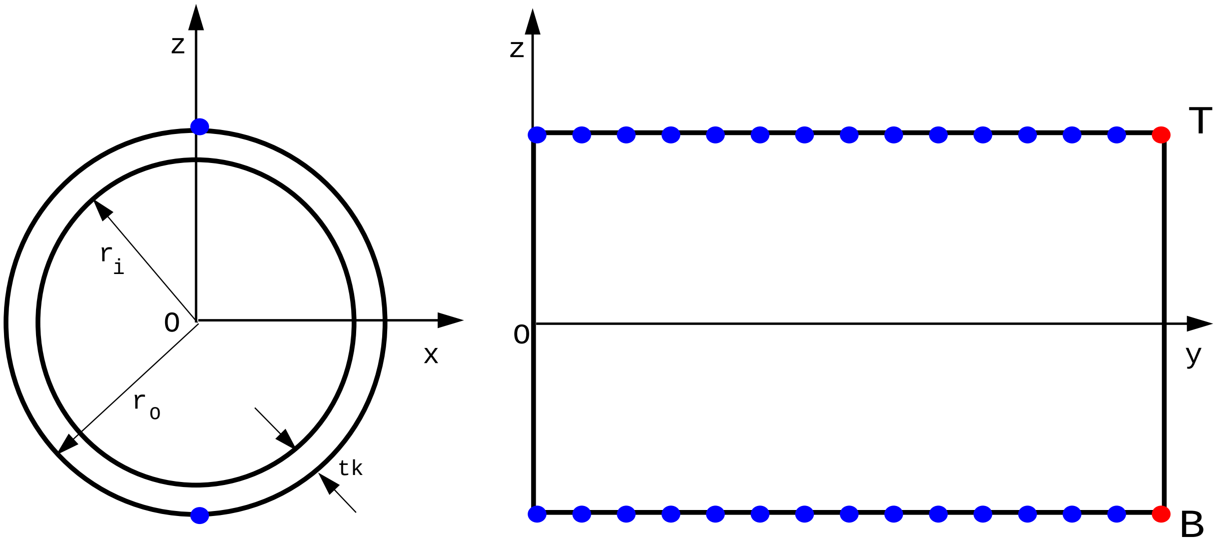


Figure 2-7: Model mechanical boundary conditions

2.3 Non-linear finite element analysis

For the FEM analysis we used a sequential coupled thermal-stress analysis. It assumes that the temperature field on the pipe can be found without knowledge or influence of the stress or deformation response [1]. Hence, an uncoupled heat transfer analysis can be conducted first and use then its results to perform a stress-deformation analysis.

2.3.1 Non-linear thermal analysis

We first conduct a thermal analysis of the pipe where we apply the heat over the line formed with the coordinates $x = 0, 0 \leq y \leq 0.25\ m, z = r_o$ (see Figure 2-3). The line heating application follows the time and speed settings defined by the user and starts at time $t = 0$ at the position $x = y = 0, z = r_o$. The computed resulting temperature field is stored in a separate file. As mentioned in the previous sections, the surface heating input is defined as a Gaussian distribution inside the inner lobe which concentrates about 70% of the total power and as a constant distribution in the outer lobe concentrating the other 30% of the total flux [22]. The heat along the pipe thickness follows a triangular distribution from the surface through a certain depth ϵ, defined as a fraction of the thickness [22][3]. We used a value of 0.09 for ϵ.

The input parameters used for the non-linear thermal and mechanical analysis example shown next are in Table 2.3

The output for the thermal analysis is the individual surface nodal temperature. Its maximum value for a node at the center of the heating path was 672.5 C^o. Figure 2-8 shows the surface temperature contours obtained during the thermal analysis. Simulations

Parameter	Value	Unit
Heating power	1300	kW
Heating speed	1.6	mm/s
Heating spot radius	12.7	mm
Heating absorption rate	0.81	

Table 2.3: Example input parameters

using different combinations of the parameters indicated above were performed and the results are indicated in Table 2.4. The parameters ranges were taken from the previous study of the effect of heating over a flat plate [23]. The results will be used in the next chapters to validate the simplified thermal model.

Case	Power input W	Source speed mm/s	Spot size radius mm	Temperature C^o
1	1300	1.6	12.7	672.6
2	1000	1.6	11.8	552.8
3	700	1.0	8.9	611.3
4	1100	1.3	10.8	732.3
5	800	0.7	10.8	646.7
6	600	0.3	12.7	528.1
7	900	1.0	11.8	566.9
8	600	0.7	8.9	527.9

Table 2.4: Non-linear FEM maximum nodal temperature prediction

2.3.2 Non-linear mechanical analysis

After the thermal analysis is completed, we conduct the non-linear FEM mechanical simulation. It uses the previously stored temperature data and analyzes the mechanical effects over the pipe due to the variation of the temperature field in time. The 20-node brick element and the 15-node triangular prism elements are of the second order type [1], and use isoparametric interpolation between their nodes defined by the local coordinates r, s, t shown in Figure 2-9. Gauss integration [9] is used to provide the most accurate strain prediction at the interpolation points. The isoparametric formulation spans the range -1 to $+1$ in the elements and provides the local coordinates to define the displacement vector

$$\mathbf{u}(r,s,t) = (u(r,s,t), v(r,s,t), w(r,s,t))^T \ , \tag{2.18}$$

within each element. Therefore for element m we have [4]

$$\mathbf{u}^{(\mathbf{m})}(r,s,t) = \mathbf{H}^{(\mathbf{m})}\,\hat{\mathbf{U}} \ , \tag{2.19}$$

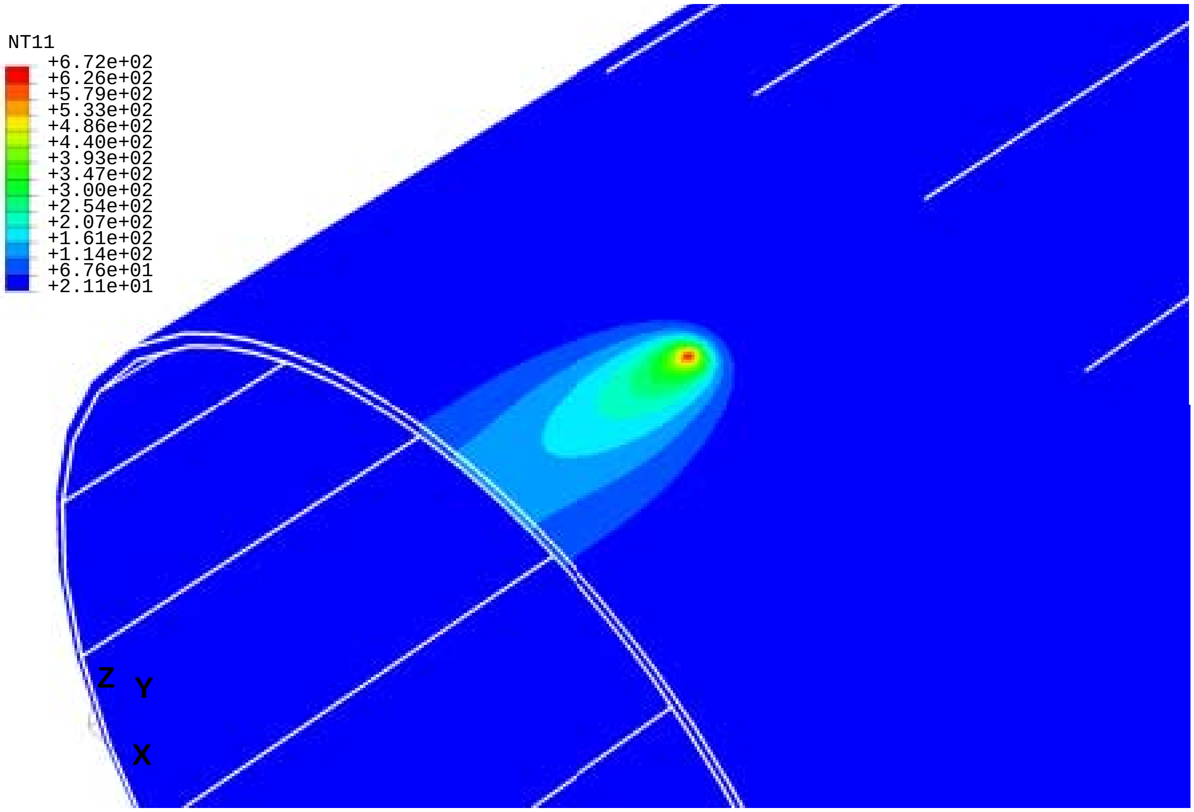


Figure 2-8: FEM line heating simulations over a pipe surface: temperature contours. Color temperature scale is in degrees Celsius

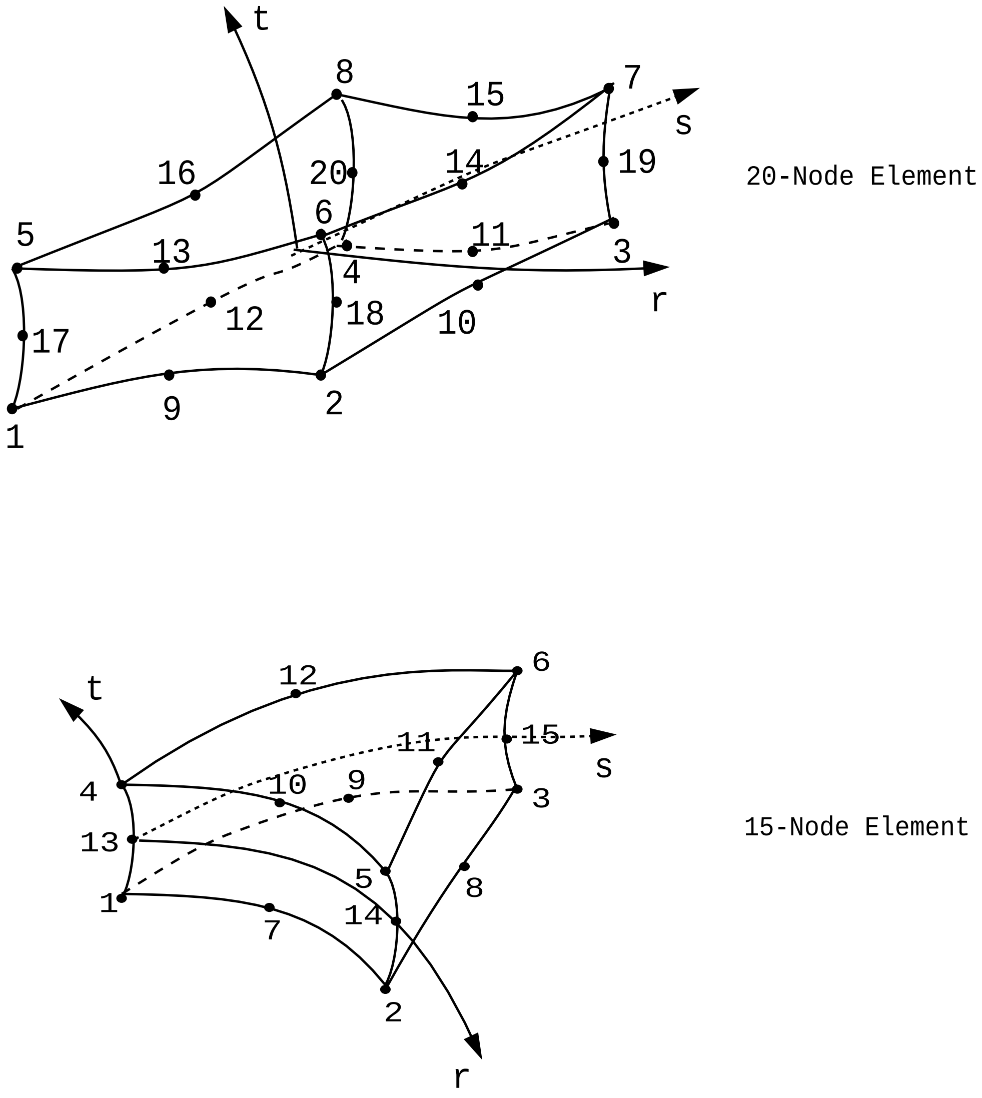


Figure 2-9: Isoparametric coordinates definition for brick and prism elements

where $\hat{\mathbf{U}}$ is the vector that represents the three global displacement components U_i, V_i and W_i at all nodal points given by

$$\hat{\mathbf{U}} = [U_1 V_1 W_1 \quad U_2 V_2 W_2 \ldots\ldots U_n V_n W_n]^T \ , \tag{2.20}$$

where n is the number of nodes in element m. The matrix $\mathbf{H^{(m)}}$ is the displacement interpolation matrix given by

$$\mathbf{H^{(m)}} = \begin{bmatrix} h_1\ 0\ 0 & h_2\ 0\ 0 & h_3\ 0\ 0 \ldots\ldots h_n\ 0\ 0 \\ 0\ h_1\ 0 & 0\ h_2\ 0 & 0\ h_3\ 0 \ldots\ldots 0\ h_n\ 0 \\ 0\ 0\ h_1 & 0\ 0\ h_2 & 0\ 0\ h_3 \ldots\ldots 0\ 0\ h_n \end{bmatrix} \tag{2.21}$$

where $h_i, i = 1, \ldots, n$ are the interpolation functions corresponding to the i^{th} node. For a 20-node brick element $h_i, i = 1, \ldots, 20$ are defined by [1]

$$\begin{aligned}
h_1 &= -\frac{1}{8}(1-r)(1-s)(1-t)(2+r+s+t) \ , \qquad (2.22) \\
h_2 &= -\frac{1}{8}(1+r)(1-s)(1-t)(2-r+s+t) \ , \\
h_3 &= -\frac{1}{8}(1+r)(1+s)(1-t)(2-r-s+t) \ , \\
h_4 &= -\frac{1}{8}(1-r)(1+s)(1-t)(2+r-s+t) \ , \\
h_5 &= -\frac{1}{8}(1-r)(1-s)(1+t)(2+r+s-t) \ , \\
h_6 &= -\frac{1}{8}(1+r)(1-s)(1+t)(2-r+s-t) \ , \\
h_7 &= -\frac{1}{8}(1+r)(1+s)(1+t)(2-r-s-t) \ , \\
h_8 &= -\frac{1}{8}(1-r)(1+s)(1+t)(2+r-s-t) \ , \\
h_9 &= \frac{1}{4}(1-r)(1+r)(1-s)(1-t) \ , \\
h_{10} &= \frac{1}{4}(1-s)(1+s)(1+r)(1-t) \ , \\
h_{11} &= \frac{1}{4}(1-r)(1+r)(1+s)(1-t) \ , \\
h_{12} &= \frac{1}{4}(1-s)(1+s)(1-r)(1-t) \ , \\
h_{13} &= \frac{1}{4}(1-r)(1+r)(1-s)(1+t) \ , \\
h_{14} &= \frac{1}{4}(1-s)(1+s)(1+r)(1+t) \ , \\
h_{15} &= \frac{1}{4}(1-r)(1+r)(1+s)(1+t) \ , \\
h_{16} &= \frac{1}{4}(1-s)(1+s)(1-r)(1+t) \ ,
\end{aligned}$$

$$\begin{aligned}
h_{17} &= \frac{1}{4}(1-t)(1+t)(1-r)(1-s)\ , \\
h_{18} &= \frac{1}{4}(1-t)(1+t)(1+r)(1-s)\ , \\
h_{19} &= \frac{1}{4}(1-t)(1+t)(1+r)(1+s)\ , \\
h_{20} &= \frac{1}{4}(1-t)(1+t)(1-r)(1+s)\ ,
\end{aligned}$$

and for the 15-node triangular prism element $h_i, i = 1, \ldots, 15$ are defined by [1]

$$\begin{aligned}
h_1 &= \frac{1}{2}((1-r-s)(2(1-r-s)-1)(1-t)-(1-r-s)(1-t^2))\ , \qquad (2.23) \\
h_2 &= \frac{1}{2}(r(2r-1)(1-t)-r(1-t^2))\ , \\
h_3 &= \frac{1}{2}(s(2s-1)(1-t)-s((1-t^2))\ , \\
h_4 &= \frac{1}{2}((1-r-s)(2(1-r-s)-1)(1+t)-(1-r-s)(1-t^2))\ , \\
h_5 &= \frac{1}{2}(r(2r-1)(1+t)-r(1-t^2))\ , \\
h_6 &= \frac{1}{2}(s(2s-1)(1+t)-s((1-t^2))\ , \\
h_7 &= 2(1-r-s)r(1-t) \\
h_8 &= 2rs(1-t)\ , \\
h_9 &= 2s(1-r-s)(1-ts)\ , \\
h_{10} &= 2(1-rs-s)r(1+t)\ , \\
h_{11} &= 2rs(1+t)\ , \\
h_{12} &= 2s(1-r-s)(1+t)\ , \\
h_{13} &= (1-r-s)(1-t^2)\ , \\
h_{14} &= r(1-t^2)\ , \\
h_{15} &= s(1-t^2)\ .
\end{aligned}$$

The integrals in the finite element analysis are conducted by Gauss quadrature [9], where both the positions of the sampling points and the weights are optimized. For the 20-node brick element the weights are equal to 1 and the sampling points are given in isoparametric coordinates defined in Figure 2-10 which shows their values at the planes $t = -0.774596669241483, t = 0$ and $t = 0.774596669241483$. Table 2.5 shows the complete set of values for the brick element used in the analysis.

For a 15-node triangular prismoid, the sampling points position are shown in Figure 2-11 at planes $t = -0.774596669241483, t = 0$ and $t = 0.774596669241483$. The values of the coordinates at each node were not included in order to have a better visualization of the picture. Table 2.6 shows the complete set of coordinates and weights for the prismoid element used in the analysis.

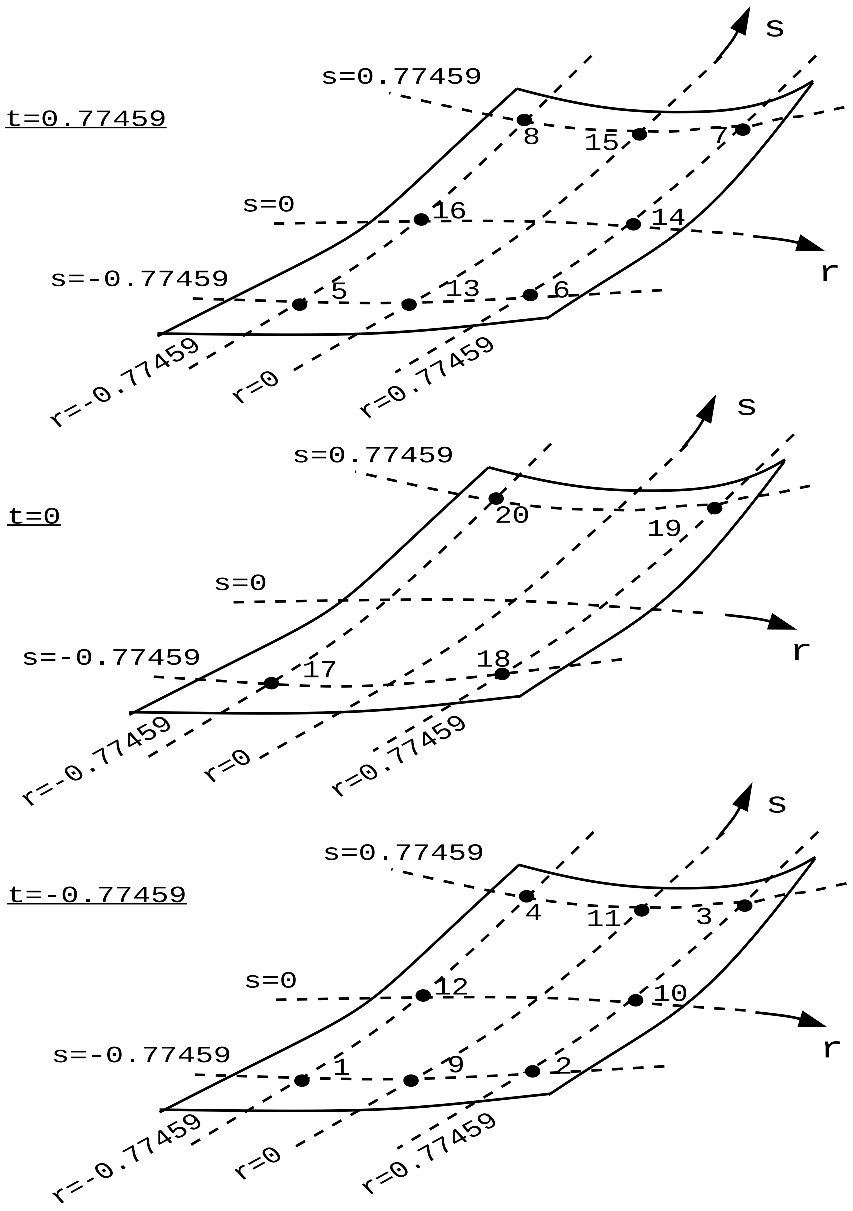


Figure 2-10: Gauss quadrature rules for a 20-node brick element

Integration order	Degrees of precision	Integration points	r-coordinates	s-coordinates	Weights
3×3	5	See Figure 2-10	r_1=-0.77459666924	s_1=-0.77459666924	$w_1 = 1.0$
			r_2= 0.77459666924	s_2=-0.77459666924	$w_2 = 1.0$
			r_3= 0.77459666924	s_3= 0.77459666924	$w_3 = 1.0$
			r_4=-0.77459666924	s_4= 0.77459666924	$w_4 = 1.0$
			r_5=-0.77459666924	s_5=-0.77459666924	$w_5 = 1.0$
			r_6= 0.77459666924	s_6=-0.77459666924	$w_6 = 1.0$
			r_7= 0.77459666924	s_7= 0.77459666924	$w_7 = 1.0$
			r_8=-0.77459666924	s_8= 0.77459666924	$w_8 = 1.0$
			r_9= 0.0	s_9=-0.77459666924	$w_9 = 1.0$
			r_{10}= 0.77459666924	s_{10}= 0.0	$w_{10} = 1.0$
			r_{11}= 0.0	s_{11}= 0.77459666924	$w_{11} = 1.0$
			r_{12}=-0.77459666924	s_{12}= 0.0	$w_{12} = 1.0$
			r_{13}= 0.0	s_{13}=-0.77459666924	$w_{13} = 1.0$
			r_{14}= 0.77459666924	s_{14}= 0.0	$w_{14} = 1.0$
			r_{15}= 0.0	s_{15}= 0.77459666924	$w_{15} = 1.0$
			r_{16}=-0.77459666924	s_{16}= 0.0	$w_{16} = 1.0$
			r_{17}=-0.77459666924	s_{17}=-0.77459666924	$w_{17} = 1.0$
			r_{18}= 0.77459666924	s_{18}=-0.77459666924	$w_{18} = 1.0$
			r_{19}= 0.77459666924	s_{19}= 0.77459666924	$w_{19} = 1.0$
			r_{20}=-0.77459666924	s_{20}= 0.77459666924	$w_{20} = 1.0$

Table 2.5: Gauss numerical integrations over quadrilateral domains

Integration order	Degrees of precision	Integration points	r-coordinates	s-coordinates	Weights
7-point	2	See Figure 2-11	r_1= 0.101286507	s_1= 0.101286507	$w_1 = 0.3333333$
			r_2= 0.797426985	s_2= 0.101286507	$w_2 = 0.3333333$
			r_3= 0.101286507	s_3= 0.797426985	$w_3 = 0.3333333$
			r_4= 0.101286507	s_4= 0.101286507	$w_4 = 0.3333333$
			r_5= 0.797426985	s_5= 0.101286507	$w_5 = 0.3333333$
			r_6= 0.101286507	s_6= 0.797426985	$w_6 = 0.3333333$
			r_7= 0.470142064	s_7= 0.059715871	$w_7 = 0.0132394$
			r_8= 0.470142064	s_8= 0.470142064	$w_8 = 0.0132394$
			r_9= 0.059715871	s_9= 0.470142064	$w_9 = 0.0132394$
			r_{10}= 0.470142064	s_{10}= 0.059715871	$w_{10} = 0.0132394$
			r_{11}= 0.470142064	s_{11}= 0.470142064	$w_{11} = 0.0132394$
			r_{12}= 0.059715871	s_{12}= 0.470142064	$w_{12} = 0.0132394$
			r_{13}= 0.101286507	s_{13}= 0.101286507	$w_{13} = 0.3333333$
			r_{14}= 0.797426985	s_{14}= 0.101286507	$w_{14} = 0.3333333$
			r_{15}= 0.101286507	s_{15}= 0.797426985	$w_{15} = 0.3333333$

Table 2.6: Gauss numerical integrations over triangular domains

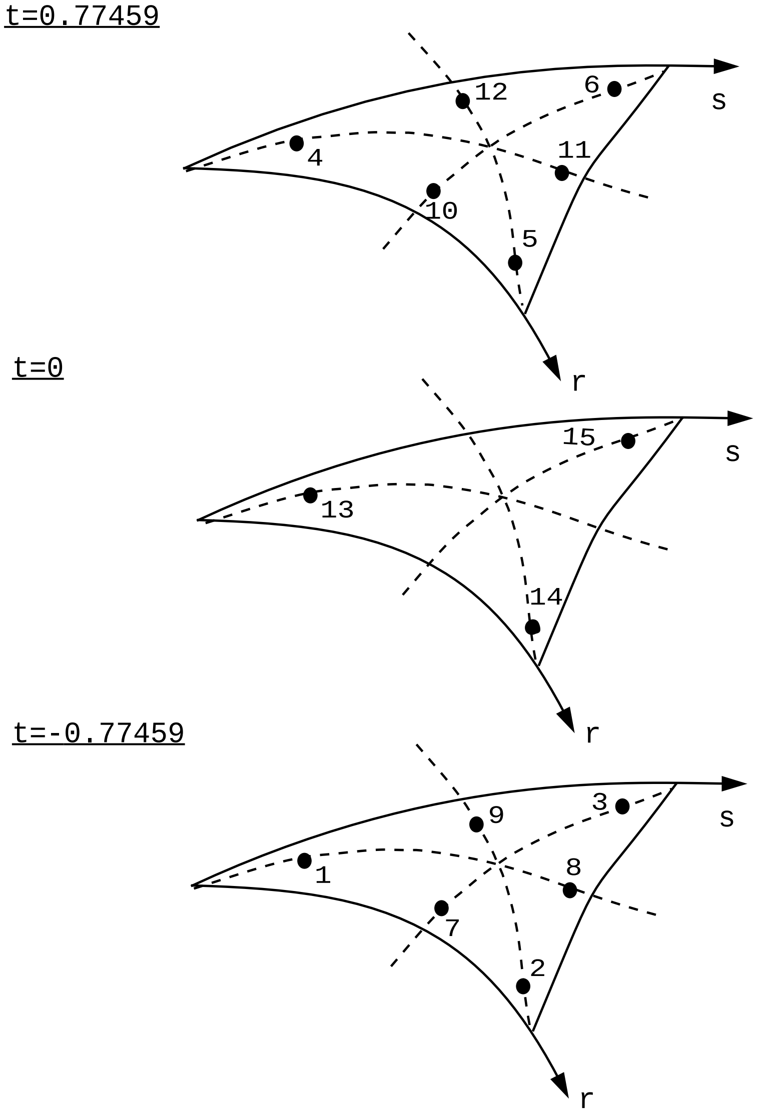


Figure 2-11: Interpolation nodes spatial position for a 15-node prism

The mechanical model starts with zero stresses for all the nodes and reads the temperature file at the corresponding time increments defined for the thermal analysis. The model applies the resulting temperature field at the nodes, varying the temperature dependent mechanical properties of the pipe inducing deformations of the pipe itself. The computational results provide the total deformation of the pipe as shown in Figure 2-12, which can also be decomposed into the x, y, z components (see Figures 2-13 to 2-15).

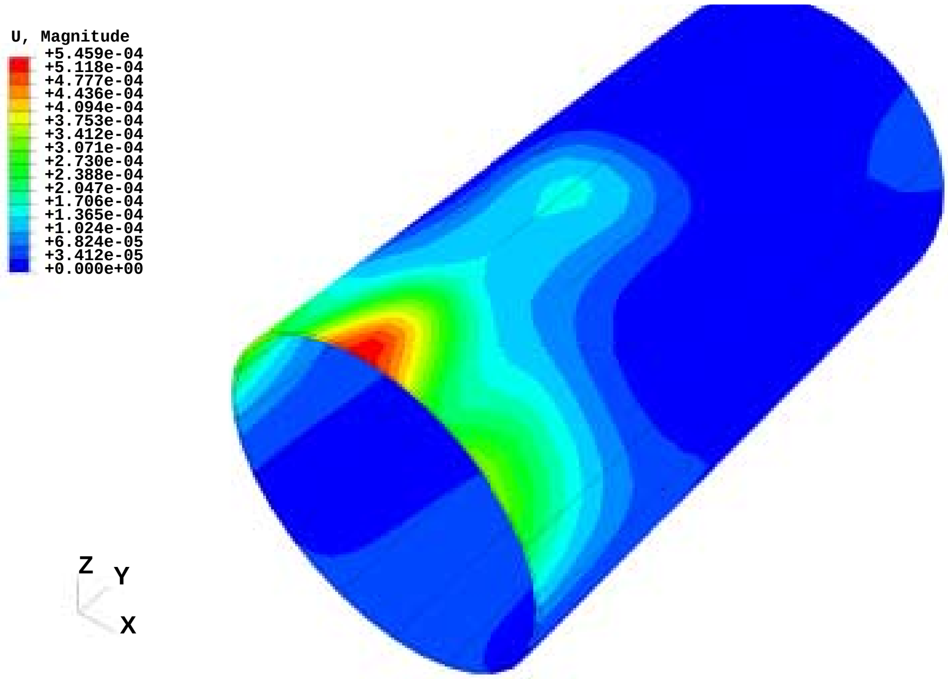


Figure 2-12: Deformation magnitude after the non-linear mechanical analysis

It can be seen that the model has distinctive deformations at three locations: at the position of application of the heat and at two positions located symmetrically distributed with respect to the heating line. The line where heat is applied along $x = 0, z = r_o$ in a positive y direction will become plastic, creating a contraction mechanism as explained in section 1.5, deforming the heated area towards the pipe center as shown in Figures 3-14 and 3-15. This produces two contiguous areas besides the heated line to deform in a radial direction away from the center of the pipe. This means that if the centerline deforms inwards, its sides will deform outwards. The effect can be seen more clearly in Figure 2-13 where there is a displacement in the x direction at both sides of the heating line. These resultant deformations are certainly important. The curvature analysis presented in Chapter 4 shows that for every deformation from the true circle to the inside or outside of the pipe, there are two almost symmetric deformation at its sides, that can be corrected taking advantage of the effect induced in the pipe under line heating. The maximum value for the deformation in the z direction using the values specified in Table 2.3 was -0.5160

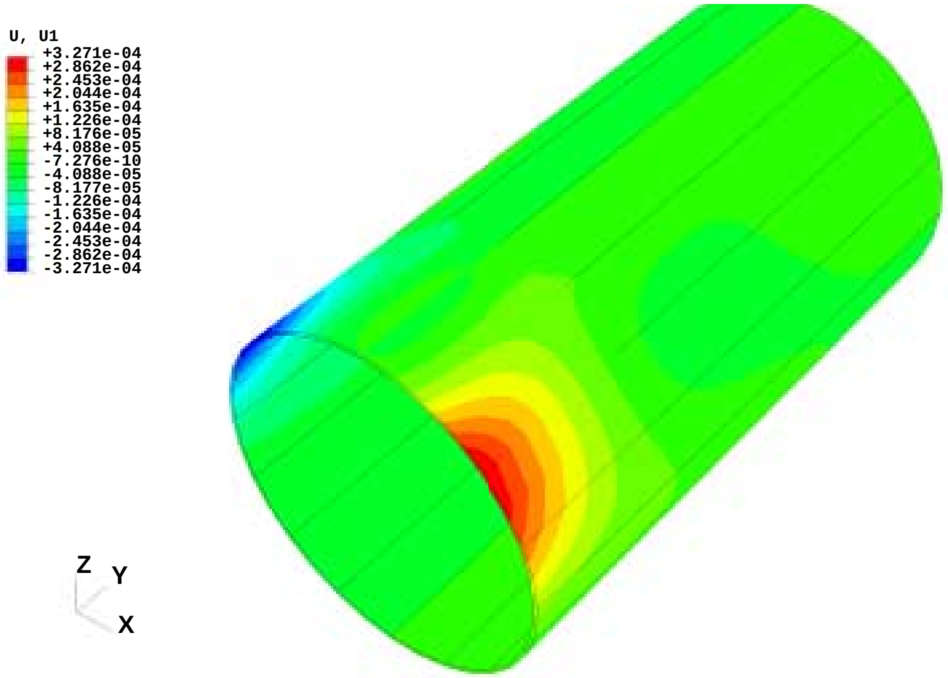


Figure 2-13: Deformation in the x direction

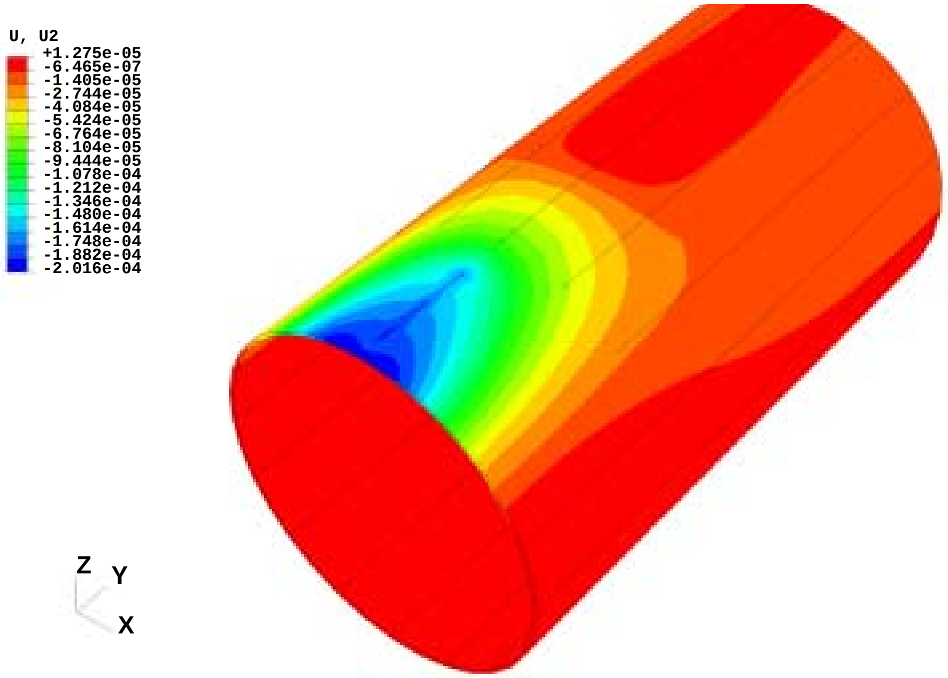


Figure 2-14: Deformation in the y direction

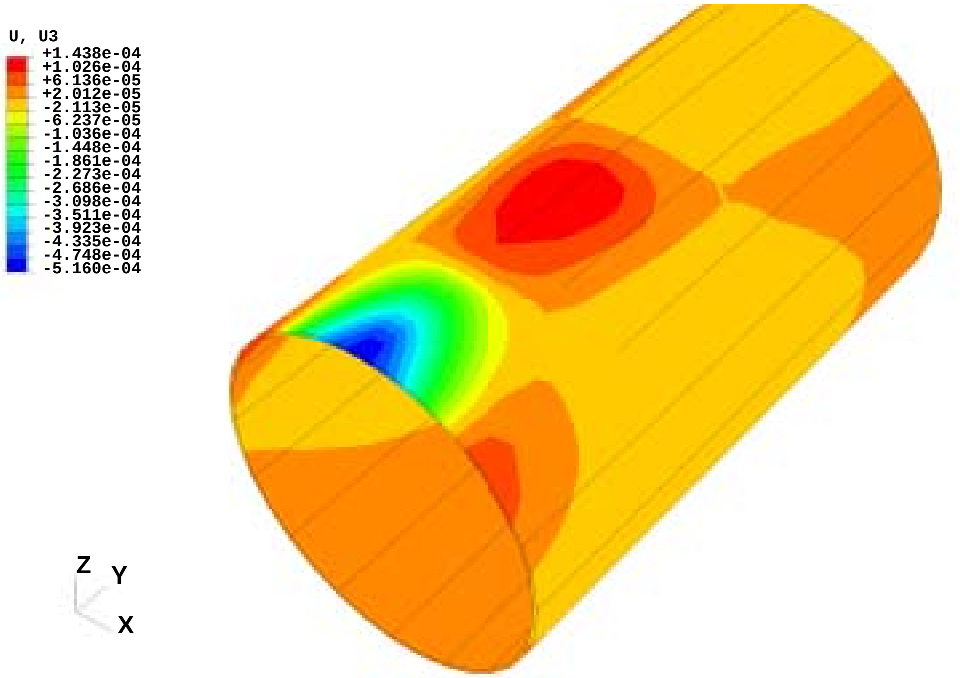


Figure 2-15: Deformation in the z direction

mm at the free end. Table 2.7 shows the results of the non-linear mechanical analysis based on the results of the heat transfer analysis under the heat conditions indicated in Table 2.4. These results are used to validate the simplified mechanical model in Chapter 3.

Case	Power input	Source speed	Spot size radius	Deformation
				at (0,0,r_o), z direction
	W	mm/s	mm	$mm \times 10^{-4}$
1	1300	1.6	12.7	-5.16
2	1000	1.6	11.8	-3.30
3	700	1.0	8.9	-3.43
4	1100	1.3	10.8	-3.59
5	800	0.7	10.8	-3.66
6	600	0.3	12.7	-4.17
7	900	1.0	11.8	-3.49
8	600	0.7	8.9	-5.35

Table 2.7: Deformation at the free end under several heat conditions

Chapter 3

Simplified model

3.1 Introduction

As mentioned in Chapter 2, the process of correction of the circularity of the ends of a pipe by line heating is a coupled nonlinear thermo-mechanical process and simulated by a full three-dimensional FEM analysis. However the required computational time for such process is typically very long, from hours to days, and is not suitable for real-time analysis. The objective of this chapter is to adapt the simplified thermo-mechanical model for the line heating process of a flat plate developed at the MIT Fabrication Laboratory [22][23] to a cylindrical body, and then compare its results with the solution obtained by the nonlinear FEM analysis. The equivalent analytic parameters predicted by the simplified model will then be used to simulate the deformation effect over a FEM model of a pipe with shell elements and with a linear application of equivalent forces and bending moments. In Sections 3.2 and 3.3 we review the simplified thermal model and simplified mechanical model developed at the MIT Fabrication Laboratory [22] [23].

3.2 Thermal model with heat loss and a distributed heat source

In the process of metal forming by line heating, residual plastic strains generated by the heating and subsequent cooling deforms a metallic body. Therefore, determination of temperature field is the prerequisite for predicting its final deformation. A method which can be applied to this problem is Rosenthal's solution of temperature distribution in a plate with a moving heat source [15].

Rosenthal makes three major assumptions which affect the solution of the temperature distribution. Rosenthal first assumes that the physical characteristics of the heated material, such as heat conductivity and specific heat, are independent of temperature. The second major assumption is that the speed of the moving heat source and the rate of heat input to the material are constant. When the heat source speed and the heat flux are considered constant, the third assumption of a quasi-stationary heat flow can be made. In a quasi-stationary heat flow state, while the temperature distribution is not constant with respect to a fixed coordinate system on the heated solid, it does remain constant with respect to a position on the moving heat source. This assumption is valid when the solid is long enough for a given heating condition for the quasi-stationary state to exist.

Rosenthal discusses the solutions of the heat equation for one-dimensional, two-dimensional, as well as three-dimensional cases. The solution of the temperature distribution of a moving point heat source with two-dimensional heat flow, as given by Rosenthal is presented in three ways [15]. The first solution is the most fundamental, where the linear source of constant strength through the thickness is applied without any surface heat losses. The second solution accounts for the effects of surface heat losses, but only for a thin plate where the temperature gradient across the thickness can be neglected. The third solution applies to a linear source of variable strength through the thickness but does not account for the surface heat losses. With the addition of a variable strength heat source, the temperature distribution varies through the thickness of the plate, effectively turning the two-dimensional heat flow solution into a three-dimensional solution [23][22]. However, none of the solutions is suitable for the line heating process, which involves a distributed heat source and heat loss, and also temperature gradient across plate thickness.

Further improvements are made in this chapter to the third solution which make the solution suitable for the line heating process. The first modification is the incorporation of the effects of surface heat losses to the solution with a source of variable strength across thickness. The second adaptation is the replacement of the single point source with a distributed source of an equivalent total heat flux [23][22]. This reflects the heat distribution during line heating to avoid melting the metal plate.

The rectangular coordinate system fixed on the solid is shown in Figure 3-1a. The heat source moves with speed v along the x-axis and the thickness is along the z-axis. For a coordinate system moving with the heat source, a quasi-stationary heat flow is observed. The distance from the point source along the x-axis is defined as $\xi = x - vt$ where t equals the time the source has been moving on the plate. The heat source-fixed coordinate system is shown in Figure 3-1b.

3.2.1 General solution of a quasi-stationary heat source

Following on Rosenthal [15], the partial differential equation of heat flow in a solid can be expressed in the rectangular coordinates (x, y, z) as

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = 2\lambda \frac{\partial T}{\partial t} \ , \tag{3.1}$$

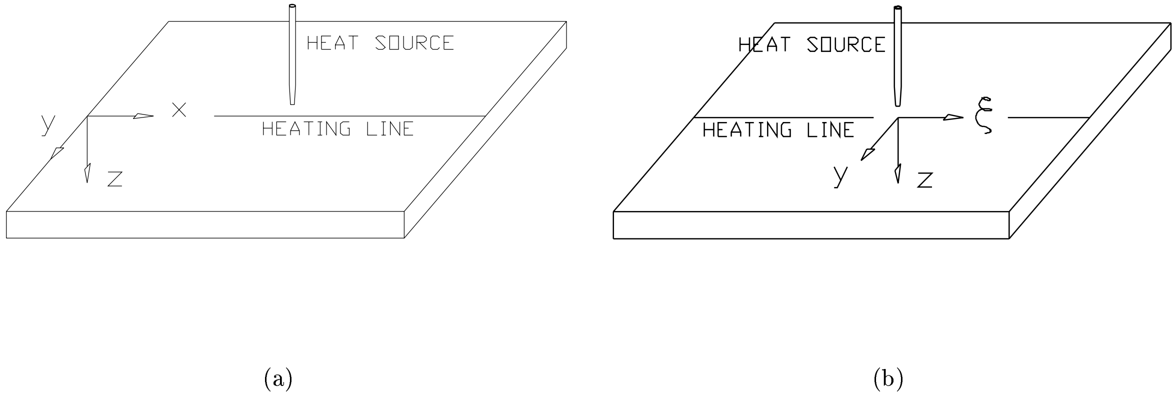


Figure 3-1: (a) Plate-fixed coordinate system (b) Heat source-fixed coordinate system.

where $\frac{1}{2\lambda}$ is the thermal diffusivity of the metal ($\frac{m^2}{s}$ in SI units) and

$$2\lambda = \frac{c_p \rho}{k} \ , \tag{3.2}$$

where k is the heat conductivity of the metal ($\frac{W}{mK}$ in SI units), c_p is the specific heat ($\frac{J}{kg-K}$ in SI units), and ρ is the density ($\frac{kg}{m^3}$ in SI units).

The coordinate transformation of $\xi = x - vt$ introduced into the heat flow partial differential equation (3.1) results in

$$\frac{\partial^2 T}{\partial \xi^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = -2\lambda v \frac{\partial T}{\partial \xi} + 2\lambda \frac{\partial T}{\partial t} \ . \tag{3.3}$$

In the moving coordinate system, the quasi-stationary heat flow condition implies that the temperature remains constant with time, $\frac{\partial T}{\partial t} = 0$, yielding

$$\frac{\partial^2 T}{\partial \xi^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = -2\lambda v \frac{\partial T}{\partial \xi} \ . \tag{3.4}$$

Equation (3.4) can be simplified by putting

$$T = T_0 + e^{-\lambda v \xi} \varphi\left(\xi, y, z\right) \ , \tag{3.5}$$

where T_0 equals the initial temperature of the plate before heating and φ is a function to be determined. When equation (3.5) is substituted into equation (3.4), the result is

$$\frac{\partial^2 \varphi}{\partial \xi^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} - \left(\lambda v\right)^2 \varphi = 0 \ . \tag{3.6}$$

3.2.2 Variable strength source with surface heat losses

Solution methodology

The solution of a moving source with a constant strength through the thickness with surface losses can not be applied directly to the line heating problem since it is the temperature gradient through the plate thickness that causes angular deformations. The effect of temperature gradient can be accounted for by allowing for a variable strength heat source [23][22].

If the method of separation of variables is applied to equation (3.6), the function φ can be expressed as a function of the depth and the radial distance from the source in the x-y plane, or

$$\varphi = Z(z)\, R(r) \ . \tag{3.7}$$

Substituting equation (3.7) into equation (3.6) results in

$$R\frac{d^2Z}{dz^2} + Z\left(\frac{d^2R}{dr^2} + \frac{1}{r}\frac{dR}{dr} - (\lambda v)^2 R\right) = 0 \ , \tag{3.8}$$

which can be rearranged to become

$$-\frac{1}{Z}\frac{d^2Z}{dz^2} = \frac{1}{R}\left(\frac{d^2R}{dr^2} + \frac{1}{r}\frac{dR}{dr} - (\lambda v)^2 R\right) = c^2 \ , \tag{3.9}$$

where c^2 is a constant to be determined by using the boundary conditions. The two ordinary differential equations which result from equation (3.9) are

$$\frac{d^2Z}{dz^2} + c^2 Z = 0 \ , \tag{3.10}$$

$$\frac{d^2R}{dr^2} + \frac{1}{r}\frac{dR}{dr} - \left((\lambda v)^2 + c^2\right) R = 0 \ . \tag{3.11}$$

The solution of equation (3.11) has the form of the modified Bessel function of the second kind and zero order, where the value of c still must be determined. The value of c is found through the solution of equation (3.10). The general solution of equation (3.10) has the form

$$Z = A' \cos(cz) + B \sin(cz) \ . \tag{3.12}$$

Taking the derivative with respect to z yields

$$\frac{dZ}{dz} = -A'c \sin(cz) + Bc \cos(cz) \ . \tag{3.13}$$

Let us assume that the value of the heat convection coefficient for the upper surface of the plate is h_U, and the value of the heat convection coefficient for the lower surface of the plate is h_L. When the heat loss due to radiation is also considered, the heat loss coefficients h_U and h_L are the equivalent convective heat loss coefficients. The boundary conditions at the surfaces of the plate with different heat loss coefficients are given below

$$k\frac{dT}{dz} = h_U\left(T - T_0\right) \qquad \text{at } z = 0\ , \tag{3.14}$$

$$k\frac{dT}{dz} = -h_L\left(T - T_0\right) \qquad \text{at } z = g\ , \tag{3.15}$$

where k is the thermal conductivity, and g is the plate thickness. After substituting Equation (3.7) into equation (3.5), and substituting equation (3.5) into equations (3.14) and (3.15), we obtain

$$ke^{-\lambda v\xi}R\left(r\right)\frac{dZ}{dz} = h_U e^{-\lambda v\xi}Z\left(z\right)R\left(r\right) \qquad \text{at } z = 0\ , \tag{3.16}$$

and

$$ke^{-\lambda v\xi}R\left(r\right)\frac{dZ}{dz} = -h_L e^{-\lambda v\xi}Z\left(z\right)R\left(r\right) \qquad \text{at } z = g\ , \tag{3.17}$$

which simplify to

$$\frac{dZ\left(0\right)}{dz} = \frac{h_U}{k}Z\left(0\right) \qquad \text{at } z = 0\ , \tag{3.18}$$

$$\frac{dZ(g)}{dz} = -\frac{h_L}{k}Z\left(g\right) \qquad \text{at } z = g\ . \tag{3.19}$$

Equation (3.12) and its derivative in Equation (3.13) are evaluated at points on the upper and lower surfaces of the plate, and substituted into relations (3.18) and (3.19), respectively. On the upper surface of the plate, $z = 0$, we have

$$Bc = \frac{h_U}{k}A'\ . \tag{3.20}$$

On the lower surface of the plate, $z = g$, we have

$$-A'c\sin\left(cg\right) + Bc\cos\left(cg\right) = -\frac{h_L}{k}\left(A'\cos\left(cg\right) + B\sin\left(cg\right)\right)\ . \tag{3.21}$$

When the result in relation (3.20) is substituted into (3.21) and simplified, the result is [23][22]

$$\tan\left(cg\right) = \frac{-\left(h_U + h_L\right)kc}{h_U h_L - k^2c^2}\ . \tag{3.22}$$

In the above equation, the $\tan(cg)$ is a periodic function, so multiple roots exist and these roots can be determined numerically. Because of the rapid decay of the right hand term in equation (3.22), the solution of the roots rapidly approaches the value of $c_n = \frac{n\pi}{g}$, which is the solution for the case when there is no surface heat loss, i.e., $\tan\left(cg\right) = 0$. It should be noted that, while $c = 0$ satisfies Equation (3.22), the root is a trivial solution and the root corresponding to A_0 term in the Fourier series representation of the heat flux distribution is the first non-zero root of c.

With the values of c_n solved for numerically, the solutions of equations (3.10) and (3.11) expressed in terms of A'_n become

$$Z_n = A'_n \left[\cos(c_n z) + \frac{h_U}{c_n k} \sin(c_n z) \right] \ , \tag{3.23}$$

$$R_n = K_0 \left[\sqrt{(\lambda v)^2 + c_n^2} \ r \right] \ , \tag{3.24}$$

where K_0 is the modified Bessel function of the second kind and zero order [23][22]. For convenience, a portion of the argument of the Bessel function in equation (3.24) is represented by Λ_n so that

$$\Lambda_n = \sqrt{(\lambda v)^2 + c_n^2} \ . \tag{3.25}$$

When the expression for Λ_n in equation (3.25) and the solution of R_n in equation (3.24) are substituted into equation (3.7), the result is

$$\varphi_n = A'_n \left[\cos(c_n z) + \frac{h_U}{c_n k} \sin(c_n z) \right] K_0 \left[\Lambda_n r \right] \ . \tag{3.26}$$

The substitution of the result for φ into the assumed form of the temperature at a point given by Equation (3.5) yields

$$T - T_0 = e^{-\lambda v \xi} \sum_{n=0}^{N} A'_n \left[\cos(c_n z) + \frac{h_U}{c_n k} \sin(c_n z) \right] K_0 \left[\Lambda_n r \right] \ . \tag{3.27}$$

Now we are left with the determination of A'_n from the following boundary condition:

$$-\frac{\partial T}{\partial r} 2\pi r k \longrightarrow q'(z) \qquad \text{as } r \longrightarrow 0 \ , \tag{3.28}$$

where $q'(z)$ is the heat flux per unit thickness at z ($\frac{W}{m}$ in the SI units). We assume the heat flux distribution across thickness follows a triangular distribution which decreases linearly from a maximum magnitude $q'(0)$ at the top surface to $q'(\epsilon) = 0$ at a depth of ϵ, where ϵ is a fraction of the thickness [23][22]. The area of the heat distribution

$$q = \frac{1}{2} q'(0) \epsilon \ , \tag{3.29}$$

is the total heat input per unit time (W in the SI units), which is kept as a constant. Therefore, $q'(0)$ increases when ϵ decreases. For very small ϵ compared with thickness g, the heat flux becomes close to the case of surface flux [3]. This triangular heat flux distribution can be represented by a Fourier series of the assumed form

$$q'(z) = q'(0) \sum_{n=0}^{N} A_n \left[\cos(c_n z) + \frac{h_U}{c_n k} \sin(c_n z) \right] \ , \tag{3.30}$$

where a trigonometric series involving c_n is used for expedience of the analytical solution.

After substituting equation (3.27) and equation (3.30) into the boundary condition (3.28), We obtain the relation between the values of the A'_n and A_n:

$$A'_n = \frac{q'(0)}{2\pi k} A_n \ . \tag{3.31}$$

The relation between the coefficients is substituted into equation (3.27) to get an expression for the temperature increase at a point in terms of the A_n coefficients of the heat flux distribution. The solution for the temperature change at a point in a plate heated with a variable strength source with different values of convective surface heat transfer coefficients for each surface is therefore [23][22]

$$T - T_0 = \frac{q'(0)}{2\pi k} e^{-\lambda v \xi} \sum_{n=0}^{N} A_n \left[\cos(c_n z) + \frac{h_U}{c_n k} \sin(c_n z)\right] K_0 \left[\Lambda_n r\right] \ . \tag{3.32}$$

Determination of A_n coefficients

When heat losses are accounted for, the solution for c_n results in a non-orthogonal behavior of the cosine and sine terms in the summations. The integration method used for solving the values of the coefficients can therefore not be applied and the coefficients must be determined with direct use of the heat flux distribution.

The values of the A_n coefficients in the summations in the expressions for the heat flux distribution and the temperature change are found by solving a linear system of equations created by substitution of values of z into the Fourier series representation of the heat flux distribution given by [23][22]

$$q'(z) = q'(0) \sum_{n=0}^{N} A_n \left[\cos(c_n z) + \frac{h_U}{c_n k} \sin(c_n z)\right] \ . \tag{3.33}$$

When the total heat flux to the plate is kept constant, and triangular heat flux distribution is used for the method of direct substitution to solve for coefficients, the value of ϵ determines not only the value of $q'(0)$, but also the minimum number of terms required to accurately represent the heat flux distribution, and therefore the minimum number of equations required in the linear system [3]. At least two points of the heat flux distribution between $z = 0$ and $z = \epsilon$ must be represented. When ϵ is chosen so that (g/ϵ) is an integer N_0, this requirement is satisfied when $N+1$ points are used to define the distribution, where N is a multiple of N_0. Since $N+1$ points are used to represent the heat flux distribution accurately, $N+1$ unknown A_n coefficients are in each equation in the linear system and, therefore, $N+1$ equations are required for a determinate system. In our application, we chose $\epsilon = 0.09$, and $N = 200$.

Each of the $N+1$ equations of the system is evaluated at a discrete value of z, which when evenly distributed across the thickness is given by

$$z_n = \left(\frac{g}{N}\right) n \qquad\qquad \text{for } n = 0, 1, ...N \ , \tag{3.34}$$

and each equation will therefore have the form

$$A_0\gamma_{m,0} + A_1\gamma_{m,1} + \cdots + A_i\gamma_{m,i} + \cdots + A_N\gamma_{m,N} = q'(z_m) \;\; m = 0, ..., N \ , \tag{3.35}$$

where the value of $\gamma_{i,m}$ is given by

$$\gamma_{m,i} = \cos(c_i z_m) + \frac{h_U}{c_i k}\sin(c_i z_m) \ . \tag{3.36}$$

With the values of A_n in the linear system (3.35) solved, they can be applied with their corresponding values of c_n to the summations required to evaluate the heat flux distribution and the temperature distribution within the plate [23][22].

3.2.3 Continuous heat source

In the previous sections, a point source is used in deriving the temperature field. The use of point source has the following disadvantages: (1) it does not reflect the actual practice of line heating when the heat source is diffused; (2) the temperature field has a singular point under the point source where temperature goes to infinity. A continuous heat source representation eliminates these disadvantages.

The continuous heat source is defined by the assumed truncated Gaussian distribution of the total heat flux [24], which is given by

$$q'''(r) = q_0'''e^{-cr^2} \ , \tag{3.37}$$

where q''' (W/m^3 in SI units) is the value of the heat flux per unit thickness and unit area on top surface at a radial distance r in the ξ-y plane from the maximum value of the heat flux q_0''' (W/m^3 in SI units). The value of the constant c is determined by the radius r_0 of the heat flux region, where the value of the heat flux is assumed to be five percent of the maximum value. The value of c is given as

$$c = -\frac{\ln(0.05)}{r_0^2} \ , \tag{3.38}$$

and the maximum heat flux is therefore given as

$$q_0''' = \frac{q'(0)c}{0.95\pi} \ , \tag{3.39}$$

where $q'(0)$ is the total heat flux per unit thickness at top surface. Equation (3.39) is obtained by equalizing the total flux per unit thickness inside the circle of radius r_0 to be

$q'(0)$ [23][22].

A differential element on the plate within the heat flux region at a distance r from the maximum value of the heat flux has an area $dA = rdrd\theta$. The integral of the products of the differential areas and the corresponding value of the heat flux at the radial distance r over the heat flux region is the total value of the heat flux per unit thickness. On the heated surface of the plate, a differential amount of heat flux is given as

$$dq' = q'''(r)rdrd\theta \tag{3.40}$$

With reference to equation (3.32), the expression of the temperature change at a point, with the application of equation (3.40), yields [23][22]

$$\begin{aligned} T - T_0 = \int_0^{r_0} \int_0^{2\pi} \quad & \frac{q'''(r)}{2\pi k} e^{-\lambda v(\xi - r\cos\theta)} \sum_{n=0}^{N} A_n \left[\cos\left(c_n z\right) + \frac{h_U}{c_n k} \sin\left(c_n z\right) \right] \\ & K_0 \left[\Lambda_n \sqrt{(\xi - r\cos\theta)^2 + (y - r\sin\theta)^2} \right] rdrd\theta \ , \end{aligned} \tag{3.41}$$

which can be computed numerically using Gaussian quadrature [7].

Elimination of singularities

When we evaluate equation (3.41) numerically using Gaussian quadrature, for all points outside the heat flux region, the modified Bessel function has no singularity. However, for points inside the heat flux region and are close to the Gaussian integration points, singularity in modified Bessel function results in very high temperature there.

The singularities in temperature computation can be removed by using a coordinate transformation [23][22]. We perform numerical integration using the polar coordinate system (R, γ) centering at the point of interest instead of the polar coordinate system (r, θ) whose origin is at the center of the heat flux region, see Figure 3-2.

Let O be the center of the heat flux region, and B be the point whose temperature is to be computed. Then we have

$$|OD| = |\xi|, \ \ |BD| = |y| \ , \tag{3.42}$$

$$|OB| = \sqrt{|OD|^2 + |BD|^2} = \sqrt{\xi^2 + y^2} \ , \tag{3.43}$$

and

$$\theta_0 = \begin{cases} \tan^{-1} \frac{y}{\xi} & \text{if } \xi > 0 \\ \pi + \tan^{-1} \frac{y}{\xi} & \text{if } \xi < 0 \\ \frac{\pi}{2} & \text{if } \xi = 0 \text{ and } y > 0 \\ -\frac{\pi}{2} & \text{if } \xi = 0 \text{ and } y < 0 \end{cases} \tag{3.44}$$

We assume the radius of the heat flux region is r_0, and consider the intersection of a ray of

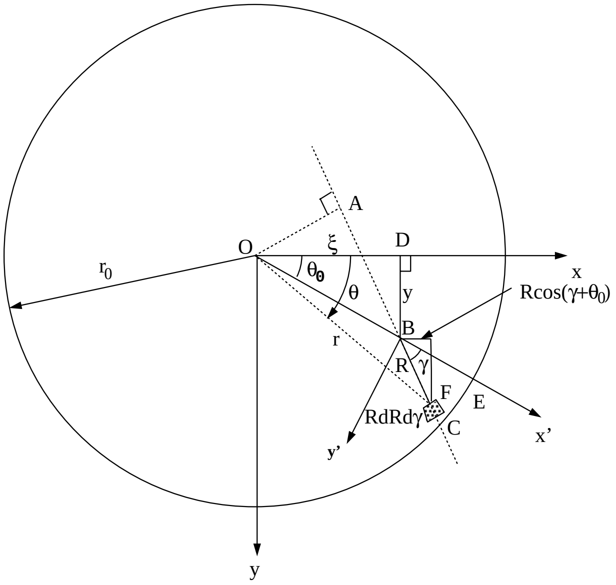


Figure 3-2: Coordinate transformation

an angle $\gamma \in (0, \frac{\pi}{2})$ with the boundary of the heat flux region. We have

$$
\begin{aligned}
|AB| &= |OB||\cos\gamma| = \sqrt{\xi^2+y^2}|\cos\gamma| \ , && (3.45)\\
|OA| &= |OB||\sin\gamma| = \sqrt{\xi^2+y^2}|\sin\gamma| \ , && (3.46)\\
|AC| &= \sqrt{r_0^2 - |OA|^2} = \sqrt{r_0^2 - (\xi^2+y^2)\sin^2\gamma} && (3.47)
\end{aligned}
$$

Therefore,

$$
R_0(\gamma) = |BC| = |AC| - |AB| = \sqrt{r_0^2 - (\xi^2+y^2)\sin^2\gamma} - \sqrt{\xi^2+y^2}\cos\gamma \quad (3.48)
$$

It can be verified that equation (3.48) also apply when $\gamma \notin (0, \frac{\pi}{2})$.

The temperature change at point B due to the differential heat input at the differential area $dA = RdRd\gamma$ located at point F is

$$
\begin{aligned}
dT &= \frac{q'''(r)}{2\pi k} e^{-\lambda v R\cos(\gamma+\theta_0)} \sum_{n=0}^{N} A_n \left[\cos(c_n z) + \frac{h_U}{c_n k}\sin(c_n z)\right] \\
&\quad K_0\left[\Lambda_n R\right] RdRd\gamma \ , && (3.49)
\end{aligned}
$$

where R is the distance between points F and B, and

$$
\begin{aligned}
r &= |OF| = \sqrt{|OB|^2 + |BF|^2 + 2|OB||BF|\cos\gamma} \ , \\
&= \sqrt{\xi^2 + y^2 + R^2 + 2R\sqrt{\xi^2+y^2}\cos\gamma} && (3.50)
\end{aligned}
$$

$$
q'''(r) = q_0 e^{-cr^2} = q_0 e^{-c(\xi^2+y^2+R^2+2R\sqrt{\xi^2+y^2}\cos\gamma)} \ . \quad (3.51)
$$

Therefore, after integrating the temperature change due to all the differential heat flux within the circle in Figure 3-2, we obtain the temperature increase at point B located inside the heat flux region as [23][22]

$$\begin{aligned} T - T_0 &= \int_0^{2\pi} \int_0^{\sqrt{r_0^2-(\xi^2+y^2)\sin^2\gamma}-\sqrt{\xi^2+y^2}\cos\gamma} \frac{q_0 e^{-c(\xi^2+y^2+R^2+2R\sqrt{\xi^2+y^2}\cos\gamma)}}{2\pi k} \\ & e^{-\lambda v R\cos(\gamma+\theta_0)} \sum_{n=0}^{N} A_n \left[\cos(c_n z) + \frac{h_U}{c_n k}\sin(c_n z)\right] K_0[\Lambda_n R] R dR d\gamma \qquad (3.52) \end{aligned}$$

Since $K_0\left[\Lambda_n R\right] R$ is finite when $R \to 0$, equation (3.52) can be evaluated numerically by using Gaussian quadrature with no singularity.

Therefore, when equation (3.41) is used for computation of the temperature at points located outside the heat flux region, and equation (3.52) is used for computation of the temperature at points located inside the heat flux region, the temperature field can be evaluated without any singularity [23][22].

3.2.4 Discussion of the thermal model

From the derivation of the thermal model, we can see that the effects of heat loss on temperature field expressed in equation (3.27) lie in (1) difference between the solution c of equation (3.22) and $\frac{n\pi}{g}$, which is the solution for the case of no surface heat loss, and (2) the term with h_U in equation (3.27). Equation (3.22) can be rewritten as [23][22]

$$\tan(cg) = \frac{\frac{h_U+h_L}{kc}}{1-\frac{h_U h_L}{k^2c^2}} \,. \qquad (3.53)$$

Therefore, as a first approximation, the difference between c_n and $\frac{n\pi}{g}$ depends on

$$\frac{h_U+h_L}{kc} \approx \frac{(h_U+h_L)g}{n\pi k} \,, \qquad (3.54)$$

$$\frac{h_U h_L}{k^2c^2} \approx \frac{h_U h_L g^2}{n^2\pi^2k^2} \,. \qquad (3.55)$$

The term with h_U in equation (3.27) has the same order of magnitude as the right hand side of equation (3.54).

It should be noted that when heat loss due to radiation is taken into account, h_U, h_L become the equivalent heat loss coefficients. For the forming process of a mild steel plate of the size 2.35 m × 1.4 m × 0.009 m equivalent to the pipe developed surface, under an average temperature of 500^oC, the equivalent heat loss coefficient due to radiation $h = 5.67 \times 10^{-8}\varepsilon(T_s^3 - T_a^3) = 5.74044 W m^{-2} K^{-1}$ is of the same order of magnitude as the convective heat loss coefficients. Since

$$\frac{(h_U+h_L)g}{\pi k} = 7.41645 \times 10^{-4} \,, \qquad (3.56)$$

$$\frac{h_U h_L g^2}{\pi^2 k^2} = 1.239 \times 10^{-7}, \tag{3.57}$$

these terms are negligible, and the temperature field with heat loss does not differ much from that without heat loss.

At high temperature, the equivalent heat loss coefficient due to radiation increases rapidly. For example, $h = 45.9 W m^{-2} K^{-1}$ at $T_s = 1000^o C$. Therefore, the effects of heat loss on temperature field become more important at high temperatures. Tungsten typically has a service temperature around $2000^o C$. If a tungsten plate is bent at high temperature, then heat loss will be significant.

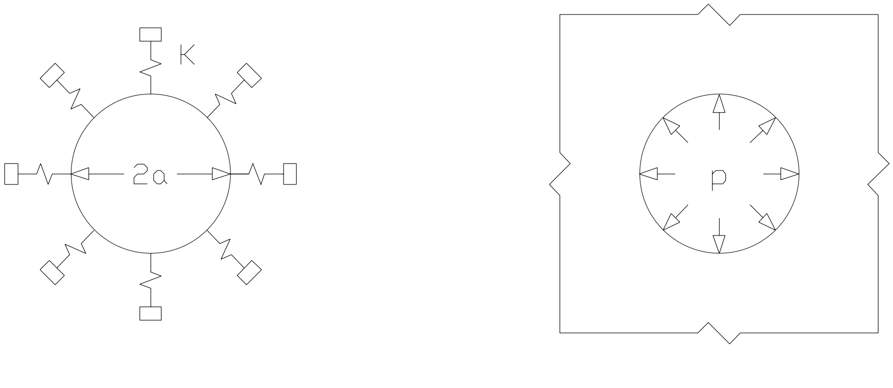
3.3 Simplified mechanical model

3.3.1 Assumptions

A simplified mechanical model was first proposed by Jang et al. [13] with the following assumptions:

(1) The elasto-plastic process is concentrated on a circular disk under the heat source and that the plate is infinite in two dimensions. Also, the diameter of the disk changes through the thickness of the plate, resulting in elliptical isothermal boundaries in the cross-section of the plate perpendicular to the direction of the heating line.

(2) The thermal elasto-plastic process occurs within the small, circular region axisymmetrically and the remaining elastic region resists the expansion and contraction of the circular plastic region. The resistance of the elastic region on the plastic zone is modeled as a set of surrounding springs which is illustrated in Figure 3-3(a). The spring constant K is evaluated by deriving the radial displacement of a circular hole of an infinite plate subjected to an inner uniform pressure p, as shown in Figure 3-3(b).



(a)

(b)

Figure 3-3: (a) Model of plastic region (b) Model of elastic region.

(3) The temperature increasing process, which results in thermal expansion, can be

thought of as a mechanical plastic loading in the plastic region and the temperature decreasing process is a mechanical plastic unloading. This plastic unloading process is the cause of the residual strains which ultimately result in the plate angular deformation.

(4) The final angular deformation results from the integration of deformation due to the residual strain in each disk. The bounds of integration are defined by the region of inherent strain produced during heating, which is assumed to have an elliptical distribution in the plane of the plate perpendicular to the heating path. The assumed elliptical distribution is based on experimental results and Rosenthal's solution [16][15].

3.3.2 Plastic strain

Under the previous assumptions, during the temperature increasing process, the central circular area goes to plastic. After the plate cools down, the residual plastic strain in the radial direction of the circular plastic region is [13]

$$\epsilon^*_{rr} = -\alpha T_c + \sigma_{yD}\left(\frac{1}{aK} + \frac{1-\nu_D}{E_D}\right) \ , \tag{3.58}$$

where α is the thermal coefficient of the disk; T_c is the critical temperature of the plate material; σ_{yD} is the yield stress of the disk; a is the radius of the disk; ν_D is the Poisson's ratio of the disk; E_D is the Young's modulus of the disk, and

$$K = \frac{p}{u_p} = \frac{E}{a\,(1+\nu)} \ , \tag{3.59}$$

is the spring constant of the surrounding area. In equation (3.59), u_p is the increment of the radius of the circle in Figure 3-3(b) due to pressure p; E and ν are the Young's modulus and Poisson's ratio of the elastic region respectively. It can be seen from equation (3.58) that Jang et al. [13] treats the temperatures inside the plastic region to be all T_c, which tends to underpredict the plastic strain. We make a reasonable modification. We use the average of the critical temperature and the maximum temperature inside the plastic region to compute the residual strain, i.e.,

$$\epsilon^*_{rr} = -\alpha T_a + \sigma_{yD}\left(\frac{1}{aK} + \frac{1-\nu_D}{E_D}\right) \ , \tag{3.60}$$

where $T_a = (T_c + T_{max})/2$, and T_{max} is the maximum temperature inside the plastic region. T_a is an approximation of the average temperature inside the plastic region.

3.3.3 Inherent strain zone dimensions

The inherent strain zone is defined as the maximum region where peak temperature is equal to or greater than the critical temperature where material strength becomes minimal and is dependent on the heating conditions.

On the plane perpendicular to the direction of the heat source, Jang's model assumes a half-elliptical isothermal, and therefore, a plastic region, as illustrated in Figure 3-4 [13]. This idealization is based on temperature distributions determined analytically by Rosenthal's solution for an infinite plate with finite thickness and a point heat source moving with constant speed without any heat loss [15].

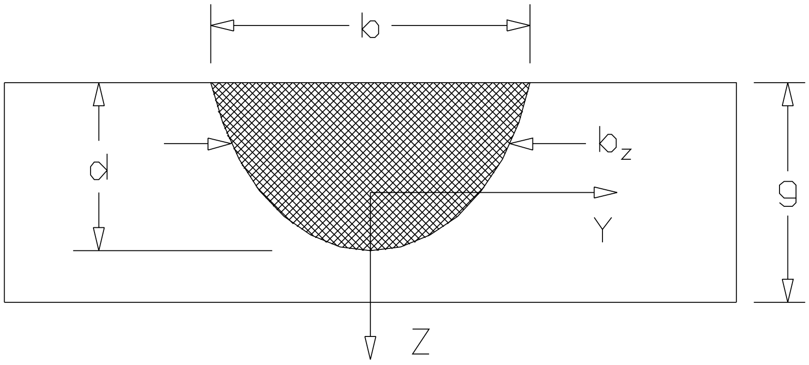


Figure 3-4: Assumed elliptical distribution of critical isothermal region and corresponding dimensions. Adapted from [13]

The isothermal region relevant to the calculation of the deformation due to heating is the region which is bounded by the critical temperature and therefore bounds the plastic region. The dimensions of the inherent strain zone are set by the maximum breadth b, and depth d, of the region as illustrated in Figure 3-4. The assumed elliptical distribution is given by

$$\frac{Y^2}{\left(\frac{b}{2}\right)^2} + \frac{\left(Z + \frac{g}{2}\right)^2}{d^2} = 1 \ , \tag{3.61}$$

where the X-Y plane is defined as the mid-plane of the plate with the Z-axis directed downward at the middle of the breadth of the inherent strain zone. The thickness of the plate is denoted by the variable g.

The width of the inherent strain zone at any depth is required in the solution of the angular deformation due to heating and is found by rearranging equation (3.61) to yield

$$b_Z = b\sqrt{1 - \frac{1}{d^2}\left(Z + \frac{g}{2}\right)^2} \ , \tag{3.62}$$

where b_Z is the width of inherent strain zone at any location through the thickness.

As mentioned previously, the dimensions and shape of the inherent strain zone are functions of the heating conditions, such as heat flux and heating source speed. The relations between inherent strain zone size and heating parameters used by Jang et al. is through an assumed linear relationship by constants determined in welding experiments [13] which is hard to verify. A more reasonable method for determining the size of the inherent strain is to use the information of isothermal lines computed from the thermal model presented in Section 3.2.

The maximum breadth and depth of the inherent strain zone are determined numerically by successively integrating the initial value problem for a system of coupled non-linear differential equations using a fourth-order Runge-Kutta method. Since the inherent strain zone is defined as the region which had reached a critical temperature, solving for the maximum dimensions of an isothermal region of the critical temperature is equivalent to solving for the maximum dimensions of the inherent strain zone [23][22].

3.3.4 Maximum breadth

Isothermal contour lines on the ξ-y plane satisfy the relation

$$T\left(\xi, y\right) = constant \ , \tag{3.63}$$

where $T\left(\xi, y\right)$ is the temperature given by equation (3.41) or equation (3.52) at a fixed value of z. The maximum breadth of the inherent strain zone occurs on the upper surface of the plate, where $z = 0$. The isotherm in the ξ-y plane on the plate can be expressed as a parametric curve defined by

$$\mathbf{r}(s) = \mathbf{r}\left(\xi(s), y(s)\right) \ . \tag{3.64}$$

Differentiating the expression for temperature along an isotherm given by expression (3.63) with respect to arc length s along the contour line yields [23][22]

$$\frac{\partial T}{\partial \xi}\frac{d\xi}{ds} + \frac{\partial T}{\partial y}\frac{dy}{ds} = 0 \ , \tag{3.65}$$

where the derivatives of the positions with respect to the arc length of the isotherm together give the direction of the contour line. The solutions of equation (3.65) are

$$\frac{d\xi}{ds} = \alpha \frac{\partial T}{\partial y} \ , \tag{3.66}$$

$$\frac{dy}{ds} = -\alpha \frac{\partial T}{\partial \xi} \ , \tag{3.67}$$

α is an arbitrary, non-zero constant which is selected to satisfy the arc length parameterization given by

$$ds^2 = d\xi^2 + dy^2 = \alpha^2 \left[\left(\frac{\partial T}{\partial \xi}\right)^2 + \left(\frac{\partial T}{\partial y}\right)^2 \right] ds^2 \ , \tag{3.68}$$

which results in

$$\alpha = \pm \frac{1}{\sqrt{\left(\frac{\partial T}{\partial \xi}\right)^2 + \left(\frac{\partial T}{\partial y}\right)^2}} \ . \tag{3.69}$$

The substitution of the solution for the value of α, using the positive solution, into equations (3.66) and (3.67) yields

$$\frac{d\xi}{ds} = \frac{1}{\sqrt{\left(\frac{\partial T}{\partial \xi}\right)^2 + \left(\frac{\partial T}{\partial y}\right)^2}} \frac{\partial T}{\partial y} \ , \tag{3.70}$$

$$\frac{dy}{ds} = \frac{-1}{\sqrt{\left(\frac{\partial T}{\partial \xi}\right)^2 + \left(\frac{\partial T}{\partial y}\right)^2}} \frac{\partial T}{\partial \xi} \ , \tag{3.71}$$

where $\frac{\partial T}{\partial \xi}$ and $\frac{\partial T}{\partial y}$ on the upper surface of the plate where $z = 0$ can be obtained by taking derivatives of equation (3.41) or equation (3.52).

Points on an isothermal contour line are computed successively by integrating equations (3.70) and (3.71) using the Runge-Kutta method [7]. The initial value required for the integration process is taken to be the point on the isotherm of the critical temperature which lies along the ξ-axis, or $y = 0$. This point is found by the method of bisection. Using the initial point on the isotherm, the Runge-Kutta method computes the position of another point along the isotherm, where the distance of the iterated point from the initial point is controlled by the value of an incremental step size, h [23][22].

3.3.5 Maximum depth

The maximum depth of the inherent strain zone is found using the same method as that used to find the maximum breadth, except the isothermal contour line lies in the ξ-z plane and all expressions are in terms of ξ and z [23][22], i.e. we solve the equation

$$T(\xi, z) = constant \ , \tag{3.72}$$

by using the fourth order Runge-Kutta method [7].

3.3.6 Maximum depth in an overheated condition

The depth of the critical isotherm will be the thickness of the plate if the heat input is high enough. For this case, in order to characterize the elliptical isotherm of the critical temperature in Figure 3-4 and given by equation (3.62) for the estimation of plate deflection, the breadth of the critical isotherm on the bottom surface needs to be evaluated. The breadth at $z = g$ can be solved using the same expressions for the Runge-Kutta method used to determine the breadth on the upper surface of the plate, except with $z = g$ instead of $z = 0$ in the equation of temperature field.

3.3.7 Angular deformation

The assumption made in the simplified mechanical model presented by Jang et al. [13] that a unit strip of plate behaves like a beam is still applicable when values of the maximum

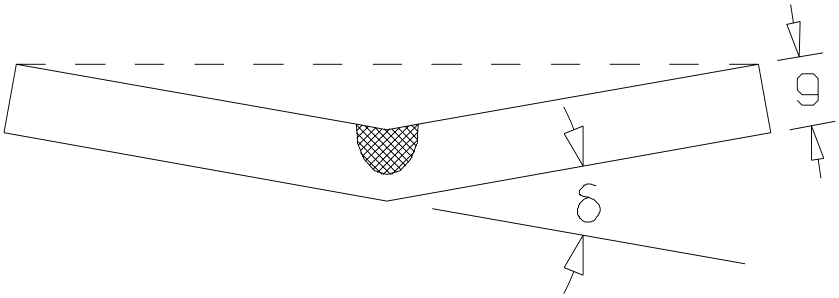


Figure 3-5: Angular deformation δ in the y-z plane. Adapted from [13]

breadth and depth of the inherent strain zone are used directly in the estimation of plate deformation.

It can be shown that the angular deformation in Figure (3-5) can be expressed as [13]

$$\delta = \frac{-12\left(1-\nu^2\right)\epsilon_{rr}^*}{g^3}\int_{\alpha}^{\beta} b_Z Z dZ \ . \tag{3.73}$$

After substituting Equation (3.62) into Equation (3.73), the actual value of the angular deformation is obtained by simply evaluating the integral in equation (3.73). Depending on the heating conditions however, the solution of the integral can be one of two forms.

With a low heat input, the depth of the inherent strain zone is less than the thickness of the plate and the bounds of integration are set by the maximum depth of the inherent strain zone. The limits of integration are from $\alpha = -\frac{g}{2}$ to $\beta = d - \frac{g}{2}$, using a coordinate system shown in Figure 3-4. With these bounds on the integral, the solution becomes

$$\delta = \frac{-2\left(1-\nu^2\right)}{g^3}\epsilon_{rr}^* bd\left[2d - \frac{3\pi}{4}g\right] \ . \tag{3.74}$$

At a certain value of heat input, the depth of the inherent strain zone equals the thickness of the plate. For this condition, and all conditions with higher values of heat input where the depth of the assumed elliptical inherent strain zone projects beyond the plate thickness, the limits of integration for Equation (3.74) become $\alpha = -\frac{g}{2}$ to $\beta = \frac{g}{2}$, or across the entire thickness of the plate and the solution becomes [23][22]

$$\delta = \frac{-2\left(1-\nu^2\right)}{g^3}\epsilon_{rr}^* bd\left[-2d\left(\left(1-\left(\frac{g}{d}\right)^2\right)^{\frac{3}{2}}-1\right)-\frac{3g}{2}\left(\frac{g}{d}\sqrt{1-\left(\frac{g}{d}\right)^2}+\sin^{-1}\frac{g}{d}\right)\right] \ . \tag{3.75}$$

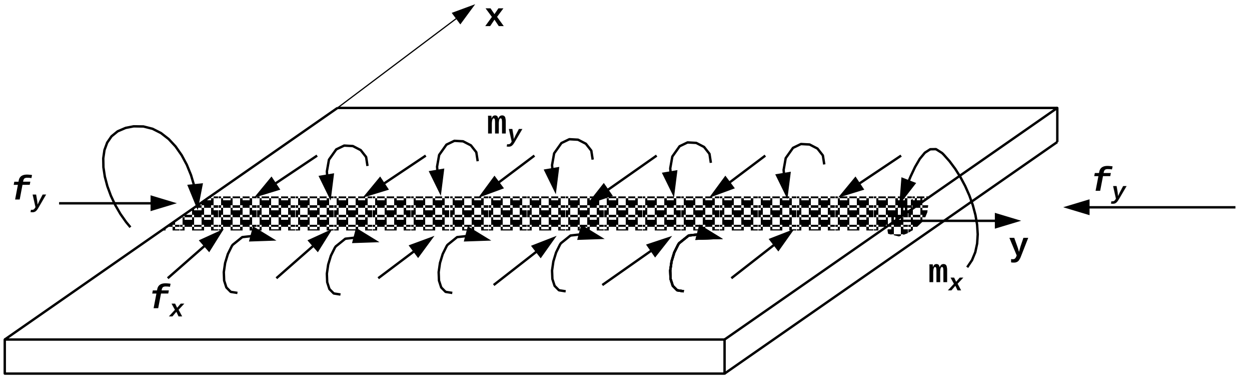


Figure 3-6: Shrinkage forces and moments due to line heating. Adapted from [13].

3.3.8 Shrinkage forces

The strains which result in the angular deformation of a plate from line heating can also be expressed by shrinkage forces and bending moments shown in Figure 3-6 [13]. These forces and moments are due to shrinkage of the inherent strain zone upon cooling and can be determined from integration of inherent strains. Since the inherent strain zone is assumed to have an elliptical distribution, the effects of the strains are represented by both a transverse moment and shrinkage force, which act perpendicular to the heating line. Similarly, the strains also create an longitudinal bending moment and shrinkage force which act parallel to the direction of the heating line, with much smaller magnitude that the transverse moment and shrinkage force.

Transverse shrinkage forces

Transverse bending moments and shrinkage forces along the heating line result from integration of inherent strains with respect to a unit longitudinal section [13]. According to the reference system defined in Figure 2-3, the transverse bending moment per unit length m_y ($\frac{Nm}{m}$ in SI units) is given by [23][22]

$$m_y = \int_{-\frac{g}{2}}^{d-\frac{g}{2}} E\epsilon^*_{rr} \frac{b_z}{b|_{z=0}} z dz = E\epsilon^*_{rr} d \left[\frac{d}{3} - \frac{g\pi}{8} \right] \ , \tag{3.76}$$

where the expression for b_z from equation 3.62 is used. The transverse shrinkage force per unit length f_x ($\frac{N}{m}$ in SI units) is given by

$$f_x = \int_{-\frac{g}{2}}^{d-\frac{g}{2}} E\epsilon_{rr}^{*} \frac{b_z}{b|_{z=0}} dz = E\epsilon_{rr}^{*} \frac{\pi}{4} d \ , \tag{3.77}$$

when the depth of the inherent strain zone is less than the thickness of the plate. Like the solution for the angular deformation however, two solutions of both the transverse bending moment and shrinkage force exist. Therefore, when the depth of the inherent strain zone exceeds the thickness, the bounds of integration change to include the entire plate thickness and the solution of the transverse bending moment per unit length becomes

$$m_y = E\epsilon_{rr}^{*} d \left[-\frac{d}{3} \left(\left(1 - \left(\frac{g}{d} \right)^2 \right)^{\frac{3}{2}} - 1 \right) - \frac{g}{4} \left(\frac{g}{d} \sqrt{1 - \left(\frac{g}{d} \right)^2} + \sin^{-1} \frac{g}{d} \right) \right] \ , \tag{3.78}$$

and the transverse shrinkage force becomes

$$f_x = \int_{-\frac{g}{2}}^{\frac{g}{2}} E\epsilon_{rr}^{*} \frac{b_z}{b|_{z=0}} dz = E\epsilon_{rr}^{*} d \left[\frac{g}{2d} \sqrt{1 - \left(\frac{g}{d} \right)^2} + \frac{1}{2} \sin^{-1} \frac{g}{d} \right] \ . \tag{3.79}$$

Because the transverse bending moment and shrinkage forces are a result of the inherent strains, they are assumed to act along the edge of the inherent strain zone.

Longitudinal shrinkage forces

The longitudinal bending moments and shrinkage forces which act along the heating line result from the integration of inherent strains with respect to a unit transverse section perpendicular to the heating line [13].

When the depth of the inherent strain zone is less than the plate thickness, the longitudinal bending moment (Nm in SI units) is expressed as [23][22]

$$m_x = \int_{-\frac{g}{2}}^{d-\frac{g}{2}} E\left(1-\nu\right) \epsilon_{rr}^{*} b_z z dz = E\left(1-\nu\right) \epsilon_{rr}^{*} bd \left(\frac{d}{3} - \frac{g\pi}{8} \right) \ , \tag{3.80}$$

and the longitudinal shrinkage force (N in SI units) is given by

$$f_y = \int_{-\frac{g}{2}}^{d-\frac{g}{2}} E\left(1-\nu\right) \epsilon_{rr}^{*} b_z dz = E\left(1-\nu\right) \epsilon_{rr}^{*} \frac{\pi}{4} bd \ . \tag{3.81}$$

When the depth of the inherent strain zone is greater than the plate thickness, the longitudinal bending moment becomes

$$m_x = \int_{-\frac{g}{2}}^{d-\frac{g}{2}} E\left(1-\nu\right) \epsilon_{rr}^{*} b_z z dz = E\left(1-\nu\right) \epsilon_{rr}^{*} bd \left[-\frac{d}{3} \left(\left(1 - \left(\frac{g}{d} \right)^2 \right)^{\frac{3}{2}} - 1 \right) \right. \ ,$$

$$-\frac{g}{4}\left(\frac{g}{d}\sqrt{1-\left(\frac{g}{d}\right)^2}+\sin^{-1}\frac{g}{d}\right)\Bigg] \tag{3.82}$$

and the longitudinal shrinkage force becomes

$$f_y = \int_{-\frac{g}{2}}^{\frac{g}{2}} E\left(1-\nu\right)\epsilon_{rr}^{*} b_z dz = E\left(1-\nu\right)\epsilon_{rr}^{*}\frac{bd}{2}\left[\frac{g}{d}\sqrt{1-\left(\frac{g}{d}\right)^2}+\sin^{-1}\frac{g}{d}\right] . \tag{3.83}$$

Within the inherent strain zone, the longitudinal bending moments and shrinkage forces cancel. Therefore, the moments and forces only act at the ends of it and are the total reaction. Because the longitudinal shrinkage forces cancel within the inherent strain zone and because they act perpendicularly to the heating path and are therefore affected by Poisson's ratio, the total relative longitudinal forces are much lower than the transverse forces. Therefore, while angular deformation occurs in both the transverse and longitudinal directions, the transverse deformation is dominant. If line heating is applied only to a fraction of the plate's total length, the inherent strain zone will have dimensions defined by the heating distance length and the thermal model calculated breadth. This is the case when the simplified model is applied to the pipe model; the heating is applied at the free end and the resultant bending moments and forces will act over the described area inducing deformation on the pipe.

3.4 Shell elements FEM analysis of a pipe

The equivalent shrinkage forces and bending moments estimated from the inherent strain zone are applied to the shell elements stress-displacement FEM model. The topology and the dimensions of the mesh were kept the same as those of the 3-D model. However instead of using 20-node parallelepiped elements and 15-node triangular prism elements, we employed 8-node rectangular shell elements and 6-node triangular elements. Shell elements allow the application of bending moments and the definition of thickness at the nodes emulating the mechanical properties of a 3-D element. The loads obtained from the simplified mechanical model were applied in one single step at the nodes corresponding to the inherent strain zone.

3.4.1 Results of the simplified thermal model

The results obtained by using the simplified thermal model for a flat plate can be used for the pipe since the radius of curvature of the heated area is large compared to the spot size, making the affected area almost flat. The simplified thermal model was used to obtain the temperature distribution and the inherent strain zone dimensions for the defined heating cases. As explained in Section 2.2.3, the heat input of the laser beam can be modeled as a truncated Gaussian distribution. The temperature distribution and the breadth and depth of the plastic region are computed by the thermal model presented in Section 3.3. For mild steel, Young's modulus and yield stress become very small at a temperature above 725^oC [12] [5]. Due to the restraint of the surrounding material during line heating process, not only the area with temperature above 725^oC will become plastic, but also the region with lower temperature is expected to become plastic. Thus we choose a critical temperature of 500^oC. The material properties of mild steel are also chosen to be those at 500^oC and the heat absorption rate is chosen to be 0.81, the same value used in Chapter 2. The results shown in this section correspond to a simulation with the same heating parameters used in the non-linear example indicated in Section 2.3. The simplified thermal model output values obtained after the calculations were:

1. Maximum temperature: *724.9* ^oC

2. Inherent strain zone breadth: *1.496 cm*

3. Inherent strain zone depth: *3.97 mm*

3.4.2 Results of the simplified mechanical model

The values obtained in the simplified thermal model are used as inputs of the simplified mechanical model to obtain the equivalent forces and bending moments to be applied to the shell stress-displacement analysis. Following the formulas defined in Section 3.3.8, the absolute magnitudes of the resultant forces and bending moments applied to the inherent strain zone (see Figure 3-6) were:

- $f_x : 2.130 \times 10^6 N/m$

- $f_y : 2.230 \times 10^4 N$

- $m_x : 6.272 \times 10^1 Nm$

- $m_y : 5.997 \times 10^3 Nm/m$

where the units of f_x and m_y are force and moment per unit length distributed along the heating length. The deformation magnitude computed using the simplified method is shown in Figure 3-7 which can also be decomposed into the x, y, z components (see Figures 3-8 to 3-10).

3.5 Comparison between the two models

3.5.1 Computational time

One of the advantages of the simplified model over the non-linear FEM analysis is the shorter computational time. The total time of the coupled non-linear FEM thermal and mechanical analysis using an SGI Irix machine running at *200 MHz* was *42,179 sec* in CPU

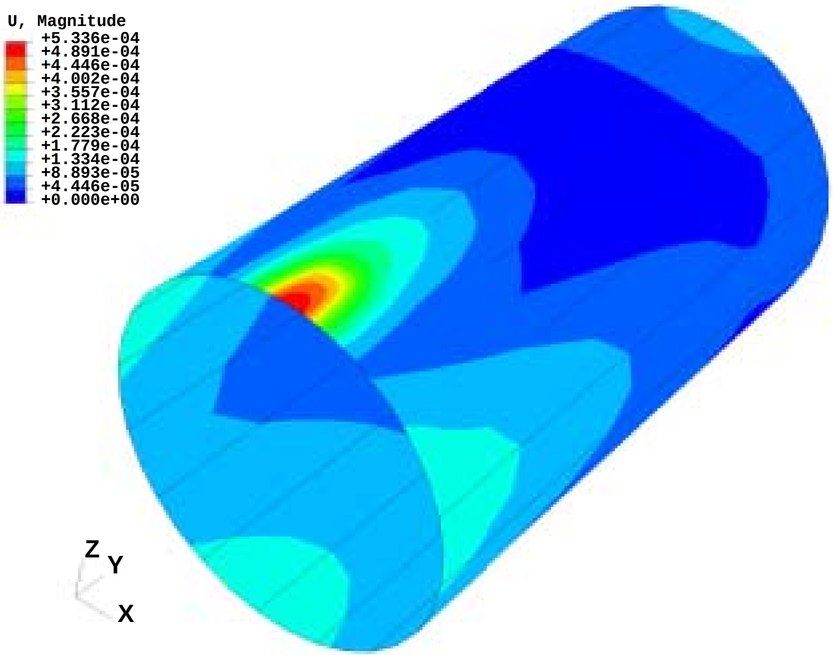


Figure 3-7: Predicted magnitude of total deformation using the simplified thermo-mechanical model (See Figure 2-12 for the corresponding 3-D FEM analysis)

time (*50,209 sec* in wall clock time). On the other hand the total average CPU time to perform both the simplified thermal model and the simplified mechanical model analysis is *1,601 sec* (*2,320 sec* in wall clock time) which represents *3.79%* of the computational time of the non-linear FEM analysis.

3.5.2 Temperature field prediction of both models

The predicted surface temperature for both models can be compared using the top view of the isothermal temperature field over the pipe's surface. The non-linear FEM analysis prediction is shown in Figure 3-11 while the temperature field using the simplified thermal model is shown in Figure 3-12. The maximum temperature and the temperature contour lines distribution in both cases have similar values and shapes. If we compare the temperature variation in time predicted by both models for a single node in the heating path we obtain Figure 3-13. The solid line plots the non-linear model prediction temperature and the dashed line plots the simplified model result. The difference in peak temperature is about 7% and the distribution in time is similar in both cases. Simulations using the different heating conditions indicated in Table 2.4 were performed in order to compare and validate the simplified thermal model.

The results obtained using both models are shown in Table 3.1. The difference in per-

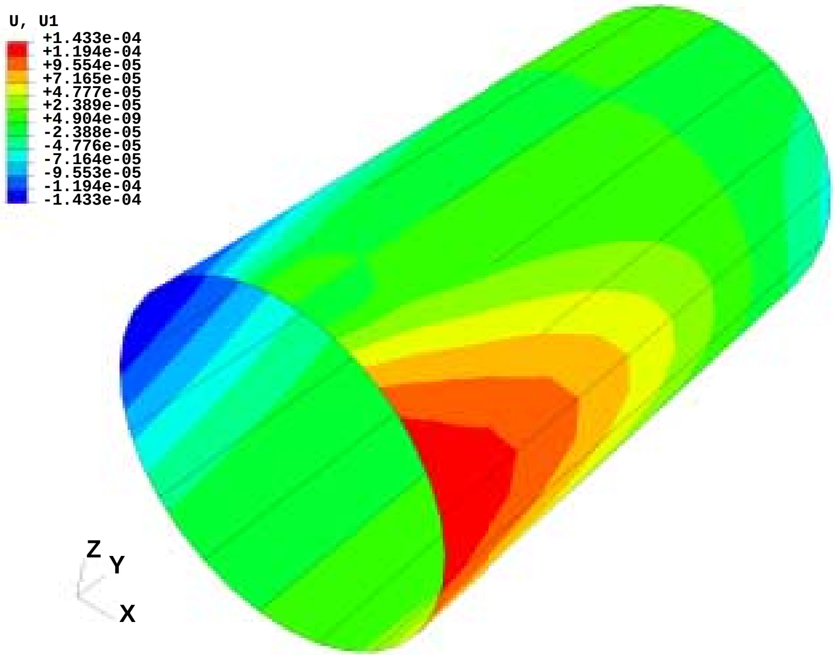


Figure 3-8: Predicted x component deformation using the simplified thermo-mechanical model (See Figure 2-13 for the corresponding 3-D FEM analysis)

centage between the non-linear and the simplified models lies around 10% of the maximum temperature. For all cases the simplified thermal model tends to overpredict the temperature value. The analysis shows a good match in the results and validates the simplified thermal model as a tool to perform multiple calculations and obtain an appropriate temperature field to be used in the simplified mechanical model. The largest error is obtained in the cases where a combination of low power and low speed is used (cases 6 and 8). This is probably because in those cases the heated area is small compared with the distance between nodes and the time it takes for the heating source to move from one node to the next is also larger. These two factors contribute to a less accurate temperature field prediction. For these cases a finer mesh in the heated area is recommended.

3.5.3 Predicted deformation comparison between the two models

The results obtained from the simplified thermal model were used to calculate the equivalent forces and bending moments defined in section 3.3.8 and then apply them to the simplified mechanical model described in section 3.4. The comparison between the predicted deformations using the heating parameters indicated in Section 2.3 for the $x - z$ and the $y - z$ planes at the pipe's free end using both procedures are shown in Figures 3-14 and 3-15. The non-linear FEM deformation prediction at the free end has a 15.5% of difference in

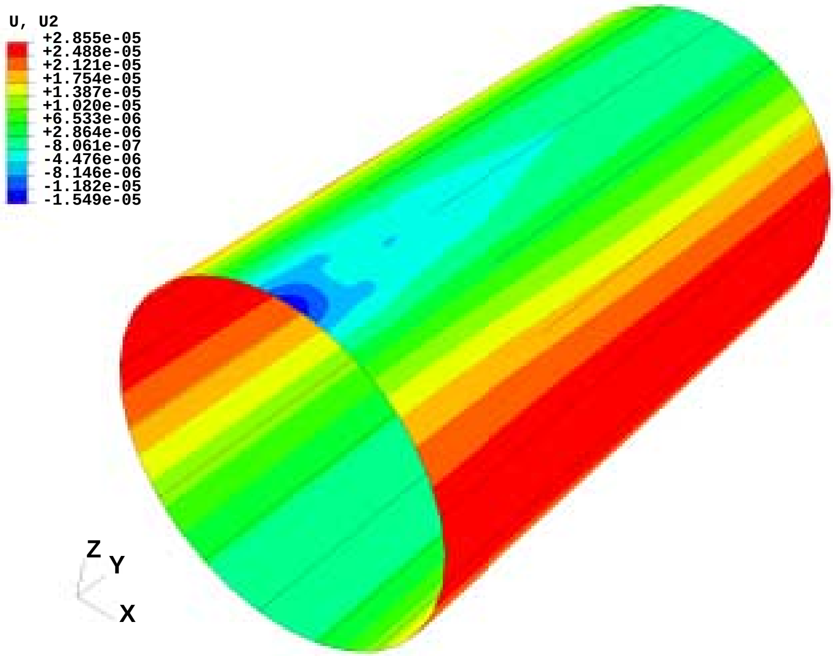


Figure 3-9: Predicted y component deformation using the simplified thermo-mechanical model (See Figure 2-14 for the corresponding 3-D FEM analysis)

Case	Power input W	Source speed mm/s	Spot size radius mm	T non linear model ^oC	T simp. model ^oC	Difference %
1	1300	1.6	12.7	672.3	724.8	7.2
2	1000	1.6	11.8	552.8	605.9	8.7
3	700	1.0	8.9	611.3	663.3	7.8
4	1100	1.3	10.8	732.3	785.1	6.7
5	800	0.7	10.8	646.7	716.1	9.6
6	600	0.3	12.7	528.1	629.6	16.0
7	900	1.0	11.8	566.9	655.3	13.4
8	600	0.7	8.9	527.9	636.5	17.0

Table 3.1: Predicted temperature comparison between both models

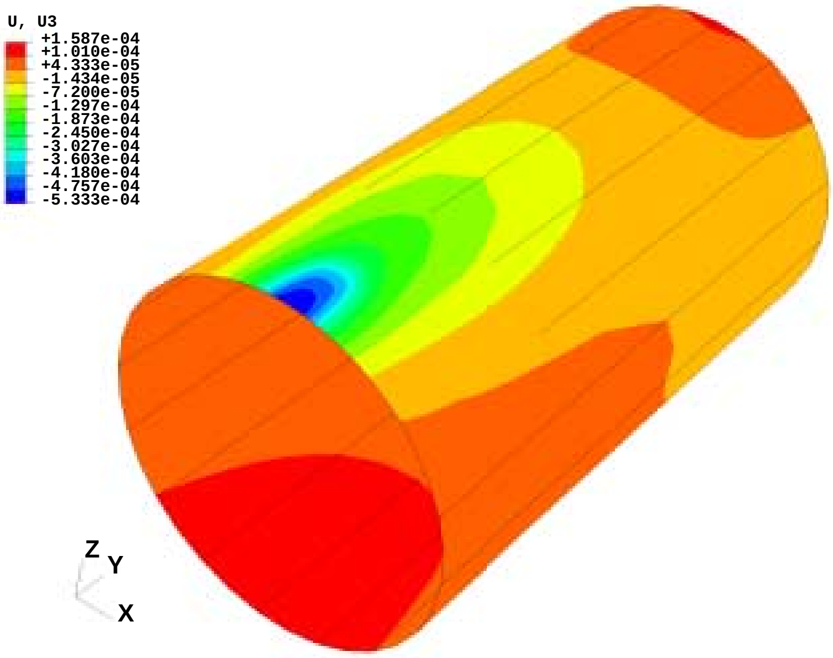


Figure 3-10: Predicted z component deformation using the simplified thermo-mechanical model (See Figure 2-15 for the corresponding 3-D FEM analysis)

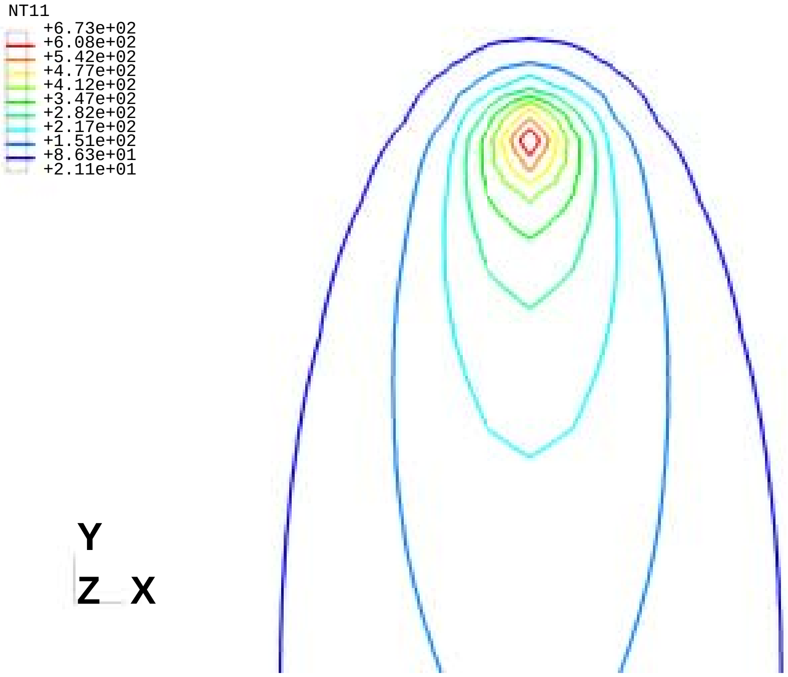


Figure 3-11: Coupled thermo-mechanical model surface temperature distribution

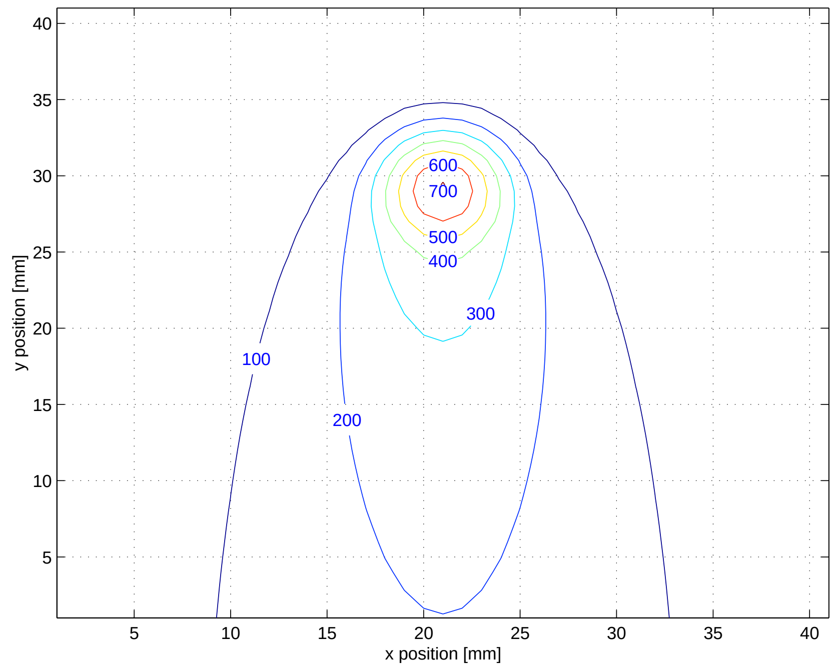


Figure 3-12: Simplified thermal model surface temperature distribution

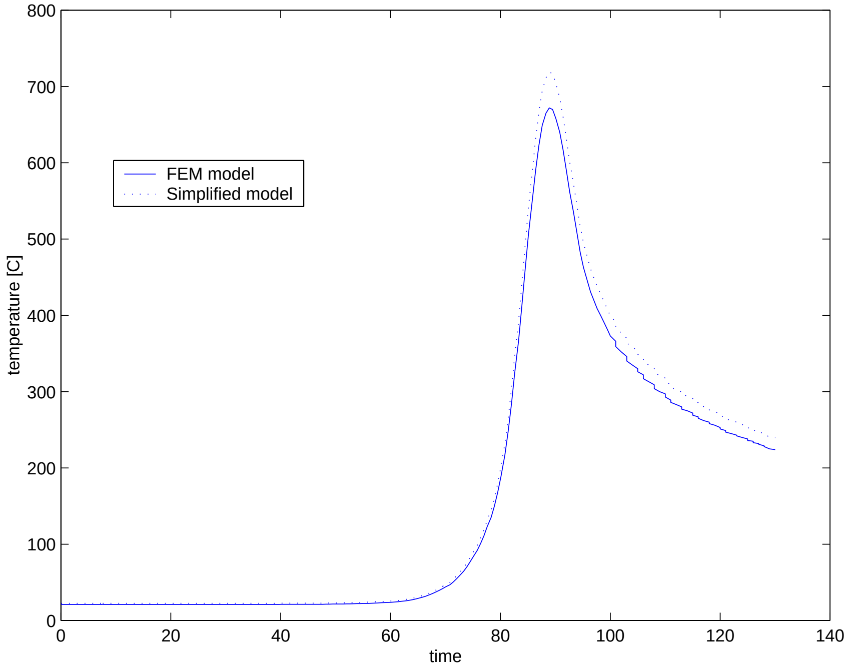


Figure 3-13: Temperature variations at a single node located at the center of the heating path

magnitude with the one obtained using the simplified mechanical model. The shape is similar in both cases. In the $x - z$ plane the figures are also similar and both models predicts similar deformations starting form the centerline ($x = 0$). A comparison of the predicted

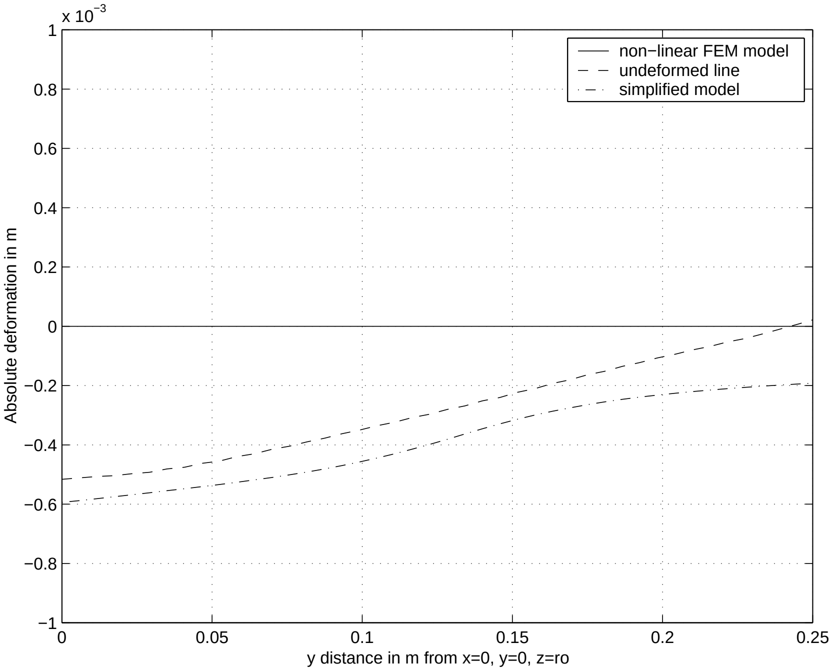


Figure 3-14: Deformation prediction in the y-z plane using the non-linear FEM model and simplified mechanical model

deformation at (0,0,r_o) in the z direction using both models under the heating conditions specified in Table 2.4 is shown in Table 3.2.

Here we see that the difference in percentage for the predicted deformation between models was increased with respect to the difference obtained in the thermal model. The percentage error lies around 20%, and for all cases the simplified model predicts a higher absolute deformation when compared with the non-linear mechanical model. The highest values are again in the cases where a slower speed and smaller spot size were used. This is probably because the difference obtained in the thermal models is carried to the mechanical analysis, resulting in an increase of the error.

In order to make a sensitivity analysis and determine the effect of the temperature difference in the deformation prediction, we performed a simplified mechanical analysis using as inputs the temperatures obtained from the non-linear thermal model for the same cases indicated. The results are shown and compared between the non-linear results versus the corrected simplified model results in Table 3.3. The difference of the final predicted deformation between the non-linear and simplified mechanical models was reduced in all cases to around 10%.

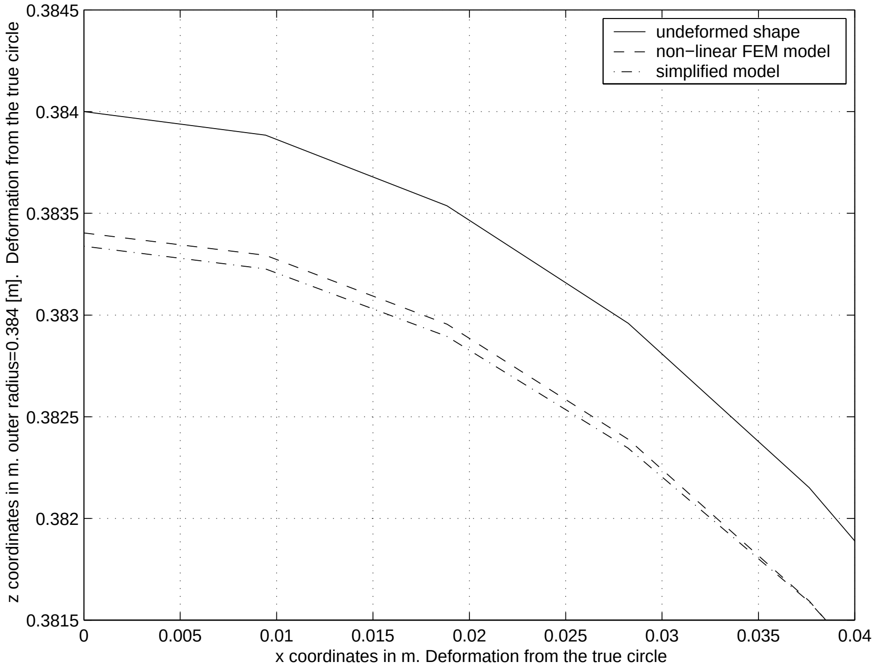


Figure 3-15: Deformation prediction in the x-z plane using the non-linear FEM model and simplified mechanical model

Case	Power input	Source speed	Spot size radius	Deform. non linear model	Deformation simplified model	Difference
	W	mm/s	mm	$mm \times 10^{-4}$	$mm \times 10^{-4}$	%
1	1300	1.6	12.7	-5.160	-5.94	15.5
2	1000	1.6	11.8	-2.203	-2.51	13.9
3	700	1.0	8.9	-2.833	-3.22	18.1
4	1100	1.3	10.8	-3.593	-4.83	25.6
5	800	0.7	10.8	-3.662	-4.44	17.5
6	600	0.3	12.7	-2.856	-3.96	27.8
7	900	1.0	11.8	-3.258	-3.59	14.3
8	600	0.7	8.9	-2.219	-3.04	27.2

Table 3.2: Predicted deformation comparison between both models

Case	Power input W	Source speed mm/s	Spot size radius mm	Deform. non linear model $mm \times 10^{-4}$	Corrected def. simp. model $mm \times 10^{-4}$	Diff. after correction %
1	1300	1.6	12.7	-5.160	-5.676	10.0
2	1000	1.6	11.8	-2.203	-2.33	5.4
3	700	1.0	8.9	-2.833	-3.01	5.8
4	1100	1.3	10.8	-3.593	-4.17	16.2
5	800	0.7	10.8	-3.662	-4.09	10.4
6	600	0.3	12.7	-2.856	-3.45	17.2
7	900	1.0	11.8	-3.258	-3.20	1.7
8	600	0.7	8.9	-2.219	-2.62	15.3

Table 3.3: Predicted deformation comparison between both models

3.6 Discussion

From the comparisons of the results obtained in the non-linear and simplified thermo-mechanical methods, we observe

1. There is a good correlation between the results of the two thermal models. However, some of the heating combinations used in the simulations increased the difference of the predicted temperature magnitude between the two models. This effect can be corrected by modifying the pipe's mesh making it finer near the heated area. This will reduce the nodal distance, providing a more accurate temperature field prediction in the model when a small spot size and speed are used.

2. The results of the simplified thermo-mechanical model predict a higher temperature and a higher deformation than the 3-D non-linear model. The temperature result can be affected by the element size and that error is carried to the mechanical analysis which can also has an additional error due to the same reason.

3. The deformation prediction difference was reduced by correcting the temperature input of the simplified mechanical model by an additive factor equal to the error difference between thermal models. This procedure made the final deformation result closer for both models.

4. There is a good agreement between the results obtained from both methods, but a more extensive research including real-scale experiments has to be considered to completely validate the results obtained in this chapter

5. The simulations performed in chapters 2 and 3 validate the simplified thermo-mechanical model as a way to obtain fast and accurate results to create the database to be used in the neural network application in the Chapter 4.

Chapter 4

Neural network application

4.1 Introduction

Line heating conditions to be applied to a pipe have to be determined in a short time in order to make an automatic process efficient. A neural network [10] is an efficient tool to perform this type of analysis since it allows the user to obtain an output almost instantaneously. The model written in MATLAB [8] uses the data obtained by both the non-linear finite element analysis and the simplified thermo-mechanical model in a manner conforming to the neural network. Using the deformation measured at the pipe's free end perimeter as input, we develop a neural network capable of predicting the necessary heating power and heating source speed to reform the pipe circularity.

4.2 Neural network principles

The neural network method is based on the concept that the human brain computes in a different way than a conventional digital computer. In fact, the brain is a highly complex, non-linear and parallel information-processing system [10]. It has the capability to organize its neurons to perform certain computations many times faster than the fastest computer. The human brain adapts to its surrounding environment through a learning process. The same principle is used by an artificial neural network: it models the way the brain performs a particular task acquiring the knowledge from the environment through a similar learning process, and uses interneuron connection strengths known as synaptic weights to store such knowledge. This ability to learn, and therefore generalize, gives the neural network its strength: the capacity of produce reasonable outputs for inputs not encountered during the learning process [10].

4.2.1 General structure

Basic neuron model

The neuron shown in Figure 4-1 is the basic and fundamental information-processing unit for the operation of a neural network. The basic elements of the neuron are [10]:

- Synapses or connecting links: Each characterized by a weight w_{jm} of its own. The signal x_i at the input of synapse i connected to neuron j is multiplied by the synaptic weight w_{ji} where j refers to the neuron in question and i refers to the input end of the synapse to which the weight refers.
- Summing junction: All the input signals weighted by the respective synapses of the neuron are added.
- Activation function: Used to limit the amplitude range of the output signal of a neuron to a finite value.
- Bias: Externally applied function that increases or reduces the net input of the activation function depending whether it is positive or negative, respectively.

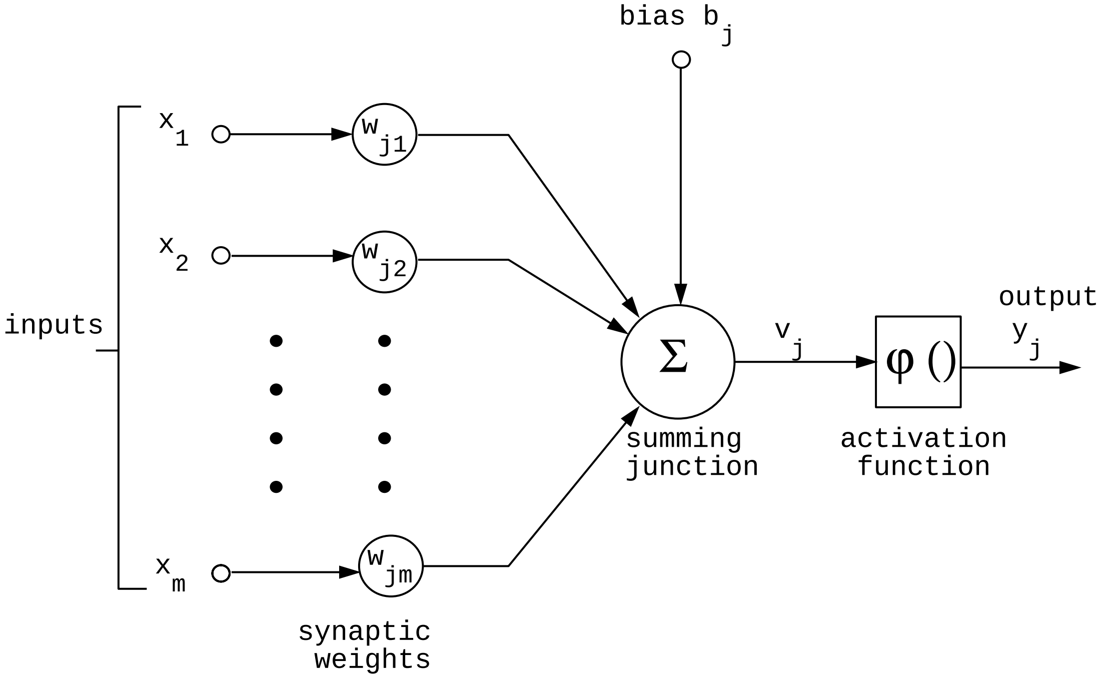


Figure 4-1: Esquematic diagram of a neuron (adapted from [10]).

The neuron j can be written by the following pair of equations [10]:

$$u_j = \sum_{i=1}^{m} w_{ji} x_i \ , \tag{4.1}$$

and

$$y_j = \varphi(u_j + b_j) \ , \tag{4.2}$$

where $x_1, x_2, ..., x_m$ are the input signals, $w_{j1}, w_{j2}, ..., w_{jm}$ are the synaptic weights of the neuron j, u_j is the linear combiner output of the input signals, b_j is the bias, φ is the activation function and y_j is the output signal of the neuron. The bias b_j applies to the output u_j of the linear combination according to the equation

$$v_j = u_j + b_j \ , \tag{4.3}$$

so if the bias is considered as the 0^{th} element, equations (4.1) and (4.2) can be rewritten as [10]

$$v_j = \sum_{i=0}^{m} w_{ji} x_i \ , \tag{4.4}$$

and

$$y_j = \varphi(v_j) \ , \tag{4.5}$$

resulting in the neuron shown in Figure 4-2

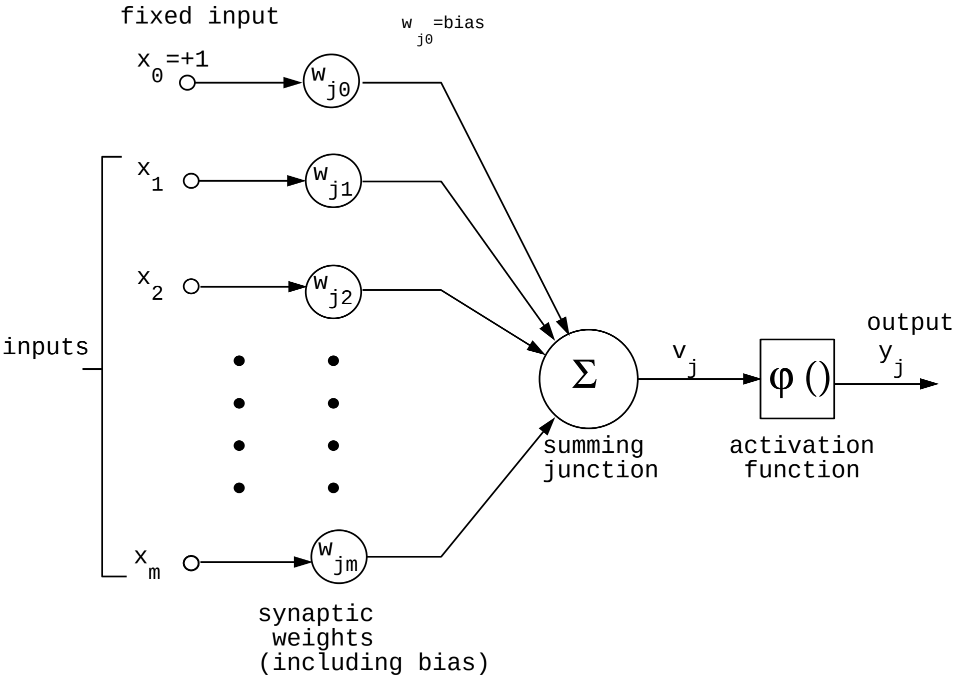


Figure 4-2: Esquematic diagram of a neuron including the bias as an input element (adapted from [10])

4.2.2 Feedforward networks

There are in general three different types of architectures that can be identified in this class. Single-layer, Multilayer and Recurrent networks [10]. We describe the first two architectures since they are the types that will be used in our analysis.

Single-layer feedforward networks

They have an input layer of source nodes that project onto an output layer of neurons, but not vice-versa. Figure 4-3 shows a single layer network with four nodes in both the input and output layers. The input layer is not considered as a layer, since no computation is performed there.

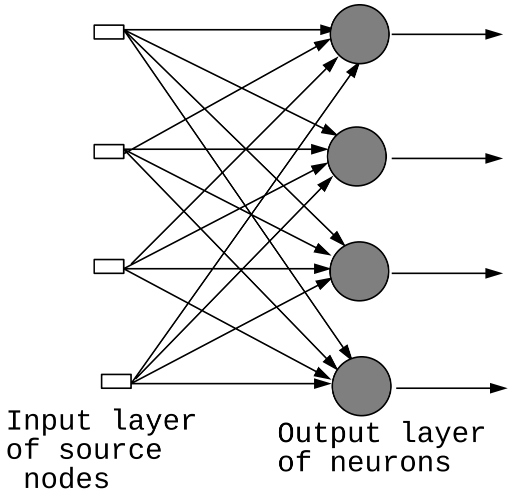


Figure 4-3: Single layer feedforward architecture (adapted from [10])

Multilayer feedforward networks

This type of network has one or more hidden layers whose computation nodes are called hidden neurons. Their function is to connect the external input and the network input in a useful manner. By adding one or more hidden layers, the network is enabled to extract high order statistics, specially valuable when the size of the input layer is large. The elements of the input vector are presented to the first layer of neurons. Their output acts as input for the second hidden layer and so on for the rest of the network until the final output is obtained. It corresponds to the overall response of the structure to the activation pattern represented by the input vector. Figure 4-4 shows the layout of a multilayer feedforward neural network for the case of a single hidden layer. To identify a particular structure, the notation $m - h_1 - h_2 - q$ is used, where m is the number of source nodes, h_1 and h_2 the number of neurons in the first and second hidden layer respectively, and q the numbers of neurons in the output layer. A neural network is said to be fully connected if every node in each layer is connected to every other node in the adjacent forward layer. If any connection is missing, it is partially connected.

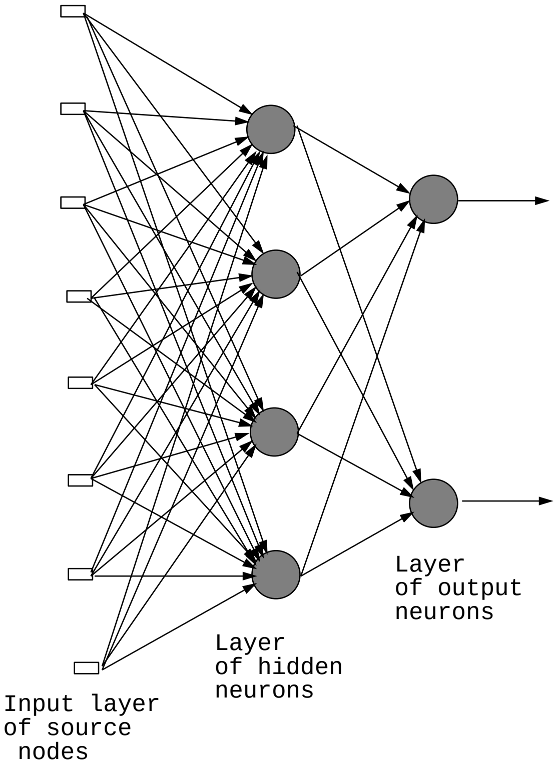


Figure 4-4: Multiple layer feedforward architecture (adapted from [10])

4.2.3 Learning process

Probably one of the most important characteristic of the neural networks is their ability to learn and improve their performance through this process [10]. The neural network learns about its environment through an interactive process of adjustment applied to its synaptic weights and bias levels, becoming more knowledgeable after each iteration of the learning process. For the learning process the neural network uses an already known set of data called training set. It consists in inputs and their corresponding outputs which are presented together to the neural network. There are many different learning processes and paradigms, but in this chapter we will focus only in describing the learning rule and the learning paradigm we used in our research.

Error correction learning

The output signal of the neuron j, $y_j(n)$ shown in Figure 4-5, where n is number of iterations performed by the net, is compared to the corresponding desired response included in the training set denoted by $d_j(n)$. Their difference is the error signal $e_j(n)$ defined by

$$e_j(n) = d_j(n) - y_j(n) \ . \tag{4.6}$$

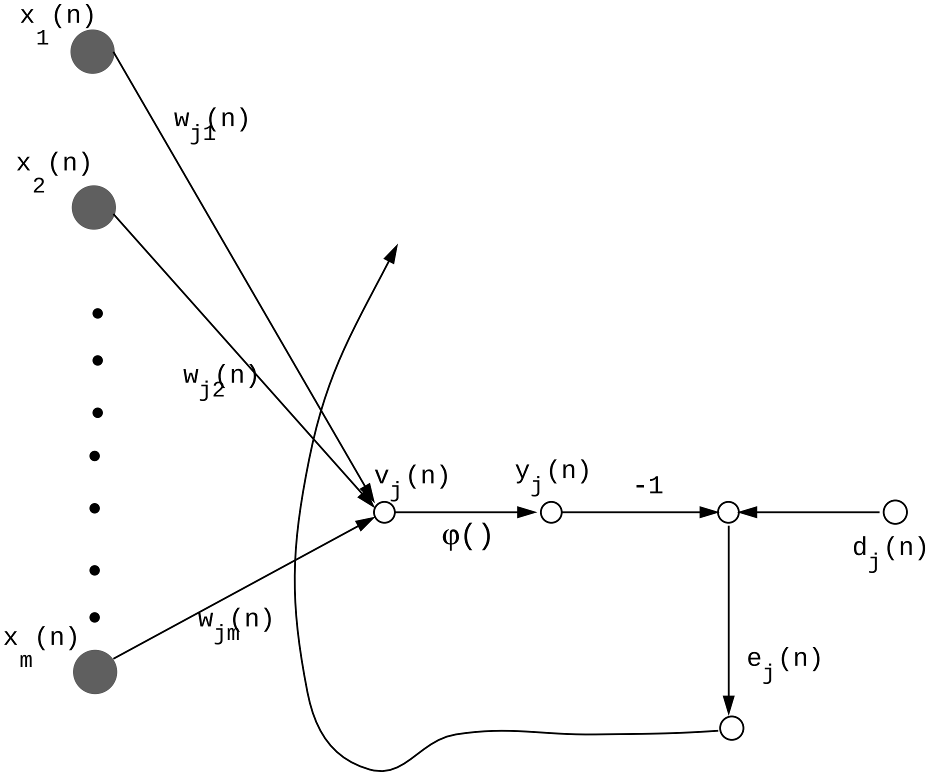


Figure 4-5: Error correction learning signal flow diagram(adapted from [10]).

The error function travels backwards acting as a control mechanism that applies corrective adjustments to the synaptic weights of the neuron j to make the output $y_j(n)$ closer to $d_j(n)$ in the next iteration. The goal is to minimize the cost function $\Upsilon(n)$ defined by

$$\Upsilon(n) = \frac{1}{2}e_j^2(n) \ . \tag{4.7}$$

The adjustment of the weights continues until the system reaches a steady state after n iterations, a point at which the learning process is terminated. Then the minimization of the cost function $\Upsilon(n)$ leads to a learning rule known as the *Widrow-Hoff rule* [10]. If $w_{ji}(n)$ denotes the value of synaptic weight w_{ji} of neuron j excited by element $x_i(n)$ of the signal vector $\mathbf{x}(n)$ at time step n, the adjustment $\Delta w_{ji}(n)$ applied to the synaptic weight w_{ji} at time step n is defined by

$$\Delta w_{ji}(n) = \eta e_j x_i(n) \ , \tag{4.8}$$

where η is a positive constant that determines the rate of learning as the learning process proceeds from one step to another. Therefore, η is referred to as the *learning rate parameter* [10]. So the adjustment of the synaptic weight of a neuron is proportional to the product of the error signal and the input signal of the synapse in question. The value of η determines the stability and convergence of the iterative learning process playing a key role in determining the performance of error-correction learning. The rate of learning is a scalar that perhaps be decreased at each iteration as learning progresses, or it may perhaps be a constant fixed value throughout the learning process. If η is selected to decrease, the rate at which it decreases affects the speed of convergence to the optimum solution.

Supervised learning

Supervised learning, also known as *learning with a teacher* [10], can be seen as a process where a teacher is available to check if a system is performing correctly and indicating a desired response, or to validate the acceptability of the system's response, or to indicate the amount of error in system performance [14]. The teacher has knowledge of the environment, with that knowledge being represented by a set of input-output examples. However the environment is unknown to the neural network. If both are exposed to a training vector, the teacher is able to provide the neural network with the desired response for a training vector. The desired response represents the optimum action to be performed by the neural network. The network parameters are adjusted under the combined influence of the training vector and the error signal defined in the previous paragraph. The adjustment is carried step by step trying to make the neural network to emulate the teacher in a statistical sense. When the knowledge of the teacher is transferred by training of the network, the teacher can let the net deal with the environment by itself.

The error correction learning is a closed loop feedback system, but the unknown environment of the supervised learning is not in the loop. The performance of the system is measured in terms of the sum of the squared errors over the training sample, which can be visualized as a multidimensional error surface averaged over all possible input-output examples with the free parameters as coordinates. Any operation of the system under the teacher's supervision is represented as a point on the error surface. For the system to improve performance over time and therefore learn from the teacher, the operating point has to move down successively toward a minimum point of the error surface that can be a local or a global minimum. A supervised learning network (explained next) is able to do this with the gradient information it has. The gradient of an error surface at any point is a vector that points in direction of the steepest descent.

4.2.4 Perceptrons

Single layer perceptrons

The perceptron is the simplest form of a neural network used for classification of patterns that are linearly separable [10]. It is built around a non-linear neuron and consist of a linear combiner followed by a hard limiter that performs the sigmum function (see Figure 4-6) limiting the amplitude of the output. The output will be equal to +1 if the hard limiter is positive or −1 if it is negative, classifying the answers in two decision regions C_1 and C_2 separated by the hyperplane defined by

$$\sum_{i=1}^{m} w_i x_i + b = 0 \; . \tag{4.9}$$

The case for two input variables x_1 and x_2 is illustrated in Figure 4-7. The goal of the perceptron is to correctly classify the set of externally applied stimuli x_m into one of the two classes C_1 or C_2. From Figure 4-7 it can be seen that if a point (x_1, x_2) lies above the boundary line, is assigned to class C_1, and if lies below it is assigned to class C_2. The effect of the bias b is only shifting the decision boundary away from the origin. The single neuron also forms the basis of an *adaptive filter* [10], a name given to the mathematical numerical process of adjustment of weights to produce the desired response. The adaptive filter numerical techniques used to adjust the synaptic weight values include the steepest descent

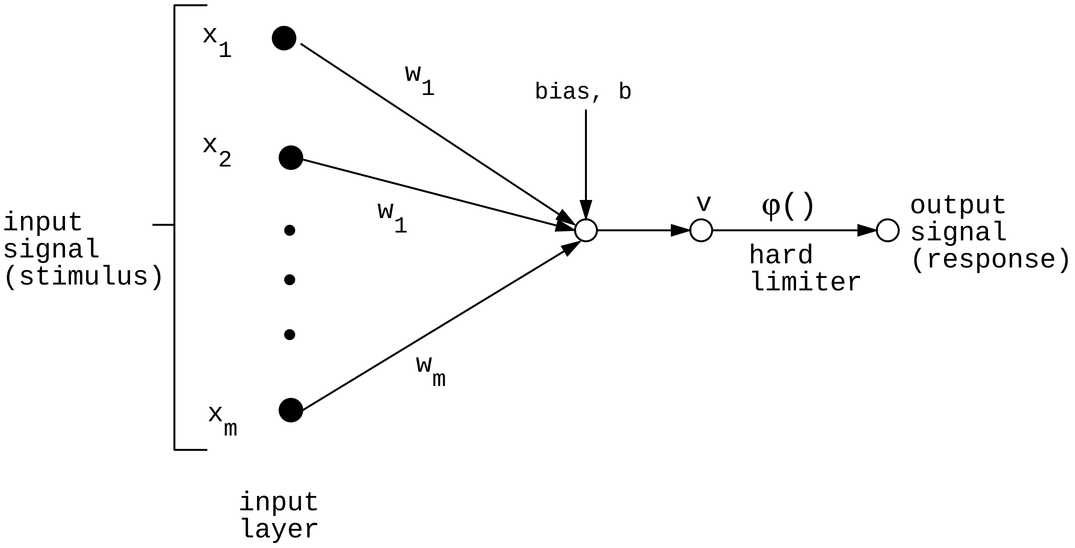


Figure 4-6: Architecture of a single layer perceptron (adapted from [10]).

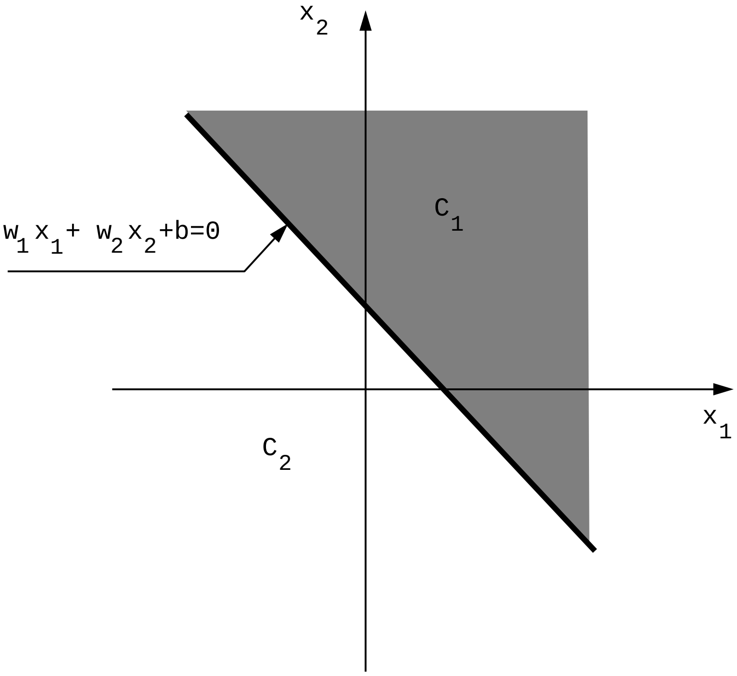


Figure 4-7: Hyperplane $w_1x_1 + w_2x_2 + b = 0$ for two input variables x_1 and x_2 (adapted from [10]).

method, Newton's method, Gauss-Newton method and Least-Mean-Square method [18]. Any of them can be implemented in a neural network.

Multilayer perceptrons

The multilayer perceptron shown in Figure 4-8 is a generalization of the single-layer perceptron and consists of a set of sensory units that constitute the input layer, one or more hidden layers of computational nodes and an output layer of computation nodes [10]. The input signal propagates through the network in a layer-by-layer basis. The network in the figure has two hidden layers and an output layer and is fully connected. The first hidden layer is fed from the input layer, and the outputs are in turn applied to the next hidden layer and so on for the rest of the network. In a multilayer perceptron there are two kinds of signal that can be identified:

1. Function signal: It is an input signal or stimulus that comes in at the input end of the network, propagates forward neuron by neuron, and emerges at the output end of the network as an output signal. At each neuron of the network through which a function signal passes, the signal is calculated as a function of the inputs and associated weights are applied to that neuron.

2. Error signal: It is originated at an output neuron and propagates backwards layer by

layer through the network as explained in Section 4.2.3.

Each hidden or output neuron of a multilayer perceptron is designed to compute the function signal appearing at the output of a neuron, which is expressed as a continuous nonlinear function of the signal and synaptic weights associated to that neuron, and to compute an estimate of the gradient vector which is needed for the backward pass. One of the most popular methods to train this type of networks is called error backpropagation algorithm (explained in Section 4.2.5), based on the error correction rule explained in Section 4.2.3. A multilayer perceptron has three distinctive characteristics [10]:

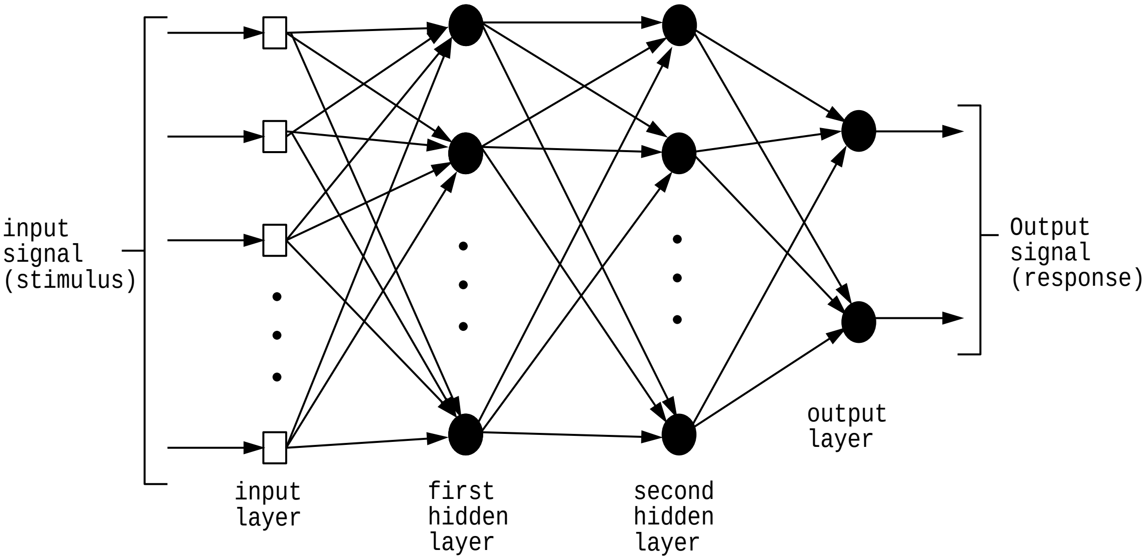


Figure 4-8: Architecture of a multilayer perceptron (adapted from [10]).

- The model of each neuron includes a non-linear activation function. Since the nonlinearity is smooth, the function has to be differentiable everywhere. A commonly used form is the sigmoid function defined by the logistic function:

$$y_j = \frac{1}{1+e^{-v_j}} \;, \tag{4.10}$$

  where v_j is the weighted sum of all synaptic inputs plus the bias of neuron j, and y_j is the output of the neuron. If non-linearities are not used, the input-output could be reduced to a single layer perceptron.

- It contains one or more layers of hidden neurons. These neurons enable the network to learn more complex tasks than the single layer case.

- There is a great degree of connectivity between the weights of the network.

The combination of this characteristics and the ability to learn from experience through training gives this tool its ability to solve complex problems.

4.2.5 Backpropagation algorithm

Backpropagation consists of two passes through the different layers of the network: a forward pass and a backward pass. In the forward pass the input vector is applied to the nodes of the network propagating its effect layer by layer obtaining at the output the actual response of the network. During the forward pass the synaptic weights are all fixed. During the backward pass the synaptic weights are adjusted in accordance with the error correction rule using the error signal generated in the comparison between the target and the output of the network. This error travels backwards correcting the synaptic weights to make the response of the network closer to the desired response in a statistical sense [10]. The backpropagation algorithm uses the error signal $e_j(n)$ defined by equation (4.6). The value of the total error energy $\Upsilon(n)$ is obtained summing equation (4.7) over all the neurons in the output layer and is defined by

$$\Upsilon(n) = \frac{1}{2}\sum_{j \epsilon C} e_j^2(n) \ , \tag{4.11}$$

where C includes all the neurons in the output layer. If N is the number of examples contained in the training set, the averaged squared error energy is obtained by summing $\Upsilon(n)$ over all n, the number of iterations performed by the net, and normalizing with respect to the set size N (ie. number of examples in the training set) as shown by

$$\Upsilon_{av} = \frac{1}{N}\sum_{n=1}^{N} \Upsilon(n) \ . \tag{4.12}$$

Both $\Upsilon(n)$ and Υ_{av} are functions of the synaptic weights and bias levels. For a training set, Υ_{av} represents the cost function or measure of learning performance. The final objective is to minimize it by the adjust of the free parameters of the network using a simple method of training, in which the weights are updated until one epoch is completed. An epoch is the complete presentation of an entire training set. The adjustments of the weights are made in accordance with the respective errors computed for each training set element presented to the network. So the arithmetic average of these individual changes over the training set is therefore an estimate of the true change that would result from modifying the weights based on minimizing the cost function Υ_{av} over the entire training set. The induced local field $v_j(n)$ produced at the input of the activation function associated with neuron j shown in Figure 4-9 is therefore

$$v_j(n) = \sum_{i=0}^{m} w_{ij}(n) y_i(n) \ , \tag{4.13}$$

where m is the total number of inputs (excluding the bias) applied to the neuron j. The

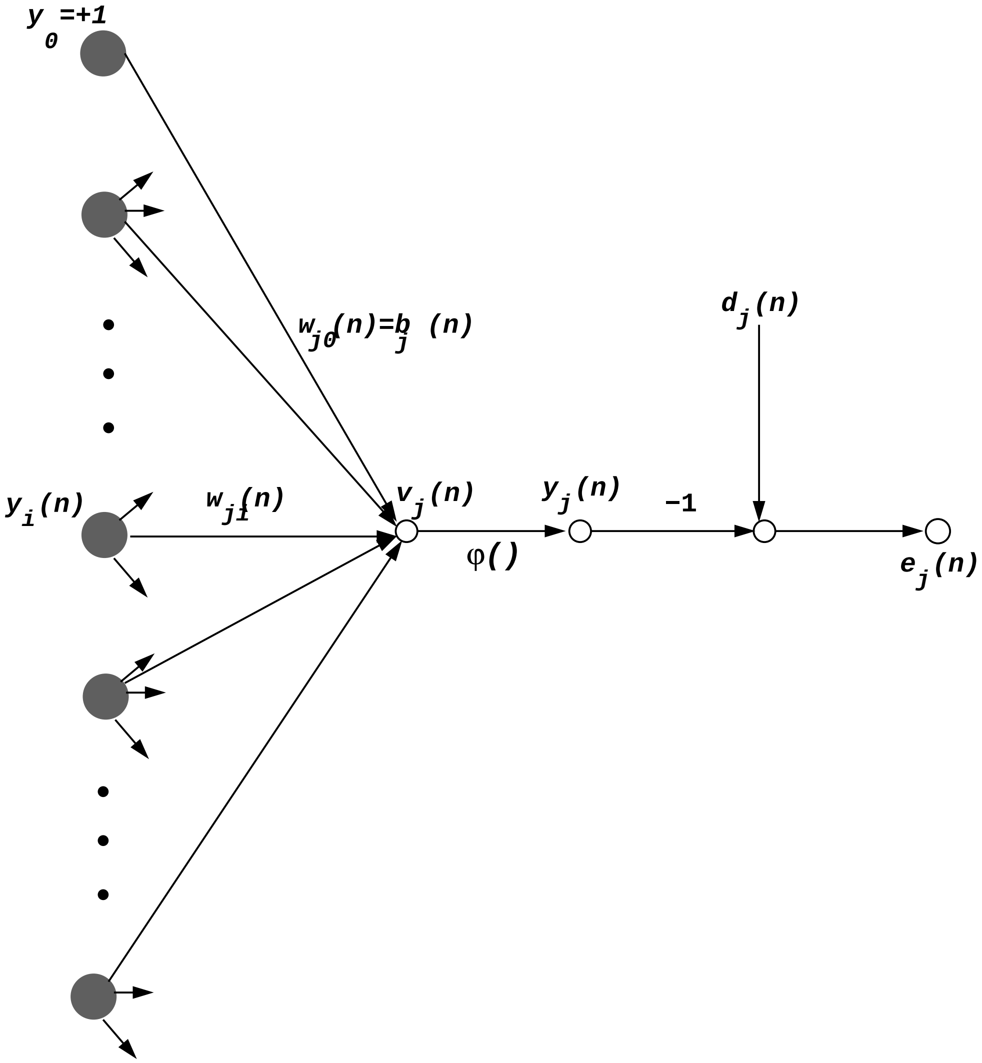


Figure 4-9: Signal-flow graph of output neuron j (adapted from [10])

synaptic weight w_{jo} (corresponding to a fixed input $y_o = +1$) equals the bias b_j applied to neuron j. Hence the function signal $y_j(n)$ appearing at the output of neuron j at iteration n is

$$y_i(n) = \varphi_j(v_j(n)) \ . \tag{4.14}$$

The backpropagation algorithm applies a correction $\Delta w_{kj}(n)$ to the synaptic weight $w_{ij}(n)$ which is proportional to the partial derivative $\partial \Upsilon(n)/\partial w_{ji}(n)$. According to the chain rule, this derivative can be expressed as:

$$\frac{\partial \Upsilon(n)}{\partial w_{ji}(n)} = \frac{\partial \Upsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \ . \tag{4.15}$$

This derivative represents the sensitivity factor, which determines the direction of search in weight space for the synaptic weight w_{ij}. Differentiating equation 4.7 with respect to $e_j(n)$ we get

$$\frac{\partial \Upsilon(n)}{\partial e_j(n)} = e_j(n) \ , \tag{4.16}$$

and differentiating equation (4.6) with respect to $y_j(n)$ we get

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \ . \tag{4.17}$$

Differentiating equation (4.14) with respect to $v_j(n)$ we get

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \varphi_j^{'}(v_j(n)) \ , \tag{4.18}$$

and differentiating equation (4.13) with respect to $w_{ji}(n)$ we get

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) \ . \tag{4.19}$$

Substituting equations (4.16) to (4.19) in (4.15) yields

$$\frac{\partial \Upsilon(n)}{\partial w_{ji}(n)} = -e_j(n)\varphi_j^{'}(v_j(n))y_i(n) \tag{4.20}$$

The correction $\Delta w_{ji}(n)$ applied to $w_{ji}(n)$ is defined by the *delta rule* [10]

$$\Delta w_{ji}(n) = -\eta \frac{\partial \Upsilon(n)}{\partial w_{ji}(n)} \ , \tag{4.21}$$

where η is the learning rate indicated previously. The minus sign accounts for the gradient descent in weight space. Substituting equation 4.20 in equation (4.21) we obtain

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \ , \tag{4.22}$$

where $\delta_j(n)$ is the local gradient defined by

$$\begin{aligned}
\delta_j(n) &= -\frac{\partial \Upsilon(n)}{\partial v_j(n)} && (4.23)\\
&= -\frac{\partial \Upsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} && (4.24)\\
&= e_j(n)\varphi_j^{'}(v_j(n)) \ .
\end{aligned}$$

This means that the local gradient $\delta_j(n)$ for output neuron j is equal to the product of the corresponding error signal $e_j(n)$ for that neuron and the derivative $\varphi_j^{'}(v_j(n))$ of the associated active function.

When the neuron j is located in a hidden layer of the network, there is no specified response for that neuron. The error signal for a hidden neuron has to be determined in terms of the error signals of all the neurons to which that hidden neuron is directly connected. For a 2-layer neural network like the one shown in Figure 4-10, the local gradient $\delta_j(n)$ for the hidden neuron j can be expressed as

$$\delta_j(n) = \varphi_j^{'}(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \ , \qquad (4.25)$$

where the index k represents the output neuron and $\sum_k \delta_k(n) w_{kj}(n)$ is the local gradient $\delta_k(n)$ at the neuron k.

Improving the backpropagation neural network performance

There exist some techniques that significantly improve the performance of the backpropagation method [10]:

1. Sequential and batch update: The sequential learning is a mode of operation where the weight update is performed after the presentation of each training example. On the other hand the batch learning mode updates its weights after the presentation of all the training examples that constitute an epoch. The first one is computationally faster, especially when the data set is large and highly redundant.

2. Maximizing information content: The information contained in the training example should be as large and different as possible.

3. Activation function: A multilayer perceptron using backpropagation can use an antisymmetric or a non-symmetric sigmoid activation function to improve its performance. A common function used to satisfy the first condition is the hyperbolic tangent sigmoid function like the one shown in Figure 4-11, defined by

$$\varphi(v) = a \ tanh(bv) \ , \qquad (4.26)$$

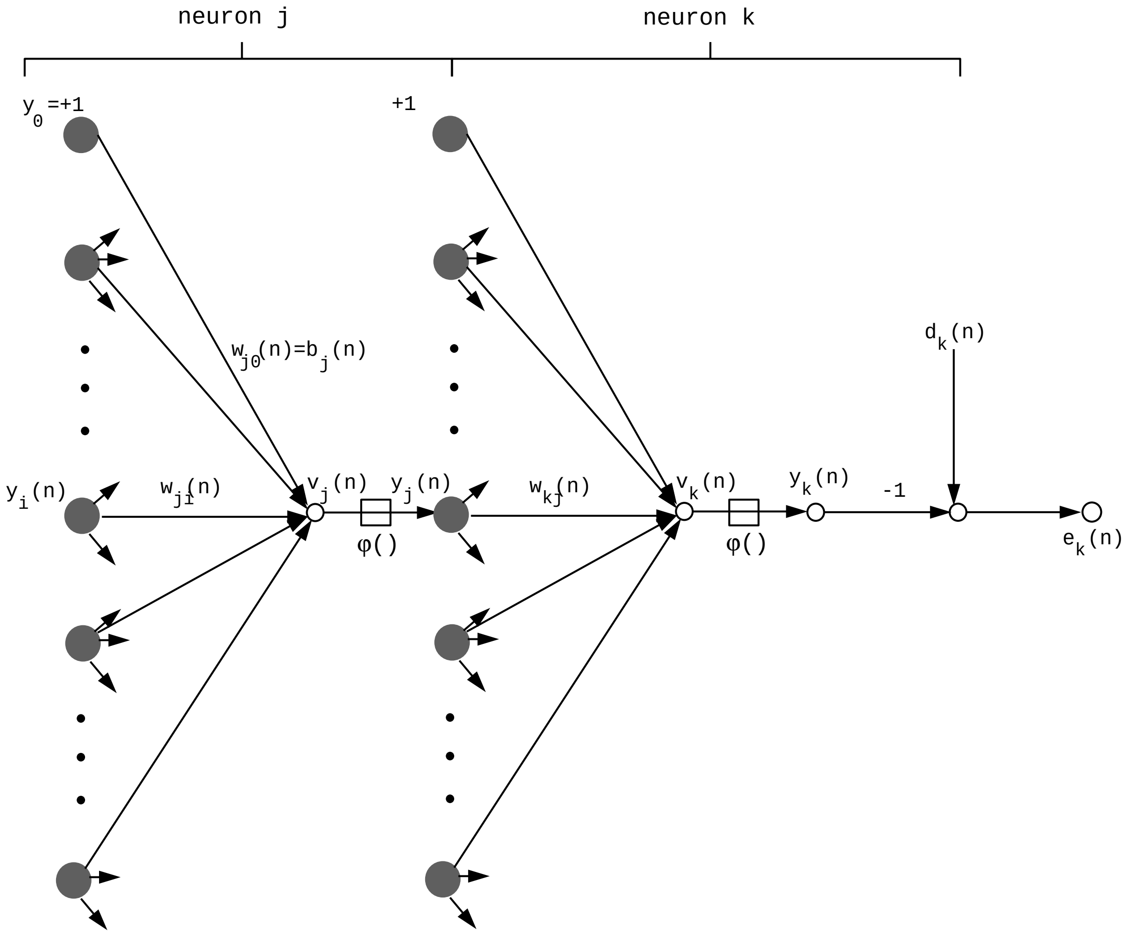


Figure 4-10: Signal-flow graph of output neuron k connected to hidden neuron j (adapted from [10]).

where a and b are constants. This function is differentiable and limits the value of the

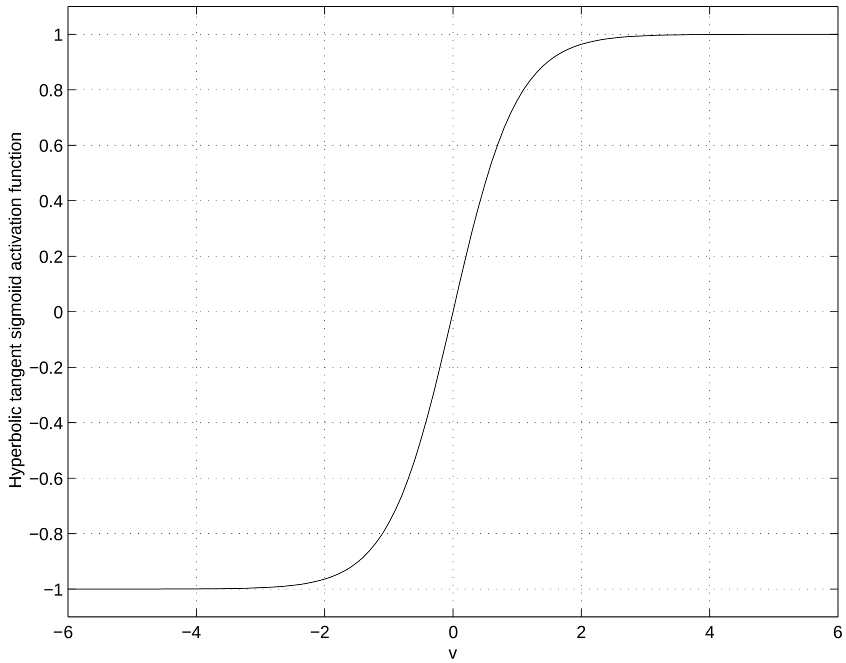


Figure 4-11: Hyperbolic tangent sigmoidal function

outputs between -1 and +1.

Another commonly used function is the log-sigmoid function shown in Figure 4-12 and defined in equation (4.10). It is also differentiable and limits the value of the outputs between 0 and +1.

4. Target values: The target values or desired response d_j for neuron j in the output layer of the multilayer perceptron should be offset by some amount ϵ away from the limiting value of the sigmoid activation function, depending on whether the limiting value is positive or negative. Otherwise the backpropagation algorithm tends to drive the free parameters of the network to infinity, and thereby slow down the learning process by driving the hidden neurons into saturation.

5. Normalizing the inputs: Each input variable should be preprocessed so that its mean value, averaged over the entire training set, is close to zero, or else it is small compared to its standard deviation. The normalization steps are the mean removal, decorrelation and covariance equalization.

Generalization

A neural network is said to generalize well when the input-output mapping computed by the network is inside the user defined allowed error for test data never used in creating or

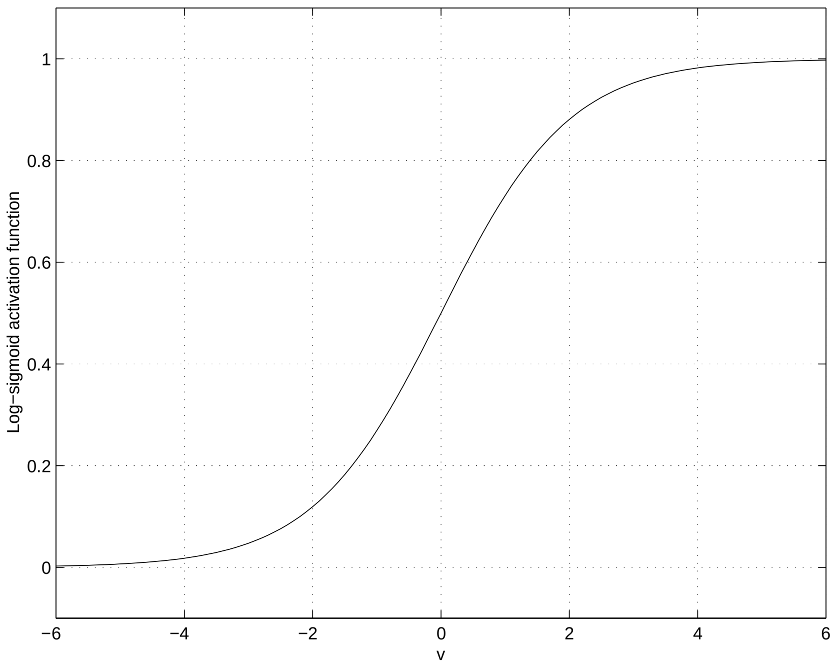


Figure 4-12: Log-sigmoid function

training the net. The learning process may be viewed as a curve fitting problem where the network performs non-linear interpolations of the input data using the continuous sigmoidal functions in the hidden layers. Generalization is influenced by three factors: (1) the size of the training set and how representative it is of the environment of interest, (2) the architecture of the neural network, and (3) the physical complexity of the problem. There is no way to control the third, but for the first two there are usually two ways of improving the generalization [10]:

- The architecture of the network is fixed and the size of the training set is defined for a good generalization to take place.

- The size of the training set is fixed and the best network architecture is to be determined.

In our case the size of the training set given by the number of simulations performed is fixed and the an appropiate definition of the neural network architecture will help to improve the generalization.

Another procedure used to improve the generalization of the network is called *Bayesian regularization* [8] and involves the modification of the performance function defined in equation (4.12). By adding a term consisting in the mean of the sum of squares of the weights and biases, the network will be less likely to overfit and will have a smoother response.

Then equation (4.12) becomes

$$\Upsilon_{reg} = \gamma \frac{1}{N} \sum_{n=1}^{N} e_j^2(n) + (1-\gamma) \frac{1}{n} \sum_{j=1}^{p} e_j^2 \ , \tag{4.27}$$

where γ is the performance ratio and msw is the added term equal to the mean square weights defined by

$$msw = \frac{1}{p} \sum_{j=1}^{p} e_j^2 \ , \tag{4.28}$$

where p is the number of weights in the network.

4.3 Neural network application

The definition of the neural network used to determine the line heating conditions to be applied over a pipe, considers the deformation as input and the heating power and heat source speed as outputs. The spot size dimension was kept fixed and a diameter of 17.0 mm was used. The training set includes the values obtained from the non-linear 3-D FEM analysis and the simplified model analysis for a pipe of the characteristics indicated in section 2.2 using a fixed spot size and varying the heating source power and speed. For the use of the already trained network, the MIT Fabrication Laboratory has developed a system that efficiently represents the deformed cross section of a pipe, identifying the positions and magnitude of the deformations from the true circle. This information will be used in future developements as the input of the trained network, obtaining the necessary heating conditions to reproduce such deformation.

4.3.1 Cross section curve fitting and curvature analysis

The distribution of in-plane shrinkage and angular distortion on a deformed pipe is estimated from the measurement of the deviation of the cross section curve of the manufactured pipe vessel from the true circle. Curve fitting has been performed [11, 2] measuring points as shown in Figure 4-13 to approximately represent the cross section of the deformed pipe. The black squares in Figure 4-13 are the measured points and the yellow green line is the approximated curve of the deformed cross section, while the blue line is the ideal circle. Also a curvature analysis of the fitted cross section curve has been conducted. The curvature plot [2], which consists of segments normals to the fitted curve emanating from a number of points on the fitted curve and whose lengths are proportional to the magnitude of the curvature, is also given in the figure. Figure 4-13 clearly shows the location where the curvature distribution is not exactly a true circle (the inner circle represents curvature of the true circle). Based on this curvature plot, it is possible to locate where to put heat and predict the amount of heat by examining the magnitude of the curvature.

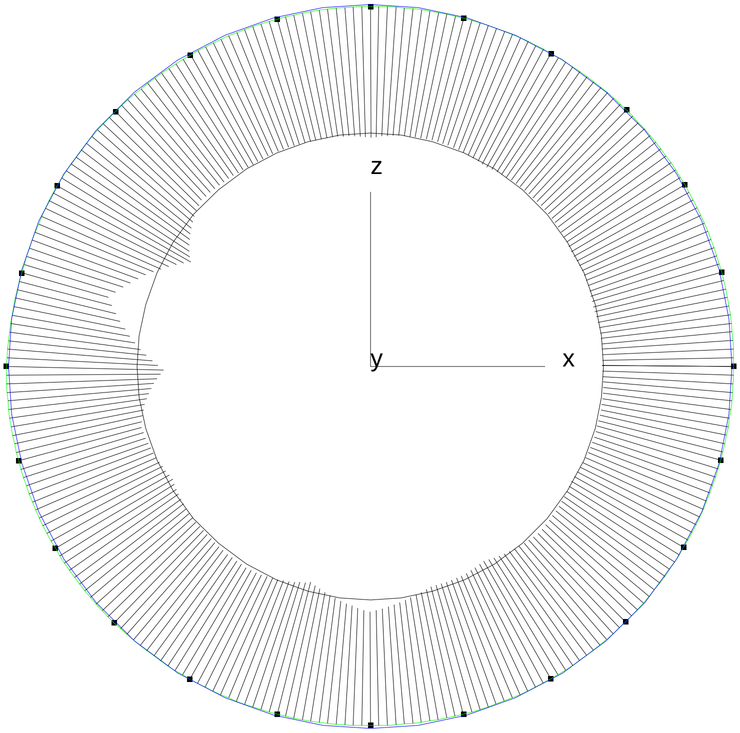


Figure 4-13: Curve fitting of measured points from the cross section of the deformed pipe vessel.

4.3.2 Neural network model

The network was designed based on the topology of a feedforward backpropagation neural network. Various combinations of layers and number of neurons in each layer were tested in order to find the topology that provides a good and computationally efficient result. There is a point where the computational time consumed in the analysis increases significantly compared with the improvement of the network performance. The combination of these properties led to the topology shown in Figure 4-14. It has a single input layer and a multilayer perceptron with two hidden layers and one output layer. The first hidden layer has six neurons, the second hidden layer has five neurons and the output layer has two, corresponding to the number of output vectors. The activation functions used in the hidden layers were logistic sigmoid functions and the one used at the output layer was a linear transfer function.

Data set

The input vector used in the analysis has dimension [224 × 1] and corresponds to the predicted deformation in the z direction for various heating conditions at the free end top surface of the pipe using the methods presented in Chapters 2 and 3. The output vector, also called *target*, has dimension [224 × 2] corresponding to the heating source power and speed required to obtain the input corresponding deformation. The data set was divided in

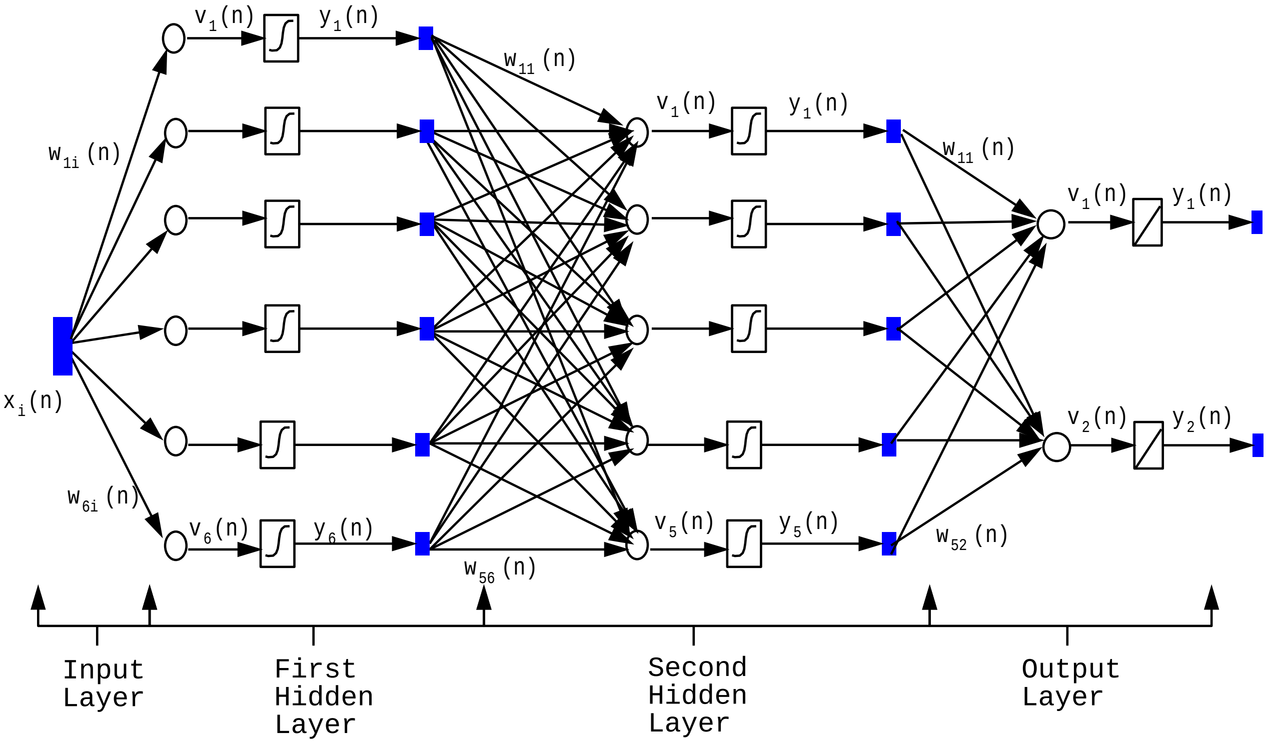


Figure 4-14: Neural network topology used in the analysis

four groups: the first and third were used to train the network (training set), the second to validate the network (validation set) and the fourth to test its work (test set).

Data analysis

The data set was presented to the network using sequential learning update, and its training was performed using the *Levenberg-Marquardt algorithm* [8]. This algorithm uses the same approach as the Newton method, minimizing the quadratic approximation of the cost function $\Upsilon(n)$ around the current point $\mathbf{w}(\mathrm{n})$ at each iteration of the process. The difference between the two methods is that the *Levenberg-Marquardt algorithm* does not compute the Hessian matrix required for the computation of the Newton method. Instead, it uses the relations

$$\mathbf{H} = \mathbf{J}^T\mathbf{J} \ , \tag{4.29}$$

to approximate the Hessian matrix and

$$\mathbf{g} = \mathbf{J}^T\mathbf{e} \ , \tag{4.30}$$

to compute the gradient. $\mathbf{J}$ is the Jacobian matrix which contains the first derivatives of the network errors with respect to the weights and biases and $\mathbf{e}$ is the vector containing the network errors. The computation of the Jacobian is less complex than the computation of the Hessian matrix, making this algorithm faster to converge than the other routines MATLAB incorporates, with a similar degree of accuracy. The learning rate is automatically adapted by the algorithm during the training, adjusting its value to have the optimal convergence of the error function. To improve the generalization of the network we used Bayesian regularization and preprocessing of the data by normalization.

4.3.3 Neural network results

The performance of a neural network can be measured to some extent by the errors on the training, validation and test sets [8]. The averaged square error performance (see equation (4.12)) obtained by evaluating the validation and test sets after the training, was equal to 3.0250×10^{-5}. This low value indicates that the network is well trained and that it provides a good representation of the outputs.

To make a more detailed analysis of the network, it is possible to make a linear regression analysis between the network response and the corresponding targets. Figures 4-15 and 4-16 show the regression curve for both network output parameters with respect to the input. The linear functions defined in each of them are of the form $y = ax + b$, where a corresponds to the slope and b is the $y - axis$ intercept of the best linear regression relating targets to network outputs. The third variable in the Figures corresponds to the correlation coefficient R between the outputs and targets. Its value is 0.894 for the heating power prediction and 0.644 for the heating source speed prediction. This indicates that the network tracks well the heating power output with respect to the corresponding target, but it does not have the same performance when predicting the source speed.

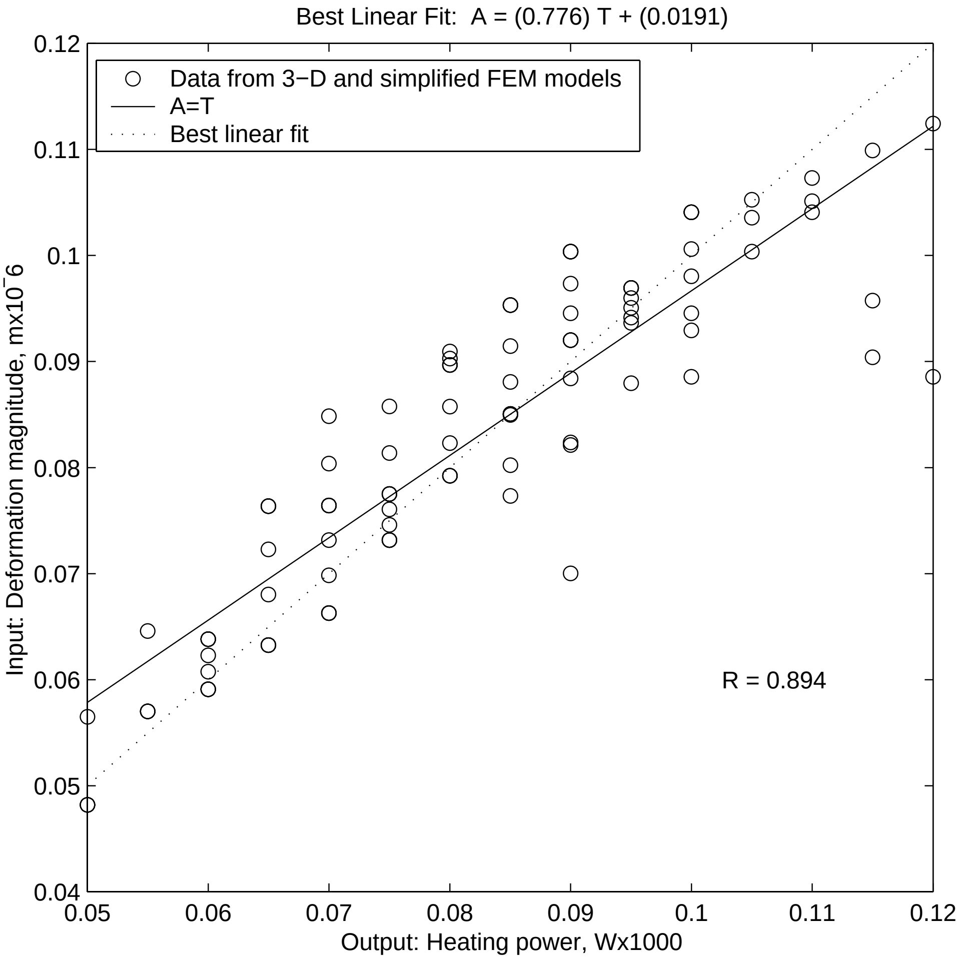


Figure 4-15: Heating power linear regression analysis

This can be seen more clearly when the validation data set is presented as new inputs to the network. Figures 4-17 and 4-18 show the neural network curve fitting for the heating power and the heating source speed given the deformation inputs contained in the validation data set. The first figure shows that there is a good generalization of the output with respect to the target. In the second case, the curve shows the best fit to the presented data. In the latter case the target data is sparse enough to make it difficult to the network to fit an accurate curve. This will produce in some cases an increase in the error of the predicted heating speed with respect to the target values. In this case it is recommended to choose the data more selectively, avoiding duplication of results and making the data less sparse.

When the test data set not included in the training is presented to the already trained network, we can compare its prediction with the target value obtained from the experiments. The results obtained are shown in Table 4.1. The difference between the predicted and the

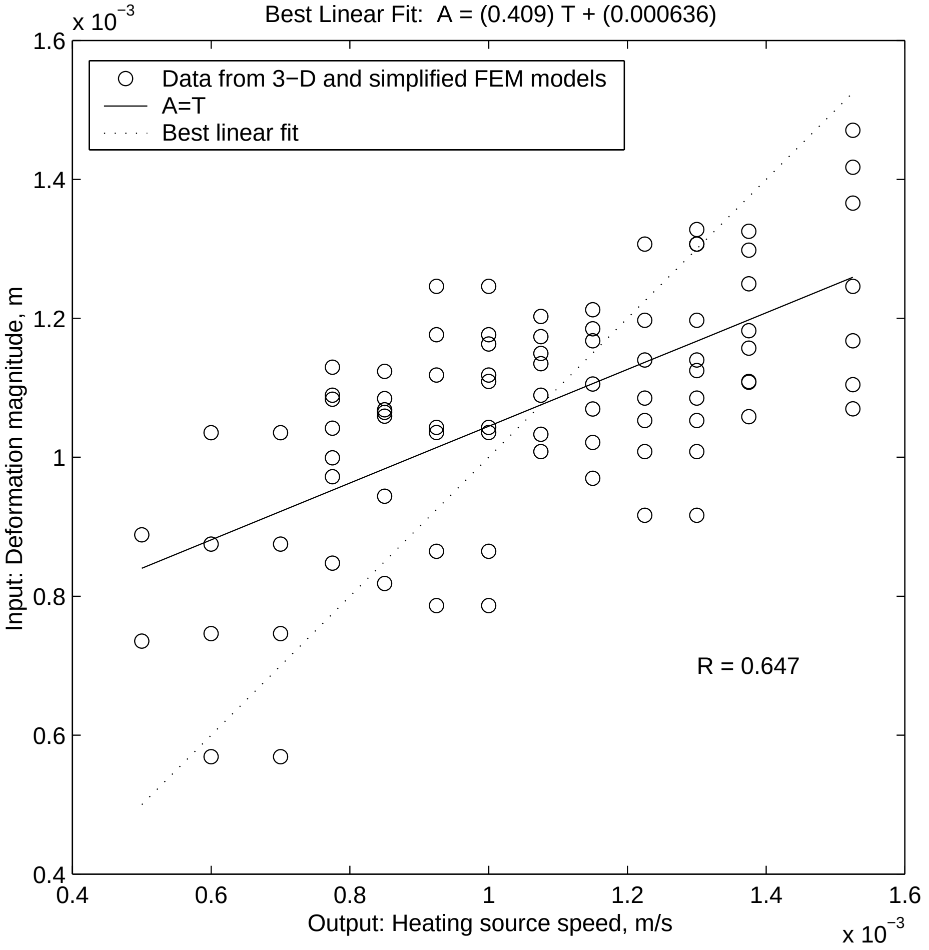


Figure 4-16: Heating source speed linear regression analysis

target heating power required to produce a given deformation is less the 10% in the presented cases. On the other hand, the error in the heating speed prediction increases to almost 30% in some examples. The network will not be highly accurate to reproduce the same heating speed result for some of the presented cases. However, the value delivered using the curve shown in Figure 4-18 will provide a continuous range of results for any given deformation, providing from the statistical standpoint, the most accurate representation of the required heating parameters to obtain a desired input using the data given.

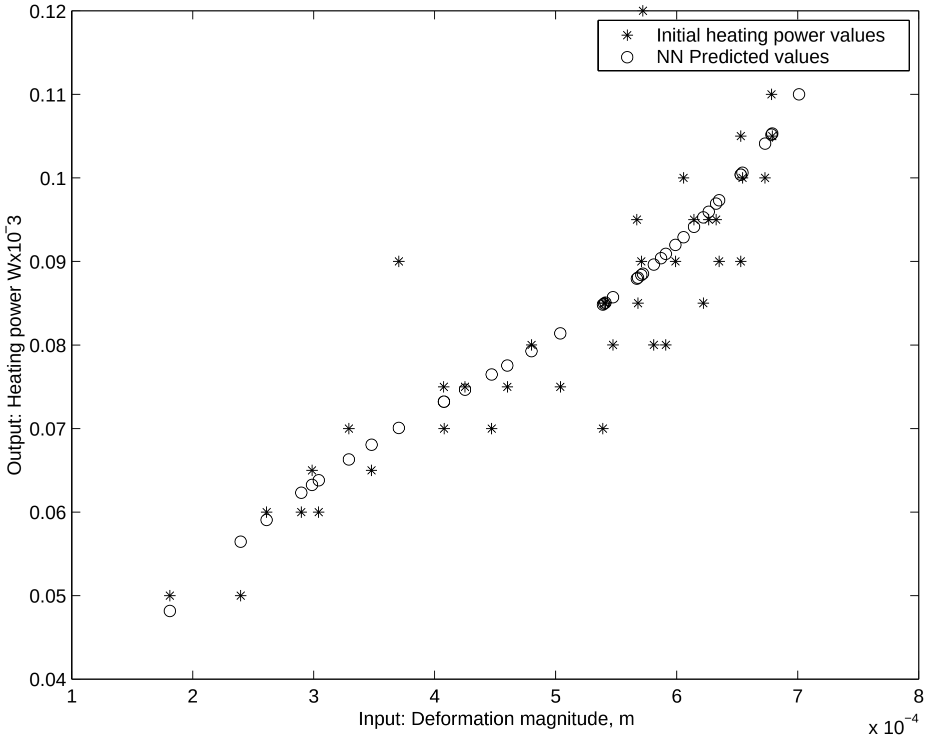


Figure 4-17: Neural network heating power prediction vs heating power original values

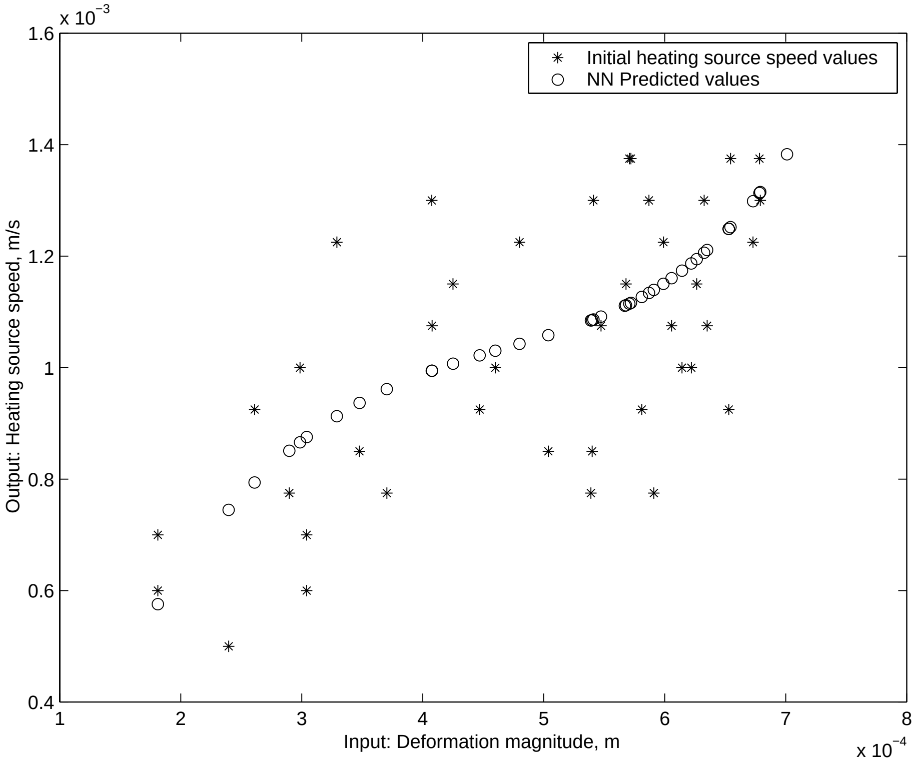


Figure 4-18: Neural network heating source speed vs heating source speed original values

Deformation	Actual	Predicted	Difference	Actual	Predicted	Difference
	power	power		speed	speed	
mm	W	W	%	mm/s	mm/s	%
0.2440	550	567	-3.05	0.7	0.746	-6.59
0.4600	750	779	-3.83	0.925	1.042	-12.74
0.2610	600	587	2.11	1.0	0.786	21.33
0.5950	850	912	-7.30	1.075	1.134	-5.55
0.3683	700	700	0.0	1.15	0.969	15.67
0.6170	900	942	-4.72	1.15	1.167	-1.53
0.4075	750	734	2.07	1.225	1.008	17.69
0.6325	950	967	-1.73	1.225	1.197	2.26
0.4800	800	796	0.54	1.30	1.053	18.99
0.6730	1000	1042	-4.19	1.30	1.306	-0.52
0.4910	850	805	5.28	1.375	1.058	23.02
0.6703	1050	1036	1.30	1.375	1.297	5.60
0.5140	900	825	8.28	1.525	1.069	29.85
0.6890	1100	1078	2.02	1.525	1.366	10.42

Table 4.1: Results obtained after presenting the test data set to the network

Chapter 5

Conclusions and recommendations

5.1 Conclusions and contributions

The major contributions of this thesis are:

1. Development of a full 3-D finite element model for a nonlinear thermo-mechanical analysis of a pipe of any dimension subjected to line heating. With this model we are able to investigate the induced deformation of a pipe due to line heating.

2. Development of a shell element finite element model of a pipe that uses distributed shrinkage forces and bending moments predicted by the semi-analytical thermal model and the simplified mechanical models developed at the MIT Fabrication Laboratory to simulate the deformation of a pipe. With this model the computational time required for a simulation was reduced by 90 %.

3. Development of a neural networks to efficiently predict the heating parameters, namely the heating power and the heating speed, necessary to generate a desired deformation at a pipe's free end.

Based on the models developed and the corresponding simulations performed, we draw the following conclusions:

1. The non-linear FEM model as well as the simplified thermo-mechanical model are able to predict the deformation of a pipe due to line heating.

2. The non-linear FEM thermal model prediction can be considered a more exact representation of the temperature distribution in a pipe when subjected to line heating than the simplified thermal model. The difference obtained between the two models for its highest temperature was of 10.8% in the average. When the temperature results of both models are used as input for the corresponding mechanical models, the difference between the predicted deformation of the two models increases in percentage. Applying a correction factor of 0.9 to the temperature obtained from the simplified thermal

model and then using it as the input in the simplified mechanical model reduces the final deformation difference between models for all the cases analyzed.

3. The neural network is a useful tool to efficiently predict the necessary heating conditions to be applied over a pipe to obtain a desired deformation. The neural network used for this application has small errors and generalizes well in most of the heating cases described.

5.2 Recommendations

The ultimate objective of this research is to develop an automatic system of circularizing the ends of a deformed pipe by laser line heating. Towards this direction we have developed a 3-D non-linear thermo-mechanical FEM model as well as a simplified thermo-mechanical model of a pipe subjected to line heating to predict the deformation for the different cases analyzed. The following topics are recommended for future research.

1. Up to now, we have applied the heat starting from the pipe free-end and moved along $(0, y, r_o)$, $0 \leq y \leq 0.2$. If we start applying the heat from $(0, 0.2, r_o)$ and move towards the free end, we may have a higher temperature at the free end and may have a higher gradient of temperature across the thickness, and hence larger deformation.

2. From previous MIT Fabrication Laboratory research (Yu [22]), we have found that successive line heating on flat plates leads to an almost linear response for the amount of deformation. This means that if several runs are performed over the same region, the final deformation should be proportional to the number of runs. Therefore we need to develop a program to automatically generate an input file for the FEM analysis to handle the ABAQUS command $''RESTART''$.

   Even though the results between the 3-D non-linear FEM model and the simplified model are in good agreement, they need to be verified by experimental data obtained from the actual pipe. This will provide the validation of the numerical data obtained and will provide information to make suitable adjustments of the models.

3. The neural network application is particular for each case and the topology and functions used in this research will not necessarily be the same for different types of problems or sets of data. The next step is to include successive heating runs over the pipe as a new variable in the data set to have a larger deformation when it is needed. This will require redefinition of the neural network topology.

Appendix A

Non-linear thermal analysis ABAQUS input file

The routine included in this appendix corresponds to the ABAQUS input file generated by the heat.inp file to perform the non-linear thermal analysis of a pipe subjected to line heating.

```
HEADING
ABAQUS job created on File name: testh.inp
*
PHYSICAL CONSTANTS, ABSOLUTE ZERO=-273.16
*
PREPRINT, ECHO=NO
RESTART, WRITE, FREQUENCY=50
*
NODE,NSET=ALL
1,        0.0000000,        0.0000000,        -0.3750000
2,        -0.0731589,        0.0000000,        -0.3677945
.
.
7786,        0.1469504,        1.4000000,        -0.3547697
7787,        0.0749147,        1.4000000,        -0.3766215
*
*
ELEMENT, TYPE=DC3D20, ELSET=HEX
1,   1, 3, 339, 337, 2074, 2076, 2412, 2410, 2, 177, 338, 176, 2075, 2250, 2411, 2249,
1569, 1570, 1675, 1674
2,   337, 339, 675, 673, 2410, 2412, 2748, 2746, 338, 513, 674, 512, 2411, 2586, 2747,
2585, 1674, 1675, 1780, 1779
```

```
.
.
1223,    5617, 5587, 5635, 5665, 7690, 7660, 7708, 7738, 5618, 5619, 5666, 5634,
7691, 7692, 7739, 7707, 6187, 6172, 6188, 6203
1224,    5665, 5635, 5683, 5713, 7738, 7708, 7756, 7786, 5666, 5667, 5714, 5682, 7739,
7740, 7787, 7755, 6203, 6188, 6204, 6219
*
*
ELEMENT, TYPE=DC3D15, ELSET=TRI
1225, 13, 15, 183, 2086, 2088, 2256, 14, 103, 102, 2087, 2176, 2175, 1575, 1576, 1628
1226, 13, 183, 349, 2086, 2256, 2422, 102, 263, 182, 2175, 2336, 2255, 1575, 1628, 1680
.
.
1673, 5117, 5175, 5173, 7190, 7248, 7246, 5148, 5174, 5147, 7221, 7247, 7220, 6021, 6038,
6037
1674, 5119, 5175, 5117, 7192, 7248, 7190, 5149, 5148, 5118, 7222, 7221, 7191, 6022, 6038,
6021
*
* hex
*
SOLID SECTION, ELSET=HEX, MATERIAL=MILDSTEE
1.,
*
* tri
*
SOLID SECTION, ELSET=TRI, MATERIAL=MILDSTEE
1.,
*
*
* mildsteel
*
MATERIAL, NAME=MILDSTEE
*
DENSITY
7800,
*
CONDUCTIVITY, TYPE=ISO
51.9, 0.
51.1, 100.
49.0, 200.
```

```
46.1, 300.
42.7, 400.
39.4, 500.
35.6, 600.
31.8, 700.
26.0, 800.
27.2, 1000.
29.7, 1500.
*
SPECIFIC HEAT
450.0, 0.
486.0, 75.
519.0, 175.
532.0, 225.
557.0, 275.
574.0, 325.
599.0, 375.
662.0, 475.
749.0, 575.
846.0, 675.
1432.0, 725.
950.0, 775.
400.0, 1500.
*
* step 1
*
*initial_cond
*
INITIAL CONDITIONS, TYPE=TEMPERATURE
ALL, 21.1
*
*
USER SUBROUTINE
SUBROUTINE DFLUX(FLUX,TEMP,KSTEP,KINC,TIME,NOEL,NPT,COORDS,JLTYP)
include 'aba_param.inc'
DIMENSION FLUX(2),TIME(2),COORDS(3)
REAL*8 R1,X1,X2,RP,PHI
X2=COORDS(3)
R1=(COORDS(2)- 0.001600*TIME(2))
X1=COORDS(1)
```

```
RP = SQRT(X1*X1 + X2*X2)
PHI = ATAN2(X2,X1)
X1=RP*ABS( 1.570796 - PHI)
IF(RP.GE. 0.383037568) THEN
IF(TIME(2).LE. 312.50000) THEN
IF((R1*R1+X1*X1).LT. 0.000161290000) THEN
FLUX(1)=3677986.8808*2.0*(RP- 0.383047568)
FLUX(1)=FLUX(1)*DEXP(-13145.660216*(R1*R1+X1*X1))
FLUX(1)=FLUX(1)/ 0.000952432/ 0.000952432
ELSE IF ((R1*R1+X1*X1).LT. 0.000645160000) THEN
R=SQRT(R1*R1+X1*X1)
FLUX(1)= 171275.3426*4.0*(RP- 0.3830)*( 0.025400000000-R)
FLUX(1)=FLUX(1)/ 0.000952432/ 0.000952432/ 0.012700000000
END IF
ELSE
FLUX(1)=0.0
END IF
END IF
RETURN
END
*
STEP, AMPLITUDE=STEP, INC=3000
HEAT TRANSFER, END=PERIOD, DELTMX=50.
0.1,         600,         0.0005,         20.
*
monitor,node=6298, dof=11
*
FILM PROPERTY, NAME=FILMUP
4.4468 , 100.0
5.1405 , 200.0
5.3252 , 300.0
5.5800 , 400.0
5.6701 , 500.0
6.2027 , 600.0
6.5913 , 700.0
6.7781 , 800.0
7.0061 , 900.0
7.2161 , 1000.0
7.4101 , 1100.0
```

```
7.5911 , 1200.0
*
FILM PROPERTY, NAME=FILMDOWN
1.8028, 100.0
2.1258, 200.0
2.2583, 300.0
2.3303, 400.0
2.4723, 500.0
2.6202, 600.0
2.8011, 700.0
2.8881, 800.0
2.9515, 900.0
3.0093, 1000.0
3.0626, 1100.0
3.1120, 1200.0
*
* conv
*
FILM, OP=NEW
*
1, F1, 21.1, FILMDOWN
2, F1, 21.1, FILMDOWN
.
.
408, F1, 21.1, FILMDOWN
817, F2, 21.1, FILMUP
818, F2, 21.1, FILMUP
.
.
1223, F2, 21.1, FILMUP
1224, F2, 21.1, FILMUP
1225, F1, 21.1, FILMDOWN
1226, F1, 21.1, FILMDOWN
.
.
1373, F1, 21.1, FILMDOWN
1374, F1, 21.1, FILMDOWN
1525, F2, 21.1, FILMUP
1526, F2, 21.1, FILMUP
.
```

```
.
1673, F2, 21.1, FILMUP
1674, F2, 21.1, FILMUP
*
RADIATE, OP=NEW
*
1, R1, 21.1, 0.000000048186
2, R1, 21.1, 0.000000048186
.
.
407, R1, 21.1, 0.000000048186
408, R1, 21.1, 0.000000048186
817, R2, 21.1, 0.000000048186
818, R2, 21.1, 0.000000048186
.
.
1223, R2, 21.1, 0.000000048186
1224, R2, 21.1, 0.000000048186
1225, R1, 21.1, 0.000000048186
1226, R1, 21.1, 0.000000048186
.
.
1373, R1, 21.1, 0.000000048186
1374, R1, 21.1, 0.000000048186
1525, R2, 21.1, 0.000000048186
1526, R2, 21.1, 0.000000048186
.
.
1673, R2, 21.1, 0.000000048186
1674, R2, 21.1, 0.000000048186
*
BOUNDARY, OP=NEW
CFLUX, OP=NEW
DFLUX, OP=NEW
HEX,BFNU
TRI, BFNU
*
NODE PRINT, FREQ=50
NT
NODE FILE, FREQ=5
```

```
NT
EL PRINT, POSITION=INTEGRATION POINT, FREQ=0
*
EL FILE, POSITION=INTEGRATION POINT, FREQ=0
*
EL PRINT, POSITION=NODES, FREQ=0
*
EL FILE, POSITION=NODES, FREQ=0
*
EL PRINT, POSITION=CENTROIDAL, FREQ=0
*
EL FILE, POSITION=CENTROIDAL, FREQ=0
*
EL PRINT, POSITION=AVERAGED AT NODES, FREQ=0
*
EL FILE, POSITION=AVERAGED AT NODES, FREQ=0
*
MODAL PRINT, FREQ=99999
*
MODAL FILE, FREQ=99999
*
PRINT, FREQ=20
END STEP
```

Appendix B

Non-linear mechanical analysis ABAQUS input file

The routine included in this appendix corresponds to the ABAQUS input file generated by the mech.inp file to perform the non-linear mechanical analysis of a pipe subjected to line heating.

```
HEADING
ABAQUS job created on File name: mechanical.inp
*
PHYSICAL CONSTANTS, ABSOLUTE ZERO=-273.16
*
PREPRINT, ECHO=NO
RESTART, WRITE, FREQUENCY=50
*
NODE,NSET=ALL
1,        0.0000000,        0.0000000,        -0.3750000
2,        -0.0731589,        0.0000000,        -0.3677945
.
.
7786,        0.1469504,        1.4000000,        -0.3547697
7787,        0.0749147,        1.4000000,        -0.3766215
*
*
ELEMENT, TYPE=DC3D20, ELSET=HEX
1,   1, 3, 339, 337, 2074, 2076, 2412, 2410, 2, 177, 338, 176, 2075, 2250, 2411, 2249,
1569, 1570, 1675, 1674
2,   337, 339, 675, 673, 2410, 2412, 2748, 2746, 338, 513, 674, 512, 2411, 2586, 2747,
2585, 1674, 1675, 1780, 1779
```

```
.
.
1223,    5617, 5587, 5635, 5665, 7690, 7660, 7708, 7738, 5618, 5619, 5666, 5634,
7691, 7692, 7739, 7707, 6187, 6172, 6188, 6203
1224,    5665, 5635, 5683, 5713, 7738, 7708, 7756, 7786, 5666, 5667, 5714, 5682, 7739,
7740, 7787, 7755, 6203, 6188, 6204, 6219
*
*
ELEMENT, TYPE=DC3D15, ELSET=TRI
1225,    13, 15, 183, 2086, 2088, 2256, 14, 103, 102, 2087, 2176, 2175, 1575, 1576, 1628
1226,    13, 183, 349, 2086, 2256, 2422, 102, 263, 182, 2175, 2336, 2255, 1575, 1628, 1680
.
.
1673,    5117, 5175, 5173, 7190, 7248, 7246, 5148, 5174, 5147, 7221, 7247, 7220, 6021, 6038,
6037
1674,    5119, 5175, 5117, 7192, 7248, 7190, 5149, 5148, 5118, 7222, 7221, 7191, 6022, 6038,
6021
*
* hex
*
SOLID SECTION, ELSET=HEX, MATERIAL=MILDSTEE
1.,
*
* tri
*
SOLID SECTION, ELSET=TRI, MATERIAL=MILDSTEE
1.,
*
*
* mildsteel
*
MATERIAL, NAME=MILDSTEE
*
DENSITY
7800,
*
ELASTIC, TYPE=ISO
200E+9, 0.3, 0.
200E+9, 0.3, 100.
200E+9, 0.3, 300.
```

```
150E+9, 0.3, 450.
109.62E+9, 0.3, 550.
88E+9,0.3, 600.
62E+9, 0.3, 720.
20E+9, 0.3, 800.
2E+9, 0.3, 1200.
2E+9, 0.3, 1550.
*
PLASTIC
290E+6, 0.0, 0.
314E+6, 1.0, 0.
260E+6, 0.0, 100.
349E+6, 1.0, 100.
200E+6, 0.0, 300.
440E+6, 1.0, 300.
150E+6, 0.0, 450.
460E+6, 1.0, 450.
120E+6, 0.0, 550.
410E+6, 1.0, 550.
110E+6, 0.0, 600.
330E+6, 1.0, 600.
9.8E+6, 0.0, 720.
58.8E+6, 1.0, 720.
9.8E+6, 0.0, 800.
58.78E+6, 1.0, 800.
9.8E+6, 0.0, 1200.
58.78E+6, 1.0, 1200.
0.98E+6, 0.0, 1550.
1E+6, 1.0, 1550.
*
* *EXPANSION, TYPE=ISO, ZERO=0.
10E-6, 0.
11E-6, 100.
12E-6, 300.
13E-6, 450.
14E-6, 550.
14E-6, 800.
```

```
15E-6,1200.
*
* step 1
*
*initial_cond
*
INITIAL CONDITIONS, TYPE=TEMPERATURE
ALL, 21.1
*
INITIAL CONDITIONS, TYPE=STRESS, USER 1,    0.000e+00, 0.000e+00, 0.000e+00,
0.000e+00, 0.000e+00, 0.000e+00 2,    0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
0.000e+00 ·
.
1673,    0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00 1674,    0.000e+00,
0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00 **
USER SUBROUTINE
SUBROUTINE SIGINI(SIGMA,COORDS,NTENS,NCRDS,NOEL,NPT,LAYER,KSPT,
* LREBAR,REBARN)
INCLUDE 'ABA_PARAM.INC'
DIMENSION SIGMA(NTENS),COORDS(NCRDS)
CHARACTER*8 REBARN
DIMENSION ST(6, 7787)
DIMENSION XI(20),ETA(20),ZETA(20),XP(20),YP(20),ZP(20)
INTEGER NTOT,NTOT0,NELEB0,NELET0,NELEBT,NELETT,NZ
INTEGER ELEMB(20),ELEMT(15)
INTEGER ELES8(12, 408),ELES6(9, 150)
DATA ((ST(I,J),I=1,6),J= 1, 100)/
0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
.
.
0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000

DATA ((ST(I,J),I=1,6),J= 7701, 7787)/ * 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
.
```

```
.
0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000
/
DATA ((ELES8(I,J),I=1,12),J= 1, 100)/
1, 3, 339, 337, 2, 177, 338, 176, 1569, 1570, 1675, 1674,
337, 339, 675, 673, 338, 513, 674, 512, 1674, 1675, 1780, 1779,
.
.
1471, 1441, 1489, 1519, 1472, 1473, 1520, 1488, 2041, 2026, 2042, 2057,
1519, 1489, 1537, 1567, 1520, 1521, 1568, 1536, 2057, 2042, 2058, 2073
/
DATA ((ELES6(I,J),I=1,9),J= 1, 100)/
13, 15, 183, 14, 103, 102, 1575, 1576, 1628,
13, 183, 349, 102, 263, 182, 1575, 1628, 1680,
.
.
971, 1029, 1027, 1002, 1028, 1001, 1875, 1892, 1891,
973, 1029, 971, 1003, 1002, 972, 1876, 1892, 1875
/
NTOT0= 2073
NZ= 3
NTOT=NTOT0*(NZ+1)
NELEB0= 408
NELET0= 150
NELEBT=NELEB0*NZ
NELETT=NELET0*NZ
IF(NOEL.LE.NELEBT) THEN
NZ=(NOEL-1)/NELEB0
ELEMB(1)=ELES8(1,NOEL-NZ*NELEB0)+NZ*NTOT0
ELEMB(2)=ELES8(2,NOEL-NZ*NELEB0)+NZ*NTOT0
ELEMB(3)=ELES8(3,NOEL-NZ*NELEB0)+NZ*NTOT0
ELEMB(4)=ELES8(4,NOEL-NZ*NELEB0)+NZ*NTOT0
ELEMB(5)=ELES8(1,NOEL-NZ*NELEB0)+(NZ+1)*NTOT0
ELEMB(6)=ELES8(2,NOEL-NZ*NELEB0)+(NZ+1)*NTOT0
ELEMB(7)=ELES8(3,NOEL-NZ*NELEB0)+(NZ+1)*NTOT0
ELEMB(8)=ELES8(4,NOEL-NZ*NELEB0)+(NZ+1)*NTOT0
ELEMB(9)=ELES8(5,NOEL-NZ*NELEB0)+NZ*NTOT0
ELEMB(10)=ELES8(6,NOEL-NZ*NELEB0)+NZ*NTOT0
ELEMB(11)=ELES8(7,NOEL-NZ*NELEB0)+NZ*NTOT0
ELEMB(12)=ELES8(8,NOEL-NZ*NELEB0)+NZ*NTOT0
ELEMB(13)=ELES8(5,NOEL-NZ*NELEB0)+(NZ+1)*NTOT0
ELEMB(14)=ELES8(6,NOEL-NZ*NELEB0)+(NZ+1)*NTOT0
ELEMB(15)=ELES8(7,NOEL-NZ*NELEB0)+(NZ+1)*NTOT0
ELEMB(16)=ELES8(8,NOEL-NZ*NELEB0)+(NZ+1)*NTOT0
ELEMB(17)=ELES8(9,NOEL-NZ*NELEB0)+NZ*NTOT0
ELEMB(18)=ELES8(10,NOEL-NZ*NELEB0)+NZ*NTOT0
ELEMB(19)=ELES8(11,NOEL-NZ*NELEB0)+NZ*NTOT0
ELEMB(20)=ELES8(12,NOEL-NZ*NELEB0)+NZ*NTOT0
DO 10 I=1,6
SIGMA(I)=0
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(1))*(-1.0/8.0*(1.0+0.77459)
* *(1.0+0.77459)*(1.0+0.77459)*(2.0-0.77459-0.77459-0.77459))
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(2))*(-1.0/8.0*(1.0+0.77459)
* *(1.0+0.77459)*(1.0+0.77459)*(2.0-0.77459-0.77459-0.77459))
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(3))*(-1.0/8.0*(1.0+0.77459)
* *(1.0+0.77459)*(1.0+0.77459)*(2.0-0.77459-0.77459-0.77459))
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(4))*(-1.0/8.0*(1.0+0.77459)
* *(1.0+0.77459)*(1.0+0.77459)*(2.0-0.77459-0.77459-0.77459))
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(5))*(-1.0/8.0*(1.0+0.77459)
* *(1.0+0.77459)*(1.0+0.77459)*(2.0-0.77459-0.77459-0.77459))
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(6))*(-1.0/8.0*(1.0+0.77459)
* *(1.0+0.77459)*(1.0+0.77459)*(2.0-0.77459-0.77459-0.77459))
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(7))*(-1.0/8.0*(1.0+0.77459)
* *(1.0+0.77459)*(1.0+0.77459)*(2.0-0.77459-0.77459-0.77459))
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(8))*(-1.0/8.0*(1.0+0.77459)
* *(1.0+0.77459)*(1.0+0.77459)*(2.0-0.77459-0.77459-0.77459))
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(9))*1.0/4.0*(1.0+0.77459)*(1.0+0.77459)
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(10))*1.0/4.0*(1.0+0.77459)*(1.0+0.77459)
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(11))*1.0/4.0*(1.0+0.77459)*(1.0+0.77459)
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(12))*1.0/4.0*(1.0+0.77459)*(1.0+0.77459)
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(13))*1.0/4.0*(1.0+0.77459)*(1.0+0.77459)
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(14))*1.0/4.0*(1.0+0.77459)*(1.0+0.77459)
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(15))*1.0/4.0*(1.0+0.77459)*(1.0+0.77459)
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(16))*1.0/4.0*(1.0+0.77459)*(1.0+0.77459)
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(17))*1.0/4.0*(1.0+0.77459)*(1.0+0.77459)
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(18))*1.0/4.0*(1.0+0.77459)*(1.0+0.77459)
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(19))*1.0/4.0*(1.0+0.77459)*(1.0+0.77459)
SIGMA(I)=SIGMA(I)+ST(I,ELEMB(20))*1.0/4.0*(1.0+0.77459)*(1.0+0.77459)
10 CONTINUE
```

```
ELSE
NZ=(NOEL-1-NELEBT)/NELET0
ELEMT(1)=ELES6(1,NOEL-NELEBT-NZ*NELET0)+NZ*NTOT0
ELEMT(2)=ELES6(2,NOEL-NELEBT-NZ*NELET0)+NZ*NTOT0
ELEMT(3)=ELES6(3,NOEL-NELEBT-NZ*NELET0)+NZ*NTOT0
ELEMT(4)=ELES6(1,NOEL-NELEBT-NZ*NELET0)+(NZ+1)*NTOT0
ELEMT(5)=ELES6(2,NOEL-NELEBT-NZ*NELET0)+(NZ+1)*NTOT0
ELEMT(6)=ELES6(3,NOEL-NELEBT-NZ*NELET0)+(NZ+1)*NTOT0
ELEMT(7)=ELES6(4,NOEL-NELEBT-NZ*NELET0)+NZ*NTOT0
ELEMT(8)=ELES6(5,NOEL-NELEBT-NZ*NELET0)+NZ*NTOT0
ELEMT(9)=ELES6(6,NOEL-NELEBT-NZ*NELET0)+NZ*NTOT0
ELEMT(10)=ELES6(4,NOEL-NELEBT-NZ*NELET0)+(NZ+1)*NTOT0
ELEMT(11)=ELES6(5,NOEL-NELEBT-NZ*NELET0)+(NZ+1)*NTOT0
ELEMT(12)=ELES6(6,NOEL-NELEBT-NZ*NELET0)+(NZ+1)*NTOT0
ELEMT(13)=ELES6(7,NOEL-NELEBT-NZ*NELET0)+NZ*NTOT0
ELEMT(14)=ELES6(8,NOEL-NELEBT-NZ*NELET0)+NZ*NTOT0
ELEMT(15)=ELES6(9,NOEL-NELEBT-NZ*NELET0)+NZ*NTOT0
DO 20 I=1,6
IF (NPT.EQ.1) THEN
SIGMA(I)=0.261401181296201*ST(I,ELEMT(1))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(2))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(3))
SIGMA(I)=SIGMA(I)+(0.261401181296201)*ST(I,ELEMT(4))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(5))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(6))
SIGMA(I)=SIGMA(I)+(0.784487892046003)*ST(I,ELEMT(7))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(8))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(9))
SIGMA(I)=SIGMA(I)+(0.784487892046003)*ST(I,ELEMT(10))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(11))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(12))
SIGMA(I)=SIGMA(I)+(0.797426985353000)*ST(I,ELEMT(13))
SIGMA(I)=SIGMA(I)+(0.797426985353100)*ST(I,ELEMT(14))
SIGMA(I)=SIGMA(I)+(0.797426985353100)*ST(I,ELEMT(15))
ELSE IF (NPT.EQ.2) THEN
SIGMA(I)=0.261401181296375*ST(I,ELEMT(1))
SIGMA(I)=SIGMA(I)+(0.261401181296201)*ST(I,ELEMT(2))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(3))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(4))
SIGMA(I)=SIGMA(I)+(0.261401181296201)*ST(I,ELEMT(5))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(6))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(7))
SIGMA(I)=SIGMA(I)+(0.784487892046003)*ST(I,ELEMT(8))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(9))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(10))
SIGMA(I)=SIGMA(I)+(0.784487892046003)*ST(I,ELEMT(11))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(12))
SIGMA(I)=SIGMA(I)+(0.797426985353100)*ST(I,ELEMT(13))
SIGMA(I)=SIGMA(I)+(0.797426985353000)*ST(I,ELEMT(14))
SIGMA(I)=SIGMA(I)+(0.797426985353100)*ST(I,ELEMT(15))
ELSE IF (NPT.EQ.3) THEN
SIGMA(I)=0.261401181296375*ST(I,ELEMT(1))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(2))
SIGMA(I)=SIGMA(I)+(0.261401181296201)*ST(I,ELEMT(3))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(4))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(5))
SIGMA(I)=SIGMA(I)+(0.261401181296201)*ST(I,ELEMT(6))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(7))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(8))
SIGMA(I)=SIGMA(I)+(0.784487892046003)*ST(I,ELEMT(9))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(10))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(11))
SIGMA(I)=SIGMA(I)+(0.784487892046003)*ST(I,ELEMT(12))
SIGMA(I)=SIGMA(I)+(0.797426985353100)*ST(I,ELEMT(13))
SIGMA(I)=SIGMA(I)+(0.797426985353100)*ST(I,ELEMT(14))
SIGMA(I)=SIGMA(I)+(0.797426985353000)*ST(I,ELEMT(15))
ELSE IF (NPT.EQ.4) THEN
SIGMA(I)=0.261401181296201*ST(I,ELEMT(1))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(2))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(3))
SIGMA(I)=SIGMA(I)+(0.261401181296201)*ST(I,ELEMT(4))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(5))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(6))
SIGMA(I)=SIGMA(I)+(0.784487892046003)*ST(I,ELEMT(7))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(8))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(9))
SIGMA(I)=SIGMA(I)+(0.784487892046003)*ST(I,ELEMT(10))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(11))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(12))
SIGMA(I)=SIGMA(I)+(0.797426985353000)*ST(I,ELEMT(13))
SIGMA(I)=SIGMA(I)+(0.797426985353100)*ST(I,ELEMT(14))
SIGMA(I)=SIGMA(I)+(0.797426985353100)*ST(I,ELEMT(15))
ELSE IF (NPT.EQ.5) THEN
SIGMA(I)=0.261401181296375*ST(I,ELEMT(1))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(2))
SIGMA(I)=SIGMA(I)+(0.261401181296201)*ST(I,ELEMT(3))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(4))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(5))
SIGMA(I)=SIGMA(I)+(0.261401181296201)*ST(I,ELEMT(6))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(7))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(8))
SIGMA(I)=SIGMA(I)+(0.784487892046003)*ST(I,ELEMT(9))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(10))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(11))
SIGMA(I)=SIGMA(I)+(0.784487892046003)*ST(I,ELEMT(12))
SIGMA(I)=SIGMA(I)+(0.797426985353100)*ST(I,ELEMT(13))
SIGMA(I)=SIGMA(I)+(0.797426985353100)*ST(I,ELEMT(14))
SIGMA(I)=SIGMA(I)+(0.797426985353000)*ST(I,ELEMT(15))
ELSE IF (NPT.EQ.6) THEN
SIGMA(I)=0.261401181296375*ST(I,ELEMT(1))
SIGMA(I)=SIGMA(I)+(0.261401181296201)*ST(I,ELEMT(2))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(3))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(4))
SIGMA(I)=SIGMA(I)+(0.261401181296201)*ST(I,ELEMT(5))
SIGMA(I)=SIGMA(I)+(0.261401181296375)*ST(I,ELEMT(6))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(7))
SIGMA(I)=SIGMA(I)+(0.784487892046003)*ST(I,ELEMT(8))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(9))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(10))
SIGMA(I)=SIGMA(I)+(0.784487892046003)*ST(I,ELEMT(11))
SIGMA(I)=SIGMA(I)+(0.784487892046002)*ST(I,ELEMT(12))
SIGMA(I)=SIGMA(I)+(0.797426985353100)*ST(I,ELEMT(13))
SIGMA(I)=SIGMA(I)+(0.797426985353000)*ST(I,ELEMT(14))
SIGMA(I)=SIGMA(I)+(0.797426985353100)*ST(I,ELEMT(15))
END IF
20 CONTINUE
END IF
RETURN
```

```
END
*
STEP, AMPLITUDE=STEP, INC=2000
STATIC
0.001, 600, 0.0001, 20.
*
monitor, node=29,dof=3
CONTROLS,ANALYSIS=DISCONTINUOUS
CONTROLS,PARAMETERS=LINE SEARCH
*
BOUNDARY, OP=NEW
1, 1,, 0.
29, 1,, 0.
.
.
7756, 1,, 0.
7756, 1, 3, 0.
7772, 1,, 0.
7772, 1, 2, 0.
* read temperature from .fil file
*
TEMPERATURE, FILE=testh.inp
*
CLOAD, OP=NEW
DLOAD, OP=NEW
*
NODE PRINT, FREQ=20
U
NODE FILE, FREQ=50
U
*
EL PRINT, POSITION=INTEGRATION POINT, FREQ=20
S
E
*
EL FILE, POSITION=INTEGRATION POINT, FREQ=0
S
```

```
E
*
EL PRINT, POSITION=NODES, FREQ=0
*
EL FILE, POSITION=NODES, FREQ=0
*
EL PRINT, POSITION=CENTROIDAL, FREQ=20
S
E
*
EL FILE, POSITION=CENTROIDAL, FREQ=40
S
E
*
EL PRINT, POSITION=AVERAGED AT NODES, FREQ=20
S
E
*
EL FILE, POSITION=AVERAGED AT NODES, FREQ=0
*
MODAL PRINT, FREQ=99999
*
MODAL FILE, FREQ=99999
*
ENERGY PRINT, FREQ=0
*
ENERGY FILE, FREQ=0
*
PRINT, FREQ=50
*
END STEP
```

Appendix C

Shell elements FEM analysis ABAQUS input file

The routine included in this appendix corresponds to the ABAQUS input file generated by the simp.inp file to perform the shell FEM analysis of a pipe subjected to line heating.

```
HEADING
ABAQUS job created on File name: simplified.inp
*
PREPRINT, ECHO=NO
RESTART, WRITE, FREQUENCY=50
*
NODE,NSET=ALL
1,        0.0000000,        0.0000000,        -0.3807146
2,        -0.0731589,        0.0000000,        -0.3733993
.
.
3640,        0.1456932,        1.4000000,        -0.3517344
3641,        0.0742737,        1.4000000,        -0.3733993
*
*
ELEMENT, TYPE=S8R, ELSET=HEX
409,   2074, 2076, 2412, 2410, 2075, 2250, 2411, 2249
410,   2410, 2412, 2748, 2746, 2411, 2586, 2747, 2585
.
.
815,   3544, 3514, 3562, 3592, 3545, 3546, 3593, 3561
```

```
816,   3592, 3562, 3610, 3640, 3593, 3594, 3641, 3609
*
ELEMENT, TYPE=STRI65, ELSET=TRI
1375,    2086, 2088, 2256, 2087, 2176, 2175
1376,    2086, 2256, 2422, 2175, 2336, 2255
.
.
1523,    3044, 3102, 3100, 3075, 3101, 3074
1524,    3046, 3102, 3044, 3076, 3075, 3045
*
*
NODAL THICKNESS
ALL, 0.009
*
* hex
*
SHELL GENERAL SECTION, ELSET=HEX, POISSON=0.3, MATERIAL=MILDSTEE,
NODAL THICKNESS
1.,
*
* tri
*
SHELL GENERAL SECTION, ELSET=TRI, POISSON=0.3, MATERIAL=MILDSTEE,
NODAL THICKNESS
1.,
*
* mildsteel
*
MATERIAL, NAME=MILDSTEE
*
ELASTIC, TYPE=ISO
2E+11, 0.3
DENSITY
7800,
*
* step 1
*
*initial_cond
*
INITIAL CONDITIONS, TYPE=STRESS
```

```
409,   0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00
410,    0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00
.
.
1523,   0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00
1524,    0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00
*
STEP
STATIC
*
monitor, node=2102, dof=3
*
BOUNDARY, OP=NEW
2074, 1,, 0.
2102, 1,, 0.
.
.
3610, 1, 3, 0.
3626, 1, 2, 0.
*
CLOAD, OP=NEW
2103, 1, -24208.7418810
2103, 5, -68.1445026
.
.
2973, 2, -7434.4704166
2973, 4, 20.9270804
*
DLOAD, OP=NEW
*
NODE PRINT, FREQ=1
U
NODE FILE, FREQ=1
U
*
EL PRINT, POSITION=INTEGRATION POINT, FREQ=1
S
E
*
EL FILE, POSITION=INTEGRATION POINT, FREQ=0
```

```
S
E
*
EL PRINT, POSITION=NODES, FREQ=0
*
EL FILE, POSITION=NODES, FREQ=0
*
EL PRINT, POSITION=CENTROIDAL, FREQ=1
S
E
*
EL FILE, POSITION=CENTROIDAL, FREQ=1
S
E
*
EL PRINT, POSITION=AVERAGED AT NODES, FREQ=1
S
E
*
EL FILE, POSITION=AVERAGED AT NODES, FREQ=0
*
MODAL PRINT, FREQ=99999
*
MODAL FILE, FREQ=99999
*
ENERGY PRINT, FREQ=0
*
ENERGY FILE, FREQ=0
*
PRINT, FREQ=1
*
END STEP
```

Appendix D

Neural Network MATLAB input file

The routine included in this appendix corresponds to the MATLAB input file to perform the neural network simulation. It considers the deformation as the input vector and the heating source power and speed as the output vector.

```
%The neural network definition
clear
%Including the input and output vectors contained in the vector.m file
pn= input matrix
tn= output matrix

    [R,Q]=size(pn);

% Definition of the training, validation and test data sets
iitst=2:4:Q;
iival=4:4:Q;
iitr=[1:4:Q 3:4:Q];

v.P=pn(:,iival); v.T=tn(:,iival); % Validation data set
t.P=pn(:,iitst); t.V=tn(:,iitst); % Test data set
ptr=pn(:,iitr); ttr=tn(:,iitr); % Training data set

% Neural network definition
net = newff([0 0.1],[6,5,2],’logsig’,’logsig’,’purelin’,’trainbr’);
% NN characteristics
% newff=Feedforward backpropagating neural network
%[0 0.1]=Range of the data in the input vector
```

```
% One input layer
% [6,5,2]=Two hidden layers. Six neurons in the first and five in the second. One output
layer with two neurons
% 'logsig'=Log sigmoid activation function in both hidden layers
% 'purelin'=Linear activation function in the output layer
% 'trainbr'=Bayesian regularization function using Levenberg-Marquardt training

net.trainParam.show=50; net.trainParam.epochs=1600;
% Neural network initialization with function 'init'
net=init(net);

% Training using the training data set with function 'train'
net=train(net,ptr,ttr);

% Simulation of the network performance using all the data sets with function 'sim'
y=sim (net,v.P);
y1=sim(net,t.P);
y2=sim(net,ptr);

% Plotting of the NN prediction
figure
plot(v.P,v.T(1,:),'*',v.P,y(1,:),'o');
xlabel('Input: Deformation magnitude, m');
ylabel('Output: Heating power Wx10^-3');
legend('Initial angle-speed values', 'NN Predicted values');
figure
plot(t.P,t.V(1,:),'*',t.P,y1(1,:),'o');
xlabel('Input: Deformation magnitude, m');
ylabel('Output: Heating power Wx10^-3');
legend('Initial angle-speed values', 'NN Predicted values');
figure
plot(ptr,ttr(1,:),'*',ptr,y2(1,:),'o');
xlabel('Input: Deformation magnitude, m');
ylabel('Output: Heating power Wx10^-3');
legend('Initial angle-speed values', 'NN Predicted values');
figure
plot(v.P,v.T(2,:),'*',v.P,y(2,:),'o');
xlabel('Input: Deformation magnitude, m ');
ylabel('Output: Heating source speed, m/s');
legend('Initial angle-speed values', 'NN Predicted values');
```

```
figure
plot(t.P,t.V(2,:),’*’,t.P,y1(2,:),’o’);
xlabel(’Input: Deformation magnitude, m ’);
ylabel(’Output: Heating source speed, m/s’ );
legend(’Initial angle-speed values’, ’NN Predicted values’);
figure
plot(pn,ttr(2,:),’*’,ptr,y2(2,:),’o’);
xlabel(’Input: Deformation magnitude, m ’);
ylabel(’Output: Heating source speed, m/s’ );
legend(’Initial angle-speed values’, ’NN Predicted values’);

% NN linear regression analysis and plot
figure
[m(1), b(1),r(1)] = postreg(y2(1,:),tn(1,:));
xlabel(’Output: Heating power,Wx1000’);
ylabel(’Input: Deformation magnitude, mx10^-3 ’ );
legend(’Data from 3-D and simplified FEM models’, ’A=T’,’Best linear fit’);
figure
[m(2), b(2),r(2)] = postreg(y2(2,:),tn(2,:));
xlabel(’Output: Heating source speed, m/s’ );
ylabel(’Input: Deformation magnitude, m ’);
legend(’Data from 3-D and simplified FEM models’, ’A=T’,’Best linear fit’);

% Error function matrix and error performance values
e=ttr-y2;
%absolute performance mean error
apme=mae(e)
%mean squared error performance
msep= mse(e)
%sum of squared error performance
ssep = sse(e)
```

Bibliography

[1] *ABAQUS User's Manual.* Hibbitt, Karlsson, and Sorensen, Inc., 1999.

[2] S. L. Abrams, L. Bardis, C. Chryssostomidis, N. M. Patrikalakis, S. T. Tuohy, F.-E. Wolter, and J. Zhou. The geometric modeling and interrogation system Praxiteles. *Journal of Ship Production*, 11(2):117–132, May 1995.

[3] R. J. Anderson. Experiments and simulation of line heating of plates. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, September 1999.

[4] K. J. Bathe. *Finite Element Procedures.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1996.

[5] S. Brown and H. Song. Finite element simulation of welding of large structures. *Journal of Engineering for Industry*, 114(4):441–451, November 1992.

[6] ASM Handbook Committee. *Metals Handbook, Volume 1 Properties and Selection: Irons, Steels, and High Performance Alloys.* ASM International, Materials Park, OH 44073, 1990.

[7] G. Dahlquist and Å. Björck. *Numerical Methods.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1974.

[8] H. Demuth and M. Beale. *Neural Network Toolbox for Use with Matlab. User's Guide.* The MathWorks, Inc., 1998.

[9] C. F. Gerald and P. O. Wheatley. *Applied Numerical Analysis.* Addison-Wesley, Reading, MA, 4th edition, 1990.

[10] S. Haykin. *Neural Networks : A Comprehensive Foundation.* Prentice Hall, Upper Saddle River, N.J., 1999.

[11] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design.* A. K. Peters, Wellesley, MA, 1993. Translated by L. L. Schumaker.

[12] Y. Iwamura and E. F. Rybicki. A transient elastic-plastic thermal stress analysis of flaming forming. *Journal of Engineering for Industry*, 95(1):163–171, 1973.

[13] C. D. Jang, S. Seo, and D. E. Ko. A study on the prediction of deformations of plates due to line heating using a simplified thermal elasto-plastic analysis. *Journal of Ship Production*, 13(1):22–27, 1997.

[14] M. Mehrotra, C. Mohan, and S. Ranka. *Elements of Artificial Neural Networks*. MIT Press, Cambridge, MA, 1997.

[15] D. Rosenthal. Theory of moving source of heat and its application to metal treatments. *Transactions of American Society of Mechanical Engineers*, 68:849–866, November 1946.

[16] K. Satoh and T. Terasaki. Effect of welding conditions on welding deformations in welded structural materials. *Journal of Japanese Welding Society*, 45(4):302–308, 1976.

[17] K. Scully. Laser line heating. *Journal of Ship Production*, 3(4):237–246, 1987.

[18] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, Wellesley, MA, 1986.

[19] L. C. Thomas. *Heat Transfer*. Prentice Hall, Eaglewood Cliffs, NJ, 1992.

[20] N. S. Tsai and T. W. Eagar. Distribution of the heat and current fluxes in gas tungsten arcs. *Metallurgical Transactions B*, 16B:841–846, 1985.

[21] J. R. Wooten, S. Fowser, K. Masubuchi, W. Maher, J. Jones, and R. Martukanitz. Final Report on Laser Forming for Flexible Fabrication (AF Report number: RD98-415). Technical report, Rocketdyne Division of Boeing North American, Inc., Canoga Park, CA, February 1999.

[22] G. Yu. *Modeling of Shell Forming by Line Heating*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, April 2000.

[23] G. Yu, R. J. Anderson, T. Maekawa, and N. M. Patrikalakis. Efficient simulation of shell forming by line heating. *International Journal of Mechanical Sciences*, 2001. To appear.

[24] G. Yu, K. Masubuchi, T. Maekawa, and N. M. Patrikalakis. A Finite Element Model for metal forming by laser line heating. In C. Chryssostomidis and K. Johansson, editors, *Proceedings of the 10th International Conference on Computer Applications in Shipbuilding, ICCAS '99, Volume 2*, pages 409–418, MIT, Cambridge, MA, June 1999.