A Theoretical and Experimental Investigation Into the Variation of Process Parameters in the Laser Heating, Melting and Welding Processes

Thesis submitted in accordance with the requirements of The University of Liverpool for the degree of Doctor of Philosophy

by D.A.V. Clucas, B.Sc.(Hons), M.Sc (Eng)

March 1997

Laser Division
Department of Mechanical Engineering
University of Liverpool
Liverpool
L69 3BX

Declaration

I declare that no part of this thesis has been submitted for any other examination

D.A.V. Clucas

Acknowledgments

During the past few years working in the laser lab, I have met many people who have helped me in some way or another; to mention everyone would take forever, I owe them all my gratitude. Most notably, I would like to mention Sarah S and Dr. Brookfield, whose help and guidance in my masters dissertation initiated my interest in lasers. Bill. 0, Karen, Martin and Neil were also of great help as the 'old' members of the lab who gave me great advice when I needed it; my fellow students in the same year, most notably Katherine and Simon, who held out to the end; and also Johnny, Raja, and the CFD boys, Gerard and Mark upstairs. Thanks also go to the technicians, Jack for making my worktop for Heriot Watt, Bob for getting the x-y table moving, and Andy for cutting up all my samples when I had finished with the laser.

During this project, I have had the chance to travel to a number of places, thanks to the sponsorship of the Eureka 113 and 194 programmes, I thank all the guys at the Ekhatherineberg plant for helping with the Russian experiments there, and for introducing me to a rather unusual cocktail of liquid nitrogen and vodka. Thanks also go to the laser team at Heriot Watt, as well as all the other friends I made up there, together with the guys down at The Welding Institute for the CO laser experiments, especially Andy and Bill McNought from Ferranti, and of course the theory boys from Essex, especially Robert Ducharme, for helping me with my model development.

Over the last few years, I have made many friends whose help and support cannot go without thanks, especially Dave L, Sarah J, Ian M, Paul M, May, Ruth P, Ruth S, Sara, Stuart and Anna. Special thanks must go to Rachel for putting up with me whingeing down the phone at her; I'm sure you've got better things to do, Rach. I would also like to thank Preston North End for winning the third division last year, as well as Lancashire cricket team, who once again showed that they were the 'one day kings' by winning two trophy's this year, but even better; beating the Yorkies twice in each semi final!!!

Finally I would like to thank my supervisor; Professor Steen, for giving me the chance to undertake this research and for helping me along the way, and my dad for supporting me throughout my educational career; Thanks dad!

Abstract

Laser material processing can become very complex, and in order to understand these processes more fully, a full description of the mechanisms involved which characterise the process must be studied. In this thesis, a step is taken towards this goal by modelling pulsed and continuous wave laser welding, as well as the pulsed material removal process.

This investigation begins with a review of previous work in the field, together with an outline of the niches yet to be studied in the processes considered. The second chapter then considers two pulsed laser beam processes. The first section deals with pulsed laser welding, this model assumes that the effects due to fluid flow in the weldpool can be neglected and that the moving boundary (i.e the keyhole surface) can be calculated from the vaporising isotherm. It then solves the heat conduction equation in order to calculate the temperature distribution, from which the moving boundaries were calculated. The second section adapts this model by for laser material removal by assuming that all molten material is removed by the assist gas jet, therefore the melting isotherm defines the free boundary in this case.

The next part of the thesis concerns the development of a model for the continuous wave laser keyhole welding process, the model takes into account the beat input into the material by plasma absorption and direct absorption at the keyhole walls, together with the heat lost in the vaporisation of the material and conduction throughout the material. As well as the heat balance, a pressure balance, which has often been neglected in previous works, is also applied at the keyhole wall. The two balance equations form part a system of equations which describe the heat input and output to the system, as well as the flow characteristics in the keyhole itself The results of this model are then compared to the results of other models in the literature in order to determine a quick method of modelling the laser keyhole welding process, which takes into account all of these factors.

The final part of this thesis concerns the validation of the above models, together with the results of the model. In this study, results from the laser welding with a new type of laser; the CO laser; with a wavelength of 5.4m as opposed to 10.6m for the $C0_2$ laser; gives a useful insight into the driving mechanisms of the laser welding process as a whole.

CONTENTS

List of Figures

List of Tables

List of Abbreviations

a	Keyhole radius (m).
A	Area (m2).
$A(\theta)$	Fraction of absorbed light.
c_0	Speed of light (m/s).
C_p	Specific heat (J/kgK).
d	Keyhole length (m).
f(r)	Function that describes the power distribution.
g	Acceleration due to gravity (m/s^2)
g_o	Statistical weight.
g bar	Gaunt factor.
h	Planks constant (J s).
h	Convective heat transfer coefficient (W/m^2K)
H	Enthalpy function (J/m^3) or (J/kg)
I_n, K_n	Modified Bessel functions of the first and second kind, and of n^{th} order.
k	Extinction coefficient.
K	Boltmanns constant (J/K).
L	Linking intensity (W/m^2).
L_m	Latent heat of melting (J/K).
L_v	Latent heat of vaporisation (J/kg).
M^2	Quality factor of the laser beam.
n	Refractive index.
n_e, n_i, n_n	Number densities of the electron, ions and the neutral atoms within the plasma ($/m^3$)
P	Power (w)
p	Pressure (Pa).
p_{abl}	Pressure due to material ablating off the keyhole walls (Pa).
P_g	Excess pressure in the keyhole due to viscous flow (Pa).
P_{hydd}	Hydrodynamics pressure in the weldpool (Pa).
P_{hyds}	Hydrostatic pressure in the weldpool (Pa).
Pe	Peclet number = $r_b U/2\alpha$
Q_B	Power absorbed per unit length by the plasma (W/m).
Q_F	Power absorbed per unit length at the keyhole walls (W/m).
Q_λ	Power used up per unit length in conducting the beat throughout the material (W/m)
Q_v	Power per unit length used up in ablating the material (W/m)
r_b	Beam radius (m)
R	General reflectivity coefficient

R_p	Reflectivity coefficient for p polarised light
R_s	Reflectivity coefficient for 5 polarised light
t	Time (s)
T	Temperature (K)
T_0	Ambient temperature (K)
T_s	Surface temperature (K)
U	Velocity of the workpiece (m/s)
V	Volume (m^3)
u,v,w	Velocity components in the x,y, and z directions respectively (m/s).
W_i	Ionisation energy (Ev)
x,y,z	Length measurements in the x, y and z directions respectively (m).
z	Degree of ionisation.
α	Thermal diffusivity (m^2/s)
β	Thermal expansion coefficient (/K)
γ	Surface tension (N/m).
δ	Small increment
Δ	Large increment
μ	Viscosity (Ns/m^2).
ρ	Density (kg/m^3).
σ	Stefan-Boltzmann constant (J/sm^{K4})
τ	Stress tensor.
Θ	Non dimensional temperature.
$\acute{\omega}$	Frequency of laser light (Hz).

Chapter 1 Introduction and Literature Survey

This chapter serves as an introduction to the background subject matter and to the thesis as whole. It begins with a brief description of the laser, and outlines the various characteristics of laser light which are relevant to material processing. A description of laser light interaction with materials follows, together with the heat transfer processes within the material, once the energy has been absorbed. Previous work in the field of laser heating/melting and vaporisation processes are reviewed (this is summarised in tables 1.1 to 1.4 at the end of this chapter), together with a full mathematical description of these processes, with a view to formulating more detailed models in the following chapters.

1.1 Lasers and Laser Light

It was Maiman in 1960 who invented the first working ruby laser, using the principles of stimulated emission discovered by Einstein decades earlier. In the years that followed, many more materials were found to lase, and in 1964 C.K.N. Patel of Bell laboratories invented the first working CO_2 laser. This laser used pure carbon dioxide and produced 1mW of power at an efficiency of O.OOOl%. By adding nitrogen to the lasing gas the power increased to 200mW, and when helium was added, the power increased to 100W at an efficiency of 6%. Today the carbon dioxide laser, together with the Nd:YAG and excimer lasers, make up the largest proportion of the laser material processing market.

The word laser itself is an acronym, standing for Light Amplification of Stimulated Emission of Radiation. This gives one an idea of what a laser is and how it works. A laser is basically a light amplifier, consisting of three main parts, these are (Steen (1991))

- an optical resonator, which consists of two mirrors which are placed parallel to each other and perpendicular to the optical axis of the active medium, where light would oscillate back and forth unless it were prevented by some other mechanism such as absorption.

- an active medium between the mirrors which is capable of amplifying the light oscillations by means of stimulated emission.

- a system to pump the active medium so that it has enough energy to be active. In the case of gas lasers, this is usually a DC or RF power supply, in the case of the Nd:YAG laser, this is a focused pulse of light.

The light is amplified inside the laser by the active medium, and oscillates from mirror to mirror inside the laser. One of these mirrors is partially transparent allowing a portion of the light

to exit the optical cavity and emerge from the laser as the working laser beam. The other mirror is as totally reflecting as possible, and usually curved to reduce diffraction losses of the oscillating light, and to aid alignment of the laser beam.

1.2 Laser Light Characteristics

The light emerging from the optical cavity will have properties that are very important in laser material processing, these are discussed individually below.

1.2.1 Wavelength

Many materials have been found to lase, each lasing medium having its own characteristic wavelength of the generated laser beam (figure 1.1). The wavelength of laser light is a function of the energy transitions inside the active medium. The laser light is not monochromatic, but has a characteristic narrow bandwidth, the size of which is very important in focused beam laser material processing due to the fact that the refractive index of the lens material varies with the wavelength of the oncoming beam. If the laser operates over a wide bandwidth, this will cause the light at different wavelengths to be focused over a range of focal points, resulting in a larger minimum spotsize than if the laser beam had been monochromatic. This effect is known as chromatic aberration.

1.2.2 Coherence

The coherence of the laser beam describes the phase relationship between the electromagnetic rays within the laser beam itself, and can be described with respect to the plane perpendicular to the direction of the beam (spatial coherence), or in the plane parallel to the direction of beam propagation (temporal coherence) The laser light would need to be monochromatic in order to obtain perfect temporal coherence, however a high degree of temporal coherence can be achieved over a distance of a few mm to m in depending on the laser itself.

1.2.3 Polarisation

An electromagnetic wave consists of an electric and a magnetic vector field, oscillating at 90^0 to each other. It is the direction of the electric field vectors that define the polarisation of a wave. The plane of polarisation for the electromagnetic waves in the laser beam will be random unless special optics are introduced in the cavity to make the majority of the electric vectors line up in one plane, in this case the beam is said to be linearly polarised, when half of these electric vectors line up at 90^0 to each other, the beam is said to be circularly polarised.

1.2.4 Mode

Inside the laser cavity, the electromagnetic field generated by the stimulated emission phenomenon is constrained to certain configurations consistent with the boundary conditions. The

laser beam will then leave the cavity in certain preferred intensity distributions or modes (Steen (1991)). The geometry of the laser cavity, the gain in the cavity, inhomogeneties in the medium and the pumping power all contribute to the intensity distribution of the output laser beam mode. These modes are known as transverse electromagnetic modes TEM_{pq}, where p denotes the number of nodes in the x or r directions, and q denotes the number of nodes in the y or θ directions, for cartesian and polar coordinate systems respectively. The spatial distribution for any particular mode can be calculated by a Laguerre Gaussian Distribution function (Steen (1991)), figure 1.2 shows a number of electromagnetic modes used in laser material processing.

1.2.5 Divergence

All laser beams diverge, in order to compare the amount of divergence of one beam with another, the concept of the M^2 factor or Q factor (Coherent (1993)) was introduced, which is a comparison of the divergence of a real laser beam with that of an ideal gaussian beam of the same beam waist, this can be expressed in the following equation,

$$M^2 = \frac{\Theta_w}{\Theta_g} \qquad 1.1$$

where θ_w is the actual beam waist, and θ_g is the beam waist of the of the diffraction limited beam.

1.3 Effects of Laser Light Upon Materials

Figure 1.3 illustrates the effects when electromagnetic radiation strikes the surface of a metal. A part of the laser beam is absorbed, some is reflected, and the remaining part is transmitted through the material, which decays according to Beer Lamberts law $I=I_0e^{-\beta z}$, where β is known as the absorption coefficient and is dependent upon the medium, the wavelength and intensity of the light. In metals laser radiation is predominantly absorbed by free electrons in the 'sea of electrons'. These electrons are free to oscillate and reradiate without disturbing the solid atomic structure. Therefore the reflectivity of metals is high in the infra red region. The laser radiation induces the free electrons in the surface to vibrate in phase thus generating a field 180° out of phase with the oncoming beam (Steen (1991)). The laser radiation is unable to penetrate to any significant depth, and metals can be considered opaque, i.e, no radiation is transmitted through the metal, and therefore the fraction of the laser light which is absorbed is equal to 1-R, where R is the reflectivity. Once part of the energy from the laser beam is absorbed at the surface, the heat is spread throughout the material by conduction, and lost from the surface, by convection and radiation. If the amount of heat input is sufficient, the sample will melt, and may even vaporise. These are the essential mechanisms involved in laser material processing.

1.3.1 Absorption

The amount of heat that is transferred to the workpiece from the laser beam is determined by the absorptivity of the material at the particular laser wavelength. Therefore, for any scientific study of laser material processing to be valid, a full understanding of the mechanisms involved in the absorption process is essential. The absorptivity of a metal is determined by the polarisation, the angle of incidence, the wavelength, and the temperature. Each of these will be discussed in turn below.

1.3.1.1 Angle of incidence and Polarisation

When the laser beam strikes the surface of a material at normal incidence, the reflection coefficient is not dependent upon polarisation. However, when the angle is increases, the difference between the two polarisations becomes more pronounced (see figure 1.4), the reflectivity of a metal increases monotonically for s polarised light (where the electric field is perpendicular to the plane of incidence), whereas for p polarised light (where the electric vector is parallel to the plane of incidence) the reflectivity decreases to a minimum at a high angle of incidence (known as the Brewster angle), and then increases as the workpiece becomes more parallel to the oncoming beam. At any angle of incidence, the reflectivity can be calculated for s and p polarisation by the Fresnel formulae,

$$R_p(\theta)=\frac{(n^2+k^2)\cos^2(\theta)-2n\cos(\theta)+1}{(n^2+k^2)\cos^2(\theta)+2n\cos(\theta)+1} \qquad 1\text{-}2$$

$$R_s(\theta)=\frac{(n^2+k^2)-2n\cos(\theta)+\cos^2(\theta)}{(n^2+k^2)+2n\cos(\theta)+\cos^2(\theta)}$$

For circularly polarised light the total reflection coefficient is the average of the above two (figure 1.4 shows the angular dependence on reflectivity for p,s and circularly polarised light). Here n and k are the real and imaginary parts of the complex refractive index respectively.

1.3.1.2 Wavelength

Within the ultra violet, visible and infra red region the absorptivity of most metals increases with increasing wavelength (Steen (1991)), this is due to the fact that the more energetic photons can be absorbed by a greater number of electrons. Figure 1.5 shows the dependence of the reflectivity of a number of materials over a wide wavelength region relevant to laser material processing.

1.3.1.3 Temperature

As the temperature rises there will be an increase in the phonon population causing more phonon-electron energy exchanges, thus the electrons are more likely to interact with the structure rather than oscillate and reradiate (Steen (1991)). The reflectivity of metals thus tends to decrease with increasing temperature, this is illustrated in figure 1.6.

1.4 Modelling of Energy Transfer in Laser Material Processing

Once the beam is absorbed, the energy will be conducted throughout the material, and will be lost from the surface by convection and radiation. The governing equation for this situation is the heat conduction equation, subject to convective and radiative boundary conditions at the surfaces of the material. This describes the process of laser transformation hardening. If the energy is large enough, melting will occur, hence the energy used up in melting the material (latent heat), together with the convection in the molten pool will have to be taken into account in order to carry out a full analysis of the process. This is described as laser surface melting or conduction limited welding. If the intensity from the laser beam is high enough, the material will begin to ablate and form a vapour hole, known as the keyhole, throughout the depth of the material (figure 1.7). The energy is absorbed at the keyhole walls, and in the plasma inside the keyhole and just above it (this region is known as the plume), this energy is then conducted away from the keyhole and into the material. This describes the deep penetration welding process. In order to model these processes, the heat transfer and absorption mechanisms need to be taken fully into account. In the following sections, the processes will be described in a mathematical way, and previous models which attempt to describe these processes will be outlined.

1.4.1 Heat Transfer Processes

Heat transfer is defined by the energy in transit due to a temperature difference. Whenever there exists a temperature difference in a medium or between media, heat transfer will occur. Heat is usually transferred by a combination of modes, these are known as conduction, convection and radiation.

1.4.1.1 Conduction

A temperature gradient within a homogenous substance results in an energy transfer from the region of high temperature to the region with a lower temperature, this is described as conduction. Fourier formulated the basic law of heat conduction from experimental observations on steady state systems. It was observed that the heat flow was proportional to the temperature difference and the area normal to the direction of heat flow, this can be described by (Croft and Stone (1977)),

$$q = -kA\frac{\partial T}{\partial n} \qquad 1.3$$

where A is the area normal to the direction through which the heat flows, n represents the normal to the surface, k is a constant known as the conductivity of the substance, and T is the temperature. This equation describes one dimensional conduction only, in order to describe the multi dimensional problem one needs to consider the energy balance on a differential element as shown in figure 1.8.

The heat input to the element in the x direction can be expressed by

$$q_{x_{1\ input}} = -(kA\frac{\partial T}{\partial x})_{x_1} \quad 1.4$$

and the heat out of the element is described by

$$q_{x_{1\ out}} = -(kA\frac{\partial T}{\partial x})_{x_2} \quad 1.5$$

if we then consider the net heat conduction in the x direction only, we have,

$$q_{x_{input}} - q_{x_{out}} = -\delta y \delta z((k\frac{\partial T}{\partial x})_{x_1} - (k\frac{\partial T}{\partial x})_{x_2}) \quad 1.6$$

using a Taylor series expansion, we can further simplify' this expression to obtain

$$q_{x_{input}} - q_{x_{out}} = -\delta y \delta z(\delta x \frac{\partial}{\partial x} k\frac{\partial T}{\partial x} + ..) \quad 1.7$$

where the extra terms can be discounted as δx tends to zero (this analysis also applies in the y and z direction). Noting that the heat stored per unit volume is defined to be

$$\frac{\partial U}{\partial t} = \rho C_p(\delta x \delta y \delta z)\frac{\partial T}{\partial t} \quad 1.8$$

and hence dividing through by δV as δx, δy and δz all approach zero we obtain,

$$\frac{\partial}{\partial x}k\frac{\partial T}{\partial x} + \frac{\partial}{\partial y}k\frac{\partial T}{\partial y} + \frac{\partial}{\partial z}k\frac{\partial T}{\partial z} + H = \rho C_p\frac{\partial T}{\partial t} \quad 1.9$$

where k, ρ and C_P are the material dependent properties of conductivity, density and specific heat respectively, and H is the heat generated per unit volume within the medium.

1.4.1.2 Convection

Convective heat transfer occurs when whole groups of molecules move from one place to another. It usually occurs in fluids and between solid and liquid boundaries; for example with a hot plate in air, the motion of the fluid (air) is induced by buoyancy forces when density changes occur due to temperature increases, this type of heat transfer is known as free convection. If the motion of the fluid is externally induced, for example by an air jet, this type of heat transfer is described as forced convection.

Newton first discovered that heat was transferred by convection was proportional to the difference between the solid and fluid temperature, and the exposed area A (Croft and Stone (1977)),

$$q=hA(T-T_0) \quad 1.10$$

where h is a constant of proportionality known as the heat transfer coefficient, and is a function of the external flow over the surface, as well as the fluid properties.

1.4.1.3 Radiation

Radiation is different from the other two modes of heat transfer. Conduction and convection occur only in solid or fluid media, radiation however occurs in all matter above absolute zero, and heat can be transferred by radiation in a vacuum. Radiative heat transfer is transported by electromagnetic rays which are caused by vibrating electrons in the molecules of the material surface. It was through the work of Stefan and later Boltzmann that established that the rate at which energy is radiated from a body is proportional to the fourth power of its absolute temperature, as well as the exposed area. The net rate of heat transfer is expressed in the following equation (Croft and Stone (1977)),

$$q=\sigma A(T_s^4-T_0^4) \quad 1.11$$

where a is a constant of proportionality known as Stefan's constant. This equation is for an ideal radiator or black body. The heat emitted from the real radiator is a fraction of this power

$$q=\sigma \epsilon A(T_s^4-T_o^4) \quad 1.12$$

where ε is known as the emissivity of the substance and is a constant between 0 and 1.

1.4.2 Modelling Single phase problems

The goal of heat transfer analysis is to solve the relevant heat equations, subject to the appropriate boundary conditions, in order to obtain the temperature at any given point in the solution zone, at any particular time. In order to solve this equation, two distinct methods can be used, these are analytical or numerical methods. Work that has been carried out in both areas will be described below. However first it will be necessary to mathematically describe the laser heating process.

During the laser heating process the workpiece is scanned under a laser beam with a known power distribution, heat is conducted throughout the medium by the process of conduction, and is lost from the surface by radiation and convection. In order to model the heat conduction process correctly one needs to take into account the scanning velocity, this can be done by assuming that the workpiece moves with uniform velocity U, which can be represented by a convection term in the energy equation, and that quasi steady sate has been reached, hence all time derivatives are equal to zero. Using these assumptions, we obtain,

$$\rho C_p U \frac{\partial T}{\partial x} = \frac{\partial}{\partial x} k \frac{\partial T}{\partial x} + \frac{\partial}{\partial y} k \frac{\partial T}{\partial y} + \frac{\partial}{\partial z} k \frac{\partial T}{\partial z} \qquad 1.13$$

At the surface, the energy balance can be expressed in terms of the absorbed power input, P_A, the convective and radiative heat losses, and the heat conducted away from the surface,

$$-k\frac{\partial T}{\partial z} = \frac{P_A}{\pi r_b^2} f(x,y) - h(T_s - T_o) - \sigma\epsilon(T_S^4 - T_o^4) \qquad 1.14$$

where f(x,y) is a function which describes the power distribution over the surface. For the bottom and side surfaces, convective and radiative boundary conditions apply, and so the heat balance can be expressed as,

$$-k\frac{\partial T}{\partial z} = -h(T_s - T_o) - \sigma\epsilon(T_S^4 - T_o^4) \qquad 1.15$$

at the bottom of the workpiece, and similarly

$$-k\frac{\partial T}{\partial y} = -h(T_s - T_o) - \sigma\epsilon(T_S^4 - T_o^4) \qquad 1.16$$

at the sides.

1.4.2.1 Multi Dimensional Analytical Models

Various analytical solutions exist for the above system of equations, a number of important ones will be mentioned here. Rosenthal (1946) transformed the heat conduction equation into a simpler form by assuming the temperature distribution could be described by the form

$$T = T_o + e^{-\frac{vx}{2\alpha}} \Phi(x,y,z) \qquad 1.17$$

which could be solved for various situations with the boundary conditions that the partial derivatives of temperature tend to zero as the length away from the source tends to infinity, and that there is no convective or radiative heat losses at the workpiece surface. The solution depends upon the nature of the heat source, two solutions that have been used extensively in other works are the solution for the line source and the point source, these will be discussed below..

Neglecting heat conduction in the z direction, Rosenthal solved the heat equation for a moving line source in a semi infinite solid, by reducing the governing equation in T to one in Φ using the above relationship, and then solving for $\Phi$and then for T. The equation for the temperature distribution was found to be,

$$T-T_o=\frac{q'}{2\pi k}e^{-\frac{vx}{2\alpha}}K_o(\frac{vr}{2\alpha}) \qquad 1.18$$

where K_o is the modified Bessel function of the second kind and zeroth order, q is the strength of the line source, k is the conductivity of the workpiece, v is the velocity of the workpiece, α is the thermal diffusivity, and x and r are the distances along the x axis and from the heat source respectively.

The second of Rosenthal's solutions which has proved very useful in later research is the treatment of the 3-dimensional heat conduction equation due to a point source travelling along the surface of a semi infinite body, with velocity v. Again employing the above transformation, Rosenthal solved for Φ, and then for T as a function of space coordinate, in terms of the point source strength, the thermal properties of the workpiece and the translation velocity,

$$T-T_o=\frac{q}{4\pi kr}e^{-\frac{vx}{2\alpha}}e^{-\frac{vr}{2\alpha}} \qquad 1.19$$

Later, Cline and Anthony (1977) solved the heat conduction equation in 3 dimensions for a moving gaussian beam with constant velocity v, in a semi infinite solid by superimposing the known solution for the thermal distribution for a point source C located at x',y',t' over the area of the gaussian beam. This analysis yields a temperature distribution, after some simplification, of,

$$T(x',y',z')=\frac{P}{C_p}\int_0^{\infty}\frac{\exp(-((x+vt)^2+y^2)(2k^2+4\alpha t')^{-1}+z^2(4\alpha t')^{-1})}{\sqrt{\pi^3\alpha t'(2R^2+4\alpha t')}}dt \qquad 1.20$$

Using the above equation, they obtained a simpler empirical relation for the temperature distribution underneath the laser beam, this was found to be

$$T=T_o e^{-\frac{z}{z_o}},\quad where \qquad z_o=10Rf(0,0,0,V),\qquad f=\int_0^{\infty}\frac{e^{-H}}{\sqrt{2\pi^3(1+\mu^2)}}d\mu$$

$$H=\frac{(X+\rho/2\mu^2)^2+Y^2}{2(1+\mu^2)}+\frac{Z^2}{2\mu^2},\qquad \mu^2=\frac{2\alpha t'}{R},\qquad \rho=\frac{R}{\alpha V} \qquad 1.21$$

$$X=\frac{X}{R},\qquad Y=\frac{y}{R},\qquad Z=\frac{z}{R}$$

They extended their analysis to consider the deep penetration welding process. This was carried out by assuming that the keyhole tip was equal to the vaporisation temperature, and that the absorption could be described by a uniform power distribution of sources from the surface to the depth of the keyhole. Using the method of images to ensure there was no heat flow across the

surface, and also the simple empirical relation described above, they solved for the temperature below the keyhole. Using this relation, and solving for the depth of the keyhole as a function of absorbed power (in this study, the whole of the power was assumed to be absorbed), the authors estimate the penetration depth of the keyholing process,

$$z_m = z_v + z_o \ln\left(\frac{T_v}{T_m}\right) \qquad 1.22$$

where z_v is calculated from

$$\frac{P}{P_v} = \frac{2(z_v/z_o)}{1-e^{-2(z_v/z_o)}} \qquad 1.23$$

where z_0 is defined above, P is the power of the laser beam, P_v is the threshold power required to reach vaporisation on the surface, and z_v is the keyhole length. The authors compared their analysis to experimental data, and found that their approximation gave good results for thicknesses less than 10mm, however tended to overestimate the penetration depth at greater thicknesses

Gratzke, Kapadia and Dowden (1991) extended the analysis of Cline and Anthony (1977) in order to take into account plates of finite thickness and then to calculate the weldpool shape as a function of Peclet number. Using the method of images to ensure no heat flow across the weldpool boundaries, they obtain an equation for non dimensional temperature,

$$\Theta(x,y,z,t) = \sum_{n=-\infty}^{\infty} \int_{n=-\infty}^{\infty} \frac{d\tau}{1+\tau^2} \left(\exp\left(\frac{-\tau^2}{1+\tau^2}\right)\left(x' - \frac{Pe}{2\tau^2}\right)^2 + y^2\right) - \tau^2(z' - 2nz_o)) \qquad 1.24$$

where

$$\Theta = k r_o \frac{\pi^{3/2}}{P}(T - T_o), \qquad x' = \frac{x}{r_o}, \qquad y' = \frac{y}{r_o}, \qquad z' = \frac{z}{r_o}, \qquad Pe = \frac{r_o U}{2\alpha} \qquad 1.25$$

Where the case of n=0 describes the semi infinite body, while the other terms are corrections due to finite thickness. Here, quasi steady state was shown to be valid for times greater than 0.1s, independent of travel speed.

The temperature on the surface and interior were calculated using asymptotic expansions for high Peclet numbers. The authors compared their results with those that solved the heat conduction equation by numerical techniques and found good agreement. They found that the length and depth of the weldpool were not that sensitive to changes in beam radius r_o, but the width was. The authors found that effects due to finite thickness were important for thin plates, and that in the case of a semi

infinite body the width and depth of the weldpool decreased dramatically with Peclet number when Pe was small, and smoothly for large values of Peclet numbers. The length of the weldpool was found to be fairly constant for Peclet numbers greater than 1 in the infinite body case. However, for sheets of finite thickness, the length decreased more rapidly with Pe, and the weldpool could then be approximated by a liquid cylinder.

Ryalkin, Uglov et al (1978) considered the temperature distribution for the case of a diffused beam with a Gaussian distribution. At high velocities they estimated the beam to be a surface line source, infinitely small in the x direction, and finite in the y direction, (equal to the beam radius). The temperature distribution could be described in terms of velocity v, and the thermal properties of the workpiece, the absorbed power P_A, and the time, t,

$$T-T_o=\frac{P_A/v}{2\pi k(t(t+t_o))^{1/2}}\exp\left(-\frac{1}{4\alpha}\left(\frac{z^2}{t}+\frac{y^2}{t+t_o}\right)\right) \qquad 1.26$$

Where t_o is defined as,

$$t_o=\frac{r_B^2}{4\alpha} \qquad 1.27$$

One of the problems with this approach was that the temperature tended to infinity near the origin. Ashby and Easterling (1984) eliminated this by assuming the process could be better described by a moving source at a distance z_0 above the surface. The temperature field could then be described by the above solution, with the exception that z was replaced by the distance between z and z_0, i.e. $z-z_0$. In order to evaluate z_0, the authors took the limiting cases of t much greater than t_o, and t much less than t_o

$$z_o^2=r_b\frac{\sqrt{\pi}}{e}\sqrt{\frac{\alpha r_b}{C_1 v}}, \qquad t>t_o$$
$$z_o^2=\frac{\pi\alpha r_b}{2eC_1 v}, \qquad t<t_0 \qquad 1.28$$

where C_1 is a constant near unity that takes into account the width of the laser beam in the x direction. The authors then compared the results of the approximate formula to the results of a numerically calculated three dimensional solution, and found that when y was equal to zero, the approximate formula was valid. They then combined these solutions with kinetic models of the carbon diffusion within the workpiece in order to describe the near surface structure, and the hardness of carbon steels after laser treatment.

This model was further modified by Ashby and Shercliffe (1986), who created master plots for predicting case depths in laser transformation hardening. They carried out this analysis by substituting dimensionless parameters into the analytical solution of heat given above, and then describing the results in the form of master plots (figure 1.9).The dimensionless parameters were,

$$\frac{1}{q^*}=\frac{r_{bk}(T-T_o)}{(1-r_f)P}, \qquad v^*=\frac{vr_b}{\alpha}, \qquad t^*=\frac{t}{t_o},$$

$$Z^*=\frac{z}{r_b}, \qquad Z_o^*=\frac{z_o}{r_b}, \qquad ,X^*=\frac{x}{r_b}, \qquad Y^*=\frac{y}{r_b} \qquad \textbf{1.29}$$

The derivation used the general equation by Bass (1983) to calculate better approximations of the near and far field limits of z_0. This development of Ashby and Easterling's model gave better results when compared with experiments.

Davis et al (1986) followed Carslaw and Jaeger (1959) in solving the heat conduction equation in a semi infinite body using a gaussian distributed heat source in order to determine the hardened depth during laser surface hardening. Following Ashby and Easterling's analysis, they simplified the conditions required for hardening to occur, these were, i) The solid has to reach and go beyond the hardening temperature T_3, ii) This temperature must be sustained in order to allow for carbon diffusion and iii) The solid must be quenched in order to achieve the transformation structure in a relatively short time. The authors discounted the relative importance of ii) and iii), and thus set about calculating the conditions for i) to occur.

They proceeded to solve the equation using asymptotic solutions in various limiting situations, which gave satisfactory results for some limiting cases, however, when compared to experimental results, they found that in many situations satisfactory results were not obtained. The authors thought that this was due to changes in specific heat with temperature, so in order to take this into account, they used a value of specific heat which was dependent upon a typical temperature of the system. This was done by making the specific heat a function of $q/\alpha^{1/2}$ where $q=P_A/ak^{3/2}(T_3-T_0)$, and $\alpha=Ua\rho C_3/k_3$. After solving the equations again the authors obtain a value for the hardened depth d (assuming the power is large enough for hardening to occur),

$$d=0.76017D\left(\frac{1}{\alpha\bar{c}/q+\pi^{1/4}(\alpha\bar{c}/q)^{1/2}}-\frac{1}{\alpha\bar{c}/q_{min}+\pi^{1/4}(\alpha\bar{c}/q_{min})^{1/2}}\left(\frac{q_{min}}{q}\right)^3\right) \qquad \textbf{1.30}$$

where q_{min} is defined as the minimum value of q at which hardening occurs, which can be approximated by,

$$q_{min}=(0.4058+0.21586\alpha\bar{c}) \quad 1.31$$

and c bar is a measure of the specific heat which can be calculated from,

$$\bar{c}=c_\infty-0.46460(c_\infty-c_o)(\alpha c_o)^{1/2}/q$$
$$c_\infty=c_o+0.74062c_1+2.45881c_2 \quad 1.32$$

with the values of c_0, c_1 and c_2 being calculated from the quadratic approximation for specific heat,

$$C_p(T)=C_3(c_o+c_1(\frac{T-T_o}{T_3-T_o})+c_2(\frac{T-T_o}{T_3-T_o})^2) \quad 1.33$$

It was found that this solution gave more satisfactory results when compared to experimental data.

Davis (1983) and Noller (1983) independently of each other, modelled the laser keyhole welding process by assuming that the keyhole could be described by a uniformly hot cylinder of radius a and a temperature T_v, surrounded by a solid material moving with a constant velocity U with thermal diffusivity α. The boundary conditions are that the temperature is equal to the vaporisation temperature at r=a, and tends to the ambient temperature T_0 as r tends to infinity. Substituting $\varepsilon=U/\alpha$ and $T=Pe^{\varepsilon x/2}$ into the heat conduction equation, the resulting partial differential equation can be solved in terms of P and then in T. The full solution can be expressed as,

$$T-T_o=2T_v\exp(\frac{1}{2}\epsilon r\cos(\theta))(\frac{I_o(\frac{\epsilon a}{2})K_o(\frac{\epsilon r}{2})}{K_o(\frac{\epsilon a}{2})}+\sum_{n=0}^{\infty}(-1)^n(\frac{I_n(\frac{\epsilon a}{2})K_n(\frac{\epsilon r}{2})}{K_n(\frac{\epsilon a}{2})}\cos(n\theta)) \quad 1.34$$

where I_o and K_o are modified Bessel functions of the first and second kind, and of n^{th} order. The above equation is known as the Davis-Noller solution and has proved to be a very important result in calculating the temperature distribution in the weld or heat flux into the weld in many studies since.

Compared to CW laser processing, relatively little theoretical work has been undertaken in the pulsed laser welding field. However, there have been a few papers in the last decade dealing with the problems specific to pulsed laser heating, welding and drilling. Due to the complexity of the problem, analytical transient solutions to the melting and evaporation models have been mostly restricted to one space dimension. El-Adawi and Elshahawey (1986) examined the problem of heating a homogenous slab of material induced by a time dependent laser irradiation. They obtained a solution using a Fourier series expansion technique in order to calculate the critical time required to initiate melting. Modest and Abiakans (1986) compared a one dimensional integral solution for a pulsed laser heat source with the exact solution for 3 space dimensions in order to find a window

where the simpler solution was applicable. In order examine the pulsed keyhole welding process, Simon et al (1993) developed a 2 dimensional solution to the heat conduction equation for a pulsed cylindrical heat source in a moving, semi infinite solid. However, in order to gain an analytical solution to the problem, they assumed that the frequency was in the kHz region, whereas pulsed laser welding is usually undertaken in the Hz region, and so it would be dubious to draw any concrete conclusions from this analysis.

1.4.2.2 Multi Dimensional Numerical Models

An alternative way for solving the heat conduction equation is by numerical treatment using either finite difference or finite element methods. These methods have become more popular recently with the advancement of computer technology as they can solve more detailed problems than analytical methods, such as varying thermal properties or radiative boundary conditions.

In the finite difference method, the solution zone is divided into a network of interlocking grid points, the governing partial differential equation is then discretised resulting in an algebraic equation for temperature at each grid point in space, in terms of the thermal properties and the temperature of the neighbouring points. This system of equations can then be solved in order to find the temperature distribution throughout the whole domain. To illustrate this method, observe figure 1.10 a) where a node in space is surrounded by 6 neighbouring nodes. If we consider the partial derivatives in the x direction only, we can approximate these equations using the forward, backward or central difference formulae, these are,

$$\frac{\partial T}{\partial x} \approx \frac{T_N - T_p}{\delta x_n}, \qquad \frac{\partial T}{\partial x} \approx \frac{T_P - T_S}{\delta x_s}, \qquad \frac{\partial T}{\partial x} \approx \frac{T_N - T_S}{\delta x_n + \delta x_s} \tag{1.35}$$

respectively, where δx_n and δx_s are the distances between points P and N, and P and S, respectively. The second order partial derivative can be discretised by noting that,

$$\frac{\partial^2 T}{\partial x^2} = \frac{\partial}{\partial x}\frac{\partial T}{\partial x} \approx \frac{1}{\Delta x}\left(\frac{T_N - T_P}{\delta x_n} - \frac{T_P - T_S}{\delta x_s}\right) \tag{1.36}$$

where Δx is half the distance between points N and S. A similar analysis can be carried out on the temperature derivatives with respect to y and z. The time derivative can be solved using the forward difference formula,

$$\frac{\partial T}{\partial t} \approx \frac{T_P^n - T_P^o}{\delta t} \tag{1.37}$$

where the superscripts n and o denote the new and old temperatures at point P respectively. Using

these approximations together with using the backward difference formula in evaluating the first order partial derivative with respect to x for the convection term, and then substituting back into the general differential equation (equation 1.13) we obtain,

$$\rho C_p\left(\frac{T_P^n - T_P^o}{\delta t} + U\frac{T_P - T_S}{\delta x_s}\right) = \frac{k}{\Delta x}\left(\frac{T_N - T_P}{\delta x_n} - \frac{T_P - T_S}{\delta x_s}\right) + \frac{k}{\Delta y}\left(\frac{T_E - T_P}{\delta y_e} - \frac{T_P - T_W}{\delta y_w}\right) + \frac{k}{\Delta z}\left(\frac{T_U - T_P}{\delta z_u} - \frac{T_P - T_D}{\delta z_d}\right) \quad \textbf{1.38}$$

where the temperature on the right hand side can either be all evaluated at the new time, in which the formulation is said to be implicit, and is solved by matrix methods, part new and part old temperatures, in which case the formulation is said to be semi implicit, or completely in terms of the old temperature, in which case the formulation is said to be explicit. The simplest case, i.e. the explicit case, will be described here. In order to apply the correct boundary conditions, equation 1.14 can be discretised in terms of a fictitious temperature T_G placed above the surface (in the place of T_D, see figure 1.10 b),

$$-k\frac{T_U - T_G}{\delta z_u + \delta z_g} = \left(\frac{Pf(x,y)Abs}{\pi r_b^2} - h(T_S - T_o) - \sigma\epsilon(T_S^4 - T_o^4)\right) \quad \textbf{1.39}$$

this can substituted back into the main discretised equation above in order to eliminate the unknown temperature at the ghost point, T_G, which will then give an equation for the surface temperature in terms of the neighbouring temperature and the power. The temperature at the bottom and sides can be calculated using a similar method. Therefore, we now have the particular equations for each node in the solution zone. In order to solve the heat conduction equation, a guessed temperature field is used for the first iteration (for example, all temperatures are set to the ambient temperature), and then the new temperatures are solved in terms of the heat input and output at each grid point. This procedure is repeated until the change in temperature difference between iterations becomes small enough to satisfy the convergence criteria, or when the time of interest has elapsed.

Steen (1976), was one of the first to apply multi dimensional finite difference methods to laser material processing, when he used a 2-dimensional cylindrical geometry in order to solve the time dependent heat conduction equation for a stationary beam in order to analyze the process of laser vapour deposition. He then built on this earlier work in 1980 with a full 3-dimensional heat conduction model (Mazumder and Steen (1980)), similar in outline to the one described above, for a moving Gaussian heat source. The model described the situation whereby a laser beam, having a defined power distribution, strikes an opaque substrate having a finite width and depth and infinite length moving in the positive x direction with a uniform velocity. The authors assumed that the absorption was a fraction less than 1 up to the boiling point; after boiling was reached, the absorption

was set to 1 (thus assuming that the substrate became a black body). Using constant thermal properties and neglecting latent heat, the authors then set about solving the governing equation with the known boundary conditions using a successive over relaxation method. The authors then used their model to predict the temperature profile, maximum welding speed, the thermal cycle at any location on the substrate, the effect of thickness or any other parameter or the effect of supplementary heating or cooling. They found that for laser transformation hardening and laser glazing of nickel alloy the results compared reasonably well with experiment.

A number of researchers have tackled the problem of a pulsed laser heat source using numerical techniques. Gellert and Egli (1988) solved the transient 1 dimensional problem for copper using Langmuir evaporation as a cooling mechanism. Smurov et al (1991) furthered this analysis by calculating the amount of melting and vaporisation caused by various shaped laser pulses, however, they did not consider the range of frequencies used in laser welding. This flaw was taken into account by Tzeng (1996), who used an explicit 1 dimensional finite deference algorithm to distinguish between the melting regimes when using various laser pulse shapes.

1.4.3 Modelling two phase processes

When the energy from the laser beam is high enough to cause melting, the phase boundary can be modelled by applying an energy balance. This can be shown to be (Davis 1983),

$$(k\frac{\partial T}{\partial n})_l-(k\frac{\partial T}{\partial n})_s=-\rho L v_n \qquad 1.40$$

ie, the energy conducted into the interface from the liquid phase, minus the energy conducted away from the interface, in the solid phase, is equal to the energy used up in the latent heat of melting. Here v_n is the normal velocity of the phase boundary, n is the normal at the phase boundary, and the subscripts s and l denote that the derivatives have been calculated in the solid and liquid region respectively. There are various methods that have been applied in the literature to solve this problem, Crank (1986) gives an extensive survey on the earlier work carried out in this area. These methods fall into three separate categories, the first of which are known as front tracking methods, where the phase boundary is computed after each iteration, this requires special treatment of partial derivatives at the phase boundary Alternatively, in order to obtain a multi dimensional solution, the grid can be transformed after each iteration in order to coincide exactly with the phase boundary. An example of this more usable approach was taken by Meyer (1970) called the method of lines. For problems in one space dimension, the time variable is discretised and the governing partial differential equation is replaced by a series of ordinary differential equations at discrete time levels. This has a number of advantages over other techniques, however the one big disadvantage is that it can only be solved over a topological rectangle. This can be overcome by employing a curvilinear transformation of variables from the solution zone to a rectangle, which takes a great amount of time, especially in

more than one dimension, and so will be disregarded here as a method of solution. However, it should be noted that a 2 dimensional method of lines model has been applied successfully to solve the flow conditions and temperature distribution within the weldpool in laser keyhole welding problem by Davis (1983).

The second type of numerical method for dealing with the moving boundary, is to fix it by a suitable choice of new space coordinates which fit the boundary exactly, by again employing a curvilinear transformation. In moving boundary problems, the region changes with time, but a fixed (ξ,η) mesh corresponds to a moving (x,y) mesh can be used for all time. The movement of the boundary and mesh points in the original region, appear only as changes in x and y at the corresponding fixed (ξ,η) points after each iteration or time step. The advantages of working in a fixed region such as a rectangle or cuboid, with no loss of accuracy near the boundary have been found to be considerable. Another example of a front fixing method is the isotherm migration method, in this algorithm, instead of solving for the temperature as a function of space coordinate, and here the governing heat conduction equation is transformed from an partial differential equation in T, to a partial differential equation in one of the space variables. This technique is particularly suited to phase problems as the boundary itself is an isotherm, providing the phase change takes place over a constant temperature; again though, front fixing methods have the very serious disadvantage of requiring curvilinear transformations, which increases the time in which a user can solve these equations.

In the previous methods, curvilinear transformations were nearly always required at the boundaries in order to satisfy the free or moving boundary. These methods can work well if the boundary monotonically increases with space coordinates and time, however in some situations, this is definitely not the case (eg pulsed laser welding, where the melting isotherm decreases and disappears once the laser is switched off and the substrate cools). Therefore, it can be seen that a method which implicitly solves the latent heat condition at the free or moving boundary which applies to the whole domain is an attractive idea. This is the driving force behind the so called enthalpy method, whereby the heat equation is solved in terms of enthalpy and the moving boundary comes out at the end of the process as a result. The method was first proposed by Eyers et al (1946). The procedure involves the introduction of the enthalpy function H(T), which accounts for the total heat content ie the rise in temperature as well as latent heat. The jump at the interface boundary is thus incorporated in the definition of H(T).

$$H(T)=\int_{T_o}^{T}\rho C_p dT, \qquad T<T_m$$

$$H(T)=\int_{T_o}^{T}\rho C_{pdT}+\rho L, \qquad T>T_m \qquad \textbf{1.41}$$

$$\int_{T_o}^{T}\rho C_p dT \le H(T) \le \int_{T_o}^{T}\rho C_{pdT}+\rho L, \qquad T=T_m$$

The function H(T) is shown in figure 1.11 a. The governing heat conduction equation is thus transformed, but is not defined at the melting temperature since the enthalpy has a jump discontinuity at T-T_m. However Shansunber and Sparrow (1975) avoided this by considering the energy balance of a control volume V with surface are A, devised the following integral formulation of the enthalpy equation,

$$\frac{d}{dt}\int_V HdV+\int_A HV\cdot dA=\int_A k\nabla T\cdot dA \qquad \textbf{1.42}$$

which they proved was valid throughout the whole of the region of interest. Variations on this formulation have been used with success when applied to laser welding This method has two main advantages, the first is that the fluid flow can easily be taken into account, and also there is no need to change coordinate systems, which can save immense time, especially with 3-dimensional models.

Kou and coworkers (Kou (1981), Kou and Le (1983), (1984)) were the first to apply these techniques to the melting and welding problems. The authors adapted the above formulation in order to solve the enthalpy equation by a finite difference method in order to simulate the gas tungsten arc (GTA) welding process. The following equation was solved.

$$\int_V \frac{\partial(\rho H)}{\partial t}dV=\int_A k\nabla T\cdot dS-U\int_V \frac{\partial(\rho H)}{\partial x}dV$$

$$\textit{where} \qquad H=\int_{T_o}^{T} C_p dT+f_l\Delta L_m \qquad \forall T \qquad \textbf{1.43}$$

$$\textit{where} \qquad f_l=0 \qquad T<T_S, \qquad f_l=\frac{T-T_S}{T_l-T_S} \qquad T_S<T<T_l, \qquad f_l=1, \qquad T>T_l$$

where the enthalpy function (figure 1.11b) has been modified in order to avoid the discontinuity at the melting point. Here, the liquid melts in a region of temperature rather than at the melting point, between T_s the solidus temperature, and T_l, the liquidus temperature. The authors then solved for the temperature in terms of enthalpy, subject to convective and radiative boundary conditions. They then compared the results they obtained from the model with experiments in GTA welding of aluminium,

with some success, although there were some differences in the two which was presumably due to the fact that the model did not take into account fluid flow effects.

1.4.4 Fluid Flow and Heat Transfer in The Weld Pool

Once the melting point has been reached, a buoyancy force in the weldpool and the surface tension force along the weldpool surface will induce a flow inside the melt pool which will affect the temperature gradient. Figure 1.7a) shows a schematic diagram of a typical fluid flow pattern in the weldpool, assuming incompressibility, the governing equations for this situation (Kou and Sun (1985)) are the equation of continuity

$$\nabla \cdot V = 0 \tag{1.44}$$

the Navier Stokes equation

$$\rho(V \cdot \nabla)V = -\nabla P - (\nabla \cdot \tau) - \rho g \beta (T - T_r) \tag{1.45}$$

and the energy equation

$$V \cdot \nabla(\rho H - P) = \nabla \cdot (k \nabla T) \tag{1.46}$$

these can be solved using the following boundary conditions. At the top boundary (z=0), there is no flow across the boundary, and the fluid flow along the surface is induced by the rate of change in surface tension (i.e Marangoni convection), thus

$$w = 0, \qquad \mu \frac{\partial u}{\partial z} = \frac{\partial T}{\partial x}\frac{\partial \gamma}{\partial T}, \qquad \mu \frac{\partial v}{\partial z} = \frac{\partial T}{\partial y}\frac{\partial \gamma}{\partial T} \tag{1.47}$$

The energy balance at z=0 can again be described by

$$-k\frac{\partial T}{\partial z} = \frac{P_A}{\pi r_b^2} f(x,y) - h(T - T_o) - \sigma\epsilon(T_S^4 - T_o^4) \tag{1.48}$$

Assuming the weldpool depth is less than the thickness, the boundary conditions at the bottom surface are,

$$u = U, \qquad v = w = 0, \qquad k\frac{\partial T}{\partial z} = h(T - T_o) + \sigma\epsilon(T_S^4 - T_o^4) \tag{1.49}$$

and along the centre plane

$$v = 0, \qquad \frac{\partial u}{\partial y} = \frac{\partial w}{\partial y} = \frac{\partial T}{\partial y} = 0 \tag{1.50}$$

At the weldpool boundary,

$$v=w=0, \qquad u=U \tag{1.51}$$

and as x,y $\rightarrow \infty$,

$$v=w=0, \qquad u=U, \qquad \frac{\partial T}{\partial x}=\frac{\partial T}{\partial y}=0 \tag{1.52}$$

where V(u,v,w) is the velocity vector, U is the velocity of the moving workpiece, γ is the surface tension, p is the pressure, μ is the viscosity of the liquid metal, β is the coefficient due to thermal expansion, and τ is the stress tensor.

1.4.4.1 Analytical Solutions For Flow Within The WeldPool

In order to take into account the above system of equations in 3 dimensions, a full numerical treatment is necessary, however useful analytical solutions have been obtained which concentrate on particular aspects of the molten flow. Postacioglu, Kapadia and Dowden (1987) investigated the upward flow of the weldpool during laser keyhole welding due to pressure gradients. The authors calculated the flow characteristics in order to estimate the shape of the elevation or depression on the top surface of the weld. The authors later studied the thermocapillary flows on the weldpool surface by developing a non linear boundary layer model of thermal conduction and fluid convection. Gratzke et al (1992), investigated the mechanisms involved in the humping phenomenon, they discussed and dismissed thermocapillary flows as the dominant factor and instead applied Rayleigh's theory of instability of a free liquid cylinder due to surface tension (i.e under certain conditions the weldpool shape is such that disturbances increase and the weldpool breaks up into droplets), and deduced that the width to length ratio had to exceed $l/2\pi$ in order to avoid humping.

A number of analytical solutions for 2 dimensional flow in the weldpool have been proved useful in later work. Colla, Vicanek and Simon (1992) modelled the two dimensional flow in the weldpool in laser keyhole welding assuming that the situation could be modelled as an ideal fluid flowing past a cylinder (the keyhole) with velocity U. At large distances, the flow is uniform with velocity U, while the flow is disturbed close to the keyhole. The authors solved the flow conditions and temperature distribution in such a system. The authors then compared their results to the Davis-Noller solution (figure 1.12). looking at the generated isotherms for Colla and Noller's temperature distribution it can be seen that melting isotherm becomes much shorter as a result of the melt flow. In fact, at Peclet numbers greater than 1, there can be a substantial difference in the length of the melting isotherms. This can be understood by the fact that the Davis-Noller solution is calculated using a uniform velocity throughout the material, hence the melt is actually flowing through the keyhole, whereas Colla's work assumes that the fluid flows around the keyhole, which tends to reduce the velocity, and hence reducing convective heat transfer

1.4.4.2 Numerical Solutions For Flow Within The WeldPool

In order to take into account the Navier Stokes equations subject to the appropriate boundary conditions, a full numerical treatment is required. This has become possible with the advances in computers over the recent years. Oreper and Szekely (1984) modelled the TIG welding process, in which they developed a mathematical formulation to represent the transient behaviour of the fluid and energy flow in a liquid pool, generated by a spatially variable heat flux and current flowing through an initially solid block. The governing equations were transformed into a non-dimensional form, and they solved the vorticity transport equation using the finite difference method, taking into account the surface tension, buoyancy and electromagnetic forces. The authors considered a number of cases, the most important of these being the case where surface tension gradients and buoyancy effects were considered as the two driving forces of the flow in the weldpool. They reported that when the surface tension derivative with respect to temperature was negative, the fluid at the top of the meltpool surface was forced radially outwards along the surface, resulting in a wider and shallower weldpool. However, when the surface tension derivative was positive, the fluid was forced radially inwards along the surface, and was heated underneath the heat source, and then pushed down to the bottom of the weldpool, thus increasing the weldpool penetration depth, and decreasing the width. The authors extended their analysis (Oreper and Szekely (1987)), in considering the fluid flow and temperature distribution in the TIG welding of aluminium (a very good conductor), steel (a reasonable conductor) and titanium (a poor conductor), although the heat flux into each material was not the same in each case, and so an absolute comparison was not possible, it did provide illuminating insight into the behaviour of these systems. The governing equations were solved subject to the boundary conditions described above, and the authors calculated the conditions were convection plays a major role. They found that convection was unlikely to be important for good conductors such as aluminium, in shallow weldpools, and was likely to be important for deep weldpools and for poor conductors. Wherever significant surface temperature gradients existed, they were likely to play an important role in defining the flow pattern and the weldpool shape.

Chan, Mazumder and Chen (1984), were the first to calculate the transient flow in a surface tension driven weldpool, due to a scanning rectangular laser beam, which was incorporated into the flow equations using a coordinate transformation. Assuming constant thermal properties, and neglecting the latent heat of fusion, they solved the 2 dimensional equations of fluid flow and temperature using a finite difference method. They found that the recirculating velocity near the surface due to surface tension gradients was one or two orders of magnitude greater than the moving heat source, and that there were two counter rotating vortices except for values of high Prandtl number. The aspect ratio of the weldpool (width/depth) increased with increasing Prandtl number, and as the surface tension gradient increased the aspect ratio increased up to a point, and then began to decrease due to the fact that the swirling of the weldpool is so rigorous, two counter rotating vortices

on each side were generated, which led to deeper penetration due to the clockwise rotating vortex. The cooling rate at the edge of the pool was found to be higher than that at the bottom of the pool. This implies that solidification will take place from the edge of the weldpool, and proceed radially inwards.

The previously mentioned studies did not take intro account the latent heat of fusion, the first researchers to apply the enthalpy method in investigating the fluid flow during the welding process were Kou and Sun (1985), who studied the steady state welding process in two dimensions. The unique part of this study was the fact that the weldpool boundary was not calculated during each iteration, instead the governing equations were solved over the whole domain, but the viscosity set to equal to infinity for temperatures less than melting point, which effectively set all velocities outside the weldpool equal to zero. The authors went on to compare the results of the numerical model to simplified 2 dimensional analytical solutions that were available, with some success. The found that for materials with a high negative surface tension gradient, the influence of the driving force on the weldpool decreases with the order of surface tension gradient at the weldpool surface, the electromagnetic force and the buoyancy force. The authors noted that the weld penetration is strongly affected by fluid flow in the weldpool under certain conditions, such as high surface tension gradients, which tended to result in a shallower and wider pool. and electromagnetic forces which tended to force the flow radially inwards, resulting in a deeper and thinner welppol. Kou and Wang (1986) extended the two dimensional treatment to consider the 3 dimensional case, and compared their results with experimental data for the melting of aluminium. The model did not consider rapid solidification and so was only valid for low translation speeds, in this region the model compared well with experimental results, which were quite similar to the 2 dimensional case. They found that, with Chan et al (1984), the recirculating velocities in the weld pool can be up to an order of 2 greater than the translation speed, and that the fluid flow was forced radially outward along the surface of the weldpool, resulting in a larger weldpool width and shallower weldpool depth. If the surface tension coefficient was made positive, the fluid was forced radially inwards and downwards resulting in a deeper, narrower weld.

The two dimensional model was improved by Tsai and Kou (1989) by taking into account the deformation of the free surface which was calculated by applying a pressure balance at the boundary, in order to calculate the Marangoni convection properly. They employed a curvilinear transformation which fitted the free surface exactly. The governing equations of fluid flow and energy transfer were solved for a particular surface until convergence, and then the pressure balance was applied at the surface in order to calculate an improved estimation of the free boundary. The grid was then deformed to meet the new calculated surface. This process was repeated until there was no discernible difference in the change of weldpool surface. The authors found that a negative surface

tension gradient produced a radially outwards surface flow which induced a slight depression on the weld at the centre and a slight elevation towards the edge. In this case the results obtained for the melting isotherm were very similar to those in the flat surface case. However, in the case of a positive surface tension, the centre of the weldpool surface was elevated, while the sides of the weldpool were depressed. This caused a 10% reduction in the weldpool depth when compared with the flat surface case, (figure 1.13), as can be seen, the flow downwards parralel to the z-axis decays as one moves away from the centreline, however this is more pronounced in the case of a deformable surface. As a result, the isotherms in the deformable surface case are pushed downwards only at the centreline, whereas those for the undeformed weldpool are pushed down in a larger region around the centreline, resulting in a larger weldpool, thus the weld depth can be overestimated if the pool surface is assumed flat, rather than deformable.

The authors later modified their analysis in order to study the volume expansion in the weldpool due to density changes caused by temperature change, and change of phase (Tsai and Kou (1990)). The authors studied the case when there was no volume expansion, the case when volume expansion was due to temperature changes in density alone, and the case were volume expansion was due to the change in density with temperature, as well as the change in density due to melting. In the first case the velocity field exhibits a typical natural convection pattern in the weldpool. In the second case the weldpool expands slightly, and the resulting velocity field and temperature distribution were found to be similar to the first case. However, when the density variation due to phase chage was taken into account, the weldpool expanded by up to three times as much as the second case. It was found that the resulting weldpool depth could be overestimated by about 10-15% in the case of a flat surface. Hence, the volume expansion associated with the phase change can be a significant factor in the laser melting process.

1.4.5 Modelling Vaporisation Processes

In laser welding the transition between conduction limited welding and deep penetration welding is dictated by the power intensity and the translation speed of the workpiece. Above a certain threshold intensity, the metal reaches the vaporisation temperature and a crater or hole known as the keyhole is formed inside the metal. During high intensity focused laser beam processes, the vapour in the keyhole and the plume can form a plasma which absorbs the laser light by a mechanism known as Inverse Bremsstrahlung absorption. The laser power is also absorbed at the keyhole walls by Fresnel absorption. In order to describe the phenomena in the keyhole, an energy and pressure balance (figure 1.14 a) and b) respectively) at the keyhole walls need to be applied in order to calculate the keyhole dimensions.

1.4.5.1 Heat Balance at the keyhole wall

Most models of the laser keyholing process apply an energy balance at the keyhole wall in order to determine the keyhole dimensions (figure 1.14)., this energy balance can be described in terms of power absorbed per unit depth, thus

$$Q_B+Q_F=Q_\lambda+Q_v \qquad 1.53$$

where Q_B and Q_F represent the energy inputs into the weldpool in the form of plasma absorption conducted to the keyhole walls and Fresnel absorption at the keyhole walls, respectively; Q_λ is the power per unit depth conducted away from the body and into the material, and Q_v is the power per unit depth used up in vaporising the material.

1.4.5.1.1 Inverse Bremsstrahlung absorption in the keyhole

If one assumes local thermodynamic equilibrium and that heat transfer inside the plasma is non radiative, the degree of ionisation can be calculated from the Saha equation (Kaplan (1994), Ducharme, Williams et al (1995))

$$\frac{n_e n_i}{n_n}=2\frac{(g_o)_i}{(g_o)_n}\frac{(2\pi m_e kT)^{3/2}2\pi}{h}e^{-\frac{W_i}{kT}} \qquad 1.54$$

where n_i, n_n and n_e are the particle densities of ions, neutrals and electrons respectively; g_o is the statistical weight, m_e is the electron mass, h is Plank's constant, k is Boltzmann's constant, and W is the ionisation energy. The dominant absorption mechanism in the plasma in CO_2 laser welding is known as inverse Bremsstrahlung. A formula has been obtained for the linear absorption coefficient (Hughes (1975)).

$$\alpha=\frac{2\pi Z^2 e^6 N_e n_i(1-e^{\frac{-k\omega}{kT}})}{6\sqrt{3}\omega^3 m_e^2 C_o \epsilon_o^3(1-\omega_{pe}/\omega)}\bar{g}\sqrt{\frac{m_e}{2\pi kT}} \qquad 1.55$$

where Z is the degree of ionisation calculated from Saha's equation, ω is the angular frequency of laser radiation, ω_{pe} is evaluated by

$$\omega_{pe}=e\sqrt{\frac{m_e}{m_e\epsilon_o}} \qquad 1.56$$

ϵ_0 is the dielectric constant, c is the speed of light in a vacuum, and e is the elementary charge; g bar is the quantum mechanical Gaunt factor and can be approximated by (Dowden, Postacioglu, Davis and Kapadia (1987)),

$$\bar{g}=\frac{2\frac{\sqrt{2}}{\pi}\ln(4e^{C\epsilon})}{h\omega/kT} \qquad 1.57$$

where E=0.5772 is Euler's constant. Ducharme et al (1995) calculated the inverse Bremsstrahlung absorption at discrete depths down the keyhole, whereas Kaplan (1994) and others used a mean value valid throughout the keyhole of $100m^{-1}$. Most researchers only assume radial conduction in the plasma (eg Dowden, Postacioglu, Davis and Kapadia (1987); Ducharme, Williams et al (1995)). The heat conduction equation in this region then becomes,

$$\frac{1}{r}\frac{\partial}{\partial r}rk\frac{\partial T}{\partial r}+q=0 \qquad 1.58$$

where q is the absorbed energy and the conductivity k can be calculated as a function of temperature from the equation in Lancaster (1986), for iron plasma. The total heat conducted to the sideways walls can be calculated by (Ducharme, Williams et al (1995)),

$$Q_p=2\pi\int_{r_s}^{0}qrdr \qquad 1.59$$

where q is evaluated from the equation $q(r)=A_{IB}F(z)$, where F(z) is the power flux at depth z, and A_{IB} is the absorptivity of the plasma.

1.4.5.1.2 Fresnel absorption at the keyhole walls

Part of the light arriving at the keyhole wall will be absorbed, this fraction of the laser power can be calculated from the Fresnel formula (equation 1.2), the amount of power absorbed per unit depth by this method can be shown to be.

$$Q_F=PA(\theta)\frac{2}{r_s}\frac{dr_s}{dz} \qquad 1.60$$

where $A(\theta)$ is the fraction of the power absorbed, calculated from the Fresnel formula.

1.4.5.1.3 Heat lost by conduction

In order to recalculate the heat lost by conduction into the workpiece, a full numerical treatment of the fluid flow and temperature distribution in the weldpool would have to be undertaken. This would take up too much time, and so in most models of laser welding, estimates are made in order to calculate the heat lost by conduction into the material, as well as with the temperature distribution in the workpiece. The heat conducted away from the material is

$$q=-k\frac{\partial T}{\partial n} \qquad 1.61$$

Steen et al (1988) and Akhter et al (1989) showed that the energy coupled into the workpiece could be modelled by a combination of a line and point source (figure 1.15). This inspired a number of workers, (eg Shuoöcker, (1990), Dowden et al (1989)) to calculate the value of the line source from first principles. Kaplan (1994) modelled the temperature in the weldpool by a combination of line sources down the keyhole, whose radial position was not necessarily along the beam axis. Ducharme et al (1995) modelled the heat conducted into the workpiece using the Davis-Noller solution described earlier.

1.4.5.1.4 Heat lost in the vaporisation process

The power absorbed per unit depth in ablating the material at the keyhole wall can be calculated from

$$Q_{abl}=2\pi r_s \rho_g u_g \Delta L \qquad 1.62$$

where ρ_g and u_g denote the gas density and the normal gas velocity evaluated at the keyhole walls respectively, and ΔL is the latent heat of vaporisation. The only way to calculate the gas velocity at the keyhole walls is to evaluate the flow conditions within the keyhole, this increases the computational time immensely, and hence this term is neglected or oversimplified in most treatments to date.

1.4.5.2 Pressure balance at the keyhole wall

As well as the energy balance, some studies also deal with the pressure balance at the keyhole walls as well (figure 1.14b). This is described by a balance between the pressure in the weldpool trying to close the keyhole, together with the pressures in the keyhole keeping the keyhole open. The overall balance is described by,

$$p_{abl}+\delta p_g = p_\gamma + p_{hyds} + p_{hydd} \qquad 1.63$$

The terms in this balance are described individually below.

1.4.5.2.1 The ablation pressure

The ablation pressure is the recoil pressure on the keyhole walls due to vaporisation, and can be calculated by

$$p_{abl}=\rho_g u_g^2 \qquad 1.64$$

1.4.5.2.2 The eccess pressure in the keyhole due to the fluid flow

The excess pressure in the keyhole due to the fluid flow, δp_g can either be estimated or calculated numerically from the Navier Stokes equations. Klein et al (1994) estimated this pressure to be,

$$\delta p_g \approx \frac{1}{3}\rho_g u_g^2 \left(\frac{d}{a}\right)^2 \qquad 1.65$$

where d and a are the keyhole length and radius respectively.

1.4.5.2.3 Surface tension

The pressure due to surface tension is defined as

$$p_\gamma = \frac{\gamma}{r} \qquad 1.66$$

assuming that the curvature effects in all but the radial direction are negligible.

1.4.5.2.4 Hydrostatic pressure

The hydrostatic pressure in the weldpool has a tendency to close the keyhole and can be evaluated from,

$$p_{hyds} = \rho_l g z \qquad 1.67$$

where P is the density of the liquid metal, g the acceleration due to gravity, and z the depth down the keyhole. For thin materials, it can easily be shown that this effect is negligible; considering welds in 5mm steel as a typical example, the density is equal to 7200kg/m^3, g=9.81 m/s^2 and the maximum value of z is 0.005m, this gives a maximum value of hydrostatic pressure equal to 353 Pa, which is 2 orders of magnitude below the pressure due to surface tension, and hence many researchers (eg Klein et al (1994), Kroos, Gratzke and Simon (1992) have justifiably neglected this value.

1.4.5.2.5 Hydrodynamic pressure

The hydrodynamic pressure in the weldpool will also tend to close the keyhole. Again, in order to evaluate this value properly, a full numerical treatment of the conditions within the weldpool would be required. However, approximate solutions can be obtained using some simplifying assumptions. Kroos, Gratzke and Simon (1992) followed Lanbdau and Lifshitz (1959) by estimating the pressure on the keyhole as equal to the pressure difference on a cylinder when ideal fluid with velocity v, flows past it, this gives,

$$p_{hyd}=\frac{\rho}{2}v^2(4\cos^2\theta-3) \qquad 1.68$$

Putting in typical values for steel, Klein showed that this was of order 1, and hence this term was neglected. Beck, Berger and Hugel (1995) used another approximate formula by neglecting viscosity and using Bernoulli's equation, which gave values of hydrodynamic pressure under 50Pa for speeds of up to 5m/min, and so again the authors justifiably neglected the hydrodynamic pressure.

Kroos, Gratzke and Simon, (1992 a) also give an estimate for the hydrodynamic pressure for viscous flows using the model of Dowden, Davis et al (1983), which calculated the fluid flow in the weldpool using the assumption that the keyhole and weldpool could be described as two cylinders of radius a and R respectively, for small Reynolds numbers, this gives

$$p_{fl}-p_b=-\frac{\rho}{2}v^2\left(\frac{8}{Re}\alpha\cos(\theta)\right) \qquad 1.69$$

where

$$\alpha=\frac{2r\left(\frac{a}{R}\right)^2+\left(\frac{R}{a}\right)^2}{\left(\frac{a}{r}\right)^2\left(1+2\ln\left(\frac{R}{a}\right)\right)-\left(\frac{R}{a}\right)^2\left(1+2\ln\left(\frac{a}{R}\right)\right)} \qquad 1.70$$

Here, θ is the angle in polar coordinates and p_b is a constant determined by the boundary conditions at the fluid solid interface. Using typical values of processing parameters for the laser welding of thin steel, the authors deduce that the pressures on the keyhole wall due to hydrodynamic forces is of order 1. Following these studies it will be assumed in the rest of this thesis then that this pressure is negligible.

1.4.5.3 Previous Models for Laser Keyhole Welding

Andrews and Atthey (1976) modelled a pressure and heat balance in the laser generated keyhole walls in order to describe the laser drilling process. They assumed that the pressure balance could be described by the balance between the surface tension, the hydrostatic pressure, and the ablation pressure. They also assumed that the vapour pressure inside the keyhole was equal to the atmospheric pressure. They prescribed the energy balance at the keyhole wall, assuming that the energy from the beam was fully absorbed, and therefore this energy was completely used up in the vaporisation process, thus neglecting conduction losses into the workpiece. The authors considered two different situations in this model, the first being when gravity was the only restoring force. In this case the pressure balance was described by neglecting viscous forces and using Bernoulli's equation to describe the flow out of the keyhole, and by employing the conservation of mass condition at the liquid/vapour interface. The resultant equations, together with the energy balance and

the correct boundary conditions, were used to obtain the keyhole dimensions. The authors noted that there were two distinct, physical regimes present. In the case of shallow holes, the depth was determined by the power density, and for deep holes the depth was determined by the power per unit radius. In the second model, surface tension was added to the pressure balance, this resulted in a reduction of penetration depth by a factor of 3. It was found that the depth was determined by two dimensionless parameters, $Q=P_d/(g\rho_g\rho\Delta L^2a)^{1/2}$ and $\tau=\gamma/(\rho ga^2)$ where P_d is the power density of the beam, a is the beam radius, ρ and ρ_g are the densities of the liquid and vapour state respectively, ΔL is the latent heat of vaporisation and γ is the surface tension gradient.

Steen et al (1988) and Akhter et al (1989), analysed a number of welds, and showed that the energy coupling into the beam could be described in terms of a point and line source. Since then, attempts have been made to determine the value of this point and line source, from first principles. A very useful attempt to calculate the line source strength was presented by Dowden, Kapadia and Postacioglu (1989), who calculated the absorption in the plasma in a qualitative way, neglecting the Fresnel absorption mechanism. The model assumed that the plasma was fully ionised (i.e. Z=1), and all the ions were considered at rest but allowance was made for the mutual interaction between electrons. Energy was transferred to the workpiece by an electron conduction process, which occurred in the radial direction only, and the source term in the conduction equation was given by the amount of absorption of laser light at that position. The authors obtained an expression for the linking intensity L which is defined by $Q^2=LP$ where Q is the power absorbed per unit depth, and P is a measure of the power in the form of optical energy at a given cross section. The authors calculated the value of L for steel at atmospheric pressure, and with a CO_2 laser beam. This was found to be $3.56KWcm^2$.

Kroos et al (1992 a) studied the stability of the cylindrical keyhole at low Peclet numbers, by using a pressure and energy balance at the keyhole wall. They assumed the keyhole could be described as a cylinder, which is valid at low Peclet numbers, and that the pressure balance in the keyhole could be described in terms of the surface tension and the ablation pressure on the keyhole wall. They didn't specify the absorption mechanism involved in the keyhole, but assumed that allthe power was absorbed uniformly around the keyhole, this energy input was used up in vaporising the material, and conduction of heat away from the keyhole and into the workpiece. The two balance equations were solved in order to find the keyhole radius as a function of process parameters and workpiece dimensions. Figure 1.16 shows how the ablation and surface tension gradients vary with increasing keyhole radius a, for this system. For small values of keyhole radius, the laser power entering the keyhole increases with increasing radius, and then saturates for large values of keyhole radius, therefore the absorbed energy flux and hence the ablation pressure decrease with increasing beam radius. The authors noted that the equilibrium values of keyhole radius are at the intersection

of the ablation and surface tension pressure curves, here it can be shown point B is the stable one. The authors then went on to calculate the various values of keyhole radius for various process parameters, and showed that for lower powers the keyhole cannot be maintained due to the fact that there is no point were the pressure curves meet, and hence there are no stable points for this system. They also calculated that the ratio of the keyhole radius to beam radius for various metals near the threshold power, this ratio was approximately 1.7 for all metals, indicating the marked affect of the laser beam size in laser keyhole welding.

Kroos et al (1992 b) developed this analysis by also taking into account the flow in the weldpool. It was assumed that the keyhole and weldpool could be described by two concentric cylinders of radius r_1 and r_2, this is valid for low Peclet numbers and thin workpiece thicknesses. The authors then solved the velocity in the weldpool in terms of radial coordinate and time, and applied a pressure balance at the keyhole wall; after eliminating r, this led to an equation that enabled the authors to calculate keyhole radius as a function of material properties. They then applied this analysis to the case of the collapse of the keyhole when the laser was switched off, assuming that the process is completely surface tension driven. They calculated the time for the radius of the keyhole to reach zero after the ablation pressure and power were set to zero, and found that for iron, copper and aluminium, the keyhole collapse time was of the order of 0.1ms. They then carried out a stability analysis on the system by examining the effect of small perturbations of two functions on the keyhole radius. The process was found to be stable for small perturbations in the system but was vulnerable to large, finite perturbations. The authors calculated the typical frequency of the process at powers just above the threshold power, and found it to be around 500Hz, this frequency was only weakly dependent upon the material.

Klein et al (1994) further developed the above model by describing the free oscillations in the cylindrical keyhole classified into radial, axial, and azimuthal modes (figure 1.17). They began their analysis by evaluating the behaviour of the steady state system by applying an energy and pressure balance at the keyhole wall similar to Kroos, Gratzke and Simon (1992 a), with one exception. In this study, the excess pressure due to fluid flow in the keyhole was also taken into account in the pressure balance equation, and estimated using equation 1.65. The authors carried out an analysis of radial modes and round that typical eigenfrequencies were in the range of 1500-2000Hz, i.e. up to four times greater than the earlier work of Kroos, Gratzke and Simon (1992 b), this was mainly due to the fact that in this case, the excess pressure inside the keyhole was taken into account. The authors then investigated the eigenfrequencies associated with the axial and azimuthal deviations, where the keyhole radius was set to $r=a+\cos(n\theta)\cos(kx\alpha(t))$; the calculations showed that there exists a whole spectrum of frequencies associated with different values of n and k. It was shown that for small workpiece thicknesses the frequencies rapidly increase with increasing n and k, however

as the thickness becomes larger, the eigenfrequencies show only a weak dependence on 1 because the associated curvature in the axial direction becomes negligible compared to that in the azimuthal direction. the authors again show that the system was stable for small perturbations above the threshold power, however, the system is susceptible to finite deviations in process parameters, but the higher the power, the more stable the system.

Ducharme et al (1993) also calculated the collapse time of the keyhole following the extinction of a CO_2 laser beam. They considered the melt/solid interface to be a cylinder much larger than the keyhole, whose profile was calculated from the integrated keyhole and weldpool model of Ducharme et al (1992). They then solved the momentum and continuity equations in the weldpool using a 2 dimensional, time dependent finite volume formulation (Patankar (1980)), while assuming that the pressure in the keyhole drops to ambient in a time much shorter than the collapse time of the keyhole, which is assumed to be purely surface tension driven. The radial velocity of the keyhole was calculated as a function of time, and the collapse time was then deduced from the initial conditions. The authors concluded that the collapse time was dependent upon the process conditions before the laser is switched off. For slow welding speeds, when the keyhole was almost a perfect cylinder, the keyhole collapses at the centre, whereas at much higher speeds, where the keyhole is more conical, the keyhole collapses at the bottom. Typical values of collapse time were found to be of the order of 0.1ms which was consistent with the analysis of Kroos, Gratzke and Simon (1992 b).

The use of cylindrical symmetry in many of the models described here meant that the high Peclet number case could not be taken into account. Kaplan (1994) following Schuöcker (1990) suggested an approach to rectify this problem by calculating the keyhole coordinates at discrete intervals down the keyhole both at the front and back of the keyhole using cartesian coordinates, but then switching to polar coordinates to describe the temperature in the workpiece and the energy balance at the keyhole walls. The heat input into the workpiece was equal to the sum of the Fresnel absorption at the keyhole walls (directly from the beam and by multiple reflections) and the heat gained from the inverse Bremsstrahlung absorption mechanism, which was modelled by taking an average absorption coefficient throughout the plasma. The heat conducted into the material was described as a distribution along the z axis of line sources. He observed that most of the beam hit the front wall whereas the rear of the wall was formed by the low intensity region of the beam, where convective heat transfer ensures a high temperature anyway. He also calculated that the plasma absorption in the keyhole is much less than the Fresnel absorption at the keyhole walls.

Ducharme et al (1995) developed an integrated mathematical model of the keyhole and weldpool during laser keyhole welding of thin sheets. They carried out this analysis by employing an energy balance at the keyhole wall by taking into account the absorption of the plasma inside the

keyhole, together with direct absorption of the laser beam at the keyhole walls. The sum of these two terms was set to equal the heat conducted away into the material, which was calculated using the Davis-Noller solution. The power input was assumed to be uniform throughout and the effects of multiple reflections were neglected. This analysis yielded a first order differential equation for the keyhole dimensions, which was solved using standard numerical techniques, with the boundary condition that the keyhole radius at z=0 was equal to the laser beam radius. A mean value of keyhole radius was then calculated and used as the cylinder radius in the Davis Noller equation in order to find the temperature distribution in the workpiece; the authors thus assumed that the flow in the weldpool together with axial conduction and the latent heat all have a negligible effect upon the temperature distribution, this is a valid assumption for thin welds only. In order to validate the model, a high speed film was made of the top weldbead dimensions during the welding process which was compared with the theoretical predictions of weldpool dimensions from the model (see figure 1.18), with considerable success.

1.5 Summary

This chapter has reviewed the current state of modelling laser heating, melting and welding processes. A summary of these models can be found in tables 1.1 to 1.4. Obviously, we have come a long way since the days of Rosenthal, the advancements in computer technology have made it possible to solve the 3 dimensional heat conduction equation with complex boundary conditions quite easily. Even so, there are a many areas in laser processing that are still not understood fully. Two models are developed in chapter 2 that describe the comparatively little studied areas of pulsed laser welding and machining. These models solve the heat conduction equation using a variation of the enthalpy method, in order to obtain the position of the ablating and melting surfaces. The results from both models are compared with experimental data, and they are then run in order to gain a better understanding of the processes.

Many aspects of the CW laser welding process have been studied. However, even with the advancements in computer technology, programs that describe the physics of the process in detail can take many hours or even days to run on even the fastest computers. Hence, there is still a need to develop simpler models, that can run in a fraction of a second on a PC. The first model presented in chapter 3 calculates the keyhole profile by applying an energy and pressure balance at the keyhole walls, in order to estimate the size of the ablation term in the energy balance. In order to evaluate the pressure balance properly, the flow in the keyhole has to be solved, this takes a few hours to run on a mainframe computer. Hence, once the effect of ablation on the overall process has been evaluated, a simpler and much faster model is developed that neglects this term. This model is then validated by comparing the theoretical results with experimental data presented in chapter 4. This chapter also contains a theoretical and experimental investigation into the effects of varying a number

of process parameters on the laser welding process. Chapter 5 concludes this thesis by summarising the main conclusions of each chapter, and outlining further work in these areas that needs to be undertaken.

Author and Year	Scope of the model and comments
Rosenthal, 1946	The author solves the heat conduction equation in order to obtain the temperature distribution for point and line heat sources.
Cline and Anthony, 1977	This paper calculates the 3-dimensional temperature distribution in a semi infinite solid, with a Gaussian beam as a heat source.
Gratzke et al, 1991	The authors developed the work of Cline and Anthony in order to account for the finite thickness of the workpiece, using asymptotic solutions for the series solution of the temperature distribution gained from applying the method of images.
Ryalikin et al, 1978	The authors assumed that for the high speed laser heating case a Gaussian beam could be approximated by a line source on the surface of the workpiece, finite in the y direction, and infinitesimally small in the x direction.
Ashby and Easterling,, 1984	This paper develops the work of Ryalkin by assuming that the line source is slightly above the surface, and by including an empirical factor to account for the finite thickness of the beam in the x direction. They combined these solutions with models of carbon diffusion in order to describe laser transformation hardening.
Ashby and Shercliffe, 1986	Developing the above work, the authors modify the location of the line source, as well as reducing the solution for the temperature distribution, with the aid of dimensionless groups, in order to produce master plots which characterise the laser transformation hardening process.
Davis et al, 1986	Following Carslaw and Jaeger (1959), the authors solve the heat conduction equation in a semi infinite workpiece subject to a scanning Gaussan beam. They used the novel approach of using a representative value of specific heat dependent upon the process parameters in order to obtain the hardened depth.
Davis, 1983 Noller, 1983	Both authors, working independently, modelled the heat flow in the weldpool in laser keyhole welding by assuming the situation could be described by a uniformly hot cylinder at the vaporisation temperature, surrounded by a solid material moving at a constant velocity.
Steen, 1976	The author modelled the laser heating process with a stationary Gaussian beam using the finite difference method.
Mazumbder and Steen, 1980	The authors modelled the laser heating process, with a scanning Gaussian beam, using the finite difference method.
Kou, 1981, Kou and Li, 1983, 1984	The authors introduced a variation of the enthalpy method in order to solve the GTA welding of aluminium, in order to take into account the latent heat effects. The resulting differential equation was solved using the finite difference method.

Table 1.1 A summary of the CW laser heating models discussed in the first chapter.

Author and Year	Scope of the model and comments
El-Awadi and Elshehawey (1986)	The authors examined the temperature field in a workpiece subject to time dependent laser radiation. in order to calculate the critical power required to initiate melting.
Modest and Abiakans (1986)	The authors compared a 1-dimensional solution for the temperature distribution of a workpiece subject to a pulsed laser beam, with the exact 3-dimensional solution in order to find a window where the simpler solution was applicable.
Simon et al (1993)	The authors modelled the pulsed laser heating process by assuming that the process could be described by a cylinder with varying heat flux moving through a semi infinite body.
Gellert and Egli (1988)	The authors calculated the temperature distribution and the position of the free boundary for pulsed heating of copper, using Langmuir evaporation as a cooling mechanism.
Smurov et al (1991)	The authors calculated the temperature distribution and the position of the evaporating and melting isotherms for various shaped pulses in the kHz region.
Tzeng (1996)	The author furthered the analysis of Smurov et al (1991), by examining the response of the workpiece to pulses in the Hz region.

Table 1.2 A summary of the CW laser heating models discussed in the first chapter.

Author and Year	Scope of the model and comments
Postacioglu et al, 1987	The authors investigated the upward flow of the weldpool due to pressure gradients in the vicinity of the keyhole, in order to evaluate the elevation or depression of the weldbead.
Postaciglu et al, 1991	The authors studied the thermocapillary flows in the weldpool surface by developing a non linear boundary layer model of the thermal conduction and fluid convection.
Gratzke et al, 1992	The authors investigated the mechanisms involved in humping during laser welding, by applying Rayleigh's theory of instability of a free liquid cylinder due to surface tension.
Colla et al, 1992	The authors modelled the 2-dimensional heat and fluid flow within the weldpool around the keyhole, by assuming that the process could be described by an ideal fluid flowing past a hot cylinder. The authors then evaluate the temperature distribution and flow characteristics for such a system.
Oreper and Szekely, 1984, 1987	The authors calculate the transient behaviour of a number of metals heated by a stationary heat source, by solving the governing flow and energy equations using the finite difference method.
Chan et al, 1984	The authors studied the 2-dimenional steady state behaviour of a metal due to a laser beam.
Kou and Sun, 1985 Kou and Wang, 1986	The authors studied the stationary and moving TIG welding process by using a finite difference method to calculate the temperature distribution and flow fields. The authors used the effective viscosity method, which meant that the boundary did not have to be recalculated every iteration, and the enthalpy method in order to take into account latent heat effects.
Tsai and Kou, 1989, 1990	The authors solve the two dimensional heat and mass transfer equations in the weldpool, due to a stationary heat source, but also included a deformed surface due to Marangoni convection (1989) or density variations (1990).

Table 1.3 A summary of the fluid flow models of welding discussed in the first chapter.

Author and Year	Scope of the model and comments
Andrews and Atthey, 1976	The authors used a heat and pressure balance at the keyhole wall in order to describe the laser drilling and welding processes. The heat balance included the power input from the beam which was all used up in vaporisation of the material, and the pressure balance took into account the hydrostatic, ablation and surface tension pressures.
Steen et al, 1988 Ahkter et al, 1989	The authors analysed a number of experimental weld cross sections, and then went onto describe how the power deposited in the keyhole to the material could be described in terms of a combination of point and line sources.
Kroos et al, 1993a	The authors described the laser keyhole welding process using a heat and pressure balance at the keyhole in order to ascertain the radius of the assumed cylindrical keyhole. The heat is assumed to be lost through conduction and vaporisation, and gained by absorption from the laser beam in the keyhole. The pressure balance is described in terms of the surface tension and ablation pressures.
Kroos et al, 1993b	The above model was developed in order to take into account the 1-dimensional flow in the weldpool, in order to calculate the collapse time of the keyhole when the laser is switched off, as well as the evaluating the stability of the system.
Klein et al, 1994	The authors went further by describing the free oscillations in the keyhole classified into radial, axial and azimuthal modes. They apply a heat and pressure balance at the keyhole walls similar to Kroos in 1993, but also estimated the excess pressure in the keyhole due to flow, the authors applied their analysis to evaluate the stability of the system.
Ducharme et al, 1993	The authors calculated the collapse time of the keyhole during laser welding by calculating the flow within the weldpool that drives the process.
Kaplan, 1994	The author calculates the keyhole and weldpool profiles in partial penetration welding, by applying a heat balance at the keyhole walls, taking into account the plasma and Fresnel absorption, as well as the conduction of heat throughout the material.
Ducharme et al, 1995	This paper calculates the keyhole and weldpool profiles for full penetration welding of thin sheets using a laser beam by applying a heat balance at the keyhole walls, which takes into account plasma and Fresnel absorption, as well as the conduction throughout the material.

Table 1.4. A summary of the laser keyhole welding models discussed in the first chapter.

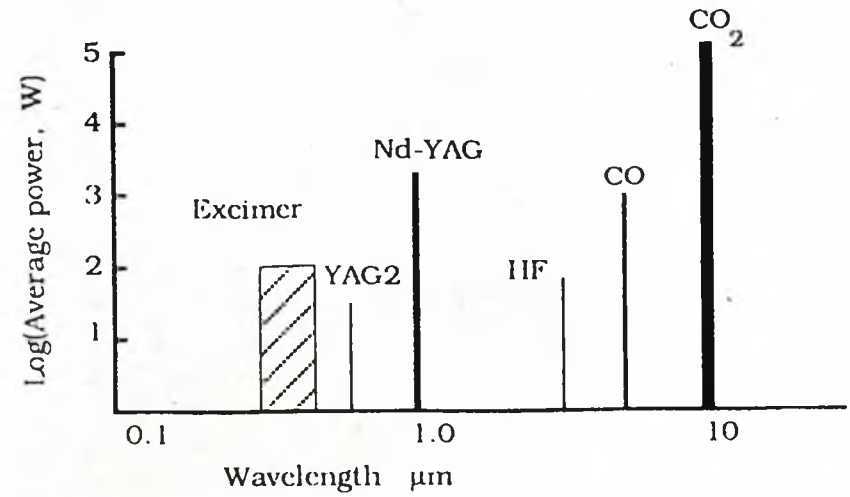


Fig.1.1 The operating bandwidths and output power ranges of various commercial lasers (Steen (1991))

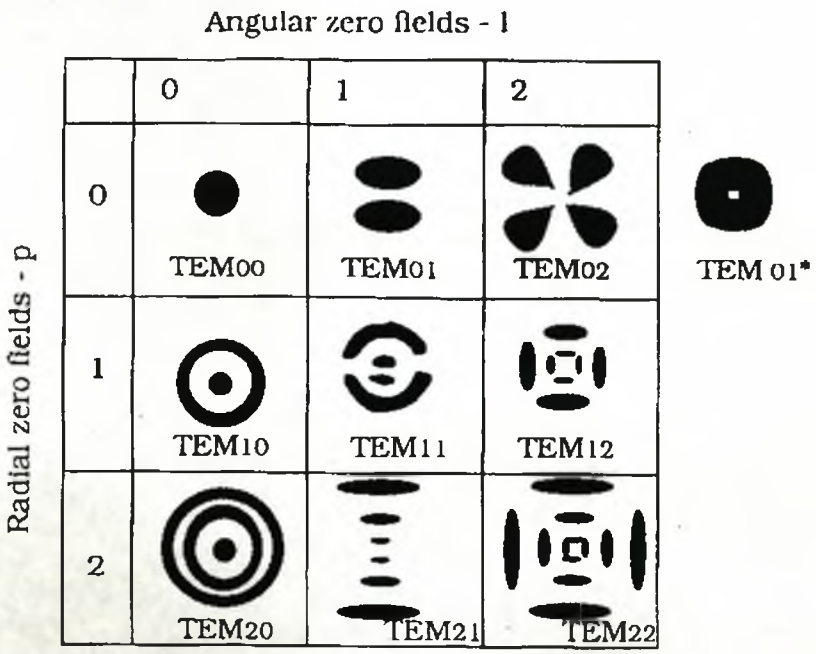


Fig. 1.2 Various mode patterns (Steen (1991))

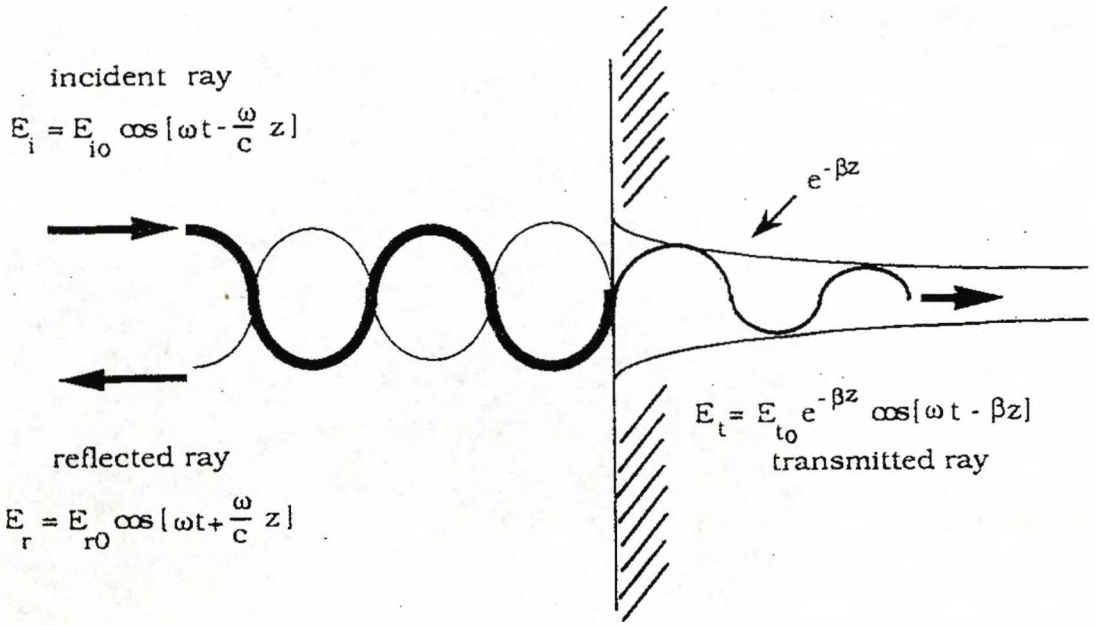


Fig. 1.3 Schematic diagram of electromagnetic radiation striking the workpiece (Steen (1991)).

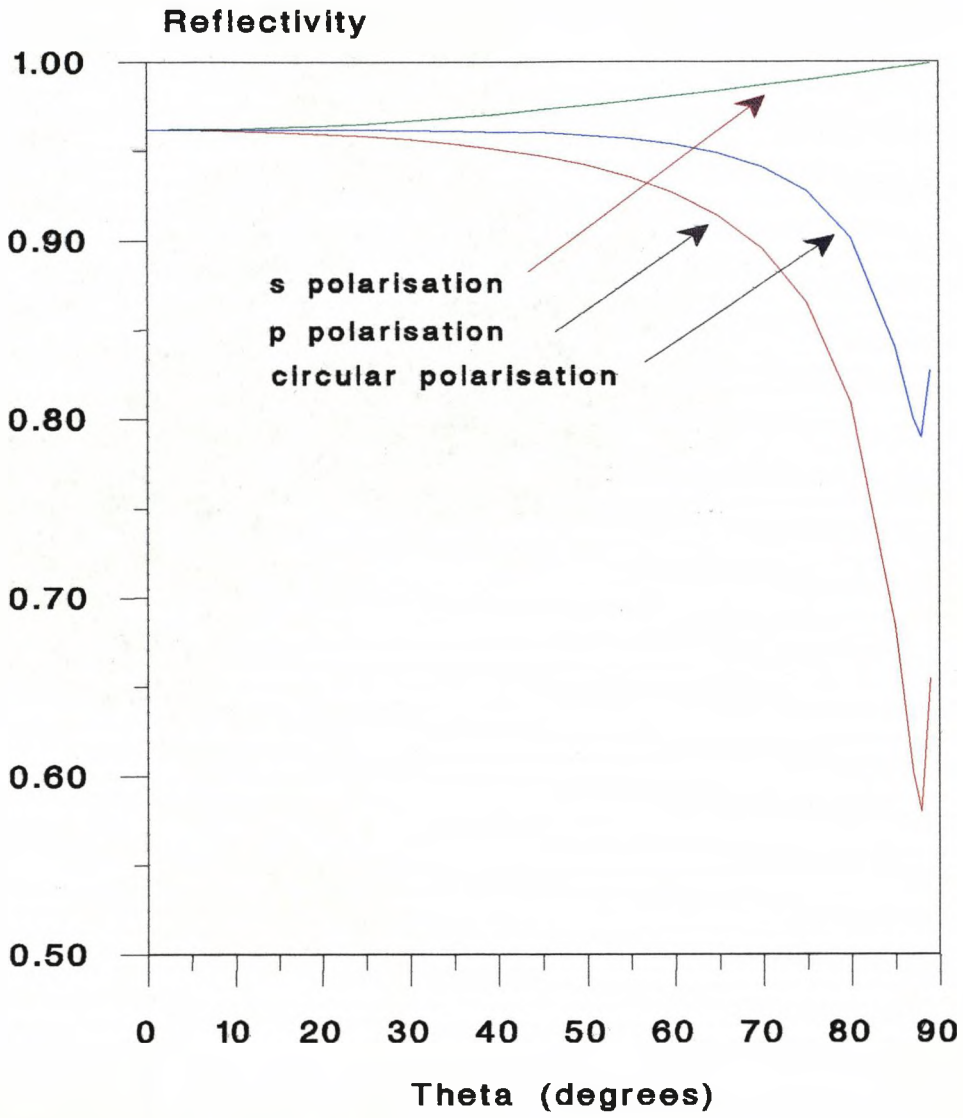


Fig. 1.4 Variation of reflectivity for s, p and circularly polarised light (Steen (1991)).

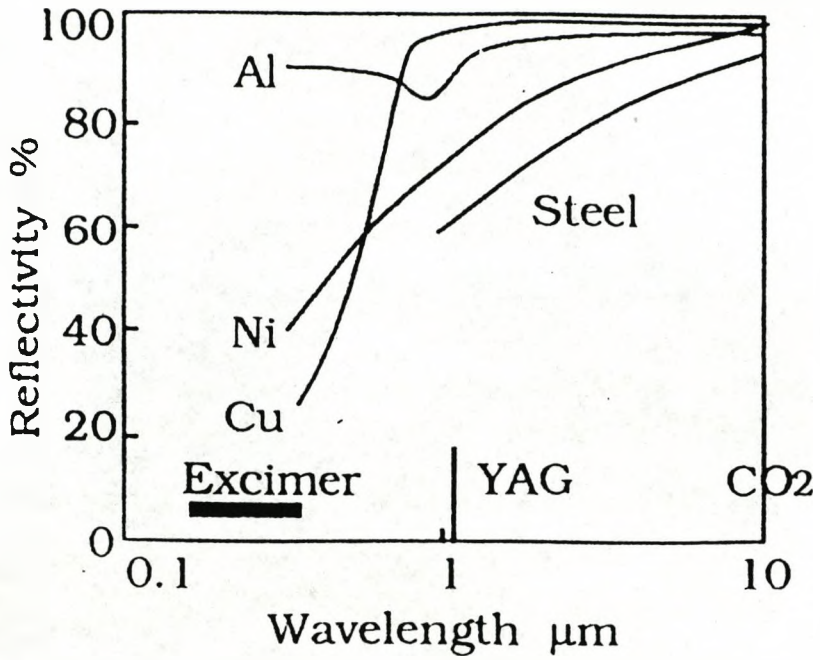


Fig. 1.5 Reflectivity of a number of metals as a function of wavelength (Steen (1991))

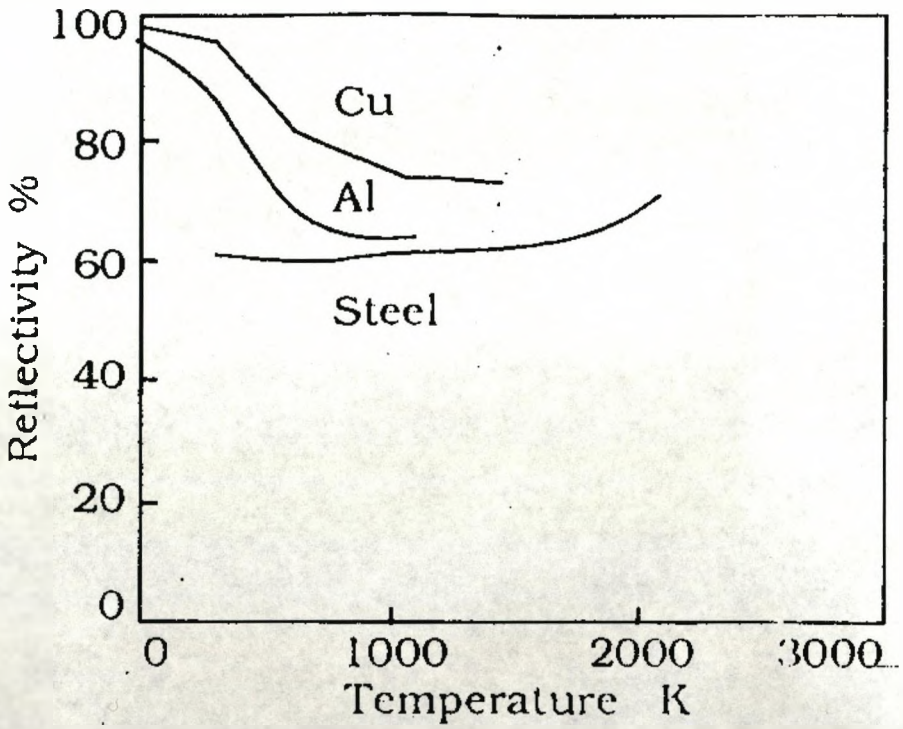
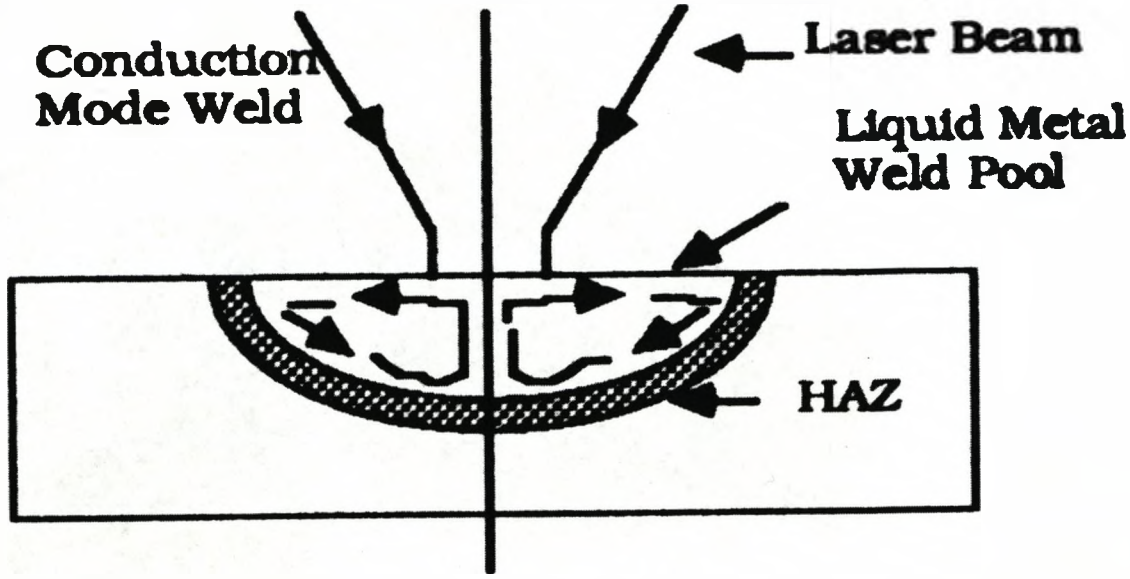


Fig. 1.6 Reflectivity of a number of metals as a function of temperature (Steen (1991))

a)



b)

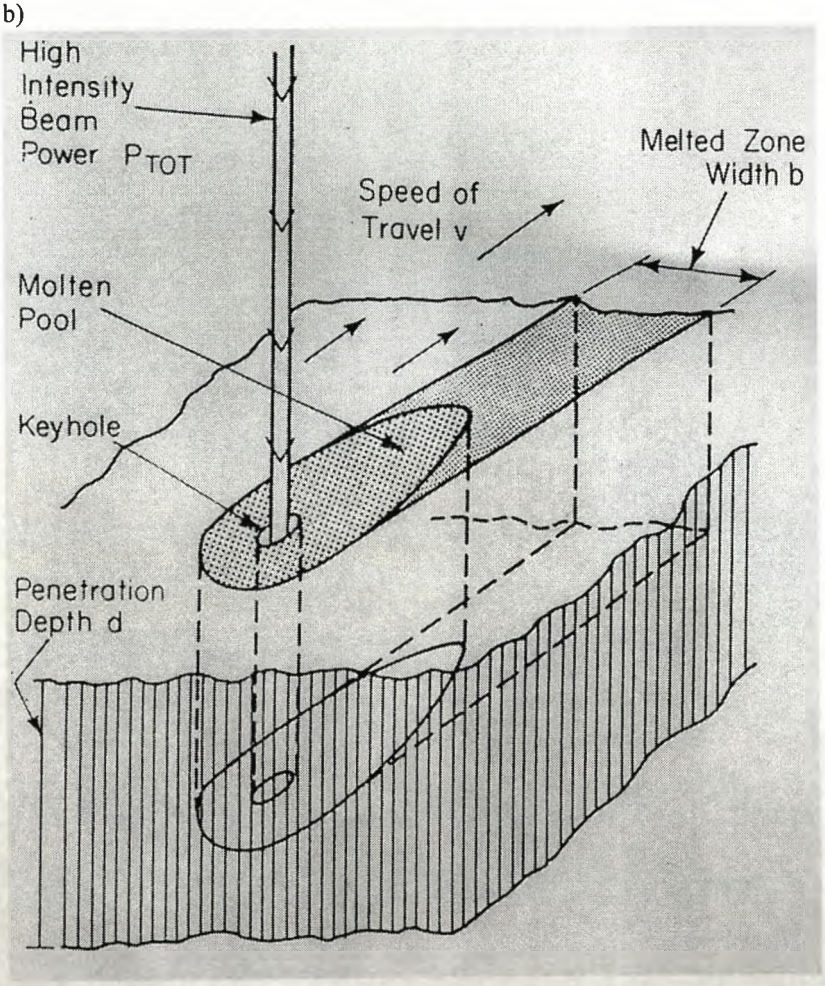


Fig 1.7, The modes of laser welding (Steen (1991)); Bass (1983)), a) Conduction limited welding, characterised by a wide and shallow weld pool, and b) Keyhole welding, characterised by a long and narrow weldpool.

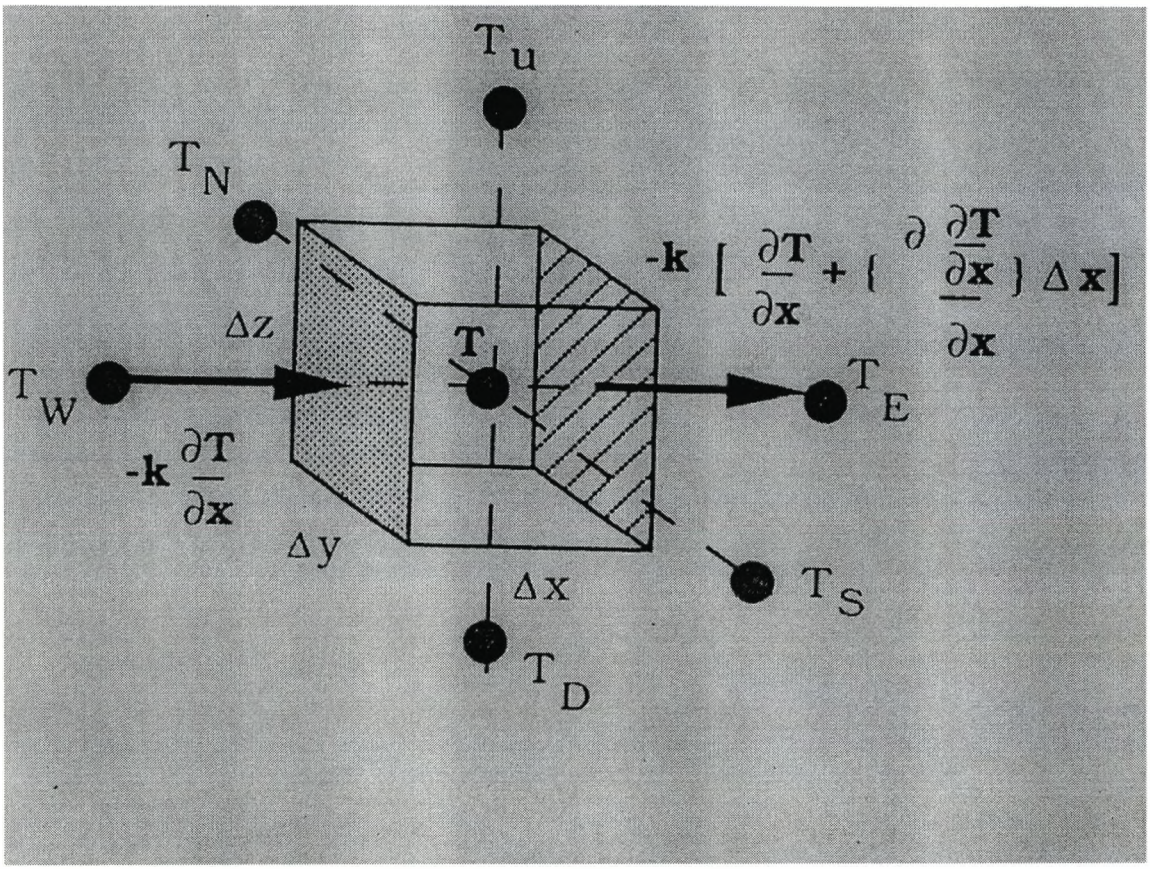


Fig. 1.8. Differential element in 3-space (Steen (1991))

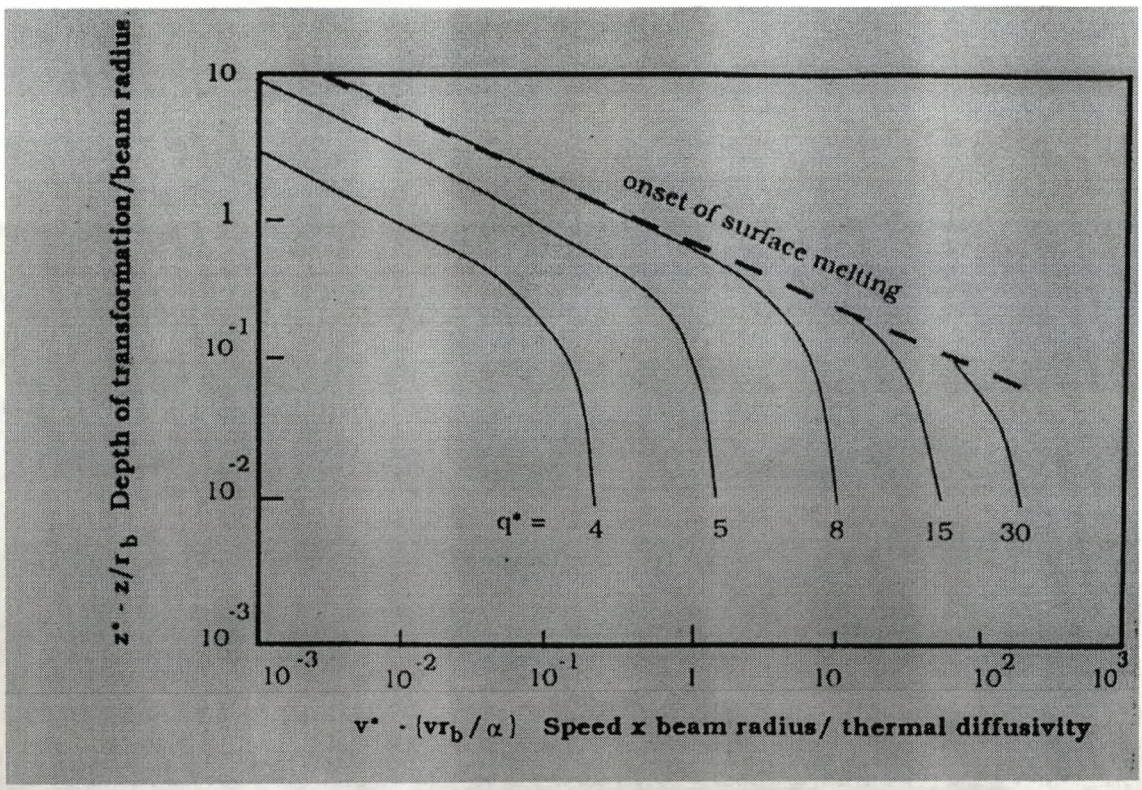


Fig.1.9 Dimensionless master plots for laser transformation hardening (Ashby and Shercliffe (1986))

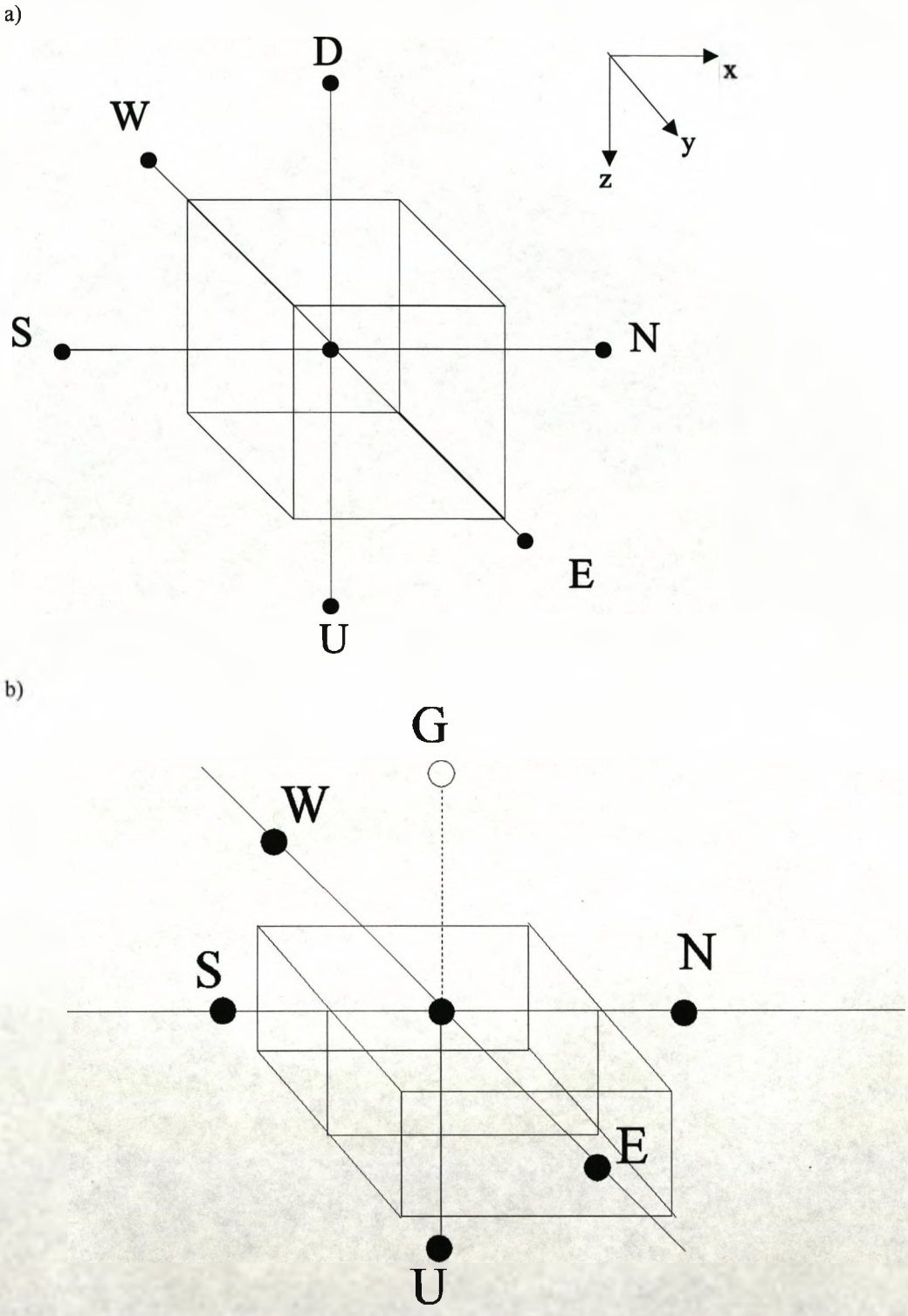
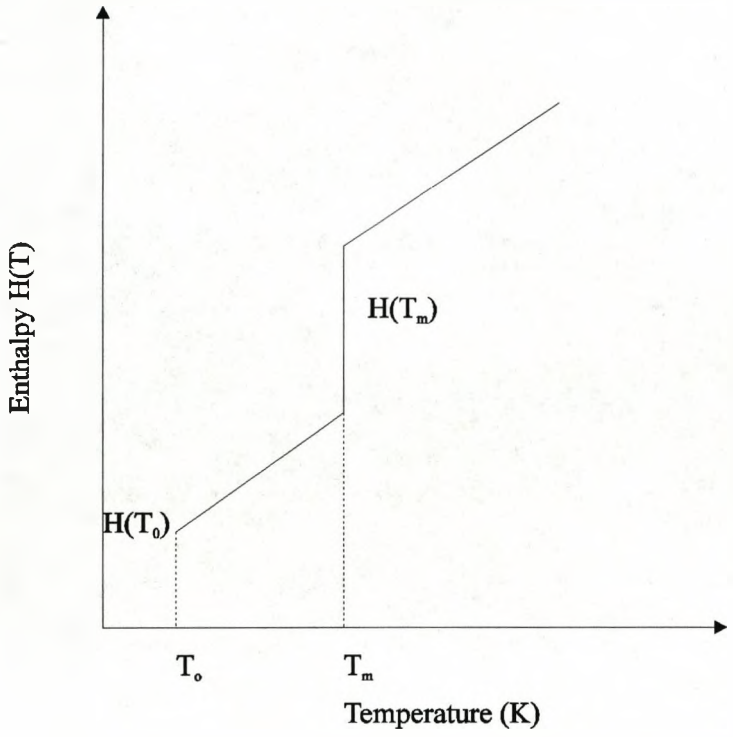


Figure 1.10. Schematic diagram of a finite difference control volume for a) an internal node, and b) a surface node, where the point G is a fictitious point above the surface used to obtain the temperature gradient in the z direction as a function of the power and the convective and radiative losses.

a)



b)

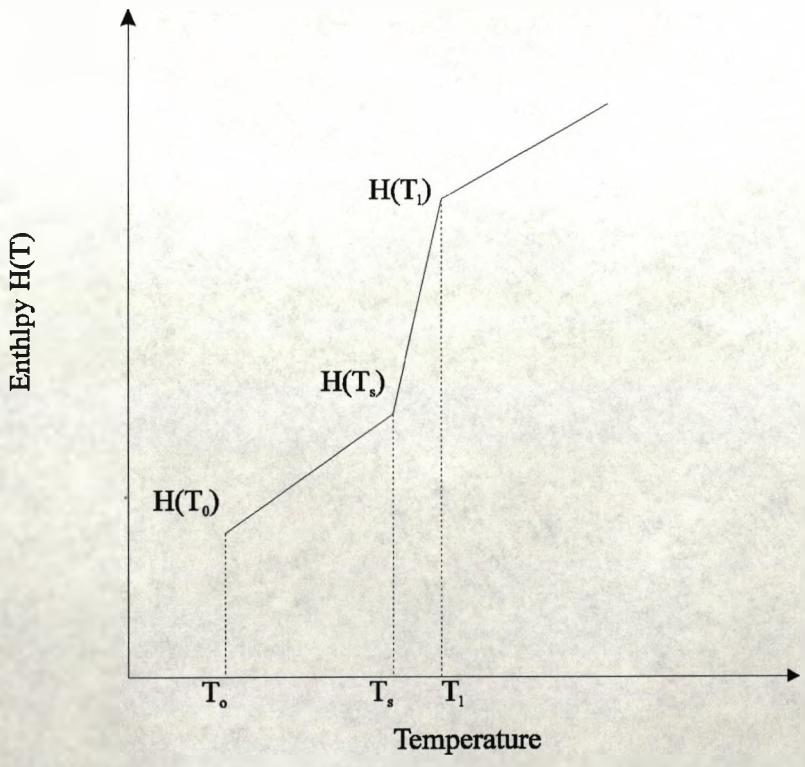


Fig. 1.11 The enthalpy functions proposed by a) Eyers et al (1946) and b) Shamsumder and Sparrow (1975) as a function of temperature.

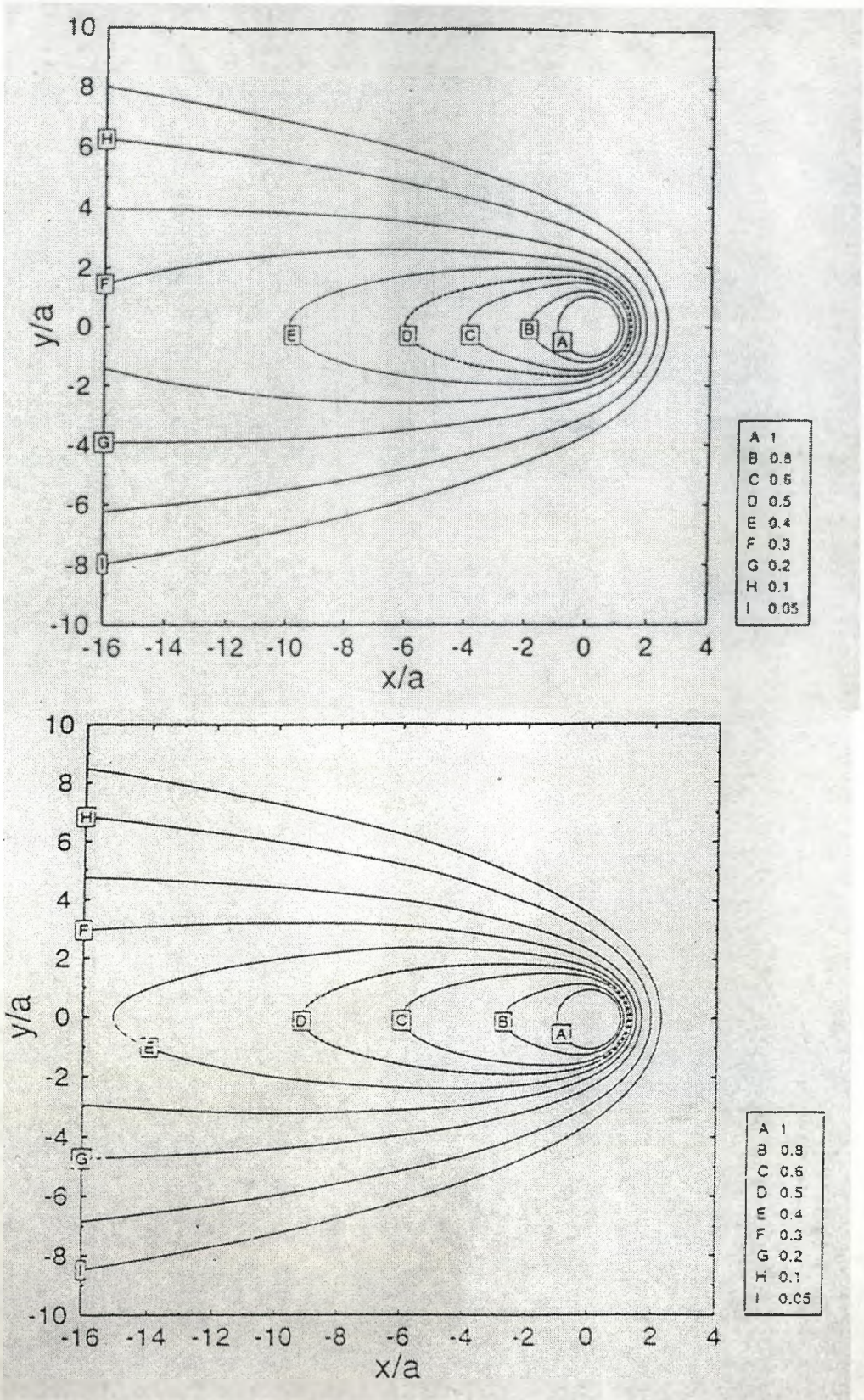


Fig. 1.12. A comparison between the temperature distribution in the weldpool in laser keyhole welding proposed by a) Colla et al (1992), which allows for fluid flow, and b) Davis (1983) and Noller (1983), which only allows for the translation of the workpiece.

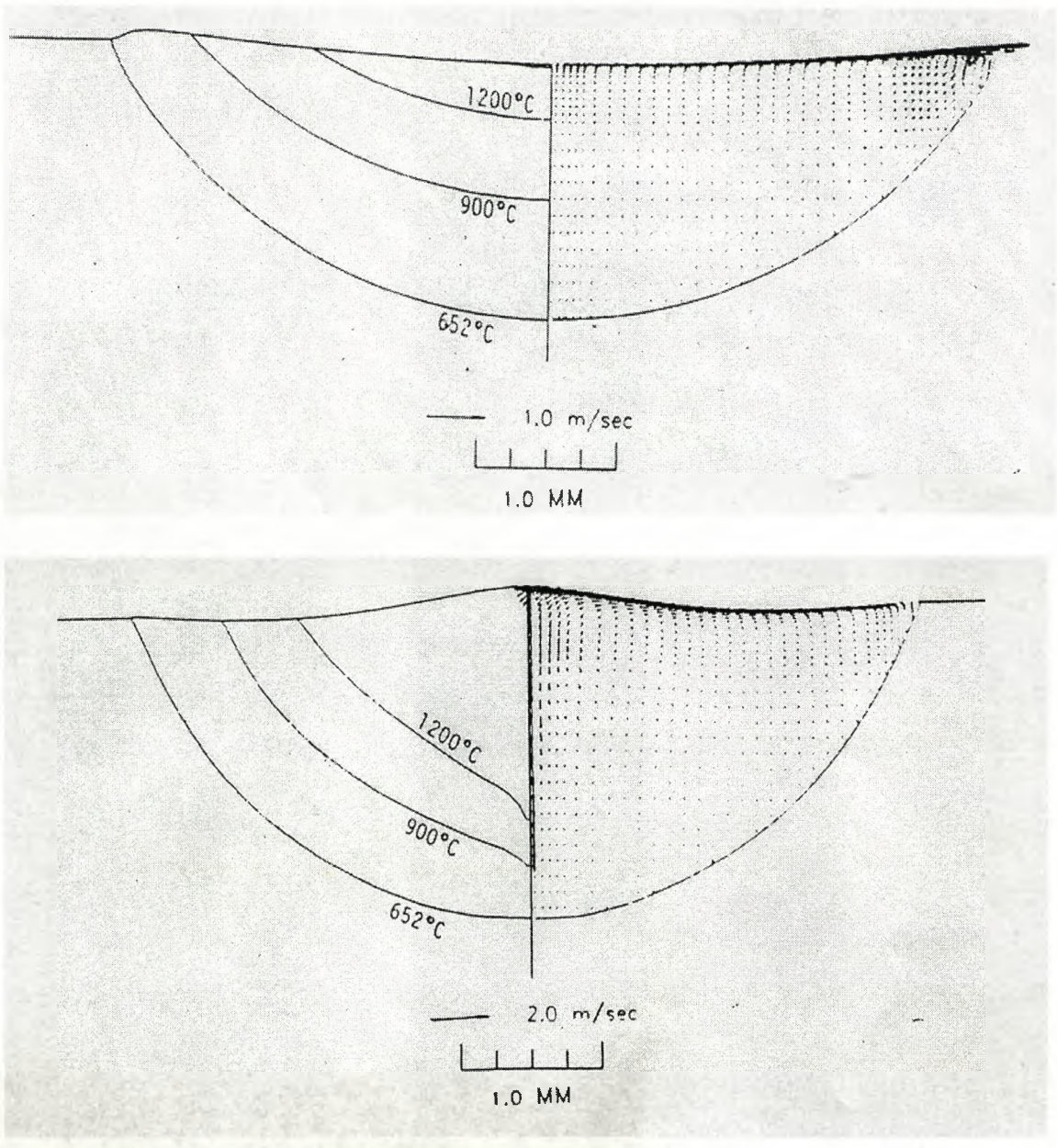


Fig. 1.13 The 2 dimensional shape of the free surface when a) the surface tension gradient is negative, and b) when the surface tension gradient is positive, calculated by Kou et al (1989).

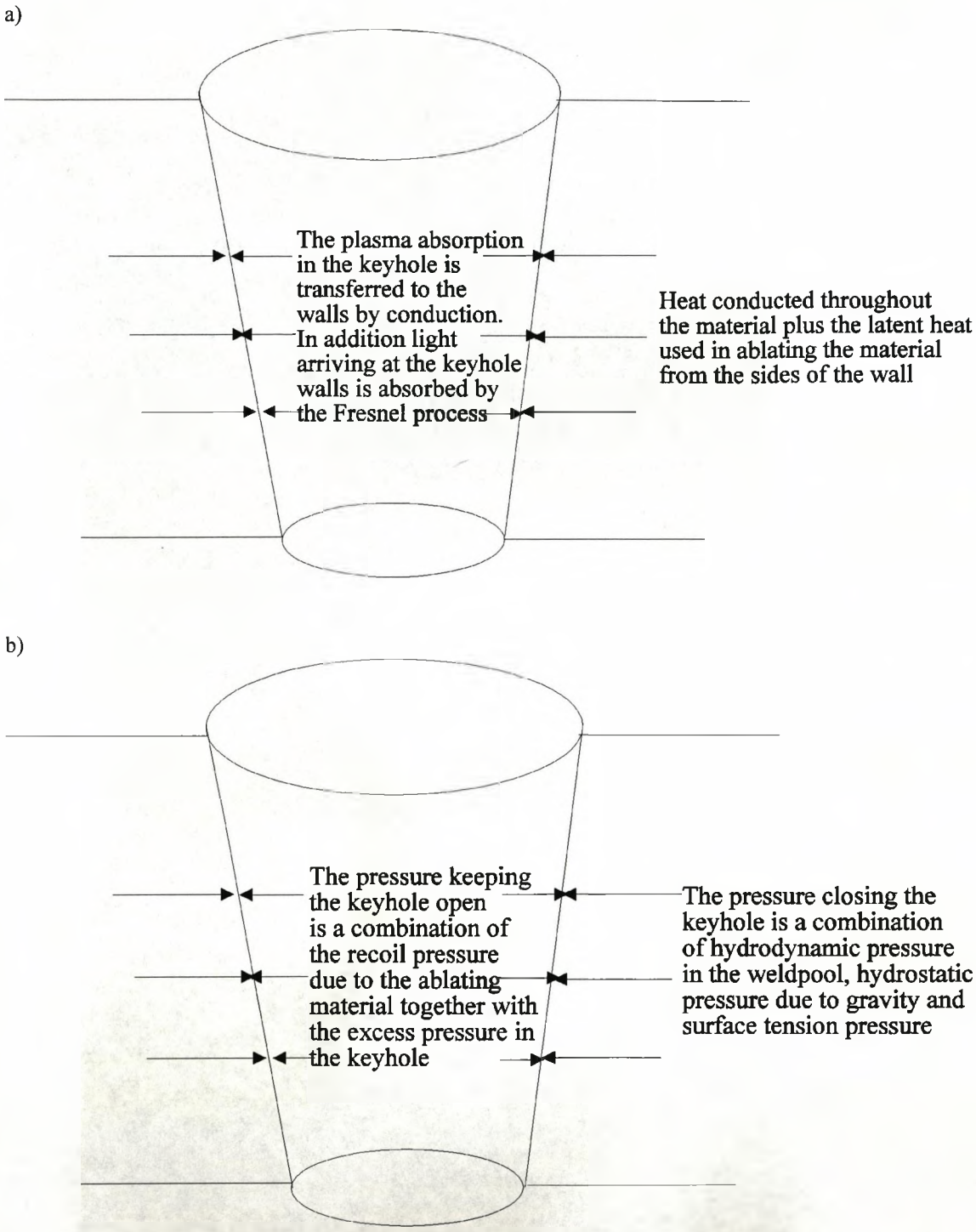


Fig. 1.14 Schematic diagram of the keyhole illustrating a) the energy balance at the keyhole walls, and b) the pressure balance at the keyhole walls.

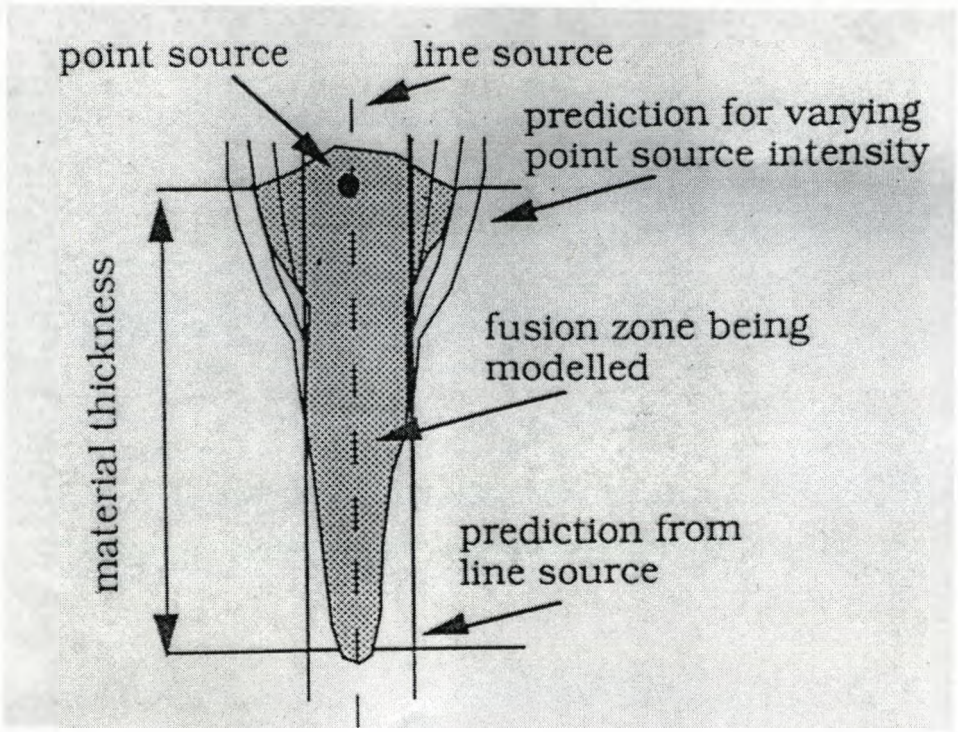


Fig. 1.15. Cross section of a laser keyhole weld showing how the energy deposited into the workpiece can be described in terms of a point and line source (Steen (1991)).

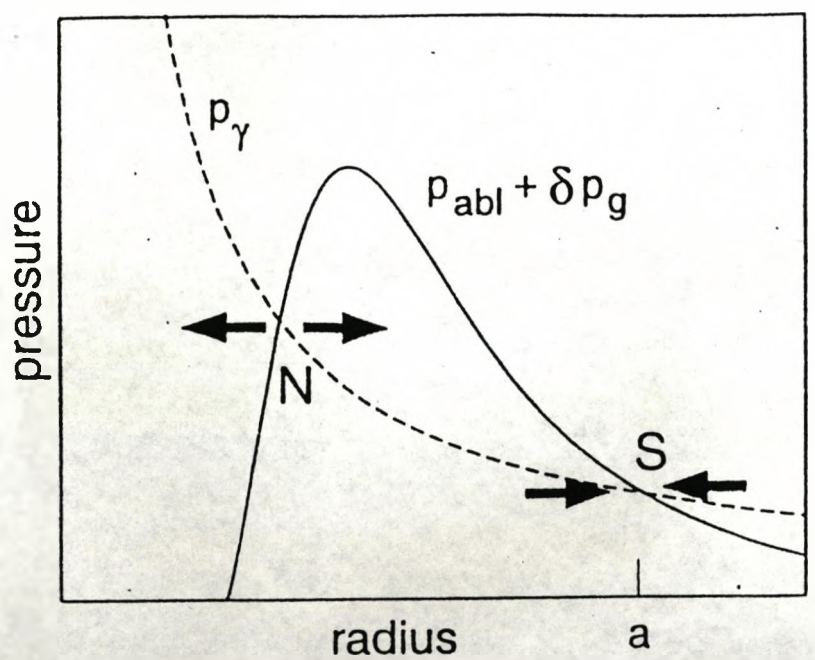


Fig.1.16 Graph of the ablation and surface tension pressure contributions in the laser generated keyhole. The point S denotes the stable point for this system. (Kroos et al (1993))

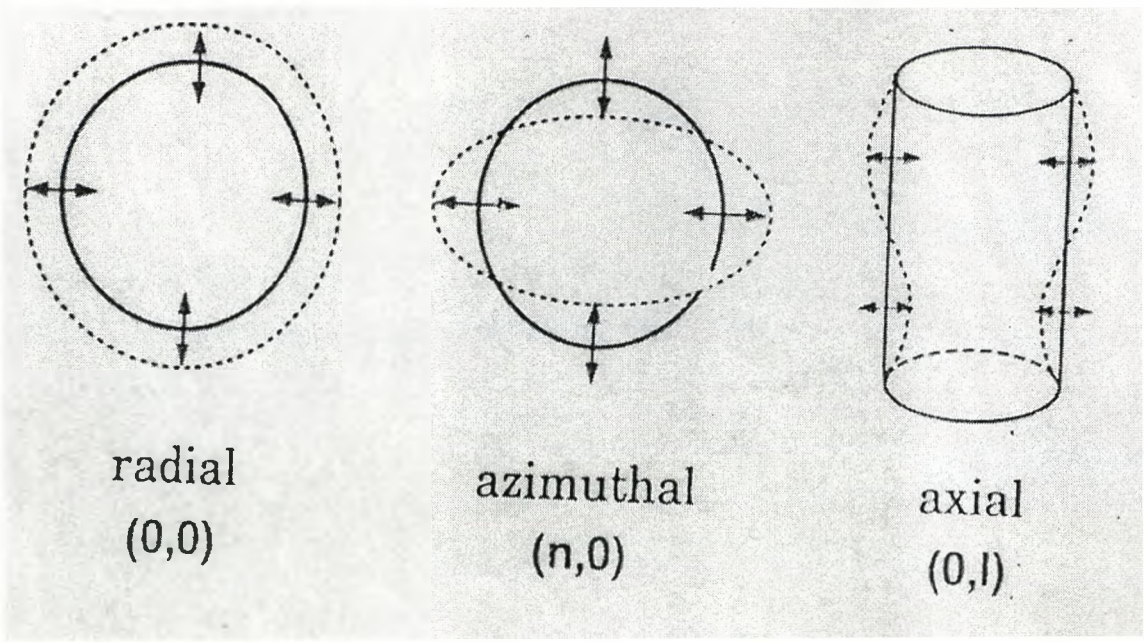


Fig. 1.17 The axial, azimuthal and radial modes of vibration in the laser generated keyhole (Klein et al, 1994).

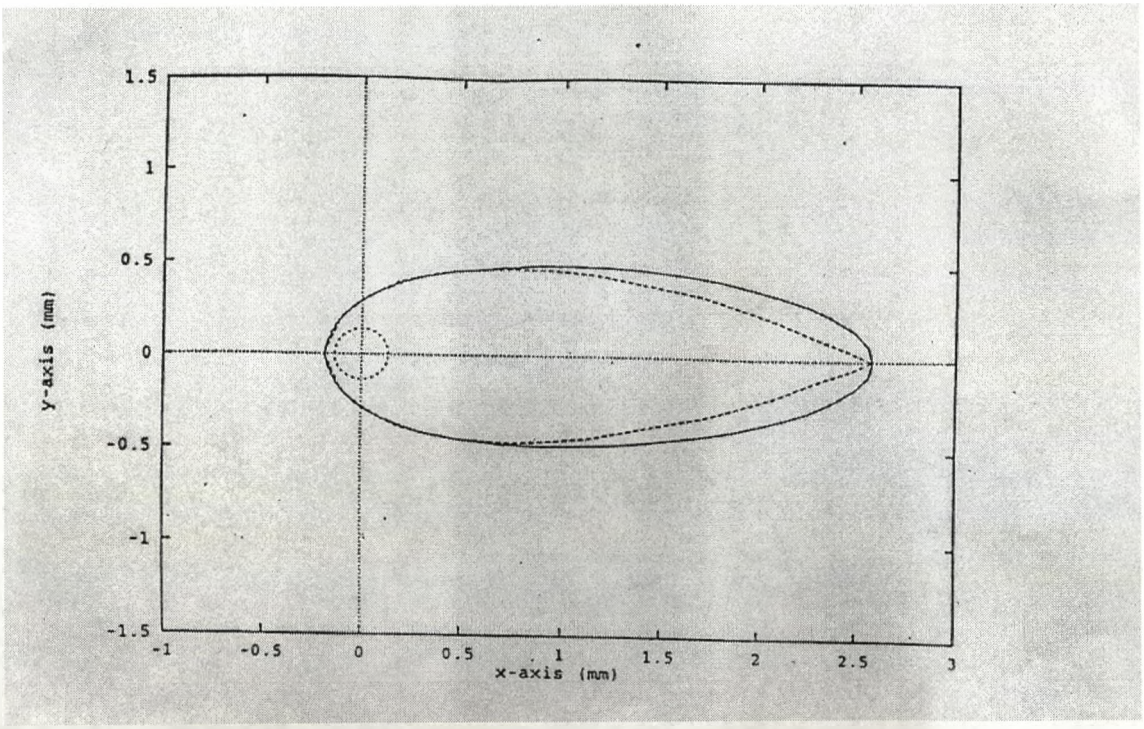


Fig. 1.18 Comparison between the theoretically calculated weldpool dimensions using the Davis-Noller solutions, and the experimentally observed weldpool dimensions by Williams (1995)

Chapter 2 Heat Flow, Melting and Ablation in Pulsed Laser Processing

In the first chapter, the physics involved in the laser heating and welding processes were described in depth, together with survey of previous work. In this chapter, a short recap will be given, and two models will be developed in order to study the pulsed laser welding, and pulsed material removal processes.

2.1 Overview of laser welding

Many industries have used welding techniques in manufacturing for quite a number of years now, however, it is only recently that the laser has become a recognised tool in this field. Currently, both CO_2 and Nd:YAG lasers are used in many welding applications, especially in the automobile, aircraft and shipbuilding industries. There are a number of reasons for the laser's success in these areas, the main ones are,

- Laser keyhole welding can achieve a large penetration depth with a high weld aspect ratio, which reduces the amount of energy required to weld a particular join. This also means that the heat affected zone, and hence the amount of distortion are small compared to conventional techniques.

- The process is faster than traditional welding techniques, and therefore more cost effective.

- Nd:YAG laser light can be delivered to the workpiece via fibre optic cables, and hence the process is easy to automate, and can be used to reach difficult areas when welding 3-d shapes.

There are, however, a number of disadvantages of using a laser for welding,

- At present, the laser is an expensive tool compared to the more traditional welding machines, hence the advantage in cost obtained with faster processing speeds will only be realised if there is a great amount of material to be welded.

- The benefit of gaining a high power density at the workpiece due to the small beam radius, has the disadvantage that a precise setup is required in order to ensure that the beam strikes the workpiece.

- The high cooling rates in the welding process can cause hardness, especially in high carbon steels.

2.1.1 Laser Welding Regimes

There are two main laser welding regimes, which describe how the workpiece transfers the energy from the beam throughout the material, and how the laser radiation is absorbed by the workpiece. The first of these is conduction limited welding (figure 2.1), where a relatively low power density beam is scanned across the workpiece in order to melt the substrate. Here, heat is transferred by conduction (enhanced by convection in the molten pool), throughout the workpiece. This process has many features in common with traditional welding techniques due to the fact that the laser beam behaves like a conventional surface heat source.

The second laser welding regime is known as keyhole welding, which occurs at higher power densities. In the continuous wave (CW) case, a high power density beam is scanned across the workpiece, causing melting and vaporisation; and forming a hole or crater in the substrate, known as the keyhole (figure 2.2). The formation of the keyhole changes the mechanisms involved in the absorption process. This is not fully understood at the moment, and a number of mechanisms have been suggested in the literature (figure 2.3). Fresnel absorption occurs at the keyhole wall from direct laser radiation, and also from multiple reflections in the keyhole. Additional contributions may come from the scattering of light in the keyhole when laser radiation hits small particles, and from plasma absorption from the inverse Bremsstrahlung mechanism. This keyholing effect results in an increase in absorption of laser light, as well as increasing the depth of penetration resulting in a in a high weld aspect ratio. This is one of the main advantages of laser welding. A detailed model of the CW welding case will be developed in chapter 3, so little more needs to be said about it here.

The pulsed laser keyhole welding process differs from the CW case, in that the output power varies with time. Figure 2.4 shows some examples of laser pulses used in welding. The advantage of using this technique is that high peak power densities can be reached using lasers with a small average powers, and that as Nd:YAG lasers are used for pulsed welding, the power absorbed by the Fresnel mechanism is higher, and so it is easier to initiate and maintain the keyholing process. The mechanism involved in pulsed laser welding is illustrated schematically in figure 2.5 a-e). Once the laser beam strikes the workpiece, the energy is absorbed, causing the workpiece to melt and vaporise forming a keyhole, which may partially or fully penetrate the material. Once the laser is switched off, the keyhole collapses, and the energy in the molten pool is conducted away, leaving a weldbead. This sequence is repeated for the next pulse and the next, and so the pulsed laser welding process can be considered as a series of spot welds along the seam of the weld.

2.1.2 Pulsed Laser Heating Model

This section describes a 3-dimensional heat conduction model of the pulsed Nd:YAG laser welding process. Once the governing equations are stated, a numerical solution procedure is described

which solves the main equation subject to the appropriate boundary conditions.

2.1.2.1 Model Assumptions

The model depends on a number of major assumptions, these are described briefly below,

1) That the fluid flow in the molten region has no effect upon the heating process whatsoever. In reality, the surface tension gradient at the surface will drive the convection in the meltpool, forcing the hotter material outwards to the sides of the weldpool, and therefore increasing the molten width. The effects of this will be discussed later.

2) That the absorption varies with temperature up to the boiling point (see figure 2.6), after which, a keyhole is formed. It is assumed that in the keyhole the absorptivity can be described by the sum of the direct Fresnel absorption and the multiple reflections within the keyhole. The direct Fresnel absorption is a function of the angle of incidence, whereas the multiple reflection term is assumed to be constant. In reality, multiple reflections in the keyhole will ensure that the absorption at any point under the beam will be an extremely complex function of the keyhole geometry and position. However it is thought that a constant value, calibrated from experimental data, can describe the absorption in the keyhole adequately enough for the purpose of this model. Here, the plasma absorption in the keyhole is neglected, this can be justified by the fact that any plasma generated in the keyhole is almost invisible to 1.06μm radiation, and will therefore not act as a significant heat source.

3) In order to model a number of pulses, the keyhole is assumed to collapse immediately after the pulse ends. In reality the keyhole collapse takes a fraction of a millisecond (Ducharme et al,1993; Kroos et al (1992)). Additionally, the temperature everywhere in the collapsed keyhole is assumed to be just under the vaporisation point. In reality the violent swirling of the fluid in the weldpool will ensure that the temperature distribution is much more complex, however, it is thought that this will not affect comparative numerical trials a significant amount.

2.1.2.2 The Governing Equations

A cartesian coordinate system is employed, where the x direction is parallel to the direction of the beam, the y direction is perpendicular to the x axis, and runs parallel to the surface, and the z axis points into the workpiece. The heat is conducted throughout the material according to the energy equation, which is expressed in terms of the enthalpy function H, and the temperature T, as well as the material properties,

$$\rho U\frac{\partial H}{\partial x}+\rho\frac{\partial H}{\partial t}=\frac{\partial}{\partial x}k\frac{\partial T}{\partial x}+\frac{\partial}{\partial y}k\frac{\partial T}{\partial y}+\frac{\partial}{\partial z}k\frac{\partial T}{\partial z} \qquad 2.1$$

where

$$H=C_pT+L_vf_v+L_mf_m$$

$$\textit{where} \quad f_m=0 \quad T<T_s, \qquad f_m=1 \quad T>T_l, \qquad f_m=\frac{T-T_s}{T_l-T_s} \quad T_s<T<T_l \qquad 2.2$$

$$\textit{where} \quad f_v=0 \quad T<T_{vl}, \qquad f_v=1 \quad T>T_{v2}, \qquad f_v=\frac{T-T_{vl}}{T_{v2}-T_{vl}} \quad T_{vl}<T<T_{v2}$$

where f_m and f_v are defined as the molten and vapour fractions of the material respectively. In order to solve this equation, a new variable C_p' will be employed, in order to take into account the latent heat of melting and vaporisation. Assuming constant specific heat,

$$C_p'=C_p+\frac{L_mf_m+L_vf_v}{T} \qquad 2.3$$

Therefore, equation 2.1 can be expressed as,

$$\rho\frac{\partial C_p'T}{\partial t}+\rho U\frac{\partial C_p'T}{\partial x}=\frac{\partial}{\partial x}k\frac{\partial T}{\partial x}+\frac{\partial}{\partial y}k\frac{\partial T}{\partial y}+\frac{\partial}{\partial z}k\frac{\partial T}{\partial z} \qquad 2.4$$

This is a second order differential equation in T, which can be solved according to the correct boundary conditions. Under the beam, the heat input can be expressed in terms of the normal temperature gradient, the heated area, A_H, the area exposed to the beam, A_E, the absorbed power, P_d, the heat transfer coefficient between the metal and the air, h, and surface temperature rise (T_S-T_o);

$$-kA_H\frac{\partial T}{\partial n}=(P_d(x,y,t)A_E-hA_H(T_S-T_o)) \qquad 2.5$$

where the latent heat at the interface has been accounted for in equation 2.2. Here P_d is a function of time and surface position.. For the Gaussian and top hat profiles this can be described by

$$P_d=\frac{P_0}{\pi r_B^2}A(\phi,T)e^{-2\frac{r^2}{r_B^2}}, \qquad P_d=\frac{P_0}{\pi r_B^2}A(\phi,T) \qquad 2.6$$

respectively, where r_B is the beam radius, ϕ is the angle between the workpiece surface and the vertical plane, and P_O is the power at the centre of the beam. Here, $A(\phi,T)$ is the absorptivity, which is given as a function of temperature in figure 2.6 before keyholing commences. After the keyhole is initiated, the absorptivity is assumed to be the sum of the direct Fresnel absorption at the keyhole

walls (Schulz et al 1987), and the contribution due to multiple reflections in the keyhole, this can be expressed as,

$$A(\phi)=(1-\frac{1}{2}(\frac{1+(1-\epsilon\cos\phi)^2)}{1+(1+\epsilon\cos\phi)^2)}+\frac{\epsilon^2-2\epsilon\cos\phi+2\cos^2\phi}{\epsilon^2+2\epsilon\cos\phi+2\cos^2\phi}))+\frac{1}{A_E}\sum C_{mr}A_{mr} \qquad 2.7$$

where ε has a value of 0.24 for mild steel with Nd:YAG laser radiation. $\sum C_{mr}A_{mr}$ is the contribution due to the multiple reflections in the keyhole. Here C_{mr} is a constant that takes into account the fraction of the power left and the area exposed (similar to A_E for the direct beam), and A_{mr} is the multiple reflection absorption coefficient, which is dependent on the angle of incidence. In reality, this is a complex function of the shape of the keyhole, and would be extremely complicated to calculate in this simulation, the second term in equation 2.7 is therefore kept constant.

The heat is lost from the bottom and side surfaces by free convection, thus;

$$-k\frac{\partial T}{\partial n}=-h(T_S-T_o) \qquad 2.8$$

where the normal here is parallel to the z axis at the bottom, and the y axis is at the sides. Finally as x tends to infinity,

$$\frac{\partial T}{\partial x}\to 0, \qquad as \quad x\to\infty \qquad 2.9$$

The above system of equations represent a well posed problem which can be readily solved using the finite difference method, the following section describes the numerical methods involved in calculating such a solution.

2.1.2.3 The Solution Procedure

2.1.2.3.1 The Solution grid

The solution to the above set of partial differential equations can be found using finite difference methods using a cartesian grid mesh. In order to increase the accuracy of the solution, a grid is chosen which concentrated the elements in the region where the temperature gradients are very high.

2.1.2.3.2 Temperature Calculation at the Internal Grid Nodes

Figure 2.7 shows a control volume associated to a particular node, with its 6 nearest neighbours in the x,y and z directions. The term in x in the right hand side of the energy equation can be approximated by,

$$\frac{\partial}{\partial x}k\frac{\partial T}{\partial x}\approx\frac{1}{\Delta x}((k\frac{\partial T}{\partial x})_n-(k\frac{\partial T}{\partial x})_s)\approx\frac{1}{\Delta x}(k_n\frac{(T_N-T_P)}{\delta x_n}-k_s\frac{(T_P-T_S)}{\delta x_s}) \qquad 2.10$$

where the subscripts N,P and S denote the temperatures at these points respectively, and the conductivity subscripts n and s denote the fact that the conductivity has been evaluated at the control volume interface between the nodes N and P, and P and S respectively. On the right hand side, the velocity term can be approximated by,

$$\rho U\frac{\partial C_p'T}{\partial x}\approx\rho U\frac{(C_p'T)_P-(C_p'T)_S}{\delta x_s} \qquad 2.11$$

and the time derivative by,

$$\rho\frac{\partial}{\partial t}C_P'T\approx\rho C_P'\frac{(T_P^n-T_P^o)}{\delta t} \qquad 2.12$$

where the superscripts n and o denote the value of the temperature at time t and t+δt respectively. Using similar expressions as 2.10 for the partial derivatives with respect to y and z, and multiplying both sides by the control volume ΔxΔyΔz, we obtain, after some rearrangement, (Patankar, 1980),

$$a_PT_P=a_NT_N+a_ST_S+a_UT_U+a_DT_D+a_ET_E+a_WT_W+b \qquad 2.13$$

where the temperature coefficients are described by the following equations,

$$a_N=\frac{\Delta y\Delta z}{\delta x_n}k_n,\qquad a_S=\frac{\Delta y\Delta z}{\delta x_s}k_s+\frac{\rho UC_P\Delta x\Delta y\Delta z}{\delta x_s},\qquad a_U=\frac{\Delta x\Delta y}{\delta z_u}k_u,$$

$$a_D=\frac{\Delta x\Delta y}{\delta z_d}k_d\qquad a_E=\frac{\Delta x\Delta z}{\delta y_e}k_e,\qquad a_W=\frac{\Delta x\Delta z}{\delta y_w}k_w \qquad 2.14$$

$$a_P^o=\frac{\rho C_P'\Delta x\Delta y\Delta z}{\delta t},\qquad b=a_P^oT_P^o,\qquad a_P=a_N+a_S+a_U+a_D+a_W+a_E+a_P^o.$$

2.1.2.3.3 Calculation of the Moving Boundary

Once the temperature of an element is raised beyond the boiling point of the substrate, that element is assumed to have completely vaporised, and no longer takes part in any calculation. This in itself describes the position of the moving boundary. Figure 2.8 illustrates this point, here, the keyhole lies above the thick black line, while the main body is the liquid and solid metal (i.e the solution zone).

The keyhole shape is therefore a result of the solution procedure. From this we can calculate the normal to the keyhole, and hence evaluate the energy balance at the keyhole walls. The keyhole

wall can be described as a function of x and y, so z=s(x,y), hence the unit inward normal is,

$$\underline{\hat{n}}=\frac{1}{l}(-\frac{\partial s}{\partial x},-\frac{\partial s}{\partial y},1) \qquad 2.15$$

where l is the modulus of the normal vector,

$$l=\sqrt{(\frac{\partial s}{\partial x})^2+(\frac{\partial s}{\partial y})^2+1} \qquad 2.16$$

Therefore the temperature gradient with respect to the normal is,

$$\frac{\partial T}{\partial n}=\underline{\hat{n}}.\nabla T=n_x\frac{\partial T}{\partial x}+n_y\frac{\partial T}{\partial y}+n_z\frac{\partial T}{\partial z} \qquad 2.17$$

where

$$n_x=-\frac{1}{l}\frac{\partial s}{\partial x}, \qquad n_y=-\frac{1}{l}\frac{\partial s}{\partial y}, \qquad n_z=\frac{1}{l} \qquad 2.18$$

After each set of iterations at time t, a new boundary is calculated by evaluating the position of the vapour front, and then used for the next set of iterations at time t+δt.

2.1.2.3.4 Calculation of Temperature at The Keyhole Surface

Figure 2.9 shows the control volume of an element at the keyhole surface. Here, T_S is part of the keyhole and not part of the solution zone. Using the above analysis we can evaluate the differential with respect to x,

$$\frac{\partial}{\partial x}k\frac{\partial T}{\partial x}\approx\frac{1}{\Delta x}(k_N(\frac{\partial T}{\partial x})_N-k_S(\frac{\partial T}{\partial x})) \qquad 2.19$$

where the partial derivative at s can be evaluated in terms of the boundary condition,

$$-k\underline{\hat{n}}.\nabla T=(P_d\frac{A_E}{A_H}-h(T_S-T_o)) \qquad 2.20$$

therefore,

$$k\frac{\partial T}{\partial x}=-\frac{1}{n_x}(P_d\frac{A_E}{A_H}-h(T-T_o)+n_y\frac{\partial T}{\partial y}+n_z\frac{\partial T}{\partial z}) \qquad 2.21$$

where A_E/A_H is the ratio of the area exposed to the laser beam to the area heated. This can be shown to be the projection of the surface of the keyhole onto the x-y plane. Therefore, if θ is the angle between the surface and the x-y plane, and the ratio is cosθ, we have,

$$\frac{A_E}{A_H}=\underline{\hat{P}}.\underline{\hat{n}}=(0,0,1).(n_x,n_y,n_z)=n_z=\frac{1}{l} \qquad 2.22$$

It can also be shown that θ is also the angle between the direction of the beam and the unit normal (i.e it is therefore equal to ϕ in the absorptivity equation). The energy equation can now be written as,

$$k\frac{\partial T}{\partial x}=-\frac{1}{n_x}\left(\frac{P_d}{l}-h(T-T_o)+n_y\frac{\partial T}{\partial y}+n_z\frac{\partial T}{\partial z}\right) \qquad 2.23$$

Similarly, if T_D and T_W are inside the keyhole, then,

$$k\frac{\partial T}{\partial y}=-\frac{1}{n_y}\left(\frac{P_d}{l}-h(T-T_o)+n_x\frac{\partial T}{\partial x}+n_z\frac{\partial T}{\partial z}\right) \qquad 2.24$$

$$k\frac{\partial T}{\partial z}=-\frac{1}{n_z}\left(\frac{P_d}{l}-h(T-T_o)+n_x\frac{\partial T}{\partial x}+n_y\frac{\partial T}{\partial y}\right) \qquad 2.25$$

By substituting back into the main discretisation equation, a generic equation that is true for all nodes in the solution zone can easily be formulated,

$$a_P T_P=a_N T_N+a_S T_S+a_U T_U+a_D T_D+a_E T_E+a_W T_W+b \qquad 2.26$$

where,

$$a_N=\frac{\Delta y\Delta z}{\delta x_n}k_n(1-C_{xb}),\qquad a_S=\frac{\Delta y\Delta z}{\delta x_s}k_s(1-C_{xf})+\frac{\rho UC_P\Delta x\Delta y\Delta z}{\delta x_s},\qquad a_U=\frac{\Delta x\Delta y}{\delta z_u}k_u$$

$$a_D=\frac{\Delta x\Delta y}{\delta z_d}k_d(1-C_z)\qquad a_E=\frac{\Delta x\Delta z}{\delta y_e}k_e\qquad a_W=\frac{\Delta x\Delta z}{\delta y_w}k_w(1-C_y) \qquad 2.27$$

$$a_P^o=\frac{\rho C_P\Delta x\Delta y\Delta z}{\delta t},\qquad a_P=a_N+a_S+a_U+a_D+a_W+a_E+a_P^o. \qquad 2.28$$

and b can be calculated by

$$b=a_P^O T_P^O+\frac{2\Delta y\Delta z}{n_x}\left(\frac{1}{n_x}\left(\frac{P_d}{l}-h(T-T_O)+n_y\frac{\partial T}{\partial y}+n_z\frac{\partial T}{\partial z}\right)\max(C_{fx},C_{fb})\right.$$

$$+\frac{2\Delta x\Delta z}{n_y}\left(\frac{1}{n_y}\left(\frac{P_d}{l}-h(T-T_O)+n_x\frac{\partial T}{\partial x}+n_z\frac{\partial T}{\partial z}\right)C_y\right. \qquad 2.29$$

$$+\frac{2\Delta x\Delta y}{n_z}\left(\frac{1}{n_z}\left(\frac{P_d}{l}-h(T-T_O)+n_x\frac{\partial T}{\partial x}+n_y\frac{\partial T}{\partial y}\right)C_z\right.$$

Here, the coefficients act as markers for the position of the boundary, so

$$C_{xf}=0 \quad if \quad T_S<T_v, \quad C_{xf}=1 \quad if \quad T_S\geq T_v, \quad C_{xb}=0 \quad if \quad T_N<T_v, \quad C_{xb}=1 \quad if \quad T_N\geq T_v \qquad 2.30$$
$$C_y=0 \quad if \quad T_W<T_v, \quad C_y=1 \quad if \quad T_W\geq T_v, \quad C_z=0 \quad if \quad T_D<T_v, \quad C_z=1 \quad if \quad T_D\geq T_v$$

2.1.2.3.5 The Numerical Solution Procedure

The solution procedure is illustrated by means of a flow chart in figure 2.10. After the variables and arrays have been initialised, with all the initial temperatures in the solution zone set to ambient. The temperature field at time t+δt is calculated from the initial temperature field, and from the power input, using a semi implicit line-by-line method described in Patankar (1980). Any nodes found to be above the boiling point are taken out of the solution zone, and the moving surface, together with the unit normal, is recalculated. This procedure is repeated again and again until the maximum time has elapsed.

2.1.3 Results and Discussion

The results from the model are presented in two parts, the first part consists of a comparison between experimental data and theoretical results, together with an analysis of the main features of the workpiece response to the pulsed laser beam. The second sections are qualitative numerical studies into the effects of varying the process parameters and the thermophysical properties on the weldpool and keyhole dimensions.

2.1.3.1 Comparison to experimental data

In order to validate the model, results from the model were compared with experimental data from Erasenthiran (1997). Figure 2.11 shows the experimental arrangement for the pulse laser welding procedure. Bead on plate welds were made on 2mm thick mild steel using a focused and pulsed Nd:YAG laser, with argon shroud gas. The experimentally determined spotsize was found to be 0.8mm ± 0.05mm, table 2.1 shows the remaining parameters.

Figure 2.12 shows the comparison between experimental and theoretical results (here the dimensions have been calculated after 5 pulses in the model). As can be seen, although there seems

to be comparatively good agreement concerning the penetration depth, the difference in melt width can be as much as 30% (it is significant however that the model predicts the trends well enough). There are number of reasons why this is so,

1) Figure 2.13 illustrates the flow in the molten pool during pulsed laser welding. In this case, there are two main driving mechanisms to consider, Marangoni convection, and the influence of the recoil pressure due to ablation.

a) Marangoni flow - This describes the flow at the surface where the molten material is drawn along the surface from a region of low surface tension to a region of higher surface tension. For significant values of surface tension gradient (as in steel), Marangoni flow can become the main driving force for flow within the weldpool. The surface tension gradient is negative for steel, causing the fluid to flow from a high temperature to a lower temperature. In the present case, therefore, the fluid will flow from inside of the beam outwards, thus heating the cooler parts of the workpiece, resulting in an increase in the weld width. Tsai and Kou (1989), developed a two dimensional, non ablating, free surface fluid flow model for weldpool convection, and observed that the conduction limited solution tended to underestimate the weld width by up to 30%. It is thought that in this case this makes a significant contribution to the increase in weld width.

b) Influence of recoil pressure - As the material ablates, a recoil pressure is experienced by the molten material, proportional to the evaporation rate. Assuming that most of the ablation occurs underneath the beam, the fluid will tend to depress in this region, again forcing the hot molten material upwards towards the cooler parts.

2) In this model it was assumed that there could be no heat input outside the beam radius, however, multiple reflections in the keyhole would ensure that some of the power was reflected into the open mouth of the keyhole, again increasing the weld width.

The combination of these mechanisms will ensure that there is a significant underestimate in the value of weld width. It is interesting to note that there is a similar, significant discrepancy between theoretically calculated weld widths and experimental observations in the CW case (see chapter 4). It can be shown that for a CW beam, the evaporation rates are not as high, but a further contribution which tends to increase the weld width is present, i.e. the presence of an additional heat source due to the absorption in the plasma plume above the workpiece. This would suggest that the contribution due to the recoil pressure is quite significant in the pulsed laser welding case.

2.1.3.1.1 Characteristics of pulsed laser welding

Figures 2.14 and 2.15 illustrate the response of 2mm thick mild steel plate to a number of laser pulses, using a power of 3.5kW, a beam radius of 0.4mm, a time on of 0.012s, and a total pulse time of 0.1s. It can be seen quite clearly that the weldpool dimensions increase with the number of pulses (it must be stated here that the model would exaggerate this a little due to the fact that it assumes that the collapsed keyhole has a constant temperature of around 3000K, in reality the temperature field would not be constant, and the average temperature would be less). This is due to the fact that during the previous pulse, the laser heats up the area in front of the beam. This area will still be comparatively hot at the start of the next pulse, and so will be easier to weld. Of course, the distance and time between pulses is crucial here, it can be seen from figure 2.14 that with a larger velocity (and hence a larger distance between pulses), the maximum value of weld dimensions is obtained after just two pulses. After this, the previous pulse will have little effect on the value of the maximum weldpool dimensions.

2.1.3.2 Variation of Power density

The beam radius and peak power were varied in order to determine the effect of varying the peak power density on the pulsed laser welding process. The results of these comparative numerical trials are shown in figures 2.16 and 2.17. Obviously, an increase in power leads to a increased heating rate, until full penetration is reached in the 3kW case. It can be seen in figure 2.17 that a decrease in beam radius from 0.4mm to 0.2mm, results in an increase in keyhole depth of 100%, and a corresponding decrease in weld width of 25%. It is interesting to note that the thinner weld bead created with 0.2mm radius, cools at a much higher rate than the shorter, fatter weld bead using a 4mm radius beam.

2.1.3.3 Variation of Pulse Duration

It is very important for a laser operator to optimise the pulse shape and length in order to use the laser to the its full capacity. This section and the next are concerned with this optimisation process. In order to determine the effect of pulse duration on the laser welding process, 3 rectangular pulses, each with the same pulse energy, were used (a) P=4000W, t_{on}=4ms b) P=2000W t_{on}=8ms, and c) P=1333W, t_{on}=12ms), the melt and keyhole dimensions are presented in figure 2.18. It can be seen here that the greatest melt depth can be obtained width the shortest pulse duration and the highest peak power. A word of caution needs to be mentioned here, however. If the pulse on time is too short, and the corresponding peak power is too high, the evaporation rate will become so high that the recoil pressure in the keyhole will become large enough to expel almost all of the molten material. This describes the onset of drilling, and must be avoided. It is also worth noting that the keyhole width reaches similar maximum values for all cases early on in the pulse, and that the behaviour of the weld widths is similar, showing that the weld width is not strongly dependent on

power once the keyhole has been initiated, but depends mainly upon the keyhole width, which in turn is mainly dependent upon the keyhole radius.

2.1.3.4 Variation of pulse shape

In order to determine the effect of varying the pulse shape on the process, four commonly used welding pulse shapes were used for comparison, these were defined as (see figure 2.4),

1) Rectangular

$$P=2000W, \quad t<t_{on}, \qquad P=0W, \quad t_{on}<t<t_p$$

2) Ramp up

$$P=1500W, \quad t<\frac{t_{on}}{2}, \qquad P=2500W, \quad \frac{t_{on}}{2}<t<t_{on}, \qquad P=0W, \quad t_{on}<t<t_p$$

3) Ramp down

$$P=2500W, \quad t<\frac{t_{on}}{2}, \qquad P=1500W, \quad \frac{t_{on}}{2}<t<t_{on}, \qquad P=0W, \quad t_{on}<t<t_p$$

4) Superpulse

$$P=P_2+2(P_1-P_2)\frac{t}{t_1}, \quad 0<t<t_1$$

$$P=\frac{2(P_2-P_1)}{t_1}\left(t-\frac{t_1}{2}\left(1-\frac{P_1}{P_2-P_1}\right)\right), \quad t_1<t<t_{on}, \qquad P=0W, \quad t_{on}<t<t_p \qquad \textbf{2.30 a-d)}$$

where P_1 and P_2 are 3000W and 1833W respectively, and $t_2=t_{on}/4$, and $t_1=t_2/2$. In these definitions, t_{on} is taken to be the time the laser is on, and t_p is the overall time for one pulse, this is set so that the average power over all time is 400W.

Figure 2.19 shows the effects of varying the pulse shape on the keyhole depth, the weldpool width and weldpool depth. As can be seen, there is no significant change in weld width, whereas for the keyhole and weld depths, there is only 10% in the difference between the shallowest (ramp up pulse) and the deepest (ramp down pulse) depth. It can be concluded therefore, that for a particular power density and pulse duration suitable for welding purposes, the weldpool dimensions are not a strong function of the pulse shape.

2.1.3.5 Variation of Thermophysical Properties

The specific heat is defined as the amount of energy required to raise 1kg of a material by 1K. Therefore, an increase in specific heat, while keeping all other properties constant, will result in

a lower temperature field and smaller weldpool dimensions. This point is illustrated well in figure 2.20. It is also worth noting that during cooling, more energy is required to be removed from the region in order to lower the temperature for the high specific heat case, and so the weldpool will take a longer time to disappear.

Increasing the conductivity of the material will allow the heat input to be conducted away from the surface much quicker, this effect is known as thermal runaway. This makes the initiation and maintenance of the keyhole more difficult and will therefore result in smaller weldpool dimensions. Once the laser pulse has finished, the highest thermal conductivity case will dissipate the heat quickest throughout the material, and will therefore have the largest cooling rate.

2.2 Pulsed Laser Material Removal

In this section after a brief introduction, a model which describes the laser edge plate material removal process will be presented. The aim of this model is to describe in a quantitative and qualitative way, the effect that the variation of a number of process parameters have upon the process itself.

2.2.1 Laser Material Removal Background

Material removal is used in many manufacturing processes in order to shape parts. Currently, there are many conventional methods for undertaking this task, such as drilling, sawing or milling, however, they all suffer from a number of disadvantages,

- The surface finish can vary significantly. In order to increase the surface quality, the feed rate has to be decreased, increasing the time taken and hence cost per part.

- Hard materials can only be machined using special tools, again increasing cost.

- Conventional tools suffer from wear and have to be replaced on a regular basis.

The laser can offer significant advantages in this area, as laser material removal is a relatively fast process, hard materials can be machined just as easily as other materials, and as laser machining is a non-contact process, tool wear is not a problem.

Currently, there are a number of laser material removal processes used in manufacturing. However, this study will concentrate on the new process of pulsed Nd:YAG laser edge plate machining for Laminated Object Manufacturing (LOM) (Erasenthiran et al, 1997). LOM is a technique used in rapid prototyping and rapid manufacture, whereby laminates from thin, shaped,

metal sheets are welded on top of each other in order to produce the required object. One problem that arises in this process however, is the 'staircase effect' illustrated in figure 2.22. In the initial stages of LOM, this problem was solved by grinding and polishing. However, in order to improve the process a number of methods have been investigated, including 'filling in' the steps with powder (Nakagawa, Kunienda and Liu (1995)), profile cutting (Sepold 1994) and EDM machining (Soar and Dickens (1996)). However, none of these have proved successful to date. This current modelling project concerns the study of an alternative solution (Erasenthiran et al, 1997), whereby material is removed from the edge of the laminate by pulsed laser machining in order to shape it.

2.2.1.1 Modelling The Pulsed Laser Edge Machining Process

A schematic diagram illustrating the mechanisms involved in the pulsed laser edge plate machining process is shown in figure 2.23. The pulsed laser beam, at an angle α to the vertical, is scanned across the material. As each pulse commences, the material is heated and undergoes melting, and in some cases evaporation. The molten or gaseous material is ejected from the cut zone by a high velocity side jet of nitrogen, which also protects the material from burning.

The aim of the model is to describe in both a quantitative and qualitative way, the effect of the variation of a number of process parameters upon the process. Of particular interest are the pulse parameters, such as energy and repetition rate, as well as the process velocity and the type of laser.

In order to model the edge plate machining process, the following assumptions were made

1) Once the cut front is initiated, the cut slope can be defined by an isotherm $T=T_S$, where T_S is the liquidus temperature of the substrate. This obviously assumes that all of the molten material is blown out of the kerf. This is adequate for most of the range of parameters studied here, where the power densities are comparatively low. However, at higher power densities, the rate of melt generation will become too high for all the molten material to be blown away by the gas jet. In this case the model will tend to overestimate the value of the removed depth. This resolidified molten material causes an increase in surface roughness, which will add to post process machining time and cost. For this reason high power densities are undesirable.

2) Radiation losses are assumed negligible. This can be justified by assuming that the maximum surface area for a heated groove at the melting point is approximately $(\pi r_B^2+2t\pi r_B)/2$ where t is the thickness of the workpiece. In such a case the heat lost through radiation can be calculated by,

$$Q_{r\ loss} = \sigma \epsilon 4 r_B t (T_m^4 - T_o^4) \qquad 2.31$$

Using a value of 1 the emissivity, and 0.2mm and 1mm as typical values of the beam radius and the plate thickness respectively, $Q_{r\ loss}$ has a value of 0.58W. This is negligible compared with the typical values of powers used in the material removal process, which are of the order of 10^2-10^3W.

3) There are no multiple reflections in the kerf, and the absorption can be estimated from the Fresnel formula. This is certainly true at the sides of the wall, although in some cases, the bottom of the kerf will experience enhanced absorption due to multiple reflections. The neglection of multiple reflections will therefore tend to underestimate the value of the machined depth.

In the next section, the governing and discretised equations for this process will be presented, together with the numerical procedure, which is similar to the one described above for the pulsed laser welding process. In order to obtain a 'steady' solution, a number of pulses are run. This also enables the program to take into account the pulse repetition rate and the velocity.

2.2.2 The Solution Procedure

The governing equations are similar to the ones in the previous sections on pulsed laser welding. The energy is passed throughout the body by heat conduction,

$$\rho U \frac{\partial H}{\partial x} + \rho \frac{\partial H}{\partial t} = \frac{\partial}{\partial x} k \frac{\partial T}{\partial x} + \frac{\partial}{\partial y} k \frac{\partial T}{\partial y} + \frac{\partial}{\partial z} k \frac{\partial T}{\partial z} \qquad 2.32$$

At the surface, the energy balance can be described by,

$$-kA_H \frac{\partial T}{\partial n} = \left(\frac{P_A(x,y,t)}{\pi r_b^2} A_E - h_{FO} A_H (T_S - T_o) - \rho v_n \Delta L \right) \qquad 2.33$$

where h_{FO} is the forced convection heat transfer coefficient, which is set to 100 in this case, v_n is the normal velocity of the melt front, and ΔL is the latent heat of melting. In this model the temperature at the moving surface is assumed to be equal just above the melting point. The latent heat of melting is again taken into account using the lumped heat capacity formulation used in the previous model, thus assuming that melting occurs over a specified temperature range, and not just at the melting point. The latent heat term in equation 2.33 is therefore dropped.

The heat is convected away from the sides and the bottom surface by free convection,

$$k\frac{\partial T}{\partial n}=h_{FR}(T_S-T_0) \qquad 2.34$$

where h_{FR} is the free convection heat transfer coefficient between the metal and the air, typical values range between 10-15 (Croft and Stone, 1977), and so, a value of 15 will be used here.

2.2.2.1 Absorbed Power Density

The equations for the power density need to be modified for the edge machining case in order to take into account the angle of the beam to the workpiece. The area of the beam is thus transformed from πr_B^2 to $\pi r_B^2/\cos\alpha$, where α is the angle the beam makes with the vertical. Therefore the functions describing the gaussian and top hat profiles are,

$$P_d=\frac{PA(\phi,T)}{\pi r_B^2}\cos(\alpha)e^{\frac{x^2+y^2\cos^2(\alpha)}{r_B^2}}, \qquad P_d=\frac{PA(\phi,T)}{\pi r_B^2}\cos(\alpha) \qquad 2.35$$

Here, ϕ is the angle between the beam axis and the unit normal, and A, the absorption, can be calculated using the direct Fresnel formula.

$$\phi=\hat{F}.\hat{n}=(0,\sin(\alpha),\cos(\alpha)).(n_x,n_y,n_z)=n_y\sin(\alpha)+n_z\cos(\alpha) \qquad 2.36$$

It can also be shown that this is also the ratio of the exposed and heated areas, and so the energy balance at the surface now reads,

$$-k\hat{n}.\nabla T=\frac{PA(\phi,T)}{\pi r_B^2}(n_y\sin(\alpha)+n_z\cos(\alpha))\cos\alpha-h(T_S-T_0) \qquad 2.37$$

for the simplest, top hat case.

2.2.3 Numerical Solution Procedure

In order to evaluate the above equations, a similar procedure to the one described for the pulsed laser welding model was used. The temperature field was calculated, the moving boundary changed if necessary, and the surface recalculated at a particular time t. The generic formulation for the discretised equation is,

$$a_PT_P=a_NT_N+a_ST_S+a_UT_U+a_DT_D+a_ET_E+a_WT_W+b \qquad 2.38$$

where the temperature coefficients are,

$$a_N=\frac{\Delta y\Delta z}{\delta x_n}k_n, \qquad a_S=\frac{\Delta y\Delta z}{\delta x_s}k_s(1-C_x), \qquad a_U=\frac{\Delta x\Delta y}{\delta z_u}k_u$$
$$a_D=\frac{\Delta x\Delta y}{\delta z_d}k_d(1-C_z) \qquad a_E=\frac{\Delta x\Delta z}{\delta y_e}k_e, \qquad a_W=\frac{\Delta x\Delta z}{\delta y_w}k_w(1-C_y) \qquad 2.39$$

for the neighbouring nodes, and

$$a_P^O=\frac{\rho C_P\delta x\delta y\delta z}{\delta t}, \qquad a_P=a_N+a_S+a_U+a_D+a_W+a_E+a_P^O. \qquad 2.40$$

$$b=a_P^O T_P^O+\frac{2\Delta y\Delta z}{n_x}\left(\frac{1}{n_x}(P_d(n_y\sin(\alpha)+n_z\cos(\alpha))-h(T-T_O)+n_y\frac{\partial T}{\partial y}+n_z\frac{\partial T}{\partial z})C_c\right.$$
$$+\frac{2\Delta x\Delta z}{n_y}\left(\frac{1}{n_y}(P_d(n_y\sin(\alpha)+n_z\cos(\alpha))-h(T-T_O)+n_x\frac{\partial T}{\partial x}+n_z\frac{\partial T}{\partial z})\max(C_{yp},C_{ym})\right. \qquad 2.41$$
$$+\frac{2\Delta x\Delta y}{n_z}\left(\frac{1}{n_z}(P_d(n_y\sin(\alpha)+n_z\cos(\alpha))-h(T-T_O)+n_x\frac{\partial T}{\partial x}+n_y\frac{\partial T}{\partial y})C_z\right.$$
$$-2\Delta y\Delta z h_{\max}(T_S-T_0)-2\Delta x\Delta z h_{\max}(T_S-T_0))y_{0f}-2\Delta x\Delta y h_{\max}(T_S-T_0)z_{mf}$$

where

$$C_x=0 \;\; if \;\; T_S<T_m, \quad C_x=1 \;\; if \;\; T_S\geq T_m, \quad C_y=0 \;\; if \;\; T_W<T_m, \quad C_y=1 \;\; if \;\; T_W\geq T_m \qquad 2.42$$
$$C_z=0 \;\; if \;\; T_D<T_m, \quad C_z=1 \;\; if \;\; T_D\geq T_m$$

and

$$y_{0f}=1 \;\; if \;\; y=y_{\min}, \quad y_{0f}=0 \;\; otherwise, \quad x_{0f}=1 \;\; if \;\; x=x_{\min}, \quad x_{0f}=0 \;\; otherwise \qquad 2.43$$
$$z_{mf}1 \;\; if \;\; z=z_{\min}, \quad z_{mf}=0 \;\; otherwise$$

2.2.3 Results and Discussion

2.2.3.1 Comparison with experimental data

In order to validate the model, results from the model were compared with experimental data from Erasenthiran(1997). The experimental arrangement is shown in figure 2.24. Here, a slightly defocussed pulsed beam, overlapping the material by 50% percent, is scanned across a workpiece in order to melt or ablate the material. A nitrogen gas jet at 7 bar is delivered from the side in order to blow away the molten material. Figure 2.25 shows the comparison between experimental results and theoretical predictions. As can be seen there seems to be reasonably good agreement between theory and experiment for the lower powers. However, as the power increases, the rate of melt generation became so high that the gas jet is unable to remove all the material, thus increasing the surface temperature, which invalidates one of the primary assumptions of the model. As can be seen, in this

region the profile depth can be overestimated by as much as 40%.

2.2.3.2 Variation of beam parameters

Figure 2.26 shows the effect of the variation of power upon the profile of the surface, using the process parameters given in table 2.5. Obviously, the higher the heat input, the larger the machined depth. An increase in the power from 1 to 2kW, increases the depth by over 80%. Decreasing the beam radius by 50% will quadruple the power density, however at slow speeds, pulses with a larger beam will overlap the previous pulse by a higher percentage, and so a point on the surface would be heated for a longer time. Figure 2.27 shows the machined profile surface after 3 pulses for beam radii of 0.1, 0.2 and 0.3mm. As can be seen, the smaller beam radius produces a deeper and much steeper profile than the large beams, which produce shallower and wider profiles. Figure 2.28 shows that there is little difference between the mode structure and the machined profile (less than 20%), the beam radius is slightly wider in the gaussian mode case

A change in wavelength will change the value of ε in the absorptivity equation, and hence the absorptivity. Figure 2.29 shows the variation in absorptivity with angle of incidence for different values of wavelength; here the values of ε used for CO_2, CO and Nd:YAG wavelengths were 0.08. 0.12 and 0.24 respectively. For angles of less than 80^0, the absorptivity for Nd:YAG laser wavelength is greater than that for the CO laser, which is greater than that for the CO_2 laser. This fact is illustrated in figure 2.30, where machining with an Nd:YAG laser will result in a increase in depth of 65% or 125% when compared with CO and CO_2 lasers respectively.

2.2.3.3 Variation of process velocity

An increase in the process velocity or time between pulses will increase the distance between the present pulse and the next. This will have the effect of decreasing the profile depth due to the fact that the material interaction time is less. This is illustrated in figure 2.31. Decreasing the speed will however decrease the material removal rate, hence increasing production costs.

2.3 Summary

The first section of chapter 2 was concerned with the development of a 3 dimensional heat conduction and ablation model for the pulsed laser welding process. The model took into account the free boundaries of the melting and ablating interfaces, as well as the heat conduction throughout the workpiece. The model neglected the fluid flow in the weldpool and used a simplified formula for the absorption due to multiple reflections.

- The results from the model were compared with experimental data, and it was found that although there was good agreement with respect to the weld depth (10-15%), there was up to 30%

difference in weld widths. The main reasons for this discrepancy were thought to be a combination of the effects of Marangoni convection, and the upward flow due to the excess ablation pressure in the keyhole. Both of these mechanisms tend to push the hot molten material from under the beam, outwards towards the sides of the weldpool, with the result of increasing the weld width.

- The weld dimensions increase with the number of pulses asymptotically. This effect is more pronounced when the velocity and hence the distance between pulses on the plate decrease, as in these cases the laser is welding a hotter region.

- As long as the keyhole is formed, the top weldbead width depends mainly upon the time taken for each pulse, the keyhole radius (which is approximately equal to the beam radius at the top of the weld), and is only weakly dependent upon the pulse shape. The size of the weldpool width is dominated by the laser on time, as well as the beam radius.

- the keyhole and the weldpool depths are strongly dependent on the power density, the pulse duration and again only weakly dependent on pulse shape. An increase in power density (i.e an increase in power or a decrease in beam radius), or the time taken for a particular pulse (using the same power), will result in an increase in keyhole and weldpool depths.

- an increase in specific heat means that the energy required to raise the temperature of the substrate by 1K will increase. This will lead to a decrease in the temperature field, and the material will be more difficult to weld. The high specific heat will also result in an increase in the amount of energy required to be taken away from the material to cool the substrate, and hence the gradient of the cooling curve will be less. Increasing the thermal conductivity will allow the heat input to dissipate throughout the material at a greater rate, the material will therefore be more difficult to weld, and will cool much faster

In the second section of chapter 2, a 3 dimensional heat conduction model was developed in order to simulate the pulsed laser edge plate machining process. The model assumed that any molten material generated by the laser was blown away by the gas jet. The model was compared to experimental data. There is reasonably good agreement at lower powers. At higher powers, however, where the melt is generated at a much quicker rate and the gas jet is unable to remove all the molten material (hence invalidating the main assumption), the model can overestimate the machined depth by up to 40%.

The behaviour of the pulsed laser machining process is very similar to the pulsed welding case. An increase in power density or time taken for a pulse (using a constant power), or a decrease

in velocity will increase the profile depth. An increase in wavelength in the infra red region will correspond to a decrease in absorptivity at the profile surface, and hence lead to an decrease in profile depth.

Chapter 2 Tables

Parameter or property	Value
Average power (laser on)	A1, 3333W A2, 2500W
Average power	400W
Pulse shape	Superpulse
Beam radius	0.4mm
Pulse on time	A1, 12ms A2, 16ms
Velocity	8-32mm/s
Keyhole absorptivity	A1, $A=A(\phi)+0.05$ A2, $A=A(\phi)+0.075$
Thermophysical properties	Conductivity=36W/mK Specific heat=600J/KgK Density=7000Kg/m^3
Wavelength	1.06μm

Table 2.1 Values of process parameters used in experiments A1 and A2, and of the thermophysical properties in the process simulation.

Parameter or property	Value
Power	2000W
Velocity	10mm/s
Average power	400W
Pulse shape	Rectangular
Beam radius	0.2mm
Pulse on time	12ms
Pulse off time	48ms
Keyhole absorptivity	$A=A(\phi)+0.1$
Thermophysical properties	As above
Wavelength	1.06μm

Table 2.2 Typical values for parameters used in numerical calculations for pulsed laser welding

Section	Parameter or property	Values
2.1.4.2 a)	Peak power	1,2,3kW
b)	Beam radius	0.2,0.3,0.4mm
2.1.4.3	Pulse duration	4,8,12ms
	1/frequency	40ms
2.1.4.4	Pulse shape	Rectangular, superpulse, ramp up, ramp down
2.1.4.5a)	Specific heat	300, 600, 1200, 2400 J/kgK
2.1.4.5b)	Thermal conductivity	18, 36, 72, 144 W/mK

Table 2.3 Summary of process parameters or thermophysical properties altered for the pulsed laser welding numerical calculations in section 2.1.4.2-2.1.4.5

Parameter or property	**Value**
Average power (pulse on)	1877.5W-1167.5W
Velocity	6mm/s
Pulse shape	0.8ms
Beam radius	0.2mm
Pulse on time	0.8ms
Frequency	100Hz
Laser beam wavelength	1.06μm
Thermophysical properties	As above

Table 2.4 Values of process parameters used in the pulsed edge machining experiments, and of the thermophysical properties in the process simulation.

Parameter or property	Value
Power	1500W
Velocity	15mm/s
Pulse shape	Rectangular
Beam radius	0.25mm
Pulse on time	0.5ms
Frequency	100Hz
Absorptivity	$A=A(\phi,T)$
Thermophysical properties	As above
Wavelength	1.06μm

Table 2.5 Typical values for parameters used in numerical calculations for pulsed edge machining

Section	Parameter or property	Values
2.2.3.2 a)	Peak power	1,1.5,2kW
b)	Beam radius	0.1,0.2,0.3mm
c)	Beam mode	Gaussian, top hat
2.2.3.4	Wavelength	λ=1.06μm, 5.4μm, 10.6μm
2.2.3.3	Process velocity	5,10,15mm/s

Table 2.6 Summary of process parameters for the pulsed edge machining numerical calculations in section 2.2.3.2-2.2.3.6

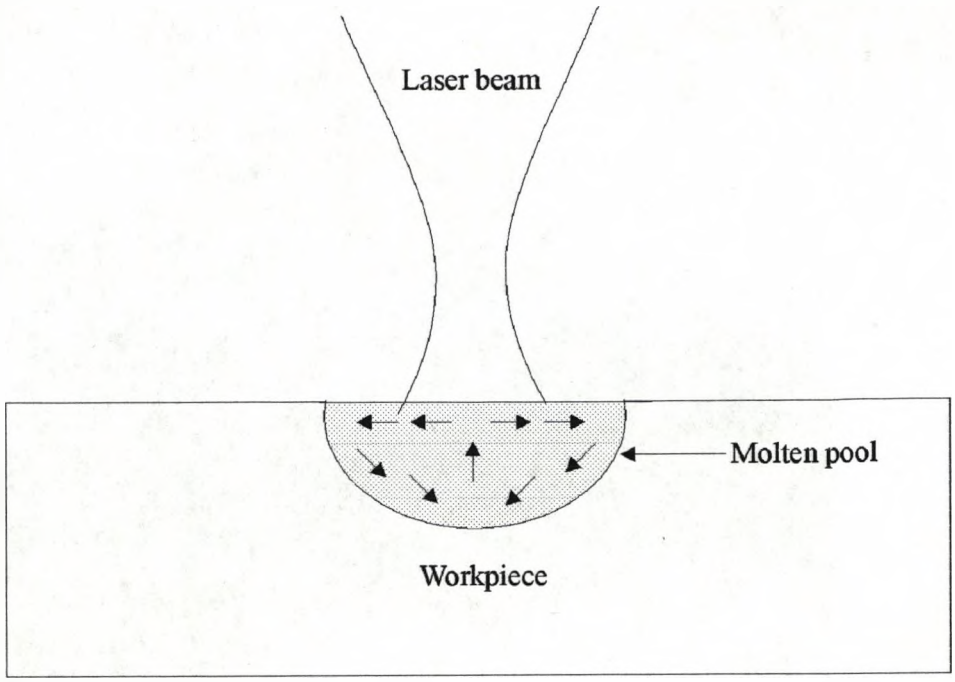


Fig.2.1 Schematic diagram of laser melting/conduction limited welding

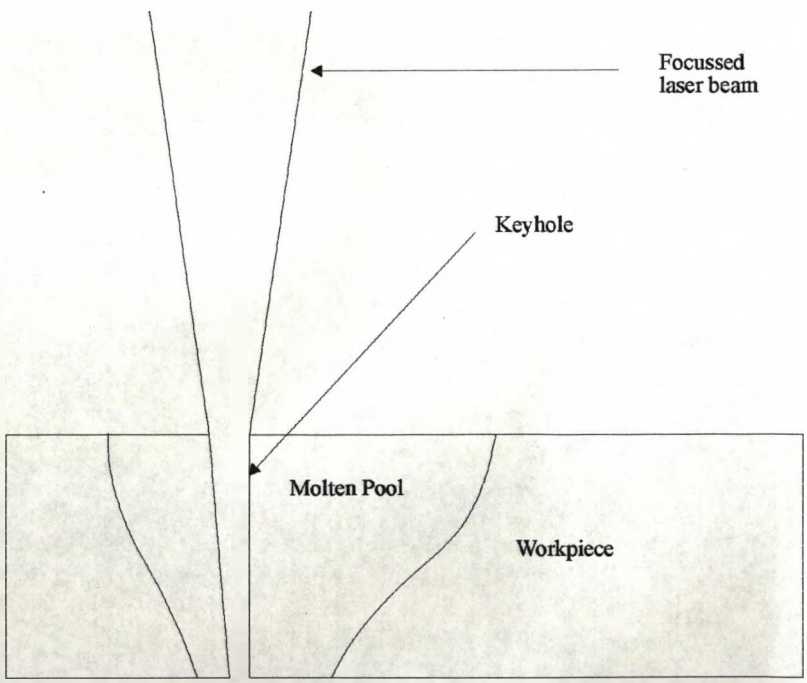


Fig.2.2 Schematic diagram of laser keyhole welding.

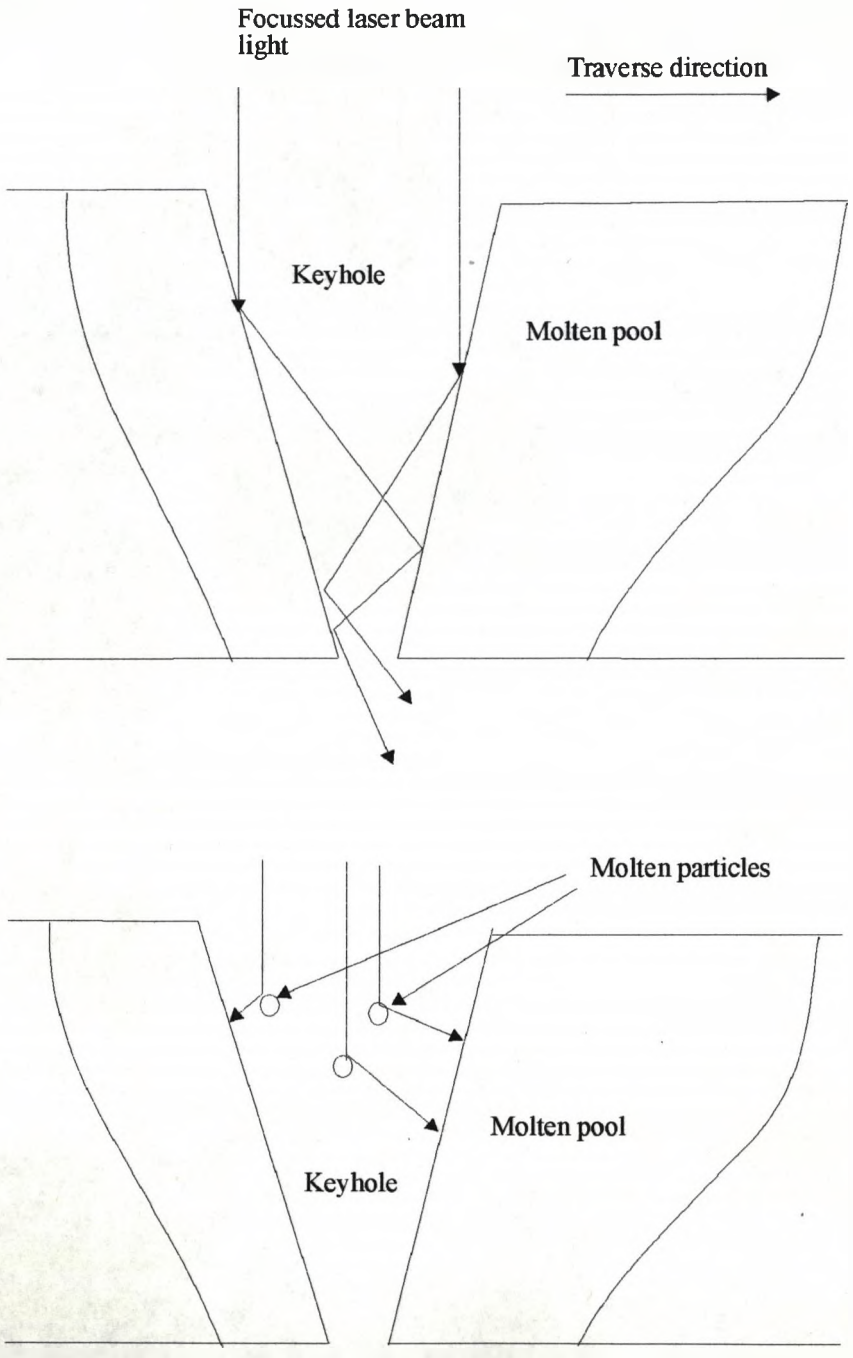


Fig.2.3 Schematic diagram of illustrating the absorption processes in laser keyhole welding, a) shows the direct and multiple reflections in the keyhole, b) illustrates the laser beam reflecting off particles in the keyhole onto the keyhole. In addition to these two mechanisms, there is absorption within the plasma in the keyhole.

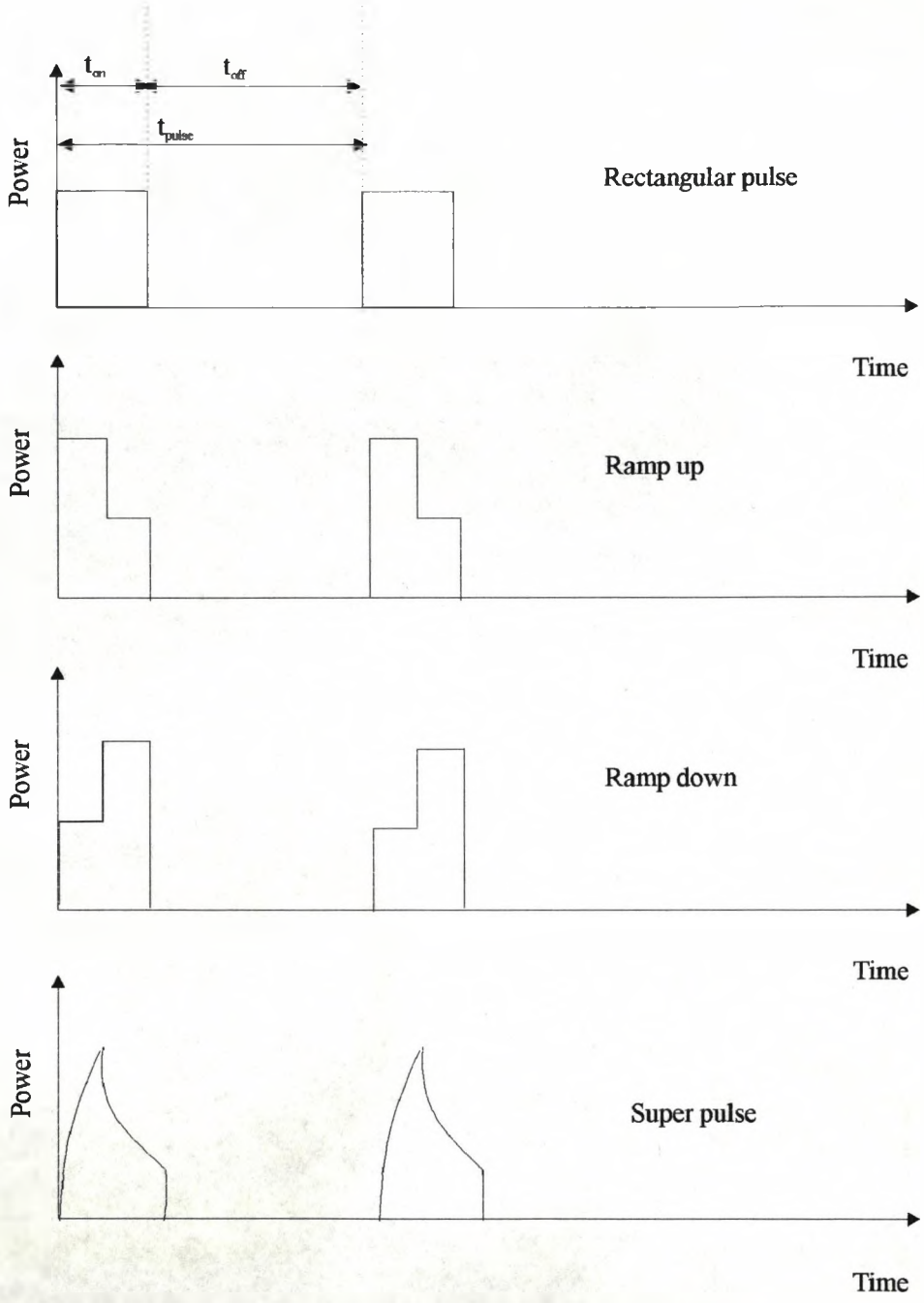


Fig.2.4 Four common pulse shapes used in pulse welding applications

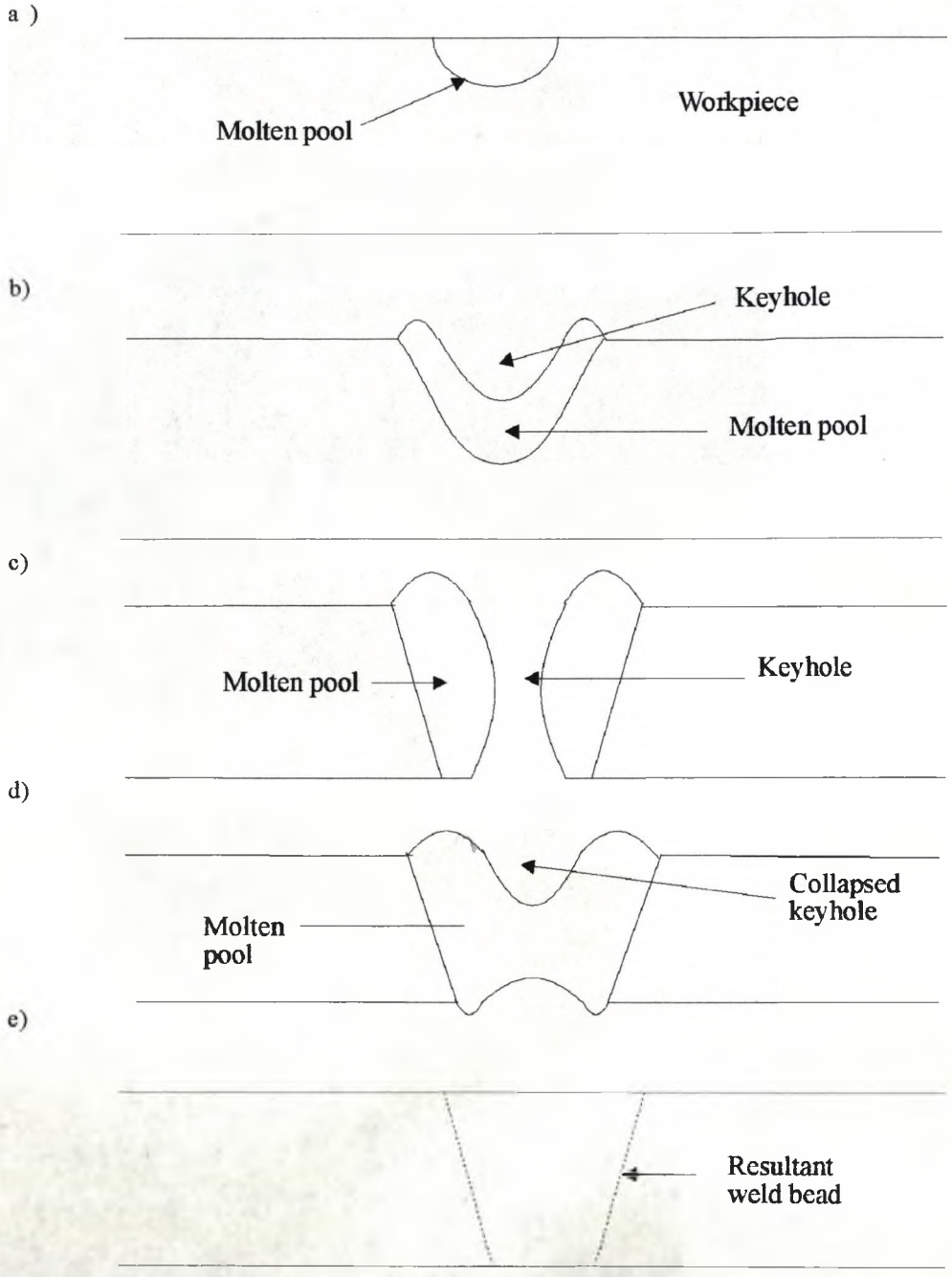


Fig.2.5 Schematic diagram showing the stages of pulsed laser keyhole welding; a) Initially the beam heats the workpiece, and causes melting, b) the worpiece absorbs enough energy to form a partially penetrating keyhole, c) in full penetration welding, the pulse is long enough to form a keyhole throughout the material, d) Once the pulse has ended, the keyhole collapses, and cooling takes place, leaving a weld bead along the seam.

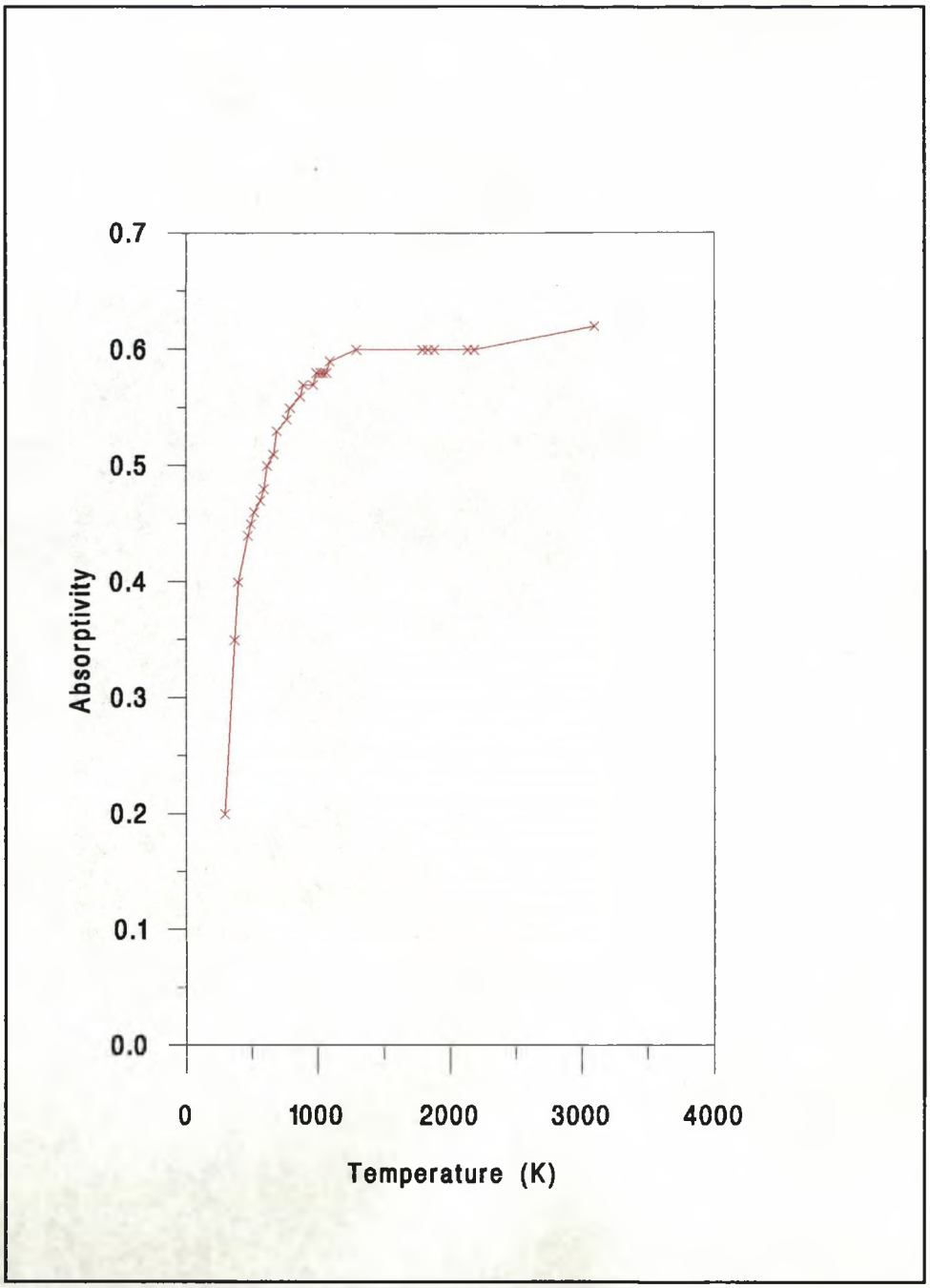


Fig.2.6 Temperature dependent properties of steel.

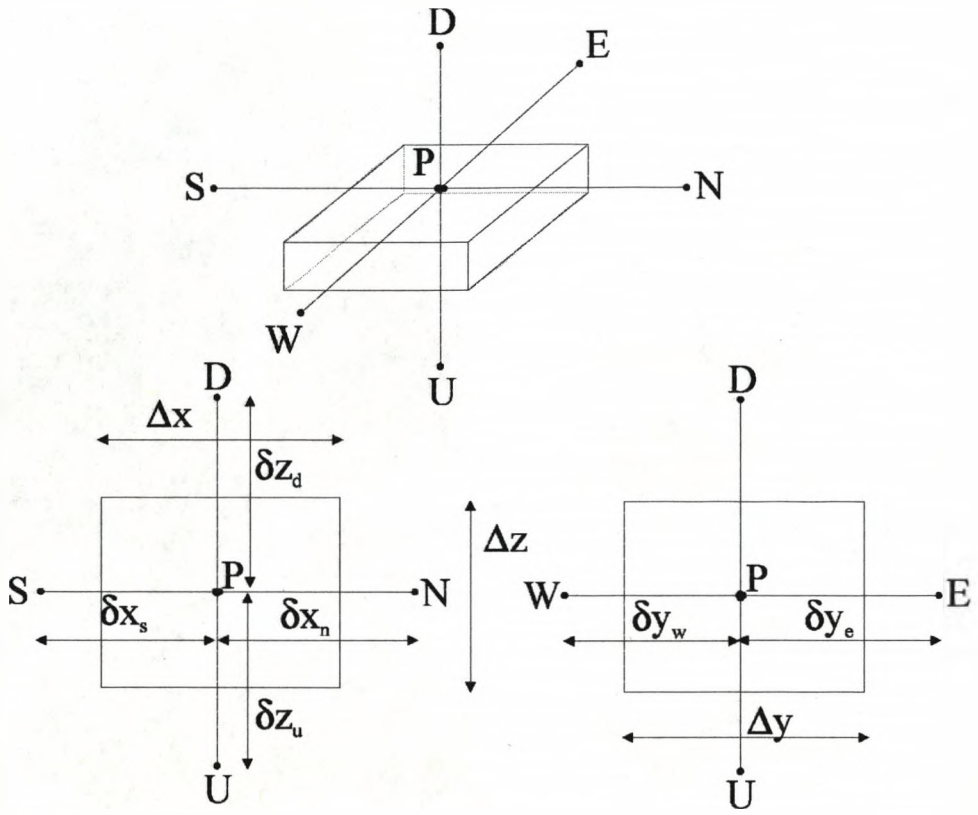


Fig.2.7 Diagram illustrating the dimensions of a control volume in relation to its nearest neighbours. These are used in the calculation to solve the temperature at a particular point.

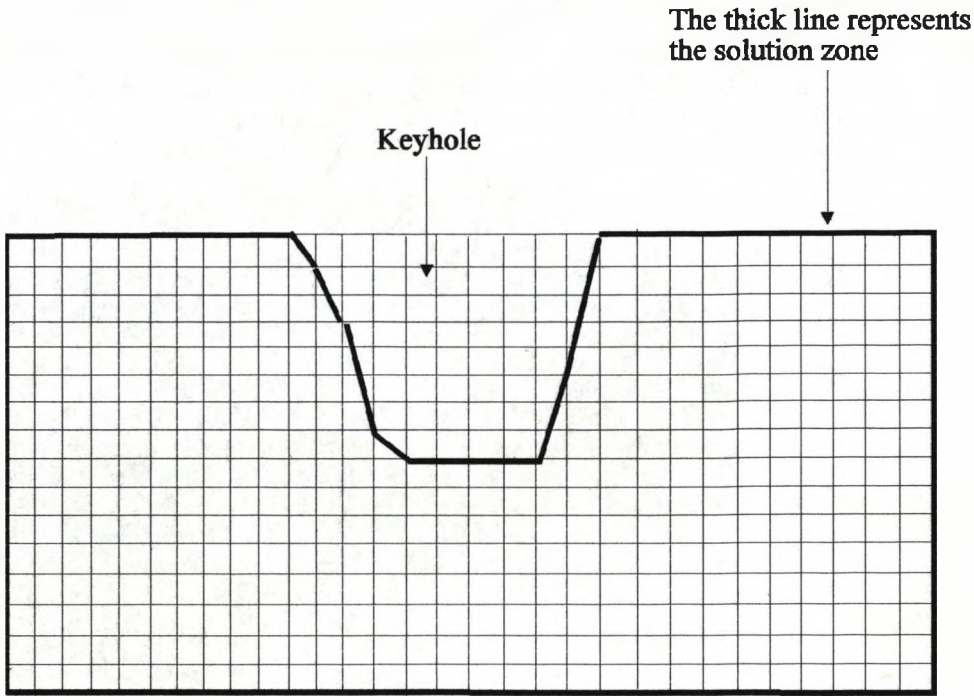


Fig.2.8 Schematic diagram illustrating the way in which the boundary is calculated.

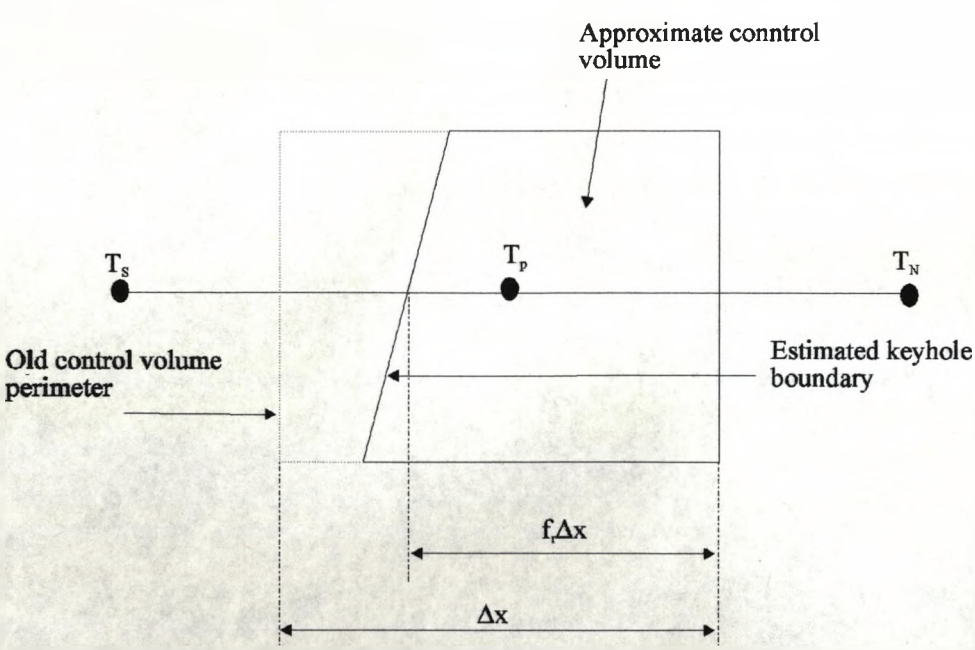


Fig.2.9 Diagram illustrating the calculation of temperature at a boundary node.

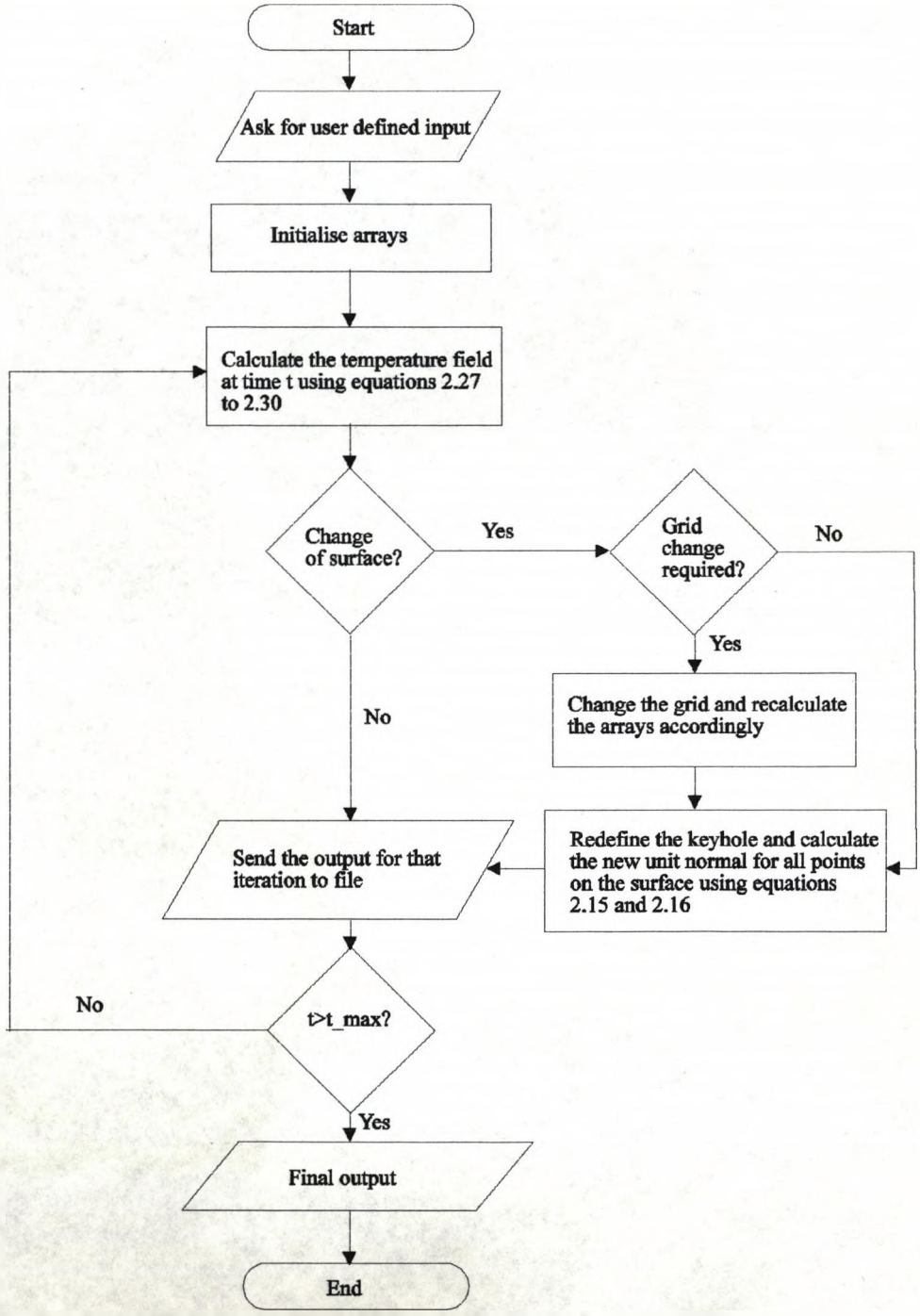


Fig.2.10 Flow chart illustrating the method for solving the pulsed laser welding problem.

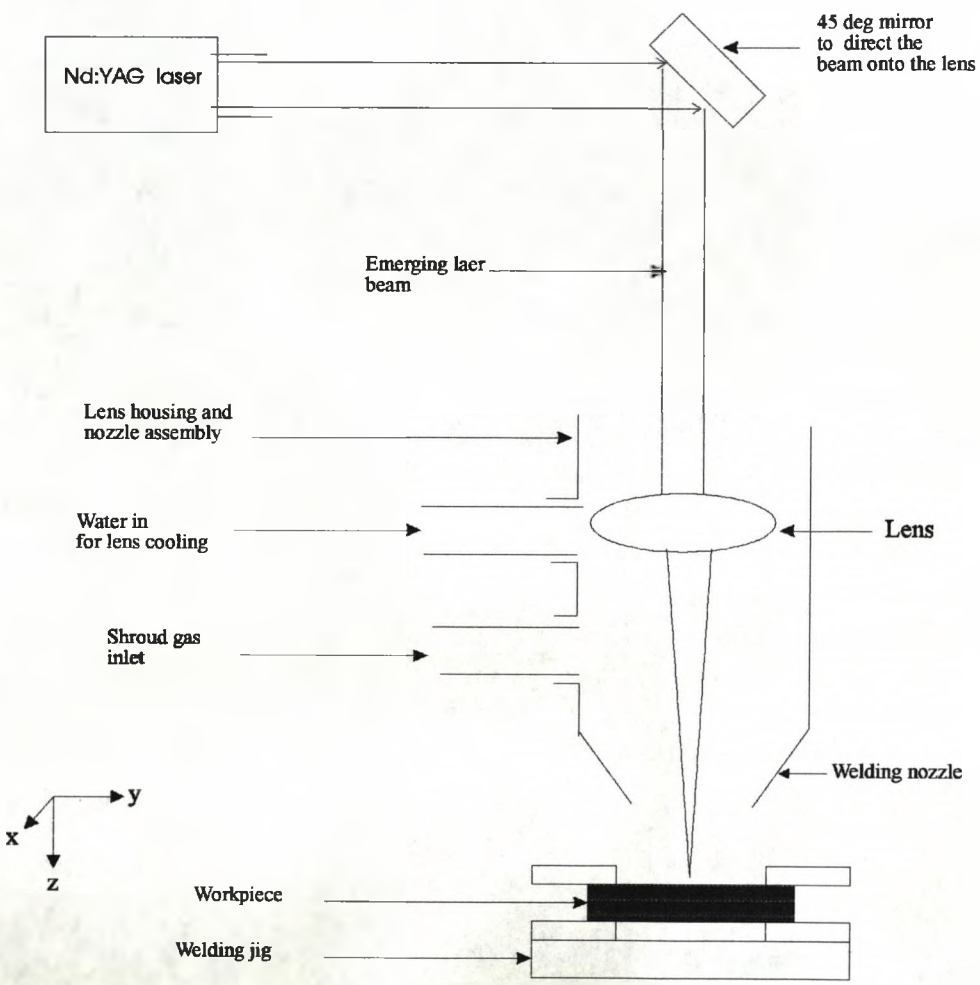
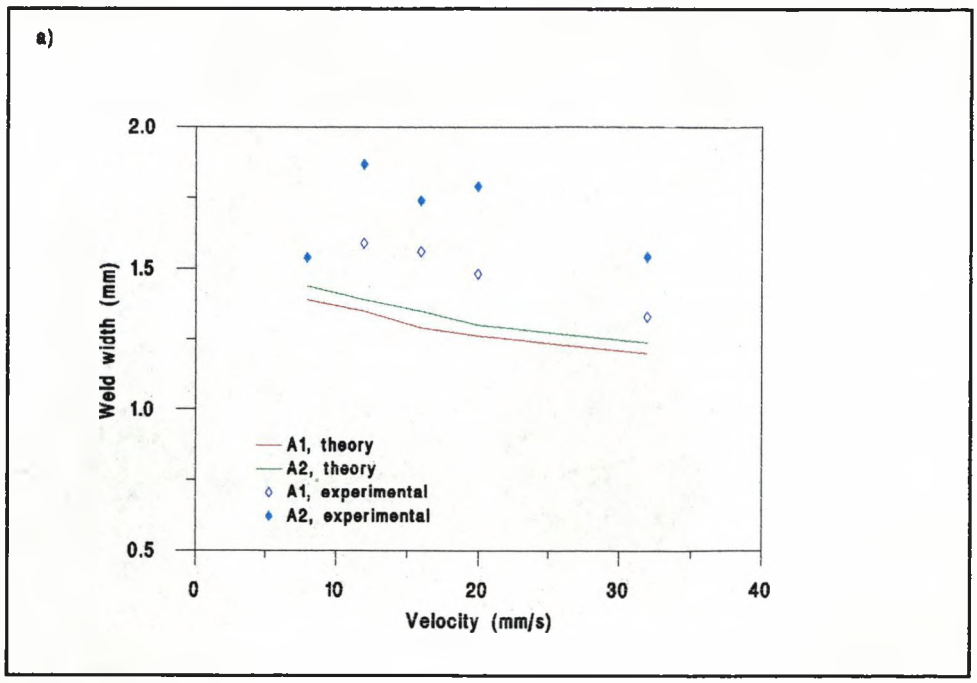


Fig.2.11 Experimental arrangement for pulsed laser welding



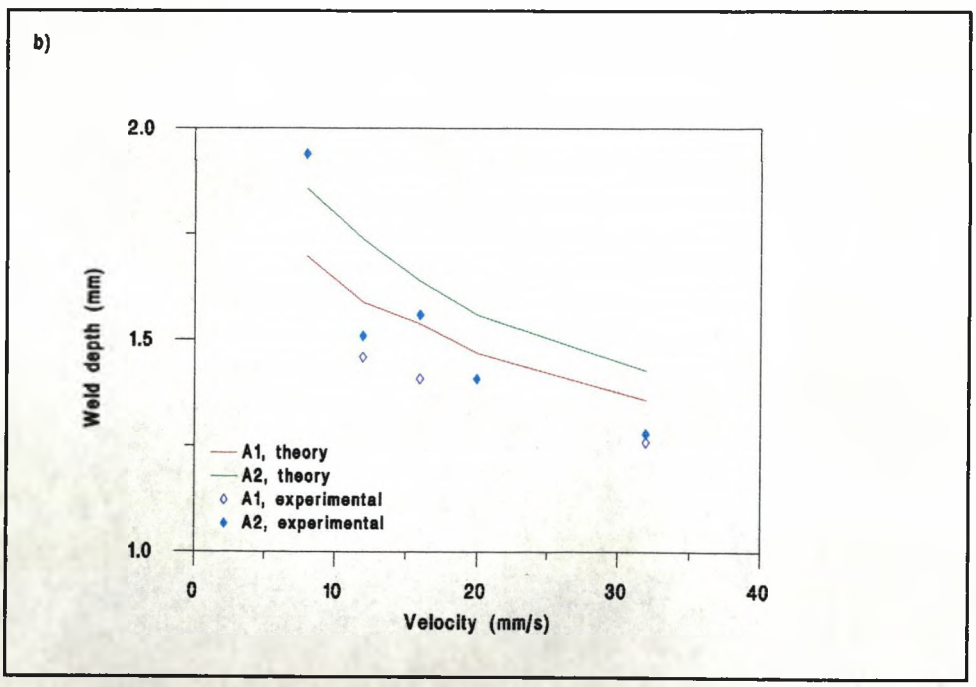


Fig.2.12 Comparison between the results from the theoretical model and experimental data. a) compares the melt width, and b) compares the melt depth. The process parameters used in this comparison are shown in table 2.1.

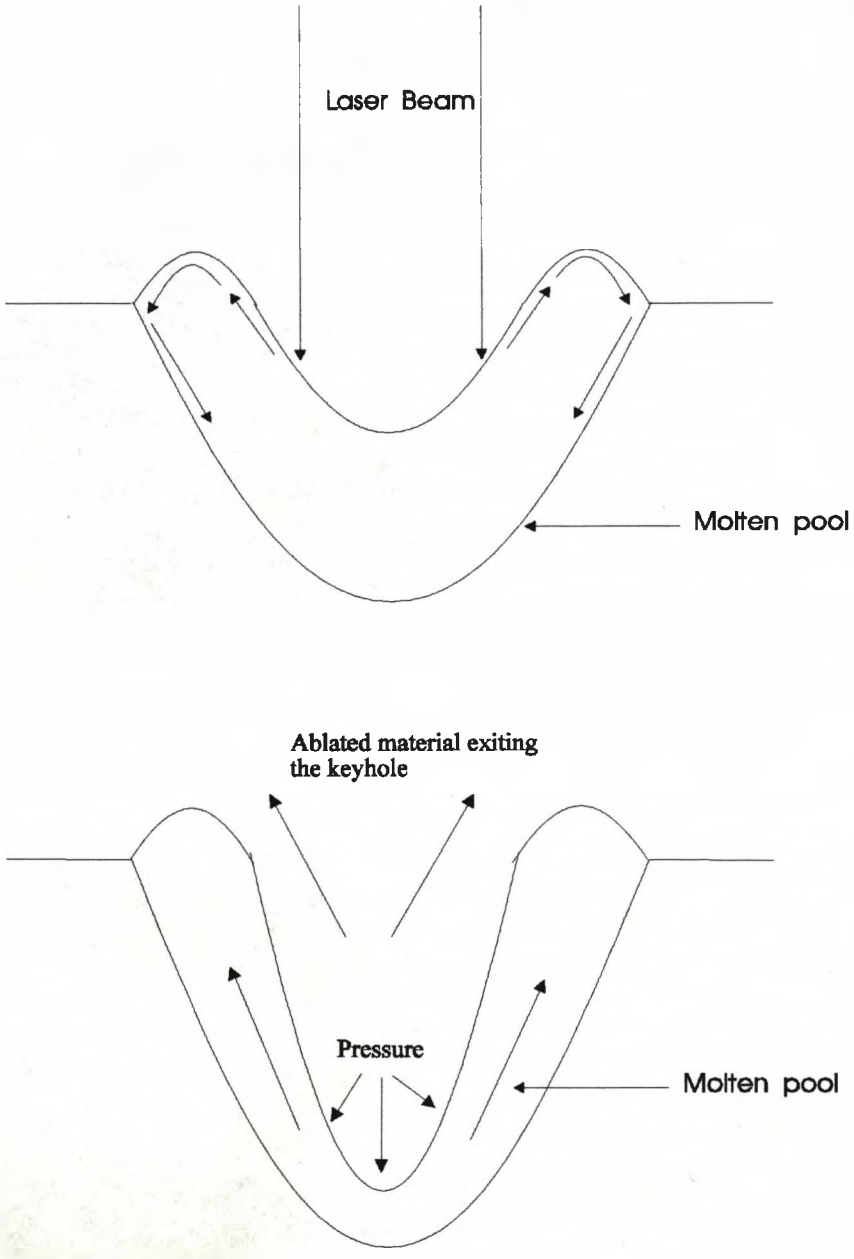


Fig.2.13 Schematic diagram illustrating two of the possible reasons for the difference between experimental and theoretical results. a) Marangoni convection. Here, the flow at the top surface is forced upwards, driven by the surface tension gradients. This causes the hot material to flow towards the cold material, increasing the molten zone, and b) Influence of recoil pressure. Here, the recoil pressure due to ablation forces the keyhole to be depressed at the centre, and causing the hot, molten material to flow upwards, again increasing the melt width.

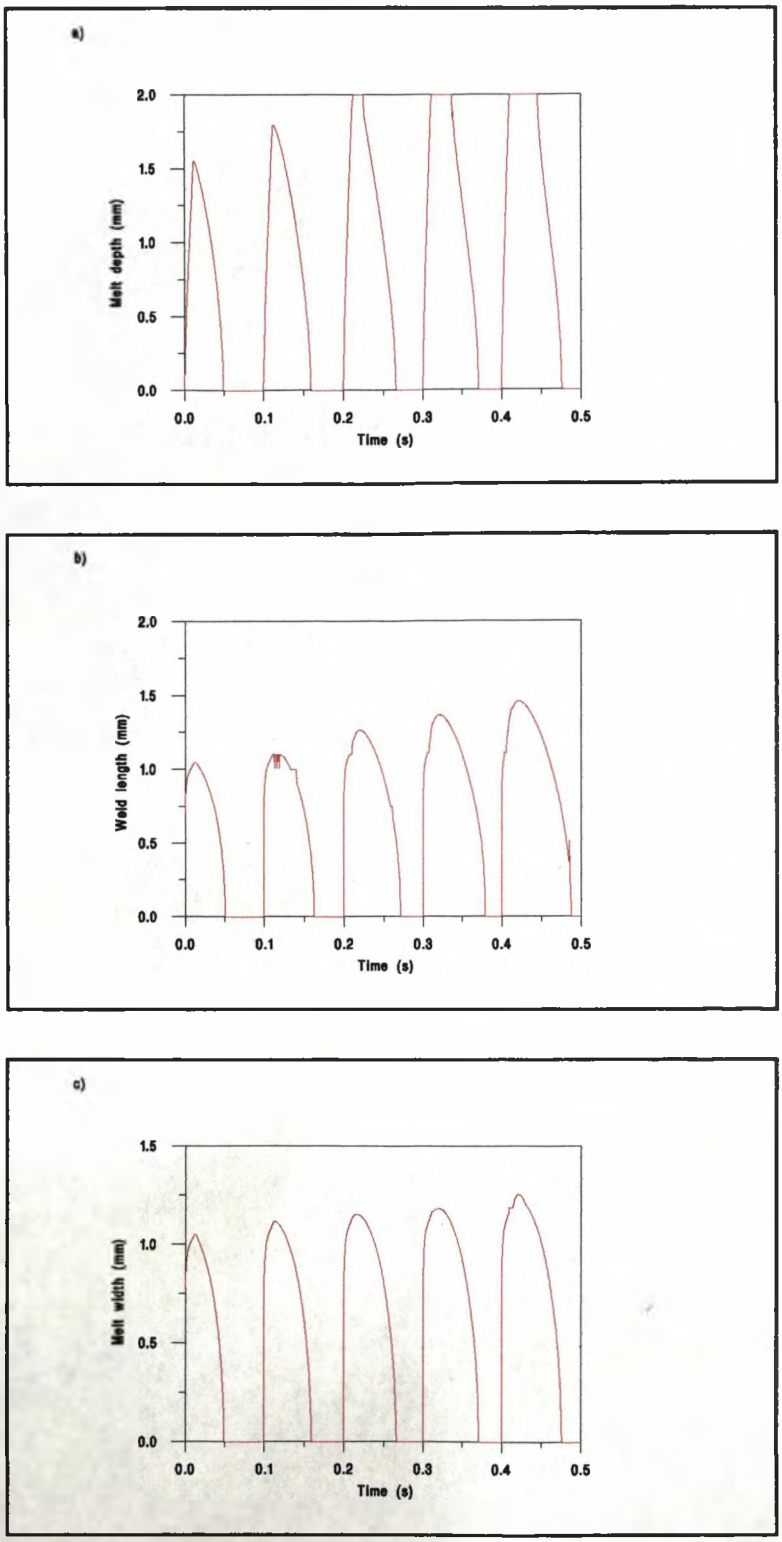


Fig.2.14 Weldpool depth, length and width using a power of 3500W, a beam radius of 0.4mm, a laser on time of 0.012s, a frequency of 10Hz, and a velocity of 8mm/s.

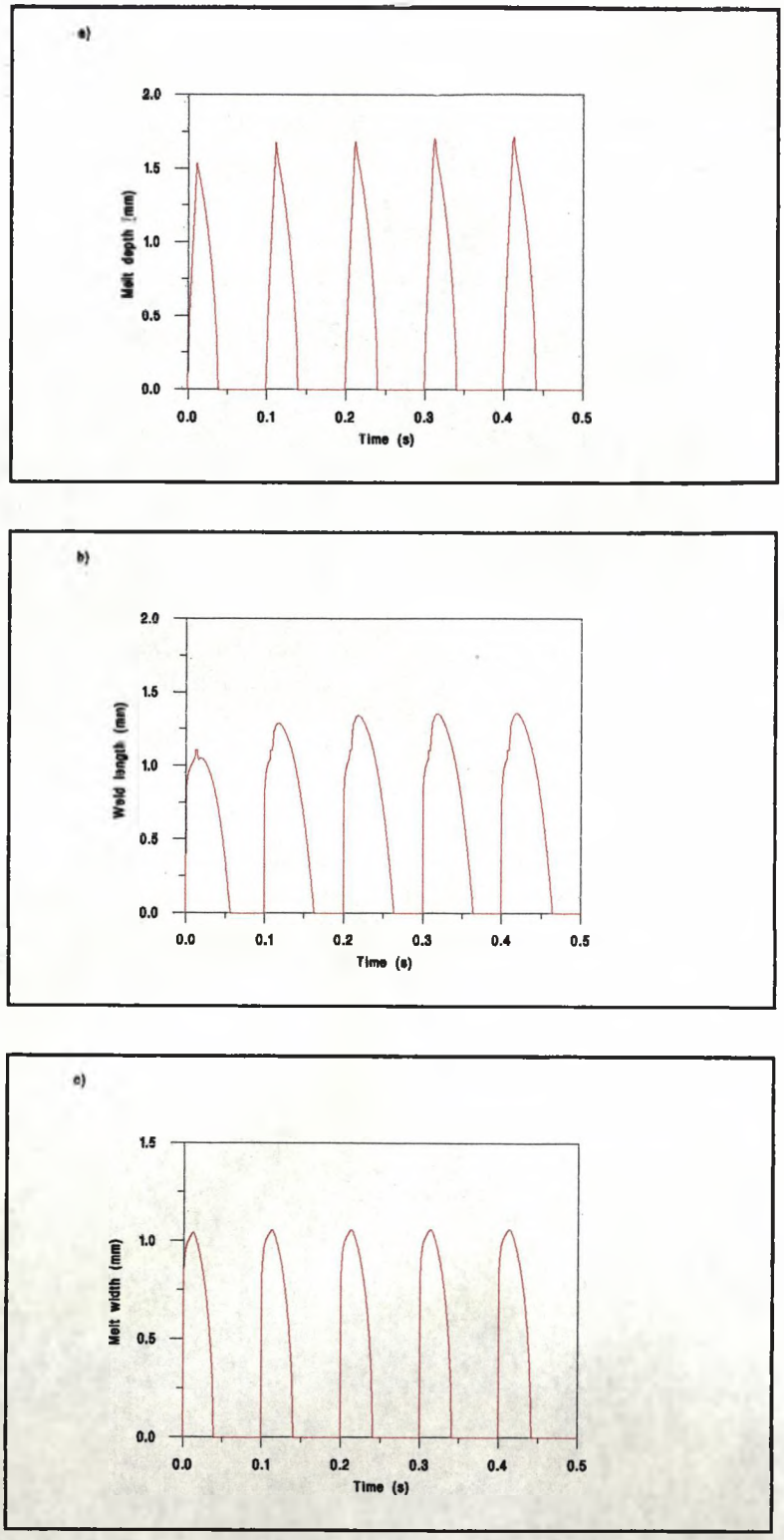
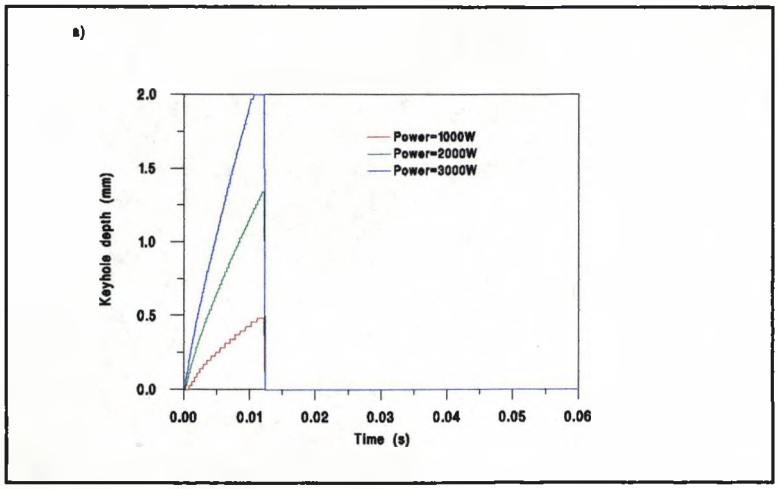
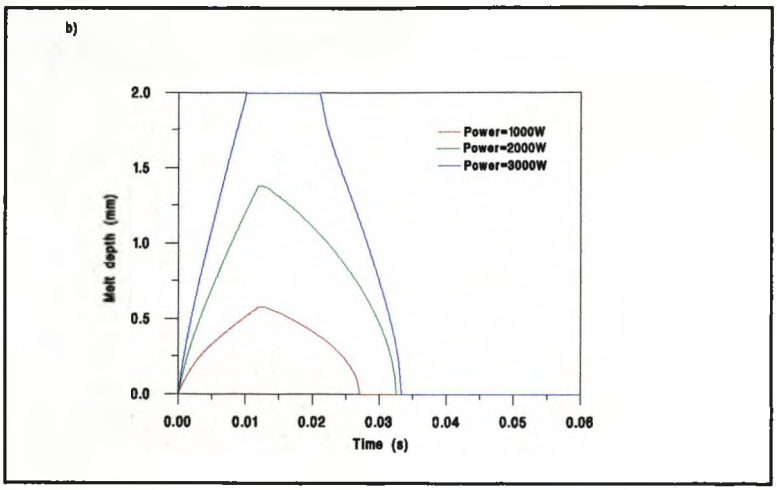


Fig.2.15 Weldpool depth, length and width using a power of 3500W, a beam radius of 0.4mm, a laser on time of 0.012s, a frequency of 10Hz, and a velocity of 32mm/s.





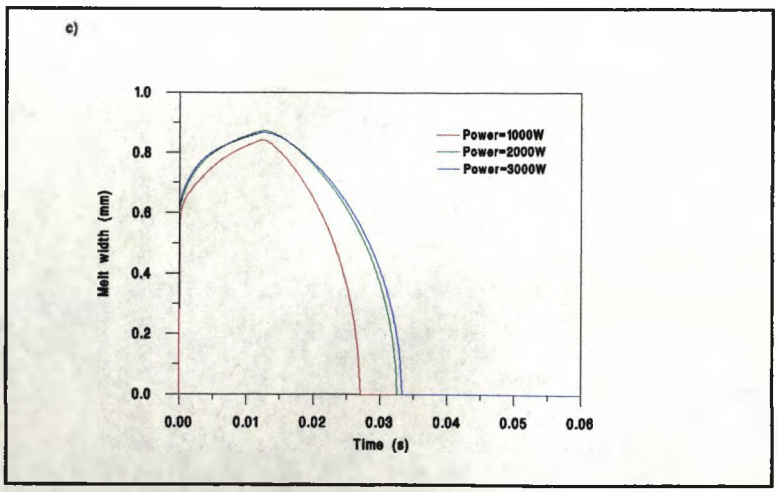


Fig.2.16 Graphs of keyhole depth, melt depth and width for 1 pulse with powers of 1kW, 2kW and 3kW. The other process parameters are shown in table 2.2.

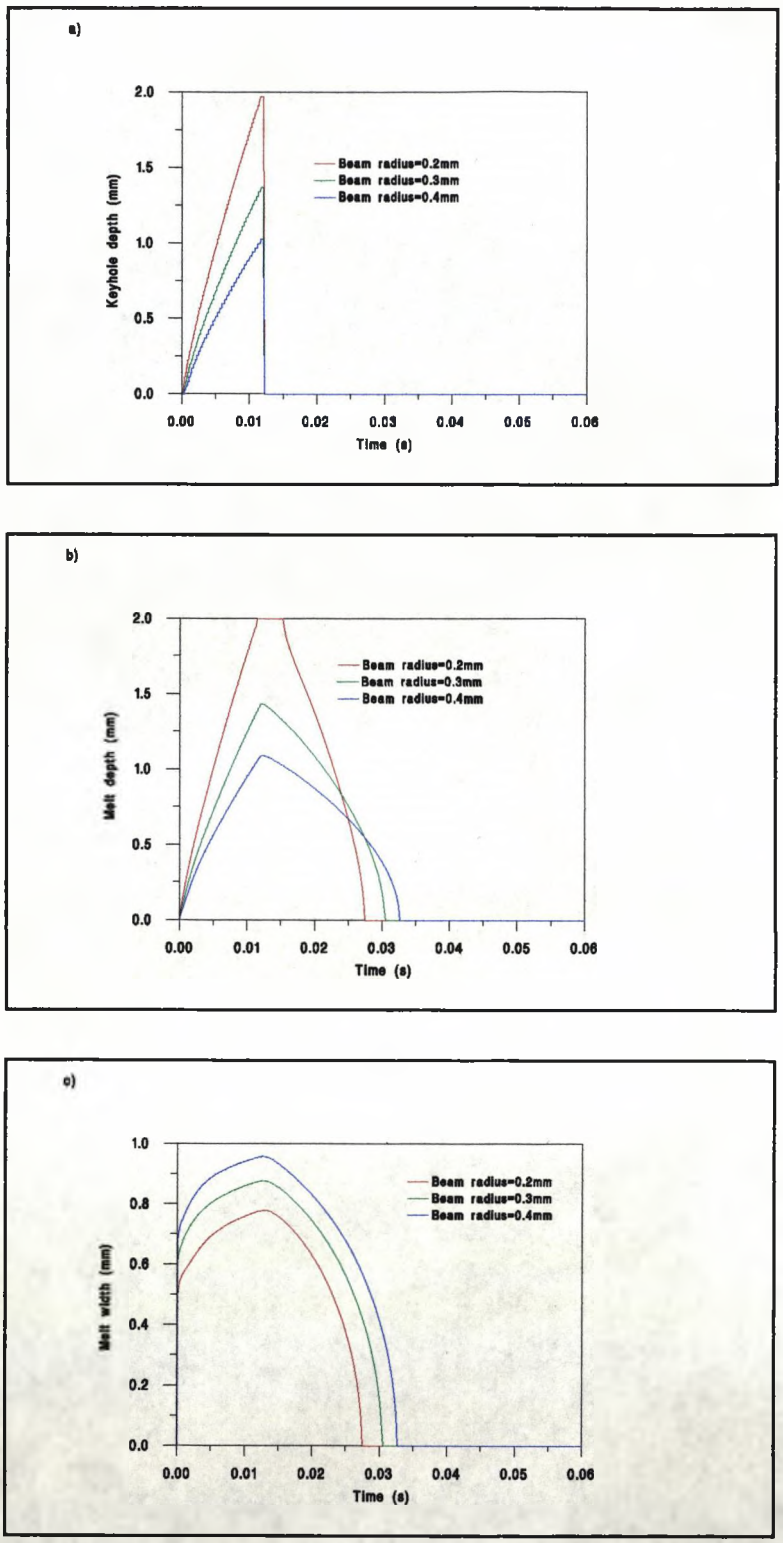


Fig.2.17 Graphs of keyhole depth, melt depth and width for 1 pulse with beam radii of 0.2, 0.3 and 0.4mm. The other process parameters are shown in table 2.2.

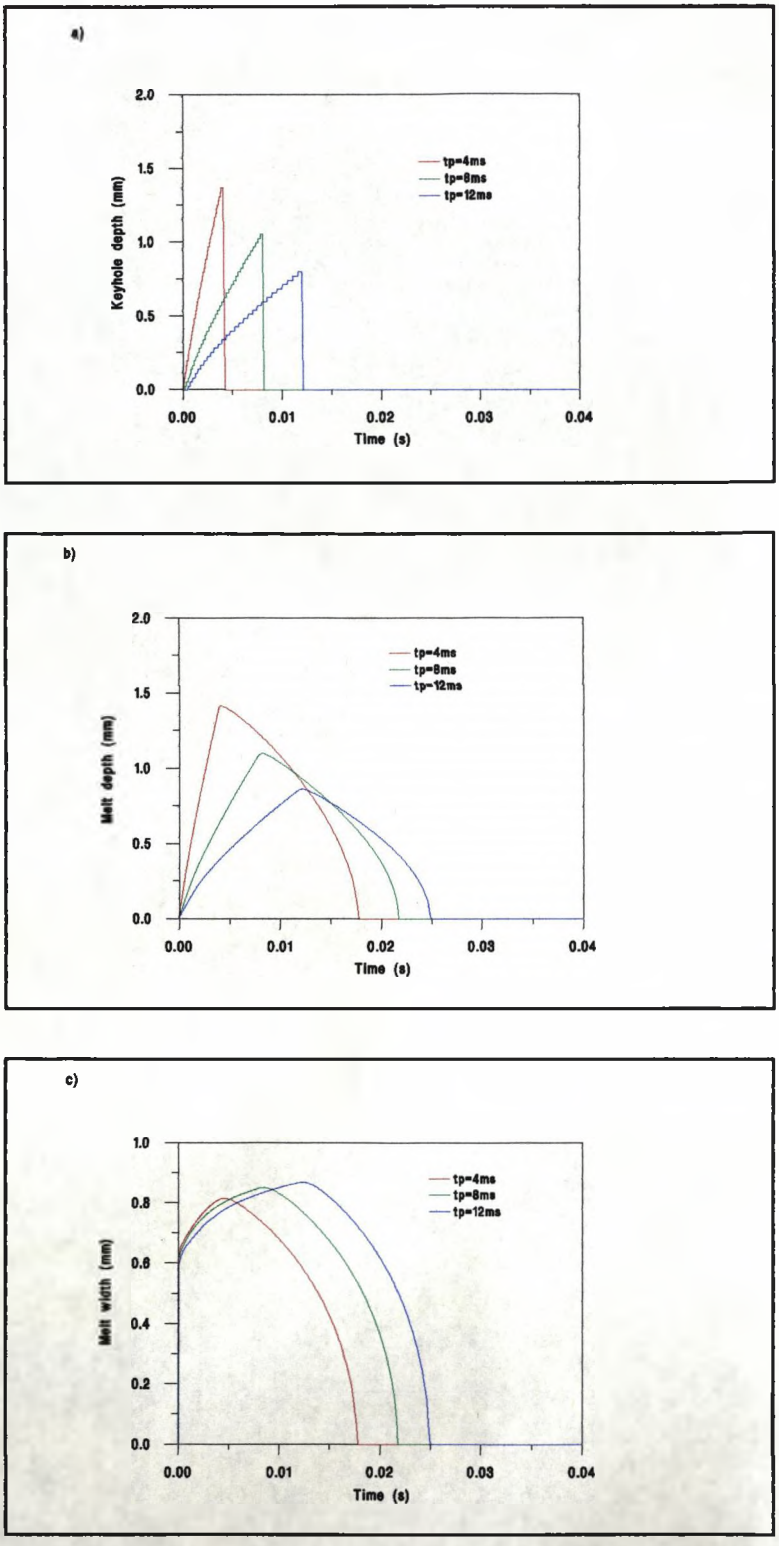


Fig.2.18 Graphs of keyhole depth, melt depth and width for 1 pulse with pulse durations of 4, 8 and 12ms. The other process parameters are shown in table 2.2.

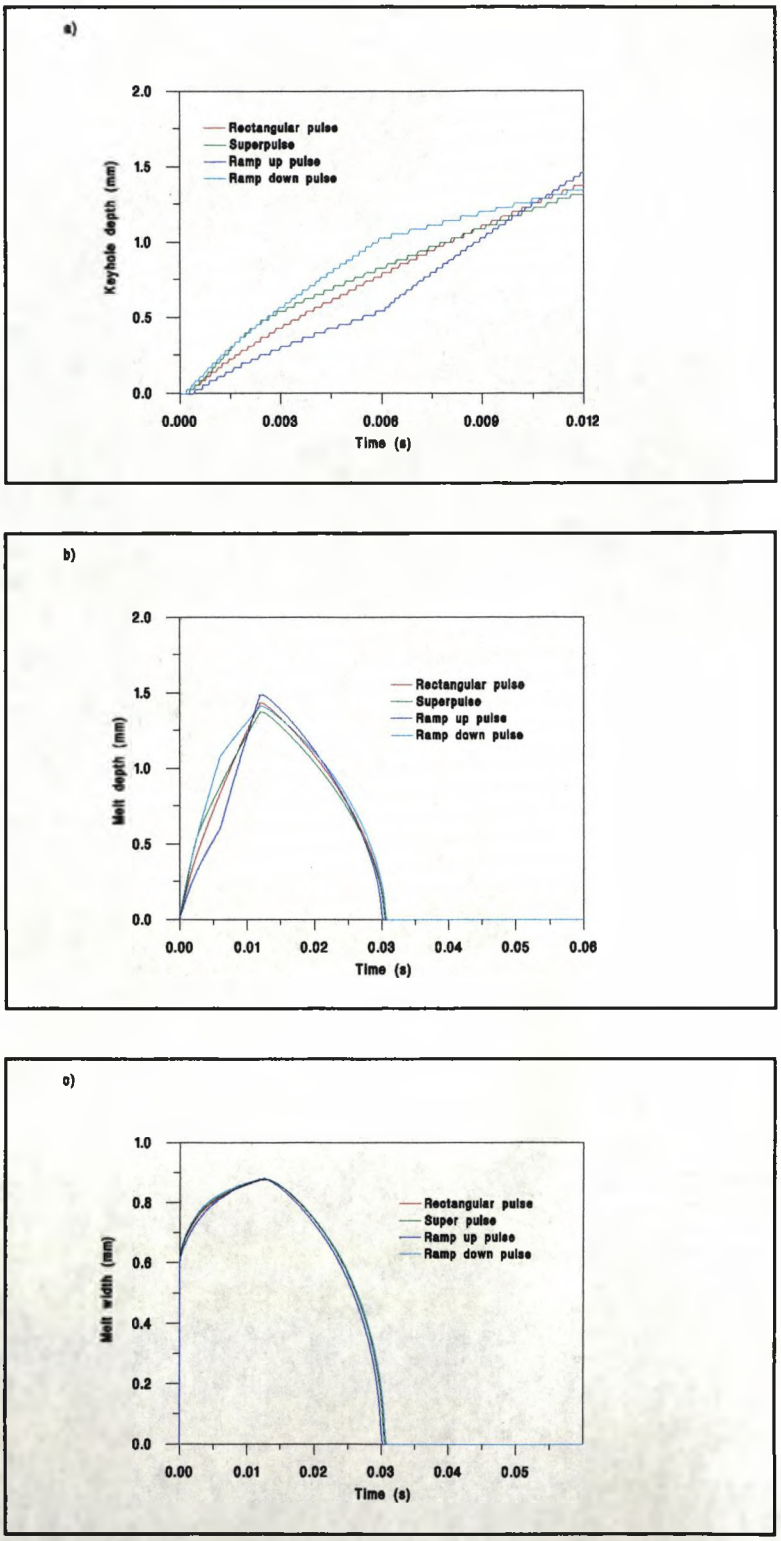
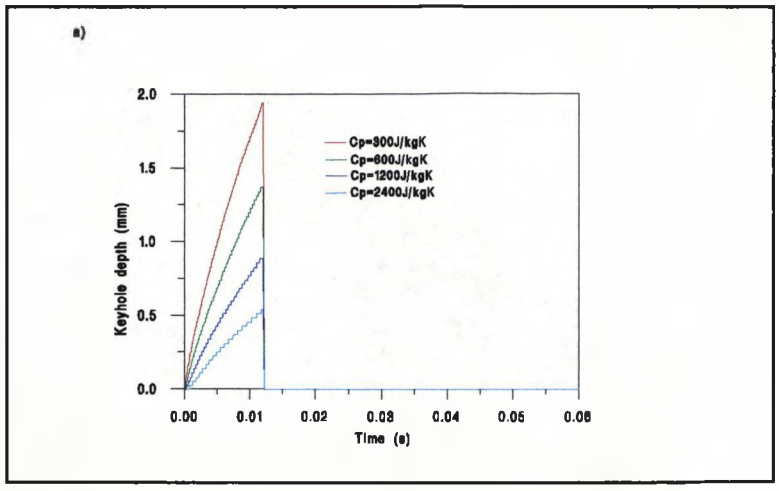
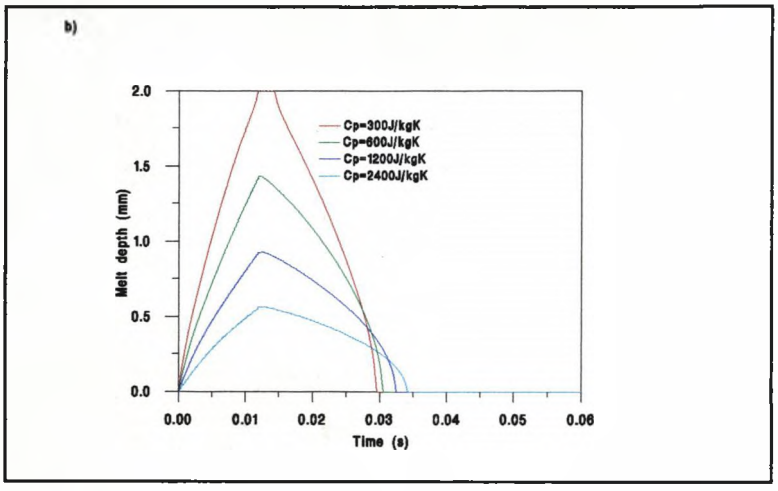


Fig.2.19 Graphs of keyhole depth, melt depth and width for 1 pulse using a rectangular, ramp up, ramp down or superpulse pulse shape. The other process parameters are shown in table 2.2.





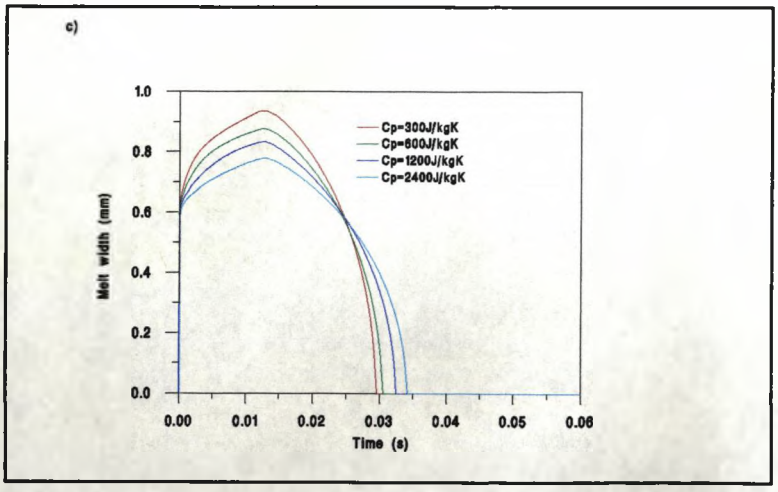


Fig.2.20 Graphs of keyhole depth, melt depth and width for 1 pulse with specific heats of 300, 600, 1200 and 2400/kgK. The other process parameters are shown in table 2.2.

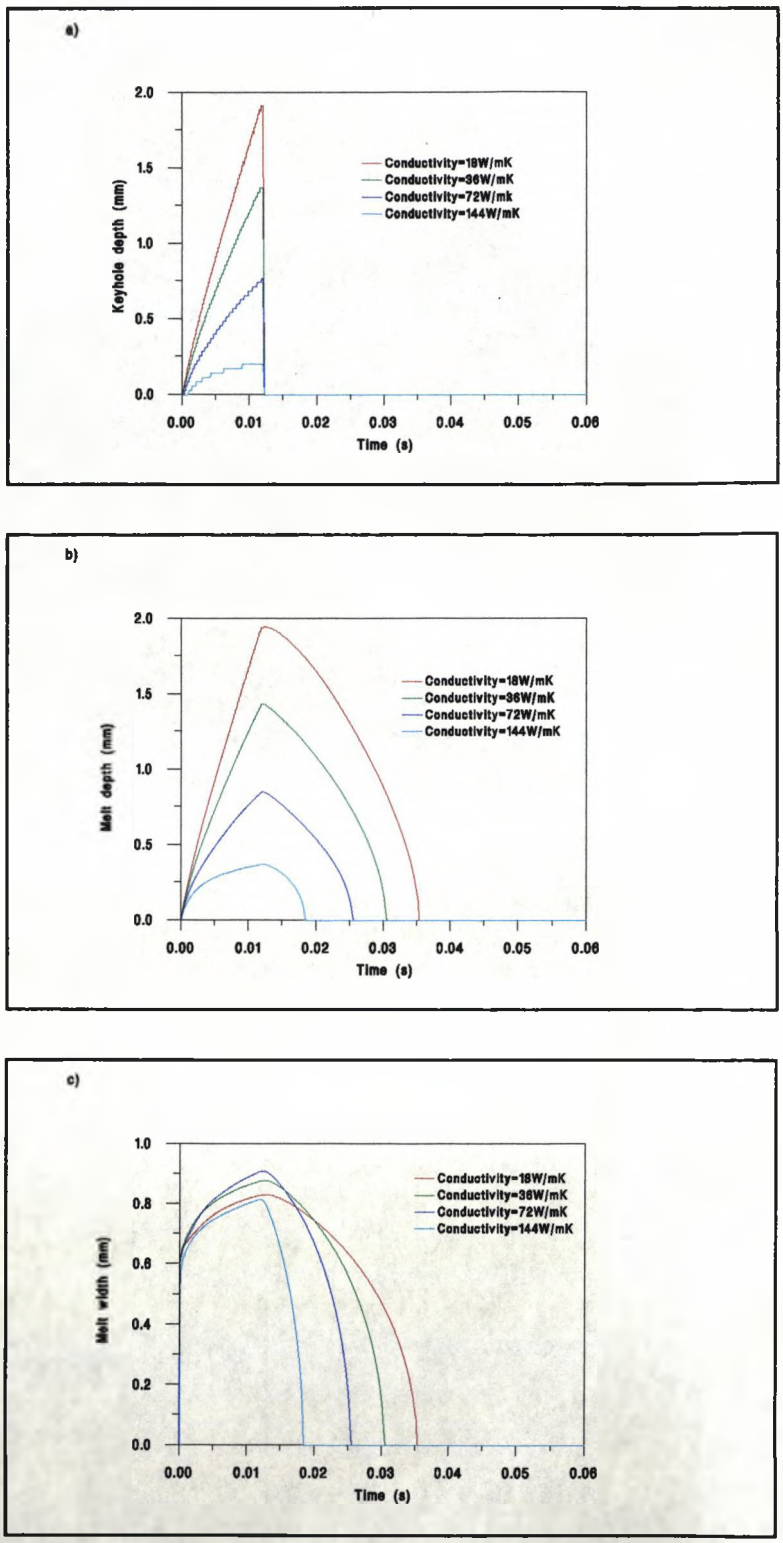


Fig.2.21 Graphs of keyhole depth, melt depth and width for 1 pulse with conductivities of 18, 36, 72 and 144W/mK. The other process parameters are shown in table 2.2.

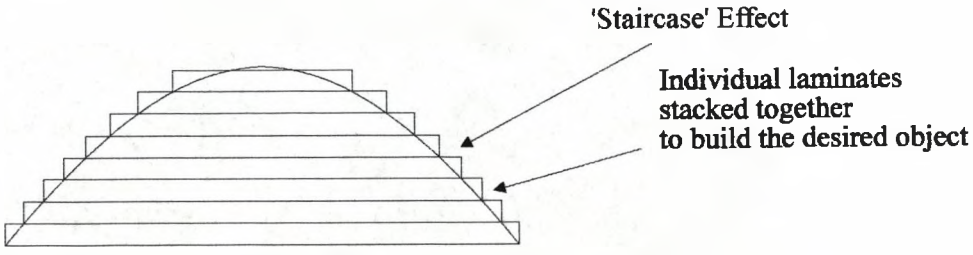


Fig.2.22 Schematic diagram illustrating an object made from thin laminates, as well as illustrating the 'staircase' effect.

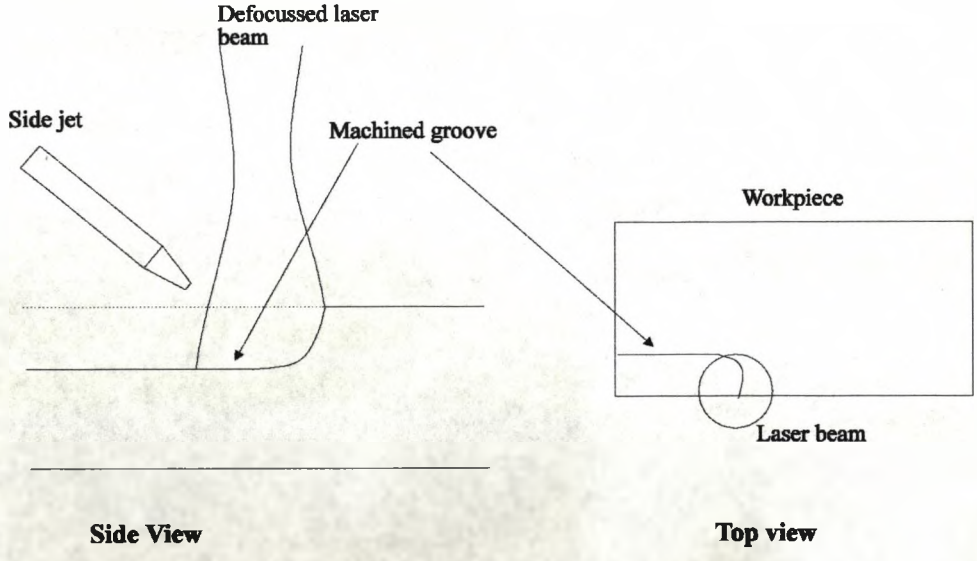


Fig.2.23 Schematic diagram illustrating the pulsed edge machining process.

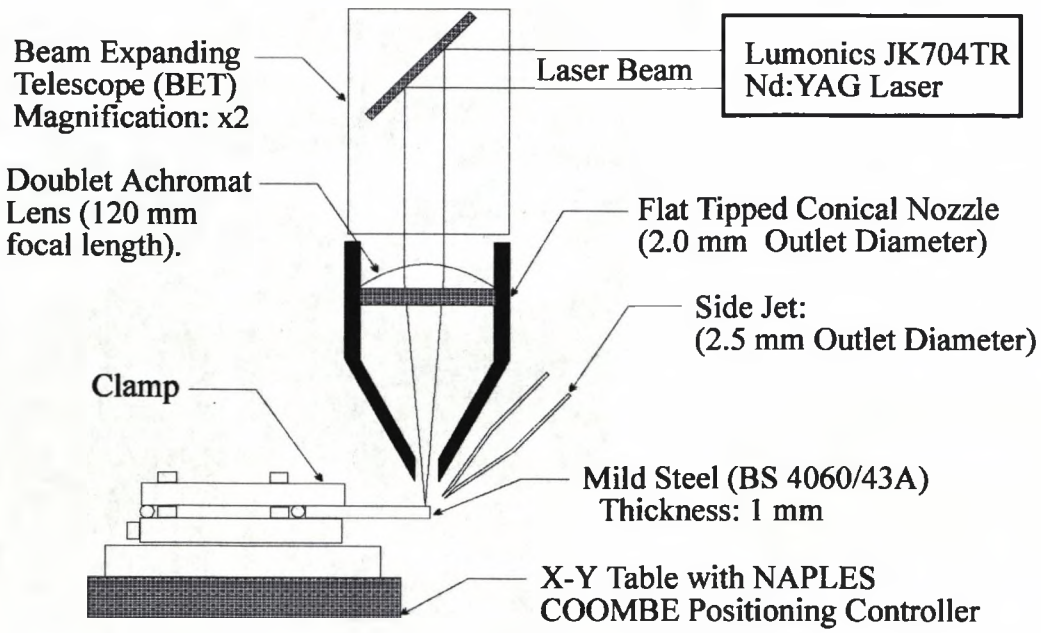


Fig.2.24 Experimental arrangement for the pulsed edge machining process.

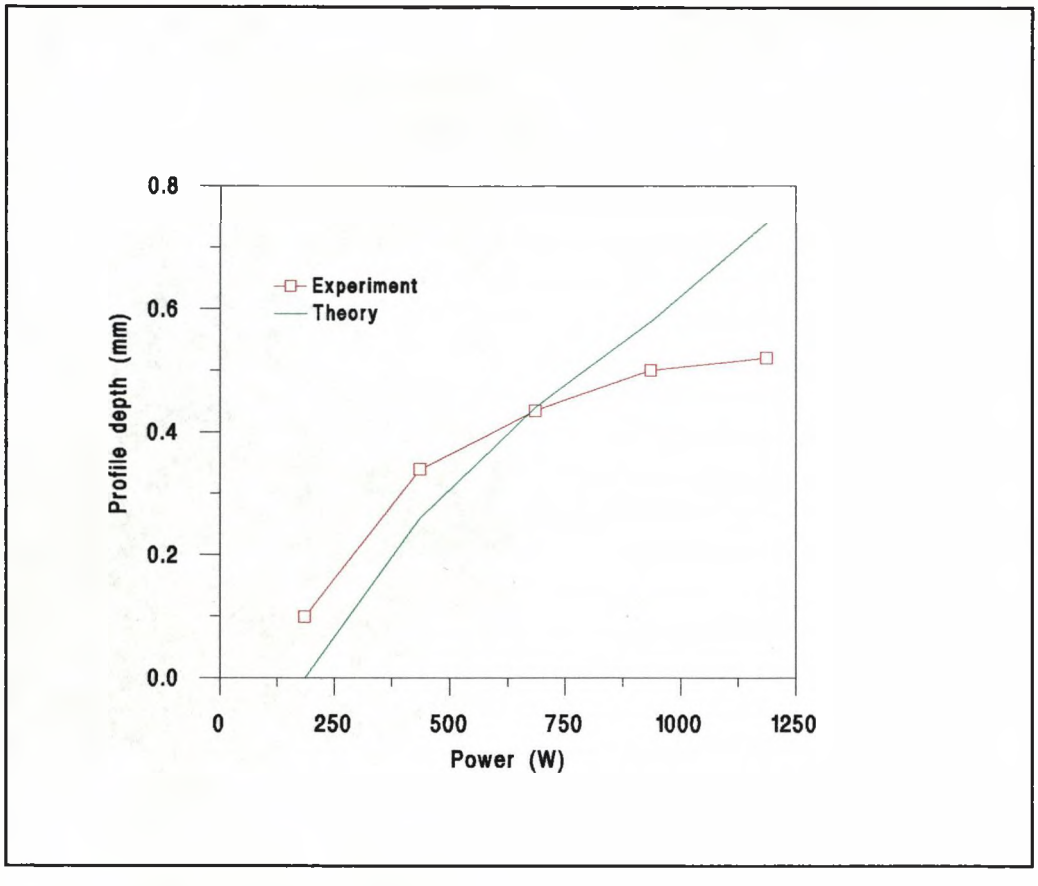


Fig.2.25 Comparison between theoretically calculated results and experimentally obtained pulsed laser machining data.

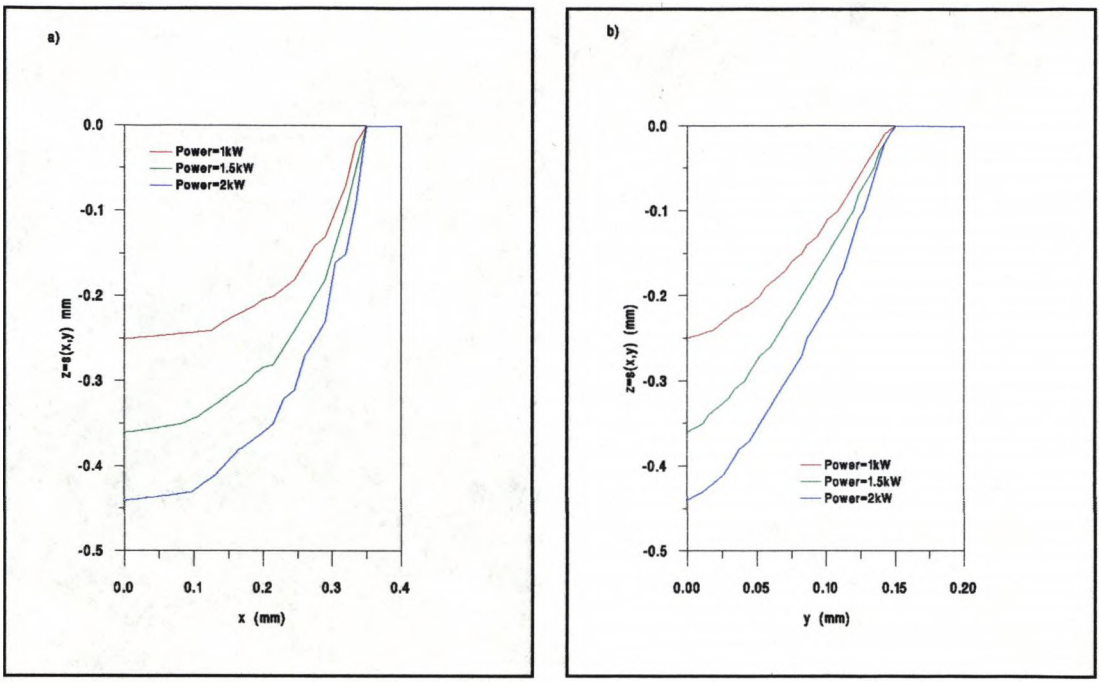


Fig.2.26 Profiles of the machined surface for powers of 1kW, 1.5kW and 2kW. The other parameters are given in table 2.5.

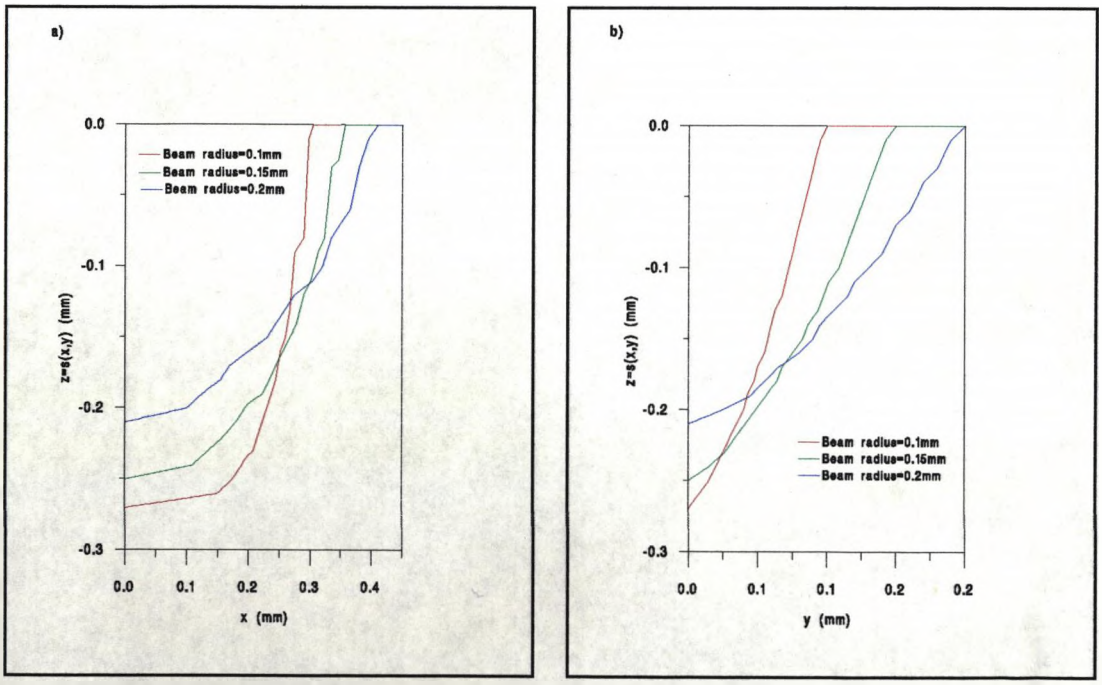


Fig.2.27 Profiles of the machined surface for beam radii of 0.1, 0.15 and 0.2mm. The other parameters are given in table 2.5.

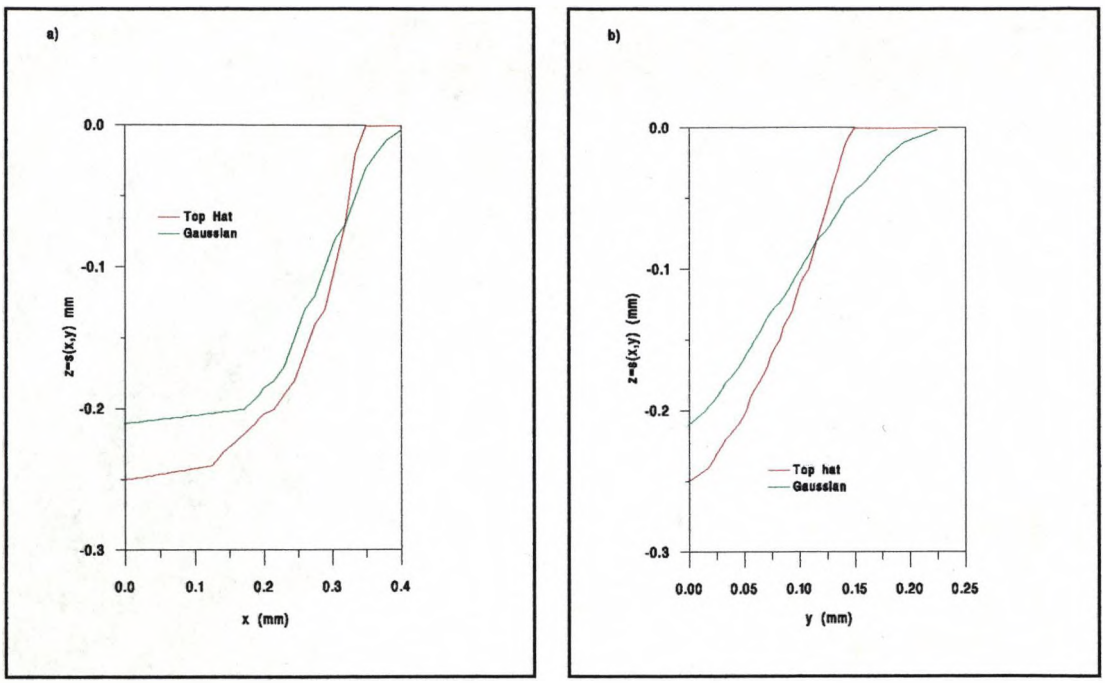


Fig.2.28 Profiles of the machined surface for Guassian and top hat shaped beams. The other parameters are given in table 2.5.

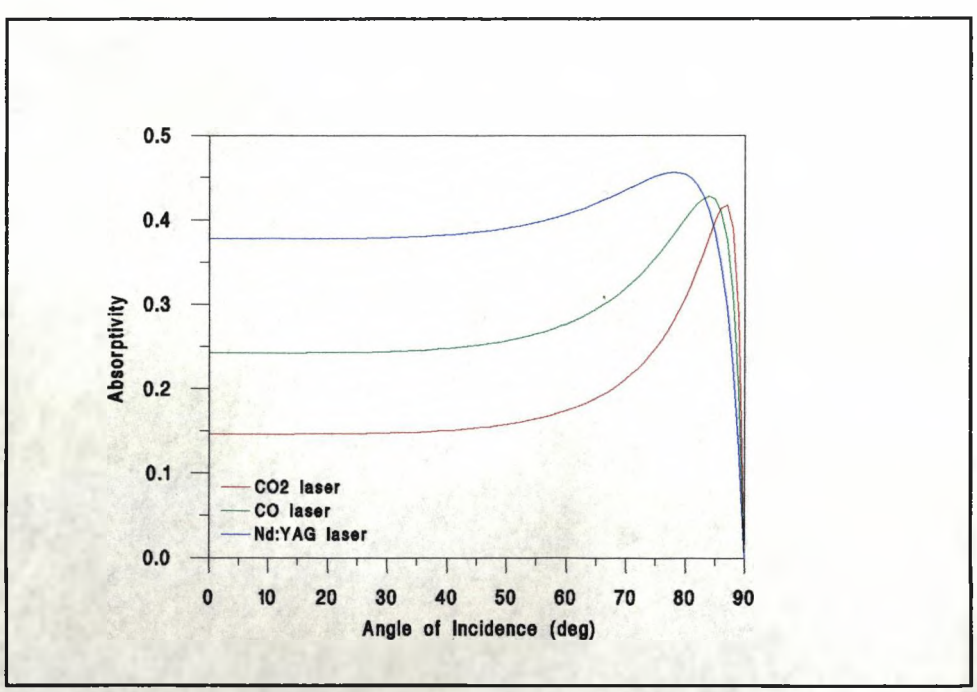


Fig.2.29 Grapth of the absorptivity of mild steel at angles of incidence for Nd:YAG, CO CO_2 lasers.

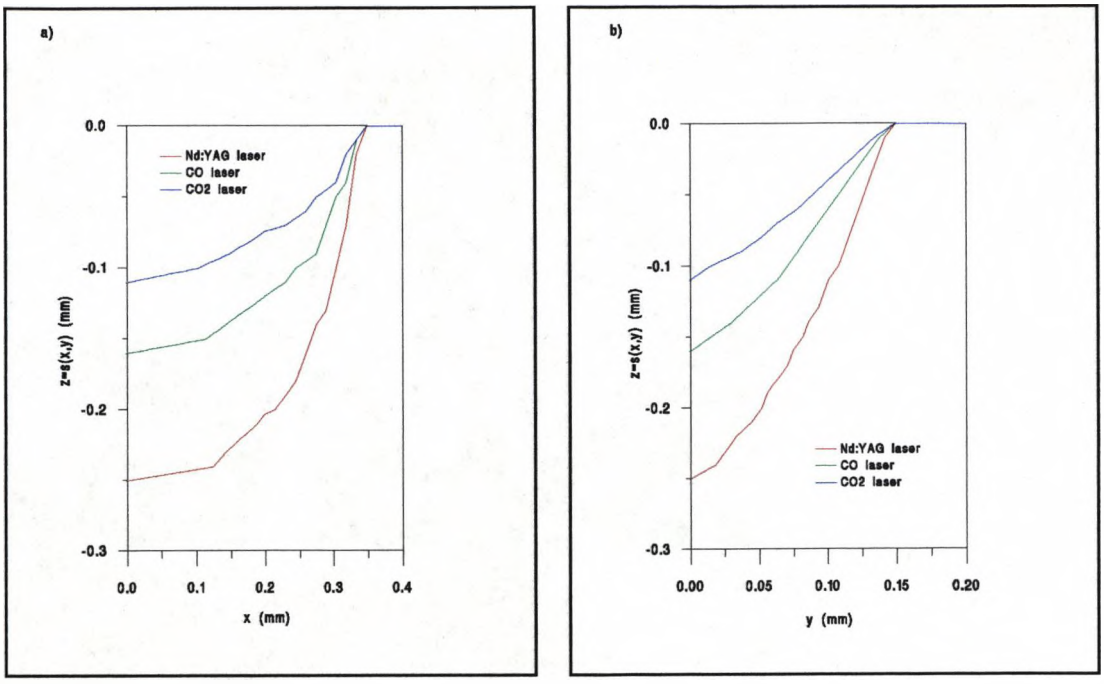


Fig.2.30 Profiles of the machined surface for Nd:YAG, CO and CO_2 lasers. The other parameters are given in table 2.5.

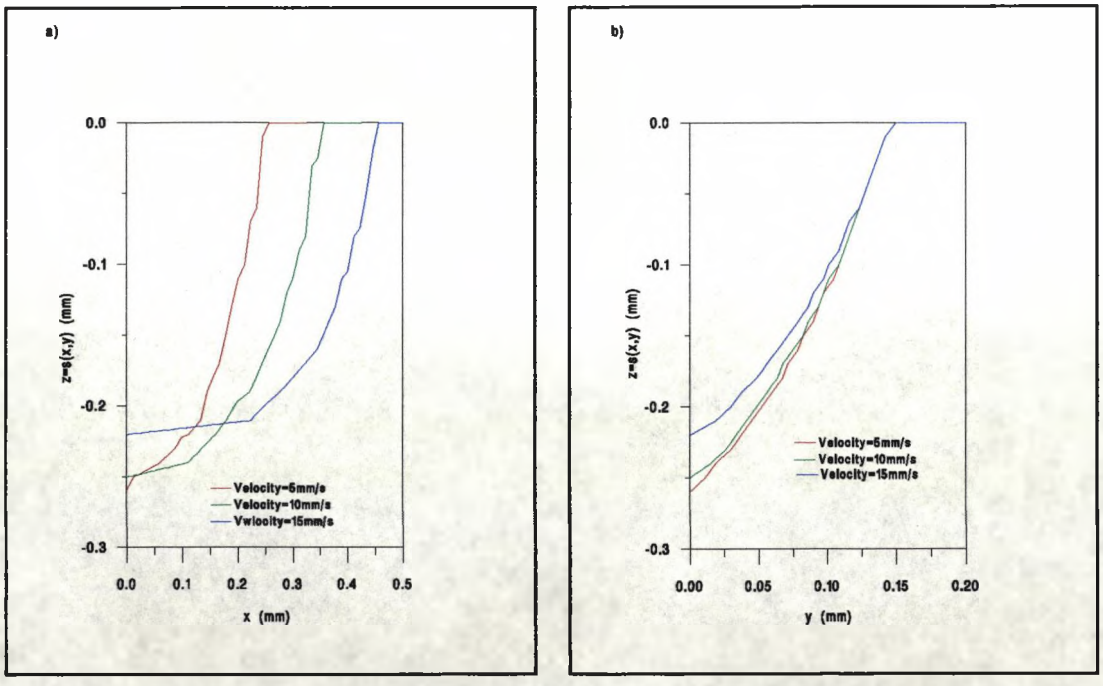


Fig.2.31 Profiles of the machined surface for velocities of 5, 10 and 15mm/s. The other parameters are given in table 2.5.

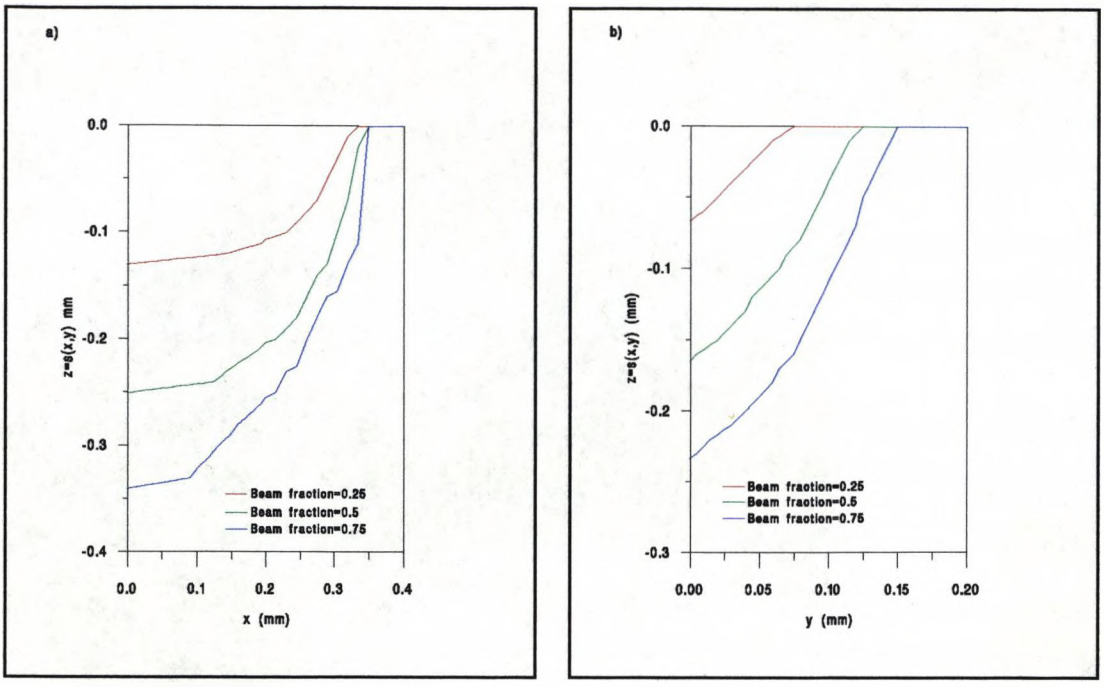


Fig.2.32 Profiles of the machined surface for beam fractions of 0.25, 0.5 and 0.75. The other parameters are given in table 2.5.

Chapter 3 A Model of the Vapour Flow in the keyhole

A brief description of the conditions during the laser keyhole welding process were described in previous chapters, a more detailed analysis will be described here, with a view to obtaining a mathematical description of the conditions within the keyhole and weldpool in order to develop a model of the process itself. The model presented later in the chapter accounts for the pressure and energy balances at the keyhole walls, as well as the flow in the keyhole. The results from the model will be analysed, and in the next chapter will be compared with experimental results.

3.1 The laser Welding Process

A weld is, according to the British standard, 'the union between pieces of metal at faces rendered plastic or liquid by heat or by pressure or by both'. The present investigation concerns the study of the laser keyhole welding of mild steel, from both a theoretical and experimental point of view. During this process, the workpiece is traversed under a focused laser beam which creates a vaporised hole, known as the keyhole, within the material. This is kept open by the recoil pressure of the ablating molecules at the keyhole walls, together with the pressure due to the fluid flow within the keyhole. The keyhole increases the coupling of the energy from the laser beam into the workpiece for a number of reasons. First, it presents a greater surface area in which the beam can interact with the material. Secondly, the workpiece also presents itself to the laser at a high angle of incidence, thus the Fresnel absorption coefficient is much greater. Finally, the vapour forms a plasma, which absorbs the laser radiation by a process known as inverse Bremsstrahlung absorption, this energy is then passed to the keyhole walls by conduction. In some circumstances however, the plasma can undergo an avalanche of ionisation which can become a hindrance to the process itself, by either blocking the beam, or creating a high thermal gradient above the keyhole which defocuses the beam due to the spatial variation in refractive index of the medium, leading to a reduction in power density. In order to protect the weld from excess plasma therefore an inert gas jet is blown across the keyhole exit, thus removing the unwanted ionised gas.

The welding process itself is dependent upon many process variables, a full description of these can be found in Steen (1991), or Williams (1996). Here, only the parameters under investigation, together with their effects on the welding process, will be described. These can be divided into three main groups, described briefly below.

3.1.1 Fundamental process characteristics

The fundamental process characteristics are defined here as the process parameters common to all types of welding, these are, the heat input, traverse speed and the workpiece thickness. Obviously the penetration depth is dependent upon the heat input into the system, and increases with increasing power for any given speed, provided that the gas shielding is adequate enough to protect

the keyhole.

For a given power, a decrease in speed will increase the energy input into the workpiece, which will result in an increase in the penetration depth and weld width. A maximum penetration depth is reached however for a particular power when the plasma in the keyhole and the plume becomes so intense that it hinders the laser welding process, by defocusing the beam above the workpiece and not allowing enough power to reach the keyhole (this effect is known as 'plasma blocking'), and so the penetration depth decreases at velocities lower than this critical velocity.

3.1.2 Laser beam characteristics

Apart from power, the main characteristics of the laser beam which have an effect on the laser welding process are spotsize, depth of focus, beam mode and wavelength of the laser light, these will be discussed below.

3.1.2.1 Spotsize and depth of focus

The laser beam can be focused by a lens or a mirror; the characteristics of which, together with the size and mode of the beam, determine the minimum spotsize and the depth of focus. Generally speaking, the spotsize decreases with decreasing focal length, this can be calculated from the following approximation formula (Kaplan (1994)),

$$d=\frac{4\lambda f M^2}{D\pi} \qquad 3.1$$

here λ is the wavelength of laser light, f is the focal length of the lens, D is the diameter of the laser beam before the lens, and M^2 is the a description of the quality of the beam

The beam radius itself can be calculated over the depth from,

$$r(z)=r_0\sqrt{1+\left(\frac{z-z_0}{z_r}\right)^2} \qquad 3.2$$

where $z-z_0$ is the distance from the focal position, and z_r is a quantity known as the Rayleigh length, which describes the depth of focus,

$$z_r=\frac{2r_0F}{d} \qquad 3.3$$

Figure 3.1 shows how the beam radius varies with depth for 100, 125 and 150 mm focal length lenses. Bearing in mind that a small spot size is required in laser keyhole welding in order to vaporise the material, it seems that the smaller focal length lenses offer the greatest potential. However, it can be seen that this advantage may be outweighed somewhat, by the fact that with small

focal length lenses, the beam rapidly diverges after the focus. This can become a major problem, especially when welding thicker material, therefore a balance has to be reached between spotsize and he depth of focus. This will depend upon the laser beam parameters and the workpiece thickness. In the low to medium power range, where refractive optics are usually employed, focal lengths between 100-200mm are the most common.

3.1.2.2 Beam mode

Figure 3.1 b shows the change in beam radius near the focal position for different values of M^2. As can be seen, the minimum spotsize and the depth of focus both increase with increasing beam mode. The advantage of the small spotsize is the reason why the TEM_{00} laser mode with an M^2 value near unity is preferred in most forms of focused laser beam processing. The lasers used in this study were tuned to give as low an order mode as possible, with the shape of the intensity distribution across the beam being similar to that of a pure gaussian profile. This was not a pure TEM_{00} mode, but consisted of a number of higher order modes which together resemble a gaussian distribution. Thus a larger spotsize than the diffraction limited value was obtained (typically between 0.3 and 0.4mm).

3.1.2.3 Wavelength of laser light

It can be seen from equations 3.1 to 3.3, that the both the minimum spotsize and the depth of focus are affected by the wavelength of light. Assuming all other parameters constant, figure 3.1 c) shows the variation of beam radius near the focus for $C0_2$, CO and Nd:YAG laser light. As can be seen, the shorter wavelengths will focus to a finer spot. However, in reality, the M^2 value is proportional to the Fresnel number, which in turn is inversely proportional to the wavelength, and therefore, a reduction in wavelength will result in a corresponding increase in the M^2 value. The result of this will be that there is no difference in the beam radius anywhere along the beam path (figure 3.1d). In reality the CO laser light, which operates over a wider band than the $C0_2$ laser light, could yield a larger spotsize due to the chromatic abberation effects mentioned earlier. The minimum Nd:YAG laser spotsize is even larger due to the very poor mode quality of the 1.06μm beam.

Both mechanisms by which the laser light is absorbed in the welding process, ie the plasma absorption in the keyhole and the plume, and the Fresnel absorption at the keyhole walls, are wavelength dependent. The amount of power absorbed in the plasma is proportional to the square of the wavelength. Thus $C0_2$ laser light, which has a wavelength of 10.6μm, will be absorbed by a plasma approximately 4 times as much as CO laser light, which operates at 5 4μm. Thus it would be expected that plasma is much less of a problem in CO laser welding than it is in CO_2 laser welding. However, the beneficial effects of the plasma presence, such as greater absorption, would be diminished. The Fresnel coefficient increases with decreasing wavelength in the infra red region,

and so there would be a gain in Fresnel absorption for most angles of incidence. The combined effects of changing these two parameters, from both a theoretical and experimental perspective, will be discussed in the next chapter.

3.1.3 Gas characteristics

Due to the fact that this plasma is not completely transparent, the laser beam is absorbed in the plume as well as the keyhole. A high temperature gradient is obtained in the region around the plume, with a corresponding spatial variation in refractive index. This will tend to defocus the beam above the workpiece. The greater absorption within the plume, the greater the amount of ionisation, which results in the region becoming more opaque to the laser beam, and so the region absorbs more energy. This cycle would continue with the plume becoming hotter and hotter and absorbing more and more energy; resulting in the reduction of laser beam intensity due to plasma blocking and defocusing of the beam above the workpiece, and thus hindering the welding process, if it were not for the fact that a gas shroud is blown across the plume in order to suppress the plasma.

The shroud gas is usually inert, and therefore has a greater ionisation energy than the iron vapour (approximately twice as much for argon, and three times as much for helium), and so can absorb much more energy from the beam before the cycle of plasma absorption described above becomes a hindrance to the laser welding process. It also removes the ionised gas from the welding region, which cools the system. The main shroud gases used in the laser welding process are argon and helium. Argon is cheaper, and is therefore used in some applications. It is also a heavier gas, and so it can remove the iron plasma more easily than helium. Helium however, has the added advantage of having a greater ionisation energy, this is more advantageous at lower welding speeds and higher powers, i.e. when the plasma absorption is most likely to cause a hindrance to the welding process. Despite the addition of the shroud gas however; the power absorbed in the shroud is not insubstantial, and at very low welding speeds becomes critical, a point is reached beyond which greater penetration cannot be obtained.

3.1.4 The Weld

Referring back to figure 1.15, which shows a diagram of a typical keyhole weld, one can see that this is characterised by the dimensions of the molten zone. The main features of the dimensions are the penetration depth, and the top and stem weld widths. The values of these dimensions describe how the power from the laser is coupled into the workpiece. As Steen et al (1988) have shown, this heat input can be approximated by a line source throughout the depth of the weld which represents the absorption of light in the keyhole, and a point source at the top of the weldbead, which represents the absorption of light in the plume just above the workpiece. It should be noted that in reality this picture is somewhat simplistic, both the keyhole and plume are volume

sources, with spatially varying energy distributions, however, these approximations have proved very useful in the past in analysing welds obtained from experiment.

3.2 The Model

3.2.1 Energy and Pressure balance at the keyhole walls

The energy and pressure balance equations at the keyhole wall have already been described in detail in chapter 1. What follows is a discussion on a keyhole model that takes into account the work of previous authors as well as taking it further by solving the two dimensional flow distribution within the keyhole.

The energy balance at the keyhole walls can be described in terms of the power absorbed per unit depth; here the total power P(z) crossing a given section of the keyhole parallel to its axis is related to the power absorbed per unit depth at the keyhole wall, Q, by the relation (Ducharme et al (1995))

$$\frac{dP(z)}{dz} = -Q \qquad 3.4$$

where z is a coordinate parallel to the axis of the keyhole. The main contributions to Q are from inverse Bremsstrahlung absorption, Q_B, in the keyhole itself, and from Fresnel absorption at the walls of the keyhole, Q_F. This energy is then used in ablating the material at the keyhole wall, Q_v, and conducting the heat throughout the material, Q_λ, thus,

$$Q_B + Q_F = Q_\lambda + Q_v \qquad 3.5$$

Q_F is the power input per unit depth at the keyhole wall due to the Fresnel absorption mechanism. This can be calculated from the power that has not been absorbed at depth z, and radial position r, P(r,z), the beam radius, r_b, the r coordinate, r; the gradient of the keyhole wall dr/dz, and the absorption coefficient A,

$$Q_F = P(r,z)A(\theta)\frac{2}{r_s}\frac{dr_s}{dz} \qquad 3.6$$

A can be calculated by (Schulz et al (1987)),

$$A(\theta) = 1 - \frac{1}{2}\left(\frac{1+(1-\epsilon\cos\theta)^2)}{1+(1+\epsilon\cos\theta)^2)} + \frac{\epsilon^2 - 2\epsilon\cos\theta + 2\cos^2\theta}{\epsilon^2 + 2\epsilon\cos\theta + 2\cos^2\theta}\right) \qquad 3.7$$

where ε is a material dependent property defined as,

$$\epsilon^2=\frac{2\epsilon_2}{\epsilon_1+\sqrt{\epsilon_1^2+(\frac{\sigma_{st}}{\omega\epsilon_o})^2}} \qquad 3.8$$

where ε_1 and ε_2 are the real parts of the dielectric constants for the metal and plasma, and σ_{st} is the electrical conductance per unit depth. Typical values for iron give $\sigma_{st}=5.0\text{x}10^5$ and $\varepsilon_1=\varepsilon_2=1$. For a $C0_2$ laser with an angular frequency of $1.78\text{x}10^{1}m^{-1}$, this yields a value for ε of 0.08 (Ducharme et al (1995)).

Q_B is calculated from the inverse Bremsstrahlung absorption in the plasma, which is calculated from Dowden's formula for linking intensity (Dowden (1987)).

$$Q_B=\sqrt{LP_0} \qquad 3.9$$

where P_0 is the total input power, and L is defined to be the linking intensity of the plasma. This is assumed to be constant for normal working conditions, and has a value is $3.56kWcm^2$ for CO_2 laser wavelength.

The power per unit depth used up in ablating the material is given by

$$Q_v=2\pi r_s\rho_g u_g\Delta L \qquad 3.10$$

where ρ_g is the density of the gas, r_s is the keyhole radius, ΔL is the latent heat of vaporisation, and u_g is the gas velocity at the keyhole wall (this is a function of the pressure difference evaluated at the keyhole wall, which is calculated by solving the Navier Stokes equations in the keyhole).

The heat lost by conduction is then calculated using the derivative of the Davis-Noller solution

$$Q_v=2\pi\lambda cyl(r/2l)$$

$$where \quad cyl(x)=\frac{I_o(x)}{K_o(x)}+2\sum_{n=0}^{\infty}(-1)^n\frac{I_n(x)}{K_n(x)} \qquad 3.11$$

where $l=\alpha/U$, and I_n and K_n are modified Bessel functions of the first and second kind respectively and of n^{th} order. This gives us an energy balance at the keyhole wall,

$$P(r,z)A(\theta)\frac{2}{r_s}\frac{dr_s}{dz}+\sqrt{LP_0}=2\pi r_s\rho_g u_g\Delta L+2\pi\lambda(T_v-T_0)cyl(\frac{r_s}{2l}) \qquad 3.12$$

This is a first order partial differential equation in r and z. In order to solve this, we note that the

above equation can be expressed as,

$$P(r,z)A(\theta)\frac{2}{r_s}\delta r_s+\frac{\delta z}{2}\sqrt{LP_0}-\pi\delta z r_s\rho_g\mu_g\Delta L-\pi\delta z\lambda(T_v-T_0)cyl(\frac{r_s}{2l})=0 \qquad 3.13$$

i.e $F(r_s)=0$. Here the differential has been replaced by small increments δr_s and δz. This can be solved in the region from z=0 to z=d, by dividing the depth by a specific number of points in order to obtain a value at each step for δz, and then finding a value of r and δr which satisfies the above equation. This was performed using a Newton-Raphson root finder method.

This energy balance is only valid when the keyhole radius is smaller than the radius of the beam, if the radius is bigger, then the beam does not strike the workpiece directly, and the only heat input into the workpiece is the heat conducted to the keyhole walls from the plasma. Thus the energy balance becomes,

$$\sqrt{LP_o}=2\pi r\rho_g\mu_g\Delta L+2\pi\lambda(T_v-T_o)cyl(\frac{r}{2l}) \qquad 3.14$$

In order to find the keyhole radius in this situation, a value of r is sought that satisfies this equation. This can easily be accomplished again by using the Newton-Raphson root finder method.

3.2.2 Modelling the flow in the keyhole

In order to model the flow inside the keyhole, a cylindrical coordinate system is used, with the axis of the keyhole aligned along the z axis. The radial and axial components of velocity are denoted by u and w respectively, the pressure denoted by p, the surface tension at the keyhole walls γ, the viscosity μ and the density ρ. Figure 3.2 illustrates the keyhole and the boundaries used in the model. Within the vapour, the continuity equation and the Navier Stokes equations are solved, these are

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho ur)+\frac{\partial}{\partial z}(\rho w)=0$$

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho uur)+\frac{\partial}{\partial z}(\rho uw)=\frac{1}{r}\frac{\partial}{\partial r}\mu r\frac{\partial u}{\partial r}+\frac{\partial}{\partial z}\mu\frac{\partial u}{\partial z} \qquad 3.15$$

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho uwr)+\frac{\partial}{\partial z}(\rho ww)=\frac{1}{r}\frac{\partial}{\partial r}\mu r\frac{\partial w}{\partial r}+\frac{\partial}{\partial z}\mu\frac{\partial w}{\partial z}$$

subject to the boundary conditions

$$p=1\ atm, \quad \frac{\partial u}{\partial z}=\frac{\partial w}{\partial z}=0, \quad on\ \ AB,\ HG$$
$$p=1\ atm \quad \frac{\partial u}{\partial r}=\frac{\partial w}{\partial r}=0 \quad on\ \ BC,\ FG \qquad 3.16$$

At the top and bottom surface of the weldpool (DC and EF respectively in the figure 3.2), no slip conditions apply, and so u=w=0, and at the centreline, as r tends to zero,

$$\frac{\partial w}{\partial r}\to 0, \quad u\to 0, \quad \frac{\partial p}{\partial r}\to 0, \quad as\ \ r\to 0 \qquad 3.17$$

The pressure balance at the keyhole wall is described by,

$$p_{\gamma}=\delta p_g+p_{abl} \qquad 3.18$$

where δp_g is the excess pressure in the keyhole due to the vapour flow. Here the pressures due to hydrostatic and hydrodynamic forces in the weldpool which tend to close the keyhole are neglected due to the reasons mentioned before in chapter 1. The pressure due to surface tension is equal to

$$p_{\gamma}=\frac{\gamma}{r} \qquad 3.19$$

and the ablation pressure is equal to

$$p_{abl}=\rho_g u_g^2 \qquad 3.20$$

The value of δp_g is evaluated from solving the continuity and Navier Stokes equations in the keyhole. Substituting these expressions back into the pressure balance, one obtains an expression for the radial velocity of the gas at the keyhole wall.

$$u_g=\sqrt{\frac{1}{\rho}\left(\frac{\gamma}{r}-\delta p_g\right)} \qquad 3.21$$

This gives the velocity at the keyhole wall as a function of the excess gas pressure and the surface tension pressure.

3.2.3 The weld width

The temperature in the workpiece can be approximated by neglecting fluid flow within the weld pool, and assuming the heat input into the weld metal can be described by a non uniform distributed point or a point and line source. In both cases the results of Rosenthal (1946) can be applied. The temperature in the workpiece due to a distributed line source can be found by assuming

that at discrete levels down the workpiece, a line source with a certain strength governs the heat input into the system, thus, using Rosenthal's equation for a line source for any particular depth z,

$$T_m = T_o + Q_l e^{\frac{Ux}{2\alpha}} K_o\left(\frac{Ur}{2\alpha}\right) \qquad 3.22$$

$$\textit{where} \qquad r^2 = x^2 + y^2$$

Here $Q_l(z)$ is the line source strength which represents the power absorbed per unit depth used in conducting heat throughout the material, at depth z down the workpiece. This is calculated from the energy balance equation.

$$Q_l = Q_{IB} + Q_F - Q_v \qquad 3.23$$

Hence, this solution models the heat input into the workpiece as a finite number of line sources down the keyhole, and the temperature can then be found at depth z, in terms of the respective line source at that depth. Due to the two dimensional nature of the line source solution, this produces a discontinuous temperature distribution, however it serves as a good indication as to the location of the melting isotherm. A more comprehensive solution which takes into account the 3 dimensional nature of the problem can be found by modelling the heat input into the material as a distribution of point sources down the keyhole, this would yield a temperature distribution described by,

$$T(x,y,z) = T_o + \int_{z'=0}^{l} q(z')f(z')dz' \qquad 3.24$$

where f(z') is defined from Rosenthal's point source solution,

$$f(c) = \frac{\exp\left(\frac{U}{2\alpha}x - \frac{U}{2\alpha}\sqrt{x^2+y^2+(z-c)^2}\right)}{4\pi k\sqrt{x^2+y^2+(z-c)^2}} \qquad 3.25$$

This solution can be enhanced by ensuring that there is no heat flow across the boundaries, this can be achieved by using the method of images, illustrated in figure 3.3, which ensures no heat flow across the top boundary in a semi infinite solid by assuming that the temperature at a particular point within the workpiece is also influenced by an exact image of point source distribution in the negative x direction. Thus the temperature distribution for a semi infinite solid becomes,

$$T(x,y,z) = T_o + \int_{z'=-l}^{l} q(z')f(z')dz' \qquad 3.26$$

where q(-l)=q(l). This method can then be extended to take into account the zero heat flow boundary conditions at the lower surface by again making an image of the weld, this time about the line z=d.

Then to ensure these further image sources have no heat flow conditions at the outer boundaries, two more images are reflected about the line z=d, and z=-2d, and so on. This procedure is repeated indefinitely and therefore the temperature distribution in the weldpool can be expressed in terms of an infinite series,

$$T(x,y,z) = T_o + \sum_{n=-\infty}^{\infty} \int_{z'=-l}^{l} q(z')f(2nd - z')dz' \qquad 3.27$$

The higher order terms have a decreasing affect on the temperature in the workpiece, and thus the solution converges. Table 3.1 gives the calculated temperature distribution for the line source solution above, together with the point source solution with increasing number of terms. As can be seen, after the fourth term in the series, there is no discernible effect on the weld width, and so in all further calculations, the weld width will be calculated from this approximation.

3.2.4 Model assumptions

The model makes various assumptions concerning the physical system under investigation in order to arrive at a system of equations that can be solved in a reasonable time on a mainframe computer. The main assumptions of the model are the fact that it disregards the action of the gas jet, together with the spatial variation in density and viscosity within the vaporised metal, and that the flow is assumed to be laminar. In addition, the model assumes that the keyhole and the flow within it can be described in terms of a 2-dimensional cylindrical coordinate system, and that the keyhole radius at the top is equal to the beam radius. The heat flow in the weldpool is assumed to be passed by conduction only, and so convection in the molten material is disregarded. These basic assumptions, together with their expected effects on the overall process, will be discussed individually below.

3.2.4.1 The effect of the gas jet

The model assumes that the metal vapour exits the top of the keyhole unhindered by the gas jet. In actual fact, the metal vapour is blown away by the gas shroud, which form a plume above the keyhole. This plume consists of ionised metal vapour and inert gas, which absorbs the laser light, and thus forms a heat source which is responsible for the 'nailhead' part of the solidified weld structure. Thus the effect of the gas jet does make an overall difference to the keyhole and weld dimensions. The effects of varying this point source upon laser the welding process will be discussed in the next chapter. However, in the absence of a mathematical description of the conditions within the plume, which is a separate study in itself, the gas jet will be assumed to have negligible effect on the process.

3.2.4.2 The effect of varying the density

The model assumes that the density is constant everywhere, and therefore the flow is

incompressible. In actual fact, the density changes with temperature according to the ideal gas equation,

$$\rho = \frac{p}{RT} \qquad 3.28$$

for one mole of gas, where p, the pressure, can be calculated from the degree of ionisation,

$$p = (n_e + n_i + n_n)kT \qquad 3.29$$

where n is the number of electrons, ions or neutrals respectively, thus, assuming that the pressure remains of the order of one atmosphere, the density varies with the ionisation of the vapour. At low degrees of ionisation, this effect will be negligible, however, at higher degrees of ionisation, the density will decrease. As the degree of ionisation is mainly dependent upon temperature, which is spatially dependent, the resulting density will vary across the solution zone, and in order to take this fully into account, the compressible Navier Stokes equation would have to be solved in the solution domain.

3.2.4.3 The effect of the fluid flow in the weldpool

The model assumes that the heat input into the weld can be described as a point source distribution, which is conducted throughout the depth of the material. No account therefore is taken of the effect of the fluid flow in the molten region. There are two main mechanisms that induce fluid flow in the CW laser welding process, the first is the convection of the flow due to the movement of the workpiece. The second is the effect of thermocapillary flow at the keyhole walls and on the top and bottom surfaces due to surface tension gradients. At high speeds and with thin workpieces, the flow is mainly driven by the workpiece velocity, and although the weld length obtained from a distributed point source or line source solution varies from experimentally obtained values, there is general agreement on the maximum weld width using various analytical solutions. (Williams (1996)). At slower speeds, surface tension forces have a greater affect upon the weld width. The effect of this parameter would be to drive the fluid outwards along the top and bottom surfaces, which would lead to an increase in weld width in these areas.

3.2.4.4 The assumption of 2 dimensional geometry

The keyhole, and the flow within it, are described in terms of a 2~dimensional axisymetric polar coordinate system, and hence the keyhole is assumed to be either cylindrical or conical. This assumption seems to be valid for low and medium welding speeds, however Kaplan (1994) showed that at higher welding speeds, the slopes of the front and back of the keyhole have different values of gradient, the latter being steeper than the former. For this reason, the model will concentrate mainly on the slow to medium range of welding speeds.

3.2.4.5 The assumption of the size of the top keyhole radius

In the present model the top keyhole radius is assumed to be equal to the theoretical beam radius. This assumption is dubious at high welding speeds as there is not enough laser-material interaction time to vaporise all the material underneath the laser beam. At slower welding speeds, the plasma in the plume absorbs more laser light. The spatial variation of refractive index due to the steep temperature gradients in the plume, result in a defocusing of the beam above the workpiece. Both these factors will tend to increase the value of the top keyhole radius.

3.2.5 Solution Method

In order to solve the above system of equations simultaneously, the initial condition is set that the top of the keyhole is equal to the beam radius. Then, assuming that the ablation term in the energy balance is zero, the keyhole coordinates are calculated down the depth of the material using the energy balance equation. A relaxed velocity is then prescribed at the keyhole wall using equation 3.21, and then the equations of fluid flow are solved in the region of interest. The pressure balance is again applied at the keyhole wall in order to set a new relaxed velocity, and the Navier Stokes equations are again solved, until convergence is reached for that particular keyhole shape. Once convergence is reached, the energy balance is reapplied in order to gain a new set of keyhole dimensions, and the process is repeated until overall convergence. A modified versions of Rosenthal's equation for a distributed point source is then solved by the Newton-Raphson technique in order to obtain the weldpool dimensions. This solution procedure is illustrated schematically in the flow chart in figure 3.4.

3.3 Results and Discussion

3.3.1 Description of the Fluid Flow within the system

The pressure at the keyhole walls is equal to γ/r above the ambient pressure, whereas at the outer boundaries of the solution zone the pressure is set to ambient. This pressure difference, together with the mass influx at the keyhole walls of ablating material, induces a flow away from the keyhole walls and the vapour accelerates out of the keyhole at both ends, and into the ambient atmosphere. The main region of interest here is the keyhole, especially the pressure and velocity at the keyhole walls which directly influences the amount of heat used in vaporising the material. Also of interest are the values of the velocities of the vapour exiting the top and bottom of the keyhole. These have been measured experimentally by Beck et al (1989) to be of the order of 100-200m/s. This should ensure some validation of the model.

3.3.2 Examination of the variation in Parameters and Material Properties that affect the Vapour Flow in the Keyhole

In the following sections, the effect of varying a number of parameters upon the keyhole and

weldpool dimensions, the energy contributions to the overall process, and the flow characteristics in the keyhole will be studied. The aim of this study is to gain a quantitative and qualitative understanding of the factors that affect the vaporisation term in the heat balance.

3.3.2.1 Change of process velocity

Figures 3.5 and 3.6 show the response of the system to a change in process velocity. As the velocity decreases from 40mm/s down to 10mm/s, the gradient of the keyhole becomes steeper, and hence less of the workpiece is exposed to the laser beam, with a corresponding decrease in Fresnel absorption (figure 3.6 c,d). The velocity of ablating vapour only increases slightly with increasing process velocity, and hence taking into account the increased area for ablation to occur, conservation of mass dictates that the vapour velocity at the top and bottom keyhole mouths will increase. Indeed this is the case, especially at the top of the keyhole (see figure 3.5 c,d). It is interesting to note that the power used in ablating the material from the keyhole walls is less than 5% of the absorbed power throughout the range of speeds considered here. The pressure difference between the vapour in the keyhole and the keyhole wall is small inside the keyhole, resulting in relatively low ablation velocities and energies. However near the keyhole mouths, the pressure becomes more influenced by the ambient pressure, and so the pressure difference increases with a corresponding increase in the velocity of the ablating vapour.

3.3.2.2 Change of beam radius

Figures 3.7 and 3.8 show comparative results obtained from varying the beam radius. An increase in beam radius will result in a decrease in power density at the workpiece, and hence increase in the slope of the keyhole, as less power is required per unit depth to keep the keyhole open (figure 3.8 c,d). The velocity of ablating material is greater in the narrower beam radius case (figure 3.8 a), however with the decrease in the value of r, this makes a negligible difference to the heat used up in ablation (figure 3.8 b). The top and bottom exit velocities are greater in the narrower beam radius case due to the increased pressure difference (figure 3.7 c,d).

3.3.2.3 Change of thickness

An increase in thickness of the workpiece will increase the surface area of the keyhole, which will result in a greater amount of mass being ablated from keyhole wall, and hence, by conservation of mass, the velocity at the keyhole exits will increase (figure 3.9 c,d). Obviously, the flow within the keyhole influences the pressure difference at the keyhole wall. As can be seen, between thicknesses of 2 and 6mm, the greater the velocity exiting the keyhole, the greater the pressure difference and hence ablating velocity at the keyhole wall (figure 3.10 a). It is interesting to note that even in the 6mm thickness case, the power used up in ablating the material is only about 10% of the total energy absorbed by the workpiece (figure 3.10 b,c).

3.3.2.4 Change of density

The vapour density effects the fluid flow in the keyhole, the ablating material velocity at the keyhole walls, and the energy used up in ablation. As a result of these changes in the heat and pressure balance equations, there is a slight decrease in keyhole coordinates with an increase in vapour density from 0.1122kgm^{-3} to 0.4480kgm^{-3} (figure 3.11 a), reflecting the relatively minor roll of the ablation energy in the energy balance. An increase in vapour density decreases the top and bottom exit velocities. The wall velocity of ablating vapour is dependent upon the product of 1 over the square root of the vapour density and the pressure difference between the keyhole wall and the fluid within keyhole. The former seems to be the stronger term, though only slightly, and the wall velocity increases with decreasing density (figure 3.12 a). However, as the power used in ablating the material is proportional to the density, one finds almost a proportional increase in the ablation contribution with increasing density. Even in the most extreme case, though, the ablation contribution is only 10% of the total power absorbed in the process.

3.3.2.5 Change of latent heat

An increase in the latent heat results in an increase in the power required to ablate the material (figure 3.14 b), which in turn will result in a slight decrease in the keyhole radius (figure 3.13 a), reflecting the greater power required to maintain the keyhole. Again it should be noted that even in the more extreme cases, the average ablation contribution to the energy balance is approximately 10% of the overall energy input.

3 3.2.6 Change of viscosity

From ideal gas theory it can be shown that the viscosity is related to the temperature by the equation

$$\mu = \frac{1}{3Q}\sqrt{\frac{4mkT}{\pi}} \qquad 3.30$$

where Q is the inter-particle collision cross section, m is the mass of a metal ion or atom, and k is Boltzmann's constant. In order to determine the effect that the change of viscosity has upon the process, the viscosities corresponding to keyhole temperature of 6000K, 12,000K 18,000K were chosen, due to the fact that both theoretical and experimental studies have estimated the temperature within the keyhole to be within this range. Using values for iron at these temperature, the corresponding viscosities are $7.07*10^{-4}$, $9.9*10^{-4}$ and $1.22*10^{-3}$ respectively. The results from these comparative numerical trials are shown in figures 3.15 and 3.16. As one would expect, the exit velocities at the top and bottom of the keyhole increase with decreasing viscosity (and hence decreasing resistance to fluid flow). This is significant as a number of researchers (eg Dowden et al (1987)) have neglected viscous terms in order to solve the flow in the keyhole analytically. The

present study however, shows that this assumption is not justified. In the range of values studied here, the viscosity does not have a significant effect on either the wall velocity or the energy balance at the keyhole wall.

3.3.3 Examination of the variation in Parameters and Material properties that affect the absorbed power

If one considers the graphs of absorbed power as a function of depth, it can be seen that the overall fraction of power used in ablating the material rarely rises above 10% of the total absorbed power, and has little effect on either the keyhole or weldpool dimensions. Due to its relative unimportance therefore, the ablation term will be dropped in all subsequent calculations, thus the keyhole coordinates are calculated by the energy balance alone. This will make the model similar to other models (eg Ducharme et al (1995)), however the current work extends the work of previous authors in a number of areas. In the following sections, the parameters that affect the contributions to the inverse Bremsstrahlung and Fresnel absorption mechanisms will be discussed, in order to obtain an qualitative understanding of system under consideration.

3.3.3.1 Examination of change in plasma absorption

The significance of the inverse Bremsstrahlung contribution to the absorbed power in the keyhole is still a matter of debate. This is not helped by experimental observations, in which the temperature in the keyhole has been measured between 6kK, where inverse Bremsstrahlung absorption will be insignificant, and 18kK, where the plasma absorption would become dominant in many cases of CO_2 laser welding. This should be borne in mind when contemplating the present study which varies the linking intensity calculated from Dowden's formula (Dowden et al (1987)). Figure 3.17 shows the effect of varying the linking intensity upon the keyhole radius, the weld width, and the power absorbed by both inverse Bremsstrahlung and Fresnel absorption mechanisms. The inverse Bremsstrahlung term provides a large part of the energy required to keep the keyhole open, and so as this contribution increases, the slope of the keyhole radius becomes steeper, which exposes less of the workpiece to the beam, with the result that the Fresnel contribution decreases.

3.3.3.2 Examination of the effects of Fresnel absorption

The Fresnel contribution is given by equation 3.6, this is a function of the angle of incidence, the power, the gradient of the keyhole walls and the absorption coefficient, which in turn is dependent upon the angle of incidence and a material and wavelength dependent property ε. In the following sections, the effect of varying these contributions will be discussed.

The inverse Bremsstrahlung contribution gives a theoretical limit to the value of the keyhole radius, and also ensures a relatively steep keyhole gradient. This will make comparisons in the

variation of the Fresnel absorption parameters difficult to discern, and so, in order to accentuate the differences between results, the inverse Bremsstrahlung term will be dropped. The heat balance at the keyhole wall then becomes,

$$P(r,z)A(\theta)\frac{2}{r_s}\frac{dr_s}{dz}=2\pi k(T_v-T_o)cyl(\frac{r}{2l}) \qquad 3.31$$

this will be the governing equation for the following sections.

3.3.3.2.1 Investigation of the change in ε for a given set of process parameters

Figure 3.18 shows the variation of absorption coefficient with angle of incidence for a number of values of ε. One should bear in mind here that the values of ε for mild steel when using CO_2, CO and Nd:YAG lasers are approximately 0.08, 0.12 and 0.24 respectively, and that in general, over the range of values considered here, ε increases with decreasing wavelength. From the graph, it can be see that an increase in ε will result in an increase in the absorption at angles of incidence upto and slightly beyond the Brewster angle. This is significant due to the fact that as the wavelength decreases, there is a corresponding decrease in the amount of power required to initiate the keyholing process.

Figure 3.19 a-b) shows the effect that the variation of ε has upon the keyhole radius for the welding of 2mm mild steel with a power of 2.0kW, a spotsize of 0.2mm, and for process speeds of 20 and 50mm/s. It can be seen that for all values of ε in the 20mm/s case, and for all but ε=0.04 in the 50mm/s case, the value of the keyhole radius at any depth is greater for lower values of ε. However, in the case of ε=0.04 at 50mm/s, the slope of the keyhole is more non linear. As can be seen at the top of the keyhole, the slope of the keyhole radius is less steep than the other cases considered, however, as r decreases the slope becomes steeper and steeper, and at the penetration depth, the value of the keyhole radius is larger than those cases where ε is greater than 0.20. This behaviour is reflected also in the values of the weld widths and the absorbed power down the keyhole (figure 3.19 c-f). This phenomena can be explained by examining the radial coordinate dependent terms in the energy balance at the keyhole wall. First, note that equation 3.31 can also be expressed as.

$$P(r,z)A(\theta)\delta r_k=r_k cyl(\frac{r_k}{2l})(\delta z 2\pi(T_v-T_o)) \qquad 3.32$$

Note that a decrease in r will result in a corresponding decrease in the Bessel function expansion term cyl(r/2l), and therefore, if the heat balance is to be satisfied, the left hand side must decrease as well. As the power input to the discretised element is assumed constant, then the heat balance must be

satisfied by a decrease in the absorptivity and/or δr. In actual fact, the value of absorptivity in this region increases as the slope approaches the Brewster angle (figure 3.19 g-h), and so a sharp decrease in δr is required. All the other cases also exhibit a less severe form of this phenomena. As the angle of incidence is greater than the Brewster angle, and hence the absorptivity decreases as the slope increases, and a comparatively smaller decrease in δr is required.

3.3.3.2.2 Change of initial radius (beam radius)

A decrease in beam radius will increase the power density at the workpiece. Mathematically, this can again be studied by examining the heat balance equation. At the keyhole mouth, the amount of heat conducted away from the keyhole decreases with decreasing radial coordinate, and hence, slimmer keyholes require less energy to be kept open. This is illustrated in figure 3.20 c), where, as can be seen, the largest amount of power is used up in the case with the widest beam radius. This is intuitively obvious when one considers the respective keyhole profiles (figure 3.20 a). The wider the beam radius, the less steep the keyhole profile, and so the beam is exposed to a larger surface area, which will result in a greater amount of energy absorbed by the workpiece.

3.3.3.2.3 The change in the power distribution

Due to the condition of axial symmetry within the keyhole, only 3 beam modes can be considered. These are the top hat distribution, Gaussian distribution, and the TEM_{01*} mode (doughnut mode). In reality these modes correspond to a certain value of M^2, which will result in the Gaussian distribution producing the finest spot of the three, followed by the TEM_{01*} mode. Bearing this in mind, the results for welding mild steel at a power of 2.0kW, a speed of 20mm/s using the three distributions with both 0.2mm and 0.3mm beam radii, are shown in figure 3.21. As can be seen, there is little difference between the top hat mode and the TEM_{01*} mode for either of the beam radii. However, as the value of the beam intensity is low at the outside radius in the case of the gaussian beam,the slope is quite gradual. However, as the radius decreases, and the beam intensity increases, the slope of the keyhole wall becomes much steeper. The extra power at the centre of the beam alone would probably be enough to increase the penetration depth in the welding process. This fact, together with it's enhanced focusing capability, illustrate the advantages of the Gaussian mode.

3.3.3.2.4 Change in conductivity

The value of conductivity describes the ability of the material to transmit heat. A good conductor will transmit heat quickly throughout the region, with a low value of temperature gradient throughout the material; whereas a poor conductor will retain the heat in a small region, with a correspondingly high temperature gradient. If the energy input into the workpiece is the same, and assuming all other material properties are constant, then the higher the conductivity, the lower the surface temperature, as in this case the heat is spread throughout the material rather than concentrated

at the surface. Figure 3.22 shows the power required to maintain a keyhole in 1mm mils steel as a function of conductivity at different values of keyhole radii. As can be seen, the greater the conductivity, the greater the amount of energy is required to keep the keyhole open. This is illustrated in figures 3.23 a) and 3.23 c). Once this energy is reached however, the lower value of temperature gradient will ensure greater weld widths are obtained for cases with a high conductivity (figure 3.20 b).

3.4 Summary

A mathematical model of the laser keyhole welding process was developed that took account of the pressure and energy balance at the keyhole walls, together with the vapour flow within the keyhole. The heat balance consisted of the Fresnel absorption at the keyhole walls and the inverse Bremsstrahlung absorption in the plasma, for the heat input; and the vaporisation of the ablated material and the conduction of heat throughout the workpiece as the heat output. The pressure balance consisted of the pressure due to surface tension, the recoil pressure due to the material ablating off the keyhole walls, and the pressure due to the fluid flow in the keyhole. The model neglected both the defocusing and the heat source effects associated with the plasma above the workpiece in the plume, as well as the action of the gas shroud or the fluid flow within the weld itself. The model was run for various process parameters and thermophysical properties and it was concluded that,

- The velocity of the ablating material from the keyhole walls is highest at the top and bottom of the weld, where the pressure difference is at its highest value, and decreases towards the centre of the keyhole. Typical values for the cases studied here range between 30-40m/s at the mouth of the keyhole, and 5-10m/s in the middle.

- Due to the pressure difference between the keyhole walls and the outside ambient pressure the ablating material from the keyhole wall accelerates out of the top and the bottom mouths of the keyhole at velocities in the order of 10^2m/s. This is consistent with the experimentally measured values of Beck et al (1989).

- As the translation speed increases, the keyhole becomes more conical increasing the pressure at the keyhole walls and inside the keyhole, and hence the pressure gradient. This results in an increase in the velocities exiting the keyhole (especially at the bottom).

- An increase in material thickness between 2 and 6mm, results in an increased value of ablation velocity at the keyhole walls, by conservation of mass it is then intuitively obvious that the velocity at the mouth of the keyhole will therefore increase.

- A decrease in beam radius also increases the pressure at the keyhole walls, which will again lead to higher exit velocities. There is also a slight increase in the ablating velocity at the keyhole walls, however, the increased value of radial coordinate compensates somewhat and there is little difference in the energy lost due to the ablation process.

- A decrease in vapour density produces an increase in ablation velocity. However, this is outweighed by the proportional relationship between the power used up in ablation and the vapour density, and a decrease in density leads to a decrease in the power used up in ablation. Increasing density produces higher exit velocities at the keyhole mouths.

- A change of latent heat has no discernable effect on either the ablating or exiting velocities in the keyhole. However, the power used in ablation increases with increasing latent heat.

- Ideal gas theory states that the viscosity increases with the square root of the temperature. The exit velocities increase with increasing viscosity and hence temperature in the keyhole. This is very significant as a number of researchers have suggested that viscosity terms in the Navier-Stokes equations are negligible, the present analysis shows that this assumption is wrong.

The calculations of the model indicate that even in the extreme cases, the overall power used in ablating the material is rarely above 5% of the overall value of absorbed power. Therefore one can conclude that this term does not affect the laser welding process significantly, and so in order to develop a simpler model, this term was dropped. Thus the keyhole boundary was described by the energy balance alone. The model was then run to assess the relative importance of the variables in this equation, and it can be concluded that,

- An increase in the linking intensity L, will ensure that most of the power is absorbed in the plasma, thus leading to steeper keyhole wall profiles and a reduced value of power absorbed by the Fresnel mechanism as there is less surface area exposed to the beam.

- In general, the absorption increases with decreasing wavelength (and therefore increasing value of ε), resulting in higher values of keyhole wall gradient with decreasing ε. However, when the gradients are within the range of the Brewster angle, this is not always the case.

- A decrease in beam radius will decrease the amount of power required to keep the keyhole open, and will therefore lead to a steeper keyhole slope.

- There is little difference between the keyhole dimensions obtained with a top hat or TEM_{01*}

mode, which produced almost linear profiles, whereas the gradient of the keyhole profile obtained when using a Gaussian beam was low at the start, where the intensity is relatively small, but increased as the radial position decreases towards the centre, where the beam intensity is much greater.

- For a particular keyhole with set dimensions, an increase in conductivity will increase the amount of absorbed power required to keep the keyhole open, which will decrease the gradient of the keyhole slope, and increase the weld width.

Tables

Depth (mm)	Weldbead width (mm)				
	i) Line source (eqn 3.22)	ii) Point source (eqn 3.24)	iii)Point source (eqn 3.26)	iv)Point source (eqn 3.27)	v)Point source (eqn 3.27)
0.0	1.05	0.195	0.931	0.957	0.957
0.2	1.017	0.510	0.925	0.953	0.953
0.4	0.983	0.642	0.908	0.942	0.942
0.6	0.949	0.718	0.882	0.925	0.925
0.8	0.915	0.760	0.865	0.904	0.904
1.0	0.881	0.780	0.824	0.880	0.880
1.2	0.846	0.786	0.773	0.857	0.857
1.4	0.812	0.783	0.709	0.833	0.833
1.6	0.777	0.775	0.629	0.813	0.813
1.8	0.741	0.768	0.521	0.800	0.800
2.0	0.705	0.765	0.355	0.796	0.796

Table 3.1. The weldbead width as a function of depth calculated from i) the line source solution presented in eqn 3.22, ii) the distributed point source solution obtained from 3.24, with no extra terms, iii) the distributed point source solution calculated from 3.26 using only the first term in the method of images (thus assuming it is a semi infinite solid). iv) the distributed point source solution with two extra terms using the method of images (eqn 3.27) and v) the distributed point source solution using 4 extra terms. The parameters used for this study were a power of 1kW, a velocity of 20mm/s, a beam radius of 0.2mm, and a workpiece thickness of 2mm.

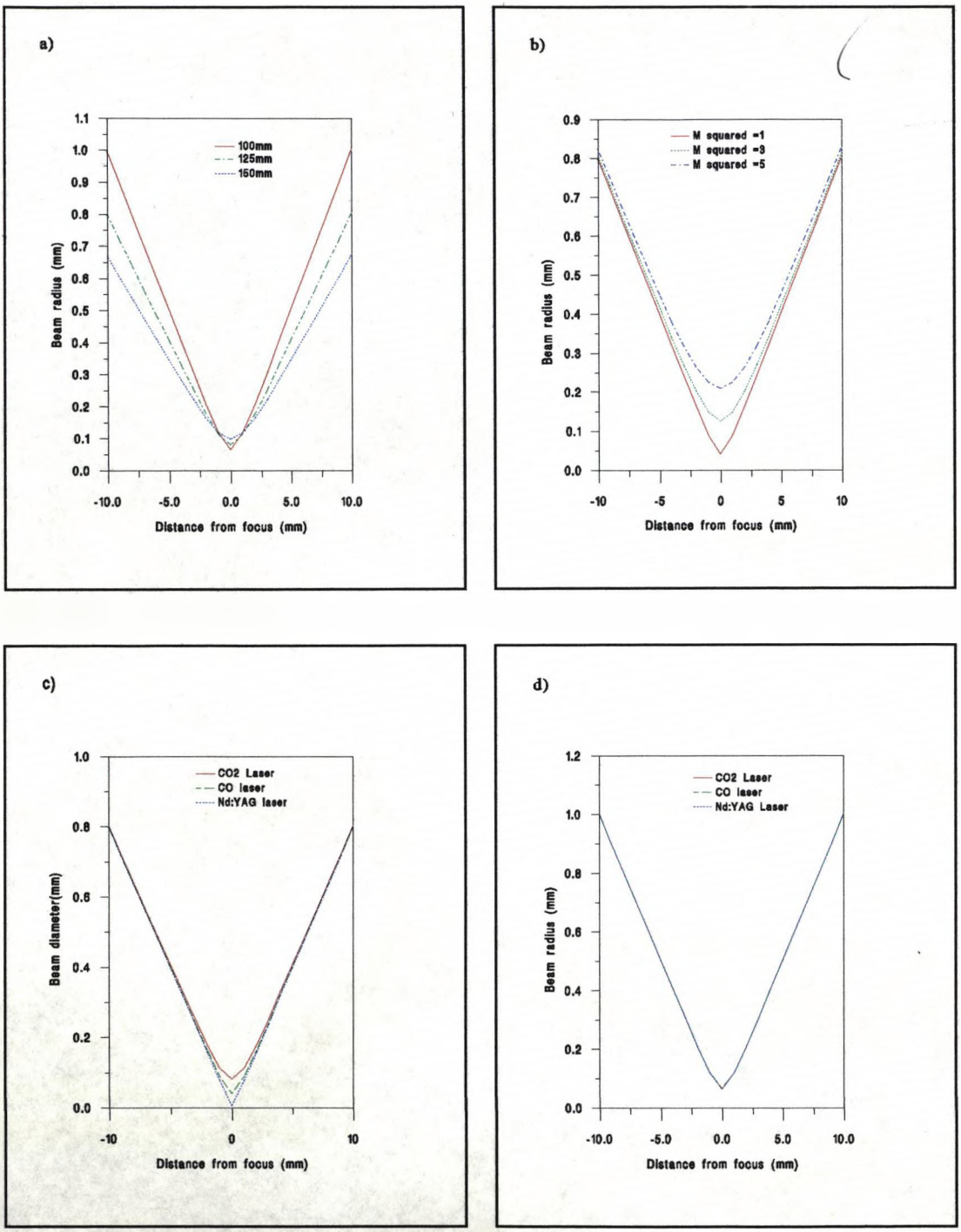


Fig.3.1 Variation of beam radius near the focus for a) different values of focal length, b)different values of M^2, c) different values of wavelength, with M^2 constant, and d) different values of wavelength with M^2 varying according to the particular value of Fresnel number for that wavelength.

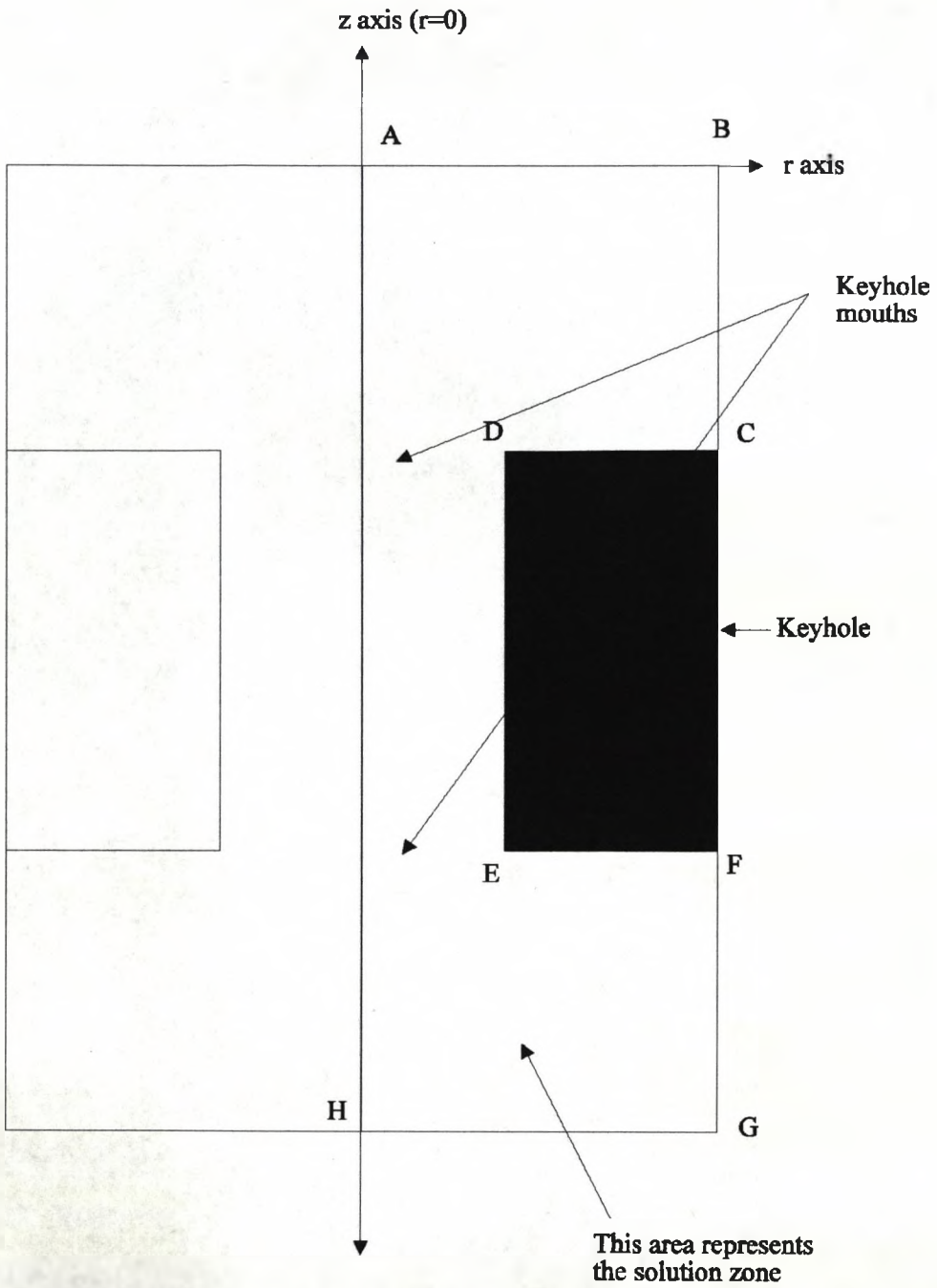


Fig.3.2 Schematic diagram of the solution zone for evaluating the dynamics of the vapour flow out of the keyhole.

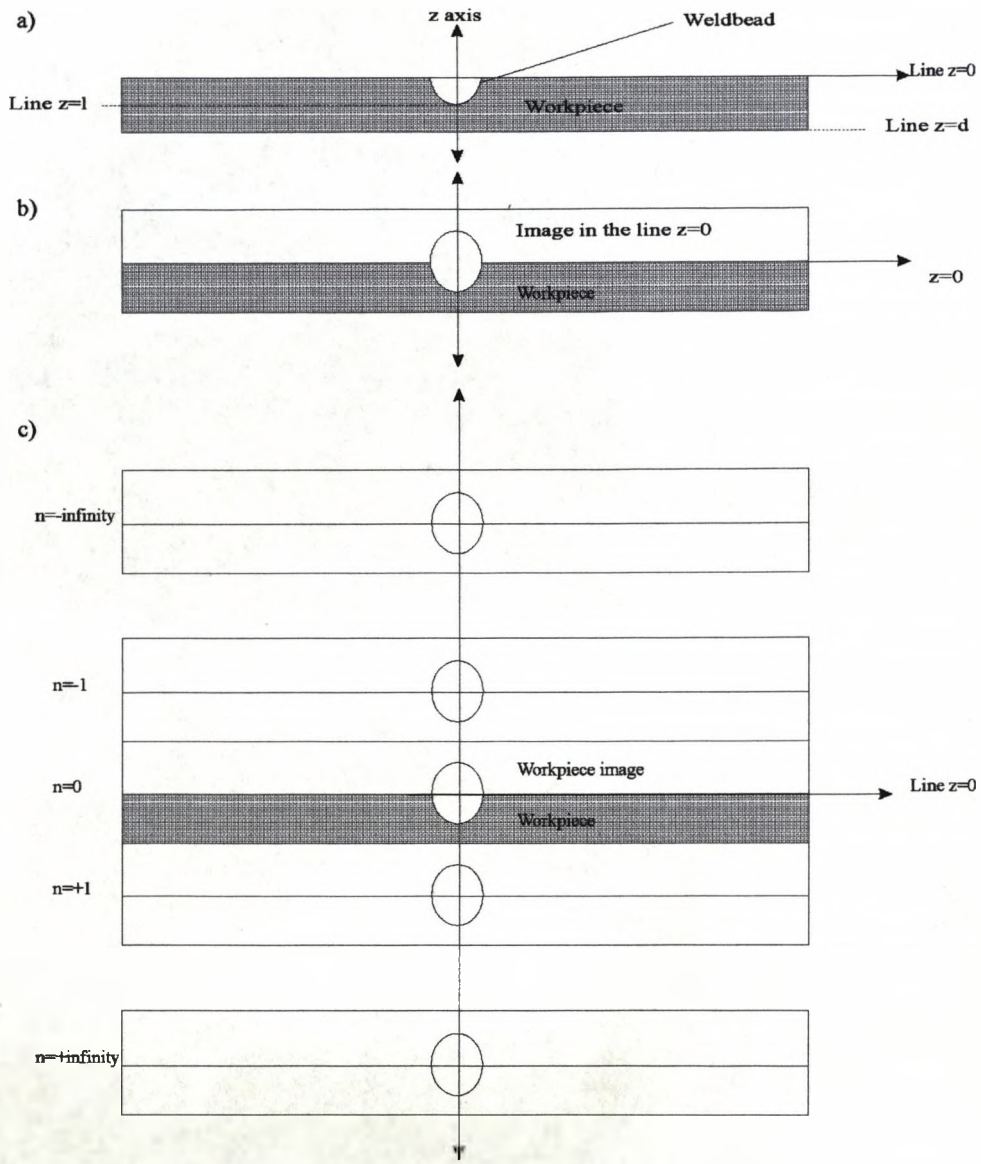


Fig.3.3 Schematic diagram illustrating the method of images.

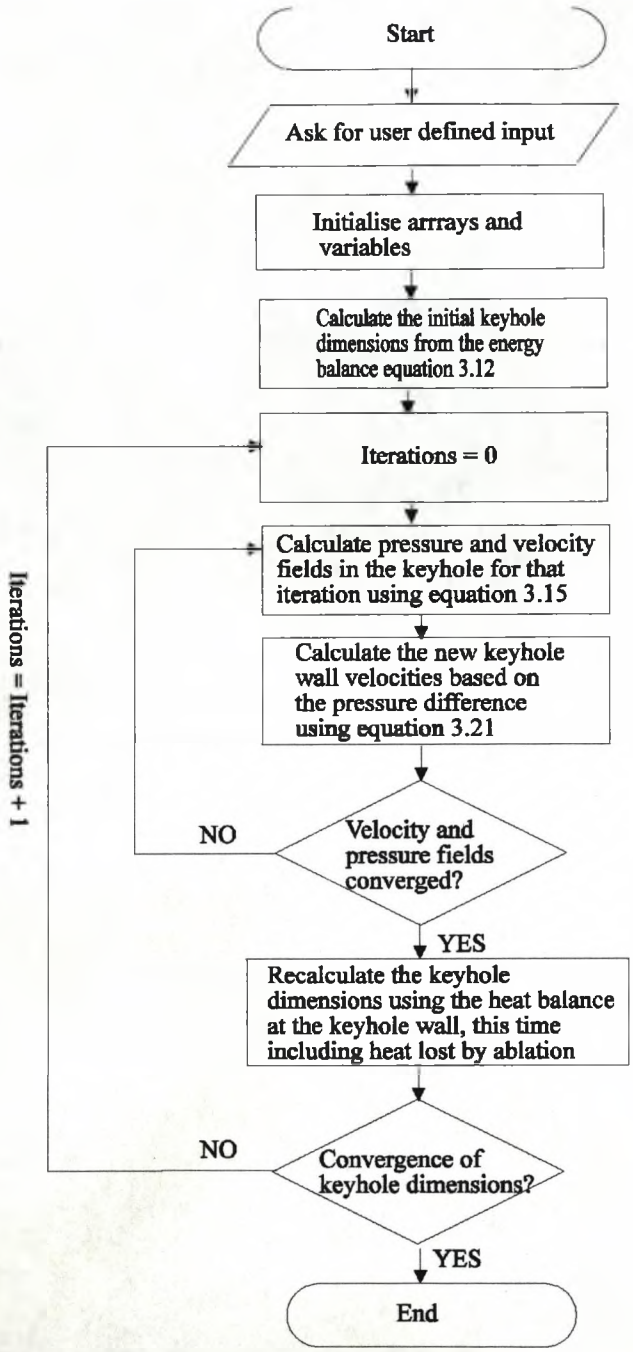


Fig.3.4 Flow chart illustrating the solution procedure for calculating the conditions during laser keyhole welding.

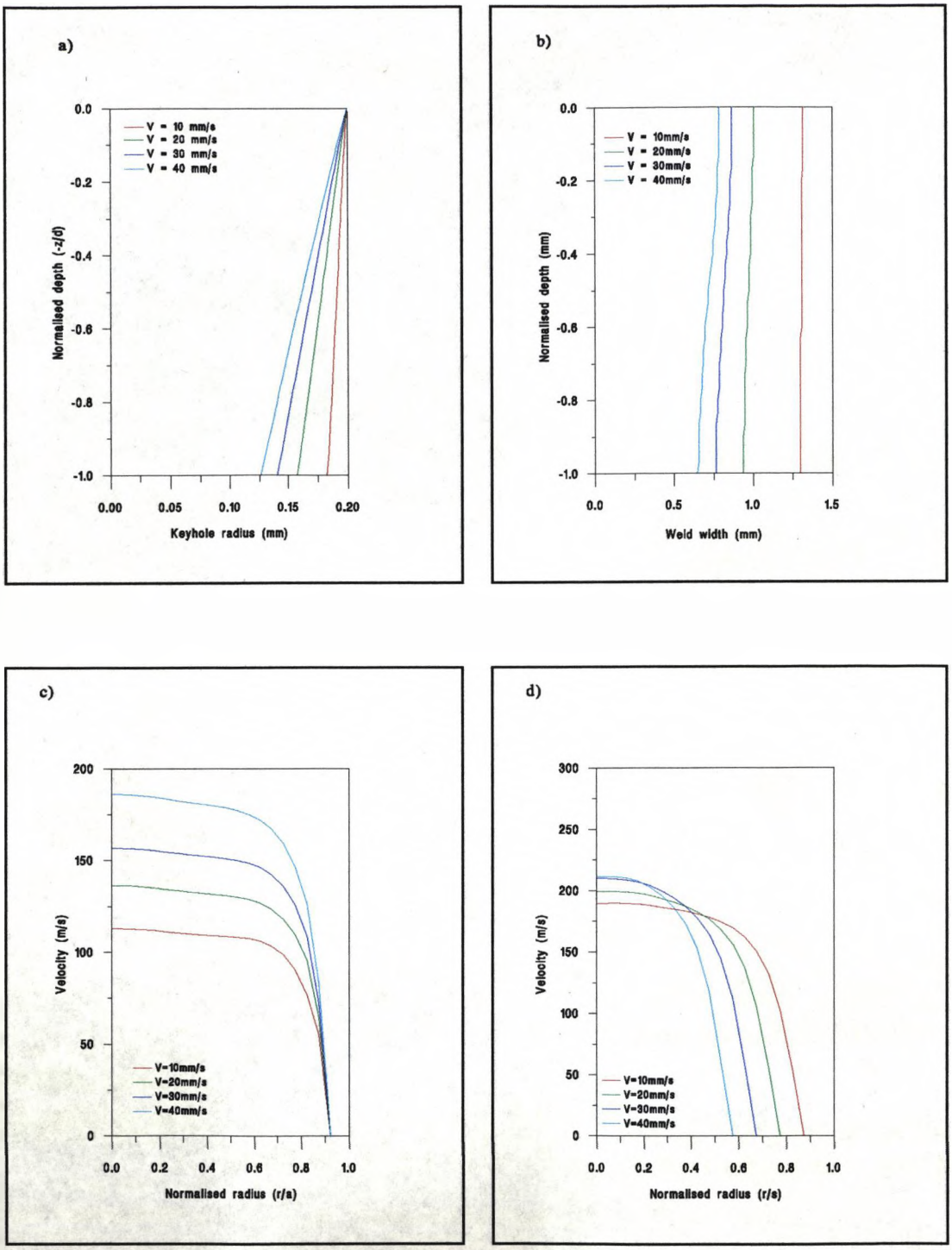
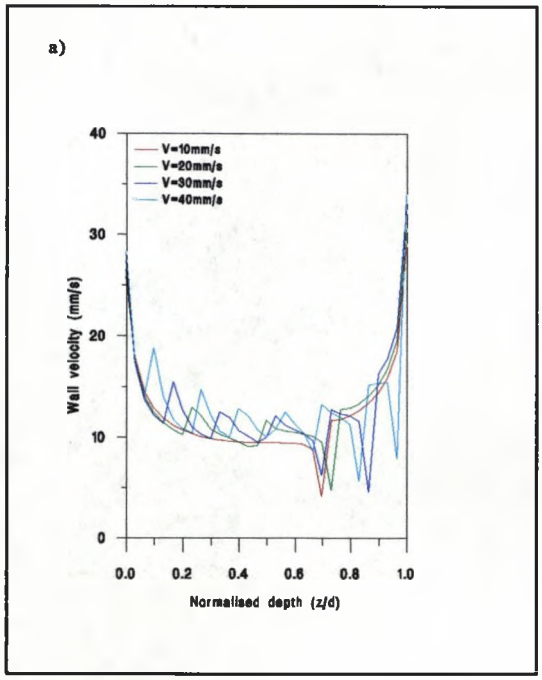
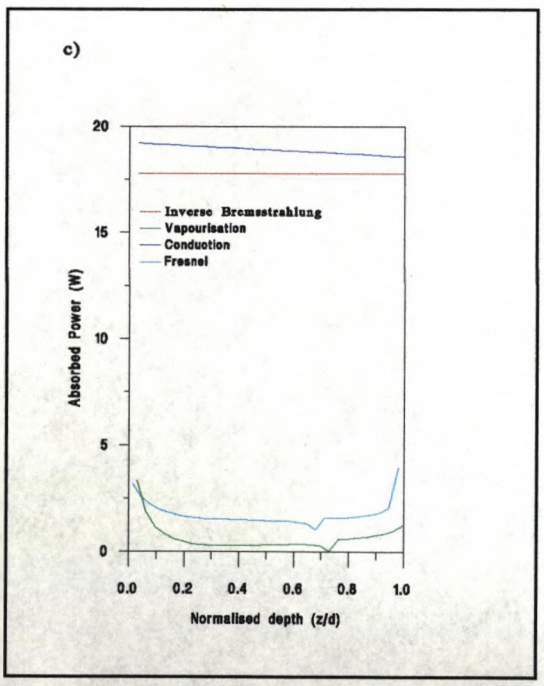


Fig.3.5 Model results from welding 2mm mild steel at 2.0kW of power, illustrating the effect of velocity upon a) the keyhole radius, b) the weld width, c-d) the velocity of vapour exiting the top and bottom of the keyhole respectively.





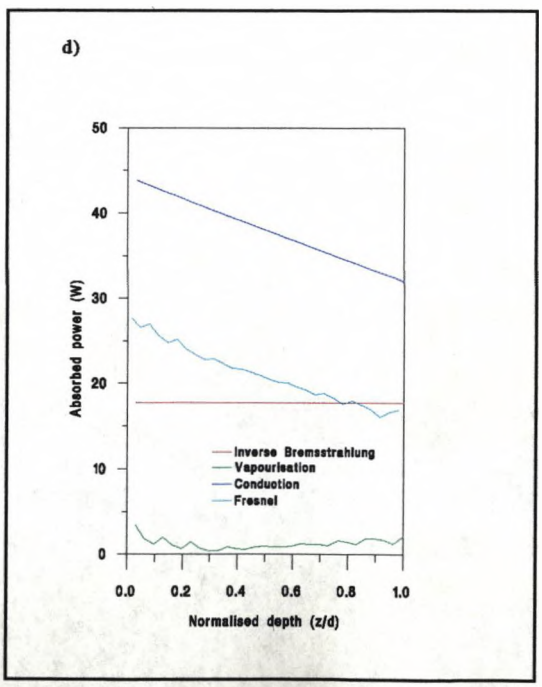


Fig.3.6 Model results from welding 2mm mild steel at 2.0kW of power, illustrating the effect of velocity upon a) the velocity of ablating material from the keyhole wall, b) the power used up in the ablation process, c-d) the contributions to the input and the output power for the cases of c) v=10mm/s, and d) v=40mm/s.

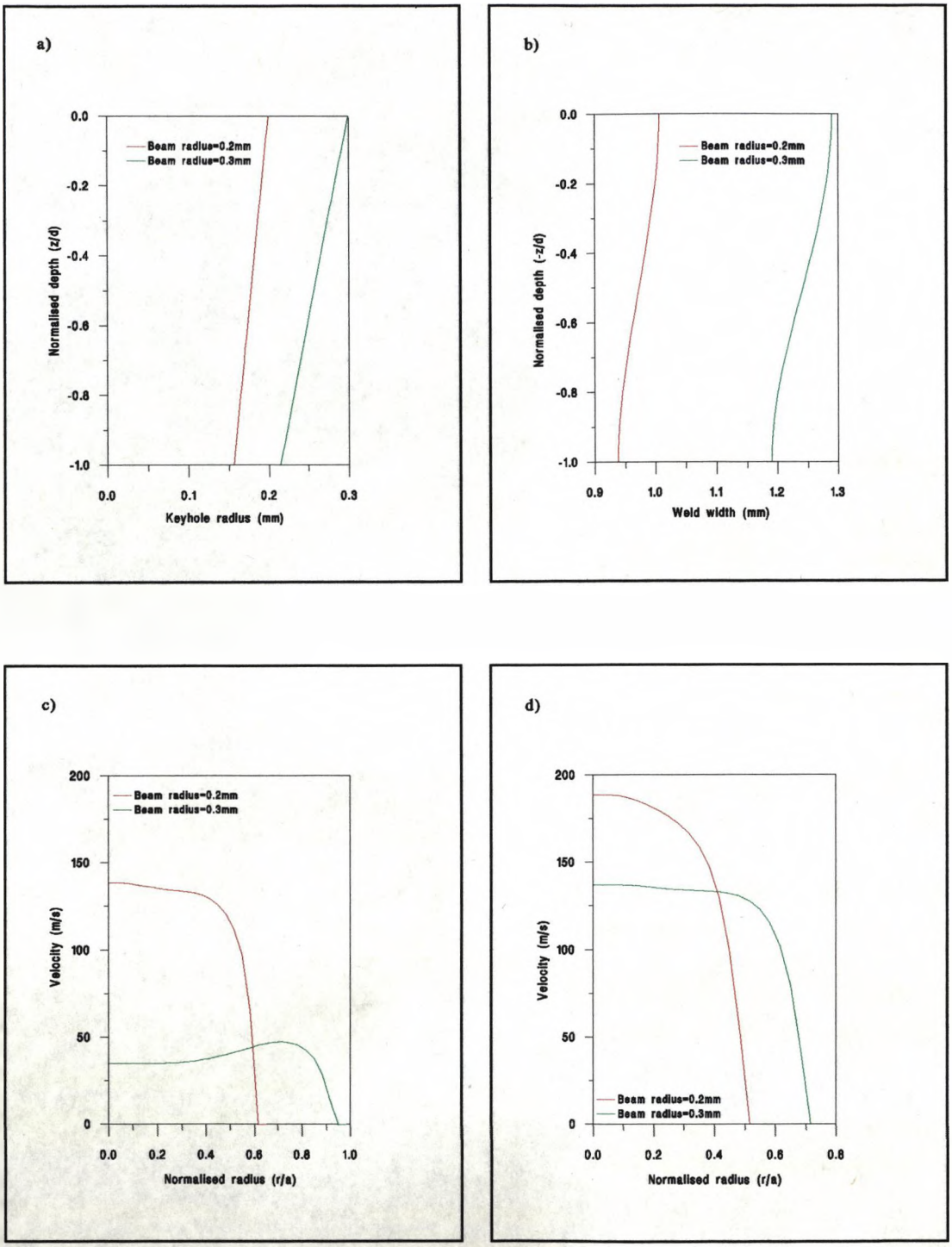


Fig.3.7 Model results for welding 2mm mild steel at 2.0kW of power, illustrating the effect of beam radius upon a) the keyhole radius, b) the weld width, c-d) the velocity of the vapour exiting the top and bottom of the keyhole respectively.

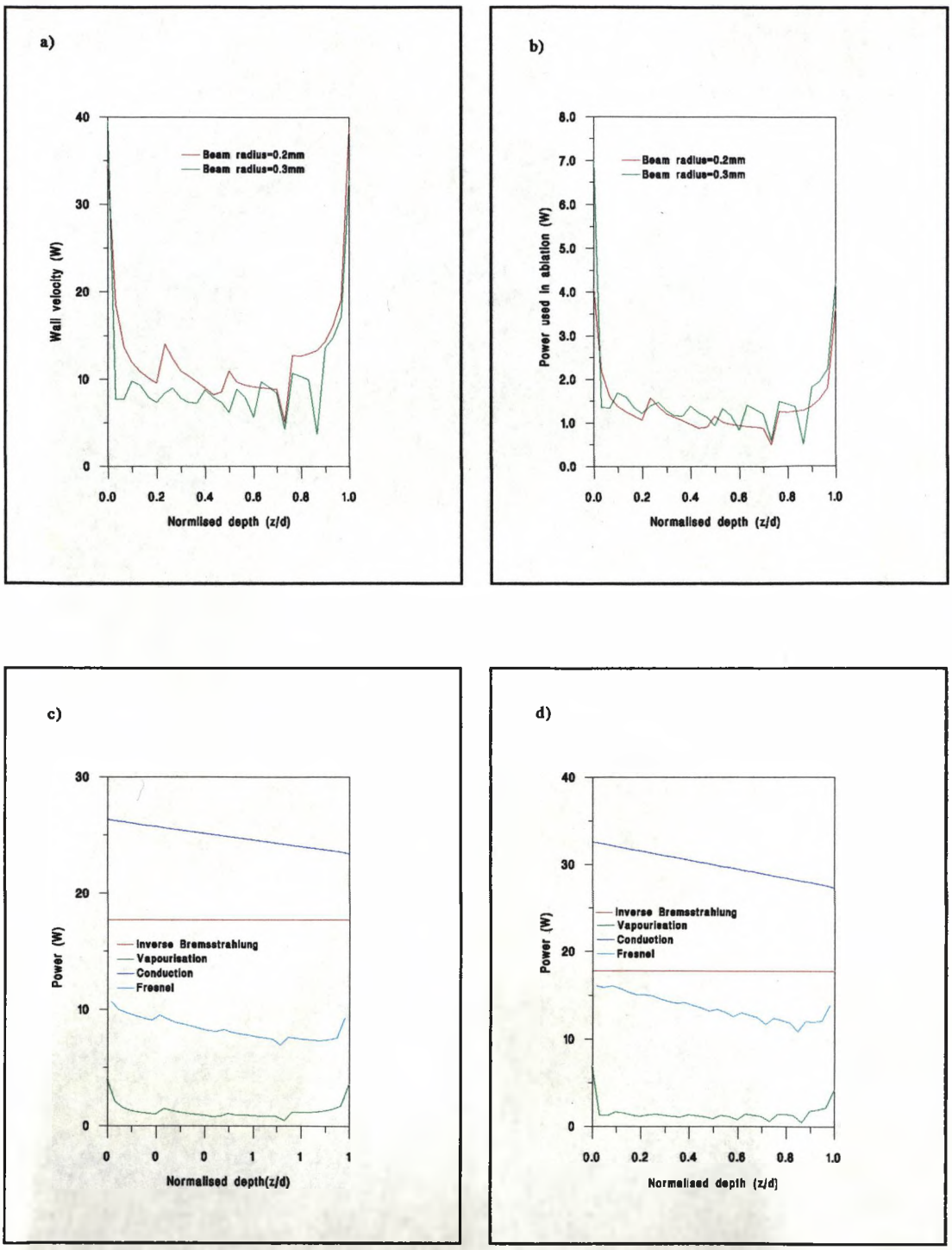


Fig.3.8 Model results from welding 2mm mild steel at 2.0kW of power, illustrating the effect of beam radius upon a) the velocity of ablating material from the keyhole wall, b) the power used up in the ablation process, c-d) the contributions to the input and the output power for the cases of c) r_b=0.2mm, and d) r_b=0.3mm.

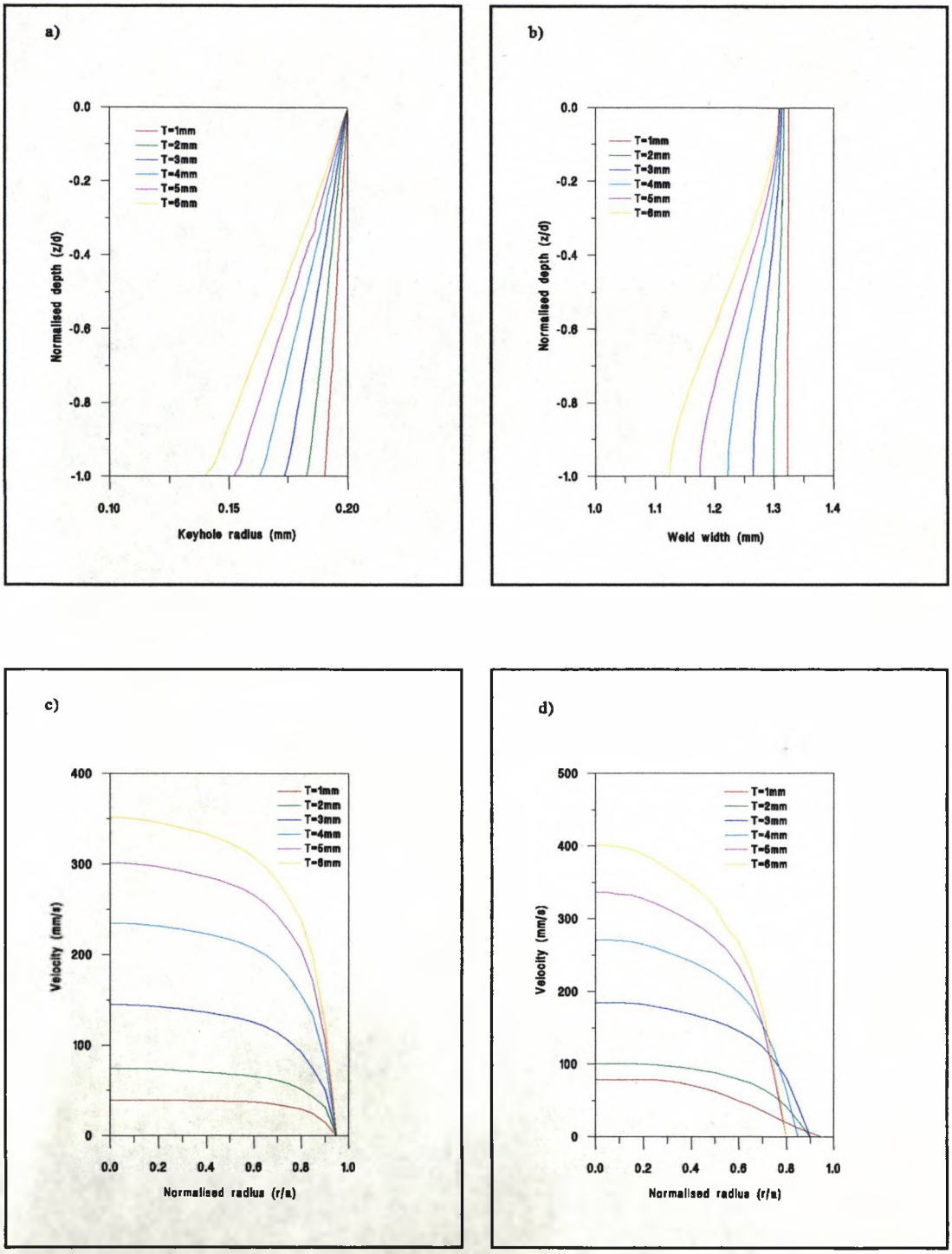
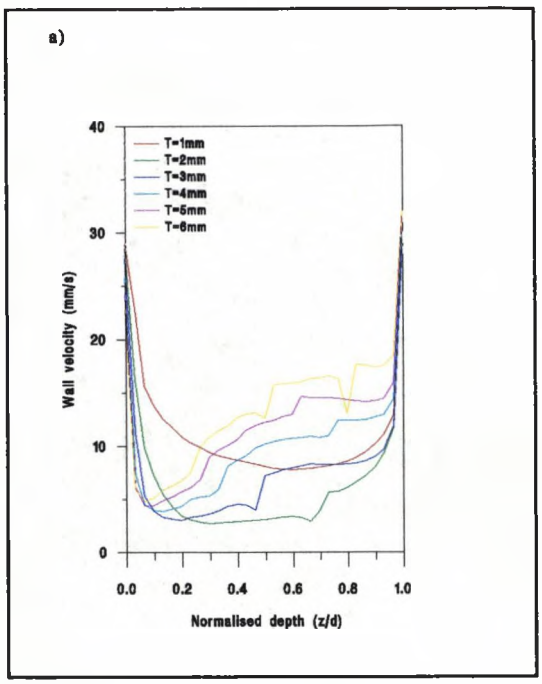
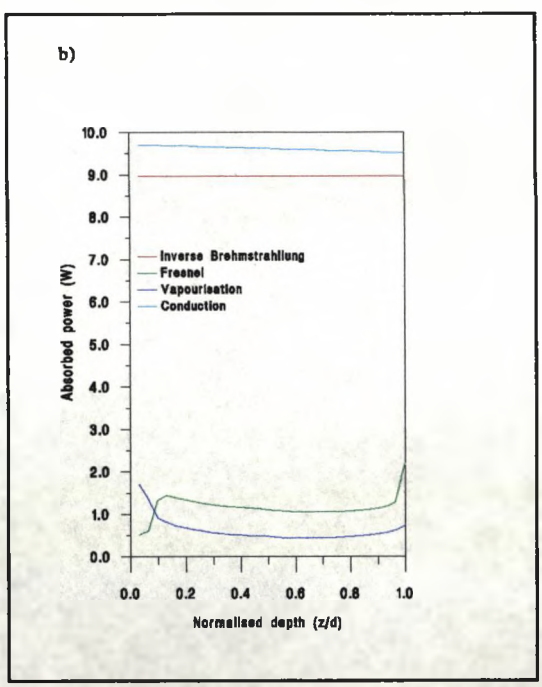


Fig.3.9 Model results for welding 2mm mild steel at 2.0kW of power, illustrating the effect of plate thickness upon a) the keyhole radius, b) the weld width, c-d) the velocity of the vapour exiting the top and bottom of the keyhole respectively.





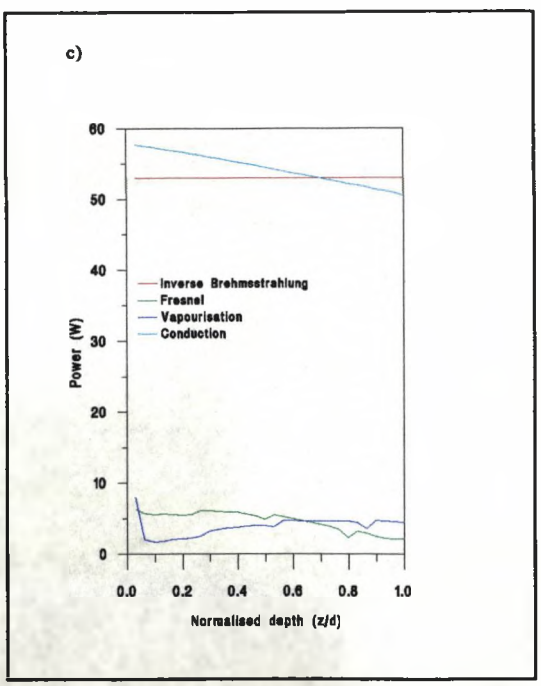


Fig.3.10 Model results from welding 2mm mild steel at 2.0kW of power, illustrating the effect of plate thickness upon a) the velocity of ablating material from the keyhole wall b-c) the contributions to the input and the output power for the cases of b) t=1mm, and c) t=6mm.

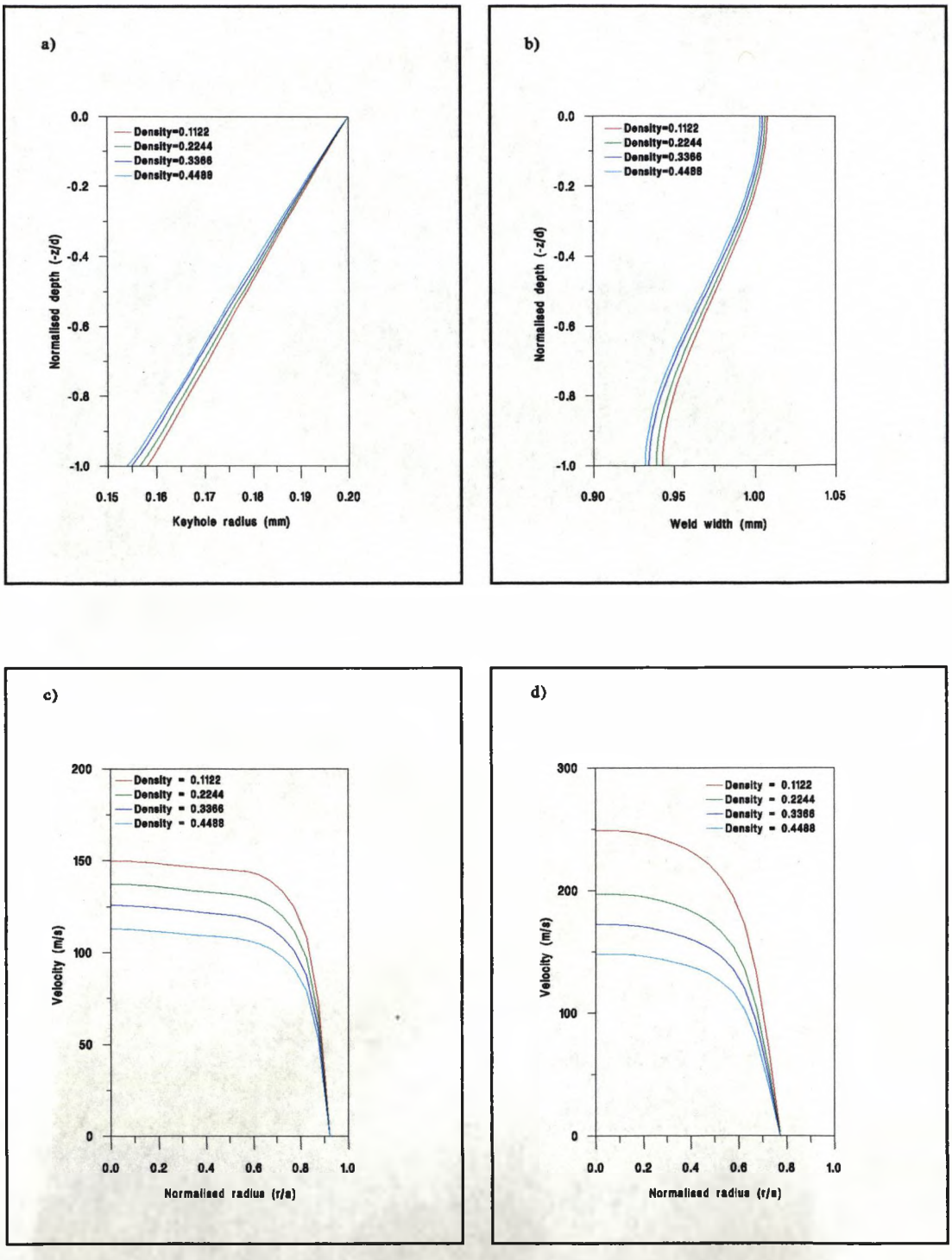


Fig.3.11 Model results for welding 2mm mild steel at 2.0kW of power, illustrating the effect of vapour density upon a) the keyhole radius, b) the weld width, c-d) the velocity of the vapour exiting the top and bottom of the keyhole respectively.

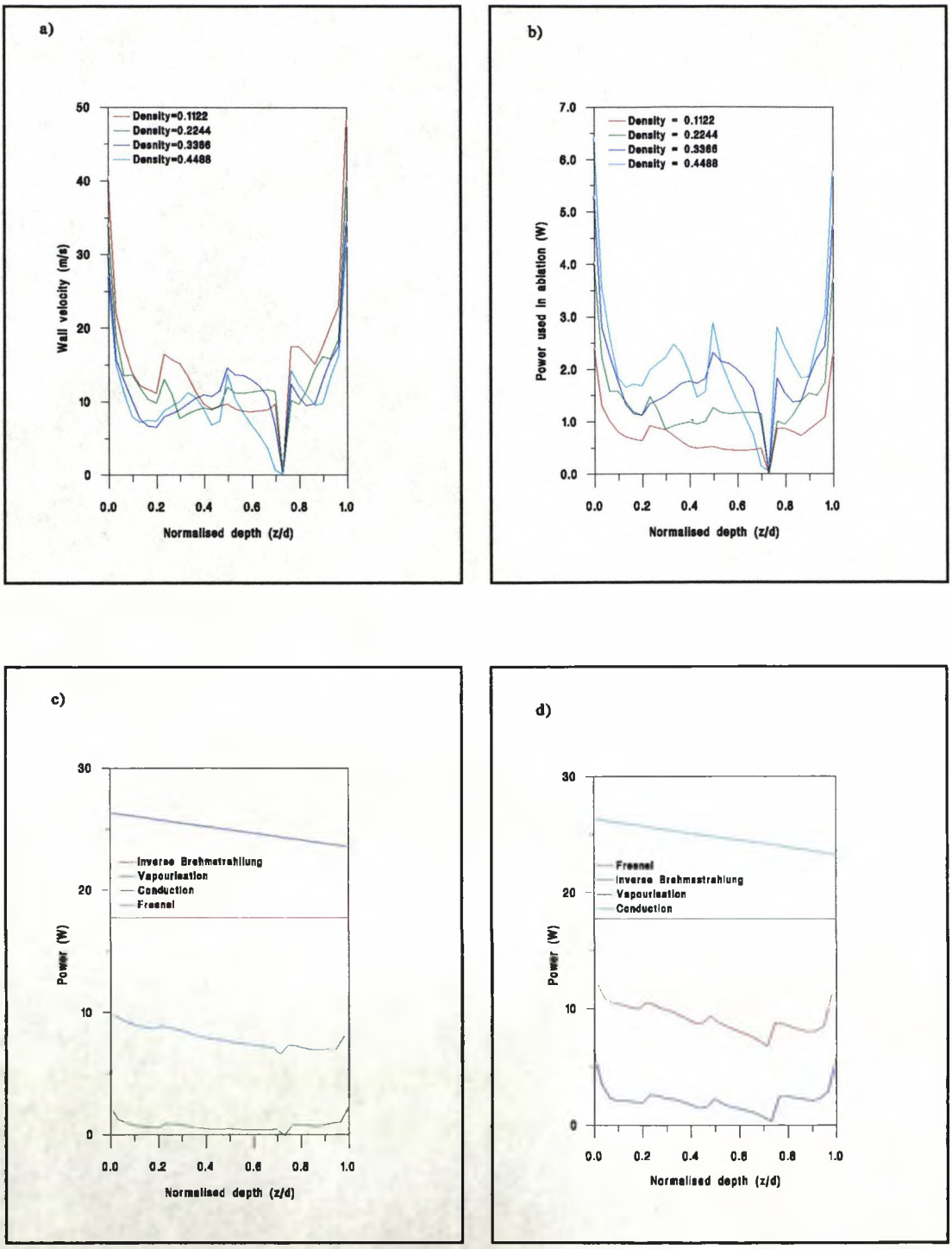


Fig.3.12 Model results from welding 2mm mild steel at 2.0kW of power, illustrating the effect of beam radius upon a) the velocity of ablating material from the keyhole wall, b) the power used up in the ablation process, c-d) the contributions to the input and the output power for the cases of c) ρ=0.1122 kg/m^3, and d) ρ=0.4488 kg/m^3.

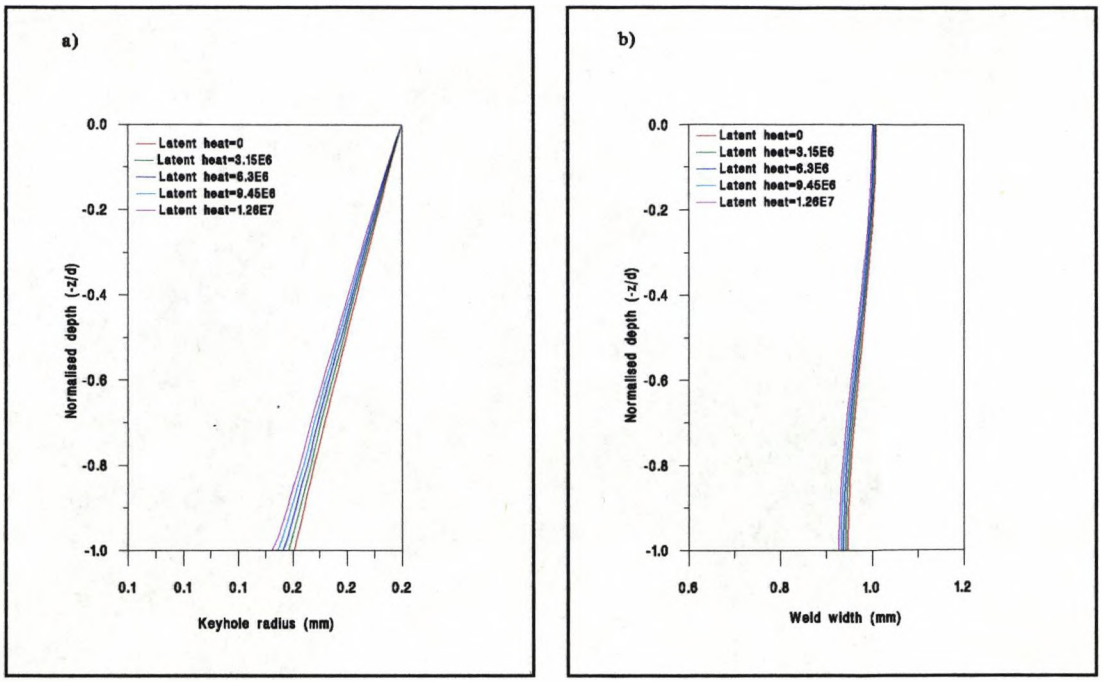


Fig.3.13 Model results for welding 2mm mild steel at 2.0kW of power, illustrating the effect of latent heat upon a) the keyhole radius, b) the weld width.

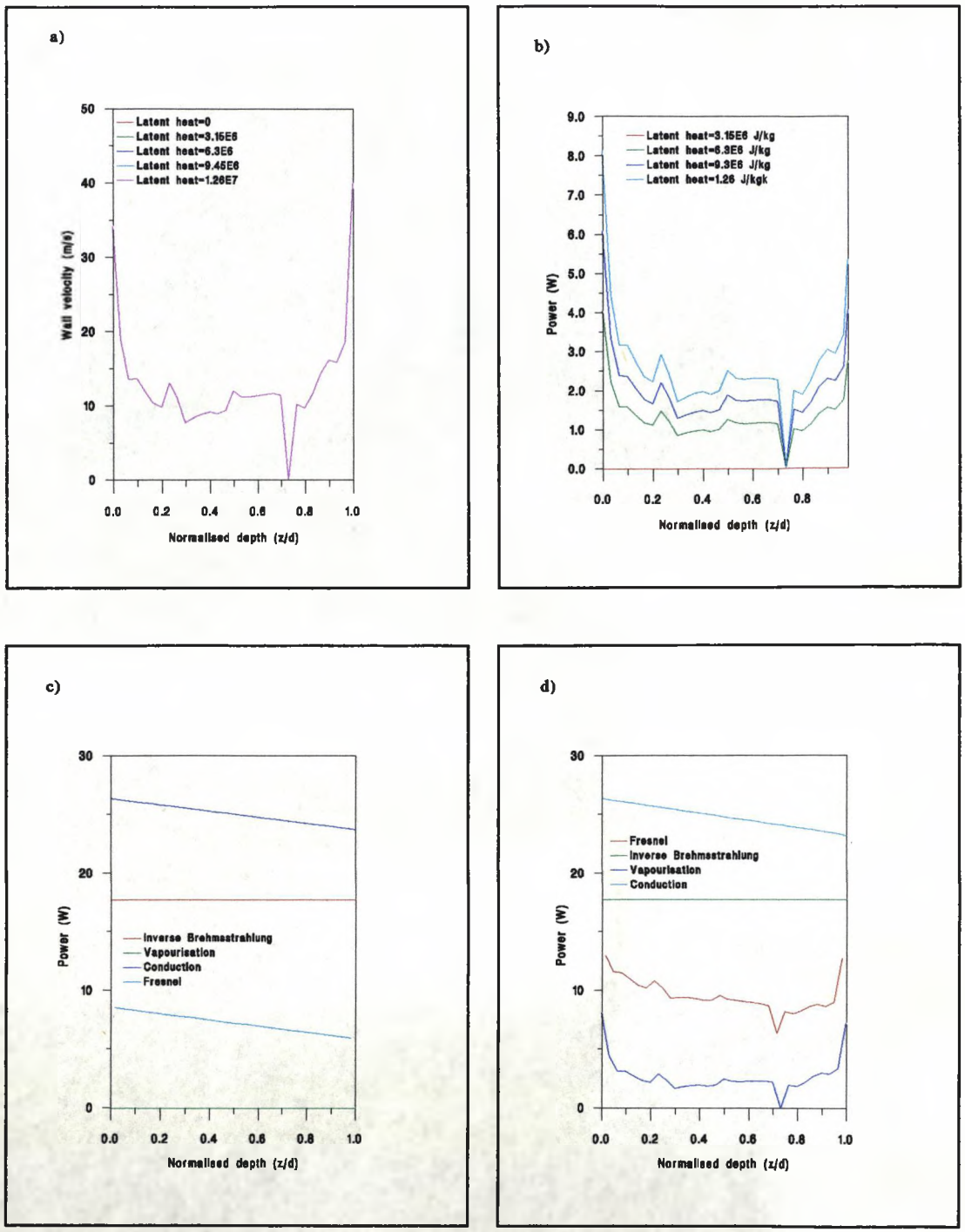


Fig.3.14 Model results from welding 2mm mild steel at 2.0kW of power, illustrating the effect of latent heat upon a) the velocity of ablating material from the keyhole wall, b) the power used up in the ablation process, c-d) the contributions to the input and the output power for the cases of c) $\Delta L=3.15.10^6$ J/Kg, and d) $\Delta L=3.15.10^7$J/kg.

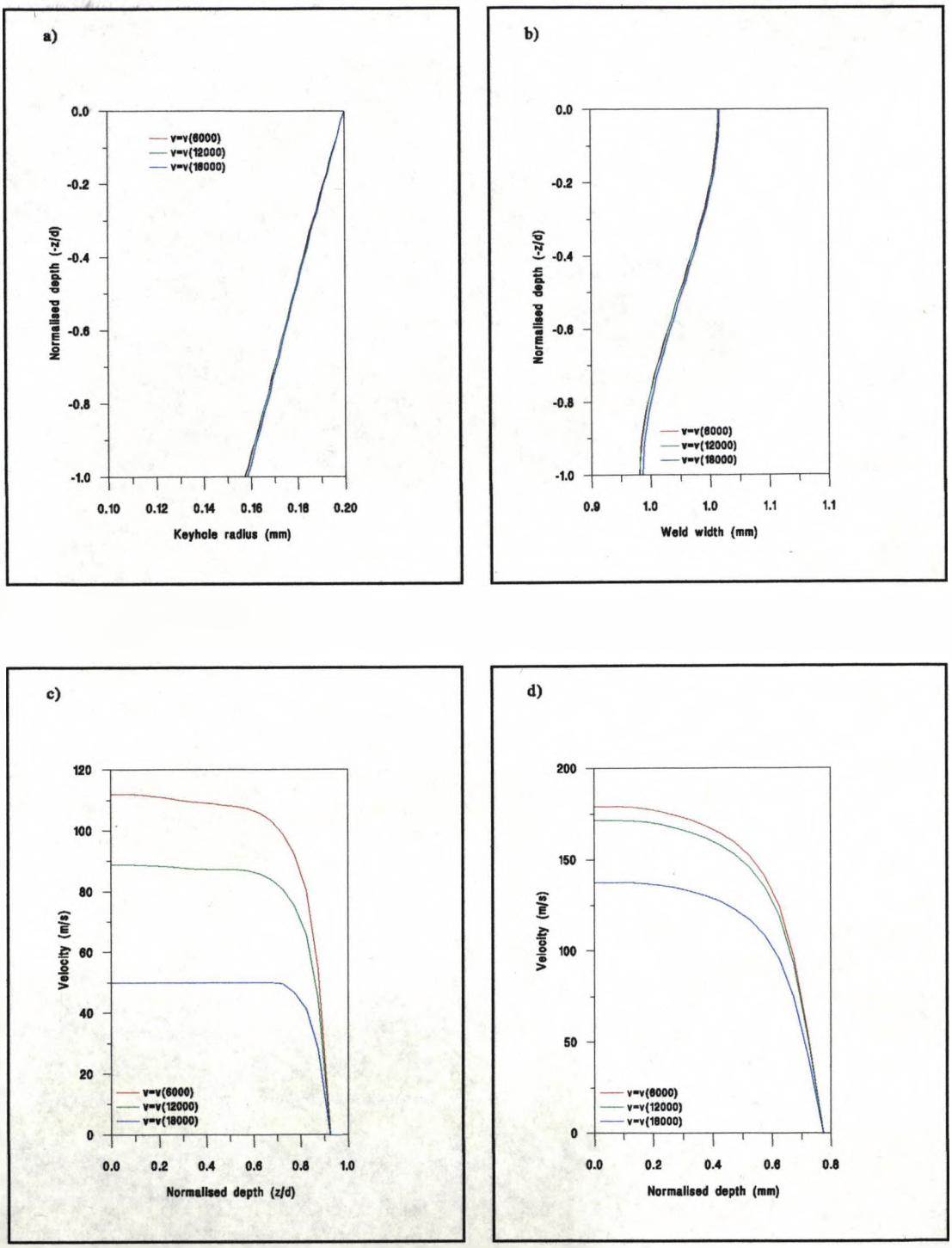


Fig. 3.15 Model results from welding 2mm mild steel at 2.0kW of power, illustrating the effect of viscosity upon a) keyhole radius, b) the weld width, c) top keyhole exit vapour velocity, and d) bottom keyhole exit vapour velocity.

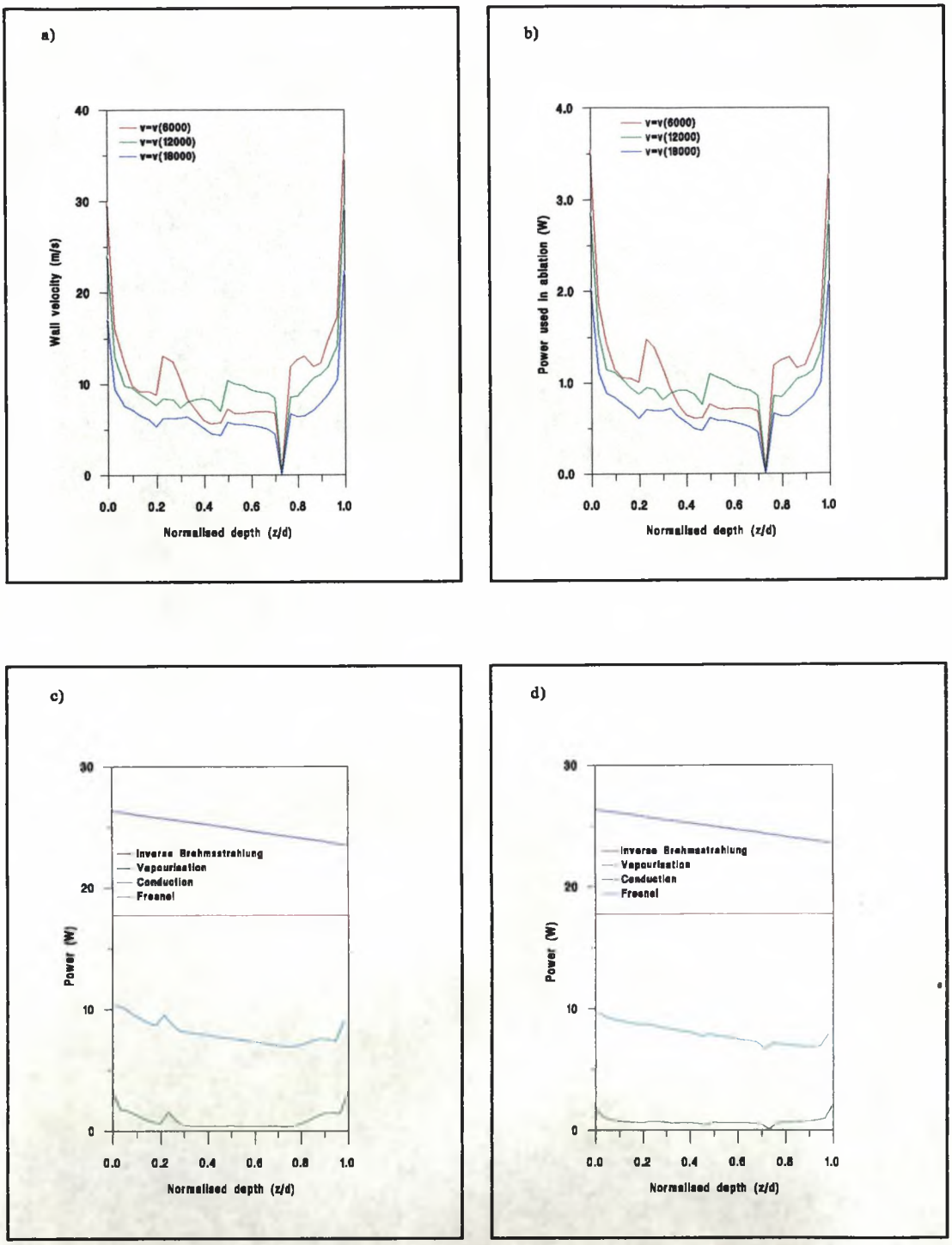


Fig 3.16 Model results from welding 2mm mild steel at 2.0kW of power, illustrating the effect of viscosity upon a) the velocity of ablating material from the keyhole wall, b) the power used in the ablation process, and c,d) the contributions to the input and output power for the cases of c) viscosity = μ(6000) and d) viscosity =(18000).

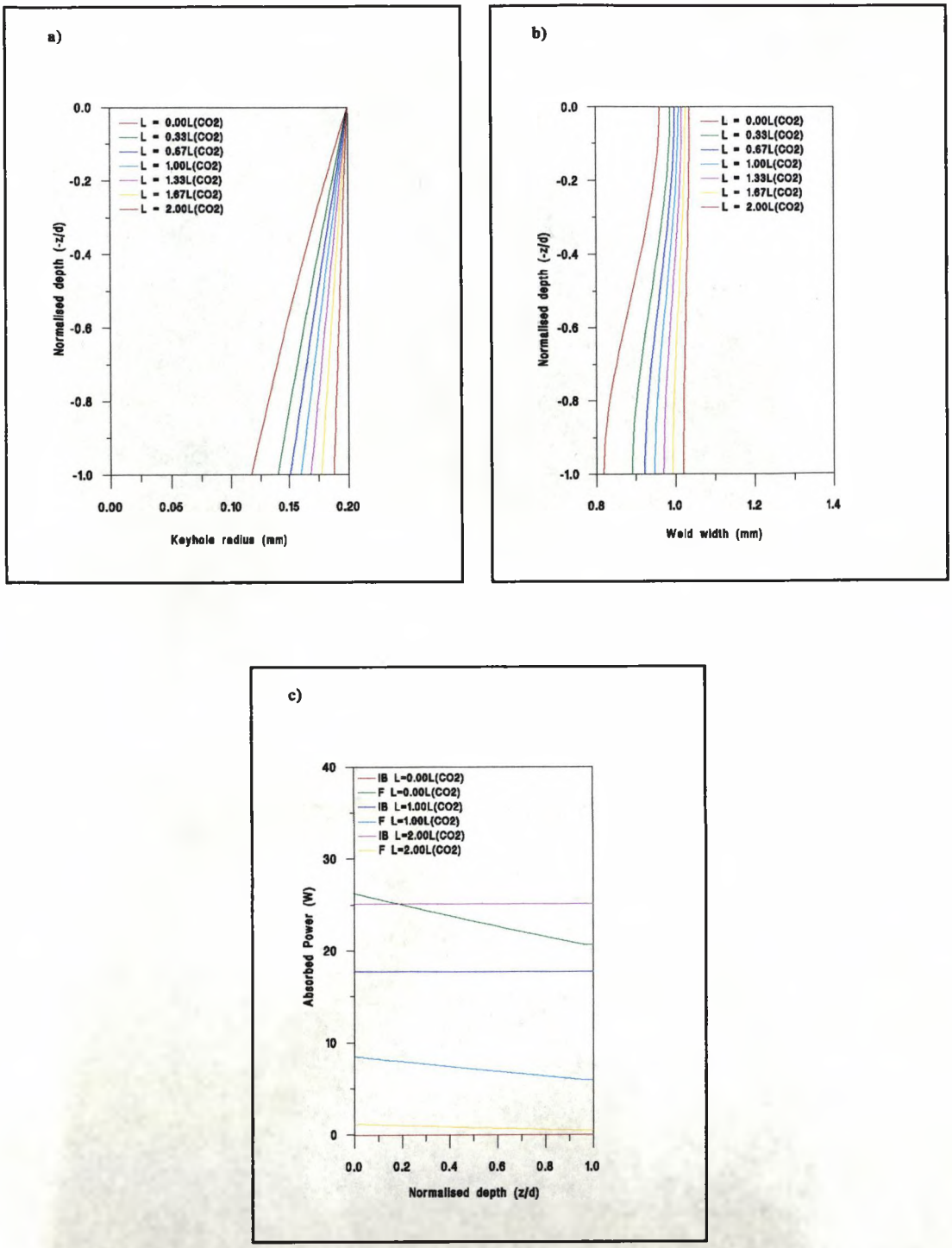


Fig. 3.17 Model results for the welding of 2mm mild steel with a 2.0kW laser beam, for different values of linking intensity: a) and b) show the keyhole and weldbead coordinates respectively, and c) shows the absorbed power from both the Fresnel and inverse Bremsstrahlung mechanisms for the linking intensity as 0,1 and 2 times the value of the CO_2 linking intensity.

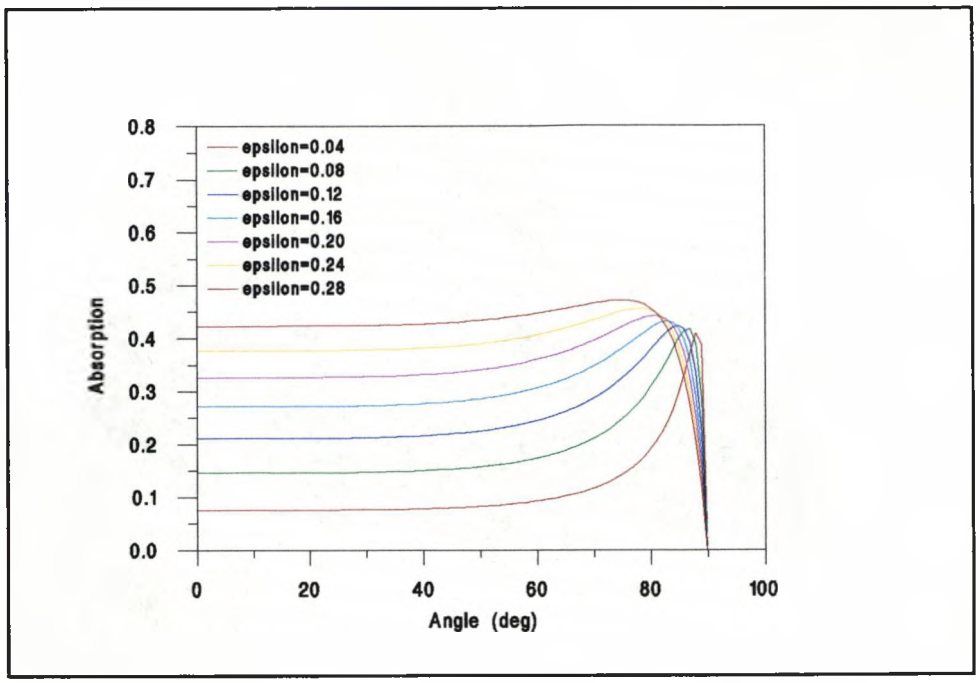
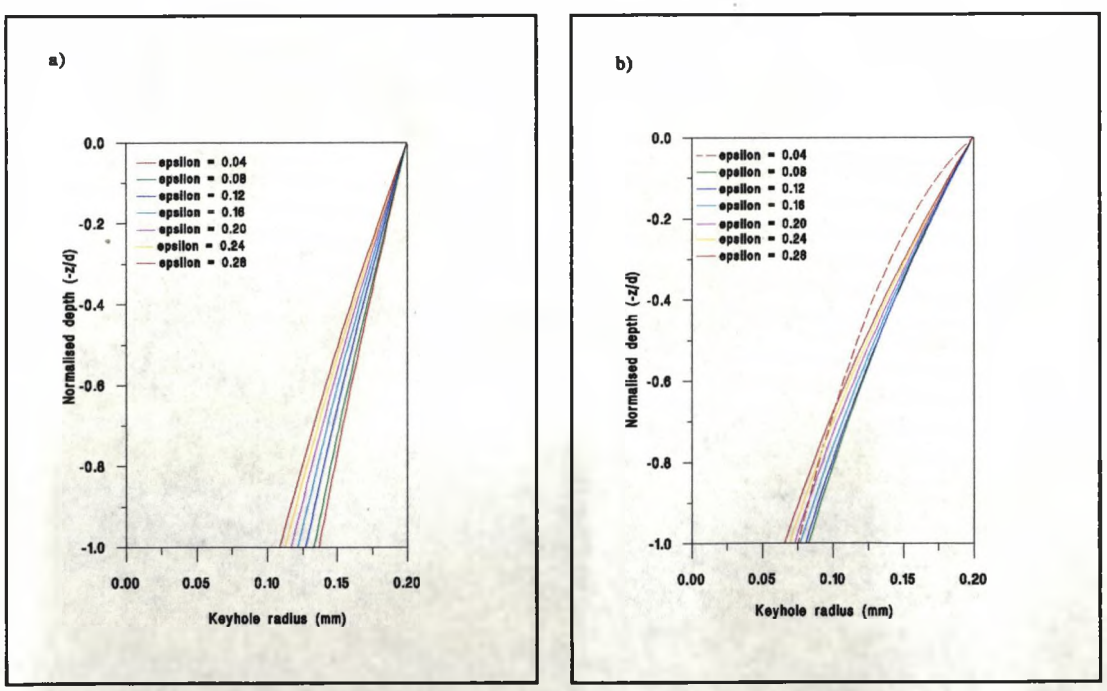
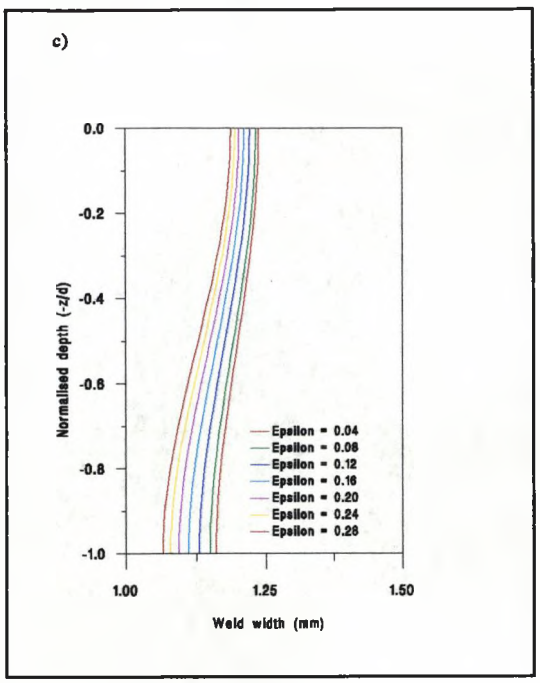
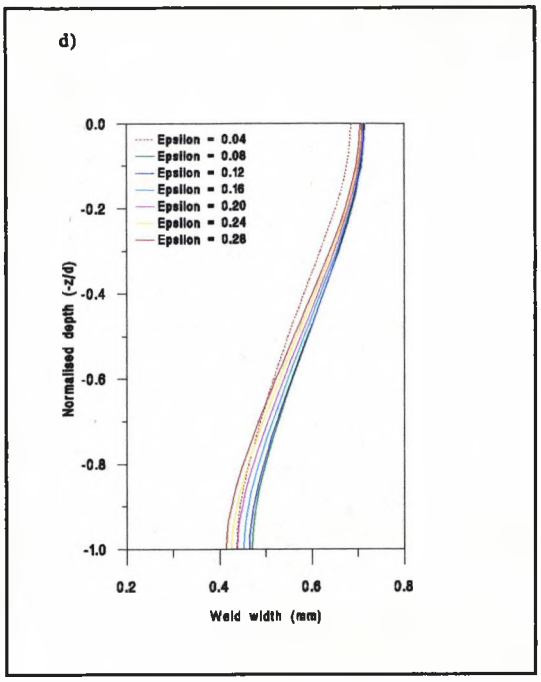
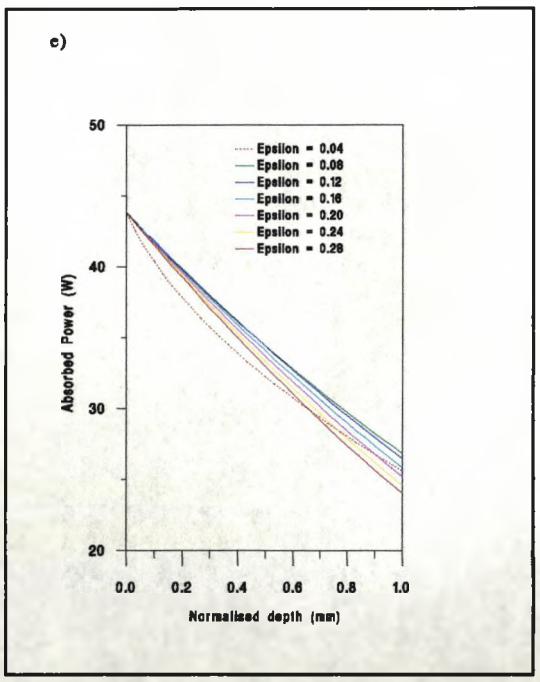


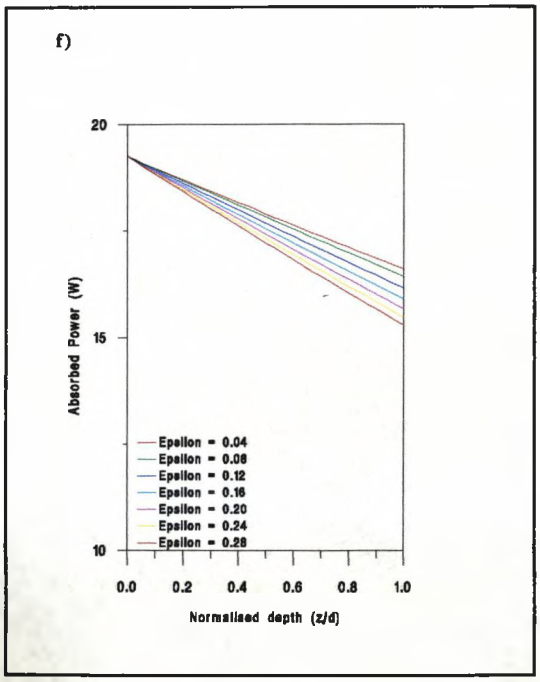
Fig.3.18 The variation of absorptivity with the angle of incidence for different values of ε.




c)
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
Normalised depth (-z/d)
1.00
1.25
1.50
Weld width (mm)
Epsilon = 0.04
Epsilon = 0.08
Epsilon = 0.12
Epsilon = 0.16
Epsilon = 0.20
Epsilon = 0.24
Epsilon = 0.28


d)
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
Normalised depth (-z/d)
0.2
0.4
0.6
0.8
Weld width (mm)
Epsilon = 0.04
Epsilon = 0.08
Epsilon = 0.12
Epsilon = 0.16
Epsilon = 0.20
Epsilon = 0.24
Epsilon = 0.28


e)
50
40
30
20
Absorbed Power (W)
0.0
0.2
0.4
0.6
0.8
1.0
Normalised depth (mm)
Epsilon = 0.04
Epsilon = 0.08
Epsilon = 0.12
Epsilon = 0.16
Epsilon = 0.20
Epsilon = 0.24
Epsilon = 0.28


f)
20
15
10
Absorbed Power (W)
0.0
0.2
0.4
0.6
0.8
1.0
Normalised depth (z/d)
Epsilon = 0.04
Epsilon = 0.08
Epsilon = 0.12
Epsilon = 0.16
Epsilon = 0.20
Epsilon = 0.24
Epsilon = 0.28

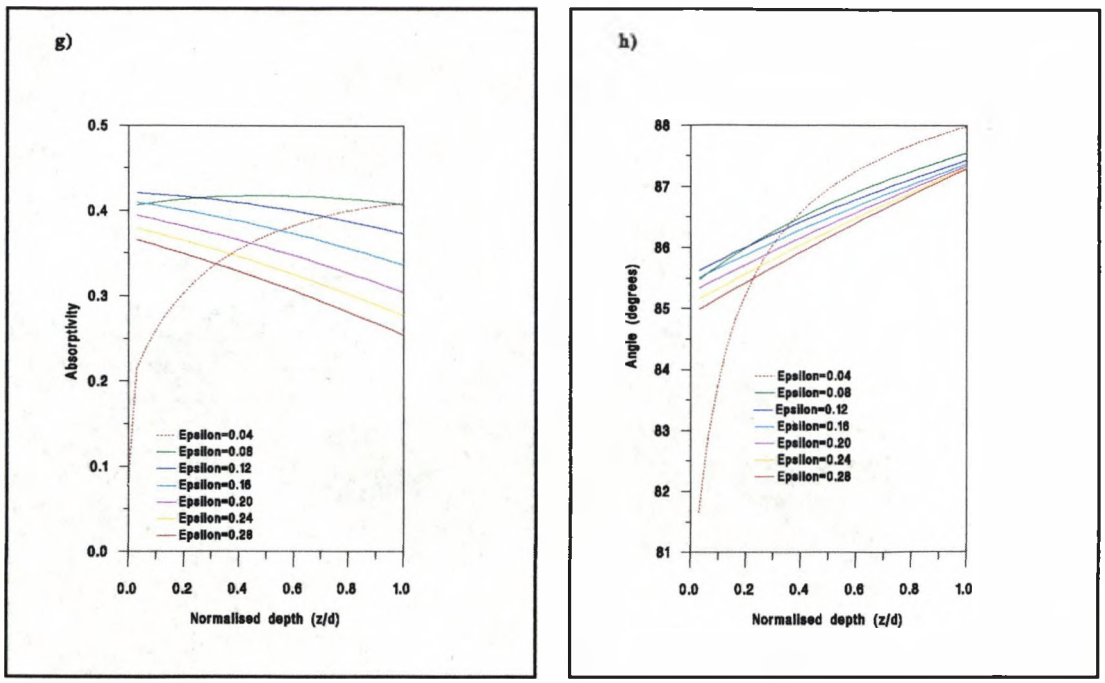


Fig. 3.19 Model results for welding 2mm mild steel with a 2.0kW CO_2 laser for different values of ε. a-b) show the keyhole coordinates for velocities of 20 and 50mm/s respectively, c-d) show the weld width for 20mm/s and 50mm/s respectively, and e-f) show the absorbed power for 20 and 50mm/s respectively, g-h) show the absorption at the keyhole wall and the angle of incidence of the laser light as a function of depth down the material in the 50mm/s case.

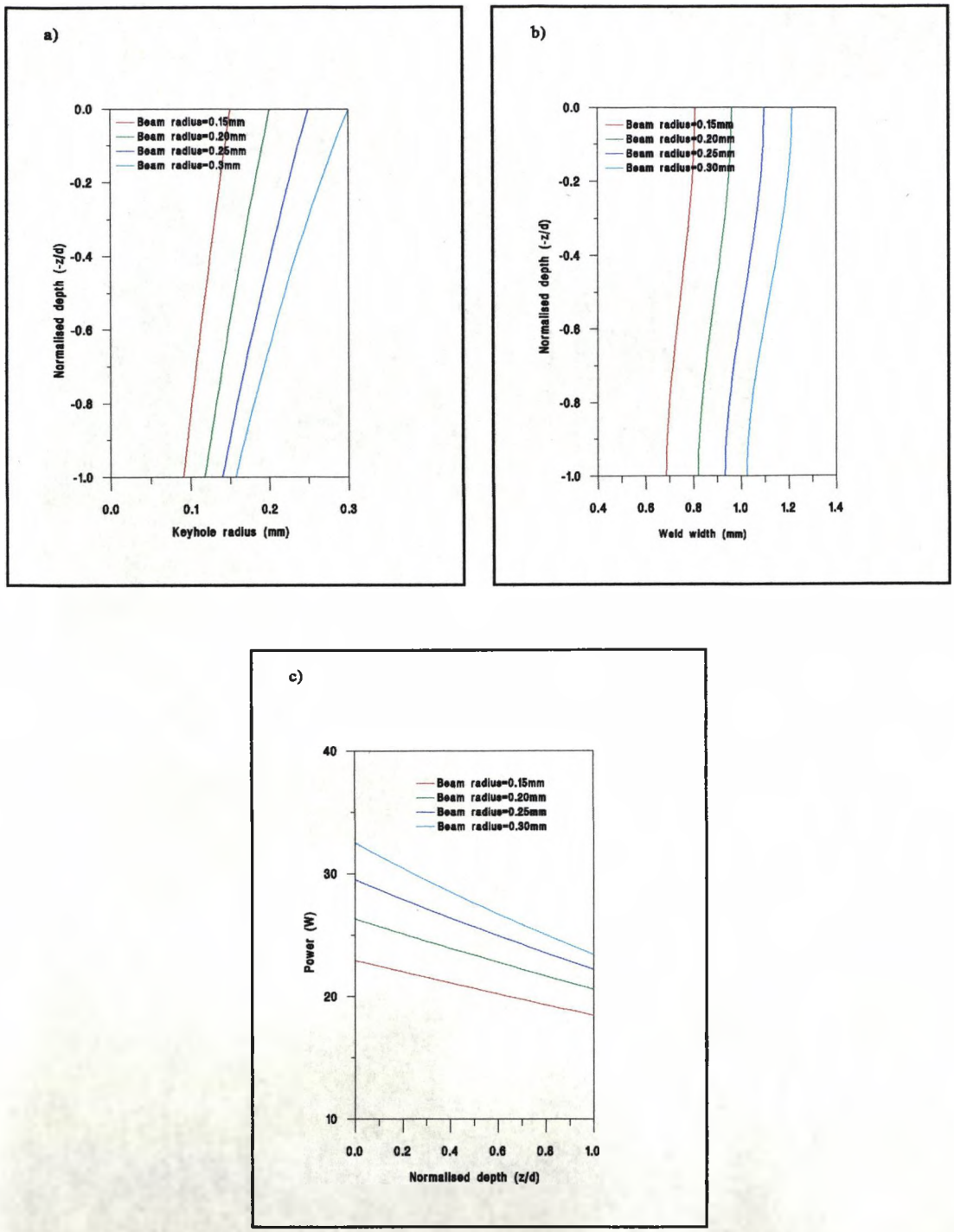


Fig. 3.20 Model results for welding 2mm mild steel with a 2.0kW beam at a speed of 20mm/s, using different beam radii, a) and b) show the keyhole and weldbead coordinates, and c) shows the variation in absorption down the keyhole.

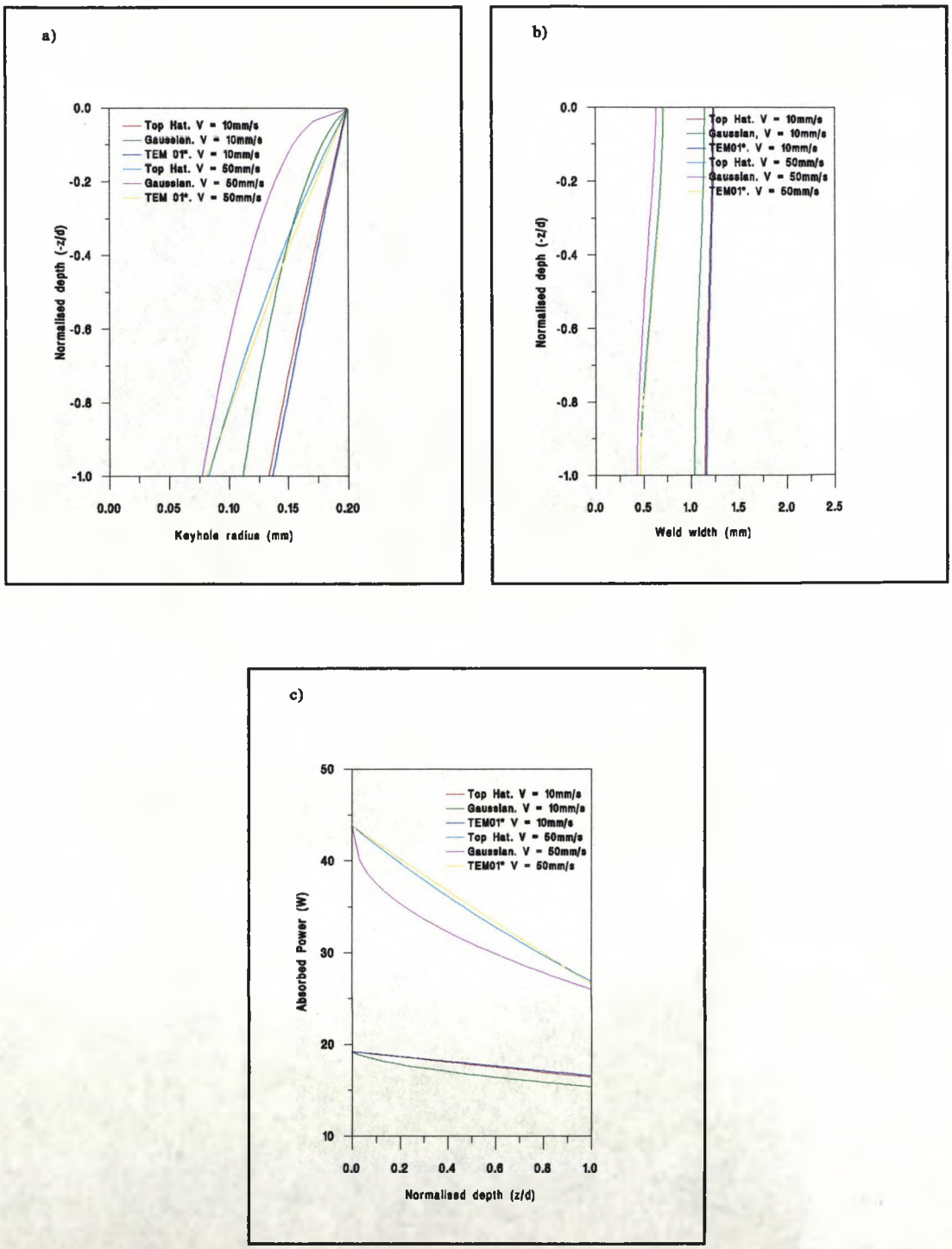


Fig.3.21 Model results for welding 2mm mild steel with a 2.0kW beam at a speed of 20mm/s, using different power distribution functions and beam radii, a) and b) show the keyhole and weldbead coordinates, and c) shows the variation in absorption down the keyhole.

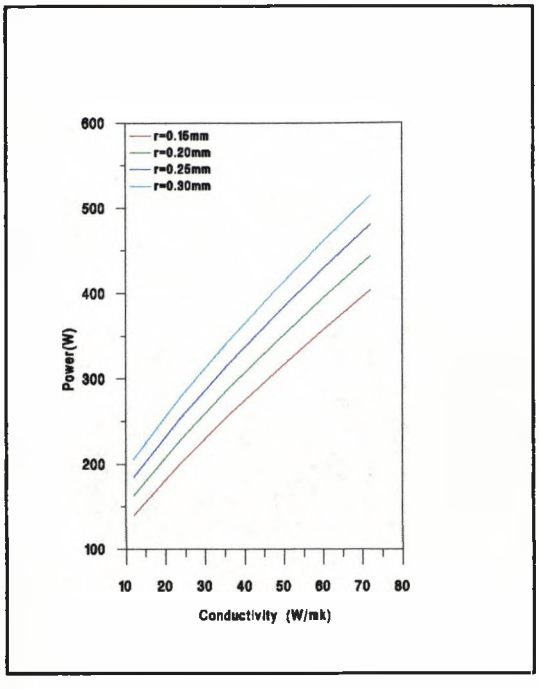
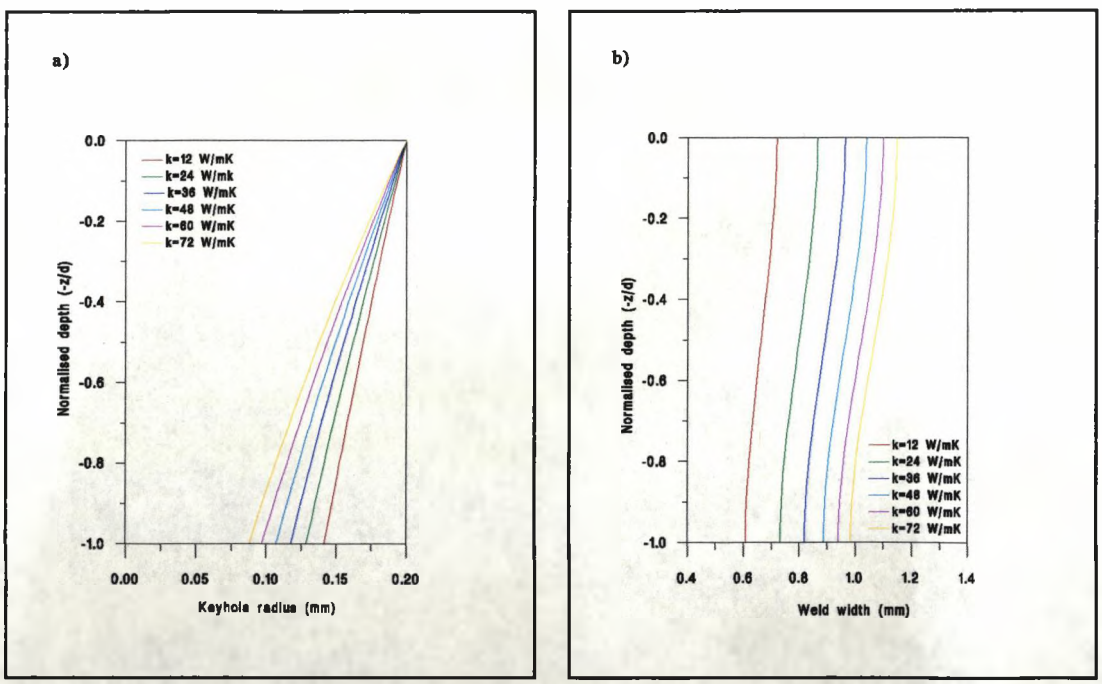


Fig.3.22 The variation of power required to conduct the material away from the keyhole against the value of conductivity, for different values of keyhole radii.



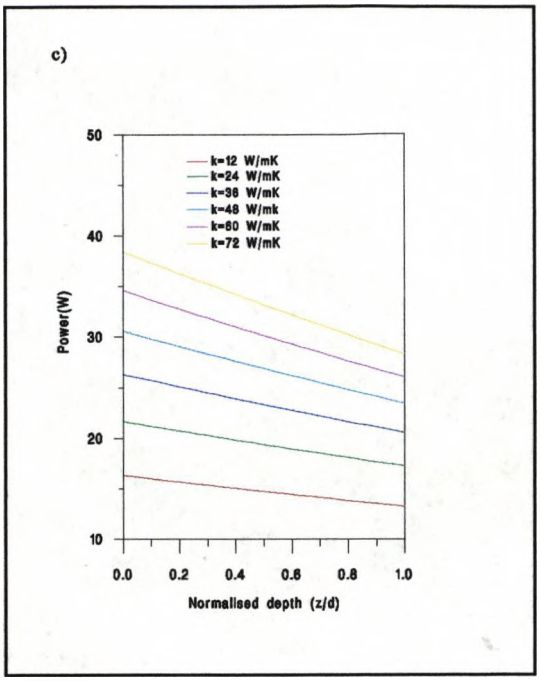


Fig.3.23 Model results for welding 2mm mild steel with a 2.0kW beam at a speed of 20mm/s, using different values of conductivity, a) and b) show the keyhole and weldbead coordinates, and c) shows the variation in absorbed power down the keyhole.

Chapter 4 Experimental Work and Validation of the Mathematical Model

This chapter presents some of the results obtained from an investigation into the effects of varying a number of process parameters on the laser welding process. Specifically, the effects of input power, process speed, workpiece thickness, beam radius, shroud gas and wavelength of laser light will be examined. These results are then compared to theoretical predictions in order to validate the model presented in the previous chapter. The full set of results obtained are given in appendix 5, here, to avoid tedious repetition, specific results that illustrate the underlying physics of the overall process will be presented and discussed.

4.1 Laser Welding Experiments

4.1.1 Experimental Arrangement and Procedure

The experimental arrangements for laser welding with refractive and reflective optics are drawn schematically in figures 4.1 and 4.2 respectively. The equipment used in these experiments consisted of the laser, the optics, the gas delivery system, the diagnostic equipment, and the clamping equipment.

4.1.1.1 The laser

Four fast axial flow (FAF) lasers were used for this study, two CO_2 lasers, one of which was used in medium power trials, the other for low power trials; similarly a medium and a low power CO laser were used. Both carbon dioxide lasers were based at Liverpool University, the AF5L was used for medium power experiments, and the OPL 2kW laser was used for low power trials. The higher power CO laser based at The Welding Institute (TWI), was the CO laser equivalent of the AF5L mentioned earlier. For low power CO laser experiments, the 800W FAF laser at Heriot Watt University was used.

4.1.1.2 The Optics

In the case of the CO laser at TWI, the beam was focused using a 150mm focal length parabolic mirror; in all other experiments lenses with focal lengths between 100 and 150mm were used for focusing. In both cases the focusing optic was cooled by water, which flowed inside the mirror, or around the lens holder in the lens housing.

4.1.1.3 The gas delivery equipment system

A 4mm diameter copper tube was used to deliver gas at the desired flow rate over the weld. It was placed approximately 3mm from the position where the beam was focused, at an angle of 45% to the horizontal. A smaller flow rate of gas was used to protect the lens from back spatter as well as from the iron vapour emerging from the keyhole. This gas was blown through the lens housing,

below the lens, and exited through the coaxial nozzle (figure 4.1 and 4.2). This gas did not effect the welding process.

4.1.1.4 The diagnostic equipment

Diagnostic equipment was used before processing in order to measure the power at the workpiece, the beam diameter and also the beam mode. The power itself was measured as the beam left the laser, however, this value is greater than the power at the workpiece, because the beam is partially absorbed by the mirrors in the beam delivery system. Therefore, before laser processing commenced, the power from the beam was measured at the workpiece using a calibrated power meter in order to determine the laser welding input power.

The spotsize at the workpiece was measured by rapidly moving a piece of perspex, at a fixed angle of incidence, across the beam path. The resulting burn print, measured using a surface profiler, revealed the size of the laser beam diameter as a function of depth (figure 4.3).

The beam mode was measured by firing an unfocussed beam at a block of perspex for 1-2 seconds. The spatial variation of the evaporated perspex then revealed the spatial intensity of the laser beam. This is shown schematically in figure 4.4.

4.1.1.5 The clamping equipment

During the laser welding process, the rapid melting and solidification, together with the steep thermal gradients generated, would induce distortion within the workpiece which would effect the overall laser welding results. In order to minimise this effect therefore, a welding jig was built in order to clamp the workpiece during welding.

4.1.2 Experimental Procedure

Figures 4.1 and 4.2 illustrate the experimental arrangement for refractive and reflective optics respectively. The beam was delivered to the focusing optic by a series of plane mirrors. After the power at the workpiece had been calibrated with the power at the laser output mirror, and a perspex impression of the beam print had been made, the beam was focused, and the workpiece was positioned so that the surface of the workpiece was at the focal point, which was found by taking spot welds with argon shroud gas at various distances from the focusing optic near the focal length. The focus was found when the spot weld produced a 'blue flash', which indicated that the power density was great enough to create an argon plasma.

Bead on plate welds were made in steel samples, using different process parameters. The welds were then cut, polished and then etched in a mixture of nitric acid and alcohol in order to

reveal the molten cross section. Photographs of these weld cross sections were then taken in order to measure the weld bead dimensions.

4.1.3 Comparison of Results

Referring back to the schematic diagram of a cross section of a weld bead in figure 1.15, it can be seen that there are three main features here worthy of note. These are the penetration depth, the top bead weld width, and the stem weld width. Taken together, these dimensions illustrate how the power from the laser beam has been coupled into the workpiece. These dimensions therefore provide vital information as to the size of the keyhole and the absorption mechanisms within it. In order for any theoretical or experimental investigation into the laser welding process to be successful therefore, these features must be taken into account in all comparisons of variations in process parameters.

4.2 Experimental Investigation into the Effects of Varying Fundamental Process Parameters Upon the Laser Keyhole Welding Process

4.2.1 Power and Speed

Figures 4.5 to 4.8 show the results of varying the power and the speed when welding 6mm mild steel plate using argon and helium shroud gases, with lens focal lengths of 150 and 100mm. It is obvious from these graphs that all weld dimensions decrease with increasing speed. More notable is the rapid increase in the weld width as the traverse speed falls below 10mm/s. This is due to the increased ionisation in the plume and the keyhole as well as the increase in beam/material interaction time. The steep temperature gradients that exist within the plume result in a variation of refractive index, this has the effect of defocussing the beam above the workpiece, and thus the effective spotsize at the mouth of the keyhole becomes larger. This increases the value of keyhole radius at the top of the keyhole and hence increases the weld widths.

Another notable feature is that at constant speeds above 10mm/s there is no significant difference in either the top or stem weld widths over the power range considered here, when argon was used as a shroud gas. As the top bead weld width is proportional to the top keyhole radius, which in turn depends upon the amount of plasma present in the plume, one can conclude that there is no significant change in the amount of ionisation in the plume over the range of powers considered here, or that if there is, it has an insignificant effect on the laser welding process. This is not the case with helium shroud gas however, where for any constant traverse speed above 10mm/s, there is a significant increase in top bead weld width with power. Therefore that there is a significant rise in plasma presence in the plume when using helium, but not argon as a shroud gas. These results can be explained in terms of ionisation energies. The more easily ionised argon gas reaches single

ionisation at lower powers than helium. The energy input required to raise the temperature further to the point where enough second ionisation has taken place to make a difference to the process was not reached under the above process conditions, and so no significant change was observed in top bead weld width using argon as a shroud gas. Helium however, is not so easily ionised, and the increasing weld width with power indicates the increasing single ionisation in the plume. With a further increase in power therefore, it would be expected that a point would be reached where the argon shroud gas would absorb enough energy from the beam for the second ionisation potential to be reached by enough ions to make a significant difference to the process, leading to an increase in keyhole radius and hence weld width.

4.2.2 Lens focal length

When considering the variation of focal length, a number of factors need to be taken into account. The first is the beam radius at the workpiece; referring back to equation 3.1, one can see that the focal spot size is proportional to the focal length. If the beam were parallel, therefore, the welding process would benefit from the shortest focal length lenses available. However, as has already been mentioned, the shorter the focal length, the greater the beam diverges after focusing. Therefore, the power density decreases rapidly down the material, this can become a problem when welding thicker materials, as there will not be enough power density at the bottom of the keyhole in some circumstances to keep the keyhole open. Thus a balance has to be made between spotsize and depth of focus. In addition to these two important factors, a higher power density would increases the amount of plasma in the plume, which, if large enough, would then defocus the beam significantly. This could mean that the effective spotsize at the workpiece would be greater with a shorter focal length lens, and the power input to the rest of the keyhole would be diminished due to the fact that more power would be absorbed in the plume, leading to a decrease in penetration depth.

Bearing these facts in mind, figures 4.9 and 4.10 show the effect of varying the lens focal length in the laser welding of 6mm mild steel, using argon and helium shroud gas, at a power of 1.5kW. In both cases the longer focal length lens gives greater penetration depths, due to problems encountered when using a short focal length lens mentioned above. When welding 3mm mild steel with focal lengths of 0.3 and 0.4mm however (figures 4.11 and 4.12), the beam-plume interaction time is much lower due to the fact that the maximum welding velocity is much higher, and hence the problems encountered in welding 6mm mild steel due to the plasma decrease. Furthermore, as the material is much thinner, the defocussing effects further down the keyhole due to beam divergence are also less significant, and hence the dominating factor in the comparison of different focal lengths is the power density at the workpiece. Thus greater penetration is achieved with the shorter focal length lens when welding with both helium and argon shroud gases. The weld widths at the top of the keyhole are proportional to the size of the focal spot, and so the greater top bead weld widths are

obtained with the longer focal length.

4.2.3 Shroud gas.

The amount of ionised gas in the plume is inversely proportional to the ionisation energy of the shroud gas used in the application, and proportional to the power density and the interaction time of the beam with the plume. The plasma can enhance the coupling of the energy from the beam into the material. A problem occurs however, when the plasma becomes highly ionised and absorbs too much laser light, in which case the power available to the keyhole is diminished dramatically, and the beam begins to defocus above the workpiece decreasing the power density. In this case the penetration depth decreases and the top surface weld width increases.

Considering that helium, argon and iron have ionisation energies of approximately 24eV, 14eV and 7eV respectively, one way of decreasing plasma problems in the plume is to use helium as a shroud gas rather than argon. Figures 4.13 and 4.14 show the results from bead on plate welds in 6mm mild steel at 2.5kW and 1.5kW with a 150mm focal length lens. Here, welding with helium shroud gas results in greater penetration depths at the higher power densities (figure 4.13) where the plasma can become a problem when using argon shroud. The corresponding weld widths are similar in size for most of the velocity range. It is significant to note that using 1.5kW (figure 4.14), there is no noticeable difference between welding with argon or helium, as the power density isn't high enough to create enough ionisation for a significant change in the absorption mechanism to occur. Another interesting set of results illustrating both the beneficial and detrimental effects of the absorption within the plume were obtained when welding 5mm mild steel using a power of 1.5kW using a focal length of 125mm (figure 4.15). Here, two regimes exist. The first is at lower speeds were the welds in argon result in a loss of penetration depth, due to the build up of plasma in the plume and the detrimental effects associated with it at low speeds. However, as the speed increases, welding with argon results in larger penetration depths, due to the increased absorption in the plume, which is not a detrimental at these speeds because the beam interaction time is small.

4.2.4 Comparison with theoretical predictions for CO_2 laser welding

Three models were chosen in order to compare theoretically calculated results with experimental data. The first model of the laser welding process, in line with other researchers, described the keyhole radius as a balance between the power absorbed from Fresnel absorption, and the heat conducted away from the material, thus neglecting plasma absorption within the keyhole. The second model considered the plasma absorption to be constant, while the third considered this quantity to vary down the depth of the material. In figures 4.16 and 4.17, these are referred to as models 1, 2 and 3 respectively.

Looking at the results of calculated and experimental top bead weld widths for full penetration welds in 2 and 3mm mild steel plate (figures 4.16-4.18), one can see that the theory tends to underestimate this value by up to 50%. This is because the model does not take the effects of the plasma above the workpiece, nor the Marangoni convection in the weld pool into account. The plasma not only acts as a heat source to the rest of the material, but also defocusses the beam above the workpiece, and hence increases the effective beam radius; this results in larger than predicted weld widths. As described in chapter 2, Marangoni convection at the top and bottom surfaces will tend to increase the width of the meltpool in these regions. The models seem to predict stem widths with much more success, although here too they tend to underestimate these values, presumably due to the fact that the model does not take into account the multiple reflections within the keyhole nor the effect of a wider top keyhole radius. It seems that model 2, with its constant plasma absorption contribution gives better results than the other 2 models, and so this will be the model used in all further calculations.

In order to estimate the penetration depth for a particular set of process parameters, it has been suggested by a number of researchers that there exists a critical ratio between top and bottom keyhole radii; once this critical ratio is met, full penetration ceases to occur. It is not fully understood as yet why this should be the case, but it is an empirical relationship which has been observed from studying a large number of welds. After comparing theoretical predictions with experimental results, the optimum value for this critical ratio was estimated to be 0.5; i.e., when the calculated keyhole radius becomes half of the value at the top of the keyhole, the laser beam fails to penetrate the workpiece fully. Using this empirical relationship, figure 4.19 shows the comparison of theoretically calculated maximum welding velocities for various workpiece thicknesses, with those obtained from experiment, using a beam radius of 0.2mm, and powers of 1.5, 2.0 and 2.5kW. As can be seen, although the model overestimates the value of the penetration depth in thicker materials, where plasma effects in the plume begin to dominate the process, the model predictions agree well with experiment for thinner workpieces.

4.2.4.2 The effect of varying lens focal length

In this section, the effect of the diverging laser beam after the focus is taken into account by assuming that the defocussing in the plume is negligible. In order to take the varying beam radius into account, a modified formula for the power absorbed by the Fresnel mechanism was developed. This was done by calculating the power input into a differential element at the keyhole walls, and then multiplying by the absorption coefficient. This power was then divided by δz in order to obtain the absorbed power per unit depth, thus

$$Q_F \approx 2\frac{P(z)A(\theta)r\delta r}{r_b(z)^2\delta z} \qquad 4.1$$

where δr is small change in r for that element and $r_b(z)$ is the beam radius at depth z, given by equation 3.2, and P(z) is the amount of power that has not been absorbed by the Inverse Bremsstrahlung mechanism.

The variation of keyhole radius with depth when welding 2mm mild steel at a velocity of 50mm/s, with different focal length lenses is shown in figure 4.20 a). Here it can be seen that, especially for the longer focal lengths, the power density at the top of the keyhole is the dominating factor in the average slope of the contour. This is illustrated by comparing the 200mm length lens case with the 100mm lens case. There is not enough power density at the workpiece with the 200mm lens to create a sustainable keyhole in the 2mm plate, whereas the 100mm lens, with four times the power density, easily does so. Therefore, at this thickness, the divergence of the laser beam is not a large problem. During the laser welding of 4mm mild steel at 2.0kW using a velocity of 10mm/s (figure 4.20b), the process becomes more dominated by the inverse Bremsstrahlung mechanism at greater depths down the keyhole. It has already been noted that this contribution defines a minimum value of keyhole radius, which becomes important at slower welding speeds. In the top half of the keyhole, one can see that the shallowest slope is founded by the 200mm lens. However, at the slower processing speed, the gradient of the keyhole contour is less steep, especially in comparison to the keyholes generated by other lenses. This is illustrated more clearly in Figure 4.21, which shows the maximum welding speed as a function of workpiece thickness for different focal lengths. Here, it can be seen that there is a huge gap between the maximum welding speeds in thin materials, where the slope is heavily dependent upon the welding speed. As the thickness increases, the difference between the focal lens used decreases, as the process becomes more and more dominated by inverse Bremsstrahlung absorption. Unfortunately, however, the dominance of this term in thicker material and at slower speeds meant that a valid comparison could not take place beyond 5mm, where one would expect the divergence of the laser beam down the material to play a more dominant role.

4.2.4.3 The effect of varying the point source in laser welding

In order to illustrate the effects that a point source generated by the presence of the plume can have upon the calculated keyhole and weldpool dimensions, the heat balance at the keyhole walls was modified to include the additional heat input of a point source. The point source can be included in the model by assuming that the heat input at the keyhole wall is equal to the amount of energy that would simulate a point source of size q in the resultant weld. This heat input can then be calculated from the solution for a stationary point source in a material.

$$T - T_o = \frac{q_s}{4\pi kr} \qquad 4.2$$

where T is the temperature of the workpiece, q_s is the point source strength, k is the conductivity of the metal, and r is the distance from the origin. Although Rosenthal's solution for a moving point source would be a better description of the situation, the assumption of axial symmetry within the keyhole dictates that only an axisymetric energy source could be used. This assumption would imply that this analysis is only valid at lower welding speeds. The resultant heat balance at the keyhole wall would then become (neglecting vaporisation losses),

$$Q_F + Q_{IB} + Q_{PS} = Q_C \qquad 4.3$$

where Q_F, Q_{IB}, and Q_{PS} are the power per unit depth absorbed by the workpiece from Fresnel absorption, inverse Bremsstrahlung absorption and from the point source respectively, and Q_C is the power used up per unit depth in the conduction of heat away from the material.

Q_{PS} can be calculated from the temperature distribution,

$$Q_{PS}(r) = -\frac{kA}{\delta z}\frac{\partial T}{\partial r} \qquad 4.4$$

where A is the area of differential element at the keyhole wall, $2\pi r\delta z$. This obviously assumes that the heat flux is constant through the area, and r represents the average value of the distance between the differential area and the point source position. The power input to the keyhole which is used in the Fresnel and inverse Bremsstrahlung equations is not the maximum power, but the maximum power minus that used up in generating the point source. The above energy balance equation can be solved by finding a value of r that satisfies the heat balance equation at a particular depth z. This value is then taken as the keyhole radius.

Appendix 5 gives the results of welds obtained during laser processing, and includes the value of the point and line source calculated from Rosenthal's equations. The value of the point source usually lies within the region of 0 to 300W, and this will be the range of point source strengths considered here. Figures 4.22 and 4.23 show the effect the variation of the size of the point source has upon the theoretically calculated keyhole radius and weld width for 2mm and 4mm mild steel plates respectively, at a power of 2KW and a speed of 10mm/s. As can be seen, the effect of the point source is to widen the keyhole at the top, and thus to present more of the keyhole to the oncoming beam. In the cases of the larger point sources, the keyhole begins to close slightly quicker near the bottom of the keyhole, as the power input to the rest of the keyhole is diminished by the large point source presence. This effect is seen more clearly in the resulting weld widths, where the large point source increases the weld width at the top of the keyhole, but decreases as the depth

increases. Figure 4.24 shows the amount of power absorbed down the workpiece for a point source of 300W. As can be seen, the absorbed power from the point source decreases rapidly down the depth of the workpiece, this is matched by the decrease in keyhole radius, and a corresponding decrease in the heat conducted away from the keyhole. The Fresnel absorption is zero at the top of the keyhole, where the keyhole radius is greater than the beam radius, and then increases to a maximum once the keyhole radius becomes less than the beam radius. As the power input decreases down the depth of the material the Fresnel contribution thus begins to decrease correspondingly,

4.3 Experimental Investigation into the Effects of Laser Wavelength upon Laser Keyhole Welding

This investigation involved comparing the laser keyhole welding capabilities of two different types of laser, the CO laser, which operates within a wavelength band of 4.8 to 5.4μm and the CO_2 laser, which operates at 10.6μm. The reasons why CO lasers were considered worthy of study were based on the following,

a) The 100% quantum efficiency of the CO laser is more than twice as high as that of the CO_2 laser. This would result in smaller and cheaper power supplies due to higher wallplug efficiencies (energy supplied to the laser / beam energy). In actual fact, wallplug efficiencies of up to 19% have been observed in CO lasers, compared to the 10-12% wallplug efficiency typically found in CO_2 lasers. In reality, CO lasers will never have twice the efficiency of CO_2 lasers due to the fact that extra power is required for nitrogen cooling.

b) The CO laser can be transmitted down certain optical fibres, thus lending itself more easily to automation, whereas 10.6μm radiation cannot as yet.

c) The shorter wavelength should give advantages during material processing due to the fineness of focus and the increased absorption of metal surfaces. It has already been noted in the previous chapter that due to the fact that the CO laser will have a poorer quality beam mode than a CO_2 laser with similar laser cavities, the advantage in the fineness of focus was not observed.

Of course the Nd:YAG laser, operating at an even shorter wavelength, will steal some of the advantages of the CO laser (such as higher absorption and the availability of fibre optic delivery). Yet here too, the CO laser was thought to offer advantages, because the output power of pseudo-continuous Nd:YAG lasers is limited to 2kW at the moment. The beam quality of the YAG laser is very poor, resulting in larger focused spotsizes than those obtained with CO_2 or CO lasers. The Nd:YAG laser also has a lower wallplug efficiency (around 6%), than either the CO or CO_2 lasers.

4.3.1 Low Power CO/CO_2 Experiments

In order to assess the comparative capabilities of each type of laser, absorptivities of a number of materials to both CO and CO_2 laser light were measured on the 800W laser at Heriot Watt University and the a CO_2 laser at Liverpool University respectively. The absorptivity was measured by a simple calorimetric technique, whereby the sample was heated by the laser for 20 seconds, and then placed in a water calorimeter. The rise in temperature of the water was then measured and from this the absorbed power could be calculated. Figures 4.25 a-d) show the measured absorptivities of aluminium, stainless steel, steel and copper respectively. These results indicate that in the first three absorbed much more CO laser radiation than CO_2. In the case of copper however, it was thought that the experiment was not sensitive enough to be carried out on a material with such a high absorptivity.

To illustrate further the increased surface absorption with 5.4μm radiation, mild steel and stainless steel samples were surface heat treated at 580W and 525W respectively, using a beam diameter of 3.2mm and an argon shroud gas. Cross sections of the sample were cut, and the heat affected zones and melt dimensions were then measured. The results are shown in figures 4.26 a-b) for mild steel, and figures 4.26 c-d) for stainless steel. As can be seen, although there is only a slight advantage in the HAZ width when heating stainless steel samples with a CO laser, between 20-25% advantage can be gained in HAZ depth. An even greater advantage can be found when heat treating mild steel samples. Here, increases in HAZ depth of over 100%, and increases in HAZ width between 25-40% were observed. Moreover, above 15mm/s, the CO_2 laser could not produce an HAZ, and melting occurred in none of the CO_2 laser trials. With the CO laser however, hardening was observed at speeds up to 30mm/s, and molten zones could be detected up to 16mm/s. This illustrates the increased coupling with the workpiece that can be obtained with the CO laser

Welding trials on mild steel and stainless steel samples were carried out on the 800W laser at Heriot Watt. The focusing optic was a 125mm ZnSe lens. The M^2 value of the laser was measured to be 2.7. This gave a theoretical value of the focus spotsize to be 0.21mm, which was validated by using the perspex burn print technique described above (this gave values between 0.2 and 0.25mm). Both argon and helium were used as shroud gases at 6l/min. Speeds between 5 and 20mm were used for comparative trials. The results from these experiments were then compared to trials at Liverpool University on the OPL CO_2 laser (M^2 value of 1.5), using the same spotsize and process conditions. The full set of results are given in Appendix 5, the welds in 1mm mild and stainless steel are shown in figures 4.27-4.29.

At sufficient power densities during the CO_2 laser welding of mild steel using argon as a shroud gas, the argon gas in the plume becomes ionised. The beneficial and detrimental effects of this plasma in CO_2 laser welding have already been discussed in the previous section. These effects will

be dampened when using a CO laser due to the decrease in plasma absorption. It was a significant observation that during CO laser welding trials at these lower powers, there were no significant changes within weld pool dimensions with shroud gas flow rate. Indeed, welding without any shielding gas gave very little difference in the resulting weldbeads. This indicates that the power intensities used in this study, were insufficient to create a plasma, or if a plasma was created, its effect upon the laser welding process itself was negligible. Looking at the comparative results for welding 1mm mild and stainless steel plate with argon and helium shroud gas at a flow rate of 6l/min, it can be seen that when argon is used the CO laser produces smaller top weld beads than the CO_2 laser, due to the lack of plasma in the plume. Hence, more power is transmitted to the keyhole, resulting in larger penetration depths in the CO laser case. The CO_2 laser welding of stainless steel is even more sensitive to the type of shielding gas used, because the constituents of stainless steel are more easily ionised. Here, the plasma created by the CO_2 laser has become a hindrance to the process, and blocks the beam from penetrating deeper into the material. The CO laser therefore again produces larger penetration depths. It can be concluded therefore, that under these process conditions, the CO laser offers a distinct advantage over the CO_2 laser.

During the laser welding of mild steel and stainless steel samples using helium as a shroud gas delivered at 6l/min, the problem of the plasma is reduced, and the beneficial effects of the plasma are enhanced. Therefore, the main factors dominating the process are the mode structure, and the amount of energy absorbed in the plume and the keyhole. Here, during keyhole dominated welding at lower speeds, the CO_2 laser with its improved mode structure and enhanced plasma absorption results in increased coupling with the workpiece. However, as the traverse speed increases, the welding passes into the conduction limited regime, where the CO laser, with its higher absorptivity at normal incidence, produces larger melt volumes.

It was mentioned earlier that there was no difference in CO laser processing results with shroud gas flow rate, this was not the case with the CO_2 laser. Figure 4.29 show a comparison between the CO and CO_2 laser welding of 1mm mild and stainless steel plates, using argon shroud at a flow rate of 20l/min, in the CO_2 laser experiments. Here, the CO_2 laser showed a marked improvement in penetration depths and weld widths compared to the CO laser. The reason for the change in results with gas flow rate is due to the fact that when the gas rate is insufficient to displace the iron vapour, the iron atoms, with a smaller ionisation potential than argon, dominate the plume which becomes very easily ionised. Hence, the plasma then begins to block the beam, and becomes even more ionised, thus hindering the welding process. Once the argon shroud has a high enough flow rate to control the plasma, the enhanced coupling obtained due to the extra heat source in the plume becomes beneficial to the welding process as a whole, and therefore produces larger welds. The presence of the plasma was too insufficient in the CO laser welding trials to have any effect

upon the laser welding process. Under these conditions, CO laser welding was totally independent of the gas flow rate, and the enhanced coupling which can occur when a plasma is present, was not apparent.

From these results one can conclude that creating and controlling a plasma is of crucial importance to the laser welding process. If the plasma cannot be controlled, as in the case of the argon experiments on mild and stainless steel at 6l/min, the plasma absorbs too much of the beam and begins to block it, with the result that not enough power can reach the keyhole in order to increase the penetration. If the plasma cannot be created, then the keyhole mouth is not widened due to the presence of the plasma, which means that the keyhole collapses sooner. Therefore, one can conclude that the penetration depth is deeper when the plasma is present and can be controlled.

4.3.2 Medium power CO/CO_2 experiments

Experiments were undertaken at TWI on the Laser Ecosse CO laser. The power was varied in these experiments between 1.5 and 2.2kW, and focused using a 150mm focal length parabolic mirror. Helium and argon shroud gas was delivered to the workpiece by a 4mm diameter plasma control jet at 10 and 20 l/min. The spotsize at the workpiece was measured with a PROMOTEC device, and was found to be 0.31mm. The full set of welding results are shown in Appendix 5, here, the main results that reveal the physics of the process will be described.

Figures 4.30 and 4.31 show the effect of the variation of power on the CO laser welding of 6mm mild steel plate, using both argon and helium shroud gas at 20l/min. Increases in power obviously result in the increase in penetration depth. However, there is no significant difference in top bead and stem weld widths with increase in power, as with CO_2 welding with argon shroud gas. This is due to the fact that there is not enough difference in the ionisation in the plume between the two cases to make a difference in the top weld bead size.

Figures 4.32 and 4.33, show the weldbead dimensions as a function of speed for welds in 6mm plate using both argon and helium shroud gases at a flowrate of 20l/min, and at a power of 1.5 and 2.2kW. As can be seen, there is no significant difference in weldpool dimensions when using either argon or helium shroud gas, even at lower speeds. This is due to the fact that although there will be some plasma present, the power densities used in this study were not large enough to create enough of a plasma to fully ignite the plume. This is due to the fact that the plasma is 4 times as transparent to CO laser light than to CO_2 radiation, and therefore the cycle of absorption in the plume leading to greater ionisation and greater absorption, that occurs in CO_2 laser welding, will require greater powers in the CO laser case than were used here. It was a significant observation that there was no 'blue flash' encountered when welding with argon shroud gas, which indicates that the argon

plasma was not ignited.

In order to determine if the shroud gas had any effect at all upon the CO laser welding process, bead on plate welds were made in 1-4mm mild steel using both argon and helium shroud gases at a flow rate of 10 and 20 l/min (see Appendix 5). If the gas shielding is inadequate in CO_2 laser welding, the plume will be dominated by the effects of the more easily ionised iron vapour. Once the iron vapour becomes ionised, the plume begins to absorb more power from the beam, and thus becomes hotter and even more ionised. This cycle ensures that the laser welding process becomes inefficient, with the resultant weld beads having shallower penetration depths and usually wider weld widths. The results for CO laser welding 4mm and 3mm mild steel plate, using argon and helium shroud gases at a flowrate of 10 and 20l/min, are shown in figures 4.34 to 4.36. In all cases, the resultant weldbeads are deeper with an increase in gas shroud flowrate. This is because the shroud gas in this case has suppressed the iron vapour in the plume, with the effect that there is much less plasma blocking. This dependence upon gas flowrate indicates that the CO laser does create enough of a plasma to have an effect on the process under these conditions, however, the lack of a 'blue flash' would suggest that the vapour was nowhere near as ionised as in the CO_2 laser welding case. Considering the comparative CO/CO_2 trials at Heriot Watt and Liverpool universities, the threshold power for the creation of such a plasma is greater in the CO laser case.

CO_2 laser welding trials were carried out at Liverpool University using the same process conditions, in order to compare the capabilities of each laser. The comparative results are presented in figures 4.37-4.40. A number of points here are worthy of note. In all but one case, the CO laser can penetrate the 6mm mild steel sample at greater speeds than the CO_2 laser, because at these slow processing speeds too much of the CO_2 laser light is being absorbed in the plume, with the result that there is less power available in the keyhole. However, in the CO laser case, less plasma is absorbed, and hence more power is available to increase the penetration in the keyhole. It is significant that in most cases considered at these speeds, the top bead weld width is greater for CO_2 laser welding, illustrating the effects of the plasma in the plume. The discrepancies between the two lasers are more apparent when using argon as a shroud gas, illustrating the increased presence of a plasma in CO_2 welding with argon as a shroud gas. One can conclude therefore, that the CO laser offers advantages over the CO_2 laser when the plasma becomes uncontrollable in the CO_2 case.

4.3.3 Validation of the Mathematical Model for CO laser processing

In order to modify the model for CO laser processing, the value of ε was changed to 0.12, and following Glowaki et al (1994), the value of the linking intensity was set to $0.65*10^{-7}$ W/m^2. A similar study to the CO_2 case was carried out in order to obtain an estimation of penetration depth in CO laser welding. Here, after comparing the keyhole dimensions with experimental data, the

critical ratio between the top and bottom keyhole radii was found to be 0.33. Figure 4.41 shows the theoretically estimated values of penetration depths, with experimental results. Once again, it can be seen that the theoretical results tend to overestimate the maximum welding speed at lower powers, due to the effect of the plasma in the plume. However, they also overestimate the maximum welding speeds for thinner materials as well. Figure 4.42 shows a comparison between theoretically calculated maximum welding speeds for CO and CO_2 laser welding at powers of 2kW and 3kW, and with keyhole radii of 0.15mm and 0.2mm. As can be seen, there are two regions generated by the curves. The first is at lower thicknesses were the maximum welding speeds are relatively high. Under these conditions, the Fresnel absorption mechanism dominates, and therefore the CO laser, with its shorter wavelength and therefore enhanced Fresnel absorption, has the advantage over the CO_2 laser. However, at greater thicknesses, where maximum welding speeds are lower, the inverse Bremsstrahlung absorption mechanism within the plasma dominates the CO_2 laser welding process, but plays a negligible roll within the CO laser welding process, and therefore the CO_2 laser can weld at a faster speed for a given thickness. In reality this advantage is not realised in thicker plates as the plasma absorption becomes detrimental to the overall process. However, this is not taken into account in the model.

Figures 4.43 and 4.44 show the comparison between theoretically calculated and experimental obtained values for top and stem weld widths. As with the CO_2 laser case, the theoretical predictions can underestimate the experimental top bead weld widths by up to 50%. The difference in the comparisons is that in the CO laser case, the model also underestimates the stem weld widths as well. It has already been stated that there is some plasma present in the plume in CO laser welding at these power densities, and this, together with the Marangoni convection at the surface of the molten pool, will be the reason for the underestimation of the top bead weld width. The underestimation of the stem weld width is probably due to the fact that internal reflections are neglected, and that in the CO_2 laser case, this is compensated somewhat by the inverse Bremsstrahlung contribution. This problem though requires further investigation.

4.4 Summary

Experimental work on CO_2 laser welding was undertaken using a 5kW Ferranti AF5L laser, and an 2.2kW OPL laser in order to determine the effect of varying a number of process parameters on the laser welding process. These results were then compared with the theoretical predictions of the second model presented in chapter 3. From these experiments it was found that,

- The weld dimensions increase with decreasing speed. Under 10mm/s, the top weld width increases rapidly as the velocity decreases, indicating not only the effect of reduced translation velocity, but also the effect of the greater amount of ionisation in the plume at these speeds.

- Referring back to section 4.2.3, it can be seen that when using argon shroud gas at a particular speed above 10mm/s, there was little difference in the observed CO_2 laser generated weld widths within a powers range of 1 to 2.5 kW. However, when helium shroud gas was used there was a large difference in weld widths. It was concluded that this was due to the fact that even at lower powers, the argon plume is ignited, whereas the helium plume, with it's higher ionisation energy, is only partially ignited, an increase in power in the latter case will therefore increase the ionisation in the plume, and thus increase the weld width.

- A decrease in lens focal length produces an increase in power density at the workpiece. Intuitively, one would therefore expect the penetration depth to increase accordingly, however, three other factors that occur with the decrease in focal length also play an important part in the process. The first two are the defocussing effect above the workpiece together with the absorption of plasma within the plume, these effects will increase with increasing power density and decreasing speed (hence increasing beam-material interaction time), resulting in a larger spotsize at the workpiece. The third factor is the divergence of the beam down the depth of the keyhole. It was a combination of these three factors that lead to a decrease in penetration with a decrease in lens focal length when welding 6mm mild steel. These effects are less pronounced when welding thinner material, where the speeds are much faster, and the beam-material interaction time is much less. Here, welding with a shorter focal length lens gave deeper penetration depths in 3mm mild steel plate.

- The role of the shroud gas is to control the amount of plasma within the plume by displacing the more easily ionised metal vapour. At high powers and slow speeds, the shroud gas fails to control the plasma, and so the beam defocussing above the keyhole and the absorption of light within the plume becomes detrimental to the process as a whole, resulting in shallower penetration depths and wider weld widths. This effect occurs first in argon, as it has a lower ionisation energy, and produces a plasma much more easily. Thus, when welding with higher powers, especially at lower speeds, welds with argon shroud produce shallower penetration than those in helium shroud.

- The mathematical model was compared to experimental CO_2 laser welding results, and it was found that the model gave a good approximation for the penetration depth for most cases, although with thicker materials at slower speeds, the model tended to overestimate this value due to the fact that it neglects the effects within the plasma in the plume.

- Due to the fact that the model neglected the heat source due to the plasma above the workpiece, it's defocussing effect on the laser beam, and the Marangoni convection at the weldpool surface, the values of weld widths were underestimated by values between 20-50%.

- In order to assess the effect of the heat source in the plume upon the laser welding process, the plume was simulated by placing a varying point source at the top of the keyhole and modifying the heat balance at the keyhole walls. The main contribution that the addition of the point source made was to open the keyhole mouth. The value of weld width increases therefore at the top of the keyhole, but decreases at the bottom due to the fact that the presence of the point source blocks a fraction of the beam, decreasing the power available in the keyhole.

- The effect of the lens focal length was modelled by reformulating the Fresnel contribution to the heat balance at the keyhole walls, in order to take into account the decreasing power density down the depth of the material (neglecting multiple reflections, and assuming that the beam is parallel to the direction of the axis). The process was more dominated by the size of the beam radius, and so the penetration depth obtained increased with decreasing focal length. As the workpiece thickness increases and as a result the maximum welding speed becomes slower, the inverse Bremsstrahlung contribution dominates the welding process and the difference between the penetration depths obtained with different lenses decreases beyond comparison.

Experiments were then carried out in the laser welding of mild steel using a CO laser, which operates at just over half the wavelength of the CO_2 laser, and compared to welding results obtained from a CO_2 laser. Theoretically, the decrease in wavelength would ensure that the linking intensity and hence the plasma absorption contribution would decrease, and that for most angles of incidence, the Fresnel absorption coefficient would increase. The decreased plasma absorption would suggest that the effect of defocussing in the plume would be reduced, and hence the CO laser could offer advantages at slower speeds and higher powers, however, the beneficial effects of the plasma absorption would be less. From these trials it can be concluded that,

- In low power comparative CO/CO_2 welding trials, the increased plasma absorption of the CO_2 laser light when using helium shroud gas at any flowrate and argon shroud gas at high flow rates, increased the coupling into the workpiece and resulted in larger penetration depths. When using an argon shroud gas at low values of flow rate however, the plasma in the plume could not be controlled in the CO_2 laser case, and so the welds generated with the CO laser resulted in greater penetration depths.

- There is little to distinguish the weld dimensions generated by a CO laser beam with either helium or argon as a shroud gas, under the process conditions studied here, even at higher powers. This is due to the fact that there is not as much ionisation and absorption in the plume in this case, and so the Fresnel absorption at the keyhole walls becomes the dominant factor in the process. This is independent of the shroud gas.

- Using higher powers, there is little to distinguish between the two lasers at medium to fast welding speeds, however as the process speed became slower, welding with a CO_2 laser using argon shroud gas results in wider weld widths and shallower penetration depths, due to the problems associated with the plasma absorption in the plume. Using a CO laser results in larger penetration depths than even CO_2 welds in helium shroud gas, although the difference here was less pronounced.

- The comparisons between the CO and CO_2 laser welding trials underline a crucial aspect of laser processing. In order for good laser welding to occur, the plasma in the plume has to be generated and controlled. If the plasma is not generated, as in the low power CO laser experiments, enhanced coupling of the laser power into the workpiece will not occur, causing a reduction in penetration depth and weld width with comparable experiments where a plasma is generated. If the plasma cannot be controlled, no matter what the shielding gas conditions are, as in the case of very low speed welding, then reduction in penetration depths will occur, and in some cases, the process may cease altogether. This is where the CO laser offers its main advantage, it can still process in a region were laser welding becomes difficult for a CO_2 laser.

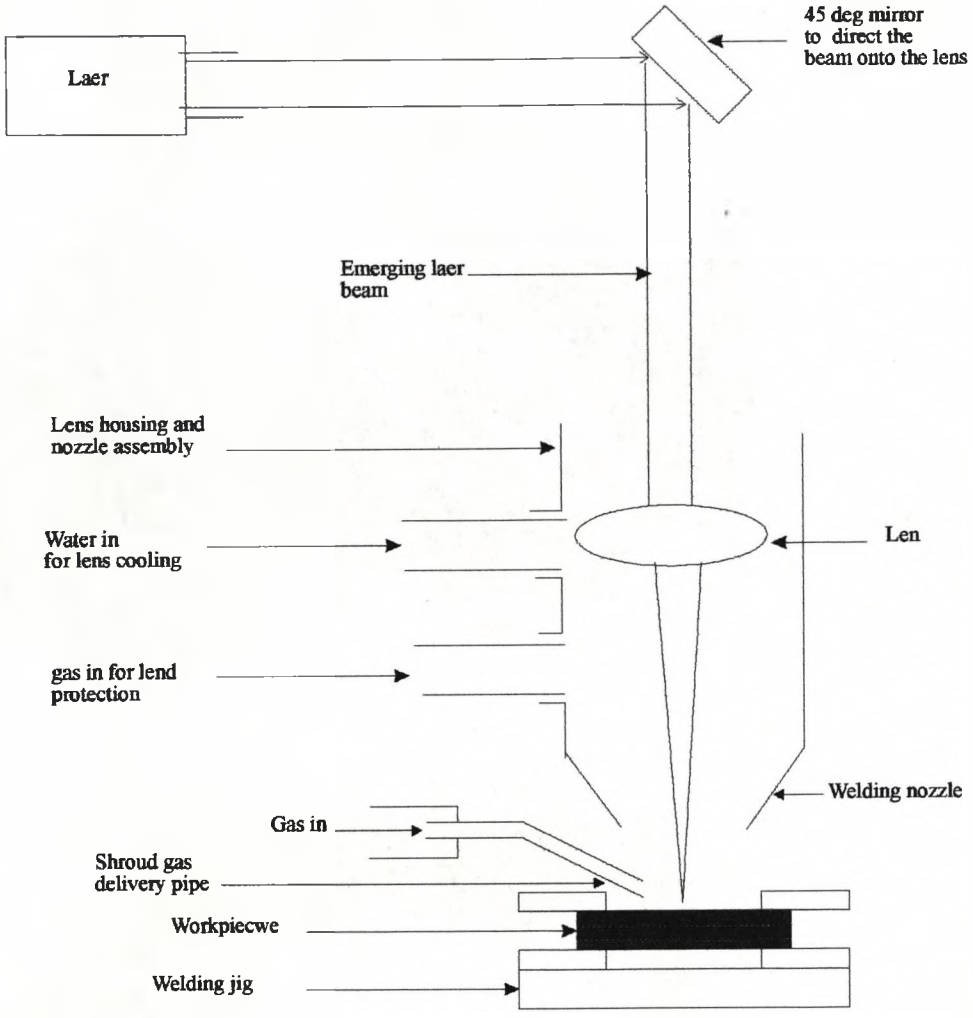


Fig.4.1 Schematic diagram of the experimental arrangement for lens focusing experiments.

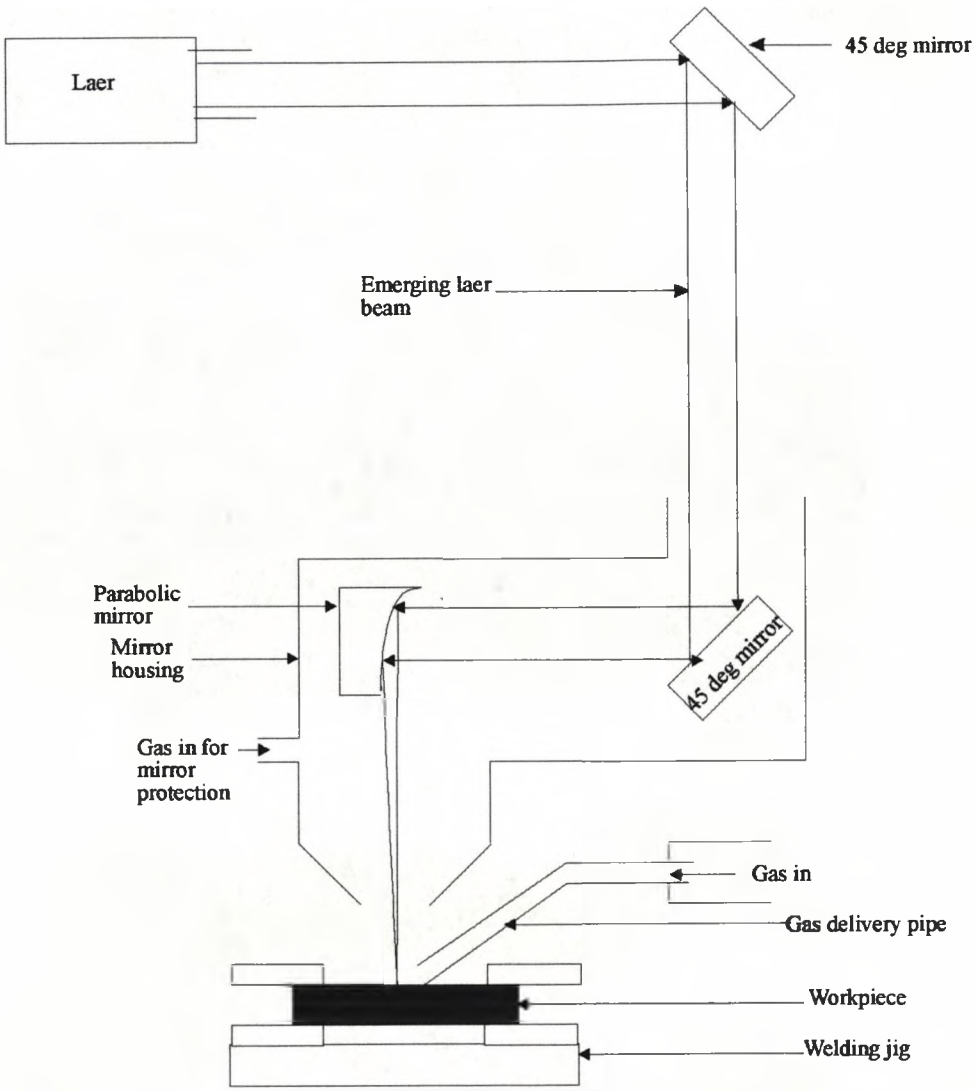


Fig.4.2 Schematic diagram of the experimental arrangement when using mirror focusing experiments at TWI.

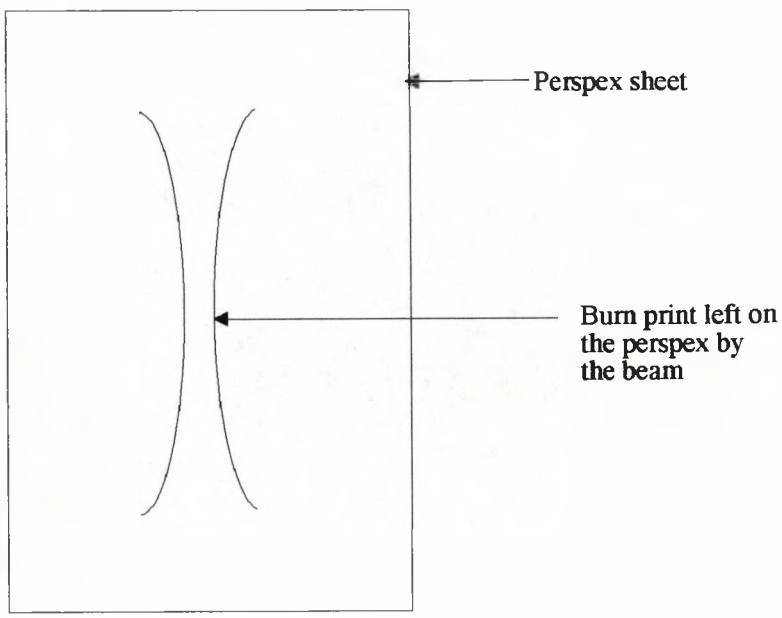


Fig.4.3 Schematic diagram of a burn print made when a perspex sheet was passed rapidly under a laser beam at a constant angle to the horizontal in order to determine the size of the beam at the focus.

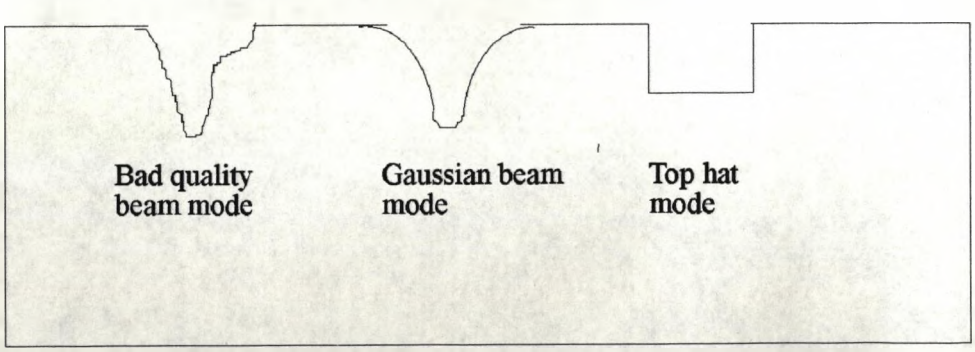


Fig.4.4 Schematic diagram of a cross section of various beam prints made in perspex.

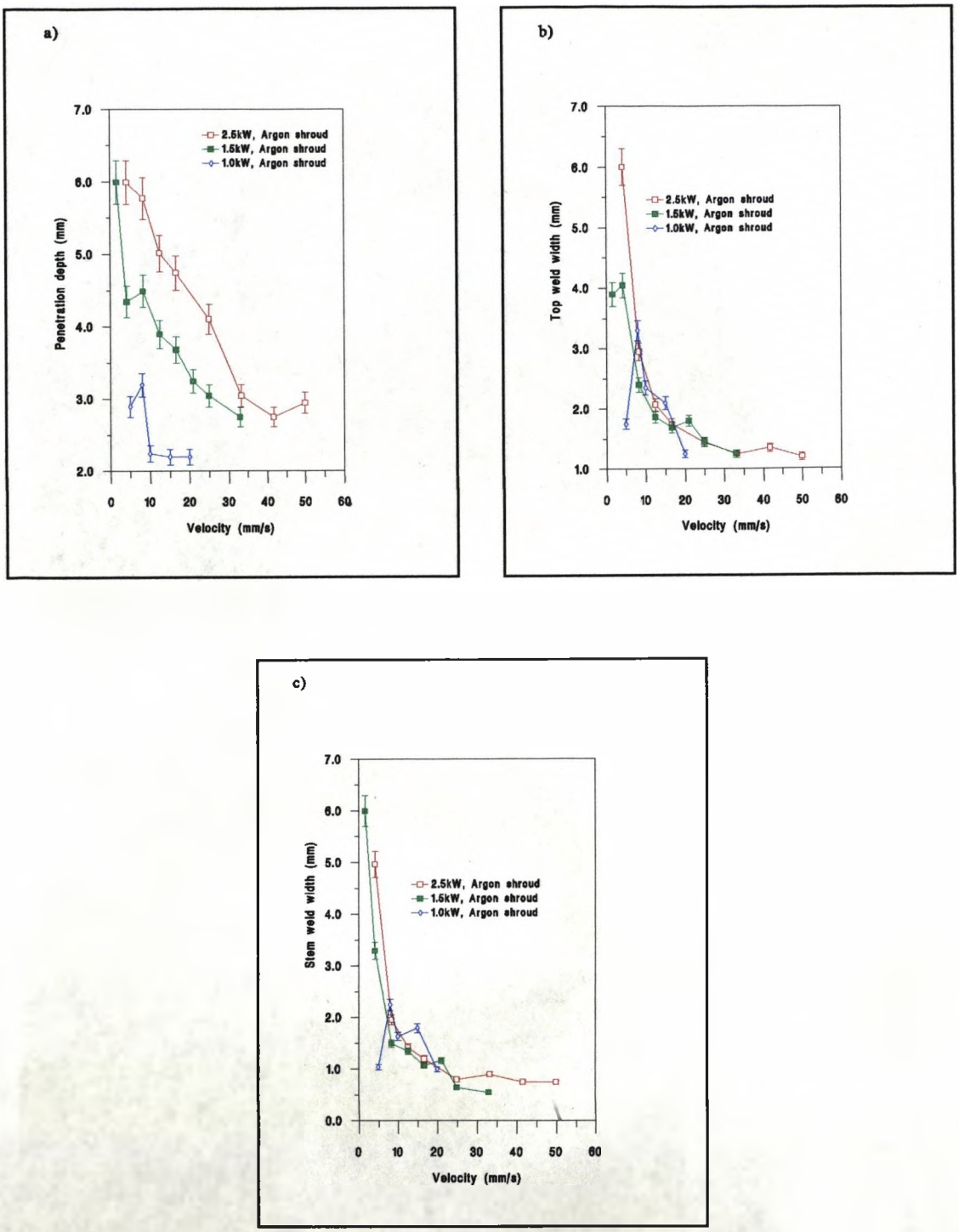


Fig 4.5 Penetration depth, top weld width, and stem weld width for bead on plate welds in 6mm mild steel, using a 150mm lens focal length and argon shroud gas.

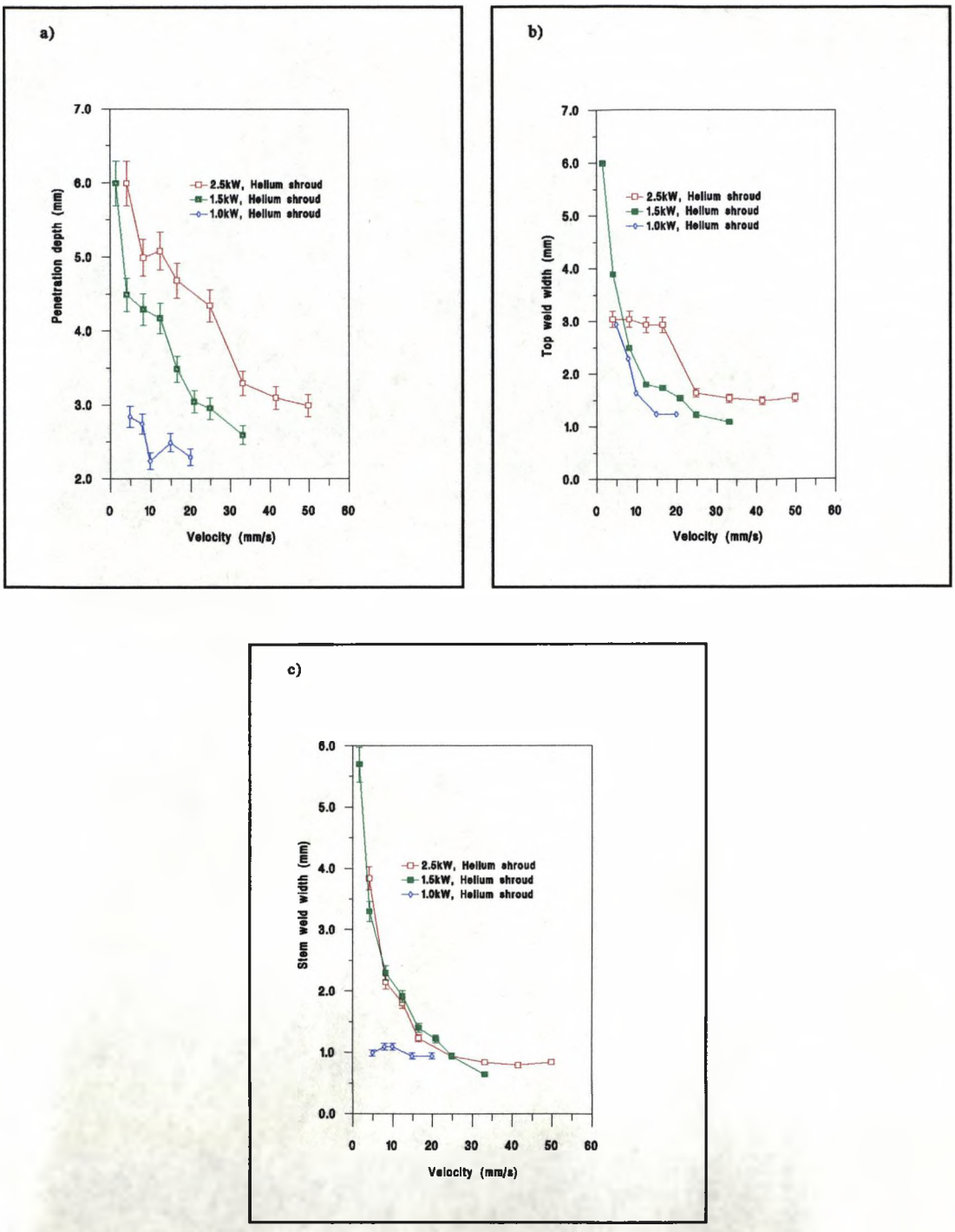


Fig.4.6 Penetration depth, top weld width, and stem weld width for bead on plate welds in 6mm mild steel, using a 150mm focal length lens and helium shroud gas.

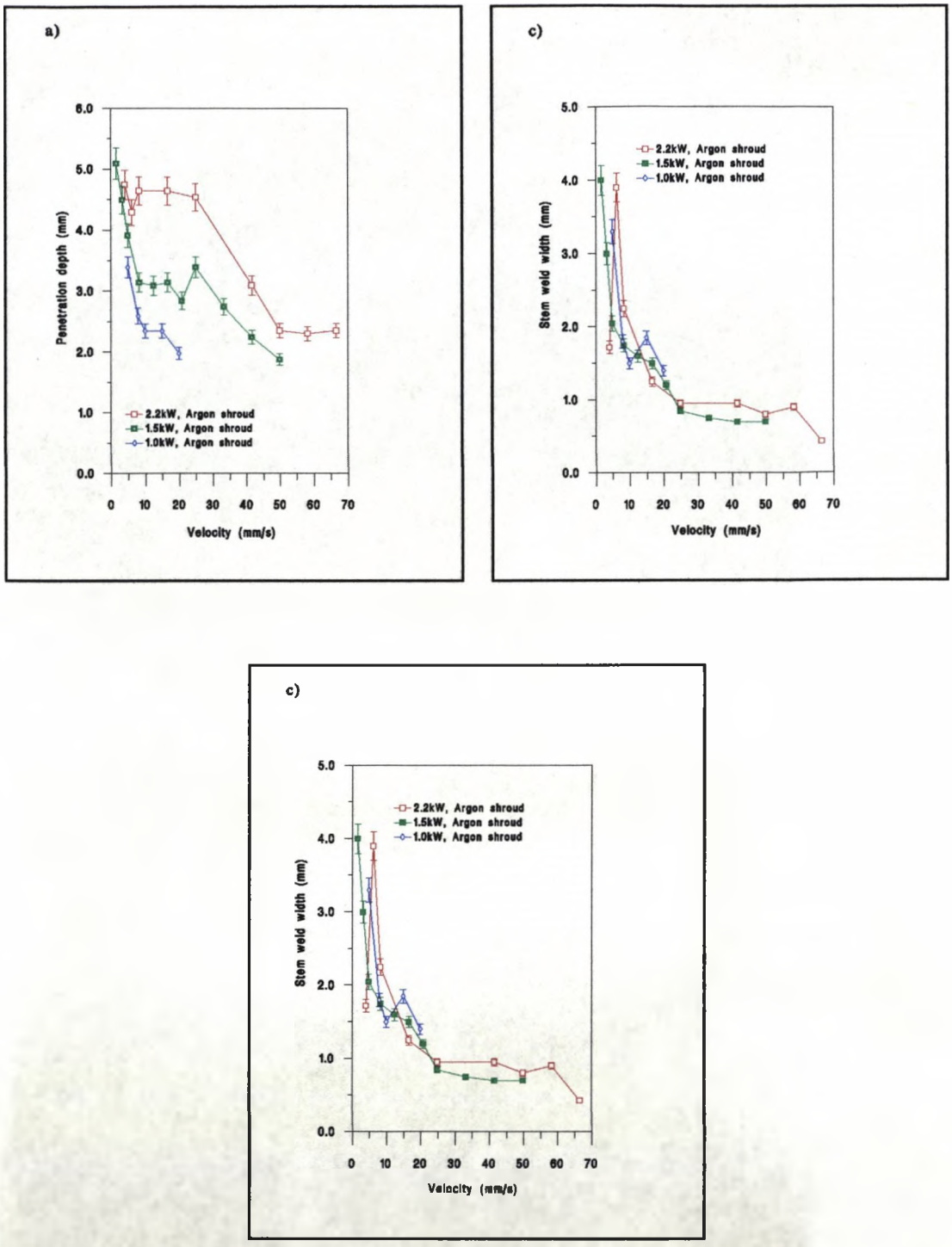


Fig.4.7 Penetration depth, top weld width, and stem weld width for bead on plate welds in 6mm mild steel, using a 100mm lens focal length and argon shroud gas.

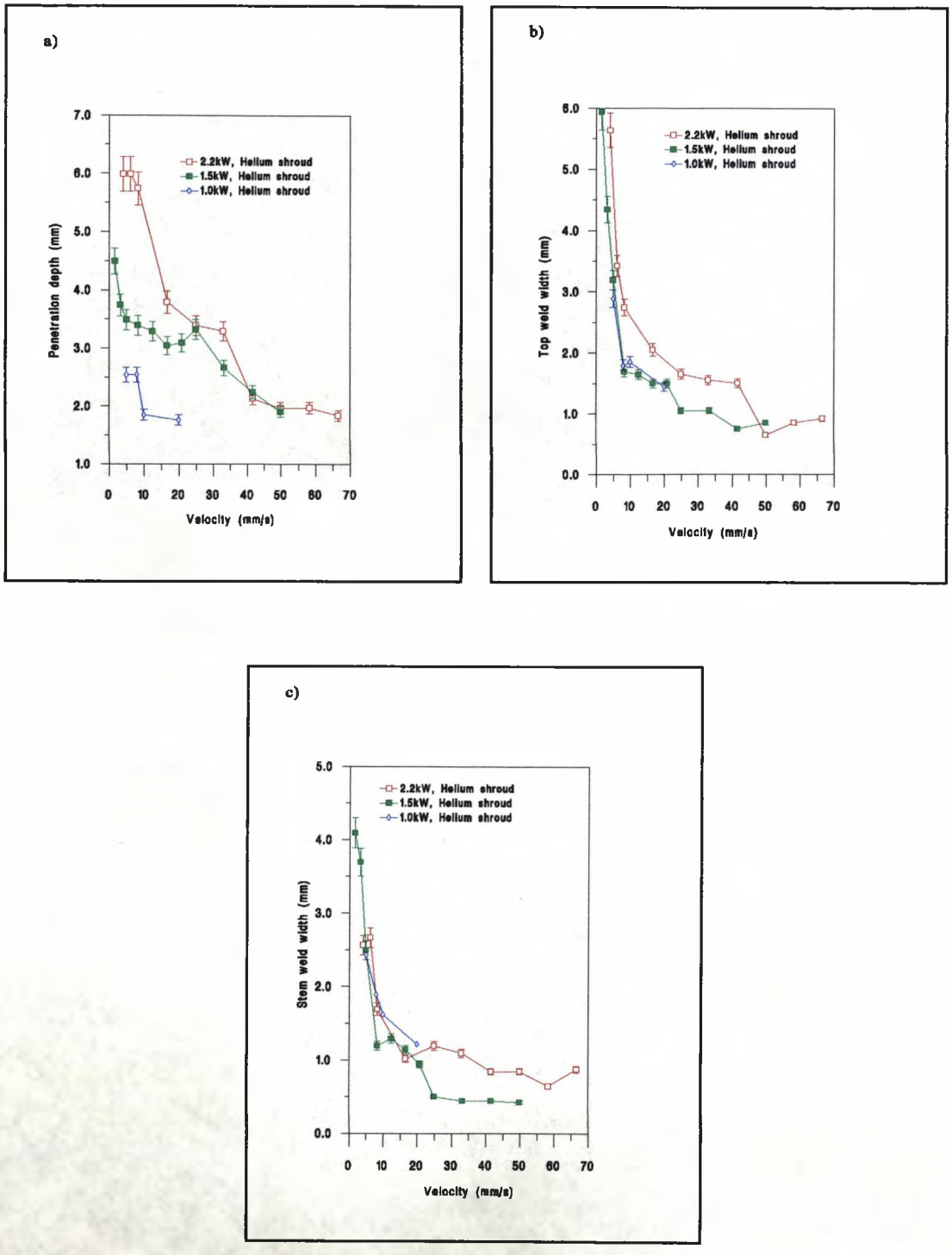


Fig.4.8 Penetration depth, top weld width, and stem weld width for bead on plate welds in 6mm mild steel, using 100 mm lens focal length and helium shroud gas.

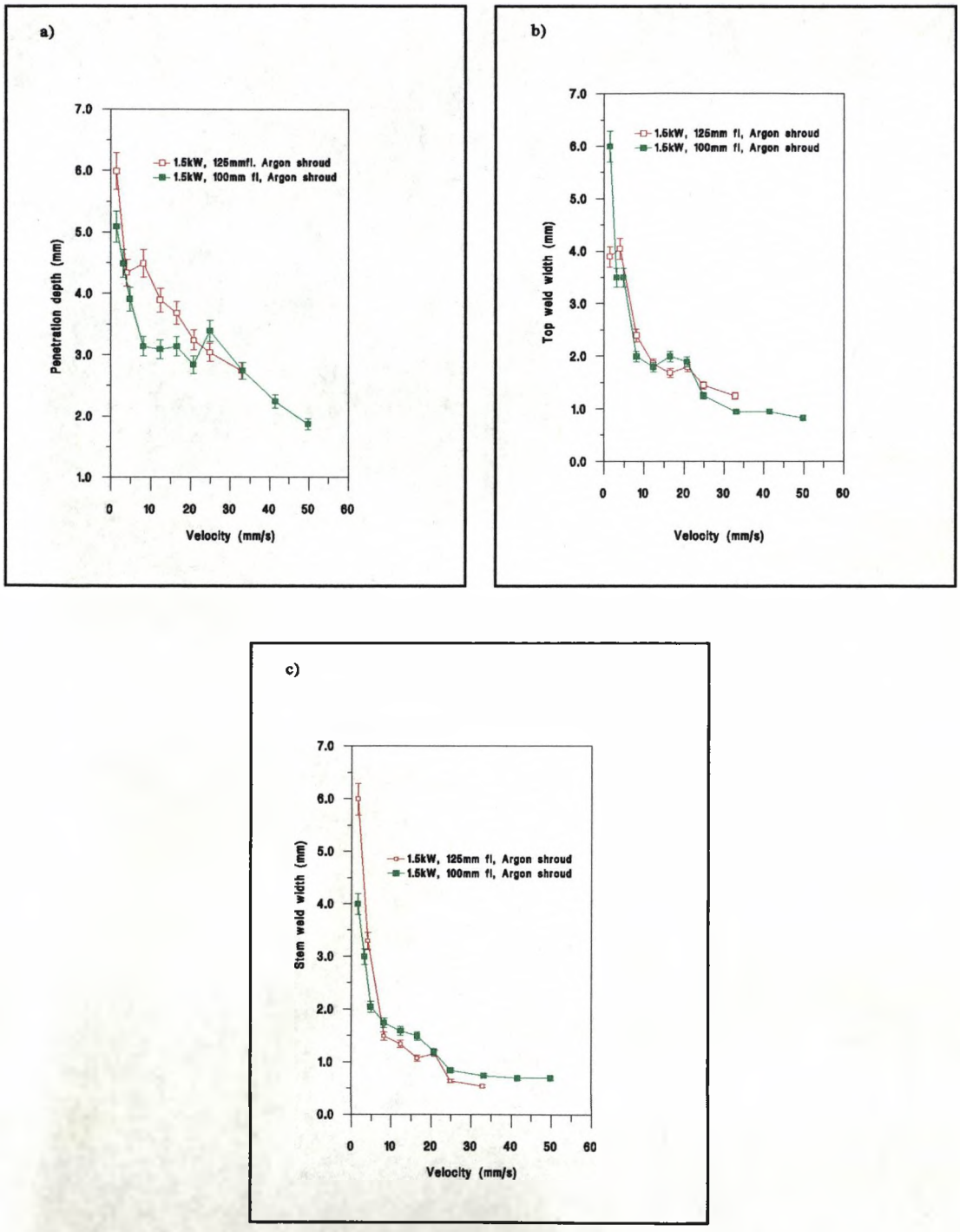


Fig.4.9 Penetration depth, top weld width, and stem weld width for bead on plate welds in 6mm mild steel at a power of 1.5kW, using a 150mm lens focal length and argon shroud gas.

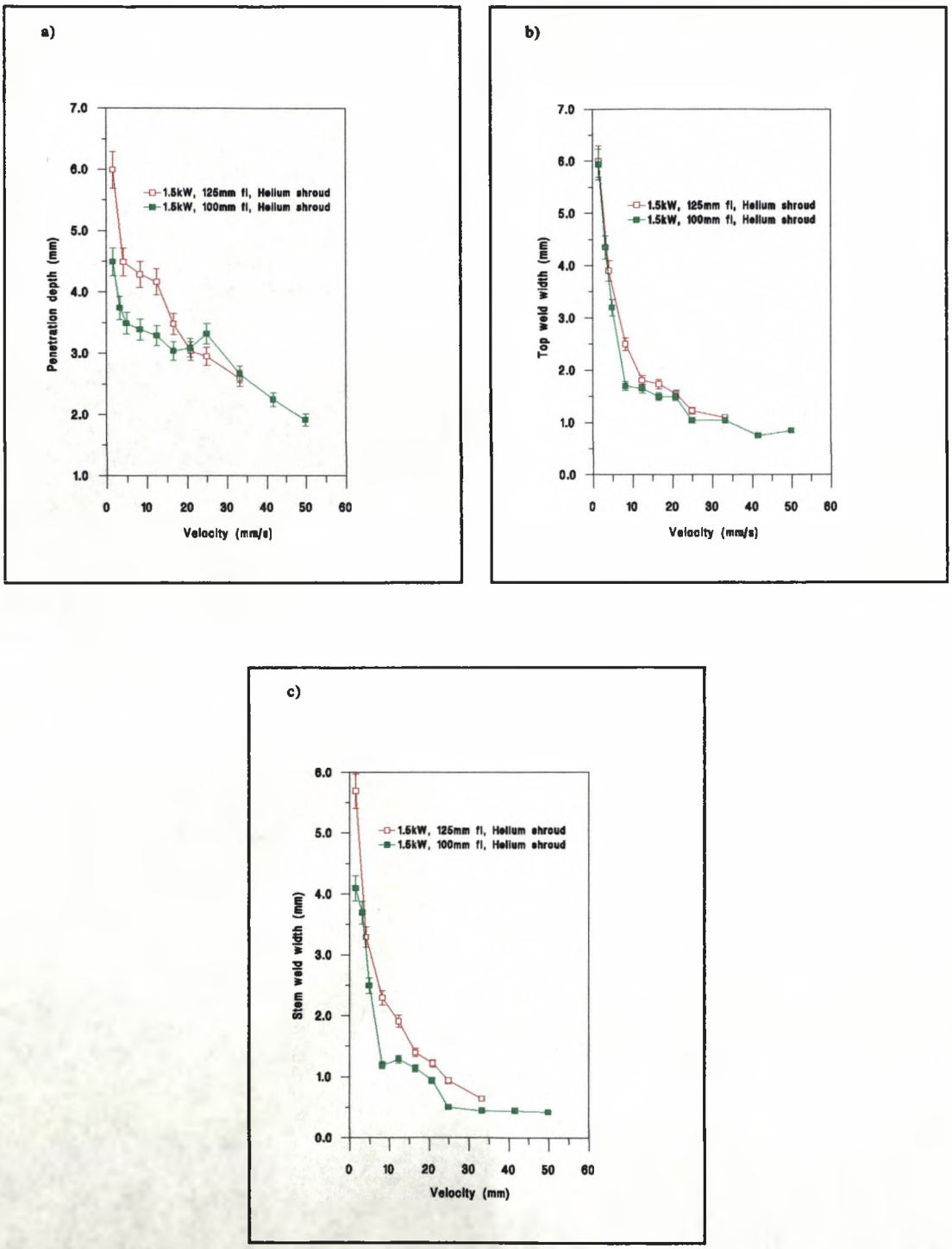


Fig.4.10 Penetration depth, top weld width, and stem weld width for bead on plate welds in 6mm mild steel, using a 150mm lens focal length and helium shroud gas at a power of 1.5kW.

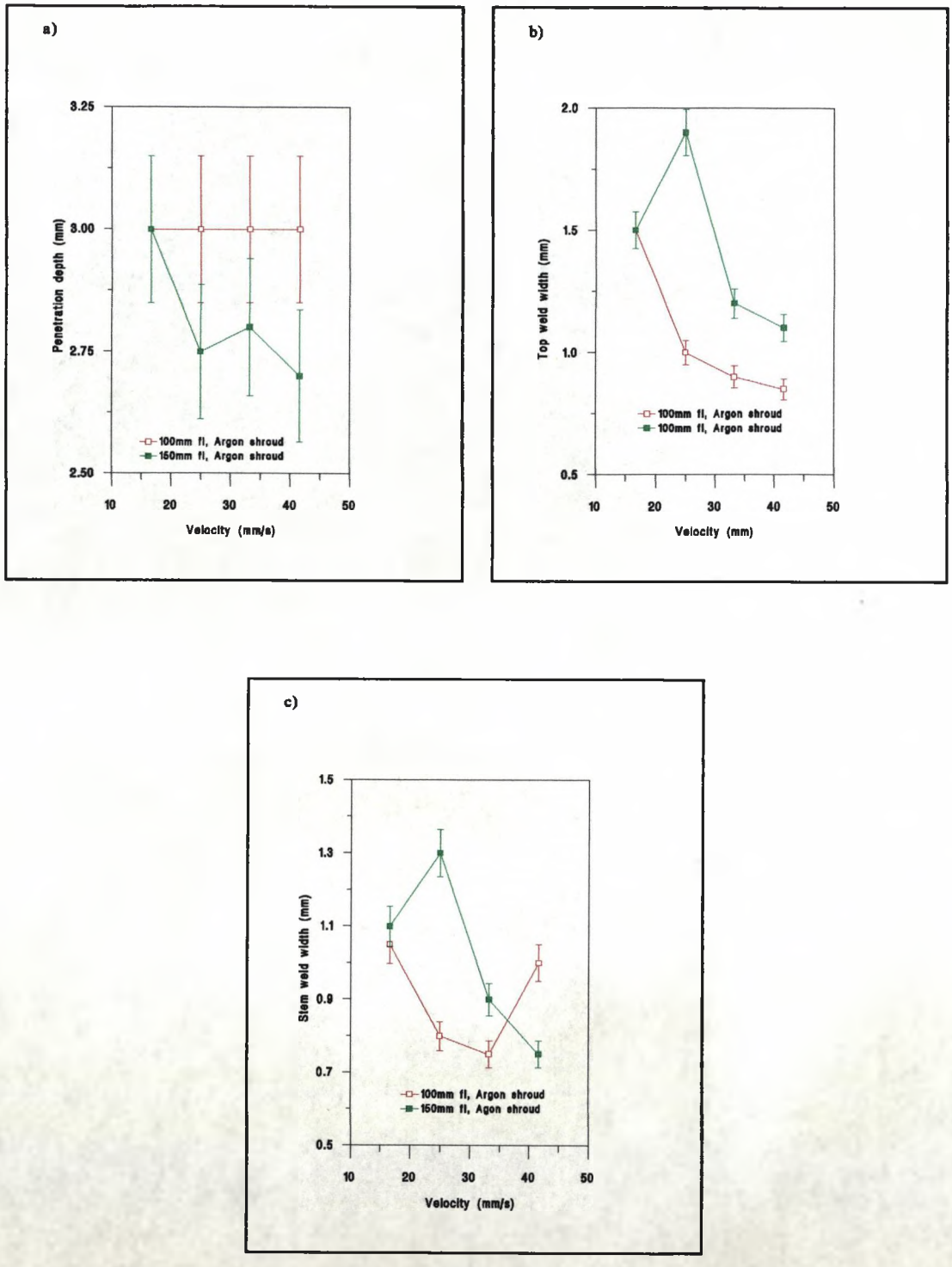


Fig.4.11 Penetration depth, top weld width, and stem weld width for bead on plate welds in 3mm mild steel at a power of 2.0kW, using a 100mm and 150mm lens focal length and argon shroud gas.

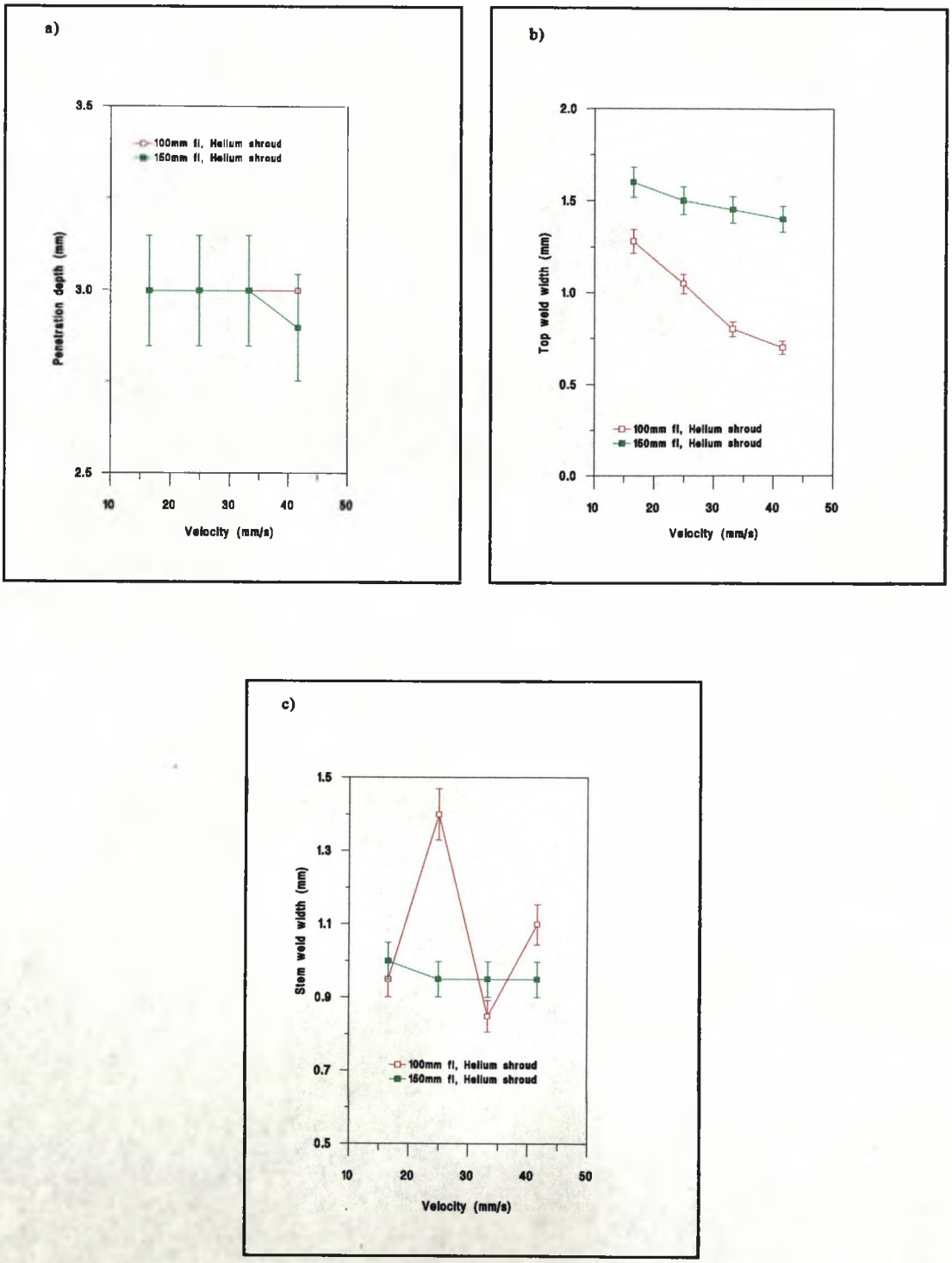


Fig.4.12 Penetration depth, top weld width, and stem weld width for bead on plate welds in 3mm mild steel at a power of 2.0kWl, using a 100mm and 150mm lens focal length of lens and helium shroud gas.

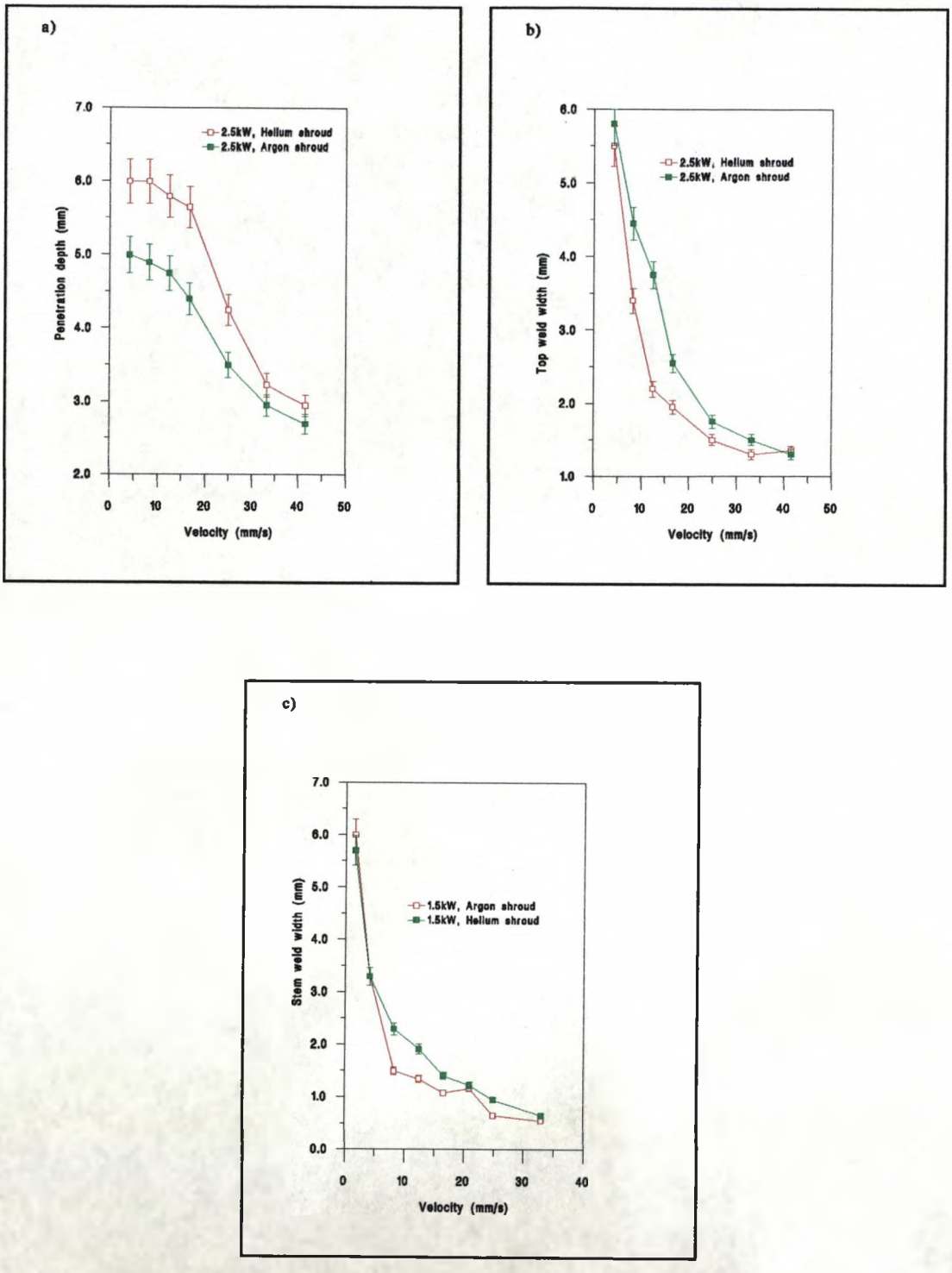


Fig 4.13 Penetration depth, top weld width, and stem weld width for bead on plate welds in 6mm mild steel at a power of 2.5kW, using a 150mm lens focal length and argon and helium shroud gas.

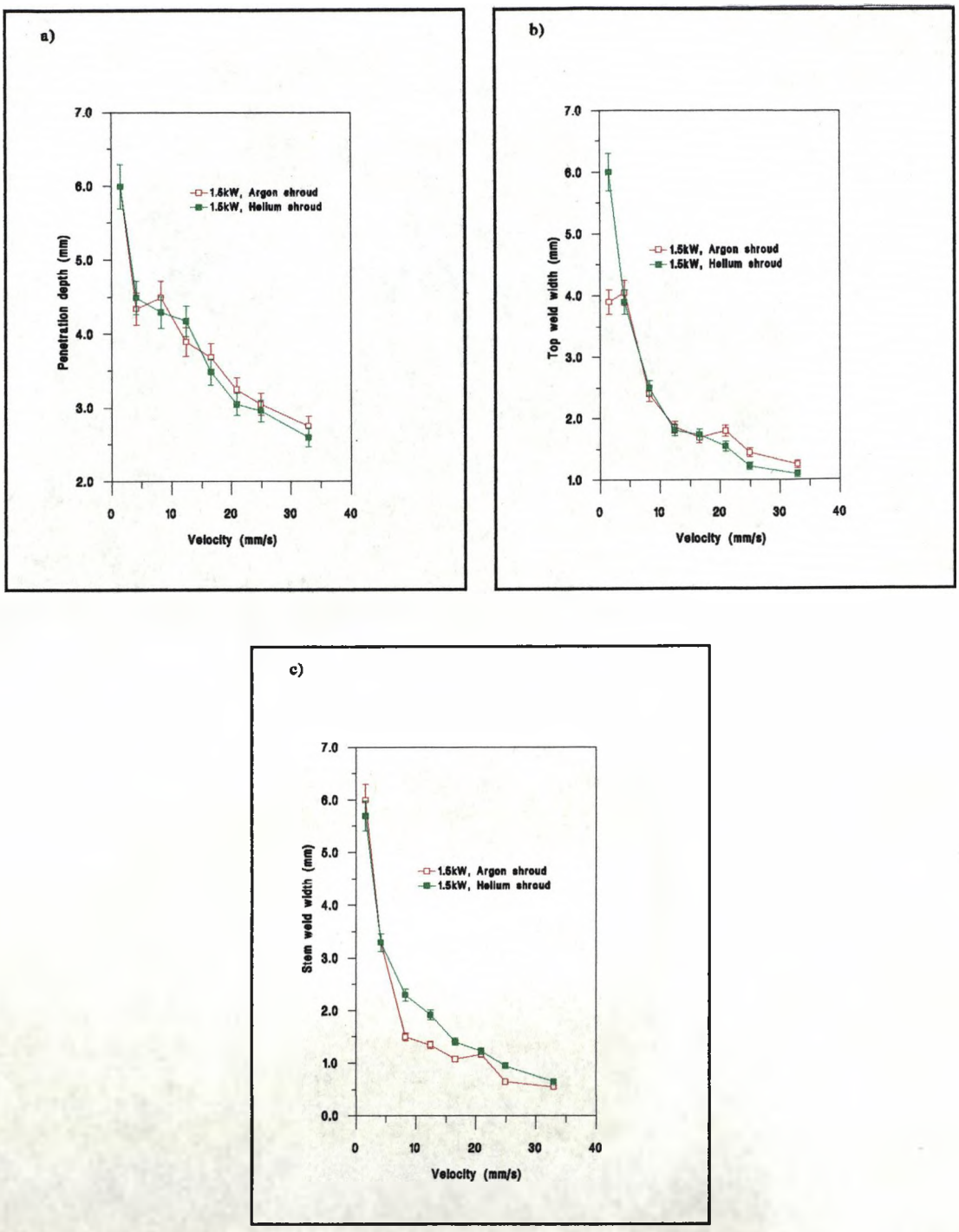


Fig 4.14 Penetration depth, top weld width, and stem weld width for bead on plate welds in 6mm mild steel at a power of 1.5kW, using 150mm lens focal length and argon and helium shroud gas.

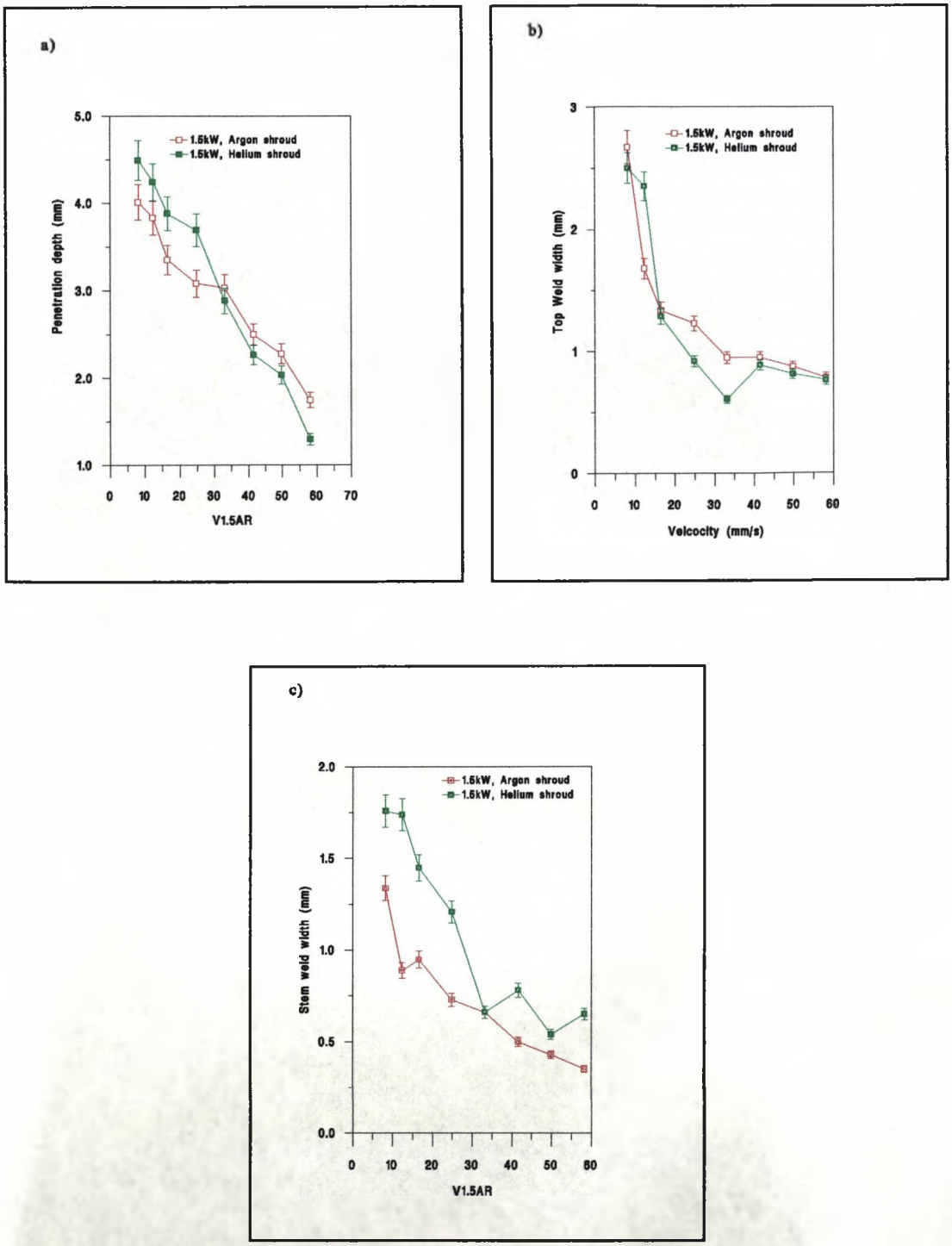


Fig 4.15 Penetration depth, top weld width, and stem weld width for bead on plate welds in 5mm mild steel at a power of 1.5kW, using a 150mm focal length lens and argon and helium shroud gas.

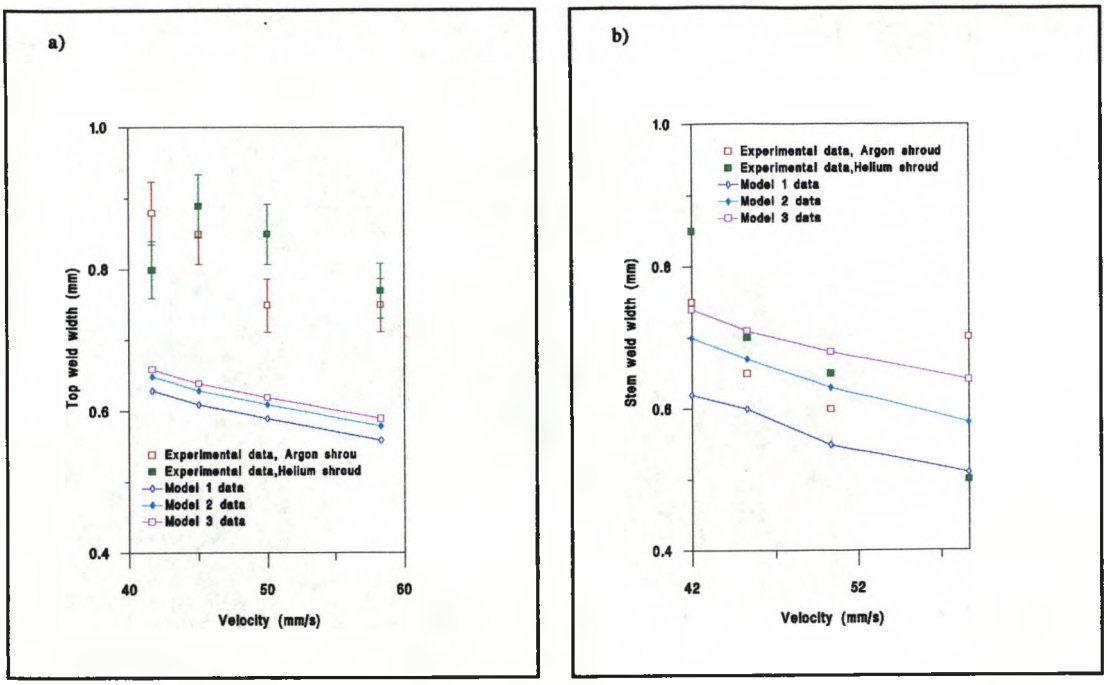


Fig.4.16 Comparison between theoretically calculated and experimentally observed top and stem weld widths using a spotsize of 0.3mm, and a power of 1.65kW on 2mm mild steel plate.

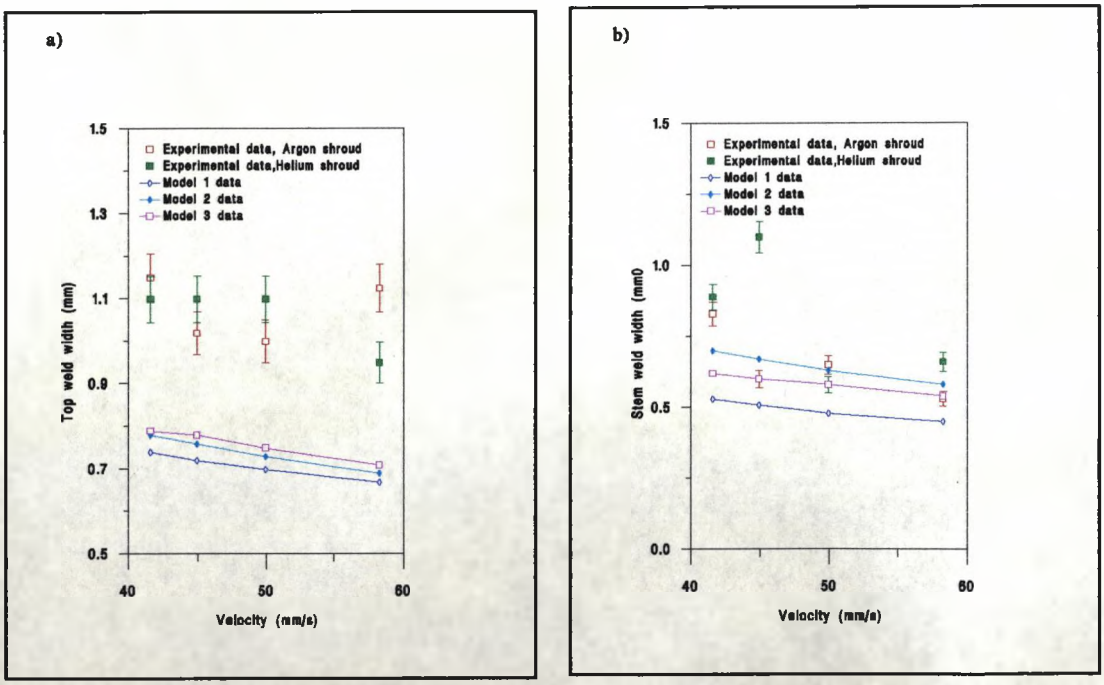


Fig.4.17 Comparison between theoretically calculated and experimentally observed top and stem weld widths using a spotsize of 0.4mm, and a power of 1.65kW on 2mm mild steel plate.

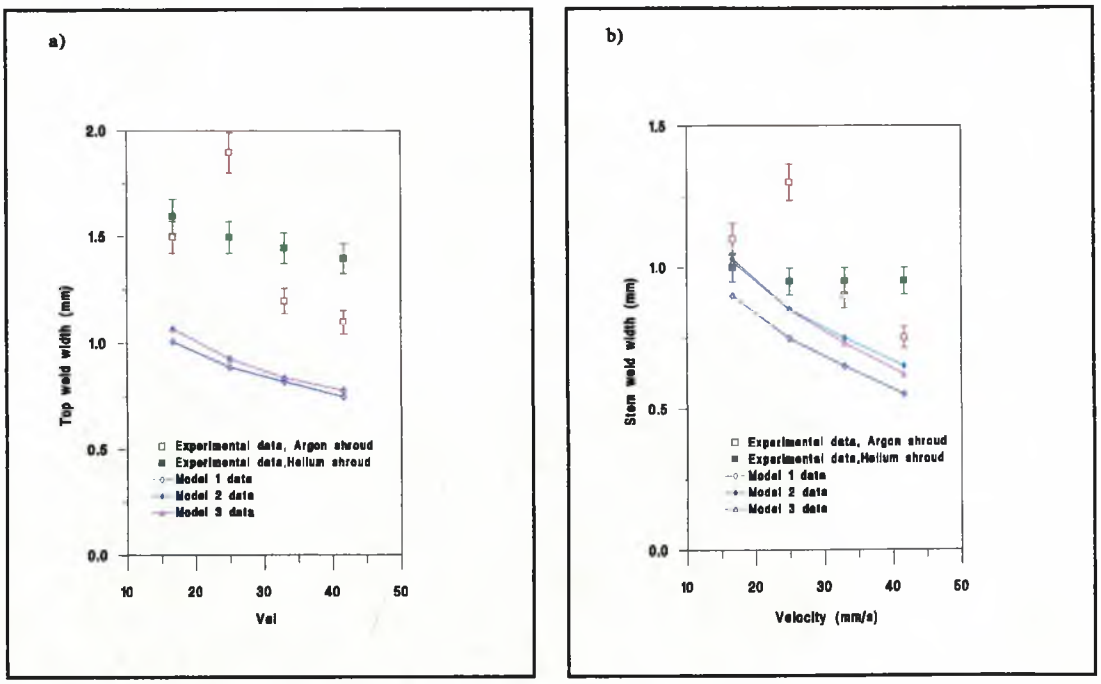


Fig.4.18 Comparison between theoretically calculated and experimentally observed top and stem weld widths using a spotsize of 0.4mm, and a power of 2.0kW on 3mm mild steel plate.

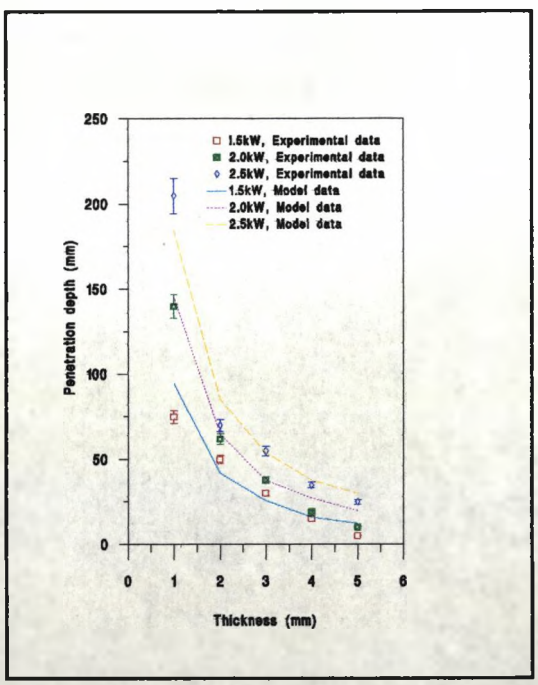


Fig.4.19 Graph comparing the theoretically calculated and experimentally observed penetration depths in mild steel using 1.5-2.5kW.

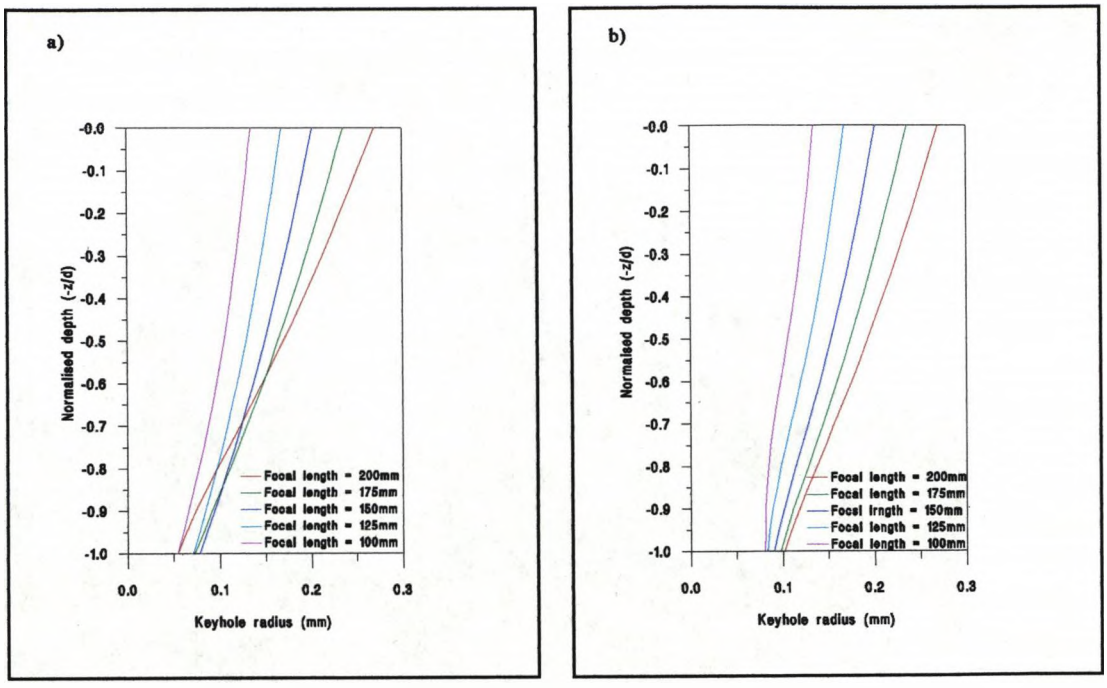


Fig.4.20 Graphs showing the calculated keyhole profiles using different focal length lenses a) using a power of 2.0kW, a thickness of 2mm, and a velocity of 50mm/s, and b) using a power of 2.0kW, a thickness of 4mm, and a velocity of 20mm/s.

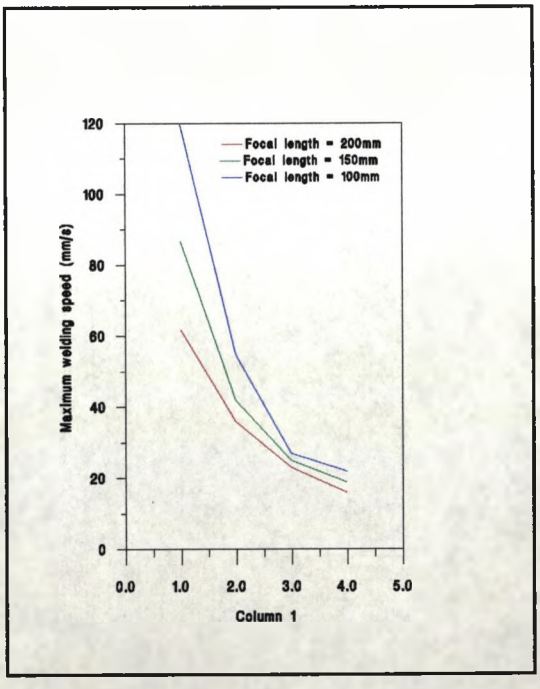


Fig.4.21 Graph showing the variation in maximum welding velocity versus plate thickness for focal lengths of 100mm, 150mm and 200mm.

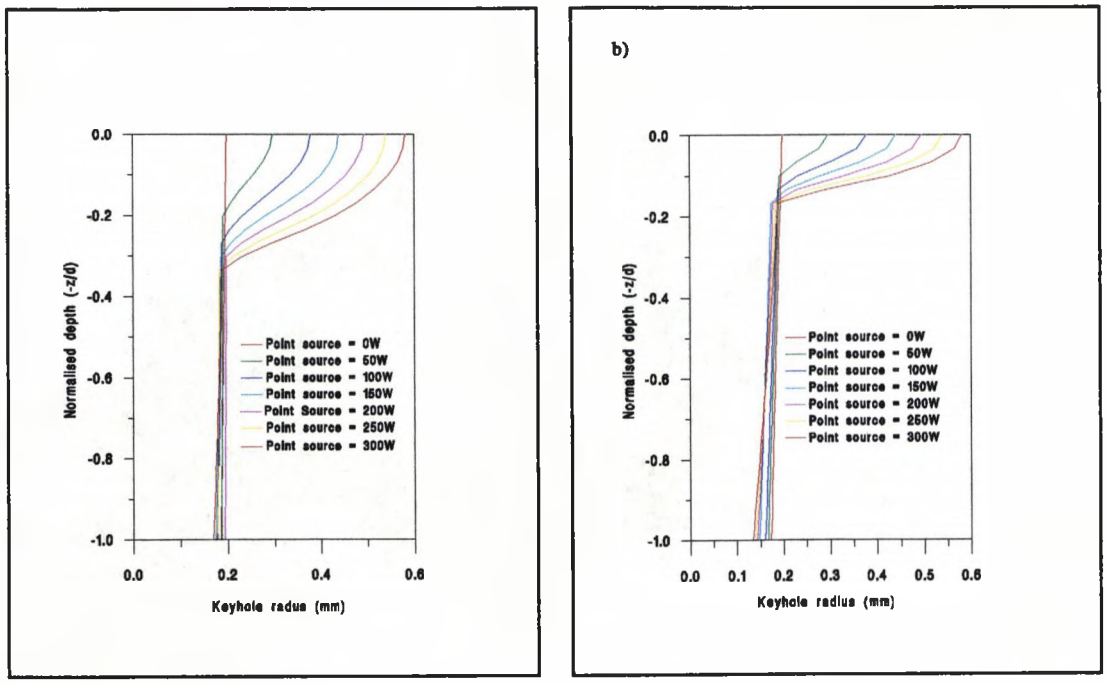


Fig.4.22 Graph showing the effect in theory of a varying point source upon the keyhole radius for 2mm and 4mm mild steel plate respectively; here the power is set to 2kW, and the beam radius to 0.2mm.

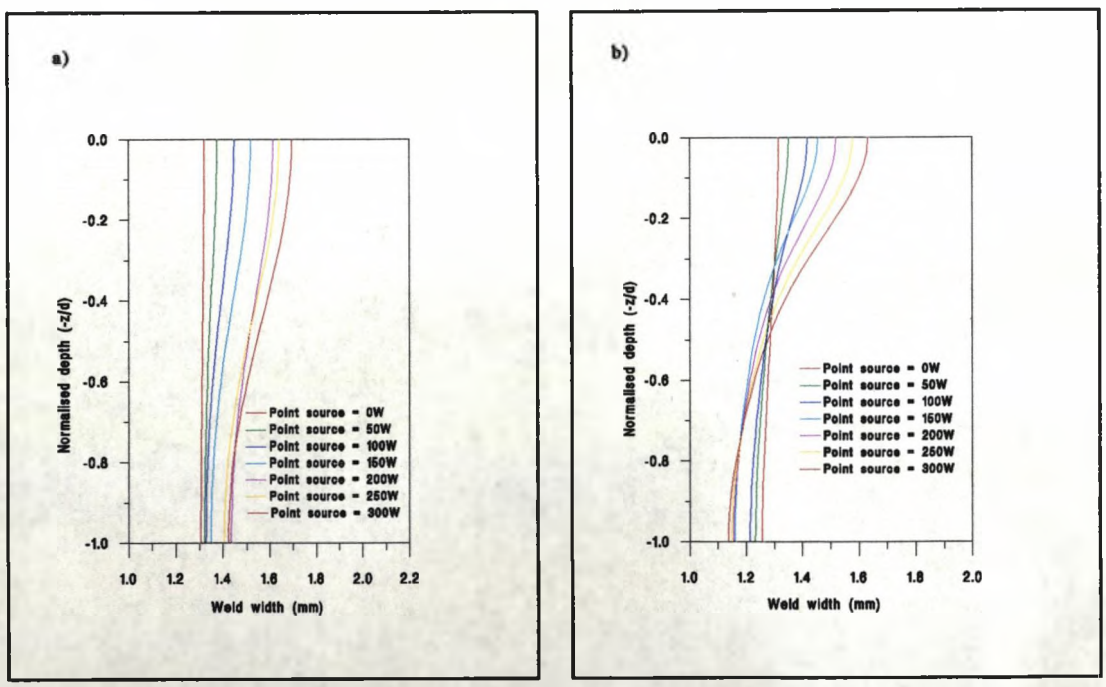


Fig.4.23 Graph showing the effect in theory of a varying point source upon the weld width for 2mm and 4mm mild steel plate respectively; here the power is set to 2kW, and the beam radius to 0.2mm.

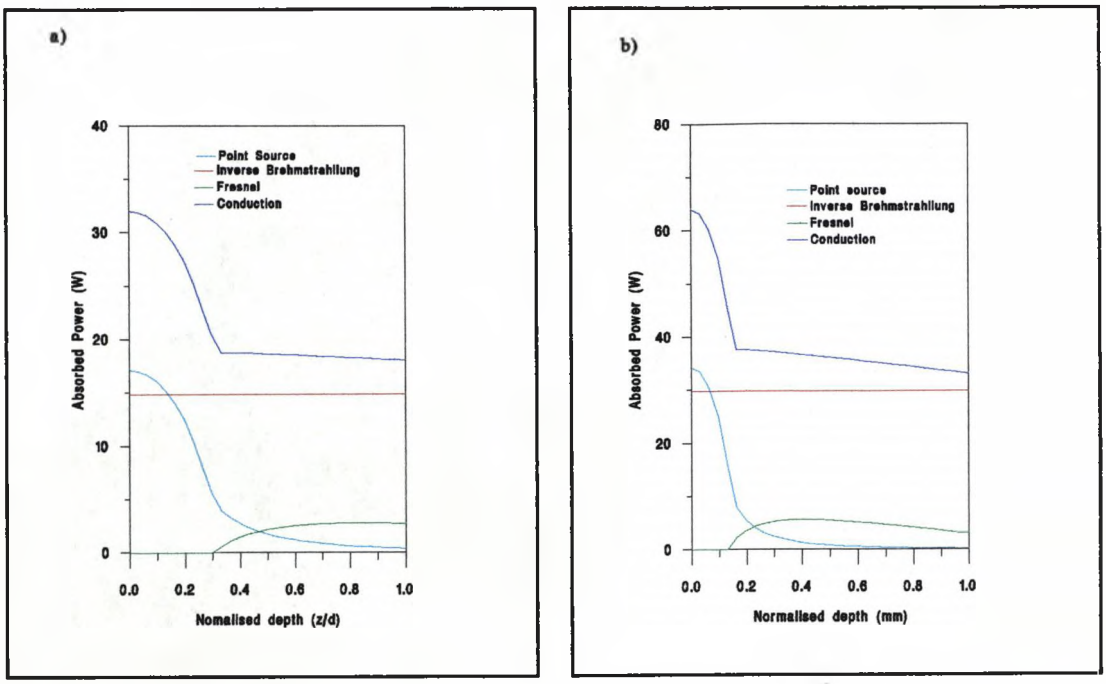


Fig 4.24 Power absorbed by the workpiece from the point source, and from the Fresnel and Inverse Bremsstrahlung mechanisms, as well as the power used up in conduction into the material, for 2mm and 4mm mild steel plates using a point source of 300W.

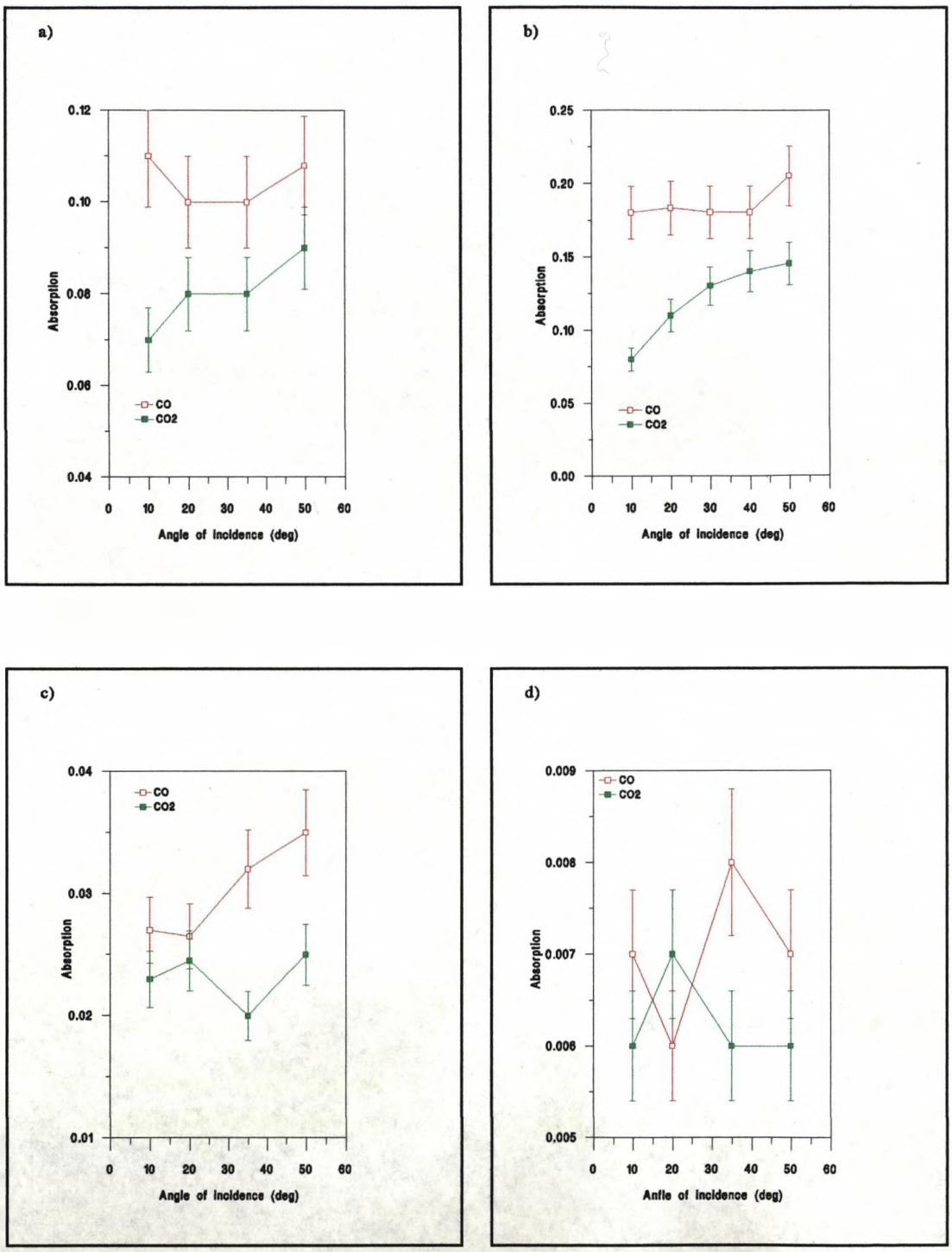


Fig.4.25 Experimentally determined absorptivities of stainless steel, titanium alloy Ti4Va6Al, aluminium alloy, Al 6061 and copper respectively, for both CO and CO_2 laser radiation.

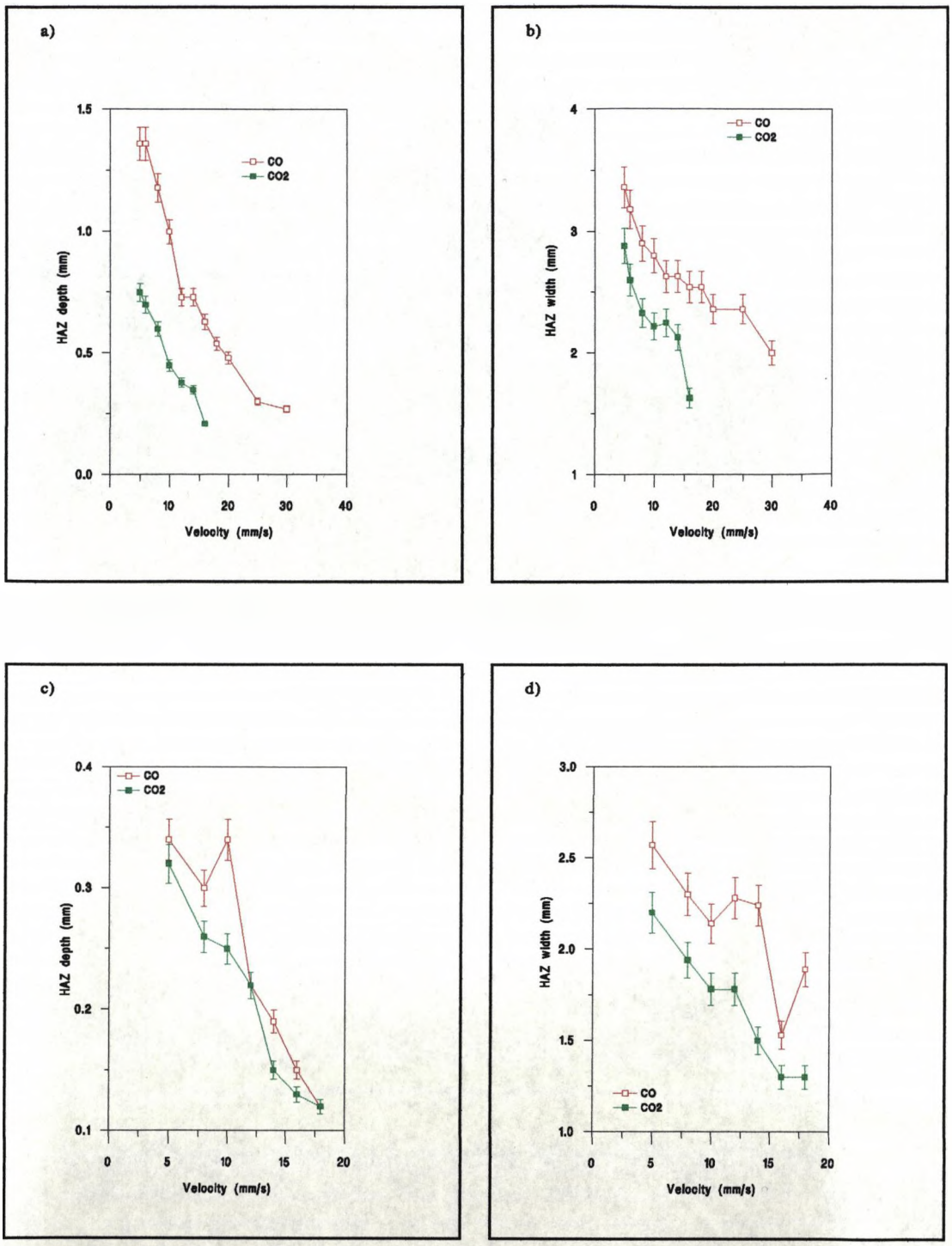


Fig.4.26 Results from comparative CO/CO_2 laser surface treatment trials using a,b) a power of 530W on mild steel, and c,d) a power of 580W on stainless steel.

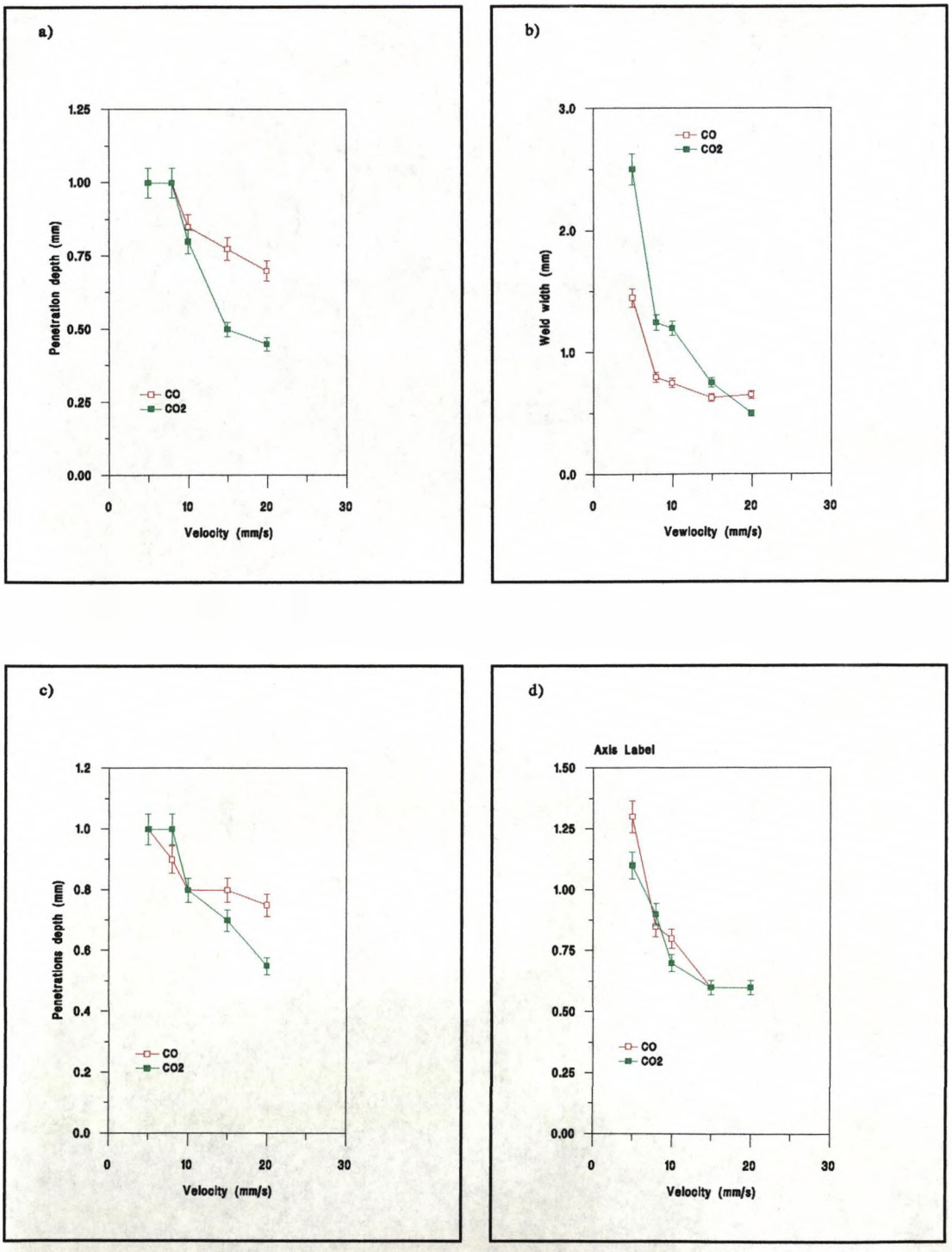


Fig.4.27 Results from comparative CO/CO_2 laser welding trials of 1mm mild steel plate a,b) using argon shroud gas at 10 l/min and a power of 670W, and c,d) using helium shroud gas at 10 l/min and a power of 600W.

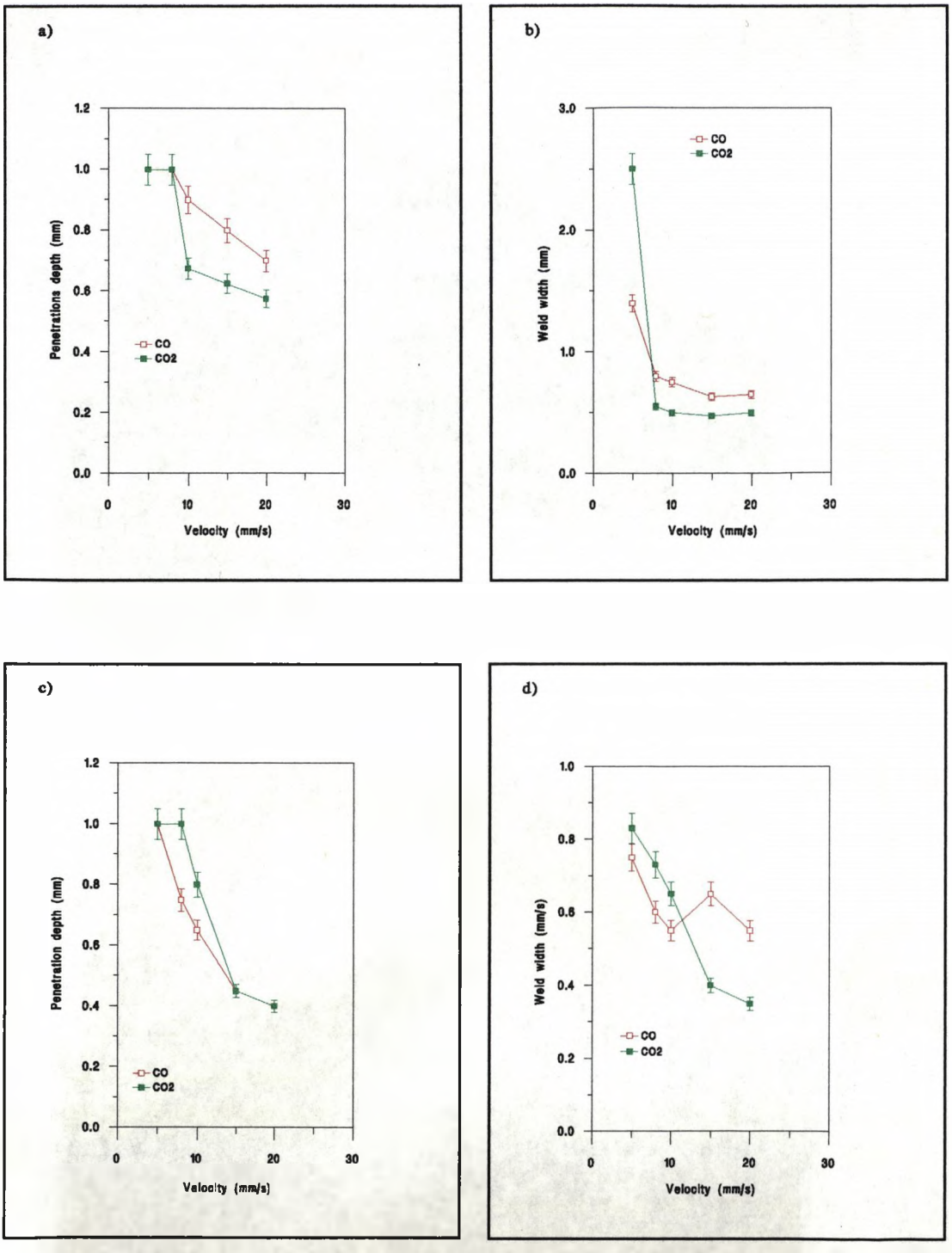


Fig.4.28 Results from comparative CO/CO_2 laser welding trials of 1mm mild steel plate a,b) using argon shroud gas at 10l/min and a power of 670W, and c,d) using helium shroud gas at 10 l/min and a power of 600W.

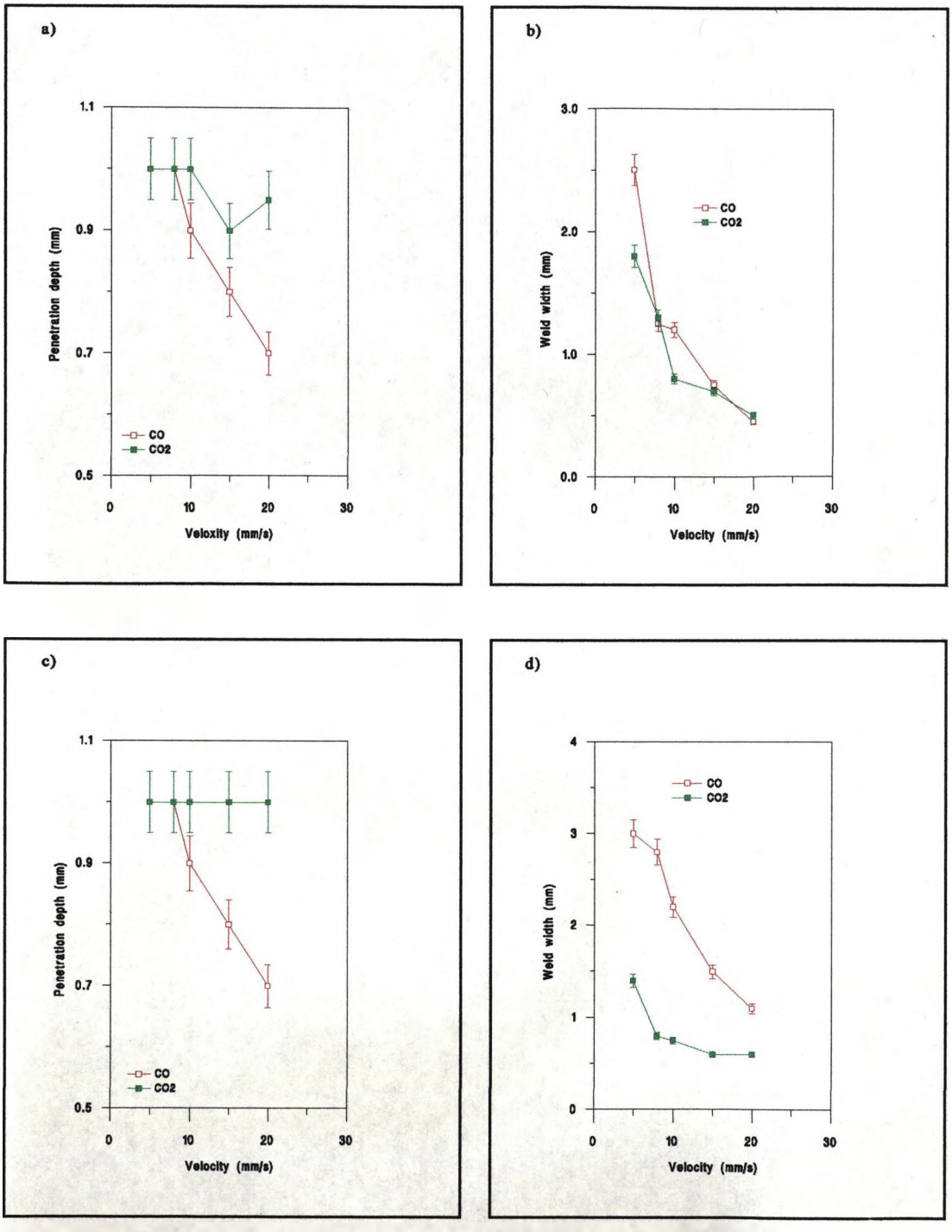


Fig. 4.29 Results from comparative CO/CO_2 laser welding trials of a,b) mild steel plate at a power of 600W, and c,d) stainless steel plate at a power of 600W. In both cases argon was delivered shroud gas was used at a rate of 20 l/min.

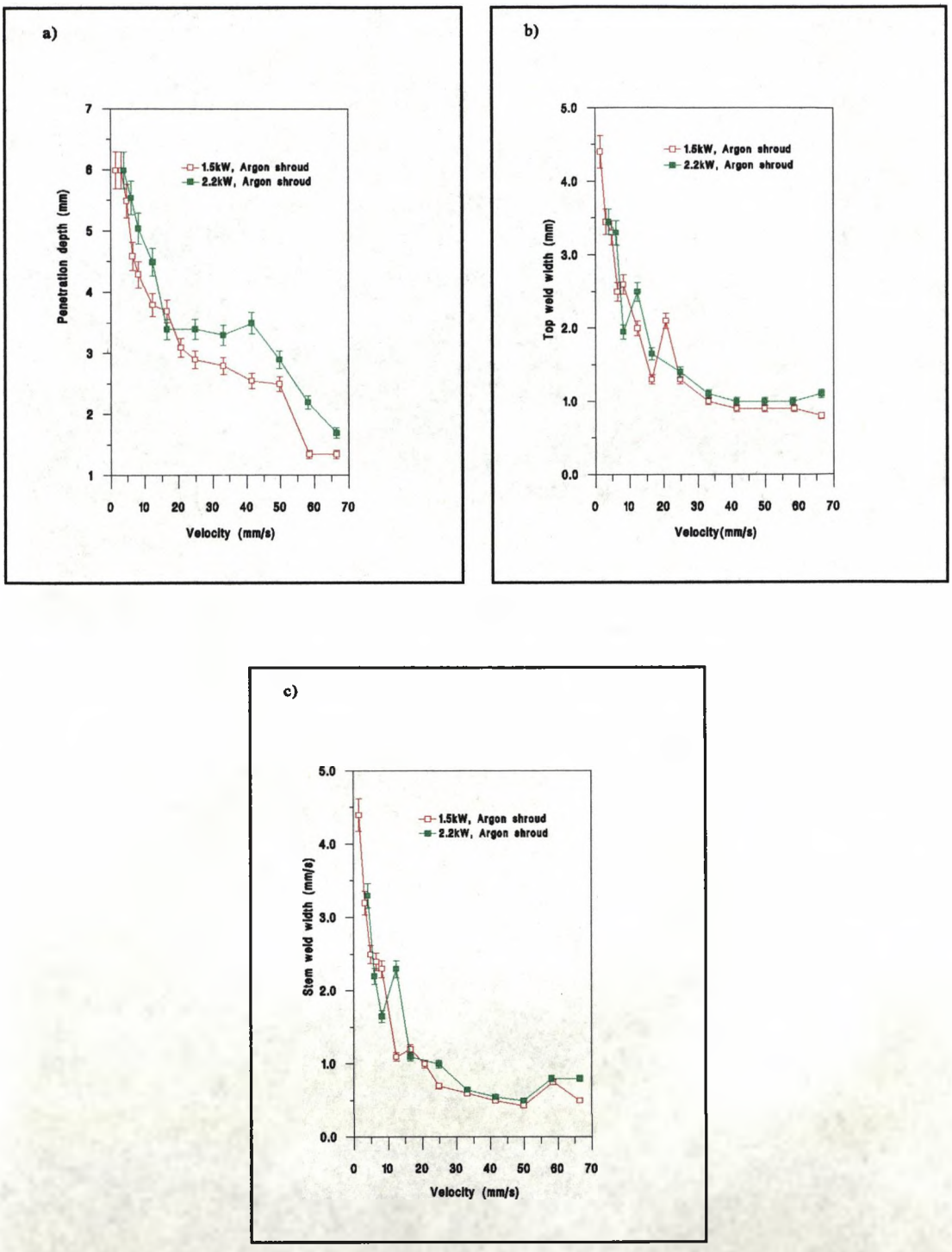


Fig.4.30 Penetration depth, top weld width and stem weld width for CO laser bead on plate welds on 6mm mild steel plate, using argon shroud gas at 20 l/min.

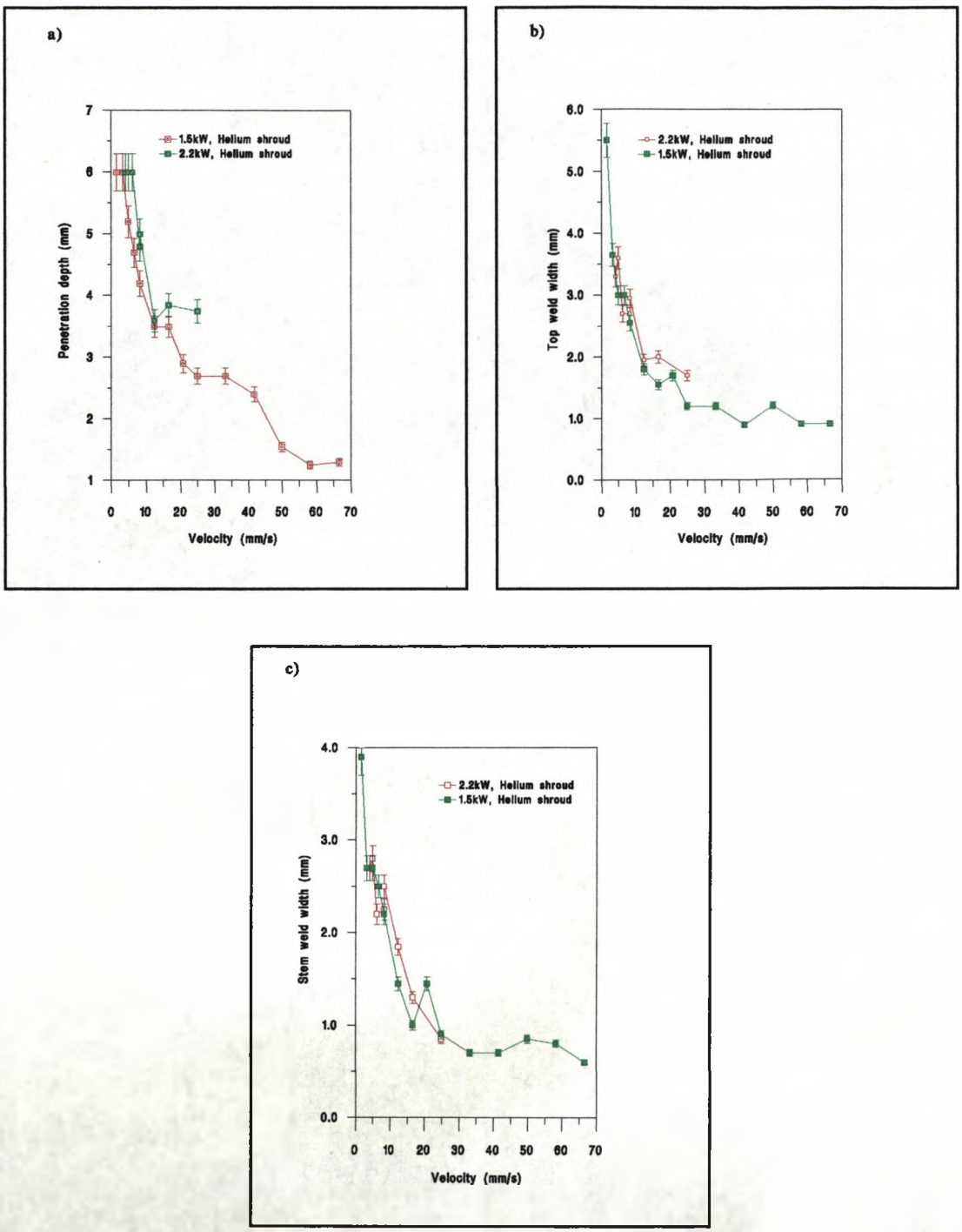


Fig.4.31 Penetration depth, top weld width and stem weld width for CO laser bead on plate welds on 6mm mild steel plate, using helium shroud gas at 20 l/min.

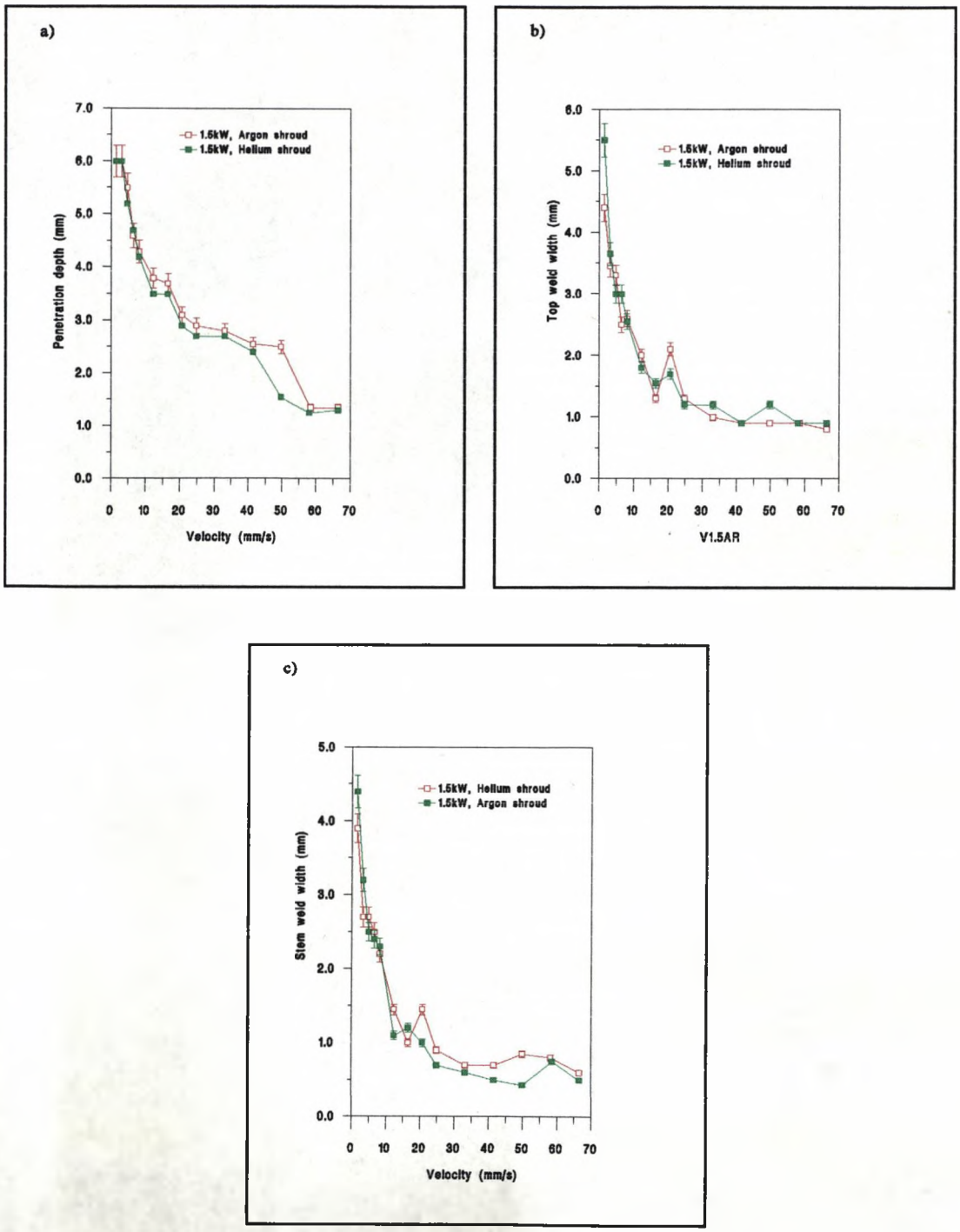


Fig.4.32 Penetration depth, top weld width and stem weld width for CO laser generated bead on plate welds in 6mm mild steel at a power of 1.5kW, using argon and helium shroud gas.

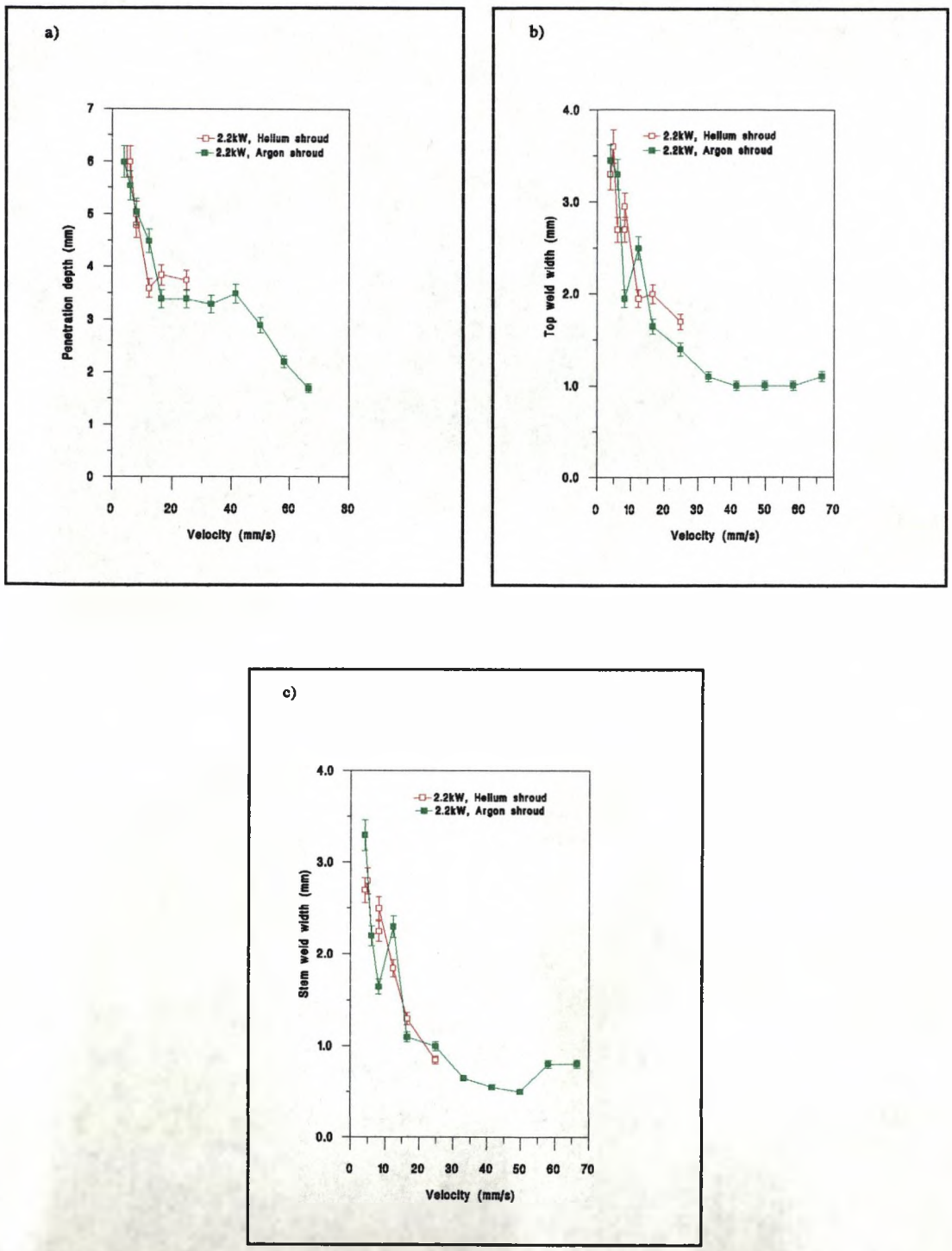


Fig.4.33 Penetration depth, top weld width and stem weld width for CO laser generated bead on plate welds in 6mm mild steel at a power of 2.2kW, using argon and helium shroud gas.

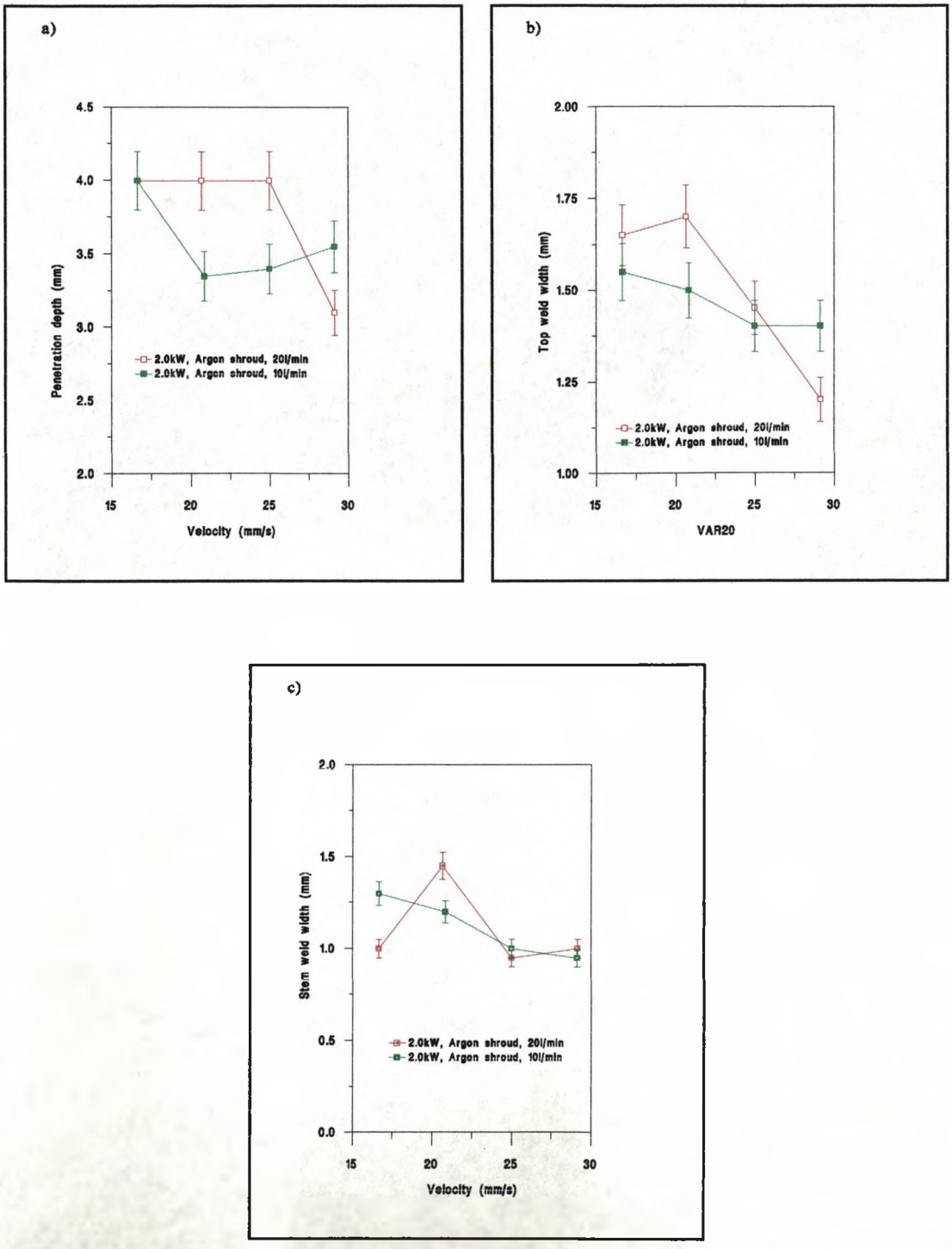


Fig.4.34 Penetration depth, top weld width and stem weld width for CO laser generated bead on plate welds in 4mm mild steel at a power of 2.0kW, using argon shroud gas at flowrates of 10 and 20 l/min.

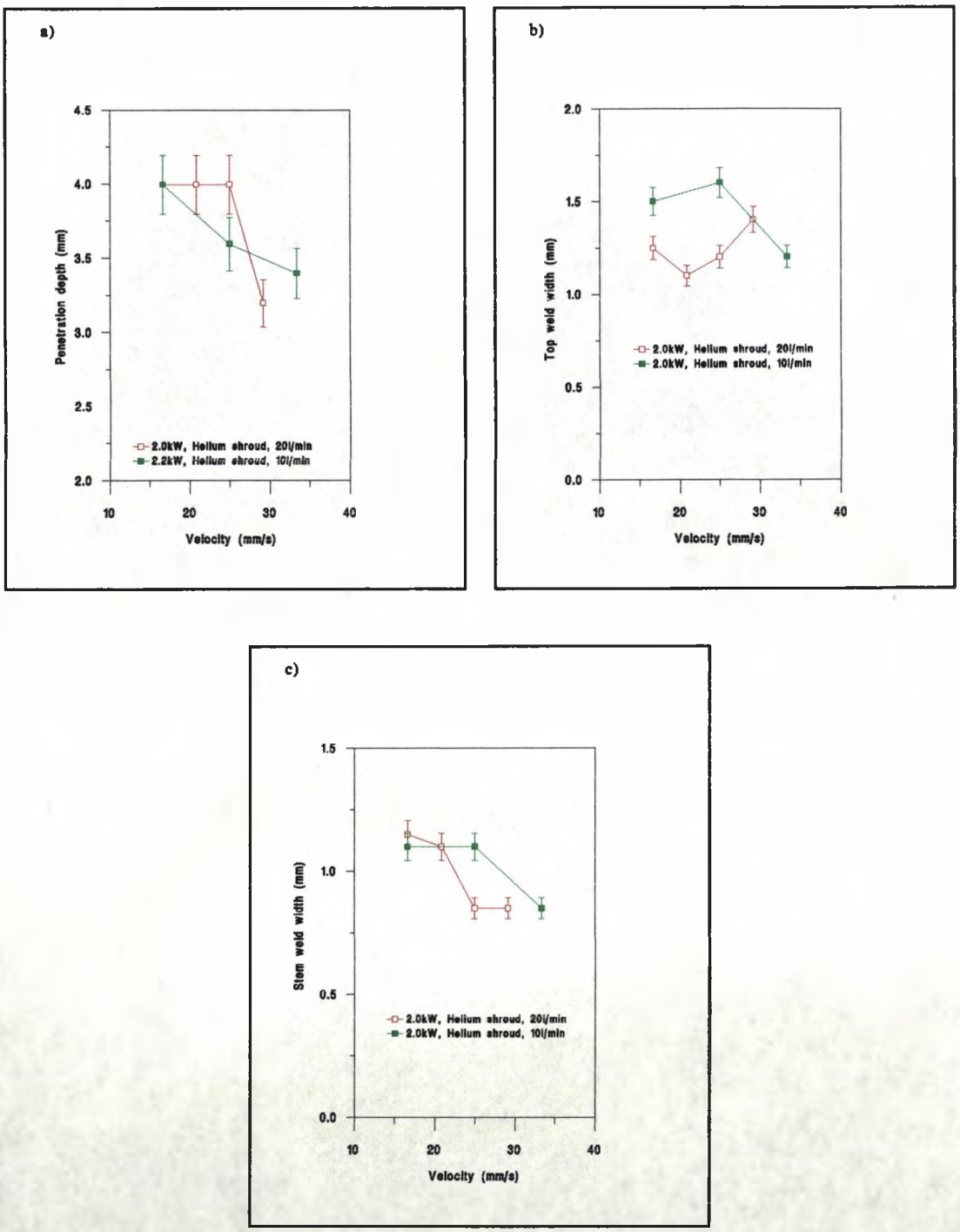


Fig.4.35 Penetration depth, top weld width and stem weld width for CO laser generated bead on plate welds in 4mm mild steel at a power of 2.0kW, using helium shroud gas at flowrates of 10 and 20l/min.

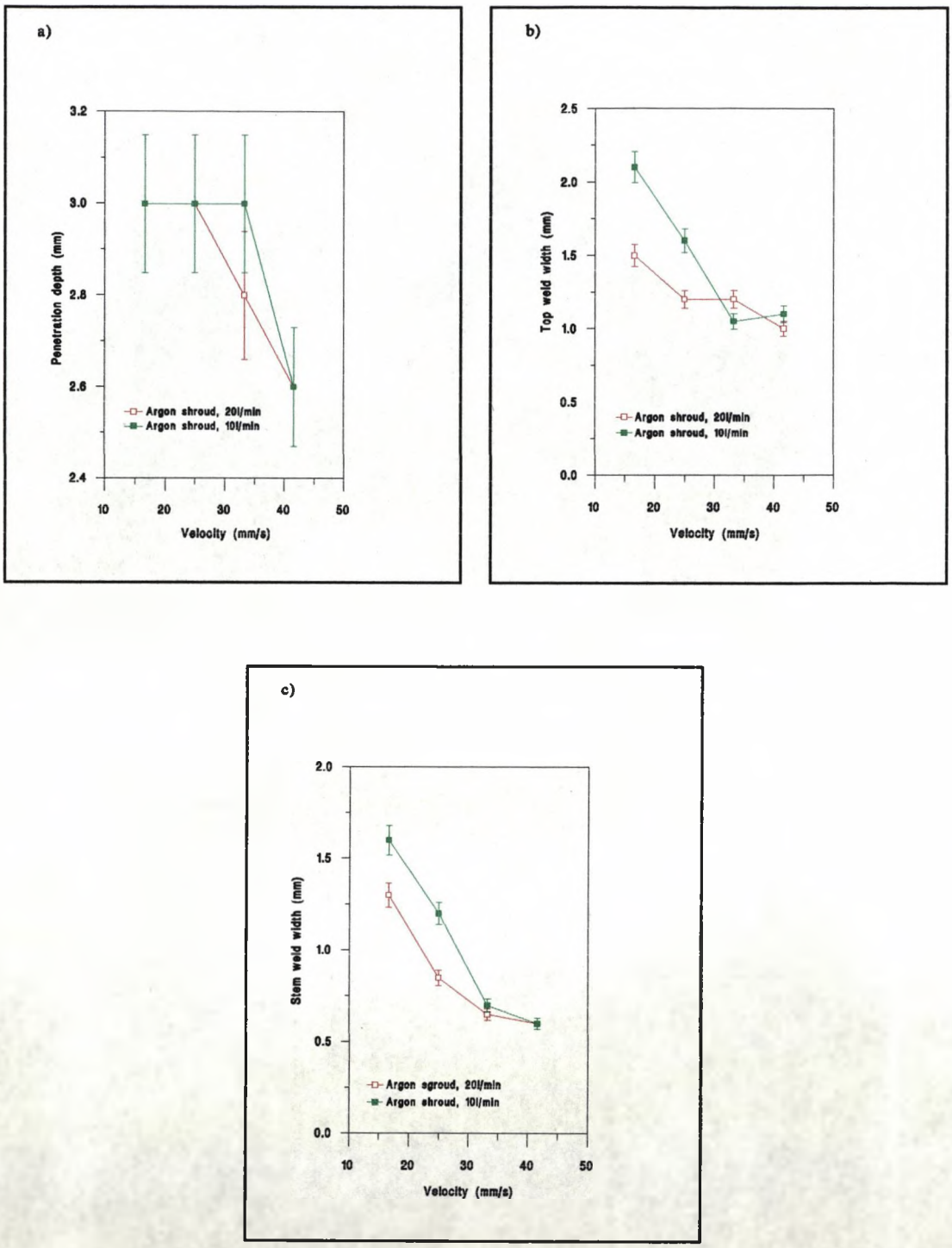


Fig.4.36 Penetration depth, top weld width and stem weld width for CO laser generated bead on plate welds in 4mm mild steel at a power of 2.0kW, using argon shroud gas at flowrates of 10 and 20 l/min.

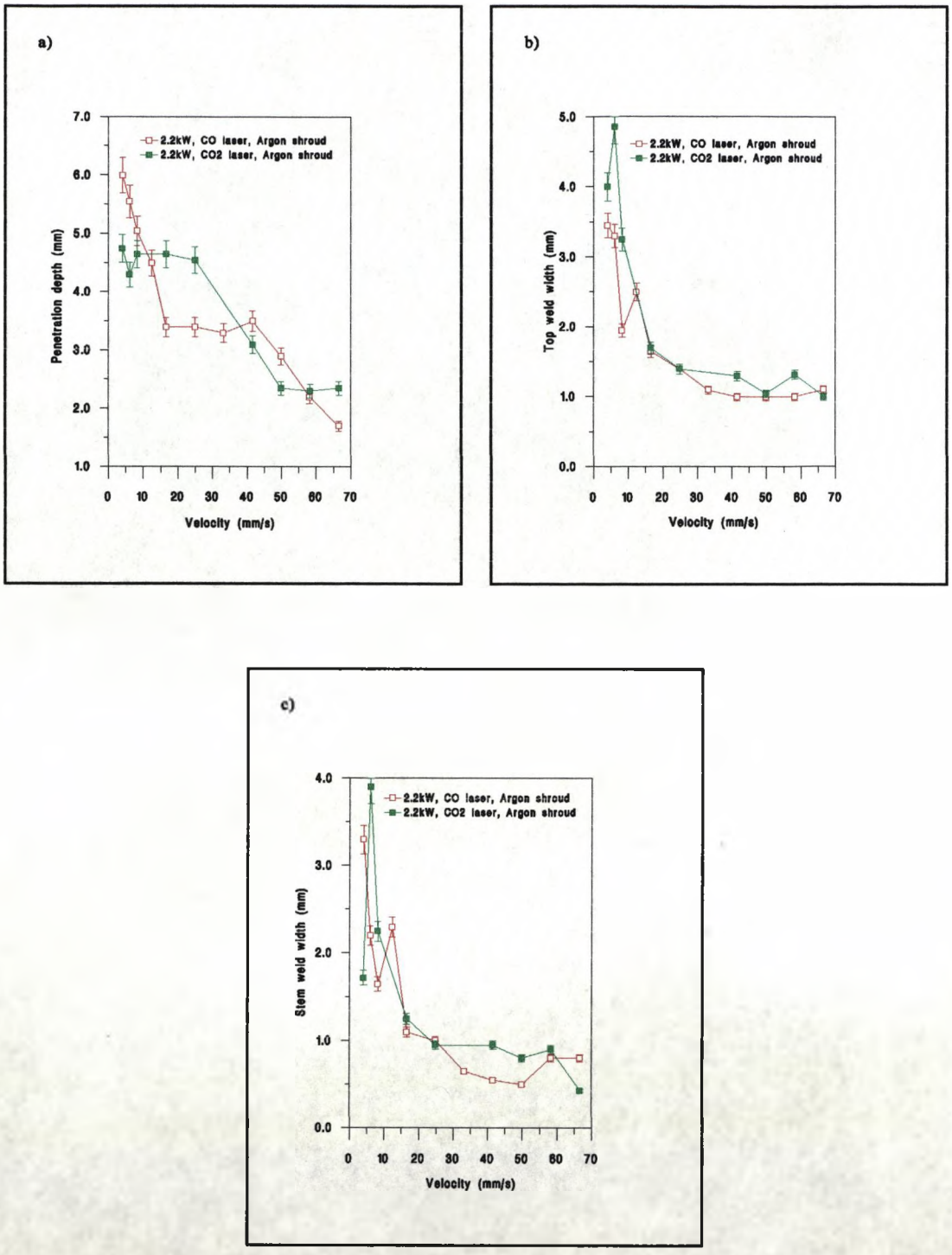


Fig.4.37 Penetration depth, top weld width and stem weld width for comparative CO/CO_2 laser welding trials using a 0.31mm spotsize, argon shroud gas and a power of 2.2kW.

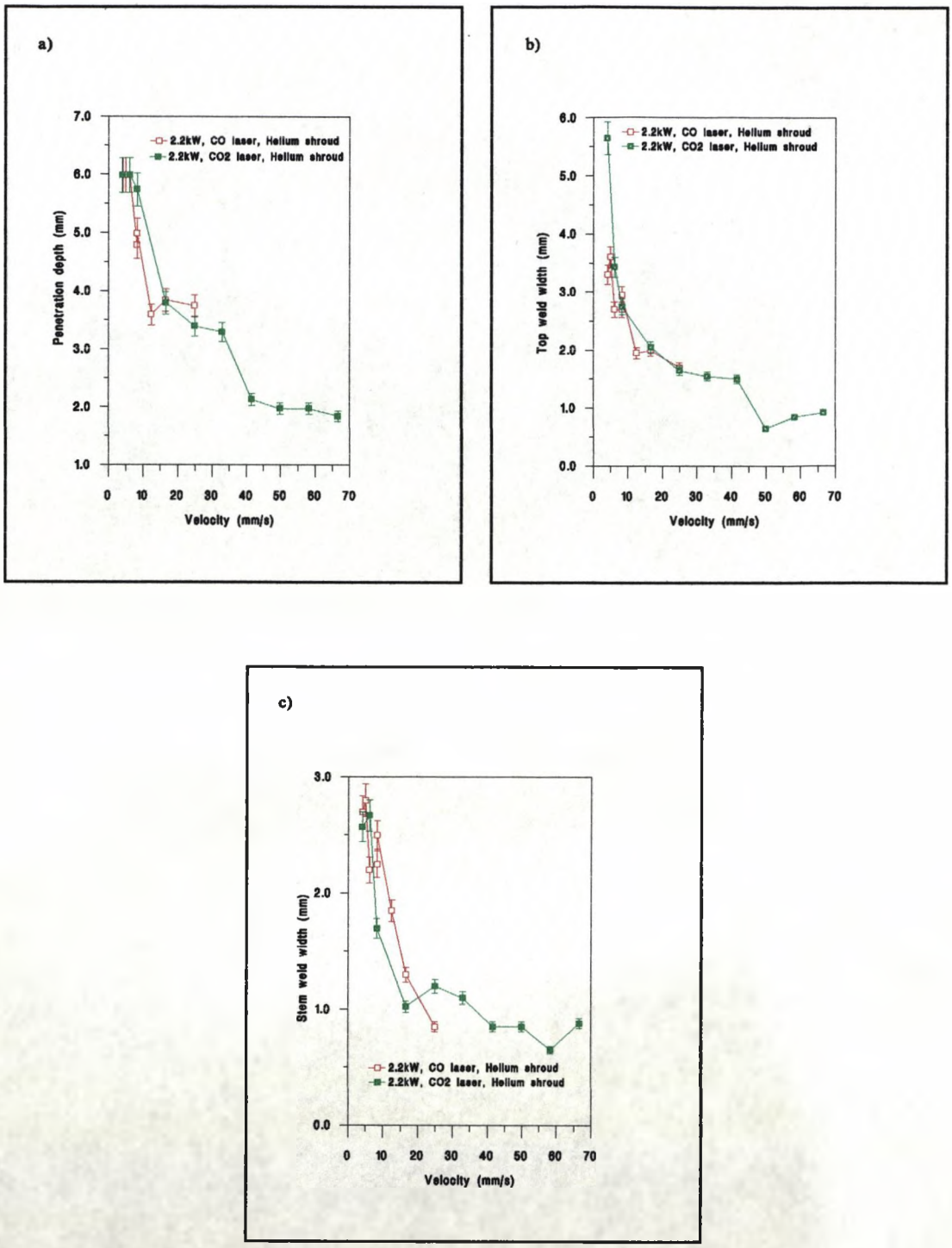


Fig.4.38 Penetration depth, top weld width and stem weld width for comparative CO/CO_2 laser welding trials using a 0.31mm spotsize, helium shroud gas and a power of 2.2kW.

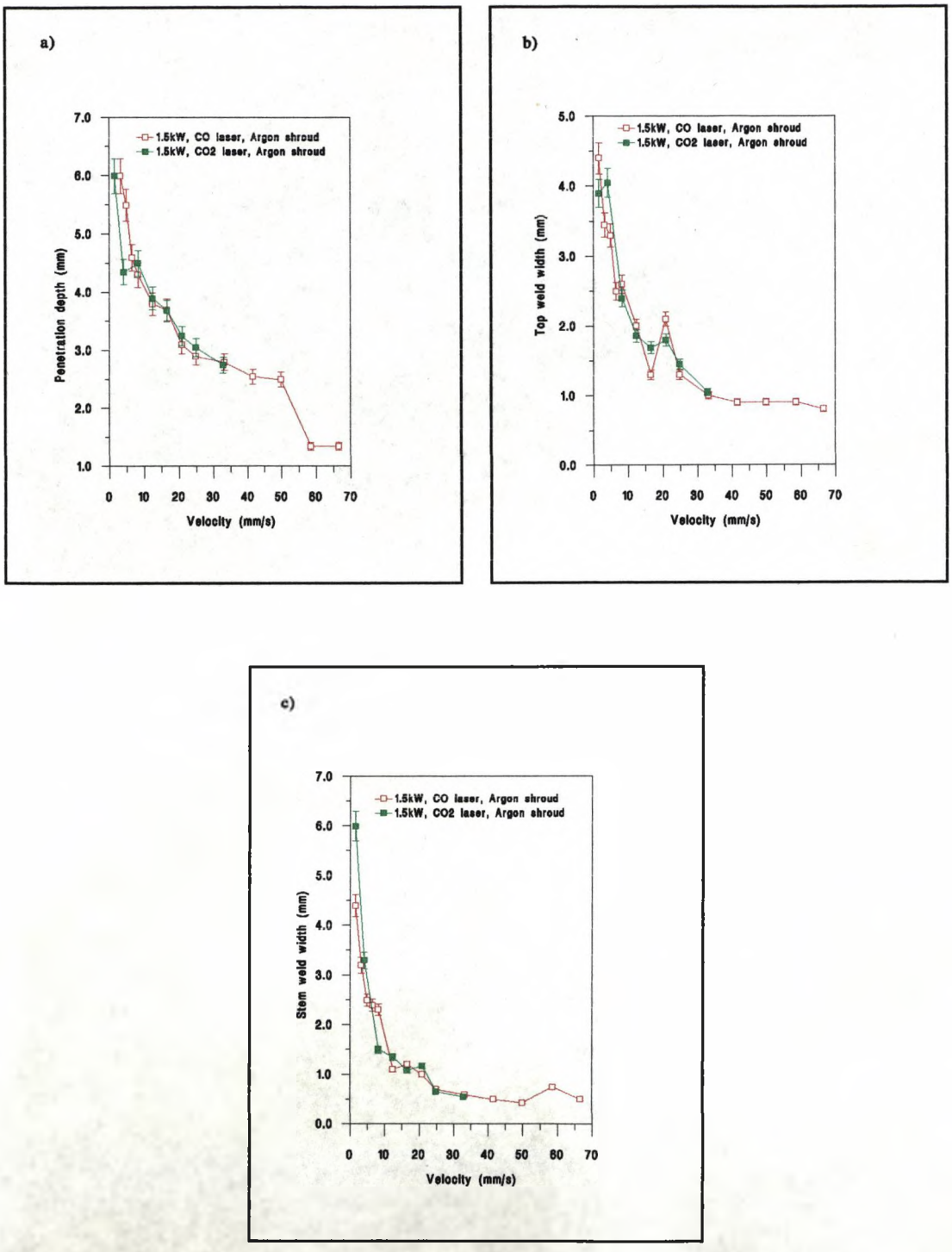


Fig.4.39 Penetration depth, top weld width and stem weld width for comparative CO/CO_2 laser welding trials using a 0.31mm spotsize, argon shroud gas and a power of 1.5kW.

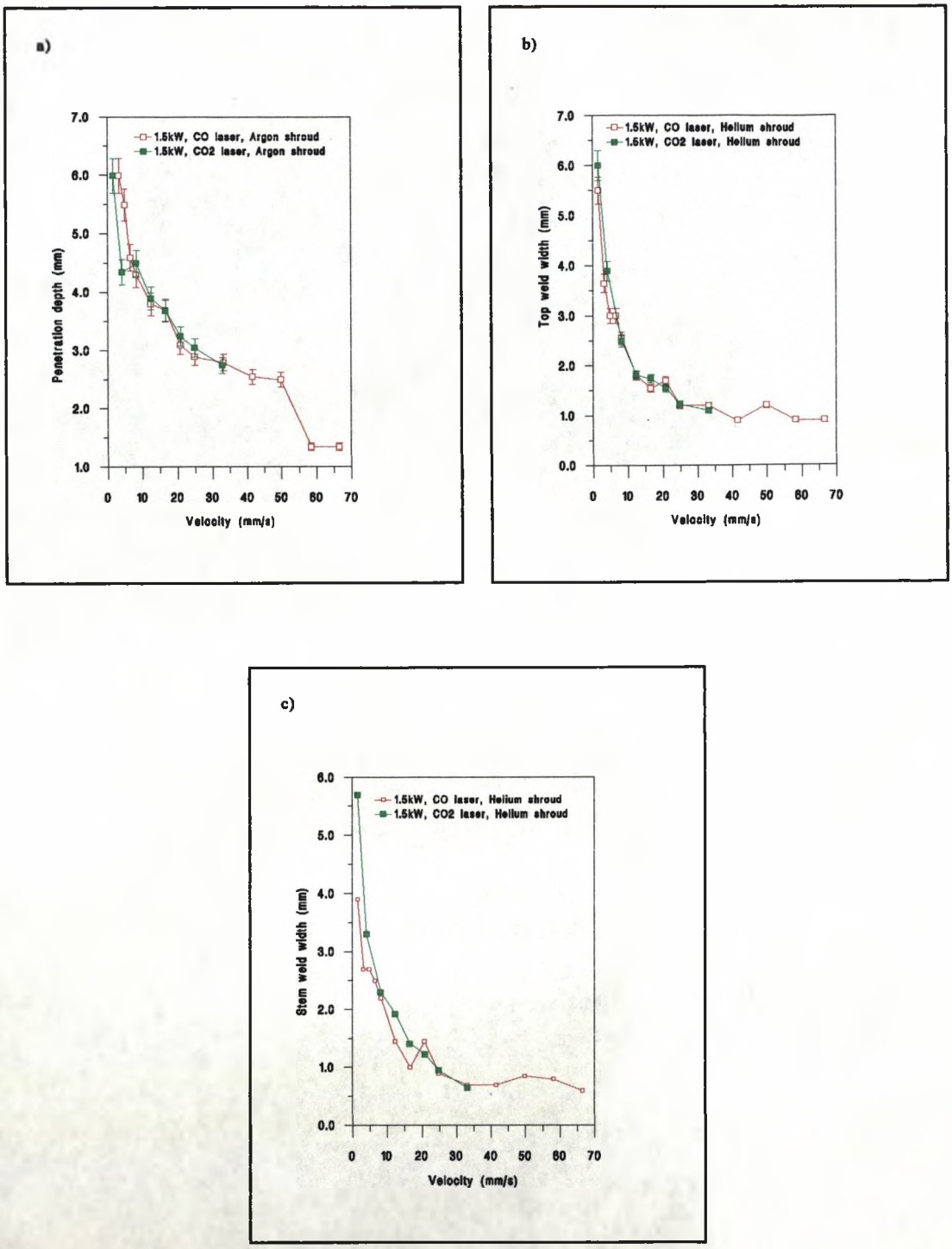


Fig.4.40 Penetration depth, top weld width and stem weld width for comparative CO/CO_2 laser welding trials using a 0.31mm spotsize, helium shroud gas and a power of 1.5kW.

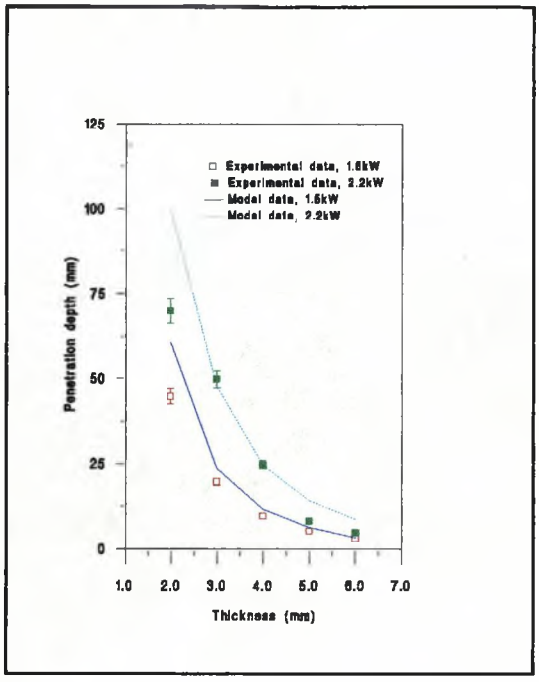


Fig.4.41 Comparison between theoretically calculated and experimentally observed maximum welding velocities for various thicknesses in CO laser welding.

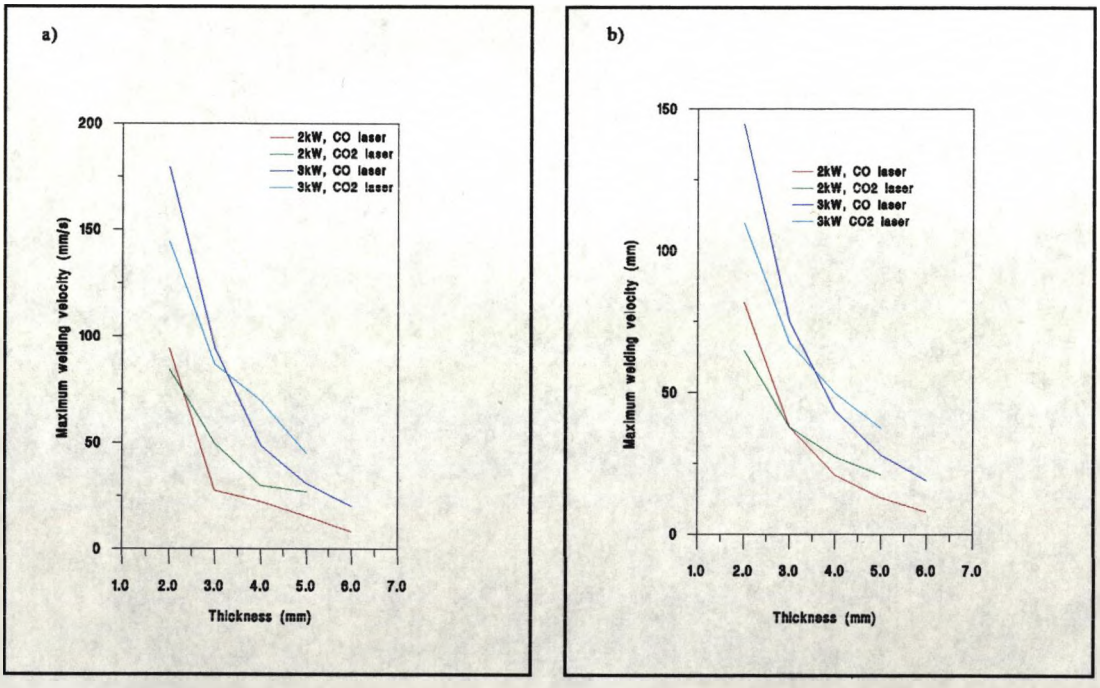


Fig.4.42 Comparison between theoretically calculated penetration depths for CO and CO_2 lasers using a power of 2 and 3kW, and a beam radius of a) 0.15mm, and b) 0.2mm.

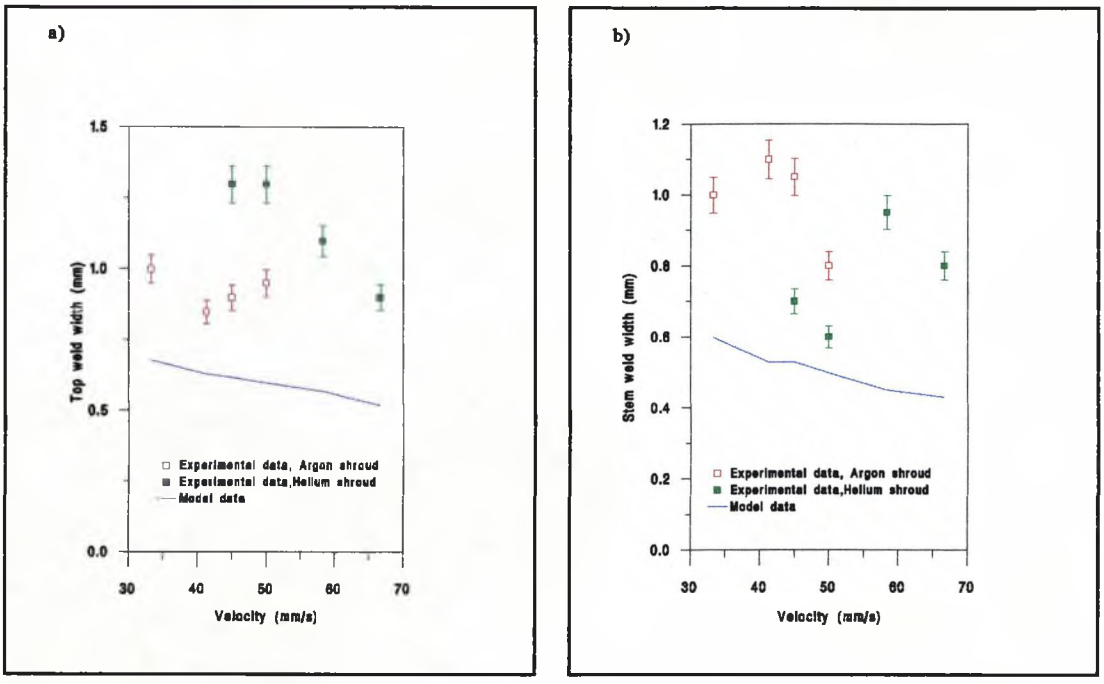


Fig.4.43 Comparison between theoretically calculated and experimentally determined weld widths for CO laser welding of 2mm mild steel using a power of 1.65kW.

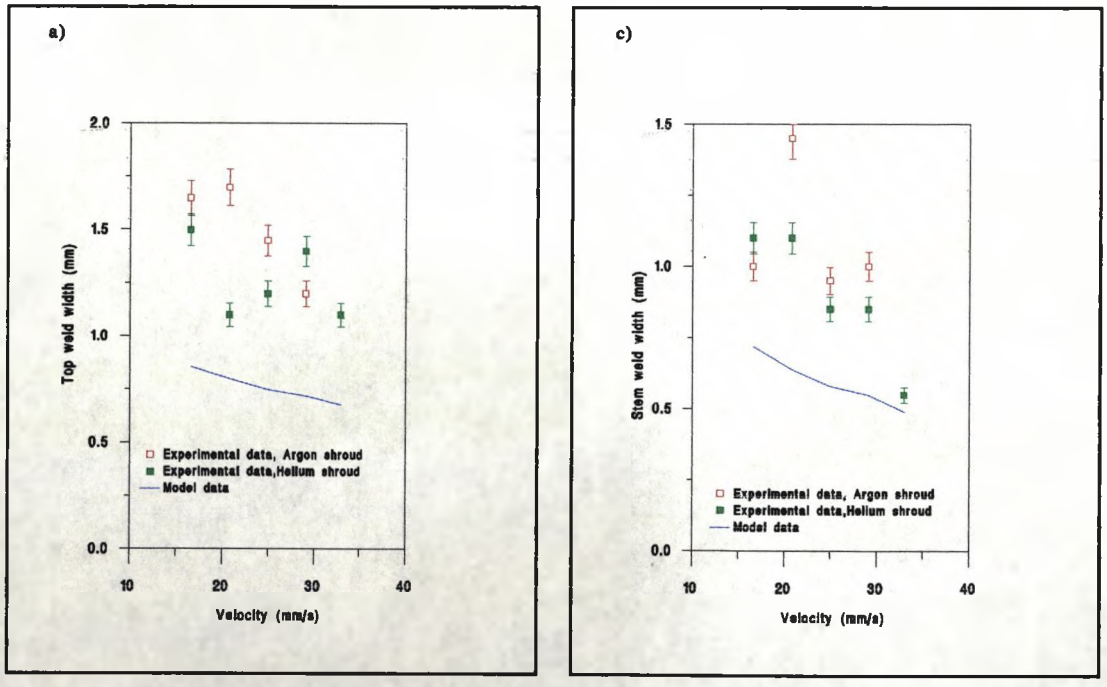


Fig.4.44 Comparison between theoretically calculated and experimentally determined weld widths for CO laser welding of 4mm using a power of 2.0kW.

5 Conclusions and Further Work

5.1 Main Conclusions

In the first section of chapter 2, a model of the pulsed laser welding process was developed. The model solved the heat conduction equation using a variation of the enthalpy formulation, in order to calculate the position of the ablating boundary. The model neglected fluid flow in the weldpool, and simplified the absorption due to multiple reflections within the keyhole. In the second section of chapter 2, a similar model for the pulsed laser edge plate machining process was developed. It assumed that all the molten material was blown out of the kerf, and so the machined front was defined by the melting isotherm. The model neglected the molten fluid flow and any multiple reflections within the kerf.

The results from both models were compared with experimental data. It was found that although the pulsed laser welding model predicted penetration depths with some success, it underestimated the top bead weld widths by upto 30%. The main reasons for this discrepancy were a combination of the effects of Marangoni convection, and the upward flow due to the excess ablation pressure in the keyhole. Both these mechanisms tend to push the hot molten material under the beam, outwards into the cooler regions, with the result of increasing the weld width.

Reasonable agreement was found when results from the pulsed laser machining model were compared with experimental data for lower powers. However, as the power increased, the generation of molten material within the kerf became so high, the gas jet was unable to remove all the melt, which invalidated one of the major assumptions of the model. In this region, the model overestimated the machined depth by upto 40%.

As long as the keyhole is formed, the top weld bead width depends mainly upon the time taken for each pulse, the keyhole radius (which is approximately equal to the beam radius at the top of the weld), and is only weakly dependent upon the pulse shape. An increase in the time taken for a particular pulse, or an increase in beam radius, will increase the weld width.

The penetration and machined depths are strongly dependent on the power density, the pulse duration, and again only weakly dependent upon pulse shape. An increase in power density (i.e an increase in power or a decrease in beam radius), or the time taken for a particular pulse (using the same power), will result in an increase in penetration or machined depth.

In chapter 3, a mathematical model of the laser keyhole welding process was developed that calculated the keyhole profile by evaluating the pressure and energy balance at the keyhole walls. The

heat balance consisted of the Fresnel absorption at the keyhole walls and the inverse Bremsstrahlung absorption in the plasma, for the heat input; the vaporisation of the ablated material and the conduction of heat throughout the workpiece as the heat output. The pressure balance consisted of the pressure due to surface tension, the recoil pressure due to the material ablating off the keyhole walls, and the pressure due to the fluid flow within the keyhole. Previous authors have either neglected the ablation term or the pressure balance altogether, or made simplifying assumptions about the flow in the keyhole in order to estimate it's value. This model solves the Navier Stokes equations in the keyhole and the surrounding area in order to obtain a more accurate evaluation of the influence of this term in the overall process. The model neglected both the defocussing and the heat source effects associated with the plasma above the workpiece in the plume, as well as the action of the gas shroud or the fluid flow within the weldpool itself.

It was found that the velocity of the ablating material is largest at the keyhole mouths, where the pressure difference at the keyhole walls is at its highest value, and decreases towards the centre of the keyhole. Typical values for the cases studied here ranged between 30-40m/s at the top of the keyhole, and 5-10m/s in the middle. Due to the pressure difference between the keyhole walls and the outside ambient pressure, the ablating material from the keyhole walls accelerates out of the top and the bottom mouths of the keyhole at velocities of the order of 10^2m/s. This is consistent with the experimentally measured values of Beck et al (1989).

A number of previous authors have simplified the flow in the keyhole, by assuming viscous forces were negligible, in order to obtain an analytical expression that estimates the ablation contribution. The results from model show that this assumption is invalid, and that there is a large difference in exit velocities with varying viscosity. Other authors that have taken into account viscous forces (eg Klein et al (1996)), still simplify the flow in the keyhole, and assume that the ablation is uniform throughout the depth of the keyhole. Again the model shows that these assumptions are invalid, and the velocity of the ablating vapour is a strong function of depth.

The model results indicated that even in extreme cases, the overall power used in ablating the material rarely rose above 5% of the overall value of absorbed power. Therefore one can conclude that this term does not affect the laser welding process significantly, and so in order to develop a simpler model, this term was dropped. Thus the keyhole boundary was described by the energy balance alone. This model was then compared with experimental results in chapter 4.

The results from the simpler model were compared with experimental data in chapter 4. It was found that although the model predicted penetration depths with some success, it overestimated the top bead weld width by 20-50%. This was due to the fact that the model neglected Marangoni

convection in the weldpool, as well as the extra heat source due to plasma absorption in the plume.

In order to assess the effect the plume has upon the laser welding process, the plume was simulated by placing a varying point source at the top of the keyhole, and modifying the heat balance at the keyhole walls accordingly. The main contribution the addition of the point source makes was to open the keyhole mouth, this ensures that the beam reaches the keyhole. The value of weld width increases therefore at the top of the weld, but decreases at the bottom due to the fact that the presence of the point source blocks a fraction of the beam, decreasing the power available in the keyhole. This will decrease the penetration depth.

Experimental work was undertaken on two CO_2 lasers at Liverpool University, in order to investigate the effects of varying a number of process parameters on the laser welding process. A full discussion of the results is presented in chapter 4. The most significant of these outline the role of the plasma in the plume during laser keyhole welding.

A decrease in lens focal length produces an increase in power density at the workpiece. Intuitively, one would therefore expect the penetration depth to increase accordingly, however, three other factors that occur with the decrease in focal length also play an important part in the process. The first two are the defocussing effect above the workpiece together with the absorption of plasma within the plume. These effects will increase with increasing power density and decreasing speed (hence increasing beam-material interaction time), resulting in a larger effective spotsize at the workpiece. The third factor is the divergence of the beam down the depth of the keyhole. It was a combination of these three factors that lead to a decrease in penetration depth with a decrease in lens focal length when welding 6mm mild steel. All these effects are less pronounced when welding thinner material, where the welding speeds are much faster, and the beam-material interaction time is much less. Here, welding with a shorter focal length lens gave deeper penetration depths in m mild steel plate.

The role of the shroud gas is to control the amount of plasma within the plume, by displacing the more easily ionised metal vapour. At high powers and slow speeds, the shroud gas fails to fully control the plasma, and so the beam defocussing above the keyhole and the absorption of light within the plume becomes detrimental to the process as a whole, resulting in shallower penetration depths, and wider weld widths. This effect occurs first in argon, as it has a lower ionisation energy, and produces a plasma much more easily. Thus, when welding with higher powers, especially at lower speeds, welds with argon shroud gas produce shallower penetration depths than those in helium shroud. It was significant that in CO laser welding, there was no significant difference in welding with argon or helium shroud gas.

Welding experiments were then carried out using two CO lasers, and compared to welds generated by a CO_2 laser under similar conditions. At lower powers the increased plasma absorption of the CO_2 laser using helium shroud gas at any flow rates and argon shroud gas at high flow rates, increased the coupling of the laser beam into the workpiece, and resulted in larger penetration depths. When CO_2 welding using an argon shroud gas at lower values of flow rate however, the plasma in the plume cannot be controlled, and begins to block the beam. Therefore, in this case, greater penetration depths can be obtained with a CO laser.

At higher powers, there is little to distinguish between the two lasers at medium to fast welding speeds, however as the process speed became slower, welding with a CO_2 laser using argon shroud gas results in wider weld widths and shallower penetration depths, due to the problems associated with the plasma absorption within the plume. The CO laser produces larger penetration depths than CO_2 welds using helium shroud gas, as well, although the difference here was less pronounced.

The comparisons between the CO and CO_2 laser welding trials underline a crucial aspect of laser welding; that in order for good laser welding to occur, the plasma in the plume has to be generated and controlled. If the plasma is not generated, as in the low power CO laser experiments, enhanced coupling of the laser power into the workpiece will not occur, causing a reduction in penetration depth and weld width with comparable experiments where a plasma is generated. If the plasma cannot be controlled, no matter what the shielding gas conditions are, as in the case of very low speed welding, then reduction in penetration depths will occur, and in some cases, the process may cease altogether. This is where the CO laser offers its main advantage, it can still process in a region were laser welding becomes difficult for a CO_2 laser.

5.2 Future Work

Important work needs to be done by modelling the assumptions of the pulsed and continuous laser welding models, and incorporating these into the overall calculations, if the discrepancies between theory and experiment are to be resolved. Most important of all for the CW laser welding model would be to incorporate the effect of the shroud gas, in order to obtain a better estimation of the keyhole radius at the top of the keyhole. Without such a model, there will always be discrepancies between theoretically calculated and experimentally obtained weld widths. The multiple reflections, which play an increasingly important role as the speed approaches the maximum welding velocity, also needs to be taken into account, as this will create an extra absorption term within the keyhole that will further increase the weld width. Finally, in order to model the conditions within the keyhole more fully, the 2 dimensional constraint should be discarded, and a fully 3 dimensional description of the keyhole profile is required. This would mean that the computational time would be greatly

increased, however, with the advancement of modern computer technology, this should not be a problem in years to come. A more advanced pulsed laser welding model could also benefit from these additions, however, in this case it is more important to study the fluid dynamics of the weldpool in order to resolve the theoretical and experimental discrepancies.

The most obvious way to further the comparison between the welding capabilities of CO and CO_2 lasers is to investigate the welding of different materials. It would be particularly useful to study the CO laser welding of stainless steel, the constituents of which easily form a plasma in CO_2 laser welding, especially when argon is used as a shroud gas; this causes a problem at slower speeds. Copper and aluminium are both difficult to weld with a CO_2 laser, due to the fact that the high absorptivities of these materials ensure that only a tiny fraction of the energy from the laser beam is absorbed by the workpiece, and their high conductivities ensure that this absorbed energy is then dissipated throughout all the material, with the result that the heat leaves the region quickly. The higher absorption of CO laser light would enhance the welding process by applying more energy to the material, whether this is enough to initiate keyholeing however, remains to be seen.

Another aspect of the CO laser welding process that needs to be further investigated, is the maximum penetration depth for any particular power. In CO_2 laser welding, as the speed decreases, there is a point where no further increase in penetration depth can be obtained, and the weld just gets wider and wider. This is due to the problems of defocussing and absorption of the laser light in the plume mentioned earlier. In theory, this should be less of a problem in the CO laser case, due to the fact that the plasma absorbs 4 times as much laser light in the CO_2 case than in the CO case. Therefore, the theoretical maximum penetration limit for a CO laser at a particular power should be much higher.

There is a need to study the effects of using different shroud gases at higher powers during CO laser welding. Under the range of powers considered (up to 2.1kW), there is no discernible difference when using either argon or helium as a shroud gas. This comparison needs to be extended to higher powers, where the differences between argon and helium shroud gases are much more pronounced. It would also be informative to weld with gases with a lower ionisation energy, such as CO_2 or N_2. Finally, with the emergence of high power pseudo continuous Nd:YAG lasers, it is imperative that a project is undertaken to compare the comparative material processing capabilities of the CO, CO_2 and Nd:YAG lasers.

6 References

Akhter, R., Davis, M., Dowden, J.M., Kapadia, P.D., Ley, M., Steen, W.M. (1989). 'A Method for Calculating the Fused Zone Profile of Laser Keyhole Welds'. J Phys. D: Appl Phys., 21: pp 23-28

Andrews, J.G., Atthey, D.R. 'Hydrodynamics Limit to Penetration of a Material by a High Power Laser Beam'. J Phys D Appl Phys. 9. pp 2181-2194.

Ashby, MF., Easterling, KE. (1984). 'The transformation hardening of steel surfaces by laser beams'. Acta Metalica 32 pp 1935-1948.

Ashby, MF, Shercliffe, H.R. (1986) 'Masterplots for predicting the case-depth in laser surface treatments'. Eng Dept, Cambridge.

Bass, M (1983). Physical processes in Laser-Material Interaction'. New York, Plenum Press.

Carslaw, HC., and Jaeger, J.C. (1959). 'Conduction of heat in solids' Oxford, Oxford University Press.

Chan, C., Mazumder J., Chen M.M. (1984) 'A Two Dimensional Transient Model for Convection in Laser Melted Pool. Met Trans. A 15A: pp 2175-2184

Cline. HE and Anthony. T.R. (1977) 'Heat treating and melting of a scanning laser or electron beam'. J.Appl. Phys 48(9): pp 3895-3900.

Clucas, DAV., MSc(Eng) Dissertion, Liverpool University.

Colla T.J., Vicanek ,M., Simon, G.,. (1992) 'Heat Transport in Melt Flowing Past The Keyhole in Deep Penetration Welding', J Phys D 27 pp2035-2040

Coherent (1993) 'M^2, the definition of beam propagation' Coherent.

Crank, J. (1984) 'Free and Moving Boundary Value Problems'. (Oxford University Press).

Croft, DR and Stone, JAR (1977) 'Heat Transfer Calculations Using Finite Difference Methods. Pavic Publications.

Davis. M (1983). PhD Thesis. University of Essex.

Davis.M, Kapadia.P.D., Dowden.J.M.,' Solution of a Stefan Problem in the Theory of Laser Welding by the Method of Lines', J Comp. Phys, 60, 534 - 538

Davis.M, Kapadia.P.D, Steen.W.M, Courtney.C.H.G. (1986) 'Heat hardening of Metal surfaces with a scanning laser beam'. J. Phys. D. Appl Phys: 19 : pp 1981-1997.

Dowden.J, Davis.M, Kapadia.P.D. (1983) Some Aspects of the Fluid Dynamics of Laser Welding" J.Fluid.Mech Vol 126 pp123-46

Dowden.J, Postacioglu.N., Davis.M., Kapadia.P.D. (1987) 'A keyhole Model in Penetration Welding with a laser'. Phys. D: Appl Phys. 20.: 35 - 44.

Dowden, J. Kapadia, P., Postacioglu,N., (1989) 'An analysis of the Laser Plasma Interaction in keyhole Laser Welding'. I Appl. Phys. 22: pp 741-749

Ducharme.R, Kapadia. P, Dowden.J, Williams.K, Steen.W.M. (1992) 'An Integrated Mathematical Model of the Keyhole and the Weldpool in the Laser Welding of Thin Metal Sheets', Proc ICALEO '92. D.Farson, W.Steen and I.Miyamoto (Eds) -LIA, Orlando, Florida 75, pp176-186

Ducharme.R, Kapadia. P and Dowden.J (1993) 'The Collapse of The Keyhole in the Laser Welding of Materials' ICALEO 93

Ducharme R, Williams.K., Kapadia, P. Dowden J., Steen, WM (1995) 'The Laser Welding of Thin Metal Sheets an Integrated Mathematical Model and Weld Pool Model with Supporting Experiments' J Phys. D:Appl Phys 27.

El-Awadi.M.K and Elshehawey.E.F (1986) 'Heating a slab induced by a time dependent laser irradiance - an exact solution' J.Appl.Phys Vol 60(7), pp2250-2255

Erasenthiran.P (1997), Private Communication, to be published, 1998 PhD Thesis. Liverpool University.

Erasenthiran.P, Jungreuthmayer .C, O'Neil.W, Steen.W.M, (1997) 'An investigation of step shaping using Nd:YAG laser for parts produced by laminated object manufacturing techniques', LANE'97.

Eyers, N.R, Hartree, DR Ingham J., Jackson, R, Sarjant R.J. Wagstaff, SM (1946). Phil. Trans. R Soc. M240

Gellert. B, and Egli.W. (1988) 'Melting of copper by an intense and pulsed heat source' J.Phys.D: Appl Phys, Vol 21, 1721-1726

Glowacki.M., Ducharme.R., Kapadia.P., and Dowden.J. 'The Welding of Thin Metal Sheets Sheets with CO_2,CO and pseudo continuous Nd:YAG Lasers Investigated by Means of an Integrated Mathematical Model' ICALEO'93 pp135-144

Gratzke.U, Kapadia.P.D. and Dowden.J (1991). 'Heat conduction in high Speed laser welding'. J.Phys D: Appl Phys 24: pp 2125-2134.

Gratzke,U., Kapadia,P.D., Dowden, Kroos.J. 'Theoretical Approach to the Humping Phenomena in welding process', J.Phys D Appl Phys 25, pp 64l~647

Huges, TP. (1975) 'Plasmas and. Laser Light'. Adam Hilger Ltd, London

Kaplan.A (1994) 'A Model of Deep Penetration Laser Welding Based on Calculation of the Keyhole Profile' J Phys D: AppI Phys. 27 pp 1805-1814

Klein T., Vicanek M, Kroos.J., Decker I., Simon G. (1994) 'Oscillations of the Keyhole in Penetration Laser Beam Welding' J Phys D:Appl Phys 2023-2030

Kou.S. (1981) 'Simulation of Heat Flow During the Welding of Thin Plates' Met Trans A. Vol 12A. pp2025-2030

Kou.S.,Le.Y, (1983) 'Three Dimensional Heat Flow and Solidification During the Autogenous GTA Welding of Aluminium Plates' Met Trans A. Vol 14A. pp2245-2253

Kou.S.,Le.Y, (1984) 'Heat Flow the Autogenous GTA Welding of Pipes' Met Trans A. Vol 15A. pp1165-1171

Kou S., Sun, D.K (1985). 'Fluid Flows and Weld Penetration in Stationery Arc Welds'. Met Trans A 16A: pp203-213.

Kou S, Wang, YR (1986) 'Convection in Laser Melted Pools'. Met Trans. A 17A pp 2265-2271.

Kroos.I, Gratzke.U., Simon G (1992 a) 'Towards a Self Consistent Model of the Keyhole in Penetration Laser Beam Welding', J Phys D: Appl Phys 26 474-480

Kroos.J., Gratzke., SimonG. (1992 b). 'Dyminamic Behaviour of the Keyhole in Laser Welding'. JPhys.D. Appl Phys. 26: pp 480 - 486.

Landau L.D. and Lifshitz. EM (1959). 'Course of Theoretical Physics. Vol 6: Hydrodynamics.Oxford. Pergamon.

Lancaster, IF. (1986). 'The Physics of welding'. Pergammon Press. Oxford.

Mazumder.J, Steen W.M (1980) Heat Transfer Model For CW Laser Material Processing'. J AppI Phys 51(2),:pp 941-947.

Meyer. G.H (1970). Niunerical Math. 16 248-67.

Modest.M.K, and Abiakans.H 'Heat conduction in a moving semi infinite solid subjected to pulsed laser irradiation' J.Heat Transfer 108(8), pp597-601

Nakagawa.T, Kunienda.M, Liu. S.D, (1995) 'Laser cut sheet laminated dies by diffusion bonding', LANE'95

Noller, F. 'The stationary shapes of vapour cavity and molten zone on eb-welding'. 3rd Int Conf on Welding and Melt, Electrons and Laser Beam 89-97.

Oreper. GM, Szekely, 1 (1984) 'Heat and Fluid Flow Phenomena in Weld Pools'. J Fluid. Mech 147: pp 53-79.

Oreper G.M, Szekely. J (1987) 'Conprehensive Representation of Transient Weldpool Development in Spot Welding Operations'.. Met. Trans.A 18A: pp 1325-1332

Patankar.S.V (1980) 'Numerical Heat Transfer and Fluid Flow', Hemisphere

Postacioglu N., Kapadia,PD., Dowden,J (1987) 'Upwelling of the Liquid Region Surrounding the Keyhole in Pnetration Welding with a Laser'. J Phys. D: Appl Phys. 20: pp 340-345.

Postacioglu, N., Kapadia P.., Dowden.J (1991) 'A Theoretical Model of Themocapillary Flows in Laser Welding', J Phys D: Appl Phys 24 15-20

Rosenthal D (1946). 'The theory of moving sources of heat and its application to metal treatments'.

Trans ASME pp 848-866.

Ryalkin, N., Uglov, A et al (1978). 'Laser machining and Welding. Oxford, Pergammon Press. Ch 3.

Schuker, D. (1990) 'Modelling of the Laser Welding Process'. Proc ECLAT 1990.Cobizrg: Spreschsaal Publishing Group. Vol 1. pp 55-.

Shansundar, N., Sparrow, E.M. (1975). 'Analysis of Multidimensional Conduction Phase Change via the Enthalpy Model' J. Heat.Transfer, Trans ASME 97: pp 333-340

Schulz W., G Simon, HM Urbassek and I Decker. (1987) 'On laser fusion cutting of metals. J Phys D: Phys. 20, 481- 488.

Sepold.G, (1994) 'LASP. A chance for future rapid prototyping of metallic large volume components', LANE'94

Simon.G, Gratzke.U, Kroos.J (1992) 'Analysis of heat conduction in deep penetration welding with a time modulated laser beam' J.Phys.D.Appl Phys Vol 26, pp862-869

Smurov. Y, Uglov A.A et al (1991) 'Modelling of pulse periodic flow action on metallic materials' Int.J.Heat.Mass.Transfer, Vol 34, No4 pp 961-971

Soar.R.C, Dickens.P.M, (1996) 'Finishing laminated tooling with stereolithography EDM Electrodes'. Proc.Int.Conf on Rapid prototyping, Nottingham

Schulz,W., Simon.G,Urbassek,H.M. and Decker.I (1987) 'On laser cutting of metals' J.Phys.D:Appl .Phys, 20, pp481-488

Steen, WM (1976) PhD Thesis. Imperial College.

Steen, WM Courtney. C (1979) Met Technol, 6: p56-62

Steen, WM, Dowden, J., Davis, M Kapadia, P. (1988) 'A point and Line Source Model of Laser Keyhole Welding' J. Phys. :Appl Phys, 21. pp 1255-1260.

Steen, WM (1991). Laser Material Processing. London, Springer Verlag.

Tsai, MC., Kou, S. (1989) 'Marangoni Convection in Weld Pools With a Free Surface'. Int J. Num Math. Fluids. 9:1503-1516.

Tsai. MC., Kou S. (1990) 'Weld Pool Convection and Expansion Due to Density Variations'. Num Heat. Transfer. 17A pp 73-89.

Tzeng.Y.F (1996) PhD Thesis. Liverpool University

MISSING PAGES ARE UNAVAILABLE

Appendix 1 Three dimeniosnal Pulsed Heat Transfer Program

This program is outlined in the first part of chapter 2, where the governing equations are described in detail. The model presented here solves the time dependent heat conduction equation subject to the apprpriate boundary conditions. The solution procedure is outlined in the flowchart fure 2.10. The model is solved using the SIMPLE finite difference model sing a 120*60*60 grid, and was iterated until the maximum number of pulses had elapsed.

```
program imp3d(input,output);
const
        epsilon=0.24;
        tab=chr(9);
        pulse_on_its=200;
        pulse_off_its=200;
        its_ppulse=pulse_on_its+pulse_off_its;
        pulse_no=5;
        runs=5;
        iters_1_max=pulse_no*its_ppulse;

        boiling_point=3000;
        latent_heat_fusion=6.3e6;
        latent_heat_melting=2.47e5;
        temp_s=1500;
        melting_pt=1600;
        temp_l=1700;
        temp_v1=3000;
        temp_v2=3155;
        boiling_pt=temp_v2;
        ambient_temp=300;

        heat_coeff_0=100;
        heat_coeff_max=10;
        beam_radius=0.00045;
        pi=3.14159;

        x_min=-35;
        x_fine_min=-30;
        x_origin=0;
        x_fine_max=30;
        x_max=35;

        x_length_min=-0.005;
        x_fine_length_min=-0.001;
        x_length_origin=0.0;
        x_fine_length_max=0.001;
        x_length_max=0.005;

        y_fine_max=30;
        y_coarse_max=5;
        y_max=35;
        y_fine_length=0.001;
        y_length=0.005;

        z_length_max=0.002;
        z_max=50;
        abs_2=0.15;
type
        temp_array=array [x_min..x_max,0..y_max,-1..z_max+1] of real;
        temp_int_array=array [x_min..x_max,0..y_max,-1..z_max+1] of integer;
        surface_array=array[x_min..x_max,0..y_max] of real;
        conductance_array=array[x_min..x_max,0..y_max] of real;
        x_dimension_array=array[x_min..x_max] of real;
        y_dimension_array=array[0..y_max] of real;
        z_dimension_array=array[0..z_max] of real;
        y_dim_int_array=array[0..y_max] of integer;
        surface_int_array=array[x_min..x_max,0..y_max] of integer;
        y_surface_int_array=array[x_min..x_max,0..z_max] of integer;
        x_surface_int_array=array[0..y_max,0..z_max] of integer;
        surface_vec=record
                x,y,z:real;
        end;

        surf_vec_array=array[x_min..x_max,0..y_max] of surface_vec;
        time_array=array[0..its_ppulse] of real;
        key_result=record
                keyhole_depth:real;
                keyhole_width:real;
        end;
        melt_result=record
                melt_depth:real;
                melt_width:real;
        end;
        key_result_array=array[1..runs,1..iters_1_max] of key_result;
        melt_result_array=array[1..runs,1..iters_1_max] of melt_result;
var
        run_no:integer;
        it_no:integer;
        velocity:real;

        key_dim_array:key_result_array;
        melt_dim_array:melt_result_array;
        flag:boolean;
```

```
drop:boolean;
big_its:integer;
i,j,k:integer;
temp_max : record
                    t:real;
                    x,y,z:integer;
end;
c:real;
x_frf:temp_array;
x_frb:temp_array;
y_fr:temp_array;
z_fr:temp_array;

d_k,old_d_k,d_m,old_d_m:real;
time_off:real;
menu_choice:char;
time:real;
dum:integer;
temp:temp_array;
C_b:temp_int_array;
power_dens:surface_array;

x_coord:x_dimension_array;
delta_x_n:x_dimension_array;
delta_x_s:x_dimension_array;
delta_x_cv:x_dimension_array;

max_key_i:y_dim_int_array;
y_coord:y_dimension_array;
delta_y_e:y_dimension_array;
delta_y_w:y_dimension_array;
delta_y_cv:y_dimension_array;

z_coord:z_dimension_array;
delta_z_u:real;
delta_z_d:real;
delta_z_cv:real;

at_n1_by_k:conductance_array;
at_n2_by_rCp:conductance_array;
at_s1_by_k:conductance_array;
at_s2_by_rCp:conductance_array;
at_e_by_k:conductance_array;
at_w_by_k:conductance_array;
at_u_by_k:conductance_array;
at_d_by_k:conductance_array;
atpo_by_rCpTodt:conductance_array;

surface:surface_int_array;
y_surface:y_surface_int_array;
x_surface_f,x_surface_b:x_surface_int_array;
s_x,s_y:surface_array;
unit_norm:surf_vec_array;
mod_un:surface_array;

l:integer;
power_time:array [0..its_ppulse] of real;
maxtime:real;
pulse_time:real;
time_on:real;
delta_t_l_on,delta_t_on:real;

delta_t_l_off,delta_t_off:real;
delta_t,delta_t_l:time_array;
t_1,t_2,t_3:real;
power_1,power_2,power_3:real;
iters_l:integer;
its_1,its_2,its_3:integer;

handle1:text;
handle2:text;
handle3:text;
handle4:text;
handle6:text;
handle7:text;
handle8:text;
temp_dum:real;
sum_dum:real;
i_dum:integer;

{-------------------------------------------------------}
{ ----------parameter functions required for program ----------}
{-------------------------------------------------------}
function absorptivity(cos_phi:real):real;
```

```
var             term_1_num,term_1_den:real;
                term_2_num,term_2_den:real;
                term_1,term_2:real;
begin
                term_1_num:=1+sqr(1-epsilon*cos_phi);
                term_1_den:=1+sqr(1+epsilon*cos_phi);
                term_1:=term_1_num/term_1_den;
                term_2_num:=sqr(epsilon)-2*epsilon*cos_phi+2*sqr(cos_phi);
                term_2_den:=sqr(epsilon)+2*epsilon*cos_phi+2*sqr(cos_phi);
                term_2:=term_2_num/term_2_den;
                absorptivity:=1-(term_1+term_2)/2;
end;
function raise_e(a,b:real):real;
begin
                raise_e:=exp(b*ln(a))
end;
function fv_of_T(temp1:real):real;
begin
                if temp1<temp_v1 then fv_of_T:=0
                else if temp1>temp_v2 then fv_of_T:=1
                else fv_of_T:=(temp1-temp_v1)/(temp_v2-temp_v1);
end;
function fm_of_T(temp1:real):real;
begin
                if temp1<temp_s then fm_of_T:=0
                else if temp1>temp_l then fm_of_T:=1
                else fm_of_T:=(temp1-temp_s)/(temp_l-temp_s);
end;

function s_Li_fi(temp1:real):real;
begin
                s_Li_fi:=latent_heat_melting*fm_of_T(temp1)
                                +latent_heat_fusion*fv_of_T(temp1);
end;
function spec_h_f(temp1:real):real;
begin
                spec_h_f:=600;
end;
function spec_h_bar(temp1:real):real;
begin
                spec_h_bar:=spec_h_f(temp1)+s_Li_fi(temp1)/temp1;
end;
function cond_f(temp1:real):real;
var             cond:real;
begin
                if (temp1-300)<800 then cond:=52.4289-0.01424*(temp1-300)
                                -0.000022858*sqr(temp1-300)
                else if ((temp1-300)<1500) and ((temp1-300)>800) then
                                cond:=21.88+0.0052*(temp1-300)
                else cond:=29.7;
end;
function dens_f(temp1:real):real;
begin
                dens_f:=7000;{8002.68-0.64*(temp1-300);}
end;

function abs_f(i,j,k:integer;temp1:real):real;
const beta=500;
begin
                if k=0 then begin
                                if (temp1-300)<175 then
                                                abs_f:=0.22552+0.00139*(temp1-300)
                                else if (temp1-300<650) and (temp1-300>175) then
                                                abs_f:=0.40689+0.00026*(temp1-300)
                                else abs_f:=0.5703+0.000018*(temp1-300);
                end
                else abs_f:=absorptivity(unit_norm[i,j].z)+abs_2;
end;
function heat_c0_f(temp1:real):real;
begin
                heat_c0_f:=heat_coeff_0*(temp1/temp1);
end;
function heat_cmax_f(temp1:real):real;
begin
                heat_cmax_f:=heat_coeff_max*(temp1/temp1);
end;
{-------------------------------------------------------}
{---------- Interpolation functions for parameters-------------}
{-------------------------------------------------------}
function mean_cond_n(i,j,k:integer):real;
begin
                mean_cond_n:=(cond_f(temp[i,j,k])+cond_f(temp[i+1,j,k]))/2
end;
function mean_cond_s(i,j,k:integer):real;
```

```
begin
        mean_cond_s:=(cond_f(temp[i,j,k])+cond_f(temp[i-1,j,k]))/2
end;
function mean_cond_e(i,j,k:integer):real;
begin
        mean_cond_e:=(cond_f(temp[i,j,k])+cond_f(temp[i,j+1,k]))/2
end;
function mean_cond_w(i,j,k:integer):real;
begin
        mean_cond_w:=(cond_f(temp[i,j,k])+cond_f(temp[i,j-1,k]))/2
end;
function mean_cond_u(i,j,k:integer):real;
begin
        mean_cond_u:=(cond_f(temp[i,j,k])+cond_f(temp[i,j,k+1]))/2
end;
function mean_cond_d(i,j,k:integer):real;
begin
        mean_cond_d:=(cond_f(temp[i,j,k])+cond_f(temp[i+1,j,k-1]))/2;
end;
{-------------------------------------------------------------}
{----------------Initialising grid routines-------------------}
{-------------------------------------------------------------}
procedure init_x;
var     i:integer;
        delta_x_1:real;
        delta_x_2:real;
        delta_x_3:real;
        delta_x_4:real;
begin
        delta_x_1:=(x_fine_length_min-x_length_min)/(x_fine_min-x_min);
        delta_x_2:=(x_length_origin-x_fine_length_min)/(x_origin-x_fine_min);
        delta_x_3:=(x_fine_length_max-x_length_origin)/(x_fine_max-x_origin);
        delta_x_4:=(x_length_max-x_fine_length_max)/(x_max-x_fine_max);

         for i:=x_min to x_fine_min do
                x_coord[i]:=x_length_min+((i-x_min)*delta_x_1);
        for i:=x_fine_min to x_origin do
                x_coord[i]:=x_fine_length_min+((i-x_fine_min)*delta_x_2);
        for i:=x_origin to x_fine_max do
                x_coord[i]:=x_length_origin+((i-x_origin)*delta_x_3);
        for i:=x_fine_max to x_max do
                x_coord[i]:=x_fine_length_max+((i-x_fine_max)*delta_x_4);

        delta_x_s[0]:=delta_x_1;
        for i:=x_min+1 to x_max do
                delta_x_s[i]:=x_coord[i]-x_coord[i-1];

        for i:=x_min to x_max-1 do
                delta_x_n[i]:=x_coord[i+1]-x_coord[i];
        delta_x_n[x_max]:=delta_x_3;

        delta_x_cv[x_min]:=delta_x_1/2;
        for i:=x_min+1 to x_max-1 do
        delta_x_cv[i]:=0.5*(x_coord[i+1]-x_coord[i-1]);
        delta_x_cv[x_max]:=delta_x_3/2;

end;

procedure init_y;
var     j               : integer;
        delta_y_1       : real;
        delta_y_2       : real;
begin
        delta_y_1:=y_fine_length/(y_fine_max-0.5);
        delta_y_2:=(y_length-y_fine_length)/y_coarse_max;

        y_coord[0]:=-delta_y_1/2;
        for j:=1 to y_fine_max do
                y_coord[j]:=(j-0.5)*delta_y_1;
        for j:=0 to y_coarse_max do
                y_coord[j+y_fine_max]:=y_fine_length+j*delta_y_2;

        for j:=0 to y_max-1 do
                delta_y_e[j]:=y_coord[j+1]-y_coord[j];
        delta_y_e[y_max]:=delta_y_2;

        delta_y_w[0]:=delta_y_1;
         for j:=1 to y_max do
                delta_y_w[j]:=y_coord[j]-y_coord[j-1];

        delta_y_cv[0]:=delta_y_1/2;
        for j:=1 to y_max-1 do
                delta_y_cv[j]:=0.5*(y_coord[j+1]-y_coord[j-1]);
        delta_y_cv[y_max]:=delta_y_2/2;
```

```
end;
procedure init_z;
var             k:integer;
                delta_z:real;
begin
                delta_z:=z_length_max/z_max;
                delta_z_u:=delta_z;
                delta_z_d:=delta_z;
                delta_z_cv:=delta_z;

                for k:=0 to z_max do
                        z_coord[k]:=k*delta_z;
end;

function length(i,j:integer):real;
begin
                length:=sqrt ( sqr(y_coord[j]) + sqr(x_coord[i]) );
end;
procedure init_power_dens;
var             i,j:integer;
                p_dens:real;
                t:real;
                l:integer;
begin
                p_dens:=1/(pi*beam_radius*beam_radius);
                for i:=x_min to x_max do begin
                        for j:=0 to y_max do
                                if length(i,j)<beam_radius then begin
                                        power_dens[i,j]:=p_dens{*exp(sqr(length(i,j)/beam_radius))};
                                end
                        else power_dens[i,j]:=0{p_dens*exp(sqr(length(i,j)/beam_radius))};
                end;
end;

procedure init_cond;
var     i,j:integer;
begin
                for i:= x_min+1 to x_max-1 do begin
                        for j:=1 to y_max-1 do begin
                                at_n1_by_k[i,j]:=delta_y_cv[j]*delta_z_cv/delta_x_n[i];
                                at_n2_by_rCp[i,j]:=-Velocity*delta_x_cv[i]*delta_y_cv[j]*delta_z_cv
                                        / (delta_x_n[i]+delta_x_s[i]);
                                at_s1_by_k[i,j]:=delta_y_cv[j]*delta_z_cv/delta_x_s[i];
                                at_s2_by_rCp[i,j]:=-at_n2_by_rCp[i,j];
                                at_e_by_k[i,j]:=(delta_x_cv[i]*delta_z_cv)/delta_y_e[j];
                                at_w_by_k[i,j]:=(delta_x_cv[i]*delta_z_cv)/delta_y_w[j];
                                at_u_by_k[i,j]:=(delta_x_cv[i]*delta_y_cv[j])/delta_z_u;
                                at_d_by_k[i,j]:=(delta_x_cv[i]*delta_y_cv[j])/delta_z_d;
                                atpo_by_rCpTodt[i,j]:=delta_x_cv[i]*delta_y_cv[j]*delta_z_cv;
                        end;
                end;
end;



procedure init_temp;
var             i,j,k:integer;
begin
                for i:=x_min to x_max do begin
                        for j:=0 to y_max do begin
                                surface[i,j]:=0;
                                mod_un[i,j]:=1;
                                s_x[i,j]:=0;
                                s_y[i,j]:=0;
                                unit_norm[i,j].x:=0;
                                unit_norm[i,j].y:=0;
                                unit_norm[i,j].z:=1;
                                for k:=-1 to z_max+1 do begin
                                        temp[i,j,k]:=ambient_temp;
                                        C_b[i,j,k]:=0;
                                end;
                        end;
                end;
                writeln('past here');
                for i:=x_min to x_max do
                        for j:=0 to y_max do
                                C_b[i,j,-1]:=1;

                for i:=x_min to x_max do
                        for k:=0 to z_max do
                                y_surface[i,k]:=0;

                for j:=0 to y_max do
```

```
                    for kc:=0 to z_max do begin
                        x_surface_f[j,k]:=0;
                        x_surface_b[j,k]:=0;
                    end;

        for j:=0 to y_max do
                    max_key_i[j]:=0;

        for i:=x_min to x_max do
                    for j:=0 to y_max do
                        for kc:=-1 to z_max+1 do begin
                            x_frf[i,j,k]:=1;
                            x_frb[i,j,k]:=1;
                            y_fr[i,j,k]:=1;
                            z_fr[i,j,k]:=1;
                        end;
end;
{-------------------------------------------------------------}
{-----------These are the functions used in calculating the temperatures----------}
{-------------------------------------------------------------}
function f_rxf(i,j,k:integer):real;
var     m:real;
begin
        f_rxf:=1
        {if (temp[i,j,k]<temp_v2) and (temp[i-1,j,k]<temp_v2) then f_rxf:=1
        else if (temp[i,j,k]<temp_v2) and (temp[i-1,j,k]>temp_v2) then begin
                m:=1/(temp[i-1,j,k]-temp[i,j,k]);
                f_rxf:=m*(temp_v2+0.5*temp[i-1,j,k]-1.5*temp[i,j,k]);
        end
        else f_rxf:=0.5;}
end;
function f_rxb(i,j,k:integer):real;
var     m:real;
begin
        f_rxb:=1
        {if (temp[i,j,k]<temp_v2) and (temp[i+1,j,k]<temp_v2) then f_rxb:=1
        else if (temp[i,j,k]<temp_v2) and (temp[i+1,j,k]>temp_v2) then begin
                m:=1/(temp[i+1,j,k]-temp[i,j,k]);
                f_rxb:=m*(temp_v2+0.5*temp[i+1,j,k]-1.5*temp[i,j,k]);
        end
        else f_rxb:=0.5;}
end;
function f_ry(i,j,k:integer):real;
var     m:real;
begin
        f_ry:=1
        {if (temp[i,j,k]<temp_v2) and (temp[i,j-1,k]<temp_v2) then f_ry:=1
        else if (temp[i,j,k]<temp_v2) and (temp[i,j-1,k]>temp_v2) then begin
                m:=1/(temp[i,j-1,k]-temp[i,j,k]);
                f_ry:=m*(temp_v2+0.5*temp[i,j-1,k]-1.5*temp[i,j,k]);
        end
        else f_ry:=0.5;}
end;
function f_rz(i,j,k:integer):real;
var     m:real;
begin
        f_rz:=1;
        {if k=0 then f_rz:=0.5 else begin
                if (temp[i,j,k]<temp_v2) and (temp[i,j,k-1]<temp_v2) then f_rz:=1
                else if (temp[i,j,k]<temp_v2) and (temp[i,j,k-1]>temp_v2) then begin
                        m:=1/(temp[i,j,k-1]-temp[i,j,k]);
                        f_rz:=m*(temp_v2+0.5*temp[i,j,k-1]-1.5*temp[i,j,k]);
                end
                else f_rz:=0.5;
        end;}
end;
function at_n2(i,j,k:integer):real;
begin
        at_n2:=(dens_f(temp[i,j,k])*spec_h_bar(temp[i,j,k])*Velocity
                        *delta_x_cv[i]*delta_y_cv[j]*delta_z_cv)/(delta_x_n[i]+delta_x_s[i]);
end;
function at_n1(i,j,k:integer):real;
begin
        if (k=0) or ((C_b[i+1,j,k]=0) and (C_b[i-1,j,k]=0)) then at_n1:=at_n1_by_k[i,j]*mean_cond_n(i,j,k)
        else if C_b[i-1,j,k]=1 then at_n1:=at_n1_by_k[i,j]*mean_cond_n(i,j,k)/x_frf[i,j,k]
        else if C_b[i+1,j,k]=1 then at_n1:=0;{at_n1_by_k[i,j]*mean_cond_n(i,j,k)/(2*x_frb[i,j,k])}
end;
function at_n(i,j,k:integer):real;
begin
        at_n:=at_n1(i,j,k)+at_n2(i,j,k);
end;
```

```
function at_s2(i,j,k:integer):real;
begin
        at_s2:=-(dens_f(temp[i,j,k])*spec_h_bar(temp[i,j,k])*Velocity
                *delta_x_cv[i]*delta_y_cv[j]*delta_z_cv)/(delta_x_n[i]+delta_x_s[i])
end;
function at_s1(i,j,k:integer):real;
begin
        if (k=0) or ((C_b[i+1,j,k]=0) and (C_b[i-1,j,k]=0)) then at_s1:=at_s1_by_k[i,j]*mean_cond_s(i,j,k)
        else if C_b[i-1,j,k]=1 then at_s1:={at_s1_by_k[i,j]*mean_cond_s(i,j,k)/(2*x_frf[i,j,k])}
        else if C_b[i+1,j,k]=1 then at_s1:=at_s1_by_k[i,j]*mean_cond_s(i,j,k)/x_frb[i,j,k];
end;
function at_s(i,j,k:integer):real;
begin
        at_s:=at_s1(i,j,k)+at_s2(i,j,k);
end;
function at_u(i,j,k:integer):real;
begin
        if k=0 then at_u:=2*at_u_by_k[i,j]*mean_cond_u(i,j,k)
        else if k<z_max then at_u:=at_u_by_k[i,j]*mean_cond_u(i,j,k)/z_fr[i,j,k]
        else if k=z_max then at_u:=0;
end;
function at_d(i,j,k:integer):real;
begin
        if k=0 then at_d:=0
        else if k=z_max then at_d:=2*at_d_by_k[i,j]*mean_cond_d(i,j,k)
        else at_d:=at_d_by_k[i,j]*mean_cond_d(i,j,k)*(1-C_b[i,j,k-1]);{/((1+C_b[i,j,k-1])*z_fr[i,j,k]);}
end;
function at_e(i,j,k:integer):real;
begin
        if k=0 then at_e:=at_e_by_k[i,j]*mean_cond_e(i,j,k)
        else at_e:=at_e_by_k[i,j]*mean_cond_e(i,j,k)/y_fr[i,j,k];
end;
function at_w(i,j,k:integer):real;
begin
        if k=0 then at_w:=mean_cond_w(i,j,k)*at_w_by_k[i,j]
        else at_w:=mean_cond_w(i,j,k)*at_w_by_k[i,j]*(1-C_b[i,j-1,k]){}/((1+C_b[i,j-1,k])*y_fr[i,j,k]);}
end;
function at_p(i,j,k:integer):real;
begin
        at_p:=at_n1(i,j,k)+at_s1(i,j,k)+at_u(i,j,k)+at_d(i,j,k)+at_e(i,j,k)+at_w(i,j,k)
                +dens_f(temp[i,j,k])*spec_h_bar(temp[i,j,k])*atpo_by_rCpTodt[i,j]/delta_t[iters_1]
end;
function bt_fn(i,j,k:integer):real;
var     C_bx_over_frx:real;
begin
        if (k=0) then
                bt_fn:=2*delta_x_cv[i]*delta_y_cv[j]*
                        ((power_dens[i,j]*power_time[iters_1]*abs_f(i,j,k,temp[i,j,k]))
                        -heat_c0_f(temp[i,j,k])*(temp[i,j,k]-ambient_temp))
                        +dens_f(temp[i,j,k])*spec_h_bar(temp[i,j,k])*temp[i,j,k]*atpo_by_rCpTodt[i,j]/delta_t[iters_1]
        else if k=z_max then
                bt_fn:=dens_f(temp[i,j,k])*spec_h_bar(temp[i,j,k])*temp[i,j,k]*atpo_by_rCpTodt[i,j]/delta_t[iters_1]
                        -heat_c0_f(temp[i,j,k])*(temp[i,j,k]-ambient_temp)*2*delta_x_cv[i]*delta_y_cv[j]
        else begin
                if (temp[i-1,j,k]<temp_v2) and (temp[i+1,j,k]<temp_v2) then C_bx_over_frx:=0
                else if temp[i-1,j,k]>=temp_v2 then C_bx_over_frx:=C_b[i-1,j,k]/x_frf[i,j,k]
                else C_bx_over_frx:=C_b[i+1,j,k]/x_frb[i,j,k];
                bt_fn:=((power_dens[i,j]*power_time[iters_1]*abs_f(i,j,k,temp[i,j,k])/mod_un[i,j]
                                        -heat_c0_f(temp[i,j,k])*(temp[i,j,k]-ambient_temp))
                                        *((C_b[i,j,k-1]*delta_x_cv[i]*delta_y_cv[j]/(mod_un[i,j]*z_fr[i,j,k]))
                                        +(C_bx_over_frx*delta_y_cv[j]*delta_z_cv*abs(s_x[i,j])/(mod_un[i,j]))
                                        +(C_b[i,j-1,k]*delta_z_cv*delta_x_cv[i]*abs(s_y[i,j])/(mod_un[i,j]*y_fr[i,j,k])))
                                        +dens_f(temp[i,j,k])*spec_h_bar(temp[i,j,k])*temp[i,j,k]*atpo_by_rCpTodt[i,j]/delta_t[iters_1]
        end
end;
function bt_cxy(i,j,k:integer):real;
begin
        bt_cxy:=at_n(i,j,k)*temp[i+1,j,k]+at_s(i,j,k)*temp[i-1,j,k]+
                at_e(i,j,k)*temp[i,j+1,k]+at_w(i,j,k)*temp[i,j-1,k]+bt_fn(i,j,k);
end;
function bt_cxz(i,j,k:integer):real;
begin
        bt_cxz:=at_n(i,j,k)*temp[i+1,j,k]+at_s(i,j,k)*temp[i-1,j,k]+
                at_u(i,j,k)*temp[i,j,k+1]+at_d(i,j,k)*temp[i,j,k-1]+bt_fn(i,j,k);
end;
function bt_cyz(i,j,k:integer):real;
begin
        bt_cyz:=at_e(i,j,k)*temp[i,j+1,k]+at_w(i,j,k)*temp[i,j-1,k]
                +at_u(i,j,k)*temp[i,j,k+1]+at_d(i,j,k)*temp[i,j,k-1]+bt_fn(i,j,k);
end;
{-------------------------------------------------------}
{-----These are the procedures that calculate the temp---------}
{-------------------------------------------------------}
function find_q(a,c,d,p_last,q_last:real):real;
```

```
begin
        find_q:=(d+(c*q_last))/(a-(c*p_last));
end;
function find_p(a,b,c,p_last:real):real;
begin
        find_p:=b/(a-(c*p_last));
end;
procedure dotdma_const_xy(i,j:integer);
var     a,b,c,d:real;
        k:integer;
        p,q:array[-1..z_max+1] of real;
          temp_t:real;
          f_rxf_t,f_rxb_t,m:real;
begin
        k:=surface[i,j]-1;
        if k<0 then begin
                p[k]:=0;
                q[k]:=0;
        end
        else begin
                p[k]:=0;
                q[k]:=temp[i,j,k];
        end;

        for k:=surface[i,j] to z_max-1 do begin
                a:=at_p(i,j,k);
                b:=at_u(i,j,k);
                c:=at_d(i,j,k);
                d:=bt_cxy(i,j,k);
                p[k]:=find_p(a,b,c,p[k-1]);
                q[k]:=find_q(a,c,d,p[k-1],q[k-1]);
        end;
        k:=z_max;
        a:=at_p(i,j,k);
        b:=0;
        c:=at_d(i,j,k);
        d:=bt_cxy(i,j,k);
        q[k]:=find_q(a,c,d,p[k-1],q[k-1]);
        temp[i,j,k]:=q[k];

        for k:=z_max-1 downto surface[i,j] do begin
        temp_t:=temp[i,j,k+1]*p[k]+q[k];
                if (temp_t>temp[i,j,k]) or (power_time[iters_l]<1) then temp[i,j,k]:=temp_t
        end;
end;

procedure dotdma_const_xz(i,k:integer);
var     a,b,c,d:real;
        j:integer;
        p,q:array[0..y_max] of real;
        temp_t:real;
          f_rxf_t,f_rxb_t,m:real;
begin
        j:=y_surface[i,k];
        if j=0 then begin
                p[j]:=1;
                q[j]:=0;
        end
        else begin
                p[j-1]:=0;
                q[j-1]:=temp[i,j-1,k];
                a:=at_p(i,j,k);
                b:=at_e(i,j,k);
                c:=at_w(i,j,k);
                d:=bt_cxz(i,j,k);
                p[j]:=find_p(a,b,c,p[j-1]);
                q[j]:=find_q(a,c,d,p[j-1],q[j-1]);
        end;

        for j:=y_surface[i,k]+1 to y_max-1 do begin
                a:=at_p(i,j,k);
                b:=at_e(i,j,k);
                c:=at_w(i,j,k);
                d:=bt_cxz(i,j,k);
                p[j]:=find_p(a,b,c,p[j-1]);
                q[j]:=find_q(a,c,d,p[j-1],q[j-1]);
        end;
        for j:=y_max-1 downto y_surface[i,k] do begin
                temp_t:=temp[i,j+1,k]*p[j]+q[j];
                if (temp_t>temp[i,j,k]) or (power_time[iters_l]<1) then temp[i,j,k]:=temp_t;
        end;
end;

procedure dotdma_const_yz_nk(j,k:integer);
```

```
var		i:integer;
		a,b,c,d:real;
		p,q:array[x_min..x_max] of real;
		temp_t:real;
begin
		p[x_min]:=0;
		q[x_min]:=300;
		for i:=x_min+1 to x_max-1 do begin
				a:=at_p(i,j,k);
				b:=at_n(i,j,k);
				c:=at_s(i,j,k);
				d:=bt_cyz(i,j,k);
				p[i]:=find_p(a,b,c,p[i-1]);
				q[i]:=find_q(a,c,d,p[i-1],q[i-1]);
		end;
		for i:=x_max-1 downto x_min+1 do begin
				temp_t:=temp[i+1,j,k]*p[i]+q[i];
				if (temp_t>temp[i,j,k]) or (power_time[iters_l]<1) then temp[i,j,k]:=temp_t;
		end;
end;

procedure dotdma_const_yz(j,k:integer);
var		i:integer;
		a,b,c,d:real;
		p,q:array[x_min..x_max] of real;
		temp_t:real;
begin
		if (x_surface_b[j,k]=0) and (x_surface_f[j,k]=0) then dotdma_const_yz_nk(j,k)
		else begin
				i:=x_surface_f[j,k]-1;
				p[i]:=0;
				q[i]:=temp[i,j,k];
				for i:=x_surface_f[j,k] to x_max-1 do begin
						a:=at_p(i,j,k);
						b:=at_n(i,j,k);
						c:=at_s(i,j,k);
						d:=bt_cyz(i,j,k);
						p[i]:=find_p(a,b,c,p[i-1]);
						q[i]:=find_q(a,c,d,p[i-1],q[i-1]);
				end;
				for i:=x_max-1 downto x_surface_f[j,k] do begin
						temp_t:=temp[i+1,j,k]*p[i]+q[i];
						if (temp_t>temp[i,j,k]) or (power_time[iters_l]<1) then temp[i,j,k]:=temp_t;
				end;

				i:=x_surface_b[j,k]+1;
				p[i]:=0;
				q[i]:=temp[i,j,k];
				for i:=x_surface_b[j,k] downto x_min+1 do begin
						a:=at_p(i,j,k);
						b:=at_s(i,j,k);
						c:=at_n(i,j,k);
						d:=bt_cyz(i,j,k);
						p[i]:=find_p(a,b,c,p[i+1]);
						q[i]:=find_q(a,c,d,p[i+1],q[i+1]);
				end;
				for i:=x_min+1 to x_surface_b[j,k] do begin
						temp_t:=temp[i-1,j,k]*p[i]+q[i];
						if (temp_t>temp[i,j,k]) or (power_time[iters_l]<1) then temp[i,j,k]:=temp_t;
				end;
		end;
end;

procedure sym_boundary;
var	i,j,k:integer;
begin
		for i:=x_min to x_max do
				for k:=0 to z_max do
				temp[i,0,k]:=temp[i,1,k];
end;

procedure change_boundary;
var		i,j,k:integer;
		i_dum,j_dum:integer;
		deepest_i_i,deepest_i_d:integer;
begin
		for i:=x_min+1 to x_max-1 do
				for j:=0 to y_max-1 do
						for k:=0 to z_max do
								if (temp[i+1,j,k]>temp_v2) and (temp[i-1,j,k]>temp_v2) and (temp[i,j,k]<temp_v2)
								then temp[i,j,k]:=temp_v2+1;
		for i:=x_min+1 to x_max-1 do begin
		for j:=1 to y_max-1 do begin
				if temp[i,j,surface[i,j]]>temp_v2 then begin
```

```
                        k:=surface[i,j];
                        repeat
                                if temp[i,j,k]<temp_v2 then temp[i,j,k]:=temp_v2+1;
                                j_dum:=0;
                                repeat
                                        if temp[i,j_dum,k]<temp_v2 then begin
                                                if temp[i,j_dum,k]<temp_v2 then
                                                        temp[i,j_dum,k]:=temp_v2+1;
                                                        if surface[i,j_dum]<k+1 then
                                                                surface[i,j_dum]:=k+1
                                        end;
                                        j_dum:=j_dum+1;
                                until j_dum=j;
                                if i<>0 then begin
                                        i_dum:=0;
                                        repeat
                                                if temp[i_dum,j,k]<temp_v2 then begin
                                                        if temp[i_dum,j,k]<temp_v2 then
                                                                temp[i_dum,j,k]:=temp_v2+1;
                                                         if surface[i_dum,j]<k+1 then
                                                                surface[i_dum,j]:=k+1
                                                end;
                                                if i>0 then i_dum:=i_dum+1 else i_dum:=i_dum-1;
                                        until i_dum=i;
                                end;
                                k:=k+1;
                        until (temp[i,j,k]<temp_v2) or (k=z_max);
                         surface[i,j]:=k;
                end;
end;
end;
for i:=x_min+1 to x_max-1 do
                surface[i,0]:=surface[i,1];
for i:=x_min+1 to x_max-1 do
                for j:=1 to y_max-1 do
                        for k:=0 to z_max do
                                if temp[i,j,k]>=temp_v2 then C_b[i,j,k]:=1;

for i:=x_min+1 to x_max-1 do
                for k:=0 to z_max do begin
                        if temp[i,0,k]>temp_v2 then begin
                                j:=y_max;
                                repeat          j:=j-1
                                until (temp[i,j-1,k]>temp_v2) or (j=0);
                                y_surface[i,k]:=j;
                                if y_surface[i,k]<>0 then begin
                                        j_dum:=0;
                                        repeat
                                                if temp[i,j_dum,k]<temp_v2 then temp[i,j_dum,k]:=temp_v2+1;
                                                j_dum:=j_dum+1;
                                        until j_dum=y_surface[i,k];
                                end;
                        end
                        else y_surface[i,k]:=0;
                end;

for j:=1 to y_max-1 do begin
                for k:=0 to z_max do begin
                        if temp[x_surface_f[j,k],j,k]>=temp_v2 then begin
                                i:=x_surface_f[j,k];
                                repeat
                                        {temp[i,j,k]:=temp_v2+1;}
                                        i:=i+1;
                                until temp[i,j,k]<temp_v2;
                                x_surface_f[j,k]:=i;
                        end;
                        if temp[x_surface_b[j,k],j,k]>=temp_v2 then begin
                                i:=x_surface_b[j,k];
                                repeat
                                        i:=i-1;
                                until temp[i,j,k]<temp_v2;
                                x_surface_b[j,k]:=i;
                        end;
                end;
end;

for j:=1 to y_max-1 do
                for k:=0 to z_max do begin
                        if (x_surface_f[j,k]<>0) and (x_surface_b[j,k]<>0) then begin
                                i:=x_surface_b[j,k]+1;
                                repeat
                                        if temp[i,j,k]<temp_v2 then temp[i,j,k]:=temp_v2+1;
                                        i:=i+1;
                                until i=x_surface_f[j,k];
```

```
                                        end;
                                end;

        for k:=0 to z_max do begin
                                x_surface_f[0,k]:=x_surface_f[1,k];
                                x_surface_b[0,k]:=x_surface_b[1,k];
        end;

        for j:=1 to y_max do begin
                                deepest_i_i:=x_min;
                                 for i:=x_min+1 to x_max-1 do
                                        if surface[i,j]>=surface[deepest_i_i,j] then deepest_i_i:=i;
                                deepest_i_d:=x_max;
                                for i:=x_max-1 downto x_min+1 do
                                        if surface[i,j]>=surface[deepest_i_d,j] then deepest_i_d:=i;
                                        if surface[deepest_i_i,j]=0 then max_key_i[j]:=0
                                        else begin
                                                if deepest_i_i=deepest_i_d then max_key_i[j]:=deepest_i_d
                                                else max_key_i[j]:=round((deepest_i_i-deepest_i_d)/2+deepest_i_d);
                                        end;
                                end;
        end;
        max_key_i[0]:=max_key_i[1];

        for i:=x_min+1 to x_max-1 do
                                for j:=1 to y_max-1 do
                                        for k:=0 to z_max do begin
                                                x_frf[i,j,k]:=f_rxf(i,j,k);
                                                x_frb[i,j,k]:=f_rxb(i,j,k);
                                                y_fr[i,j,k]:=f_ry(i,j,k);
                                                z_fr[i,j,k]:=f_rz(i,j,k);
                                                if x_frf[i,j,k]<0.5 then writeln('whoa');
                                        end;
end;

procedure calc_normal;
var     i,j,k,i_dum,k_dum:integer;
        do_rln:boolean;
begin
        do_rln:=false;
        for i:=x_min+1 to x_max-1 do begin
        for j:=1 to y_max-1 do begin
                                        if (surface[i,j]=0) then begin
                                                s_x[i,j]:=0;
                                                s_y[i,j]:=0;
                                                mod_un[i,j]:=sqrt(1+sqr(s_x[i,j])+sqr(s_y[i,j]));
                                                unit_norm[i,j].x:=-s_x[i,j]/mod_un[i,j];
                                                unit_norm[i,j].y:=-s_y[i,j]/mod_un[i,j];
                                                unit_norm[i,j].z:=1/mod_un[i,j];
                                        end
                                        else begin
                                                if surface[i,j]<>surface[i,j-1] then
                                                        s_y[i,j]:=-(z_coord[surface[i,j-1]]-z_coord[surface[i,j]])/delta_y_w[j]
                                                else if (surface[i,j]=surface[i,j-1]) and (y_surface[i,surface[i,j]]<>y_surface[i,surface[i,j]-1]) then
                                                        s_y[i,j]:=delta_z_u/(y_coord[y_surface[i,surface[i,j]]]
                                                                                -y_coord[y_surface[i,surface[i,j]-1]])
                                                else s_y[i,j]:=0;

                                                if i>max_key_i[j] then begin
                                                        if surface[i-1,j]<>surface[i,j] then
                                                                s_x[i,j]:=-(z_coord[surface[i-1,j]]-z_coord[surface[i,j]])/delta_x_s[i]
                                                        else if (surface[i-1,j]=surface[i,j]) and
                                                                (x_surface_f[j,surface[i,j]-1]<>x_surface_f[j,surface[i,j]]) then
                                                s_x[i,j]:=-delta_z_u/(x_coord[x_surface_f[j,surface[i,j]-1]]-x_coord[x_surface_f[j,surface[i,j]]])
                                                        else s_x[i,j]:=0;
                                                end
                                                else if i<max_key_i[j] then begin
                                                        if surface[i+1,j]<>surface[i,j] then
                                                                s_x[i,j]:=(z_coord[surface[i+1,j]]-z_coord[surface[i,j]])/delta_x_n[i]
                                                        else if (surface[i+1,j]=surface[i,j]) and
                                                                (x_surface_b[j,surface[i,j]]<>x_surface_b[j,surface[i,j]-1]) then
                                                s_x[i,j]:=delta_z_u/(x_coord[x_surface_b[j,surface[i,j]]]-x_coord[x_surface_b[j,surface[i,j]-1]])
                                                        else s_x[i,j]:=0;
                                                end
                                                else if i=max_key_i[j] then s_x[i,j]:=0;
                                                mod_un[i,j]:=sqrt(1+sqr(s_x[i,j])+sqr(s_y[i,j]));
                                                unit_norm[i,j].x:=-s_x[i,j]/mod_un[i,j];
                                                unit_norm[i,j].y:=-s_y[i,j]/mod_un[i,j];
                                                unit_norm[i,j].z:=1/mod_un[i,j];
                                        end;
                                end;
        end;
end;
```

```
procedure reset_temp;
var i,j,k:integer;
begin
        for i:=x_min to x_max do
        for j:=0 to y_max do
                for k:=0 to z_max do begin
                        if temp[i,j,k]>=temp_v2 then
                                temp[i,j,k]:=temp_v2+1;
                end;
end;

procedure calc_temp_xy;
var     i,j:integer;
begin
        for i:=x_min+1 to x_max-1 do
                for j:=1 to y_max-1 do
                        dotdma_const_xy(i,j);
                sym_boundary;
                if power_time[iters_l]>1 then begin
                        change_boundary;
                        change_boundary;
                        calc_normal;
                        reset_temp;
                end;
                for j:=1 to y_max-1 do
                        for i:=x_min+1 to x_max-1 do
                                dotdma_const_xy(i,j);
                sym_boundary;
                if power_time[iters_l]>1 then begin
                change_boundary;
                change_boundary;
                calc_normal;
                reset_temp;
        end;
end;

procedure calc_temp_xz;
var     i,k:integer;
begin
        for i:=x_min+1 to x_max-1 do
                for k:=0 to z_max do
                        dotdma_const_xz(i,k);
        sym_boundary;
        if power_time[iters_l]>0 then begin
                change_boundary;
                change_boundary;
                reset_temp;
                calc_normal;
        end;

        for k:=0 to z_max do
                for i:=x_min+1 to x_max-1 do
                        dotdma_const_xz(i,k);
        sym_boundary;
        if power_time[iters_l]>0 then begin
                change_boundary;
                change_boundary;
                reset_temp;
                calc_normal;
        end;
end;

procedure calc_temp_yz;
var     j,k:integer;
begin
        for j:=1 to y_max-1 do
                for k:=0 to z_max do
                        dotdma_const_yz(j,k);
        sym_boundary;
        if power_time[iters_l]>0 then begin
                change_boundary;
                change_boundary;
                reset_temp;
                calc_normal;
        end;

        for k:=0 to z_max do
                for j:=1 to y_max-1 do
                        dotdma_const_yz(j,k);
                                sym_boundary;
        if power_time[iters_l]>0 then begin
                change_boundary;
                change_boundary;
                reset_temp;
```

```
                    calc_normal;
            end;
end;

procedure reset_boundary;
var     i,j,k:integer;
begin
            for i:=x_min to x_max do
                    for j:=0 to y_max do begin
                            if surface[i,j]>0 then begin
                                    k:=0;
                                    repeat
                                            temp[i,j,k]:=temp_v1;
                                            k:=k+1;
                                    until k>=surface[i,j];
                            end;
                            surface[i,j]:=0;
                    end;

            for i:=x_min to x_max do
                    for k:=0 to z_max do
                            y_surface[i,k]:=0;

            for j:=0 to y_max do
                    for k:=0 to z_max do begin
                            x_surface_f[j,k]:=0;
                            x_surface_b[j,k]:=0;
                    end;
            for i:=x_min to x_max do
                    for j:=0 to y_max do begin
                            mod_un[i,j]:=1;
                            s_x[i,j]:=0;
                            s_y[i,j]:=0;
                            unit_norm[i,j].x:=0;
                            unit_norm[i,j].y:=0;
                            unit_norm[i,j].z:=1;
                            for k:=0 to z_max do begin
                                    C_b[i,j,k]:=0;
                                    if temp[i,j,k]>temp_v1 then temp[i,j,k]:=temp_v1;
                            end;
                    end;
end;
procedure calc_temp;
begin
            calc_temp_xz;
            calc_temp_xy;
            calc_temp_yz;
end;

{-------------------------------------------------------}
{-----------output to screen and files----------------}
{-------------------------------------------------------}
function interp(T1,T2,T3,z1:real):real;
begin
            interp:=delta_z_u*(T1-T3)/(T1-T2)+z1;
end;

function y_interp(T1,T2,T3,y1,d_y:real):real;
begin
            y_interp:=d_y*(T1-T3)/(T1-T2)+y1;
end;
procedure output1(time:real);
var         s_temp:real;
            j_dum,k_dum:integer;
            v_k,v_m:real;
            ww1,ww2:real;
            w_k:integer;
begin
            s_temp:=temp[0,0,surface[0,0]];
            if surface[0,0]=0 then begin
                    d_k:=0;
                    v_k:=(d_k-old_d_k)/delta_t_l[iters_l];
                    if temp[0,0,0]<melting_pt then begin
                            d_m:=0;
                            v_m:=(d_m-old_d_m)/delta_t_l[iters_l];
                            ww1:=0.0;
                    end
                    else begin
                            k_dum:=0;
                            repeat k_dum:=k_dum+1;
                            until temp[0,0,k_dum]<melting_pt;
                            d_m:=interp(temp[0,0,k_dum-1],temp[0,0,k_dum],melting_pt,z_coord[k_dum-1]);
                            v_m:=(d_m-old_d_m)/delta_t_l[iters_l];
                            j_dum:=0;
```

```
                                repeat          j_dum:=j_dum+1;
                                until temp[0,j_dum,0]<melting_pt;
                                ww1:=2*y_interp(temp[0,j_dum-1,0],temp[0,j_dum,0],melting_pt,y_coord[j_dum-1],delta_y_w[j_dum]);
                                k_dum:=round(d_m/2);
                                j_dum:=0;
                                repeat          j_dum:=j_dum+1;
                                until temp[0,j_dum,k_dum]<melting_pt;
                                ww2:=2*y_interp(temp[0,j_dum-1,k_dum],temp[0,j_dum,k_dum],melting_pt,y_coord[j_dum-1],delta_y_cv[j_dum]);

                        end
                end
                else begin
                        k_dum:=surface[0,0];
                        d_k:=interp(temp[0,0,k_dum-1],temp[0,0,k_dum],
                                                boiling_pt,z_coord[k_dum-1]);
                        v_k:=(d_k-old_d_k)/delta_t_l[iters_l];
                        k_dum:=0;
                        repeat k_dum:=k_dum+1;
                        until temp[0,0,k_dum]<melting_pt;
                        d_m:=interp(temp[0,0,k_dum-1],temp[0,0,k_dum],
                                melting_pt,z_coord[k_dum-1]);
                        v_m:=(d_m-old_d_m)/delta_t_l[iters_l];
                        j_dum:=0;
                        repeat          j_dum:=j_dum+1;
                        until temp[0,j_dum,0]<melting_pt;
                        writeln('j:=',(j_dum-1):4,'y:=',(2000*y_coord[j_dum-1]):12:8,'t:=',temp[0,j_dum,0]:8:2);
                        ww1:=2*y_interp(temp[0,j_dum-1,0],temp[0,j_dum,0],melting_pt,y_coord[j_dum-1],delta_y_cv[j_dum]);
                        k_dum:=round(d_m/2);
                        j_dum:=0;
                        repeat          j_dum:=j_dum+1;
                        until temp[0,j_dum,k_dum]<melting_pt;
                        ww2:=2*y_interp(temp[0,j_dum-1,k_dum],temp[0,j_dum,k_dum],melting_pt,y_coord[j_dum-1],delta_y_cv[j_dum]);
                end;

                writeln(handle1,time:8:6,tab,power_time[iters_l]:8:2,tab,s_temp:8:2,tab,(ww1*1000):10:8,tab,(ww2*1000):10:8,tab,(d_k*1000):10:8,
                        tab,(d_m*1000):10:8,tab,(v_k*1000):8:6,tab,(v_m*1000):8:6);
                writeln(time:8:6,tab,'p:=',power_time[iters_l]:8:2,tab,'ww1:=',(ww1*1000):10:8,tab,'ww2:=',(ww2*1000):10:8
                        ,tab,'weld d:=',(d_m*1000):10:8);
                old_d_k:=d_k;
                old_d_m:=d_m;
                writeln('run_no:=',run_no:4,'it_no:=',it_no:4);
                if y_surface[0,0]=0 then key_dim_array[run_no,it_no].keyhole_width:=0 else
                        key_dim_array[run_no,it_no].keyhole_width:=2*1000*y_interp(temp[0,y_surface[0,0]-1,k_dum],
                                temp[0,y_surface[0,0],k_dum],boiling_pt,y_coord[y_surface[0,0]-1],delta_y_cv[y_surface[0,0]-1]);
                key_dim_array[run_no,it_no].keyhole_depth:=d_k*1000;
                writeln('key_dim_array[run_no,it_no].keyhole_width:=',key_dim_array[run_no,it_no].keyhole_width:12:8);
                writeln('key_dim_array[run_no,it_no].keyhole_depth:=',key_dim_array[run_no,it_no].keyhole_depth:12:8);
                melt_dim_array[run_no,it_no].melt_depth:=d_m*1000;
                melt_dim_array[run_no,it_no].melt_width:=ww1*1000;
                writeln('melt_dim_array[run_no,it_no].melt_depth:=',melt_dim_array[run_no,it_no].melt_depth:12:8);
                writeln('elt_dim_array[run_no,it_no].melt_width:=',melt_dim_array[run_no,it_no].melt_width:12:8);
end;

procedure output2(time:real);
var             i,j,k:integer;
                melt_p,surface_p:array[x_min..x_max,0..y_max] of real;
begin
                for i:=x_min to x_max do begin
                        for j:=0 to y_max do begin
                                if surface[i,j]=0 then surface_p[i,j]:=0
                                else surface_p[i,j]:=interp(temp[i,j,surface[i,j]-1],temp[i,j,surface[i,j]],boiling_pt,z_coord[surface[i,j]-1]);
                                 if temp[i,j,0]<melting_pt then melt_p[i,j]:=0
                                else begin
                                        k:=0;
                                        repeat k:=k+1;
                                        until temp[i,j,k]<melting_pt;
                                        melt_p[i,j]:=interp(temp[i,j,k-1],temp[i,i,k],melting_pt,z_coord[k-1]);
                                end;
                        end;
                end;
                for i:=x_min to x_max do begin
                        for j:=0 to y_max do begin
                                writeln(handle2,(x_coord[i]*1000):8:8,tab,(y_coord[j]*1000):8:8,tab,(surface_p[i,j]*1000):7:5);
                                writeln(handle3,(x_coord[i]*1000):8:8,tab,(y_coord[j]*1000):8:8,tab,(melt_p[i,j]*1000):7:5);
                         end;
                 end;
                for i:=x_min to x_max do begin
                        for j:=0 to y_max do begin
                                writeln(handle2,(x_coord[i]*1000):8:6,tab,(-y_coord[j]*1000):8:8,tab,(surface_p[i,j]*1000):7:5);
                                writeln(handle3,(x_coord[i]*1000):8:8,tab,(-y_coord[j]*1000):8:8,tab,(melt_p[i,j]*1000):7:5);
                        end;
                end;
end;
```

```
procedure output3(time:real);
var		j_dum,k_dum:integer;
		wk_max:integer;
		k:integer;
		weld_width:real;
begin
		if temp[0,0,0]<melting_pt then begin
				writeln(handle6,'time:=',time:8:5);
				writeln(handle6,'nought here');
				writeln(handle6);
				writeln(handle6);
		end
		else begin
				writeln(handle6,'time:=',time:8:5);
				writeln(handle6);
				k_dum:=0;
				repeat		k_dum:=k_dum+1;
				until (temp[0,0,k_dum]<melting_pt) or (k_dum>=z_max);
				if k_dum=z_max then wk_max:=z_max
				else wk_max:=k_dum-1;
				for k:=0 to wk_max do begin
						j_dum:=0;
						repeat		j_dum:=j_dum+1;
						until temp[0,j_dum,k]<melting_pt;
						weld_width:=y_interp(temp[0,j_dum-1,k],temp[0,j_dum,k],melting_pt,
												y_coord[j_dum-1],delta_y_w[j_dum]);
						writeln(handle6,k:4,tab,(z_coord[k]*1000):8:5,tab,(2*weld_width*1000):8:5);
				end;
				writeln(handle6);
				writeln(handle6);
		end;
end;

procedure output4;
var		i,j,k:integer;
begin
		for i:=x_fine_min to x_fine_max do
				for j:=0 to y_fine_max do
						writeln(handle4,x_coord[i]:12:8,tab,y_coord[j]:12:8,tab,temp[i,j,0]:8:2);
end;

procedure major_output;
var run_dum,it_dum:integer;
begin
		for it_dum:=1 to iters_l_max do begin
				for run_dum:=1 to runs do begin
						write(handle7,key_dim_array[run_dum,it_dum].keyhole_depth:12:8,tab,
								key_dim_array[run_dum,it_dum].keyhole_width:12:8);
						write(handle8,melt_dim_array[run_dum,it_dum].melt_depth:12:8,tab,melt_dim_array[run_dum,it_dum].melt_width:12:8)
				end;
				writeln(handle7);
				writeln(handle8);
		end;
end;


procedure set_power_rect(power,t_on,t_off,t_p:real);
var		mean_power,energy:real;
		l:integer;
		time_dum:real;
begin
		mean_power:=power*t_on/t_p;
		energy:=power*t_on;
		writeln('mean power:=',mean_power:12:4);
		writeln('energy ppulse:=',energy:12:4);
		writeln('time off:=',t_off);

		l:=0;
		time_dum:=0;
		repeat
				if time_dum<=t_on then power_time[l]:=power
				else power_time[l]:=0;
				l:=l+1;
				time_dum:=time_dum+delta_t_l[l];
		until l>=its_ppulse;
end;

procedure set_power_step(P1,P2,t1,t2,t_off,t_p:real);
var		mean_power,energy:real;
		l:integer;
		time_dum:real;
begin
		mean_power:=(P1*t1+P2*t2)/t_p;
		energy:=P1*t1+P2*t2;
```

```
                writeln('mean power:=',mean_power:12:4);
                writeln('energy ppulse:=',energy:12:4);
                readln;
                l:=0;
                time_dum:=0;
                repeat
                        if time_dum<=t1 then power_time[l]:=P1
                        else if (time_dum>t1) and (time_dum<=(t1+t2)) then power_time[l]:=P2
                        else power_time[l]:=0;
                        l:=l+1;
                        time_dum:=time_dum+delta_t_l[l];
                until l>=its_ppulse;
end;

procedure set_power_spulse(P1,P2,t1,t2,t_p:real);
var             mean_power,energy:real;
                l:integer;
                time_dum:real;
begin
                mean_power:=((t1/2)*(P1+P2)+(t2*P2))/t_p;
                energy:=((t1/2)*(P1+P2)+(t2*P2));
                writeln('mean power:=',mean_power:12:4);
                writeln('energy ppulse:=',energy:12:4);
                readln;
                l:=0;
                time_dum:=0;
                repeat
                        if time_dum<=t1/2 then power_time[l]:=P2+(2*(P1-P2)/t1)*time_dum
                        else if (time_dum>t1/2) and (time_dum<=t1) then power_time[l]:=(2*(P2-P1)/t1)*(time_dum-t1/2*(1-(P1/(P2-P1))))
                        else if (time_dum>t1) and (time_dum<=(t1+t2)) then power_time[l]:=P2
                        else power_time[l]:=0;
                        l:=l+1;
                        time_dum:=time_dum+delta_t_l[l];
                until l>=its_ppulse;
end;

procedure input_params;
var             shape_choice:integer;
                time_off:real;
                l:integer;
begin
                writeln('Please enter the pulse shape');
                writeln('1) Recatngular');
                writeln('2) Super pulse');
                writeln('3) Step up');
                writeln('4) Step down');
                readln(shape_choice);

                if shape_choice=1 then begin
                        writeln('Please enter the peak power');
                        readln(power_1);
                        writeln('Please enter time on');
                        readln(t_1);
                        writeln('Please enter pulse time');
                        readln(pulse_time);
                        time_off:=pulse_time-t_1;
                        writeln('time_off:=',time_off:8:2);
                        time_on:=t_1;
                        its_1:=pulse_on_its;
                        its_2:=0;
                        its_3:=pulse_off_its;
                        delta_t_l_on:=time_on/pulse_on_its;
                        delta_t_l_off:=time_off/pulse_off_its;
                        delta_t_on:=delta_t_l_on/6;
                        delta_t_off:=delta_t_l_off/6;
                        for l:=0 to its_ppulse do begin
                                if l<=(its_1+its_2) then begin
                                        delta_t[l]:=delta_t_on;
                                        delta_t_l[l]:=delta_t_l_on;
                                end
                                else begin
                                        delta_t[l]:=delta_t_off;
                                        delta_t_l[l]:=delta_t_l_off;
                                end;
                        end;
                        set_power_rect(power_1,t_1,time_off,pulse_time);
                end
                else if shape_choice=2 then begin
                        writeln('Please enter ist power (peak power)');
                        readln(power_1);
                        writeln('Please enter 2nd power');
                        readln(power_2);
                        writeln('Please enter time for 1st power');
                        readln(t_1);
```

```
                writeln('Please enter time for 2nd power');
                readln(t_2);
                time_on:=t_1+t_2;
                writeln('Please enter pulse time');
                readln(pulse_time);
                time_off:=pulse_time-t_1;
                its_1:=round((t_1/time_on)*pulse_on_its);
                its_2:=pulse_on_its-its_1;
                its_3:=pulse_off_its;
                delta_t_l_on:=time_on/pulse_on_its;
                delta_t_l_off:=time_off/pulse_off_its;
                delta_t_on:=delta_t_l_on/6;
                delta_t_off:=delta_t_l_off/6;
                for l:=0 to its_ppulse do begin
                 if l<=(its_1+its_2) then begin
                                delta_t[l]:=delta_t_on;
                                delta_t_l[l]:=delta_t_l_on;
                end
                else begin
                                delta_t[l]:=delta_t_off;
                                delta_t_l[l]:=delta_t_l_off;
                end;
                end;
                set_power_spulse(power_1,power_2,t_1,t_2,pulse_time);
        end
        else if (shape_choice=3) or (shape_choice=4) then begin
                writeln('Please enter ist power');
                readln(power_1);
                writeln('Please enter 2nd power');
                readln(power_2);
                writeln('Please enter time for 1st power');
                readln(t_1);
                writeln('Please enter time for 2nd power');
                readln(t_2);
                time_on:=t_1+t_2;
                writeln('Please enter pulse time');
                readln(pulse_time);
                time_off:=pulse_time-t_1;
                its_1:=round((t_1/time_on)*pulse_on_its);
                its_2:=pulse_on_its-its_1;
                 its_3:=pulse_off_its;
                delta_t_l_on:=time_on/pulse_on_its;
                delta_t_l_off:=time_off/pulse_off_its;
                delta_t_on:=delta_t_l_on/6;
                delta_t_off:=delta_t_l_off/6;
                for l:=0 to its_ppulse do begin
                if l<=(its_1+its_2) then begin
                                delta_t[l]:=delta_t_on;
                                delta_t_l[l]:=delta_t_l_on;
                end
                else begin
                                delta_t[l]:=delta_t_off;
                                 delta_t_l[l]:=delta_t_l_off;
                end;
        end;
        set_power_step(power_1,power_2,t_1,t_2,time_off,pulse_time);
        end;
end;

procedure init_prog;
begin
        rewrite(handle7,'key_dimy41.dat');
        rewrite(handle8,'melt_dimy41.dat');
end;

procedure initialise;
begin
        d_k:=0;
        old_d_k:=0;
        d_m:=0;
        old_d_m:=0;

        init_x;
        init_y;
        init_z;
        init_temp;

        {input_params;}
        power_1:=3000;
        power_2:=2416.66667;
        t_1:=0.004;
        t_2:=0.012;
        pulse_time:=0.1;
        time_on:=t_1+t_2;
```

```
        time_off:=pulse_time-(t_1+t_2);
        its_1:=round((t_1/time_on)*pulse_on_its);
        its_2:=pulse_on_its-its_1;
        its_3:=pulse_off_its;
        delta_t_l_on:=time_on/pulse_on_its;
        delta_t_l_off:=time_off/pulse_off_its;
        delta_t_on:=delta_t_l_on/6;
        delta_t_off:=delta_t_l_off/6;
        writeln('dt_on:=',delta_t_on:12:8,' dt_off:=',delta_t_off:12:8);
        c:=0;
        for l:=0 to its_ppulse do begin
         if l<=pulse_on_its then begin
                delta_t[l]:=delta_t_on;
                delta_t_l[l]:=delta_t_l_on;
         end
        else begin
                delta_t[l]:=delta_t_off;
                delta_t_l[l]:=delta_t_l_off;
         end;
    end;

        set_power_spulse(power_1,power_2,t_1,t_2,pulse_time);
        init_power_dens;
        init_cond;
        init_temp;
        case run_no of
                1:begin
                        velocity:=0.008;
                        rewrite(handle1,'/tmp/clucy1/infoy438');
                        writeln(handle1,'hi');
                        rewrite(handle2,'/tmp/clucy1/keyy436p8.dat');
                        rewrite(handle3,'/tmp/clucy1/melty436p8.dat');
                        rewrite(handle4,'/tmp/clucy1/tempsy436p8.dat');
                        rewrite(handle6,'/tmp/clucy1/welddimy436p8');
                        writeln('vel:=8');
                end;
                2:begin
                        velocity:=0.012;
                        rewrite(handle1,'/tmp/clucy1/infoy436p12');
                        rewrite(handle2,'/tmp/clucy1/keyy436p12.dat');
                        rewrite(handle3,'/tmp/clucy1/melty436p12.dat');
                        rewrite(handle4,'/tmp/clucy1/tempsy436p12.dat');
                        rewrite(handle6,'/tmp/clucy1/welddimy436p12');
                end;
                3:begin
                        velocity:=0.016;
                        rewrite(handle1,'/tmp/clucy1/infoy436p16');
                        rewrite(handle2,'/tmp/clucy1/keyy436p16.dat');
                        rewrite(handle3,'/tmp/clucy1/melty436p16.dat');
                        rewrite(handle4,'/tmp/clucy1/tempsy436p16.dat');
                        rewrite(handle6,'/tmp/clucy1/welddimy436p16');
                end;
                4:begin
                        velocity:=0.02;
                        rewrite(handle1,'/tmp/clucy1/infoy436p20');
                        rewrite(handle2,'/tmp/clucy1/keyy436p20.dat');
                        rewrite(handle3,'/tmp/clucy1/melty436p20.dat');
                        rewrite(handle4,'/tmp/clucy1/tempsy436p20.dat');
                        rewrite(handle6,'/tmp/clucy1/welddimy436p20');
                end;
                5:begin
                        velocity:=0.032;
                        rewrite(handle1,'/tmp/clucy1/infoy436p32');
                        rewrite(handle2,'/tmp/clucy1/keyy436p32.dat');
                         rewrite(handle3,'/tmp/clucy1/melty436p32.dat');
                         rewrite(handle4,'/tmp/clucy1/tempsy436p32.dat');
                         rewrite(handle6,'/tmp/clucy1/welddimy436p32');
                end;
        end;
        for l:=0 to its_ppulse do
                writeln('l:=',l:4,' dtl:=',delta_t_l[l]:12:8,'power=',power_time[l]:12:8);
end;

procedure screen_output;
begin
        writeln('here we are');
        if iters_l/1=round(iters_l/1) then begin
                writeln('i:=',iters_l:4,' iters_l:=',iters_l:5,'  power:=',power_time[iters_l]:8:4,
                        ' s:=',surface[x_origin,0]:4,' temp:=',temp[x_origin,0,surface[x_origin,0]]:8:4);
                writeln({handle5,}time:8:4,'z  ',temp[x_origin,0,0]:8:4,' ',temp[x_origin,0,10]:8:4,' ',temp[x_origin,0,20]:8:4
                        ,temp[x_origin,0,30]:8:4,' ',temp[x_origin,0,40]:8:4,' ',temp[x_origin,0,50]:8:4
                        ,tab,temp[x_origin,0,30]:8:4);
                writeln({handle5,}time:8:4,'y ',temp[x_origin,0,0]:8:4,' ',temp[x_origin,10,0]:8:4,' ',temp[x_origin,20,0]:8:4,' ',temp[x_origin,30,0]:8:4
                        {,temp[x_origin,40,0]:8:4,' ',temp[x_origin,50,0]:8:4OD});
```

```
                writeln({handle5,}time:8:4,'x- ',temp[x_origin+0,2,0]:8:4,' ',temp[x_origin-10,2,0]:8:4,' '
                        ,' ',temp[x_origin-20,2,0]:8:4,' ',temp[x_origin-30,2,0]:8:4);
                writeln({handle5,}time:8:4,'x+ ',temp[x_origin+0,0,0]:8:4,' ',temp[x_origin+10,0,0]:8:4,' '
                        ,temp[x_origin+20,2,0]:8:4,' ',temp[x_origin+30,2,0]:8:4);
                writeln('y436p(8)a2');
                writeln('i:=',temp_max.x:4,' j:=',temp_max.y:4,' k:=',temp_max.z:4,' t:=',temp_max.t:8:2);
                writeln('y_surf:=',y_surface[0,0]:4,' x_surf_b:=',x_surface_b[0,0]
                        ,' x_surf_f:=',x_surface_f[0,0]);
                output1(time);
                output3(time);
        end;
end;

procedure run_sim;
begin
        time:=0;
        it_no:=0;
        for big_its:=1 to pulse_no do begin
                iters_l:=1;
                repeat
                        it_no:=it_no+1;
                        writeln('it_no:=',it_no:6);
                        screen_output;
                        if iters_l=pulse_on_its+1 then begin
                                if big_its=pulse_no then begin
                                        output2(time);
                                        output4;
                                        reset_boundary;
                                end
                                else begin
                                        writeln('in here');
                                        reset_boundary;
                                end;
                        end;
                        calc_temp;
                        writeln('its_l:=',its_l);
                        time:=time+delta_t_l[iters_l];
                        iters_l:=iters_l+1;
                until (iters_l>its_ppulse) or (drop=true);
        end;
end;
procedure close_down;
begin
        close(handle1);
        close(handle2);
        close(handle3);
        close(handle4);
        close(handle6);
end;

procedure close_prog;
begin
        major_output;
        close(handle7);
        close(handle8);
end;
begin
        init_prog;
        for run_no:=1 to runs do begin
                initialise;
                run_sim;
                close_down;
                if (run_no<1) or (run_no>=runs) then exit;
        end;
        close_prog;
end.
```

Appendix 2 Detailed Laser Keyhole Welding Model

The conditions in the keyhole during deep penetration welding with a laser are described in detail at the beginning of chapter 3. The idea of the model is to describe the pressure balance and energy balance at the keyhole wall in order to determine the keyhole dimensions, and the power per unit depth deposited in the keyhole, in order to calculate the penetration depth and the weldpool dimensions. The solution is calculated by applying only an energy balance equation at the keyhole walls, and then calculating the flow within the keyhole given the correct boundary conditions in order to obtain the vapour velocity at the keyhole walls. Once the solution to the flow within the keyhole has converged, the energy balance is again applied to the keyhole walls in order to obtain the new keyhole dimensions in terms of conductive and ablative energy losses. The flow equations for the keyhole are then solved again in order to find the new velocities at the keyhole walls, and then the energy balance is applied again at the keyhole. This procedure is repeated until a converged solution is obtained for the keyhole dimensions. The program then calculates the weldpool dimensions.

```
program simplerz(input,output);
const
        tab= chr(9);
        massdens= 0.2244;
        viscoscity= 5.0e-4;
        { program model quantitative behaviour }
        rac_max= 90;
        r_max= rac_max;
        rac_u_max= rac_max-1;
         r_max_length   = 0.001;
        rac_max_length= r_max_length;
        rb_max= 30;
        rb_u_max= rb_max-1;
        r_fine_max_length= 0.0003;
        rb_max_length= r_fine_max_length;

        za_length= 0.004;
        zb_length = 0.002;
        zc_length= 0.004;
        za_max_length= 0.004;
        zb_max_length= 0.006;
        zc_max_length= 0.01;
        z_max_length= zc_max_length;
        za_max= 20;
        zb_max= 40;
        zc_max= 60;
        z_max= zc_max;

        { miscellaneous constants }
        n_iters_1= 5000;{ maximum number of iterations }
        n_iters_2= 5000;

        { ambient pressure }
        p_xs= 10000.0;
        p_ambt= 100000.0;
        ambient_temp=300;
        melting_pt=1675;
         boiling_pt=3000;
        latent_heat_vap=6.3E8;
        delta_T=boiling_pt-ambient_temp;

        {solid properties}
        linking_intensity=0.356;
        conductivity_solid=35.97;
        specific_heat_solid=753;
        density_solid=7200;
        density=0.22;
         pi=3.14159;
        surface_ten=2;
        max_order=4;
        epsilon = 0.08;

type
        r_point_int= integer;  {[0..r_max]}
        z_point_int= integer;  {[0..z_max]}
        i_point_int= integer;  {[0 .. n_iters_1]}
        g_point_int= integer;  {[0 .. 100]}

        r_point_grid= array[0..r_max] of real;
        z_point_grid= array[0..z_max] of real;
        g_point_grid= array[0..100] of real;
        s_point_grid= array[0..r_max, 0..z_max] of real;
        cond_array= array[0..r_max, 0..z_max] of real;

        keyhole_coord = record
                        r_c:real;
                        z_c:real;
        end;
        array_of_keyhole_coords=array[0..20] of keyhole_coord;
        weldbead_coord = record
                        x,y,z:real;
        end;
        array_of_weldbead_coords=array[0..20] of weldbead_coord;
var
        alpha:real;
        k:integer;
        temp:s_point_grid;
        uc_u,uc_d,uc_n,uc_s:cond_array;
        wc_u,wc_d,wc_n,wc_s:cond_array;
        tc_u,tc_d,tc_n,tc_s:cond_array;
        tc_p: cond_array;
        bt_top_a,bt_top_b,bt_bot :array[0..r_max] of real;
```

```
        g_pt:g_point_int;
        i_iti_point_int;
        j_it:integer;
        counter:integer;
        relax_count:integer;

        ru_max_initial,rw_max_initial,rp_max_initial:integer;
        zu_max,zw_max,zp_max:array[0..r_max] of integer;
        zp_min_td,zu_min_td,zw_min_td:array[0..r_max] of integer;


        { finite difference mesh }
        radius_max:real;
        ru_max,rw_max,rp_max:array[0..z_max] of integer;
        pw_r_coord:r_point_grid;              { r coordinate of grid point      }
        pw_delta_r_n:r_point_grid;            { r spacing between grid points   }
        pw_delta_r_s:r_point_grid;            { r spacing between grid points   }
        pw_r_face_coord:r_point_grid;         { r coordinate of control face    }
        pw_delta_r_cv:r_point_grid;           { r spacing between control faces }
        u_r_coord:r_point_grid;               { r coordinate of grid point      }
        u_delta_r_n:r_point_grid;             { r spacing between grid points   }
        u_delta_r_s:r_point_grid;             { r spacing between grid points   }
        u_r_face_coord:r_point_grid;          { r coordinate of control face    }
        u_delta_r_cv:  r_point_grid;          { r spacing between control faces }

        pu_z_coord:z_point_grid;              { z coordinate of grid point      }
        pu_delta_z_u:z_point_grid;            { z spacing between grid points   }
        pu_delta_z_d:z_point_grid;            { z spacing between grid points   }
        pu_z_face_coord:z_point_grid;         { z coordinate of control face    }
        pu_delta_z_cv:z_point_grid;           { z spacing between control faces }

        w_z_coord:    z_point_grid;           { z coordinate of grid point      }
        w_delta_z_u:  z_point_grid;           { z spacing between grid points   }
        w_delta_z_d:  z_point_grid;           { z spacing between grid points   }
        w_z_face_coord:z_point_grid;          { z coordinate of control face    }
        w_delta_z_cv:z_point_grid;            { z spacing between control faces }

  { dependent variables }

        u_vel:s_point_grid;                   { radial velocity field     }
        w_vel:s_point_grid;                   { axial velocity field      }
        pressure:s_point_grid;                { pressure field                }
        p_cor:s_point_grid;                   { pressure correction field }

        e_max:real;                           { maximum endpoint }
        e_min:real;                           { minimum endpoint }
        velocity:real;
        max_power:real;

function u_interp(i,k:integer):real;
begin
        u_interp:=(u_vel[i,k]+u_vel[i-1,k])/2;
end;
function w_interp(i,k:integer):real;
begin
        w_interp:=(w_vel[i,k]+w_vel[i,k-1])/2;
end;
function mid_u_n(i,k:integer):real;
var     f_e_near,f_e_far:real;
begin
        f_e_near:=(u_delta_r_cv[i]/2)/u_delta_r_n[i];
        f_e_far:=(u_delta_r_cv[i+1]/2)/u_delta_r_n[i];
        mid_u_n:=f_e_near*u_vel[i+1,k]+f_e_far*u_vel[i,k];
end;
function mid_u_s(i,k:integer):real;
 var    beam_radius:real;
        keyhole:array_of_keyhole_coords;
        weldbead:array_of_weldbead_coords;
        f_e_near,f_e_far:real;
begin
        f_e_near:=(u_delta_r_cv[i]/2)/u_delta_r_s[i];
        f_e_far:=(u_delta_r_cv[i-1]/2)/u_delta_r_s[i];
        mid_u_s:=f_e_near*u_vel[i-1,k]+f_e_far*u_vel[i,k];
end;
function mid_w_u(i,k:integer):real;
var     f_e_near,f_e_far:real;
begin
        f_e_near:=(w_delta_z_cv[k]/2)/w_delta_z_u[k];
        f_e_far:=(w_delta_z_cv[k+1]/2)/w_delta_z_u[k];
        mid_w_u:=f_e_near*w_vel[i,k+1]+f_e_far*w_vel[i,k];
end;
```

```
function mid_w_d(i,k:integer):real;
var     f_e_near,f_e_far:real;
begin
                f_e_near:=(w_delta_z_cv[k]/2)/w_delta_z_d[k];
                f_e_far:=(w_delta_z_cv[k-1]/2)/w_delta_z_d[k];
                mid_w_d:=f_e_near*w_vel[i,k-1]+f_e_far*w_vel[i,k];
end;
{-------------------------------------------------------------------}
{-------------------- s p e c i a l    f u n c t i o n s ------------------}
{-------------------------------------------------------------------}
function max( x, y: real ): real;
begin
                if x < y then max:=y else max:=x;
end {max};

function min0( x: real ): real;
begin
                if x < 0.0 then min0:=0.0 else min0:=x;
end {min0};

function tan(a:real):real;
begin
                tan:=sin(a)/cos(a);
end;

function raise(a,b:real):real;
{calculates a^b}
begin
                if b=0 then raise:=1
                else raise:=exp(b*ln(a));
end;
function raiseneg(a:real;b:integer):real;
{calcs a^b where a is a negative number}
var             j:integer;
                temp_raise:real;
begin
                temp_raise:=a;
                if b>1 then
                for j:=2 to b do
                                temp_raise:=a*temp_raise;
                                raiseneg:=temp_raise;
end;

{-------------------------------------------------------------------}
{-----The following functions are used in the evaluation of the heat balance------}
{-------------------------------------------------------------------}
function absorptivity(theta:real):real;
var             term_1_num,term_1_den:real;
                term_2_num,term_2_den:real;
                term_1,term_2:real;
begin
                {    absorptivity:=(theta/theta)*0.22;}
                term_1_num:=1+sqr(1-epsilon*cos(theta));
                term_1_den:=1+sqr(1+epsilon*cos(theta));
                term_1:=term_1_num/term_1_den;
                term_2_num:=sqr(epsilon)-2*epsilon*cos(theta)+2*sqr(cos(theta));
                term_2_den:=sqr(epsilon)+2*epsilon*cos(theta)+2*sqr(cos(theta));
                term_2:=term_2_num/term_2_den;
                absorptivity:=1-(term_1+term_2)/2;
end;

{-------------------------------------------------------------------}
{-----------------Modified Bessel function order-----------------}
{-------------------------------------------------------------------}
function Mod_Bessel_1_0 ( X: real ): real;
var             y,z:  real;
                 i:integer;
                 y_to_the_2_times:array[0..6] of real;
                 z_to_the:array [0..8] of real;
begin
                y :=    X / 3.75;
                z :=    3.75 / X;
                y_to_the_2_times[0]:=1;
                for i:=1 to 6 do
                                y_to_the_2_times[i]:=sqr(y)*y_to_the_2_times[i-1];
                z_to_the[0]:=1;
                for i:=1 to 8 do
                                z_to_the[i]:=z_to_the[i-1]*z;
                if      X <= 3.75
                then    Mod_Bessel_1_0:=  1.0 + 3.5156229 * y_to_the_2_times[1]
                                + 3.0899424 * y_to_the_2_times[2]
                                + 1.2067492 * y_to_the_2_times[3]
                                + 0.2659732 * y_to_the_2_times[4]
                                + 0.0360768 * y_to_the_2_times[5]
```

```
                        + 0.0045813 * y_to_the_2_times[6]
            else
            Mod_Bessel_1_0:=  exp( X ) * sqrt(1.0 / X) * ( 0.39894228 + 0.01328592 * z
                        + 0.00225319 * z_to_the[2]
                        - 0.00157565 * z_to_the[3]
                        + 0.00916281 * z_to_the[4]
                        - 0.02057706 * z_to_the[5]
                        + 0.02635537 * z_to_the[6]
                        - 0.01647633 * z_to_the[7]
                        + 0.00392377 * z_to_the[8]);
end; {Mod_Bessel_1_0}

function Mod_Bessel_1_1 ( X: real ): real;
var         y, z:   real;
            i:integer;
            y_to_the_2_timeXs:array[0..6] of real;
            z_to_the:array [0..8] of real;
begin
            y :=    X / 3.75;
            z :=    3.75 / X;
            y_to_the_2_times[0]:=1;
            for i:=1 to 6 do
                        y_to_the_2_times[i]:=sqr(y)*y_to_the_2_times[i-1];

            z_to_the[0]:=1;
            for i:=1 to 8 do
                        z_to_the[i]:=z_to_the[i-1]*z;

            if      X <= 3.75
                        then    Mod_Bessel_1_1:=  X * ( 0.5
                        + 0.87890594 * y_to_the_2_times[1]
                        + 0.51498869 * y_to_the_2_times[2]
                        + 0.15084934 * y_to_the_2_times[3]
                        + 0.02658733 * y_to_the_2_times[4]
                        + 0.00301532 * y_to_the_2_times[5]
                        + 0.00032411 * y_to_the_2_times[6] )

            else    Mod_Bessel_1_1:=  exp( X ) * sqrt(1.0 / X) *
                        ( 0.39894228 - 0.03988024 * z
                        - 0.00362018 * z_to_the[2]
                        + 0.00163801 * z_to_the[3]
                        - 0.01031555 * z_to_the[4]
                        + 0.02282967 * z_to_the[5]
                        - 0.02895312 * z_to_the[6]
                        + 0.01787654 * z_to_the[7]
                        - 0.00420059 * z_to_the[8])
end;    {Mod_Bessel_1_1}

function Mod_Bessel_1 ( order: integer; X: real ): real;
var         I:                  integer;
            Buffer:         array [0..max_order] of real;
begin
            if  order = 0  then  Mod_Bessel_1:= Mod_Bessel_1_0 ( X )
            else if  order = 1  then  Mod_Bessel_1:= Mod_Bessel_1_1 ( X )
            else begin
                        Buffer [ 0 ] := Mod_Bessel_1_0 ( X );
                        Buffer [ 1 ] := Mod_Bessel_1_1 ( X );
                        for I:= 2 to order  do
                                    Buffer [ I ] := Buffer [ I-2 ] - ( 2.0 * (I-1) / X ) * Buffer [ I-1 ];
                        Mod_Bessel_1:=  Buffer [ order ]
            end;
end;    {Mod_Bessel_1}

function Mod_Bessel_2_0 ( X: real): real;
var         y, z:   real;
            i:integer;
            y_to_the_2_times:array[0..6] of real;
            z_to_the:array [0..8] of real;
begin
            y :=  X / 2.0;
            z :=    2.0 / X;
            z_to_the[0]:=1;
            y_to_the_2_times[0]:=1;
            for i:=1 to 6 do begin
                        y_to_the_2_times[i]:=sqr(y)*y_to_the_2_times[i-1];
                        z_to_the[i]:=z_to_the[i-1]*z;
            end;
            if      X <= 2.0
            then    Mod_Bessel_2_0:=  -ln( y ) * Mod_Bessel_1 ( 0, X )
                        - 0.57721566
                        + 0.42278419 * y_to_the_2_times[1]
                        + 0.23069756 * y_to_the_2_times[2]
                        + 0.03488589 * y_to_the_2_times[3]
                        + 0.00262698 * y_to_the_2_times[4]
```

```
                        + 0.00010750 * y_to_the_2_times[5]
                        + 0.00000740 * y_to_the_2_times[6]
        else    Mod_Bessel_2_0:=  exp( -X ) * sqrt(1.0 / X)
                        * ( 1.25331414 - 0.07832358 * z
                        + 0.02189568 * z_to_the[2]
                        - 0.01062446 * z_to_the[3]
                        + 0.00587872 * z_to_the[4]
                        - 0.00251540 * z_to_the[5]
                        + 0.00053208 * z_to_the[6])
end;    {Mod_Bessel_2_0}

function Mod_Bessel_2_1 (X: real): real;
var             y,z :   real;
                i:integer;
                y_to_the_2_times:array[0..6] of real;
                z_to_the:array [0..8] of real;
begin
                y :=    X / 2.0;
                z :=    2.0 / X;
                z_to_the[0]:=1;
                y_to_the_2_times[0]:=1;
                for i:=1 to 6 do begin
                        y_to_the_2_times[i]:=sqr(y)*y_to_the_2_times[i-1];
                        z_to_the[i]:=z_to_the[i-1]*z;
                end;
                if      X <= 2.0
                then    Mod_Bessel_2_1:=  ( X * ln( y )
                                        * Mod_Bessel_1 ( 1, X ) + 1.0
                        + 0.15443144 * y_to_the_2_times[1]
                        - 0.67278579 * y_to_the_2_times[2]
                        - 0.18156897 * y_to_the_2_times[3]
                        - 0.01919402 * y_to_the_2_times[4]
                        - 0.00110404 * y_to_the_2_times[5]
                        - 0.00004686 * y_to_the_2_times[6]) / X
                else    Mod_Bessel_2_1:=  exp( -X ) * sqrt(1.0 / X) *
                        ( 1.25331414 + 0.23498619 * z
                        - 0.03655620 * z_to_the[2]
                        + 0.01504268 * z_to_the[3]
                        - 0.00780353 * z_to_the[4]
                        - 0.00325614 * z_to_the[5]
                        - 0.00068245 * z_to_the[6])
end;    {Mod_Bessel_2_1}

function Mod_Bessel_2 ( order: integer; X: real ): real;
var             I:   integer;
                Buffer: array[0..max_order] of real;
begin
                if  order = 0  then  Mod_Bessel_2:= Mod_Bessel_2_0 ( X )
                 else if  order = 1  then  Mod_Bessel_2:= Mod_Bessel_2_1 ( X )
                else begin
                        Buffer [ 0 ] := Mod_Bessel_2_0 ( X );
                        Buffer [ 1 ] := Mod_Bessel_2_1 ( X );
                        for I:= 2 to order do
                                Buffer [ I ] := Buffer [ I-2 ] + ( 2.0 * (I-1) / X ) * Buffer [ I-1 ];
                         Mod_Bessel_2:=  Buffer [ order ]
                end;
end;    {Mod_Bessel_2}

function cyl(x:real):real;
var             temp:real;
                i:integer;
begin
                temp:=Mod_Bessel_1(0,x)/Mod_Bessel_2(0,x);
                for i:=1 to max_order do
                        temp:=temp+(Mod_Bessel_1(i,x)/Mod_Bessel_2(i,x))*2*raiseneg(-1,i);
                cyl:=temp;
end;

{-------------------------------------------------------------------}
{-----------functions and procedures to find the melt temp isotherm ----------}
{-------------------------------------------------------------------}
function l_prime(l:real):real;
begin
                l_prime:=l*velocity/(2*alpha);
 end;
function dist_from_z_axis(x,y:real):real;
begin
                dist_from_z_axis:=sqrt(sqr(l_prime(x))+sqr(l_prime(y)));
end;
function dist_from_c(x,y,z,c:real):real;
begin
                dist_from_c:= sqrt(sqr(l_prime(x))+sqr(l_prime(y))+sqr(l_prime(z-c)));
end;
```

```
function temp_due_to_point_source(point_source_strength,x,y,z,c:real):real;
begin
        temp_due_to_point_source := point_source_strength*velocity/(2*alpha*conductivity_solid)
                *exp(l_prime(x)-dist_from_c(x,y,z,c))/(4*pi*dist_from_c(x,y,z,c));
end;
function temp_due_to_line_source(line_source_strength,x,y:real):real;
begin
        temp_due_to_line_source:=line_source_strength/(2*pi*conductivity_solid)
                *Mod_Bessel_2_0(dist_from_z_axis(x,y))*exp(l_prime(x));

end;
function temperature(x,y,z,c,point_source_strength,line_source_strength:real):real;
begin
        temperature:=temp_due_to_line_source(line_source_strength,x,y)+ambient_temp
                +temp_due_to_point_source(point_source_strength,x,y,z,c);
end;

function F_of_y(x,y,z,c,point_source_strength,line_source_strength:real):real;
var     term_1,term_2:real;
begin
        term_1:=temperature(x,y,z,c,point_source_strength,line_source_strength);
        term_2:=-melting_pt;
        F_of_y:=term_1+term_2;
end;
function dF_by_dy(x,y,z,c,point_source_strength,line_source_strength:real):real;
var     delta_y:real;
begin
        delta_y:=0.01/1000;
        dF_by_dy:=(F_of_y(x,y+delta_y,z,c,point_source_strength,line_source_strength)
                -F_of_y(x,y,z,c,point_source_strength,line_source_strength))/delta_y;
end;

procedure calc_weld_profile(z,cyl_pe,r_val:real;k_weld:integer);
var     x,y,c:real;
        delta_x,delta_y:real;
        point_source_strength,line_source_strength:real;
        isotherm_y,old_isotherm_y:real;
        max_y,old_max_y:real;
        dum:integer;
        e_min,e_max:real;
begin
        point_source_strength:=0;
        line_source_strength:=2*pi*conductivity_solid*(boiling_pt-ambient_temp)*cyl_pe+
                (massdens*latent_heat_vap*2*pi*r_val*u_vel[rp_max[k_weld],k_weld]);

        delta_x:=0.00001;
        delta_y:=0.00001;
        x:=0;
        y:=0;
        c:=0;

        weldbead[k_weld].z:=z;
        x:=0.0001;
        max_y:=0.0;
        old_max_y:=0.00001;

        e_min:=0;
        e_max:=0.01;
        for dum:=1 to 20 do begin
                max_y:=0.5*(e_min+e_max);
                if temperature(x,max_y,z,c,point_source_strength,line_source_strength)<melting_pt
                then e_max:=max_y
        else e_min:=max_y;
        end;
        isotherm_y:=0.5*(e_min+e_max);
        x:=x+delta_x;

        repeat
                old_max_y:=isotherm_y;
                old_isotherm_y:=isotherm_y;
                for dum:=1 to 3 do begin
                        max_y:=old_max_y-(F_of_y(x,old_max_y,z,c,point_source_strength,line_source_strength)
                                /dF_by_dy(x,old_max_y,z,c,point_source_strength,line_source_strength));
                        old_max_y:=max_y;
                end;
                isotherm_y:=max_y;
                x:=x+delta_x;
        until (isotherm_y<old_isotherm_y);
        isotherm_y:=old_isotherm_y;
        x:=x-delta_x;
        weldbead[k_weld].x:=x;
        weldbead[k_weld].y:=old_isotherm_y*2;
        y:=old_isotherm_y;
end;
```

```
{-------------------------------------------------------------------}
{----------functions and procedures to find the keyhole dimensions----------}
{-------------------------------------------------------------------}
 function minor_f_of_r(r_val:real;k:integer):real;
var             l:real;
begin
                l:=alpha/velocity;
                minor_f_of_r:=(delta_T*pi*2*conductivity_solid*cyl(r_val/(2*l)))+
                                (massdens*latent_heat_vap*2*pi*r_val*u_vel[rp_max[k],k])
                                        -sqrt(linking_intensity*max_power);
end;

function abs_power_pud(r_val:real;k:integer):real;
begin
                abs_power_pud:=minor_f_of_r(r_val,k);
end;

function power(trans_power{,r_val,delta_z}:real):real;
begin
                power:=trans_power{+abs_power_pud(r_val)*delta_z};
end;

function find_theta(dum_r,last_r,delta_z:real):real;
var             dz_by_dr:real;
begin
                dz_by_dr:=delta_z/(dum_r-last_r);
                find_theta:=abs(arctan(dz_by_dr));
end;

function major_f_of_r(trans_power,dum_r,last_r,delta_z:real;k:integer):real;
var             mid_dum_r:real;
                theta:real;
                term_1,term_2:real;
begin
                term_1:=dum_r-last_r;
                theta:=find_theta(dum_r,last_r,delta_z);
                mid_dum_r:=(dum_r+last_r)/2;
                term_2:=((delta_z*mid_dum_r)/(2*absorptivity(theta)*
                                power(trans_power{,mid_dum_r,delta_z})))
                                *minor_f_of_r(mid_dum_r,k);
                major_f_of_r:=term_1+term_2;
end;

function d_major_f_by_dr(trans_power,guess_r,last_r,delta_z:real;k:integer):real;
var delta_r:real;
begin
                delta_r:=beam_radius/1000;
                d_major_f_by_dr:=(major_f_of_r(trans_power,guess_r+delta_r,last_r,delta_z,k)
                                -major_f_of_r(trans_power,guess_r,last_r,delta_z,k))/delta_r;
end;
procedure input_data;
var             dummy_vel,dummy_rad:real;
begin
                write('Please enter te maximum power(W)  ');
                readln(max_power);
                write('Please enter the velocity(mm/s) ');
                readln(dummy_vel);
                velocity:=dummy_vel/1000;
                writeln('vel=',velocity:8:4);
                 write('Please enter the beam radius(mm)  ');
                 readln(dummy_rad);
                 beam_radius:=dummy_rad/1000;
end;

procedure newt;
var
                 n,k,i:integer;
                 guess_r,new_guess_r:real;
                 last_r:real;
                 l:real;
                 cyl_pe:real;
                 k_weld:integer;
                 trans_power:real;
                 delta_z:real;
                 mid_r:real;
begin
                 input_data;
                 writeln('hi');
                 trans_power:=max_power;
                 delta_z:=0.002/20;
                 k_weld:=0;
                 k:=0;
                 keyhole[k].z_c:=0;
                 keyhole[k].r_c:=beam_radius;
```

```
                l:=alpha/velocity;
                cyl_pe:=cyl(keyhole[k].r_c/(2*l));
                calc_weld_profile(keyhole[k].z_c,cyl_pe,keyhole[k].r_c,k_weld);

                k_weld:=1;
                kc=1;
                last_r:=keyhole[0].r_c;
                guess_r:=keyhole[0].r_c-(delta_z/tan((8/9)*(pi/2)));
                keyhole[k].z_c:=delta_z;
                for n:=1 to 5 do begin
                        new_guess_r:=guess_r-(major_f_of_r(trans_power,guess_r,last_r,delta_z,k+za_max)/
                                                d_major_f_by_dr(trans_power,guess_r,last_r,delta_z,k+za_max));
                        guess_r:=new_guess_r;
                end;
                keyhole[k].r_c:=guess_r;
                mid_r:=(keyhole[k].r_c+keyhole[k-1].r_c)/2;
                trans_power:=trans_power-(abs_power_pud(mid_r,k+za_max)*delta_z);

                l:=alpha/velocity;
                cyl_pe:=cyl(keyhole[k].r_c/(2*l));
                calc_weld_profile(keyhole[k].z_c,cyl_pe,keyhole[k].r_c,k_weld);

                k_weld:=2;
                kc=2;
                writeln('hi2');

                repeat
                        keyhole[k].z_c:=k*delta_z;
                        last_r:=keyhole[k-1].r_c;
                        guess_r:=(keyhole[k-1].r_c-keyhole[k-2].r_c)+keyhole[k-1].r_c;
                        for n:=1 to 5 do begin
                                new_guess_r:=guess_r-(major_f_of_r(trans_power,guess_r,keyhole[k-1].r_c,delta_z,k+za_max)
                                                /d_major_f_by_dr(trans_power,guess_r,keyhole[k-1].r_c,delta_z,k+za_max));
                                guess_r:=new_guess_r;
                        end;
                        keyhole[k].r_c:=guess_r;
                        mid_r:=(keyhole[k].r_c+keyhole[k-1].r_c)/2;
                        trans_power:=trans_power-(abs_power_pud(mid_r,k+za_max)*delta_z);

                        l:=alpha/velocity;
                        cyl_pe:=cyl(keyhole[k].r_c/(2*l));
                        calc_weld_profile(keyhole[k].z_c,cyl_pe,keyhole[k].r_c,k_weld);

                        k_weld:=k_weld+1;
                        kc=k+1;
                until k>20;
                kc=k-1;
                writeln('pd:=',keyhole[k].z_c:8:6,'   r_max:=',keyhole[k].r_c:8:6);
                for i:=0 to 20 do
                writeln('wx=',(weldbead[i].x*1000):8:4,'   wy:=',(weldbead[i].y*1000):8:4
                        ,'   wz:=',(weldbead[i].z*1000):8:4);
                for i:=0 to 20 do
                        writeln('kr=',(keyhole[i].r_c*1000):8:4,'   kz:=',  (keyhole[i].z_c*1000):8:4);
                readln;
end;
{ ------------------------------------------------------------------------ }
{---- ------------------------- define grid ------------------------------- }
{ ------------------------------------------------------------------------ }
procedure defineboundaries;
var             i,k:integer;
begin

                for k :=0 to za_max-1 do begin
                        ru_max[k]:= rac_u_max;
                        rw_max[k]:= rac_max;
                        rp_max[k]:= rac_max;
                end;
                for k :=za_max to zb_max do begin
                        ru_max[k]:= rb_u_max;
                        rw_max[k]:= rb_max;
                        rp_max[k]:= rb_max;
                end;
                rw_max[zb_max]:=rac_max;
                for k :=zb_max+1 to zc_max do begin
                        ru_max[k]:= rac_u_max;
                        rw_max[k]:= rac_max;
                        rp_max[k]:= rac_max;
                end;

                {zu_max:=z_max;
                zw_max:=z_max-1;
                zp_max:=z_max;}
```

```
        for i:=0 to rb_max-1 do begin
                zp_max[i]:=z_max;
                zu_max[i]:=z_max;
                zw_max[i]:=z_max-1;
                zp_min_td[i]:=0;
                zu_min_td[i]:=0;
                zw_min_td[i]:=0;
        end;
        zu_max[rb_max-1]:=za_max;
        zu_min_td[rb_max-1]:=zb_max;

        for i:=rb_max to rac_max do begin
                zp_max[i]:=za_max;
                zu_max[i]:=za_max;
                zw_max[i]:=za_max-1;
                zp_min_td[i]:=zb_max;
                zu_min_td[i]:=zb_max;
                zw_min_td[i]:=zb_max;
        end;

end {defineboundaries};

procedure define_new_boundaries;
var     i,k:integer;
        flag_up:array[0..r_max] of boolean;
begin
        for i:=0 to r_max do flag_up[i]:=false;
        kc:=za_max;
        for kc:=za_max to zb_max do begin
                i:=0;
                repeat
                        i:=i+1;
                until pw_r_coord[i]>keyhole[k-za_max].r_c;
                i:=i-1;
                rp_max[k]:=i;
                rw_max[k]:=i;
                ru_max[k]:=i-1;
                pressure[rp_max[k],k]:=p_ambt+(surface_ten/pw_r_coord[rp_max[k]]);

                if rp_max[k]<>rp_max[k-1] then begin
                        zp_max[i]:=k;
                        zu_max[i-1]:=k;
                        zw_max[i]:=k-1;
                        flag_up[i]:=true;
                end;
                for i:=rp_max[k]+1 to r_max do begin
                        w_vel[i,k]:=0;
                        u_vel[i,k]:=0;
                        pressure[i,k]:=p_ambt;
                        p_cor[i,k]:=0;
                end;
                pressure[rp_max[k],k]:=p_ambt;
        end;


        k:=zb_max;
        if (rp_max[k]<>1) or (rp_max[k]<>0) then begin
                zp_max[0]:=z_max;
                zw_max[0]:=z_max-1;
                zu_max[0]:=z_max;
                zp_min_td[0]:=0;
                zw_min_td[0]:=0;
                zu_min_td[0]:=0;
                flag_up[0]:=true;
                zp_max[1]:=z_max;
                zw_max[1]:=z_max-1;
                zp_min_td[1]:=0;
                zw_min_td[1]:=0;
                flag_up[1]:=true;

                for i:=2 to rp_max[zb_max]-1 do begin
                        zp_max[i]:=z_max;
                        zu_max[i-1]:=z_max;
                        zw_max[i]:=z_max-1;

                        zp_min_td[i]:=0;
                        zu_min_td[i-2]:=0;
                        zw_min_td[i]:=0;
                        flag_up[i]:=true;
                end;
                zu_min_td[rp_max[zb_max]-1]:=zb_max+1;
                for i:=rp_max[zb_max] to rac_max do begin
                        zp_min_td[i]:=zb_max;
                        zu_min_tD[i]:=zb_max;
```

```
                                        zw_min_td[i]:=zb_max;
                                end;
                                for i:=rp_max[za_max] to rac_max do begin
                                        zp_max[i]:=za_max;
                                        zu_max[i-1]:=za_max;
                                        zw_max[i]:=za_max-1;
                                end;

                end
                else if rp_max[k]=1 then begin
                                zp_max[0]:=z_max;
                                zw_max[0]:=z_max-1;
                                zp_min_td[0]:=0;
                                zw_min_td[0]:=0;
                                flag_up[0]:=true;
                                for i:=rp_max[zb_max] to r_max do begin
                                        zp_min_td[i]:=zb_max;
                                        zu_min_td[i-1]:=zb_max;
                                        zw_min_td[i]:=zb_max ;
                                end;
                end
                else if rp_max[k]=0 then begin
                                for i:=rp_max[k] to r_max do begin
                                        zp_min_td[i]:=zb_max;
                                        zu_min_td[i]:=zb_max;
                                        zw_min_td[i]:=zb_max;
                                end;
                end;
                for i:=0 to r_max do
                                if flag_up[i]=false then begin
                                        zp_max[i]:=zp_max[i+1];
                                        zu_max[i-1]:=zw_max[i];
                                         zw_max[i]:=zw_max[i+1];
                                end;
end;

procedure initgrid_r;
var             i:r_point_int;
                delta_r1,delta_r2:real;
begin
                delta_r1:=rb_max_length/rb_max;
                delta_r2:=(rac_max_length-rb_max_length)/(rac_max-rb_max);
                writeln('dr1:=',delta_r1:8:6,'  dr2:=',delta_r2:8:6);

                for i:=0 to rb_max do
                                pw_r_coord[i]:=i*delta_r1;
                for i:=rb_max to rac_max do
                                pw_r_coord[i]:=rb_max_length+((i-rb_max)*delta_r2);
                for i:=0 to r_max-1 do
                                pw_delta_r_n[i]:=pw_r_coord[i+1]-pw_r_coord[i];
                                        for i:=1 to r_max do
                                pw_delta_r_s[i]:=pw_r_coord[i]-pw_r_coord[i-1];
                pw_r_face_coord[0]:=0.0;
                for i:=1 to r_max do
                                pw_r_face_coord[i]:=(pw_r_coord[i]+pw_r_coord[i-1]) / 2.0;

                pw_delta_r_cv[0]:=pw_r_face_coord[1];
                pw_delta_r_cv[r_max]:=pw_r_coord[r_max]-pw_r_face_coord[r_max];

                for i :=1 to r_max-1 do
                                pw_delta_r_cv[i]:=pw_r_face_coord[i+1]-pw_r_face_coord[i];

                for i:=1  to r_max do
                                u_r_coord[i-1]:=pw_r_face_coord[i];

                u_r_coord[r_max]:=u_r_coord[r_max-1]+pw_delta_r_s[r_max];
                for i:=0 to r_max-1 do
                                u_delta_r_n[i]:=u_r_coord[i+1]-u_r_coord[i];

                for i:=1 to r_max do
                                u_delta_r_s[i]:=u_r_coord[i]-u_r_coord[i-1];

                u_r_face_coord[0]:=0.0;

                for i:=1 to r_max do
                                u_r_face_coord[i]:=(u_r_coord[i]+u_r_coord[i-1]) / 2.0;
                u_delta_r_cv[0]:=2.0*u_r_face_coord[1];
                u_delta_r_cv[r_max]:=u_r_coord[r_max]-u_r_face_coord[r_max];

                for i :=1 to r_max-1 do
                                u_delta_r_cv[i]:=u_r_face_coord[i]-u_r_face_coord[i-1];

end {initgrid_r};
```

```
procedure initgrid_z;
var             k:z_point_int;
                delta_z1,delta_z2,delta_z3:real;
begin

                delta_z1:=za_max_length/za_max;
                delta_z2:=(zb_max_length-za_max_length)/(zb_max-za_max);
                delta_z3:=(zc_max_length-zb_max_length)/(zc_max-zb_max);
                writeln('dz1:=',delta_z1:8:7,'  dz2:=',delta_z2:8:6,'  dz3:=',delta_z3:8:6);

                for k:=0 to za_max do
                                pu_z_coord[k]:=k*delta_z1;
                for k:=za_max to zb_max do
                                pu_z_coord[k]:=za_max_length+((k-za_max)*delta_z2);
                for k:=zb_max to z_max do
                                pu_z_coord[k]:=zb_max_length+((k-zb_max)*delta_z3);

                for k:=0 to z_max-1 do
                                pu_delta_z_u[k]:=pu_z_coord[k+1]-pu_z_coord[k];
                for k:=1 to z_max do
                                pu_delta_z_d[k]:=pu_z_coord[k]-pu_z_coord[k-1];

                pu_z_face_coord[0]:=0.0;
                for k:=1 to z_max do
                                pu_z_face_coord[k]:=(pu_z_coord[k]+pu_z_coord[k-1]) / 2.0;
                pu_delta_z_cv[0]:=pu_z_face_coord[1]{depth_max/z_max};
                pu_delta_z_cv[z_max]:={depth_max/z_max}pu_z_coord[z_max]-pu_z_face_coord[z_max];

                for k :=1 to z_max-1 do
                                pu_delta_z_cv[k]:=pu_z_face_coord[k+1]-pu_z_face_coord[k];

                for k:=1  to z_max do
                                w_z_coord[k-1]:=pu_z_face_coord[k];
                w_z_coord[z_max]:=w_z_coord[z_max-1]+pu_delta_z_d[z_max];

                for k:=0 to z_max-1 do
                                w_delta_z_u[k]:=w_z_coord[k+1]-w_z_coord[k];

                for k:=1 to z_max do
                                w_delta_z_d[k]:=w_z_coord[k]-w_z_coord[k-1];

                w_z_face_coord[0]:=0.0;
                for k:=1 to z_max do
                                w_z_face_coord[k]:=(w_z_coord[k]+w_z_coord[k-1]) / 2.0;
                w_delta_z_cv[z_max]:=w_z_coord[z_max]-w_z_face_coord[z_max];

                for k :=0 to z_max-1 do
                                w_delta_z_cv[k]:=w_z_face_coord[k+1]-w_z_face_coord[k];
end {initgrid_z};


{ ------------------------------------------------------------ }
{ ----------------- i n i t i a l i s a t i o n  r o u t i n e s ----------------- }
{ ------------------------------------------------------------ }
procedure initgrid_u;
var             i:r_point_int;
                k:Z_point_int;
begin
                for i:=0 to r_max do
                                for k:=0 to z_max do
                                                u_vel[i,k]:=0.0;

end {initgrid_u};

procedure initgrid_w;
var             i:r_point_int;
                k:z_point_int;
begin
                for i:=0 to r_max do
                                for k:=0 to z_max do
                                                w_vel[i,k]:=0.0;
end {initgrid_w};

procedure initgrid_p;
var             i:r_point_int;
                k:z_point_int;
begin
                for k:=0 to z_max do
                                for i:=0 to r_max do
                                                pressure[i,k]:=p_ambt;
end {initgrid_p};
```

```
procedure initgrid_q;
var             i:r_point_int;
                k:z_point_int;
begin
                for i:=1 to r_max-1 do begin
                                p_cor[i,0]:=0.0;
                                p_cor[i,z_max]:=0.0;
                                for k:=1  to z_max-1 do
                                                p_cor[i,k]:=0.0;
                end;
                for k:=0  to z_max do begin
                                p_cor[0   ,k]:=0.0;
                                p_cor[r_max,k]:=0.0;
                end;
end {initgrid_q};
{-------------------------------------------------------------}
{-----------------define conductance arrays-------------------}
{-------------------------------------------------------------}
procedure init_conductance_u;
var i,k:integer;
begin

                for k:=1 to z_max-1 do begin
                                for i:=1 to ru_max[0]-1 do begin
                                                uc_n[i,k]:=viscoscity*u_r_face_coord[i+1]*pu_delta_z_cv[k]/ u_delta_r_n[i];
                                                uc_s[i,k]:=viscoscity*u_r_face_coord[i]*pu_delta_z_cv[k] / u_delta_r_s[i];
                                                uc_u[i,k]:=viscoscity*u_r_coord[i]* u_delta_r_cv[i] / pu_delta_z_u[k];
                                                uc_d[i,k]:=viscoscity*u_r_coord[i]*u_delta_r_cv[i] / pu_delta_z_d[k];
                                end;
                end;

                i:=ru_max[0];
                for k:=1 to z_max-1 do begin
                                uc_n[i,k]:=0;
                                uc_s[i,k]:=viscoscity*u_r_face_coord[i]*pu_delta_z_cv[k]/ u_delta_r_s[i];
                                 uc_u[i,k]:=viscoscity*u_r_coord[i]* u_delta_r_cv[i]  / pu_delta_z_u[k];
                                 uc_d[i,k]:=viscoscity*u_r_coord[i]*u_delta_r_cv[i]  /  pu_delta_z_d[k];
                end;
                i:=0;
                for k:=1 to z_max-1 do begin
                                uc_n[i,k]:=viscoscity*u_r_face_coord[i+1]*pu_delta_z_cv[k] / u_delta_r_n[i];
                                uc_s[i,k]:=0;
                                uc_u[i,k]:=viscoscity*u_r_coord[i]* u_delta_r_cv[i] / pu_delta_z_u[k];
                                uc_d[i,k]:=viscoscity*u_r_coord[i]* u_delta_r_cv[i] / pu_delta_z_d[k];
                end;

                k:=0;
                for i:=1 to ru_max[0]-1 do begin
                                uc_n[i,k]:=viscoscity*u_r_face_coord[i+1]* pu_delta_z_cv[k] / u_delta_r_n[i];
                                 uc_s[i,k]:=viscoscity*u_r_face_coord[i]*pu_delta_z_cv[k] / u_delta_r_s[i];
                                uc_u[i,k]:=viscoscity*u_r_coord[i]* u_delta_r_cv[i] / pu_delta_z_u[k];
                                uc_d[i,k]:=0;
                end;

                k:=zu_max[0];
                for i:=1 to ru_max[0]-1 do begin
                                uc_n[i,k]:=viscoscity*u_r_face_coord[i+1]*pu_delta_z_cv[k] / u_delta_r_n[i];
                                 uc_s[i,k]:=viscoscity*u_r_face_coord[i]*pu_delta_z_cv[k]/ u_delta_r_s[i];
                                uc_u[i,k]:=0;
                                uc_d[i,k]:=viscoscity*u_r_coord[i]*u_delta_r_cv[i] / pu_delta_z_d[k];
                end;

                i:=0;
                k:=0;
                uc_n[i,k]:=viscoscity*u_r_face_coord[i+1]*pu_delta_z_cv[k] / u_delta_r_n[i];
                uc_s[i,k]:=0;
                uc_u[i,k]:=viscoscity*u_r_coord[i]* u_delta_r_cv[i] / pu_delta_z_u[k];
                uc_d[i,k]:=0;

                i:=0;
                k:=zu_max[0];
                uc_n[i,k]:=viscoscity*u_r_face_coord[i+1]*pu_delta_z_cv[k] / u_delta_r_n[i];
                uc_s[i,k]:=0;
                uc_u[i,k]:=0;
                uc_d[i,k]:=viscoscity*u_r_coord[i]*u_delta_r_cv[i] / pu_delta_z_d[k];

                i:=ru_max[0];
                k:=0;
                uc_n[i,k]:=0;
                uc_s[i,k]:=viscoscity*u_r_face_coord[i]*pu_delta_z_cv[k] / u_delta_r_s[i];
                uc_u[i,k]:=viscoscity*u_r_coord[i]* u_delta_r_cv[i] / pu_delta_z_u[k];
                uc_d[i,k]:=0;

                i:=ru_max[0];
```

```
                k:=zu_max[0];
                uc_n[i,k]:=0;
                uc_s[i,k]:=viscoscity*u_r_face_coord[i]*pu_delta_z_cv[k]/ u_delta_r_s[i];
                uc_u[i,k]:=0;
                uc_d[i,k]:=viscoscity*u_r_coord[i]*u_delta_r_cv[i] / pu_delta_z_d[k];

end;

procedure init_conductance_w;
var             i,k:integer;
begin
                for k:=1 to zw_max[0]-1 do begin
                                for i:=1 to rw_max[0]-1 do begin
                                                wc_n[i,k]:=viscoscity*pw_r_face_coord[i+1]*w_delta_z_cv[k] / pw_delta_r_n[i];
                                                wc_s[i,k]:=viscoscity*pw_r_face_coord[i]*w_delta_z_cv[k] / pw_delta_r_s[i];
                                                wc_u[i,k]:=viscoscity*pw_r_coord[i]*pw_delta_r_cv[i] / w_delta_z_u[k];
                                                wc_d[i,k]:=viscoscity*pw_r_coord[i]*pw_delta_r_cv[i] / w_delta_z_d[k];
                                end;
                end;

                i:=0;
                for k:=1 to zw_max[0]-1 do begin
                                wc_n[i,k]:=viscoscity*pw_r_face_coord[i+1]*w_delta_z_cv[k] / pw_delta_r_n[i];
                                wc_s[i,k]:=0;
                                wc_u[i,k]:=viscoscity*pw_r_coord[i]*pw_delta_r_cv[i] / w_delta_z_u[k];
                                wc_d[i,k]:=viscoscity*pw_r_coord[i]*pw_delta_r_cv[i] / w_delta_z_d[k];
                end;

                i:=rw_max[0];
                for k:=1 to zw_max[0]-1 do begin
                                wc_n[i,k]:=0;
                                wc_s[i,k]:=viscoscity*pw_r_face_coord[i]*w_delta_z_cv[k] / pw_delta_r_s[i];
                                wc_u[i,k]:=viscoscity*pw_r_coord[i]*pw_delta_r_cv[i] / w_delta_z_u[k];
                                wc_d[i,k]:=viscoscity*pw_r_coord[i]*pw_delta_r_cv[i] / w_delta_z_d[k];
                end;

                k:=0;
                for i:=1 to rw_max[0]-1 do begin
                                wc_n[i,k]:=viscoscity*pw_r_face_coord[i+1]*w_delta_z_cv[k] / pw_delta_r_n[i];
                                wc_s[i,k]:=viscoscity*pw_r_face_coord[i]*w_delta_z_cv[k] / pw_delta_r_s[i];
                                wc_u[i,k]:=viscoscity*pw_r_coord[i]*pw_delta_r_cv[i] / w_delta_z_u[k];
                                wc_d[i,k]:=0;
                end;

                k:=zw_max[0];
                for i:=1 to rw_max[0]-1 do begin
                                wc_n[i,k]:=viscoscity*pw_r_face_coord[i+1]*w_delta_z_cv[k] / pw_delta_r_n[i];
                                wc_s[i,k]:=viscoscity*pw_r_face_coord[i]*w_delta_z_cv[k] / pw_delta_r_s[i];
                                wc_u[i,k]:=0;
                                wc_d[i,k]:=viscoscity*pw_r_coord[i]*pw_delta_r_cv[i] / w_delta_z_d[k];
                end;

                i:=0;
                k:=0;
                wc_n[i,k]:=viscoscity*pw_r_face_coord[i+1]*w_delta_z_cv[k] / pw_delta_r_n[i];
                wc_s[i,k]:=0;
                wc_u[i,k]:=viscoscity*pw_r_coord[i]*pw_delta_r_cv[i] / w_delta_z_u[k];
                wc_d[i,k]:=0;

                i:=0;
                k:=zw_max[0];
                wc_n[i,k]:=viscoscity*pw_r_face_coord[i+1]*w_delta_z_cv[k] / pw_delta_r_n[i];
                wc_s[i,k]:=0;
                wc_u[i,k]:=0;
                wc_d[i,k]:=viscoscity*pw_r_coord[i]*pw_delta_r_cv[i] / w_delta_z_d[k];

                i:=rw_max[0];
                k:=0;
                wc_n[i,k]:=0;
                wc_s[i,k]:=viscoscity*pw_r_face_coord[i]*w_delta_z_cv[k] / pw_delta_r_s[i];
                wc_u[i,k]:=viscoscity*pw_r_coord[i]*pw_delta_r_cv[i] / w_delta_z_u[k];
                wc_d[i,k]:=0;

                i:=rw_max[0];
                k:=zw_max[0];
                wc_n[i,k]:=0;
                wc_s[i,k]:=viscoscity*pw_r_face_coord[i]*w_delta_z_cv[k] / pw_delta_r_s[i];
                wc_u[i,k]:=0;
                wc_d[i,k]:=viscoscity*pw_r_coord[i]*pw_delta_r_cv[i] / w_delta_z_d[k];

end;
```

```
procedure init_temp;
var i,k:integer;
begin
        for i:=0 to r_max do
                for k:=0 to z_max do
                        temp[i,k]:=300;
end;
{ ------------------------------------------------------------------------ }
{ ------------------------ define coefficients ------------------------- }
{ ------------------------------------------------------------------------ }
function au_n( i: r_point_int; k: z_point_int ): real;
var     flowrate_n:real;
        conductance_n:real;
        afunc_n:real;
        pectno, x:real;
begin
        if i = ru_max[k] then au_n:=0.0 else begin
                flowrate_n    := massdens*u_r_face_coord[i+1]*mid_u_n(i,k)* pu_delta_z_cv[k];
                conductance_n := uc_n[i,k];
                pectno := flowrate_n / conductance_n;
                x:=1.0-0.1*abs( pectno );
                afunc_n:= min0(x*x*x*x*x );
                au_n:=conductance_n*afunc_n+min0 ( -flowrate_n );
        end;
end {au_n};

function au_s( i: r_point_int; k: z_point_int ): real;
var     flowrate_s:     real;
        conductance_s:real;
        afunc_s:        real;
        pectno, x:      real;
begin
        if i <= 0 then au_s:=0.0 else begin
                flowrate_s:= massdens*u_r_face_coord[i]*mid_u_s(i,k)* pu_delta_z_cv[k];
                conductance_s := uc_s[i,k];
                pectno := flowrate_s / conductance_s;
                x:=1.0-0.1*abs( pectno );
                afunc_s:= min0(x*x*x*x*x );
                au_s:=conductance_s*afunc_s+min0 ( flowrate_s );
        end;
end {au_s};

function au_u( i: r_point_int; k: z_point_int ): real;
var     flowrate_u:real;
        conductance_u:real;
        afunc_u:real;
        pectno, x:Real;

begin
        if k = zu_max[i] then au_u:=0.0
        else begin
                flowrate_u := 0.5*massdens*u_r_coord[i]*(w_vel[i,k]+w_vel[i+1,k])*u_delta_r_cv[i] ;
                conductance_u := uc_u[i,k];
                pectno := flowrate_u / conductance_u;
                x:=1.0-0.1*abs( pectno );
                afunc_u:= min0(x*x*x*x*x );
                au_u:=conductance_u*afunc_u+min0 ( -flowrate_u );
        end;
end {au_u};

function au_d( i: r_point_int; k: z_point_int ): real;
var     flowrate_d:real;
        conductance_d:real;
        afunc_d:real;
        pectno, x:real;
begin
        if k = 0 then au_d:=0.0
        else begin
                flowrate_d    := 0.5*massdens*u_r_coord[i]*(w_vel[i,k-1]+w_vel[i+1,k-1])*u_delta_r_cv[i];
                conductance_d := uc_d[i,k];
                pectno := flowrate_d / conductance_d;
                x:=1.0-0.1*abs( pectno );
                afunc_d:= min0(x*x*x*x*x );
                au_d:=conductance_d*afunc_d+min0 ( flowrate_d );
        end;
end {au_d};

function au_h( i: r_point_int; k: z_point_int ): real;
begin
        au_h:=  au_n (i,k)+au_s (i,k)+au_u (i,k)+au_d (i,k)
            +(viscoscity*u_delta_r_cv[i]*pu_delta_z_cv[k]  / u_r_coord[i]);
end {au_h};
```

```
function bu_m( i: r_point_int; k: z_point_int ): real;
var             sourceterm_u:real;
begin
                if i = rp_max[k]
                then sourceterm_u:=0.0
                else sourceterm_u:=- u_r_coord[i]*( (pressure[i+1,k]-pressure[i,k]) / pw_delta_r_n[i] );
                bu_m:=  sourceterm_u*u_delta_r_cv[i]*pu_delta_z_cv[k];
end {bu_m};

function bu_r( i: r_point_int; k: z_point_int ): real;
{ source term+axial conduction }
begin
                bu_r:=  au_u (i,k)*u_vel[i,k+1]+au_d (i,k)*u_vel[i,k-1]+bu_m( i,k);
end {bu_r};

function bu_z( i: r_point_int; k: z_point_int ): real;
{ source term+axial conduction }
begin
                bu_z:=  au_n (i,k)*u_vel[i+1,k]+au_s (i,k)*u_vel[i-1,k]+bu_m( i,k);
end {bu_z};

{ 2. the axial momentum equation }
function aw_n( i: r_point_int; k: z_point_int ): real;
var             flowrate_n:real;
                conductance_n:real;
                afunc_n:real;
                pectno, x:real;
begin
                if i = rw_max[k] then aw_n:=0.0
                else begin
                                flowrate_n := 0.5*massdens*pw_r_face_coord[i+1]*(u_vel[i,k]+u_vel[i,k+1])
                                                                * w_delta_z_cv[k];
                                conductance_n := wc_n[i,k];
                                pectno := flowrate_n / conductance_n;
                                x:=1.0-0.1*abs( pectno );
                                afunc_n:= min0(x*x*x*x*x );
                                aw_n:=conductance_n*afunc_n+min0 ( -flowrate_n );
                end;
end {aw_n};

function aw_s( i: r_point_int; k: z_point_int ): real;
var             flowrate_s:real;
                conductance_s:real;
                afunc_s:real;
                pectno, x:real;
begin
                flowrate_s:= 0.5*massdens*pw_r_face_coord[i]*(u_vel[i-1,k]+u_vel[i-1,k+1])*w_delta_z_cv[k] ;
                conductance_s:= wc_s[i,k];
                pectno := flowrate_s / conductance_s;
                x:=1.0-0.1*abs( pectno );
                afunc_s:= min0(x*x*x*x*x );
                aw_s:=conductance_s*afunc_s+min0 ( flowrate_s );
end {aw_s};

function aw_u( i: r_point_int; k: z_point_int ): real;
var             flowrate_u:real;
                conductance_u:real;
                afunc_u:real;
                pectno, x:real;
begin
                if i = 0        then aw_u:= 0.0
                else if k = zw_max[i] then aw_u:= 0.0
                else if i>= rw_max[k+1] then aw_u:= 0.0
                else begin
                                flowrate_u := massdens*pw_r_coord[i] *mid_w_u(i,k)        * pw_delta_r_cv[i];
                                conductance_u:= wc_u[i,k];
                                pectno := flowrate_u / conductance_u;
                                x:=1.0-0.1*abs( pectno );
                                afunc_u:= min0(x*x*x*x*x );
                                aw_u:=conductance_u*afunc_u+min0 ( -flowrate_u );
                end;
end {aw_u};

function aw_d( i: r_point_int; k: z_point_int ): real;
var             flowrate_d:real;
                conductance_d:real;
                afunc_d:real;
                pectno, x:real;
begin
                if i = 0 then aw_d:=0
                else if k = 0 then aw_d:=0.0
                else if i>= rp_max[k-1] then aw_d:=0.0
                else begin
                                flowrate_d := massdens*pw_r_coord[i]*mid_w_d(i,k)* pw_delta_r_cv[i];
```

```
                        conductance_d:= wc_d[i,k];
                        pectno := flowrate_d / conductance_d;
                        x:=1.0-0.1*abs( pectno );
                        afunc_d:= min0(x*x*x*x*x );
                        aw_d:=conductance_d*afunc_d+min0 ( flowrate_d );
                end;
end {aw_d};

function aw_h( i: r_point_int; k: z_point_int ): real;
begin
                aw_h:=  aw_n (i,k)+aw_s (i,k)
                    +aw_u (i,k)+aw_d (i,k)
end {aw_h};

function bw_m( i: r_point_int; k: z_point_int ): real;
var             sourceterm_w:real;
begin
                sourceterm_w:= pw_r_coord[i]*( (pressure[i,k+1]-pressure[i,k]) / pu_delta_z_u[k] );
                bw_m:=  sourceterm_w*pw_delta_r_cv[i]*w_delta_z_cv[k]
end {bw_m};

function bw_r( i: r_point_int; k: z_point_int ): real;
{ source term+axial conduction }
begin
                bw_r:=  aw_u (i,k)*w_vel[i,k+1]+aw_d (i,k)*w_vel[i,k-1]+bw_m (i,k);
end {bw_r};

function bw_z( i: r_point_int; k: z_point_int ): real;
{ source term+axial conduction }
begin
                bw_z:=  aw_n (i,k)*w_vel[i+1,k]+aw_s (i,k)*w_vel[i-1,k]+bw_m (i,k);
end {bw_z};

{ 3. the pressure field equation }
function ap_n( r_pn: r_point_int; z_pn: z_point_int ): real;
var             d_n:real;
begin
                if r_pn = rp_max[z_pn]  then ap_n:=0.0
                else begin
                        d_n:= u_r_coord[r_pn]*pu_delta_z_cv[z_pn] / au_h( r_pn, z_pn );
                        ap_n:=massdens*d_n* u_r_coord[r_pn]*pu_delta_z_cv[z_pn] ;
                end;
end {ap_n};

function ap_s( r_pn: r_point_int; z_pn: z_point_int ): real;
var             d_s:real;
begin
                d_s:=u_r_coord[r_pn-1]*pu_delta_z_cv[z_pn]  / au_h( r_pn-1, z_pn );
                ap_s:=massdens*d_s*u_r_coord[r_pn-1]*pu_delta_z_cv[z_pn];
end {ap_s};

function ap_u( r_pn: r_point_int; z_pn: z_point_int ): real;
var             d_u:real;
begin
                if z_pn >= z_max-1 then ap_u:=0.0
                else begin
                        d_u:= pw_r_coord[r_pn]*pw_delta_r_cv[r_pn] / aw_h( r_pn, z_pn );
                        ap_u:=massdens*d_u*pw_r_coord[r_pn]*pw_delta_r_cv[r_pn];
                end;
end {ap_u};

function ap_d( r_pn: r_point_int; z_pn: z_point_int ): real;
var             d_d:            real;
begin
                if z_pn <= 1 then ap_d:=0.0
                else begin

                        d_d:= pw_r_coord[r_pn]*pw_delta_r_cv[r_pn]  / aw_h( r_pn, z_pn-1 );
                        ap_d:=  massdens*d_d* pw_r_coord[r_pn]*pw_delta_r_cv[r_pn];

                end;
end {ap_d};

function ap_h( r_pn: r_point_int; z_pn: z_point_int ): real;
begin
                ap_h:= ap_n(r_pn,z_pn)+ap_s(r_pn,z_pn)+ap_u (r_pn,z_pn)+ap_d(r_pn,z_pn);
end {ap_h};

{ 4. the pressure correction equation }
function bq_m( r_pn: r_point_int; z_pn: z_point_int ): real;
begin
                if  r_pn = 0 then
                        bq_m:= ( massdens*pw_r_coord[r_pn]*w_vel[r_pn,z_pn-1]
                                        -massdens*pw_r_coord[r_pn]*w_vel[r_pn,z_pn] )
```

```
                                        *pw_delta_r_cv[r_pn]
                else
                        bq_m:= ( massdens*u_r_coord[r_pn-1]*u_vel[r_pn-1,z_pn]-
                                massdens*u_r_coord[r_pn]*u_vel[r_pn ,z_pn] )*pu_delta_z_cv[z_pn]
                        +( massdens*pw_r_coord[r_pn]*w_vel[r_pn,z_pn-1]
                                - massdens*pw_r_coord[r_pn]*w_vel[r_pn,z_pn])*pw_delta_r_cv[r_pn];
end {bq_m};

function bq_r( r_pn: r_point_int; z_pn: z_point_int ): real;
{ source term+axial conduction }
begin
                bq_r:=  ap_u (r_pn,z_pn)*p_cor[r_pn,z_pn+1]
                        +ap_d (r_pn,z_pn)*p_cor[r_pn,z_pn-1]
                        +bq_m (r_pn,z_pn);
end {bq_r};

function bq_z( r_pn: r_point_int; z_pn: z_point_int ): real;
{ source term+axial conduction }
begin
                if  r_pn = 0 then bq_z:=  ap_n (r_pn, z_pn)*p_cor[r_pn+1,z_pn]
                                +bq_m (r_pn, z_pn)
                else if r_pn = rp_max[z_pn] then bq_z:= ap_s (r_pn, z_pn)*p_cor[r_pn-1,z_pn]
                                +bq_m (r_pn, z_pn)
                else bq_z:=   ap_n (r_pn,z_pn)*p_cor[r_pn+1,z_pn]
                                +ap_s (r_pn,z_pn)*p_cor[r_pn-1,z_pn]
                                +bq_m (r_pn,z_pn);
end {bq_z};
{TDMA coefficients}
function find_q(a,c,d,p_last,q_last:real):real;
begin
                find_q:=(d+(c*q_last))/(a-(c*p_last));
end;
function find_p(a,b,c,p_last:real):real;
begin
                find_p:=b/(a-(c*p_last));
end;

{ ------------------------------------------------------------------------ }
{--- ------------------------ d i f f.  e q.  s o l v e r s ---------------------------- }
{ ------------------------------------------------------------------------ }
{ 1. calculate the pressure correction field }
procedure dotdma_qr(k:integer);
var             i:                              r_point_int;
                p, q,line_q:    r_point_grid;
                a,b,c,d:        real;

begin { dotdma }
                p[0]:=0.0;
                q[0]:=0.0;
                for i:=1 to rp_max[k]-1 do begin
                        a:=ap_h(i,k);
                        b:=ap_n(i,k);
                        c:=ap_s(i,k);
                        d:=bq_r(i,k);
                        p[i]:=find_p(a,b,c,p[i-1]);
                        q[i]:=find_q(a,c,d,p[i-1],q[i-1]);
                end;

                for i:=rp_max[k]-1 downto 0 do
                begin
                        line_q[i]:=p_cor[i,k];
                        p_cor[i,k]:=p[i]*p_cor[i+1,k]+q[i];
{                       p_cor[i,k]:=0.5*line_q[i]+0.5*p_cor[i,k];}
                end;
end {dotdma_qr};

procedure dotdma_qrb(k:integer);
var             i: r_point_int;
                line_q:         r_point_grid;
                p, q:           r_point_grid;
                a,b,c,d:real;
begin { dotdma }
                p[rp_max[k]]:=0;
                q[rp_max[k]]:=0;
                for i:=rp_max[k]-1 downto 1 do begin
                        a:=ap_h(i,k);
                        b:=ap_s(i,k);
                        c:=ap_n(i,k);
                        d:=bq_r(i,k);
                        p[i]:=find_p(a,b,c,p[i-1]);
                        q[i]:=find_q(a,c,d,p[i-1],q[i-1]);
                end;
```

```
                for i:=1 to rp_max[k]-1 do
                begin
                                line_q[i]:=p_cor[i,k];
                                p_cor[i,k]:=p[i]*p_cor[i-1,k]+q[i];
                               p_cor[i,k]:=0.5*line_q[i]+0.5*p_cor[i,k] ;
                end;
                p_cor[0,k]:=p_cor[1,k];
end {dotdma_qr};

procedure dotdma_qz(i:integer);
var             k: z_point_int;
                p,q,line_q:     z_point_grid;
                a,b,c,d:real;
begin
                p[1]:=0.0;
                q[1]:=0.0;
                for k:=2 to zp_max[i]-2 do begin
                                :=ap_h(i,k);
                                b:=ap_u(i,k);
                                c:=ap_d(i,k);
                                d:=bq_z(i,k);
                                p[k]:=find_p(a,b,c,p[k-1]);
                                q[k]:=find_q(a,c,d,p[k-1],q[k-1]);
                end;
                for k:=zp_max[i]-2 downto 2 do begin
                                line_q[k]:=p[k]*p_cor[i,k+1]+q[k];
                                p_cor[i,k]:=0.5*line_q[k]+0.5*p_cor[i,k];
                end;
end;

procedure dotdma_qzb(i:integer);
var             k: z_point_int;
                p,q,line_q:     z_point_grid;
                a,b,c,d:        real;
begin
                p[zp_max[i]-1]:=0;
                q[zp_max[i]-1]:=0;
                for k:=zp_max[i]-2 downto zp_min_td[i]+2 do begin
                                a:=ap_h(i,k);
                                b:=ap_d(i,k);
                                c:=ap_u(i,k);
                                d:=bq_z(i,k);
                                p[k]:=find_p(a,b,c,p[k+1]);
                                q[k]:=find_q(a,c,d,p[k+1],q[k+1]);
                end;
                for k:=zp_min_td[i]+2 to zp_max[i]-1 do begin
                                line_q[k]:=p[k]*p_cor[i,k-1]+q[k];
                                p_cor[i,k]:=0.5*line_q[k]+0.5*p_cor[i,k];
                end;
end;

procedure calcp_cor_r;
{ this routine calculates the pressure correction using the tdma for lines of constant z }
var k:integer;
begin
                for k:=2 to zp_max[0]-2 do dotdma_qr(k);
                for k:=zp_max[0]-2 downto 2 do dotdma_qr(k);
end {calcp_cor_r};

procedure calcp_cor_z;
var i:integer;
begin
                for i:=1 to rp_max[1]-1 do dotdma_qz(i);
                for i:=rp_max[1]-1 downto 1 do dotdma_qz(i);
end; {calcp_cor_z}

procedure calcp_cor_rz;
var             n:integer;
begin
                for n:=0 to 3 do begin
                                calcp_cor_r;
                                calcp_cor_z;
                end;
end {calcp_cor_rz};

procedure dotdma_ur(k:integer);
var             i:                              r_point_int;
                line_u:                         r_point_grid;
                p, q:                           r_point_grid;
                a,b,c,d:real;
begin { dotdma }
                p[0]:=0.0;
                q[0]:=0.0;
                for i:=1 to ru_max[k]-1 do begin
```

```
                        a:=au_h(i,k);
                        b:=au_n(i,k);
                        c:=au_s(i,k);
                        d:=bu_r(i,k);
                        p[i]:=find_p(a,b,c,p[i-1]);
                        q[i]:=find_q(a,c,d,p[i-1],q[i-1]);
                end;
                for i:=ru_max[k]-1 downto 1 do begin
                        line_u[i]:=p[i]*u_vel[i+1,k]+q[i];;
                        u_vel[i,k]:=0.5*line_u[i]+0.5*u_vel[i,k];
                end;
end {dotdma_ur};

procedure dotdma_urb(k:integer);
var             i: r_point_int;
                line_u:  r_point_grid;
                p, q:  r_point_grid;
                a,b,c,d:real;
begin { dotdma }
                p[ru_max[k]]:=0.0;
                q[ru_max[k]]:=u_vel[ru_max[k],k];
                for i:=ru_max[k]-1 downto 1 do begin
                        a:=au_h(i,k);
                        b:=au_s(i,k);
                        c:=au_n(i,k);
                        d:=bu_r(i,k);
                        p[i]:=find_p(a,b,c,p[i-1]);
                        q[i]:=find_q(a,c,d,p[i-1],q[i-1]);
                end;
                for i:=1 to ru_max[k]-1  do begin
                         line_u[i]:=p[i]*u_vel[i-1,k]+q[i];
                        u_vel[i,k]:=0.5*line_u[i]+0.5*u_vel[i,k];
                end;
end {dotdma_ur};

procedure dotdma_uz(i:integer);
var              k:  z_point_int;
                  p, q:  z_point_grid;
                line_u: z_point_grid;
                a,b,c,d:real;
begin { dotdma }
                 p[0]:=1.0;
                q[0]:=0.0;
                for k:=1 to zu_max[i]-1 do begin
                        a:=au_h(i,k);
                        b:=au_u(i,k);
                        c:=au_d(i,k);
                        d:=bu_z(i,k);
                        p[k]:=find_p(a,b,c,p[k-1]);
                        q[k]:=find_q(a,c,d,p[k-1],q[k-1]);
                end;
                for k:=zu_max[i]-1 downto 0  do begin
                        line_u[k]:=p[k]*u_vel[i,k+1]+q[k];
                        u_vel[i,k]:=0.5*line_u[i]+0.5*u_vel[i,k];
                end;
                if zu_max[i]=z_max-1 then u_vel[i,zu_max[i]]:=u_vel[i,zu_max[i]-1];
end {dotdma_uz};

procedure dotdma_uzb(i:integer);

var              k:z_point_int;
                p, q: z_point_grid;
                a,b,c,d:real;
                temp_u:real;
begin { dotdma }
                p[zu_max[i]]:=1;
                q[zu_max[i]]:=0;
                for k:=zu_max[i]-1 downto zu_min_td[i]+1 do begin
                        a:=au_h(i,k);
                        b:=au_d(i,k);
                        c:=au_u(i,k);
                        d:=bu_z(i,k);
                        p[k]:=find_p(a,b,c,p[k+1]);
                        q[k]:=find_q(a,c,d,p[k+1],q[k+1]);
                end;
                for k:= zu_min_td[i]+1 to zu_max[i] do begin
                        temp_u:=p[k]*u_vel[i,k-1]+q[k];
                        u_vel[i,k]:=0.5*temp_u+0.5*u_vel[i,k];
                end;
                if zu_min_td[i]=0 then u_vel[i,0]:=u_vel[i,1];
end;

procedure calcufield_r;
{ this routine calculates the radial flow velocities using the tdma for lines of constant z }
```

```
var             k:integer;
begin
                for k:=1 to zu_max[0]-1 do dotdma_ur(k);
                for k:=zu_max[0]-1 downto 1 do dotdma_ur(k);
end {calcufield_r};

procedure calcufield_z;
{   this routine calculates the radial flow velocities using the tdma for lines of constant z }
var             i:integer;
begin
                for i:=1 to ru_max[0]-1 do dotdma_uz(i);
                for i:=ru_max[0]-1 downto 1 do dotdma_uz(i);
end {calcufield_z};

{ 2. calculate the radial velocity field }
procedure calcufield_rz;
var n:integer;
begin
                for n:=0 to 3 do begin
                                calcufield_r;
                                calcufield_z;
                end;
end {calcufield_rz};


{ 3. calculate the axial velocity field }
procedure dotdma_wr(k:integer);
var             i:r_point_int;
                line_w:r_point_grid;
                p, q:r_point_grid;
                a,b,c,d:real;
begin { dotdma }
                p[0]:=1.0;
                q[0]:=0.0;
                for i:=1 to rw_max[k]-1 do begin
                                a:=aw_h(i,k);
                                b:=aw_n(i,k);
                                c:=aw_s(i,k);
                                d:=bw_r(i,k);
                                p[i]:=find_p(a,b,c,p[i-1]);
                                q[i]:=find_q(a,c,d,p[i-1],q[i-1]);
                end;
                for i:=rw_max[k] -1 downto 0 do begin
                                line_w[i]:=p[i]*w_vel[i+1,k]+q[i];
                                w_vel[i,k]:=0.5*line_w[i]+0.5*w_vel[i,k];
                end;
end {dotdma_wr};

procedure dotdma_wrb(k:integer);
var             i:  r_point_int;
                line_w: r_point_grid;
                p, q:  r_point_grid;
                a,b,c,d: real;
begin { dotdma }
                p[rw_max[k]]:=0.0;
                q[rw_max[k]]:=0.0;
                for i:=rw_max[k] -1 downto 1 do begin
                                a:=aw_h(i,k);
                                 b:=aw_s(i,k);
                                 c:=aw_n(i,k);
                                 d:=bw_r(i,k);
                                  p[i]:=find_p(a,b,c,p[i-1]);
                                  q[i]:=find_q(a,c,d,p[i-1],q[i-1]);
                end;
                 for i:=rw_max[k] -1 downto 0 do begin
                                line_w[i]:=p[i]*w_vel[i-1,k]+q[i];
                                w_vel[i,k]:=0.5*line_w[i]+0.5*w_vel[i,k];
                end;
                w_vel[0,k]:=w_vel[1,k];
end {dotdma_wr};

procedure dotdma_wz(i:integer);
var             k:  z_point_int;
                p,q,line_w:z_point_grid;
                a,b,c,d: real;
                max:integer;
begin { dotdma }
                p[0]:=1.0;
                q[0]:=0.0;
                if (i<rb_max) then max:=zw_max[i]-1 else max:=zw_max[i];
                for k:=1 to max do begin
                                a:=aw_h(i,k);
                                b:=aw_u(i,k);
                                c:=aw_d(i,k);
```

```
                    d:=bw_z(i,k);
                    p[k]:=find_p(a,b,c,p[k-1]);
                    q[k]:=find_q(a,c,d,p[k-1],q[k-1]);
            end;
             for k:=max downto 0 do begin
                    line_w[k]:=p[k]*w_vel[i,k+1]+q[k];
                    w_vel[i,k]:=0.5*line_w[k]+0.5*w_vel[i,k];
            end;
            if max=zw_max[i]-1 then w_vel[i,zw_max[i]]:=w_vel[i,zw_max[i]-1];
end {dotdma_wz};

procedure dotdma_wzb(i:integer);
var         k: z_point_int;
            p,q:z_point_grid;
            a,b,c,d: real;
            min:  integer;
            temp_w: real;
begin
            if i<rb_max then min:=zw_min_td[i]+1 else min:=zw_min_td[i];
            p[zw_max[i]]:=1;
            q[zw_max[i]]:=0;
            for k:= zw_max[i]-1 downto min+1 do begin
                    a:=aw_h(i,k);
                    b:=aw_d(i,k);
                    c:=aw_u(i,k);
                    d:=bw_z(i,k);
                    p[k]:=find_p(a,b,c,p[k+1]);
                    q[k]:=find_q(a,c,d,p[k+1],q[k+1]);
            end;
            for k:=min+1 to zw_max[i] do begin
                    temp_w:=p[k]*w_vel[i,k-1]+q[k];
                    w_vel[i,k]:=0.5*temp_w+0.5*w_vel[i,k];
            end;
            if min=zw_min_td[i]+1 then w_vel[i,0]:=w_vel[i,1];
end;

procedure calcvfield_r;
{ this routine calculates the axial flow velocities using the tdma for lines of constant z }
var         i,k:integer;
begin
            for k:=1 to zw_max[0]-1 do dotdma_wr(k);
            for i:=0 to rw_max[1]-1  do w_vel[i,0]:=w_vel[i,1];
            for i:=1 to rw_max[zw_max[0]-1]-1  do w_vel[i,zw_max[0]]:=w_vel[i,zw_max[0]-1];
            for k:=zw_max[0]-1 downto 1 do dotdma_wr(k);
            for i:=0 to rw_max[1]-1 do w_vel[i,0]:=w_vel[i,1];
            for i:=1 to rw_max[zw_max[0]-1]-1 do w_vel[i,zw_max[0]]:=w_vel[i,zw_max[0]-1];

            for k:=1 to zw_max[0]-1 do dotdma_wrb(k);
            for i:=0 to rw_max[1]-1  do w_vel[i,0]:=w_vel[i,1];
            for i:=1 to rw_max[zw_max[0]-1]-1  do w_vel[i,zw_max[0]]:=w_vel[i,zw_max[0]-1];
            for k:=zw_max[0]-1 downto 1 do dotdma_wrb(k);
            for i:=0 to rw_max[1]-1 do w_vel[i,0]:=w_vel[i,1];
            for i:=1 to rw_max[zw_max[0]-1]-1 do w_vel[i,zw_max[0]]:=w_vel[i,zw_max[0]-1];
end {calcvfield_r};
procedure calcvfield_z;
{ this routine calculates the axial flow velocities using the tdma for lines of constant r }
var i:integer;
begin
            for i:=1 to rw_max[0]-1 do dotdma_wz(i);
            for i:=rw_max[0]-1 downto 1 do dotdma_wz(i);
            for i:=1 to rw_max[0]-1 do dotdma_wzb(i);
            for i:=rw_max[0]-1 downto 1 do dotdma_wzb(i);
end {calcvfield_z};

procedure calcvfield_rz;
{var j_it:integer;}
begin
            for j_it:=0 to 5 do begin
                    calcvfield_r;
                    calcvfield_z;
            end;
end {calcvfield_rz};

{ 4a. correct the radial velocities }
procedure correctu_vel;
var         d_n:                    real;
            i:                      r_point_int;
            k:                      z_point_int;

begin
            for k:=1 to zu_max[0]-1 do begin
                    for i:=1 to ru_max[k]-1 do begin
                            d_n:= u_r_coord[i]*pu_delta_z_cv[k] / au_h( i, k );
                            u_vel[i,k]:=u_vel[i,k]+d_n*(p_cor[i,k]-p_cor[i+1,k]);
```

```
                end;
        end;
        for i:=0 to ru_max[0]-1 do begin
                u_vel[i,0]:=u_vel[i,1];
                u_vel[i,zu_max[0]]:=u_vel[i,zu_max[0]-1];
        end;
end {correctu_vel};

{ bb. correct the axial velocities }
procedure correctw_vel;
var     d_u:                    real;
        i:                      r_point_int;
        k:                      z_point_int;
begin
        for k:=1 to zw_max[0]-1 do
        begin
                for i:=1 to rw_max[k]-1 do begin
                        d_u:= pw_r_coord[i]*pw_delta_r_cv[i]/ aw_h( i, k );
                        w_vel[i,k]:=w_vel[i,k]+d_u*(p_cor[i,k]-p_cor[i,k+1]);
                end;
        end;
        for k:=0 to zw_max[0] do
                w_vel[0,k]:=w_vel[1,k];
        for i:=1 to rw_max[k]-1 do begin
                w_vel[i,0]:=w_vel[i,1];
                w_vel[i,zw_max[0]]:=w_vel[i,zw_max[0]-1];
        end;

end {correctw_vel};

{ 5. correct the pressure }
procedure correctpressure;
var     i:                      r_point_int;
        k:                      z_point_int;
begin
        for k:=2 to zp_max[0]-2 do
                for i:=1 to rp_max[k]-1 do
                        pressure[i,k]:=pressure[i,k]+0.01*p_cor[i,k];
        for k:=0 to zp_max[0] do
                pressure[0,k]:=pressure[1,k];

end {correctpressure};

procedure set_u_bound_1;
var k:integer;
begin
      for k:=za_max to zb_max do begin
                if pressure[rp_max[k],k]>pressure[ru_max[k],k] then
                        u_vel[ru_max[k],k]:=0.95*u_vel[ru_max[k],k]-0.05*sqrt((1/density)*(pressure[rp_max[k],k]-pressure[ru_max[k],k]))
                else u_vel[ru_max[k],k]:=0.9*u_vel[ru_max[k],k]+0.1*sqrt(abs((1/density)*(pressure[rp_max[k],k]-pressure[ru_max[k],k])));
                writeln('pe:=',pressure[rp_max[k],k]:10:6,'     pem1:=',pressure[ru_max[k],k]:10:6);
                if pressure[rp_max[k],k]<((surface_ten/pw_r_coord[rp_max[k]])+p_ambt) then
                        pressure[rp_max[k],k]:=pressure[rp_max[k],k]+
                                                1*((surface_ten/pw_r_coord[rp_max[k]]+p_ambt)-pressure[rp_max[k],k]);
        end;
end;

procedure set_u_bound;
var k:integer;
begin

        for k:=za_max to zb_max do begin
                if (pressure[ru_max[k],k]-p_ambt)>(surface_ten/pw_r_coord[rp_max[k]]) then begin
                        u_vel[ru_max[k],k]:=u_vel[ru_max[k],k]*0.9+
                                        0.1*sqrt(abs((1/density)*(pressure[rp_max[k],k]-pressure[rp_max[k]-1,k])));
                end
                else begin
                        u_vel[ru_max[k],k]:=0.9*u_vel[ru_max[k],k]
                                        -0.1*sqrt( (1/density)*(pressure[rp_max[k],k]-pressure[rp_max[k]-1,k]));
                end;
        end;
        k:=za_max+5;
        writeln('u:=',u_vel[ru_max[k],k]:8:4,'  p:=',pressure[ru_max[k],k]:8:4);
        k:=za_max+10;
        writeln('u:=',u_vel[ru_max[k],k]:8:4,'  p:=',pressure[ru_max[k],k]:8:4);
        k:=za_max+15;
        writeln('u:=',u_vel[ru_max[k],k]:8:4,'  p:=',pressure[ru_max[k],k]:8:4);
end;

{-------------------------------------------------------}
{-----------------Masterminds the simple algorithm------------------}
{-------------------------------------------------------}
```

```
procedure simple;
begin
        calcufield_rz;
        calcvfield_rz;
        calcp_cor_rz;
        correctpressure;
        correctu_vel;
        correctw_vel;
end {simple};

{ ------------------------------------------------------------ }
{ ------------------- output to screen -------------------}
{ ------------------------------------------------------------ }
{** additional output to screen procedures can be found in the file: arcauxproc **}
procedure showw_vel;
var i,k:integer;
begin
        writeln( ' axial velocities ' );
        writeln;
        for i:=0 to r_max do begin
                write (i:3,'  ');
                for k:=0 to 15 do
                        write(w_vel[i,k]:4:1,' ');
                writeln;
        end;
        readln;
        for i:=0 to r_max do begin
                write(i:3,'  ');
                for k:=15 to zw_max[0] do
                        write(w_vel[i,k]:4:1,'  ');
                writeln;
        end;
        readln;

end {showw_vel};

procedure showu_vel;
var     i,k:integer;
begin
        writeln( ' radial velocities ' );
        writeln;
        for i:=0 to r_max do begin
        write (i:3);
                for k:=0 to 15 do
                        write(u_vel[i,k]:4:1,'  ');
                writeln;
        end;
        readln;
        for i:=0 to r_max do begin
                for k:=15 to 30 do
                        write(u_vel[i,k]:4:1,'  ');
                writeln;
        end;
        readln;

end {showu_vel};

procedure showp_cor;
var     i,k:integer;
begin
        writeln( ' pressure corrections ' );
        writeln;
        for i:=0 to r_max do begin
                write(i:3);
                k:=2;
                repeat
                        write(p_cor[i,k]:7:0,' ');
                        k:=k+4;
                until k>=z_max;
                writeln;
        end;
        readln;
end {showp_cor};

procedure showpressure;
var     i,k:integer;
begin
        write( ' pressure field ' );
        writeln;
        for i:=0 to r_max do begin
                write(i:3);
                for k:=0 to 15 do
                        write(pressure[i,k]:6:0,' ');
                writeln;
```

```
                end;
                readln;

                for i:=0 to r_max do begin
                        for k:=9 to 19 do
                                write(pressure[i,k]:7:0,' ');
                        writeln;
                end;
                readln;

                for i:=0 to r_max do begin
                 for k:=19 to 29 do
                         write(pressure[i,k]:7:0,'  ');
                writeln;
        end;

end {showpressure};

procedure showbqmfield;
var             i,k:integer;
begin
                writeln( ' bqm field ' );
                writeln;
                for i:=1 to r_max-1 do begin
                        write(i:3);
                        k:=2;
                        repeat
                                write(au_u (i,k):12:6);
                                k:=k+4;
                        until k>=z_max;
                        writeln;
                end;
                readln;
                for k:=0 to z_max do
                begin
                        write(pu_delta_z_u[k]:12:3);
                        writeln;
                end;
                readln;
end {showbqmfield};

procedure showtfield;
var             i,k:integer;
begin
                writeln( ' temp field ' );
                writeln;
                for k:=0 to z_max do
                begin
                        writeln;
                        write({f,}k:3,' ');
                        i:=0;
                        repeat
                                write({f,}temp[i,k]:4:0,'  ');
                                i:=i+20;
                        until i>=z_max div 2;
                        i:=z_max div 2;
                        repeat
                                write({f,}temp[i,k]:6:0,'  ');
                                 i:=i+20;
                        until i>r_max ;
                end;
end {showbqmfield};

procedure file_oput(data_f:string);
var             oput:text;
                i,k:integer;
begin
                {open(oput,data_f,'old');}
                rewrite(oput,data_f);

                writeln(oput);
                writeln(oput,'no vel relax, 10 ns each side, and 2 i bars sections');
                writeln(oput,'za_max:=',za_max:6);
                writeln(oput,'zb_max:=',zb_max:6);
                writeln(oput,'zc_max:=',zc_max:6);
                writeln(oput,'rac_max:=',rac_max:6);
                writeln(oput,'rb_max:=',rb_max:6);

                writeln(oput,'za_length:=',za_length:12:6);
                writeln(oput,'zb_length:=',zb_length:12:6);
                writeln(oput,'zc_length:=',zc_length:12:6);
                writeln(oput,'r_max_length:=',r_max_length:12:6);
                writeln(oput,'r_fine_max_length:=',r_fine_max_length:12:6);
```

```
                writeln(oput);
                writeln(oput,'U VELOCITIES');
                writeln(oput);
                for k:=0 to z_max do begin
                                for i:=0 to rac_u_max do begin
                                                write(oput,u_vel[i,k]:6:1);
                                end;
                        writeln(oput);
                end;

                writeln(oput);
                writeln(oput,'W VELOCITIES');
                writeln(oput);
                for k:=0 to zw_max[0] do begin
                                for i:=0 to r_max do begin
                                                write(oput,w_vel[i,k]:6:1);
                                end;
                        writeln(oput);
                end;

                writeln(oput);
                writeln(oput,'PRESSURES');
                writeln(oput);
                for k:=0 to z_max do begin
                                for i:=0 to r_max do begin
                                                write(oput,pressure[i,k]:6:1,' ');
                                end;
                        writeln(oput);
                end;

                writeln(oput);
                writeln(oput,'PRESSURE CORRECTION');
                writeln(oput);
                for k:=0 to z_max do begin
                                for i:=0 to r_max do begin
                                                write(oput,p_cor[i,k]:6:1);
                                end;
                        writeln(oput);
                end;
                writeln(oput);
                writeln(oput);
                close(oput);
end;

procedure ouput_g(data_f:string);
var             oput:text;
                i,k:integer;
begin
                rewrite(oput,data_f);

                for k:=0 to zw_max[0] do begin
                        if (k/2)=round(k/2) then begin
                                for i:=0 to rb_max+10 do begin
                                        if ((i/2)=round(i/2)) and ( (i<>0) and (k<>0) ) then
                                                writeln(oput,pw_r_coord[i]:8:6,tab,pu_z_coord[k]:8:6,tab,
                                        u_interp(i,k):10:3,tab,w_interp(i,k):10:3);
                                end;
                        end;
                end;
                close(oput);
end;
{-------------------------------------------------------------------}
{------------------------ Main program -----------------------------}
{-------------------------------------------------------------------}
begin
{ 1- initialisation }
                alpha:=conductivity_solid/(density_solid*specific_heat_solid);
                newt;
                defineboundaries;
                initgrid_r;
                initgrid_z;
                initgrid_u;
                initgrid_w;
                initgrid_p;
                initgrid_q;
                init_temp;
                init_conductance_u;
                init_conductance_w;
                define_new_boundaries;
                for k:=za_max to zb_max do
                        writeln('p_max:=',pressure[rp_max[k],k]:8:4,'   p act:=',(surface_ten/pw_r_coord[rp_max[k]]):8:4);
```

```
{ 2- iteration sequence }
        relax_count:=250;
        counter:=249;
        for i_it:=0 to n_iters_1 do
        begin
                simple;
                counter:=counter+1;
                writeln('hi i:=',i_it:6,'  w:=',w_vel[15,10]:15:6,'u:=',u_vel[15,10]:10:6,'q:=',p_cor[15,10]:10:6);
                writeln('hi i:=',i_it:6,'  w:=',w_vel[15,25]:15:6,'u:=',u_vel[15,25]:10:6,'q:=',p_cor[15,25]:10:6);
                if counter=relax_count then
                begin

                        writeln('hi i:=',i_it:6,'  w:=',w_vel[15,5]:15:6,'u:=',u_vel[15,5]:10:6,'q:=',p_cor[15,5]:10:6);
                        writeln('qu:=',p_cor[15,5]:10:4,'q d:=',p_cor[15,25]:10:8);
                        set_u_bound_1;
                        counter:=0;
                        kc=za_max+5;
                        writeln('u:=',u_vel[ru_max[k],k]:8:4,'  p:=',pressure[ru_max[k],k]:8:4);
                        k:=za_max+10;
                        writeln('u:=',u_vel[ru_max[k],k]:8:4,'  p:=',pressure[ru_max[k],k]:8:4);
                        kc=za_max+15;
                writeln('u:=',u_vel[ru_max[k],k]:8:4,'  p:=',pressure[ru_max[k],k]:8:4);
                        for k:=za_max to zb_max do
                        writeln('  p:=',pressure[rp_max[k],k]:8:4);
                end;
        end;}
        {file_oput('2pne3a');}

        for kc=za_max to zb_max do
                pressure[rp_max[k],k]:=p_ambt+(surface_ten/pw_r_coord[rp_max[k]]);

        counter:=0;
        relax_count:=50;
        for i_it:=0 to n_iters_2 do
        begin
                simple;
                counter:=counter+1;
                writeln('hi i:=',i_it:6,'  w:=',w_vel[15,10]:15:6,'u:=',u_vel[15,10]:10:6,'q:=',p_cor[15,10]:10:6);
                writeln('hi i:=',i_it:6,'  w:=',w_vel[15,25]:15:6,'u:=',u_vel[15,25]:10:6,'q:=',p_cor[15,25]:10:6);

                if counter=relax_count then begin
                        writeln('i:=',i_it:6,'  w:=',w_vel[15,10]:15:6,'u:=',u_vel[15,10]:10:6,'q:=',p_cor[15,10]:10:8);
                        writeln('qu:=',p_cor[15,5]:10:4,'q d:=',p_cor[15,25]:10:8);
                        set_u_bound;
                        counter:=0;
                        for kc=za_max to zb_max do begin
                                writeln('  p:=',pressure[rp_max[k],k]:8:4,' pem1:=',pressure[ru_max[k],k]:8:4);
                                writeln('old_uvel:=',u_vel[ru_max[k],k]:8:4);
                        end;
                end;
        end;


        for i_it:=0 to n_iters_2 do   begin
                simple;
                counter:=counter+1;
                writeln('hi i:=',i_it:6,'  w:=',w_vel[15,10]:15:6,'u:=',u_vel[15,10]:10:6,'q:=',p_cor[15,10]:10:6);
                 if counter=relax_count then begin
                        counter:=0;
                        kc=za_max+5;
                        writeln('u:=',u_vel[ru_max[k],k]:8:4,'  p:=',pressure[ru_max[k],k]:8:4);
                        kc=za_max+10;
                        writeln('u:=',u_vel[ru_max[k],k]:8:4,'  p:=',pressure[ru_max[k],k]:8:4);
                        k:=za_max+15;
                        writeln('u:=',u_vel[ru_max[k],k]:8:4,'  p:=',pressure[ru_max[k],k]:8:4);
                end;
                writeln('hi i:=',i_it:6,'  w:=',w_vel[15,10]:15:6,'u:=',u_vel[15,10]:10:6,'q:=',p_cor[15,10]:10:6);
         end;

        showw_vel;
        readln;
        showu_vel;
        readln;
        showp_cor; writeln;
        readln;
        showpressure; writeln;
        readln;

        file_oput('2pne3c');
        ouput_g('d1.dat');
end {simplerz}.
```

Appendix 3 The Quick Welding program

The program presented in this appendix is a shortened version of the program presented in appendix 2, which takes a number of hours to obtain a converged solution on a UNIX workstation, and cannot run on a PC given the time constraints. In order to obtain the relevant information a welder requires from a mathematical model (eg maximum welding speeds, penetration depth and weld width), a much quicker, more practical model that runs instantaneously on a PC was developed. The results of the comparison with the larger model and with experimental results are presented in chapter 4. The essence of the model is to calculate the keyhole dimensions as a function of depth down the keyhole, in terms of the energy balance alone. The ablating material velocity is calculated using an approximate formula, which enables the heat loss by vaporising the material to be taken into account, which together with conductive heat losses (calculated using the Davis-Noller solution) make up the energy lost from the system. The energy into the system is calculated from the Fresnel absorption at the keyhole walls together with the contribution from the plasma absorption by inverse Brehmsstrahlung, which is calculated using Dowden's approximate formula (Dowden, 1989). The resulting differential equation is solved using a Newton - Raphson root finder method, which enables the penetration depth and weld width to be found.

```
program weld(input,output);
{new formula for temp using q_l int q_p dc}
type
        keyhole_coord = record
                        r_c:real;
                        z_c:real;
        end;
        array_of_keyhole_coords=array[0..50] of keyhole_coord;
        weldbead_coord = record
                                x,y,z:real;
        end;
        array_of_weldbead_coords=array[0..50] of weldbead_coord;
const
        pi=3.14159;
        max_order=4;
        epsilon = 0.08;
        boiling_pt=3000;
        melting_pt=1600;
        ambient_temp=300;
        delta_T=boiling_pt-ambient_temp;
        conductivity=35.97;
        specific_heat=753;
        density=7200;
        alpha=conductivity/(density*specific_heat);
var
        velocity:real;
        max_power:real;
        beam_radius:real;
        delta_h:real;
        keyhole:array_of_keyhole_coords;
        weldbead:array_of_weldbead_coords;
{-------------------------------------------------------------}
{-----------------------Maths functions-----------------------}
{-------------------------------------------------------------}
function tan(a:real):real;
begin
        tan:=sin(a)/cos(a);
end;
{-------------------------------------------------------------}
{--------The following functions are used in the evaluation of the heat balance------}
{-------------------------------------------------------------}
function absorptivity(theta:real):real;
var     term_1_num,term_1_den:real;
        term_2_num,term_2_den:real;
        term_1,term_2:real;
begin
        {   absorptivity:=(theta/theta)*0.5;}
        term_1_num:=1+sqr(1-epsilon*cos(theta));
        term_1_den:=1+sqr(1+epsilon*cos(theta));
        term_1:=term_1_num/term_1_den;
        term_2_num:=sqr(epsilon)-2*epsilon*cos(theta)+2*sqr(cos(theta));
        term_2_den:=sqr(epsilon)+2*epsilon*cos(theta)+2*sqr(cos(theta));
        term_2:=term_2_num/term_2_den;
        absorptivity:=1-(term_1+term_2)/2;
end;
{-------------------------------------------------------------}
{-----------------These functions calculate a to the power of b----------------}
{-------------------------------------------------------------}
function raise(a,b:real):real;
{calculates a^b}
begin
        if b=0 then raise:=1
        else raise:=exp(b*ln(a));
end;
function raiseneg(a:real;b:integer):real;
{calcs a^b where a is a negative number}
var     j:integer;
        temp_raise:real;
begin
        temp_raise:=a;
        if b>1 then
        for j:=2 to b do
                temp_raise:=a*temp_raise;
        raiseneg:=temp_raise;
end;
{-------------------------------------------------------------}
{--------------------Modified Bessel function order-----------------------}
{-------------------------------------------------------------}
function Mod_Bessel_1_0 ( X: real ): real;
var     y,z:  real;
        i:integer;
        y_to_the_2_times:array[0..6] of real;
        z_to_the:array [0..8] of real;
begin
```

```
                y :=            X / 3.75;
                z :=            3.75 / X;
                y_to_the_2_times[0]:=1;
                for i:=1 to 6 do
                        y_to_the_2_times[i]:=sqr(y)*y_to_the_2_times[i-1];
                z_to_the[0]:=1;
                for i:=1 to 8 do
                        z_to_the[i]:=z_to_the[i-1]*z;
                if X <= 3.75 then Mod_Bessel_1_0:=  1.0 + 3.5156229 * y_to_the_2_times[1]
                        + 3.0899424 * y_to_the_2_times[2]
                        + 1.2067492 * y_to_the_2_times[3]
                        + 0.2659732 * y_to_the_2_times[4]
                        + 0.0360768 * y_to_the_2_times[5]
                        + 0.0045813 * y_to_the_2_times[6]
                else Mod_Bessel_1_0:=  exp( X ) * sqrt(1.0 / X) *
                        ( 0.39894228 + 0.01328592 * z
                        + 0.00225319 * z_to_the[2]
                        - 0.00157565 * z_to_the[3]
                        + 0.00916281 * z_to_the[4]
                        - 0.02057706 * z_to_the[5]
                        + 0.02635537 * z_to_the[6]
                        - 0.01647633 * z_to_the[7]
                        + 0.00392377 * z_to_the[8]);
end; {Mod_Bessel_1_0}
function Mod_Bessel_1_1 ( X: real ): real;
var             y, z:   real;
                i:integer;
                y_to_the_2_times:array[0..6] of real;
                z_to_the:array [0..8] of real;
begin
                y :=            X / 3.75;
                z :=            3.75 / X;
                y_to_the_2_times[0]:=1;
                for i:=1 to 6 do
                        y_to_the_2_times[i]:=sqr(y)*y_to_the_2_times[i-1];
                z_to_the[0]:=1;
                for i:=1 to 8 do
                        z_to_the[i]:=z_to_the[i-1]*z;
                if X <= 3.75 then Mod_Bessel_1_1:=  X * ( 0.5
                        + 0.87890594 * y_to_the_2_times[1]
                        + 0.51498869 * y_to_the_2_times[2]
                        + 0.15084934 * y_to_the_2_times[3]
                        + 0.02658733 * y_to_the_2_times[4]
                        + 0.00301532 * y_to_the_2_times[5]
                        + 0.00032411 * y_to_the_2_times[6] )

                else  Mod_Bessel_1_1:=  exp( X ) * sqrt(1.0 / X) *
                        ( 0.39894228 - 0.03988024 * z
                        - 0.00362018 * z_to_the[2]
                        + 0.00163801 * z_to_the[3]
                        - 0.01031555 * z_to_the[4]
                        + 0.02282967 * z_to_the[5]
                        - 0.02895312 * z_to_the[6]
                        + 0.01787654 * z_to_the[7]
                        - 0.00420059 * z_to_the[8])
end; }Mod_Bessel_1_1}

function Mod_Bessel_1 ( order: integer; X: real ): real;
var             I: integer;
                Buffer:array [0..max_order] of real;
begin
                if  order = 0  then  Mod_Bessel_1:= Mod_Bessel_1_0 ( X )
                else if  order = 1  then  Mod_Bessel_1:= Mod_Bessel_1_1 ( X )
                else begin
                        Buffer [ 0 ] := Mod_Bessel_1_0 ( X );
                        Buffer [ 1 ] := Mod_Bessel_1_1 ( X );
                        for I:= 2 to order  do
                                Buffer [ I ] := Buffer [ I-2 ] - ( 2.0 * (I-1) / X ) * Buffer [ I-1 ];
                        Mod_Bessel_1:=  Buffer [ order ]
                end;
end;            {Mod_Bessel_1}

function Mod_Bessel_2_0 ( X: real): real;
var             y, z:   real;
                i:integer;
                y_to_the_2_times:array[0..6] of real;
                z_to_the:array [0..8] of real;
begin
                y :=X / 2.0;
                z :=2.0 / X;
                z_to_the[0]:=1;
                y_to_the_2_times[0]:=1;
                for i:=1 to 6 do begin
                        y_to_the_2_times[i]:=sqr(y)*y_to_the_2_times[i-1];
```

```
                        z_to_the[i]:=z_to_the[i-1]*z;
                end;
                if X <= 2.0
                then Mod_Bessel_2_0:=  -ln( y ) * Mod_Bessel_1 ( 0, X )
                        - 0.57721566
                        + 0.42278419 * y_to_the_2_times[1]
                        + 0.23069756 * y_to_the_2_times[2]
                        + 0.03488589 * y_to_the_2_times[3]
                        + 0.00262698 * y_to_the_2_times[4]
                        + 0.00010750 * y_to_the_2_times[5]
                        + 0.00000740 * y_to_the_2_times[6]
                else  Mod_Bessel_2_0:=  exp( -X ) * sqrt(1.0 / X)
                        * ( 1.25331414 - 0.07832358 * z
                        + 0.02189568 * z_to_the[2]
                        - 0.01062446 * z_to_the[3]
                        + 0.00587872 * z_to_the[4]
                        - 0.00251540 * z_to_the[5]
                        + 0.00053208 * z_to_the[6])
end; {Mod_Bessel_2_0}

function Mod_Bessel_2_1 (X: real): real;
var             y,z :   real;
                i:integer;
                y_to_the_2_times:array[0..6] of real;
                z_to_the:array [0..8] of real;
begin
                y :=X / 2.0;
                z :=2.0 / X;
                z_to_the[0]:=1;
                y_to_the_2_times[0]:=1;
                for i:=1 to 6 do begin
                        y_to_the_2_times[i]:=sqr(y)*y_to_the_2_times[i-1];
                        z_to_the[i]:=z_to_the[i-1]*z;
                end;
                if X <= 2.0
                then Mod_Bessel_2_1:=  ( X * ln( y )* Mod_Bessel_1 ( 1, X ) + 1.0
                        + 0.15443144 * y_to_the_2_times[1]
                        - 0.67278579 * y_to_the_2_times[2]
                        - 0.18156897 * y_to_the_2_times[3]
                        - 0.01919402 * y_to_the_2_times[4]
                        - 0.00110404 * y_to_the_2_times[5]
                        - 0.00004686 * y_to_the_2_times[6]) / X
                else  Mod_Bessel_2_1:=  exp( -X ) * sqrt(1.0 / X) *
                        ( 1.25331414 + 0.23498619 * z
                        - 0.03655620 * z_to_the[2]
                        + 0.01504268 * z_to_the[3]
                        - 0.00780353 * z_to_the[4]
                        - 0.00325614 * z_to_the[5]
                        - 0.00068245 * z_to_the[6])
end;            {Mod_Bessel_2_1}

function Mod_Bessel_2 ( order: integer; X: real ): real;
var             I:integer;
                Buffer:array[0..max_order] of real;
begin
                if  order = 0  then  Mod_Bessel_2:= Mod_Bessel_2_0 ( X )
                else if  order = 1  then  Mod_Bessel_2:= Mod_Bessel_2_1 ( X )
                else begin
                        Buffer [ 0 ] := Mod_Bessel_2_0 ( X );
                        Buffer [ 1 ] := Mod_Bessel_2_1 ( X );
                        for I:= 2 to order do
                                Buffer [ I ] := Buffer [ I-2 ] + ( 2.0 * (I-1) / X ) * Buffer [ I-1 ];
                        Mod_Bessel_2:=  Buffer [ order ]
                end;
end;            {Mod_Bessel_2}

function cyl(x:real):real;
var             temp:real;
                i:integer;
begin
                temp:=Mod_Bessel_1(0,x)/Mod_Bessel_2(0,x);
                for i:=1 to max_order do
                        temp:=temp+(Mod_Bessel_1(i,x)/Mod_Bessel_2(i,x))*2*raiseneg(-1,i);
                cyl:=temp;
end;

{-------------------------------------------------------------}
{-----------functions and procedures to find the melt temp isotherm ----------}
{-------------------------------------------------------------}
function l_prime(l:real):real;
begin
                l_prime:=l*velocity/(2*alpha);

end;
```

```
function dist_from_z_axis(x,y:real):real;
begin
        dist_from_z_axis:=sqrt(sqr(l_prime(x))+sqr(l_prime(y)));
end;
function dist_from_c(x,y,z,c:real):real;
begin
        dist_from_c:= sqrt(sqr(l_prime(x))+sqr(l_prime(y))+sqr(l_prime(z-c)));
end;
function temp_due_to_point_source(point_source_strength,x,y,z,c:real):real;
begin
        temp_due_to_point_source := point_source_strength
                *velocity/(2*alpha*conductivity)
                *exp(l_prime(x)-dist_from_c(x,y,z,c))/(4*pi*dist_from_c(x,y,z,c));
end;
function temp_due_to_line_source(line_source_strength,x,y:real):real;
begin
        temp_due_to_line_source:=line_source_strength/(2*pi*conductivity)
                *Mod_Bessel_2_0(dist_from_z_axis(x,y))*exp(l_prime(x));
end;
function temperature(x,y,z,c,point_source_strength,line_source_strength:real):real;
begin
        temperature:=temp_due_to_line_source(line_source_strength,x,y)+ambient_temp
        +temp_due_to_point_source(point_source_strength,x,y,z,c);
end;
function F_of_y(x,y,z,c,point_source_strength,line_source_strength:real):real;
var     term_1,term_2:real;
begin
        term_1:=temperature(x,y,z,c,point_source_strength,line_source_strength);
        term_2:=-melting_pt;
        F_of_y:=term_1+term_2;
end;
function dF_by_dy(x,y,z,c,point_source_strength,line_source_strength:real):real;
var     delta_y:real;
begin
        delta_y:=0.01/1000;
        dF_by_dy:=(F_of_y(x,y+delta_y,z,c,point_source_strength,line_source_strength)
                -F_of_y(x,y,z,c,point_source_strength,line_source_strength))/delta_y;
end;

procedure calc_weld_profile(z,cyl_pe:real;k_weld:integer);
var     x,y,c:real;
        delta_x,delta_y:real;
        point_source_strength,line_source_strength:real;
        isotherm_y,old_isotherm_y:real;
        max_y,old_max_y:real;
        dum:integer;
        e_min,e_max:real;
begin
        line_source_strength:=2*pi*conductivity*(boiling_pt-ambient_temp)*cyl_pe;
        point_source_strength:=0;

        delta_x:=0.00001;
        delta_y:=0.00001;
        x:=0;
        y:=0;
        c:=0;

        weldbead[k_weld].z:=z;
        x:=0.0001;
        max_y:=0.0;
        old_max_y:=0.00001;

        e_min:=0;
        e_max:=0.01;
        for dum:=1 to 20 do begin
                max_y:=0.5*(e_min+e_max);
                if temperature(x,max_y,z,c,point_source_strength,line_source_strength)<melting_pt
                then e_max:=max_y
                else e_min:=max_y;
        end;
        isotherm_y:=0.5*(e_min+e_max);
        x:=x+delta_x;
        repeat
                old_max_y:=isotherm_y;
                old_isotherm_y:=isotherm_y;
                for dum:=1 to 3 do begin
                        max_y:=old_max_y-(F_of_y(x,old_max_y,z,c,point_source_strength,line_source_strength)
                                /dF_by_dy(x,old_max_y,z,c,point_source_strength,line_source_strength));
                        old_max_y:=max_y;
                end;
                isotherm_y:=max_y;
                x:=x+delta_x;
        until (isotherm_y<old_isotherm_y);
```

```
                isotherm_y:=old_isotherm_y;
                x:=x-delta_x;
                weldbead[k_weld].x:=x;
                weldbead[k_weld].y:=old_isotherm_y*2;
                y:=old_isotherm_y;
end;

{-------------------------------------------------------------------}
{----------functions and procedures to find the keyhole dimensions----------}
{-------------------------------------------------------------------}
function minor_f_of_r(r_val:real):real;
var l:real;
begin
                 l:=alpha/velocity;
                minor_f_of_r:=delta_T*pi*2*conductivity*cyl(r_val/(2*l));
end;

function find_theta(dum_z,last_z,delta_r:real):real;
var             dz_by_dr:real;
begin
                dz_by_dr:=abs((dum_z-last_z)/delta_r);
                find_theta:=(arctan(dz_by_dr));
end;

function abs_power_pud(r_val:real):real;
begin
                abs_power_pud:=minor_f_of_r(r_val);
end;

function power(trans_power{,r_val,guess_z,last_z}:real):real;
begin
                power:=trans_power{+abs_power_pud(r_val)*(guess_z-last_z)};
end;

function major_f_of_rz(trans_power,dum_z,last_z,delta_r,mid_r:real):real;
var             theta:real;
                term_1,term_2:real;
begin
                term_1:=dum_z-last_z;
                theta:=find_theta(dum_z,last_z,delta_r);
                term_2:=(2*delta_r*absorptivity(theta)*power(trans_power{,mid_r,dum_z,last_z}))
                                /(mid_r*minor_f_of_r(mid_r));
                 major_f_of_rz:=term_1+term_2;
end;

function dF_by_dz(trans_power,guess_z,last_z,delta_r,mid_r:real):real;
var             delta_z:real;
begin
                delta_z:=0.01/1000;
                dF_by_dz:=(major_f_of_rz(trans_power,guess_z+delta_z,last_z,delta_r,mid_r)
                                -major_f_of_rz(trans_power,guess_z,last_z,delta_r,mid_r))/delta_z;
end;

{-------------------------------------------------------------}
{----------Performs Newton Raphson method to find dr/dz----------}
{-------------------------------------------------------------}
procedure newt;
var             n,kc:integer;
                guess_z,new_guess_z:real;
                last_z:real;
                delta_r:real;
                mid_r:real;
                l:real;
                cyl_pe:real;
                k_weld:integer;
                trans_power:real;
begin
                trans_power:=max_power;
                delta_r:=-(beam_radius-beam_radius/3)/50;
                k_weld:=0;
                kc=0;
                keyhole[k].z_c:=0;
                keyhole[k].r_c:=beam_radius;
                last_z:=keyhole[k].z_c;
                guess_z:=keyhole[k].z_c+abs((delta_r*tan((8/9)*(pi/2))));

                l:=alpha/velocity;
                cyl_pe:=cyl(keyhole[k].r_c/(2*l));
                calc_weld_profile(keyhole[k].z_c,cyl_pe,k_weld);
                k_weld:=k_weld+1;

                kc=1;
                keyhole[1].r_c:=beam_radius+delta_r;
                mid_r:=(keyhole[0].r_c+keyhole[1].r_c)/2;
```

```
                writeln('n:=',n:6,'  val1:=',guess_z:12:10);
                for n:=1 to 5 do begin
                new_guess_z:=guess_z-(major_f_of_rz(trans_power,guess_z,last_z,delta_r,mid_r)
                                /dF_by_dz(trans_power,guess_z,last_z,delta_r,mid_r));
                guess_z:=new_guess_z;
                end;
                trans_power:=trans_power-(abs_power_pud(mid_r)*(guess_z-last_z));
                keyhole[1].z_c:=guess_z;
                l:=alpha/velocity;
                cyl_pe:=cyl(keyhole[k].r_c/(2*l));
                calc_weld_profile(keyhole[k].z_c,cyl_pe,k_weld);
                k_weld:=k_weld+1;
                kc:=2;

                repeat
                keyhole[k].r_c:=beam_radius+(k*delta_r);
                last_z:=keyhole[k-1].z_c;
                guess_z:=(keyhole[k-1].z_c-keyhole[k-2].z_c)+keyhole[k-1].z_c;
                mid_r:=(keyhole[k].r_c+keyhole[k-1].r_c)/2;
                                for n:=1 to 5 do begin
                                new_guess_z:=guess_z-(major_f_of_rz(trans_power,guess_z,keyhole[k-1].z_c,delta_r,mid_r)
                                  /dF_by_dz(trans_power,guess_z,keyhole[k-1].z_c,delta_r,mid_r));
                                guess_z:=new_guess_z;
                end;
                keyhole[k].z_c:=guess_z;
                                trans_power:=trans_power-(abs_power_pud(mid_r)*(guess_z-last_z));
                                l:=alpha/velocity;
                                cyl_pe:=cyl(keyhole[k].r_c/(2*l));
                                calc_weld_profile(keyhole[k].z_c,cyl_pe,k_weld);
                                k_weld:=k_weld+1;
                kc:=k+1;
                until k>50;
                kc:=k-1;
                writeln('pd:=',keyhole[k].z_c:8:6);
                writeln('power:=',trans_power:10:4);
                readln;
end;

{-------------------------------------------------------------------------}
{-------------------------Data Input routine-------------------------------}
{-------------------------------------------------------------------------}
procedure input_data;
var dummy_vel,dummy_rad:real;
begin
                write('Please enter te maximum power(W)  ');
                readln(max_power);
                write('Please enter the velocity(mm/s) ');
                readln(dummy_vel);
                velocity:=dummy_vel/1000;
                writeln('vel=',velocity:8:4);
                write('Please enter the beam radius(mm)  ');
                readln(dummy_rad);
                beam_radius:=dummy_rad/1000;
end;

{-------------------------------------------------------------------------}
{-------------------------Output to screen--------------------------------}
{-------------------------------------------------------------------------}
procedure ouput;
var             i:integer;
begin
                for i:=0 to 50 do
                                writeln('wx=',(weldbead[i].x*1000):8:4,'   wy:=',(weldbead[i].y*1000):8:4
                                  ,'   wz:=',(weldbead[i].z*1000):8:4);
                readln;
                 for i:=0 to 50 do
                                writeln('kr=',(keyhole[i].r_c*1000):8:4,'   kz:=',  (keyhole[i].z_c*1000):8:4);
                readln;
end;
{-------------------------------------------------------------------------}
{-------------------------Main Procedure----------------------------------}
{-------------------------------------------------------------------------}
begin
                input_data;
                newt;
                ouput;
end.
```

Appendix 4 Tables of Experimental Results

4.1 Results from the CO_2 lasers at Liverpool University

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
F1	2.5	25.0	Ar,20	1.44	0.8	4.11	182	1263
F2	2.5	16.67	Ar,20	1.75	1.2	4.75	132	1462
F3	2.5	12.5	Ar,20	2.07	1.42	5.02	141	1413
F4	2.5	8.33	Ar,20	2.95	1.96	5.78	206	1531
F5	2.5	4.2	Ar,20	6.0	4.97	6.0		
F6	2.5	25.0	He,20	1.65	0.95	4.35	225	1528
F7	2.5	16.67	He,20	2.94	1.24	4.69	626	1480
F8	2.5	12.5	He,20	2.94	1.81	5.09	329	1726
F9	2.5	8.33	He,20	3.05	2.15	5.00	193	1419
F10	2.5	4.2	He,20	3.05	3.84	6.00	0	1577
F11	1.5	16.67	Ar,20	1.69	1.08	3.69	141	1048
F12	1.5	12.5	Ar,20	1.86	1.35	3.90	101	1057
F13	1.5	8.33	Ar,20	2.4	1.5	4.5	157	986
F14	1.5	4.2	Ar,20	4.05	3.3	4.35	118	1026
F15	1.5	1.67	Ar,20	3.9	6.0	6.0	0	1141
F16	1.5	16.67	He,20	1.74	1.41	3.49	80	1218
F17	1.5	12.	He,20	1.81	1.92	4.18	0	1485
F18	1.5	8.33	He,20	2.5	2.30	4.30	37	1283
F19	1.5	4.22	He,20	3.9	3.3	4.50	92	1065
F20	1.5	1.67	He,20	6.0	5.7	6.0	32	1105

Table A4.1: Results from welding 6mm mild steel samples with a CO2 laser at powers of 2.5 and 1.5kW

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
L1	2.2	4.116	Ar,20	4.0	1.72	4.75	342	729
L2	2.2	6.16	Ar,20	4.85	3.9	4.3		
L3	2.2	8.33	Ar,20	3.25	2.25	4.65	226	1365
L4	2.2	16.67	Ar,20	1.70	1.35	4.65	83	1568
L5	2.2	25.0	Ar,20	1.40	0.95	4.55	126	1598
L6	2.2	33.0	He,20					
L7	2.2	41.67	He,20	2.5	0.95	3.1	1099	1648
L8	2.2	50.0	He,20	1.05	0.8	2.35	103	1259
L9	2.2	58.33	He,20	1.32	0.9	2.3	246	1539
L10	2.2	66.67	He,20	1.0	0.43	2.35	280	956
L11	2.2	4.116	Ar,20	5.64	2.57	6.0	598	1179
L12	2.2	6.16	Ar,20	3.43	2.67	6.0	141	1600
L13	2.2	8.33	Ar,20	2.75	1.70	5.75	205	1375
L14	2.2	16.67	Ar,20	2.05	1.025	3.8	276	1039
L15	2.2	25.0	Ar,20	1.65	1.2	3.4	147	1442
L16	2.2	33.0	He,20	1.55	1.1	3.3	179	1635
L17	2.2	41.67	He,20	1.5	0.85	2.14	308	1039
L18	2.2	50.0	He,20	0.65	0.85	1.98	0	1115
L19	2.2	58.33	He,20	0.85	0.65	1.93	78	988
L20	2.2	66.67	He,20	0.92	0.88	1.85	19	1550

Table A4.2: Results from welding 6mm mild steel samples with a CO_2 laser at a power of 2.2 kW

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
L21	1.5	1.67	He,20	5.94	4.1	4.11	196	622
L22	1.5	3.33	He,20	4.85	3.70	3.55	160	721
L23	1.5	5.	He,20	3.20	2.50	3.16	104	694
L24	1.5	6.67	He,20	2.95	2.7	3.0	44	855
L25	1.5	8.33	He,20	1.70	1.2	3.4	0	642
L26	1.5	12.5	He,20	1.65	1.3	3.3	63	870
L27	1.5	16.67	He,20	1.50	1.15	3.05	74	908
L28	1.5	20.85	He,20	1.50	0.95	3.10	139	946
L29	1.5	25	He,20	1.05	0.53	3.33	111	756
L30	1.5	33.3	He,20	1.05	0.45	2.67	162	661
L31	1.5	41.67	He,20	0.75	0.45	2.25	75	654
L32	1.5	50.0	He,20	0.85	0.43	1.92	137	620
L33	1.5	1.67	Ar,20	6.00	4.0	5.1	215	760
L34	1.5	3.33	Ar,20	3.5	3.0	4.5	69	849
L35	1.5	5.0	Ar,20	3.5	2.05	3.92	229	754
L36	1.5	6.67	Ar,20	2.9	1.9	3.5	172	775
L37	1.5	8.33	Ar,20	2.0	1.75	3.15	38	769
L38	1.5	12.5	Ar,20	3.5	2.0	3.7	508	1357
L39	1.5	16.67	Ar,20	2.0	1.50	3.15	136	1155
L40	1.5	20.85	Ar,20	1.9	1.20	2.85	10	877
L41	1.5	25	Ar,20	1.25	0.85	3.4	101	1095
L42	1.5	33.3	Ar,20	0.95	0.75	2.75	52	1004
L43	1.5	41.67	Ar,20	0.95	0.70	2.25	79	930
L44	1.5	50.0	Ar,20	0.83	0.70	1.88	43	902

Table A4.3 Results from welding 6mm mild steel sample with a CO_2 laser using a power of 1.5 kW

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
M35	1.1	5	He,20	2.95	1.0	2.85	272	357
M36	1.1	8	He,20	2.3	1.1	2.75	197	479
M37	1.1	10	He,20	1.65	1.1	2.25	83	451
M38	1.1	15	He,20	1.25	0.95	2.5	50	399
M39	1.1	20	He,20	1.25	0.95	2.30	63	680
M40	1.1	5	Ar,20	1.75	1.05	2.9	68	375
M41	1.1	8	Ar,20	3.3	2.25	3.2	232	911
M42	1.1	10	Ar,20	2.35	1.64	2.25	142	598
M43	1.1	15	Ar,20	2.1	1.8	2.20	78	859
M44	1.1	20	Ar,20	1.25	1.0	2.20	53	676

Table A4.4 Results from welding 6mm mild steel samples with a CO2 laser using a power of 1.1 kW and a spotsize of 0.3mm

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
C26	1.1	5	He,20	2.90	2.44	2.55	64	551
C27	1.1	8	He,20	1.80	1.90	2.55	0	641
C28	1.1	10	He,20	1.86	1.63	1.86	38	492
C29	1.1	15	He,20					
C30	1.1	20	He,20	1.44	1.22	1.77	63	636
C31	1.1	5	Ar,20	4.15	3.30	3.4	156	909
C32	1.1	8	Ar,20	1.47	1.80	2.60	0	636
C33	1.1	10	Ar,20	1.55	1.50	2.35	7	585
C34	1.1	15	Ar,20	2.35	1.85	2.35	144	936
C35	1.1	20	Ar,20	1.28	1.40	1.98	0	794

Table A4.5 Results welding 6mm mild steel samples with a CO_2 laser using 1.1kW and a spotsize of 0.4mm

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
N35	2.5	4.2	He,20	5.5	3.5	6.0	395	1436
N36	2.5	8.33	He,20	3.4	2.7	6.0	1166	2026
N37	2.5	12.5	He,20	2.2	1.75	5.8	104	1915
N38	2.5	16.67	He,20	1.95	1.36	5.65	156	1916
N39	2.5	25.0	He,20	1.5	0.95	4.25	163	1493
N40	2.5	4.2	Ar,20	5.8	4.45	5	287	1429
N41	2.5	8.33	Ar,20	4.45	2.25	4.9	688	143
N42	2.5	12.5	Ar,20	3.75	2.125	4.75	587	1828
N43	2.5	16.67	Ar,20	2.55	1.45	4.4	364	1571
N44	2.5	25.0	Ar,20	1.75	1.0	3.75	254	1372

Table A4.6 Results from welding 6mm mild steel samples with a CO_2 laser using 2.5 kW

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
D68	1.5	58.33	Ar,20	0.78	0.35	1.75	148	543
D69	1.5	50	Ar,20	0.87	0.43	2.28	147	736
D70	1.5	41.67	Ar,20	0.95	0.50	2.50	138	790
D71	1.5	33.33	Ar,20	0.95	0.66	3.04	75	1003
D72	1.5	25	Ar,20	1.23	0.73	3.09	124	885
D73	1.5	16.67	Ar,20	1.34	0.95	3.36	75	869
D74	1.5	12.5	Ar,20	1.68	0.89	3.84	142	777
D75	1.5	8.33	Ar,20	2.67	1.34	4.02	251	816
D76	1.5	58.33	He,20	0.76	0.65	1.3	309	665
D77	1.5	50	He,20	0.81	0.54	2.04	86	790
D78	1.5	41.67	He,20	0.89	0.78	2.27	339	1025
D79	1.5	33.33	He,20	0.61	0.66	2.89	0	954
D80	1.5	25	He,20	0.92	1.21	3.7	0	1500
D81	1.5	16.67	He,20	1.29	1.45	3.89	0	1388
D82	1.5	12.5	He,20	2.35	1.74	4.25	148	1397
D83	1.5	8.33	He,20	1.50	1.23	2.45	33	470

Table A4.7 Results from welding 5mm mild steel samples with a CO_2 laser using 1.5 kW

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
D84	2.5	58.33	Ar,20	0.66	0.79	2.71	0	1622
D85	2.5	50.0	Ar,20	0.79	0.74	3.33	16	1673
D86	2.5	41.67	Ar,20	0.92	0.71	3.33	64	1392
D87	2.5	33.3	Ar,20	1.10	0.92	3.68	53	1587
D88	2.5	25	Ar,20	1.17	1.25	5.0	0	2192
D89	2.5	16.67	Ar,20	1.10	1.33	5.0	0	1492
D92	2.5	58.33	He,20	0.77	0.51	3.27	91	1370
D93	2.5	50.0	He,20	0.92	0.78	3.68	31	1931
D94	2.5	41.67	He,20	1.0	0.71	4.16	95	1739
D95	2.5	33.3	He,20	1.25	0.83	4.33	138	1716
D96	2.5	25	He,20	1.21	0.95	4.48	65	1573
D97	2.5	16.67	He,20	1.29	0.96	5.0	61	1302

Table A4.8 Results from welding 5mm mild steel samples with a CO_2 laser using 2.5 kW

Code	Power (kW)	Speed (mm/s)	Gas flow rate (l/min)	Top weld width (mm)	Stem weld width (mm)	Penetration depth (mm)	Point source strength (W)	Line source strength (W)
M19	2.0	16.67	He, 20	1.28	0.95	3.0	61	775
M20	2.0	25	He,20	1.05	1.4	3.0	0	1444
M21	2.0	33.33	He,20	0.8	0.85	3.0	0	1154
M22	2.0	41.67	He, 20	0.7	1.1	3.0	0	1800
M23	2.0	16.67	Ar,20	1.5	1.05	3.0	95	834
M24	2.0	25	Ar,20	1.0	0.8	3.0	42	922
M25	2.0	33.33	Ar,20	0.9	0.75	3.0	38	1087
M26	2.0	41.67	Ar,20	0.85	1.0	3.0	0	1675

Table A4.9 Results from welding 3mm mild steel samples with a CO_2 laser using 2.0kW and a 100mm KCl lens.

Code	Power (kW)	Speed (mm/s)	Gas flow rate (l/min)	Top weld width (mm)	Stem weld width (mm)	Penetration depth (mm)	Point source strength (W)	Line source strength (W)
N19	2.0	16.67	He, 20	1.60	1.0	3.0	133	805
N20	2.0	25	He,20	1.50	0.95	3.0	163	1053
N21	2.0	33.33	He,20	1.45	0.95	3.0	188	1327
N22	2.0	41.67	He, 20	1.4	0.95	2.9	204	1552
N23	2.0	16.67	Ar,20	1.5	1.10	3.0	85	864
N24	2.0	25	Ar,20	1.90	1.30	2.75	223	1245
N25	2.0	33.33	Ar,20	1.20	0.9	2.8	96	1285
N26	2.0	41.67	Ar,20	1.10	0.75	2.7	126	1181

Table A4.10 Results from welding 3mm mild steel samples with a CO_2 laser using 2.0kW and a 150mm KCl lens.

Code	Power (kW)	Speed (mm/s)	Gas flow rate (l/min)	Top weld width (mm)	Stem weld width (mm)	Penetration depth (mm)	Point source strength (W)	Line source strength (W)
M11	1.65	58.33	He, 20	0.8	0.89	2.0	0	1326
M12	1.65	50	He,20	0.89	1.1	2.0	0	1389
M13	1.65	45	He,20	0.85	0.58	2.0	82	755
M14	1.65	41.33	He, 20	0.77	0.66	2.0	29	748
M15	1.65	58.33	Ar,20	0.88	0.83	2.0	20	1252
M16	1.65	50	Ar,20	0.85	0.60	2.0	84	845
M17	1.65	45	Ar,20	0.75	0.65	2.0	38	828
M18	1.65	41.33	Ar,20	0.75	0.55	2.0	50	676

Table A4.11 Results from welding 2mm mild steel samples with a CO_2 laser using 1.65kW and a 100mm KCl lens.

Code	Power (kW)	Speed (mm/s)	Gas flow rate (l/min)	Top weld width (mm)	Stem weld width (mm)	Penetration depth (mm)	Point source strength (W)	Line source strength (W)
N11	1.65	41.33	He, 20	1.1	0.5	2.0	207	628
N12	1.65	45.0	He, 20	1.1	0.65	2.0	171	828
N13	1.65	50.0	He,20	1.1	0.7	1.9	169	912
N14	1.65	58.33	He, 20	0.95	0.85	1.3	44	830
N15	1.65	58.33	Ar,20	1.125	0.75	1.8	188	1032
N16	1.65	50	Ar,20	1.0	0.65	2	136	902
N17	1.65	45	Ar,20	1.02	0.6	2	139	776
N18	1.65	41.33	Ar,20	1.15	0.7	2	164	815

Table A4.12 Results from welding 2mm mild steel samples with a CO_2 laser using 1.65kW and a 150mm KCl lens.

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Penetration depth (mm)
C56	0.63	5	Ar,20	0.84	0.9
C57	0.63	8	Ar,20	0.8	0.72
C58	0.63	10	Ar,20	0.5	0.46
C59	0.63	15	Ar,20	0.55	0.38
C60	0.63	20	Ar,20	0.25	0.26
C61	0.63	5	He,20	0.8	0.75
C62	0.63	8	He,20	0.6	0.50
C63	0.63	10	He,20	0.4	0.50
C64	0.63	15	He,20	0.33	0.52
C65	0.63	20	He,20	0.33	0.25

Table A4.13 Results from welding 2mm mild steel samples with a CO_2 laser using 0.63kW.

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Penetration depth (mm)
C1	0.65	5	Ar,20	2.5	1.0
C2	0.65	8	Ar,20	1.2	1.0
C3	0.65	10	Ar,20	1.14	0.83
C4	0.65	15	Ar,20	0.74	0.5
C5	0.65	20	Ar,20	0.45	0.45
C21	0.65	5	He,20	0.75	1.0
C22	0.65	8	He,20	0.6	0.75
C23	0.65	10	He,20	0.55	0.65
C24	0.65	15	He,20	0.65	0.45
C25	0.65	20	He,20	0.55	0.4

Table A4.14 Results from welding 1mm mild steel samples with a CO_2 laser using 0.65kW.

4.2 Results from the CO lasers at Heriot Watt and TWI

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
AC63	2.1	6.17	He,20					
AC65	2.1	8.33	He,20					
AC66	2.1	16.67	He,20					
AC67	2.1	25.0	He,20					

Table A4.15 Results from welding 6mm mild steel with a CO laser using 2.1 kW.

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
AC55	2.1	4.16	He,100	3.3	2.7	6	80	1227
AC56	2.1	8.33	He,100	2.7	2.25	4.8	88	1409
AC57	2.1	6.17	He,100	3.2	2.5	5.0	123	1272
AC58	2.1	5.0	He,100	3.6	2.8	6	130	1426
AC59	2.1	12.5	He,100	1.95	1.85	3.6	21	1241
AC60	2.1	16.67	He,100	1.7	1.65	3.6	0	1424

Table A4.16 Results from welding 6mm mild steel with a CO laser using 2.1 kW.

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
AC98	1.5	1.67	Ar,20	4.4	4.4	6.0	0	945
AC99	1.5	3.33	Ar,20	3.45	3.2	6.0	29	1186
AC100	1.5	5.0	Ar,20	3.3	2.5	5.5	122	1208
AC101	1.5	6.67	Ar,20	2.5	2.4	4.6	15	1202
AC102	1.5	8.33	Ar,20	2.6	2.3	4.3	57	1283
AC103	1.5	12.5	Ar,20	2.0	1.1	3.8	187	889
AC104	1.5	16.67	Ar,20	1.3	1.2	3.7	19	1139
AC105	1.5	20.85	Ar,20	2.1	1.0	3.1	269	984
AC106	1.5	25.0	Ar,20	1.3	0.7	2.9	155	804
AC107	1.5	33.33	Ar,20	1.0	0.6	2.8	106	859
AC108	1.5	41.67	Ar,20	0.9	0.5	2.55	117	806
AC109	1.5	50.0	Ar,20	0.9	0.43	2.5	155	838
AC110	1.5	58.67	Ar,20	0.9	0.75	1.35	62	778
AC111	1.5	66.67	Ar,20	0.8	0.5	1.35	123	620

Table A4.17 Results from welding 6mm mild steel with a CO laser using 1.5 kW and argon shroud.

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
AC112	1.5	66.67	He,20	0.9	0.6	2.5	138	708
AC113	1.5	58.3	He,20	0.9	0.8	1.25	42	759
AC114	1.5	50.0	He,20	1.2	0.85	1.55	162	873
AC115	1.5	41.67	He,20	0.9	0.7	2.4	60	992
AC116	1.5	33.3	He,20	1.2	0.7	2.7	156	933
AC117	1.5	25.0	He,20	1.2	0.9	2.7	74	909
AC118	1.5	20.85	He,20	1.7	1.45	2.9	72	1238
AC119	1.5	16.67	He,20	1.55	1.0	3.5	118	938
AC120	1.5	12.5	He,20	1.8	1.45	3.5	68	1001
AC121	1.5	8.33	He,20	2.55	2.2	4.2	65	1212
AC122	1.5	6.67	He,20	3.0	2.5	4.7	90	1265
AC123	1.5	5.0	He,20	3.0	2.7	5.2	43	1204
AC124	1.5	3.33	He,20	3.65	2.7	6.0	116	1064
AC125	1.5	1.67	He,20	5.5	3.9	6.0	162	883

Table A4.18 Results from welding 6mm mild steel with a CO laser using 1.5 kW and helium shroud.

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
AC129	2.1	4.16	Ar,20	3.45	3.3	6.0	20	1406
AC130	2.1	6.17	Ar,20	3.3	2.2	5.55	196	1289
AC131	2.1	8.33	Ar,20	1.95	1.65	5.05	45	1182
AC132	2.1	12.5	Ar,20	2.5	2.3	4.5	52	1846
AC133	2.1	16.67	Ar,20	1.65	1.1	3.4	125	979
AC134	2.1	25.0	Ar,20	1.4	1.0	3.4	113	1244
AC135	2.1	33.3	Ar,20	1.1	0.65	3.3	130	1076
AC136	2.1	41.67	Ar,20	1.0	0.55	3.5	146	1199
AC137	2.1	50.0	Ar,20	1.0	0.5	2.9	189	1057
AC138	2.1	58.3	Ar,20	1.0	0.8	2.2	91	1336
AC139	2.1	66.67	Ar,20	1.1	0.8	1.7	169	1150

Table A4.19 Results from welding 6mm mild steel with a CO laser using 2.1 kW and argon shroud

4mm CO laser

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
AC48	2.0	29.16	He,20	1.4	0.85	3.2	176	1162
AC49	2.0	25.0	He,20	1.2	0.85	4.0	86	1288
AC50	2.0	20.85	He,20	1.1	1.1	4.0	0	11368
AC51	2.0	16.67	He,20	1.25	1.15	4.0	18	1192
AC52	2.0	16.67	He,20	1.5	1.1	4.0	84	1152
AC53	2.0	25.0	He,20	1.6	1.1	3.6	158	1422
AC54	2.0	33.3	He,20	1.1	0.55	3.4	157	975
AC148	2.0	16.67	Ar,20	1.55	1.3	4.0	55	1311
AC149	2.0	20.85	Ar,20	1.5	1.2	3.35	77	1228
AC150	2.0	25.0	Ar,20	1.4	1.0	3.4	113	1225
AC151	2.0	29.16	Ar,20	1.4	0.95	3.55	145	1331
AC152	2.0	29.16	Ar,20	1.2	1.0	3.1	58	1283
AC153	2.0	20.67	Ar,20	1.7	1.45	4.0	72	1696
AC154	2.0	25.0	Ar,20	1.45	0.95	4.0	144	1405
AC155	2.0	16.67	Ar,20	1.65	1.0	4.0	147	1072

Table A4.20 Results from welding 4mm mild steel with a CO laser using 2.0 kW

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
AC140	2.0	25	Ar,20	1.2	0.85	3.0	86	966
AC141	2.0	33.3	Ar,20	1.2	0.65	3.0	170	978
AC142	2.0	41.67	Ar,20	1.0	0.6	2.6	130	949
AC143	2.0	16.67	Ar,20	1.5	1.3	3.0	42	982
AC144	2.0	16.67	Ar,20	2.1	1.6	3.0	142	1158
AC145	2.0	25.0	Ar,20	1.6	1.2	3.0	128	1272
AC146	2.0	33.3	Ar,20	1.05	0.7	2.8	98	968
AC147	2.0	41.67	Ar,20	1.1	0.6	2.6	176	949

Table A4.21 Results from welding 3mm mild steel with a CO laser using 2.0 kW.

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
AC83	1.7	50	Ar,20	1.0	1.05	1.9	0	1272
AC84	1.7	41.33	Ar,20	0.9	1.05	2.0	0	1147
AC85	1.7	45	Ar,20	0.85	1.1	2.0	0	1277
AC86	1.7	33.3	Ar,20	0.95	0.8	2.0	39	769
AC87	1.7	33.3	Ar,20	1.1	0.7	2.0	116	691
AC88	1.7	41.33	Ar,20	1.25	0.9	2.0	141	1010
AC89	1.7	45.0	Ar,20					
AC90	1.7	45.0	Ar,20	1.0	1.05	2.0	0	1229
AC91	1.7	50.0	Ar,20	1.5	1.3	1.9	118	15033
AC92	1.7	58.33	Ar,20	1.3	1.1	1.3	118	1019
AC21	1.7	66.67	He,20	0.9	0.8	1.8	47	1218
AC22	1.7	58.33	He,20	1.1	0.95	1.85	76	1292
AC23	1.7	50.0	He,20	1.3	0.6	2.0	338	845
AC24	1.7	41.33	He,20	1.3	0.7	2.0	244	821

Table A4.22 Resluts from welding 2mm mild steel with a CO laser using 1.7 kW.

Code	Power (kW)	Speed (mm/s)	Gas flow rate (l/min)	Top weld width (mm)	Penetration depth (mm)
C56b	0.63	5	Ar,10	1.0	0.9
C57b	0.63	8	Ar,10	0.8	0.6
C58b	0.63	10	Ar,10	0.8	0.55
C59b	0.63	15	Ar,10	0.5	0.4
C60b	0.63	20	Ar,10	0.65	0.35
C61b	0.63	5	He,10	1.2	0.6
C62b	0.63	8	He,10	0.9	0.45
C63b	0.63	10	He,10	0.8	0.45
C64b	0.63	15	He,10	0.8	0.4
C65b	0.63	20	He,10	0.65	0.35

Table A4.23 Results from welding 2mm mild steel with a CO laser at a power of 0.63kW.

Code	Power (kW)	Speed (mm/s)	Gas, Flow (l/min)	Top Weld Width (mm)	Stem Weld Width (mm)	Penetration depth (mm)	Point source strength (W)	Line Source Strength (W)
AC1	1.7	91.67	He,10	0.75	0.60	1.0	81	691
AC2	1.7	100.00	He,10	0.70	0.55	1.0	82	691
AC3	1.7	116.67	He,10	0.90	0.50	1.0	313	723
AC4	1.7	133.33	He,10	0.70	0.52	1.0	130	823
AC5	1.7	150.00	He,10	0.80	0.60	0.6	188	922
AC14	1.7	158.33	He,25	0.65	0.50	1.0	119	906
AC15	1.7	158.33	He,25	0.50	0.30	0.6	119	363
AC16	1.7	116.67	He,20	0.40	0.25	1.0	53	410
AC17	1.7	116.67	He,20	0.75	0.20	1.0	342	339
AC18	1.7	133.33	He,20	0.35	0.25	1.0	36	456
AC19	1.7	150.00	He,20	0.85	0.45	1.0	379	804
AC20	1.7	158.33	He,20	0.75	0.40	1.0	309	784
AC71	1.7	116.67	Ar,20	0.75	0.50	1.0	166	723
AC72	1.7	133.33	Ar,20	0.75	0.60	1.0	117	914
AC73	1.7	150.00	Ar,20	0.75	0.60	0.87	133	858
AC155	1.7	116.67	Ar,10	0.875	0.60	1.0	216	831
AC156	1.7	100.00	Ar,10	0.875	0.65	1.0	151	787
AC157	1.7	133.33	Ar,10	0.85	0.85	1.0	175	963

Table A4.24 Results from welding 1mm mild steel plate using a CO laser at a power of 1.7kW.

Code	Power (kW)	Speed (mm/s)	Gas flow rate (l/min)	Top weld width (mm)	Penetration depth (mm)
C1b	0.65	5	Ar,10	1.75	1.00
C2b	0.65	8	Ar,10	0.80	0.75
C3b	0.65	10	Ar,10	0.55	0.65
C4b	0.65	15	Ar,10	0.55	0.45
C5b	0.65	20	Ar,10	0.35	0.40

Table A4.25 Results from welding 1mm mild steel using a CO laser at a power of 0.65kW.