

A computational model of arc welding – from a research tool to a software product

Anthony B. Murphy^a and David G. Thomas^b

^a CSIRO Manufacturing, PO Box 218, Lindfield NSW 2070, Australia

^b CSIRO Data61, Box 312, Clayton South VIC 3169, Australia

Email: tony.murphy@csiro.au

Abstract: CSIRO has a long history of modelling arc plasma processes, including arc welding, plasma torches and plasma waste conversion. The models use the methods of computational fluid dynamics applied to viscous incompressible flows, extended to include an energy conservation equation, Maxwell's equations, and additional source terms to take into account plasma effects. The models have been used to gain a scientific understanding of arc plasma physics and chemistry and, as they have gained sophistication and accuracy, to improve and optimise industrial processes.

The computer codes that have been developed have, until recently, relied on expert users with familiarity with the detailed operation of the code, and access to a FORTRAN compiler. In a typical implementation, input parameters are provided using user-editable text files, progress towards convergence is monitored by viewing a continuously-updated text file, and the results of the calculations are written as text files and as data files readable by graphics programs. Input and output file handling is not automated; the user is required to rename files to avoid overwriting, and to ensure that all relevant files are stored together. The provision of an appropriate start-up file (for example, a solution obtained for a similar set of input parameters) has required the user to select the file; this relies on the user carefully documenting the input parameters for all solution files.

One of CSIRO's arc welding codes has now been packaged into a software product, ArcWeld, using CSIRO's Workspace workflow framework. This advance was motivated by a customer, General Motors, requesting that the code be easily usable by its welding engineers and technicians. The model is three-dimensional and treats the full arc welding process, with solid, liquid and plasma regions included in the computational domain. Despite these factors, physically-based simplifications are used to ensure that ArcWeld can be run under 64-bit Windows on standard desktop computers. A simple GUI is used for entry of input parameters, starting the computer code, displaying progress towards convergence, and access to graphical output. The most appropriate start-up file is automatically selected based on the input parameters. All input and output data are written to a user-selected directory.

This work has had important benefits. For the scientist, using a workflow platform means that common components and functionality are available as prewritten and pretested operations, so scientists can focus more on their core science and not as much on software development. For end users, the code can now be run by non-experts, who require only very basic training and access to a desktop computer. The input parameters are automatically stored with the output data, reducing the reliance on user documentation, and the user can easily visualise the results. These changes mean that the science can be advanced more quickly and the software can be used 'on the factory floor' to help optimise welding processes. More broadly, the software can easily be demonstrated to potential customers, and has greatly increased the attractiveness of our arc plasma modelling work to potential customers.

Keywords: *Arc welding, graphical user interface, CFD modelling, workflow, Workspace*

1. INTRODUCTION

Thermal plasma processes, including arc welding, plasma cutting, plasma spraying and plasma waste treatment, are used very widely in industry. The processes make use of the high temperatures (up to 25 000 K) and heat fluxes that are provided by the plasma, which is usually produced by an electric arc between two electrodes.

Thermal plasmas are typically at or near atmospheric pressure, and can be modelled using the methods of computational fluid dynamics, with some important adaptations, such as inclusion of Maxwell's equations to treat electromagnetic phenomena. CSIRO has developed such models over more than 30 years. During this time, there have been several significant advances. In particular:

1. Initially the models considered only the arc plasma, but now they take into account the electrodes and other solid and liquid materials that interact with the plasma;
2. The models were initially two-dimensional and steady-state, but are now three-dimensional and time-dependent;
3. The original aim of the modelling effort was to provide scientific insights into experimental results, but they have more recently been applied to improve and optimise industrial processes.

This has greatly increased the value of the models, and their attractiveness to industry. However, until recently, the models have largely been research tools – they have only been useable by highly-trained expert users. This has limited their usefulness to industrial and other collaborators, since unless the collaborator was willing to spend several weeks understanding the intricacies of running the computer code, the code could only be run by CSIRO researchers. Further, even within CSIRO and within the plasma modelling team, there were difficulties transferring models, keeping records of results, maintaining adequate version control, and so on.

Perhaps the most sophisticated thermal plasma model that CSIRO has developed is of metal–inert-gas (MIG) welding. MIG welding is the arc welding process that is most widely used in manufacturing industry. The development work was partly funded by General Motors, General Motors Holden and the AutoCRC, with a focus on welding of aluminium sheets for automotive applications. General Motors requested that the computer code be packaged so that it was easily useable by its welding engineers and technicians. As a consequence, the code was packaged using CSIRO's Workspace workflow software. This allows the code to be run under Windows on a standard desktop or laptop computer, with a simple GUI for entry of input parameters, starting the computer code, displaying progress towards convergence, and access to graphical output.

In this paper, we will briefly present the computational model. We will then outline the changes to the workflow that were implemented when the code was packaged with a GUI using the Workspace software. Finally, we will discuss the benefits of these changes to researchers and to industrial collaborators.

2. THE COMPUTATIONAL MODEL OF MIG WELDING

In MIG welding (also known as gas metal arc welding or GMAW), an arc plasma is struck in a shielding gas between a wire electrode and the metal parts that are being joined, which are known as the workpiece. A schematic of the process is shown in Figure 1. In MIG welding of aluminium, the shielding gas is typically argon; for welding of steel, oxygen or carbon dioxide is usually added to the argon. The arc current is typically in the range from 80 to 400 A, depending on the thickness and type of the metal being welded, the speed of welding, and other factors. The energy transferred by the arc partially melts the workpiece, forming the weld pool. The wire electrode melts, forming droplets that pass through the arc into the weld pool; accordingly the wire is continuously fed (Norrish, 1992).

The arc plasma and the liquid metal are modelled using the methods of computational fluid dynamics applicable to viscous incompressible flows, extended to include an energy conservation equation, Maxwell's equations, and additional source terms to take into account plasma effects (Murphy, 2011). To take into account the production of metal vapour from the liquid metal surfaces, and the transport of the vapour in the arc, an additional conservation equation for the metal vapour mass fraction has to be solved (Murphy, 2013b). If the mixing of the wire and workpiece alloys in the weld pool is modelled, a further conservation equation for the wire alloy mass fraction is required (Murphy et al., 2017). The energy

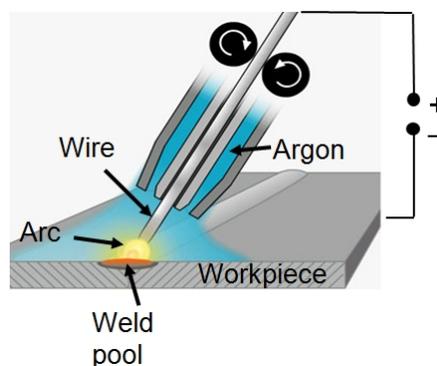


Figure 1. Schematic of the MIG welding process.

conservation and electromagnetic equations are also solved in the solid regions of the workpiece and the electrode. Internal boundary conditions are used to treat the energy and momentum transfers at the arc–weld pool boundary.

In almost all arc welding, the wire electrode and arc are moved along the weld seam. This means that the equations have to be solved in three dimensions. Also, it requires that the equations be transformed into the frame of reference of the moving wire and arc, maintaining the wire in the central region of the computational domain. Further, the workpiece is often not two adjacent flat plates (bead-on-plate geometry), but rather one plate partly overlapping the other (lap-fillet geometry); this again requires three-dimensional geometry. Steady-state solution of the equations is often acceptable; only if the welding parameters change, or near the start or end of the weld, is time-dependent solution required. The equations used have been presented by Murphy et al. (2017). They are solved using an in-house FORTRAN code based on the finite-volume method presented by Patankar (1980). The transfer of momentum and energy from the arc to the droplets, and from the droplets to the weld pool, is treated approximately by averaging their influence over time and the cross-sectional area of the droplets (Murphy, 2013a). The shape of the free surface between the weld pool and arc is calculated by minimising the energy of the surface, taking into account surface tension, buoyancy, and the pressure applied by the arc and the droplets (Murphy, 2011).

3. INTEGRATION OF THE ARC WELDING MODEL WITH A WORKFLOW PLATFORM

The arc welding code was integrated with a GUI using the Workspace workflow platform (Bolger et al., 2014; Cleary et al., 2014; Bolger et al., 2015; Cleary et al., 2015). We describe in this section the typical operation of the code before this integration, and following the integration, and then discuss the benefits of the integration.

3.1. Before integration

The model, as initially developed, was run on a desktop computer under either Unix or Windows. The source code was compiled using an Intel Fortran compiler. Figure 2 (left-hand side) shows a flow diagram of a typical operation of the code.

Before the code can be run, three text files defining the input parameters have to be edited. The files contain respectively:

1. The welding parameters (such as arc current, welding speed, wire electrode feed rate and droplet frequency), computational control factors (e.g. the maximum number of iterations, the convergence criteria and relaxation factors) and options (e.g. type of workpiece geometry, start from scratch by using the code to calculate an approximate solution or use a previous solution file, include or exclude mixing of wire and workpiece alloys);
2. The geometric and meshing parameters (the wire electrode diameter and orientation, the arc length, the workpiece thickness, the dimensions of the computational domain, the number of control volumes in different regions, etc.); separate files are used for bead-on-plate and lap-fillet geometries;
3. The type or composition of metal alloys used for wire electrode and workpiece.

Solution files are text files listing the temperature, velocity, current density, electric potential, magnetic potential, metal vapour concentration, wire alloy concentration and metal fraction in every control volume. If a previous solution file is used to start the run, it has to be selected from a library of such files, containing previous fully- and partially-converged solutions. The solution file has to be copied to the same folder as the input parameter text file, and renamed so that the code will recognise it. Alternatively, the code can be started ‘from scratch’, in which case an approximate initial solution is calculated.

The code is run from a command prompt, and text showing progress towards convergence is written to the screen. Several output files are written following each iteration, including text files showing progress towards convergence, code diagnostic information and binary Tecplot graphics files. A solution file is written after the final iteration. At each manual stage of this process there is ample scope for user error, and similarly there is little scope for reproducible testing of the set up process. A non-technical user can easily get into trouble with such a process.

3.2. After integration

The GUI that was developed to run the arc welding code controls all aspects of code operation. A flow diagram showing operation of the code using the GUI is shown in Figure 2 (right-hand side) The GUI contains five input screens, respectively headed:

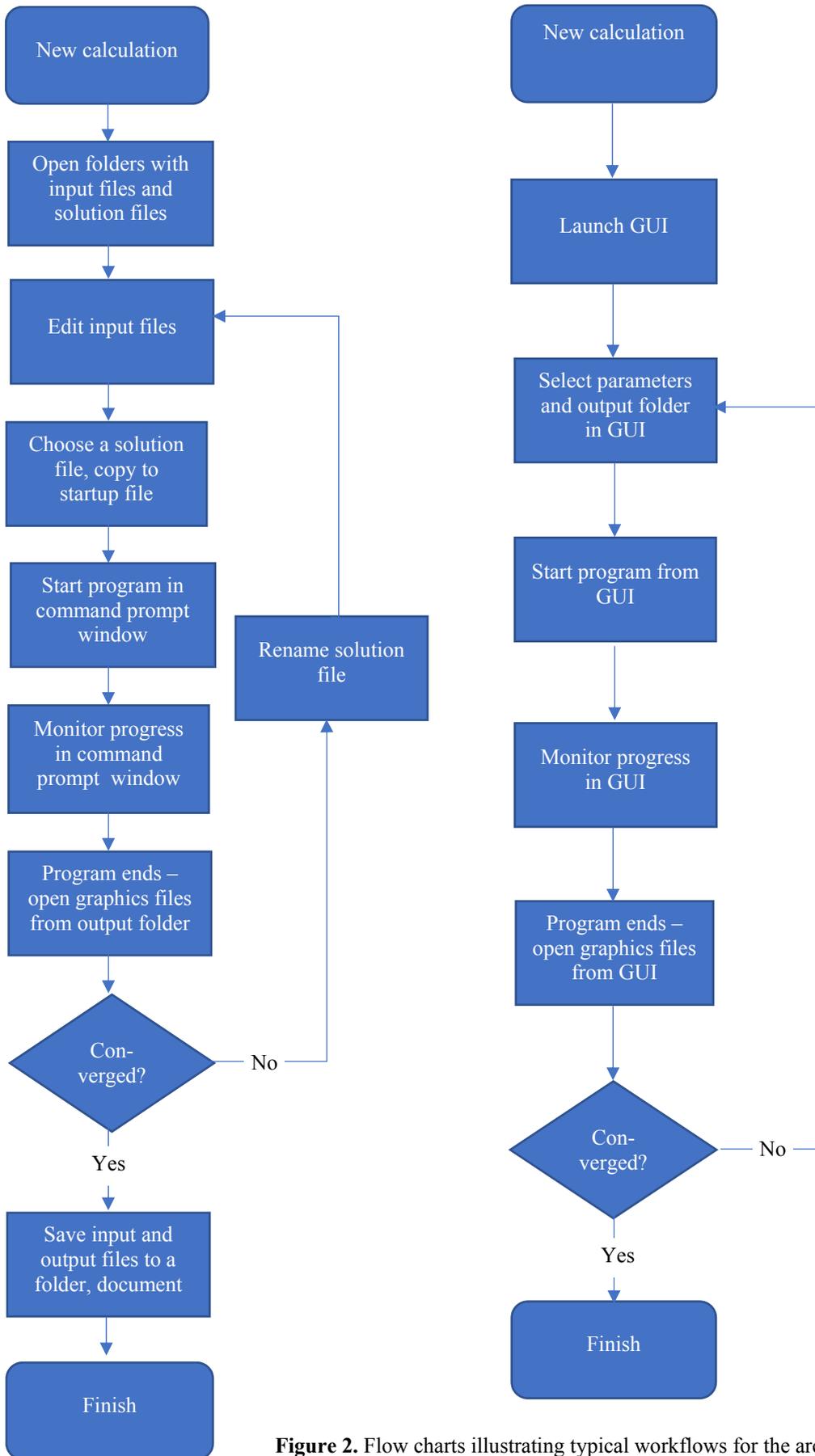


Figure 2. Flow charts illustrating typical workflows for the arc welding code before (left) and after (right) integration with the GUI using Workspace.

1. Project: project name and storage location;
2. Solver: code control options, including relaxation factors, maximum number of iterations, and the choice of using of a solution file selected by the program based on the welding and geometrical parameters, the solution file from the previous run, or starting from scratch;
3. Welding parameters: arc current, shielding gas flow, wire feed rate, droplet frequency, welding speed, workpiece geometry;
4. Geometry: workpiece thickness, arc length, wire electrode diameter, orientation of wire electrode;
5. Alloys: choice of alloy used for the wire electrode and the workpiece – either from a drop-down list of alloys, or a user-defined composition.

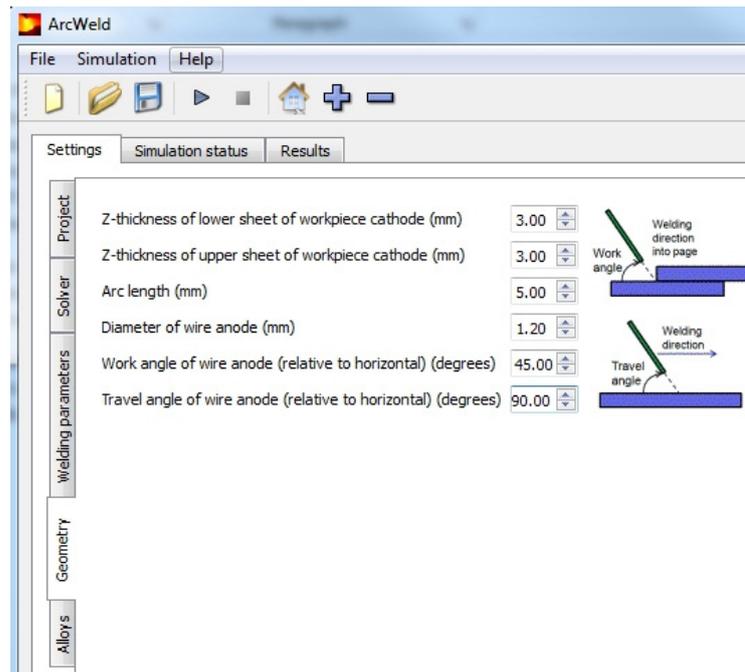


Figure 3. Screenshot of the GUI, showing the Geometry input screen.

When the code is started, the status of the simulation (number of iterations, progress towards convergence) is shown in a colour-coded screen. At any time, the results can be viewed using hyperlinks to Tecplot graphics files; these open Tecplot, and stored macros can be used to create default plots. The default plots for typical welding conditions are shown in Figure 4 to Figure 6.

A folder determined by the project name and location is created on starting the code. A file containing all parameters input to the GUI, and the input files (the text files and solution file read by the code) are written to this folder. The code execution is not changed by the GUI. All output files are written to the folder.

3.3. Benefits

The previous implementation of the code had several drawbacks.

- The user had to modify the input files manually; this required detailed knowledge of the form of these files. Different versions of the code often had different input file formats, so it was not always possible to re-use old input files without editing. There were no restrictions on the values of the parameters that could be entered; bad or incompatible values were not identified until the code was run, adding an unnecessary administrative overhead to the solver code.
- The input files and output files were saved to a default folder. The user had to document the input parameters and results, and manage storage of the input files and output manually. This situation opens itself to problems with user–file interactions (e.g. looking in the wrong location without checking timestamps). Using the Workspace platform-generated GUI application to handle the management of files obviates the need for such user interaction and care.
- The choice of the solution file used as the starting parameters was left to the user’s judgement. The user had to choose the appropriate file and manually move it to the input file folder and rename it.
- The screen display of progress towards convergence required detailed understanding to interpret.
- Output files including graphics files had to be opened from the default folder, again requiring user knowledge and maintenance of file locations.

One of the consequences of these drawbacks was strongly uneven documentation and storage practices, which depended on the user’s work practices. There was strong variation from user to user, and a given user’s practices often changed depending on the work pressure he or she was under. Other consequences included the significant scope for user error in both the editing of the files and the management of the file locations, and the

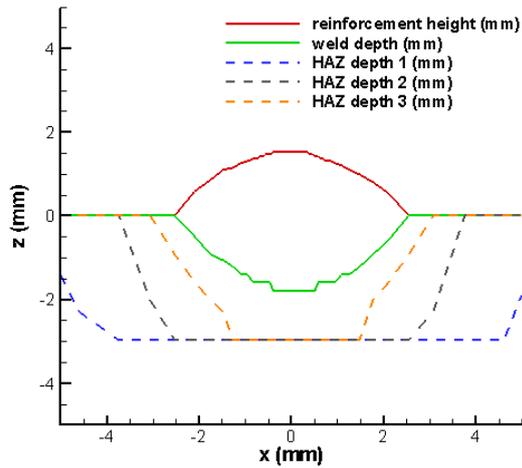


Figure 4. Weld cross-section and extent of heat-affected zones for 300°C, 400°C and 500°C for bead-on-plate welding of a 3-mm-thick sheet of aluminium AA 5754, for 95 A welding current, 0.90 m/min welding speed, 1.2-mm-diameter aluminium AA4043 wire fed at 4.32 m/min.

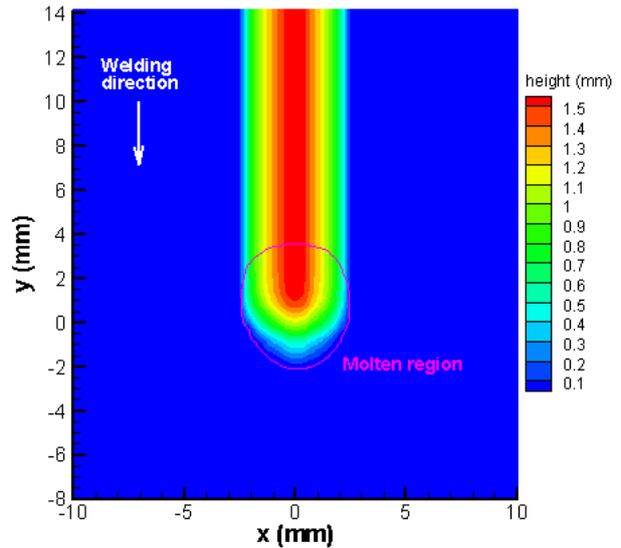


Figure 5. Height of weld reinforcement and boundary of molten region for conditions of Figure 4.

difficulty of testing of and reproducibility of the whole process of setting up, running and viewing and storing the results of a simulation.

Following integration of the code with the GUI, the operation of the code was automated. Based on the parameters entered into the GUI by the user, the most appropriate solution file is selected from a set of such files; alternatively the user can choose to use the solution file from the previous run, or to have the code calculate the initial solution. This means that the user does not have to copy or rename previous solution files, eliminating a common source of error. All parameters and files associated with a run are stored in a single folder; the only documentation required is to associate any information with the name of the folder.

Other advantages are that default values for all parameters are preselected, that the GUI enforces that the selected parameters fall within the range under which the code has been tested, and that the GUI ensures validity of all other parameters (including those that may interact with each other). Reliability is therefore increased. Monitoring of the progress of the code is much more intuitive, and viewing of results is simplified.

Even for an expert user, the Workspace-integrated code speeds operation, reduces errors, and allows the user to concentrate on generating research results rather than details of the code, storage locations and

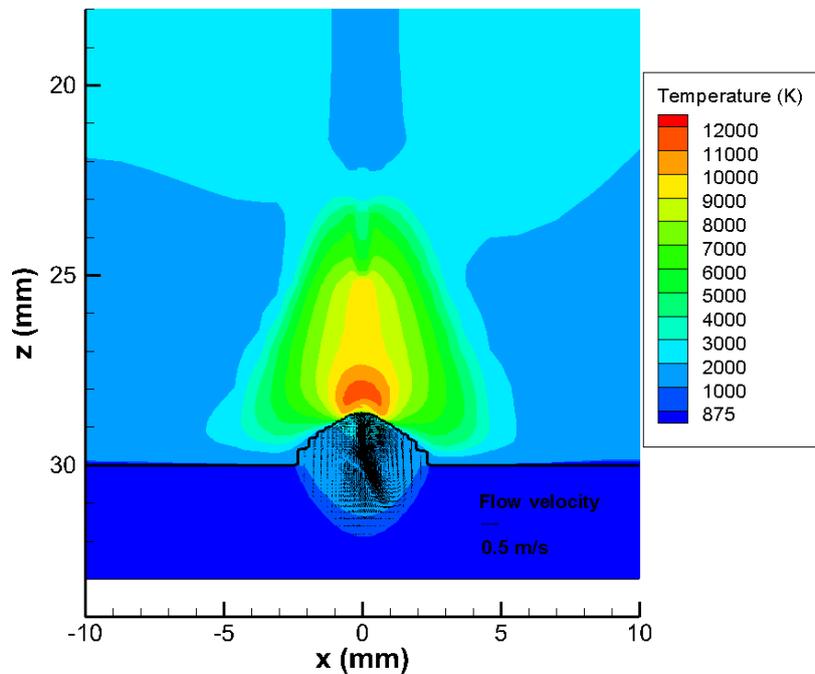


Figure 6. Temperature distribution in arc, wire electrode and workpiece, showing flow velocity vectors in weld pool; for the conditions of Figure 4. The edge of the weld pool corresponds to the contour at 875 K (the liquidus temperature of AA 5754).

documentation. After a period in which the code has not been used, it is possible to start running the code immediately, without spending time reviewing documentation and input file formats and locations. It is therefore possible to make use of even small chunks of available time to run the code. The code is also easily transferrable between researchers. A notional penalty might be a perceived decrease in flexibility when the user wishes to increase the range or number of input parameters (for example, by adding additional workpiece alloys to the list of options). However, the application is easily updated using existing Workspace tools; this requires a little more work than simply editing the FORTRAN code, but ensures that the change is documented and clearly visible to the user. Our experience is that the advantages greatly outweigh any drawbacks, and the overall consequence is that the science can be advanced more quickly, and with superior documentation.

A major advantage is that the code can now be run by non-experts, who require only brief training, a basic competence with Windows, and access to a desktop computer. Welding engineers almost without exception do not have the skills to run a ‘research’ code, but typically use desktop computers routinely. The software can thus be used routinely ‘on the factory floor’ to help optimise welding processes. More broadly, the software can easily be demonstrated to potential customers, and has greatly increased the attractiveness of our arc plasma modelling work to potential customers.

4. CONCLUSIONS

The integration of the computer code developed to model arc welding with a GUI using the Workspace workflow platform has been highly advantageous. It has simplified the workflow required to run the code, improved documentation and storage of results, reduced errors and saved researchers’ time. Moreover, it has greatly increased accessibility of the code, allowing it to be used by welding engineers in a commercial industrial environment.

ACKNOWLEDGEMENTS

The support provided for the work reported here by General Motors, General Motors Holden, and the Commonwealth of Australia, through the AutoCRC, is gratefully acknowledged. Drs Damien Watkins, John Lowke and Eugene Tam of CSIRO and Dr Hui-Ping Wang of General Motors are thanked for useful discussions.

REFERENCES

- Bolger, M., Cleary, P., Hetherington, L., Rucinski, C., Thomas, D. and Watkins, D. (2014). Workspace: Scientific Workflow Platform. Version 2. CSIRO. doi:10.4225/08/54D03170101B7.
- Bolger, M., Cleary, P., Hetherington, L., Rucinski, C., Thomas, D. and Watkins, D. (2015). Workspace: Scientific workflows and applications for multiple environments. Paper presented at the eResearch Australasia Conference, Brisbane, 19-23 October 2015.
- Cleary, P., Bolger, M., Hetherington, L., Rucinski, C., Thomas, D. and Watkins, D. (2014). Workspace: A platform for delivering scientific applications. Paper presented at the eResearch Australasia, Melbourne, 27-31 October 2014.
- Cleary, P.W., Thomas, D., Bolger, M., Hetherington, L., Rucinski, C. and Watkins, D. (2015). Using Workspace to automate workflow processes for modelling and simulation in engineering. MODSIM2015, 21st International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, Gold Coast.
- Murphy, A.B. (2011). A self-consistent three-dimensional model of the arc, electrode and weld pool in gas-metal arc welding. *Journal of Physics D: Applied Physics* 44(19), 194009.
- Murphy, A.B. (2013a). Influence of droplets in gas-metal arc welding – a new modelling approach, and application to welding of aluminium. *Science and Technology of Welding and Joining* 18(1), 32-37.
- Murphy, A.B. (2013b). Influence of metal vapour on arc temperatures in gas-metal arc welding: convection versus radiation. *Journal of Physics D: Applied Physics* 46(22), 224004.
- Murphy, A.B., Nguyen, V., Feng, Y., Thomas, D.G. and Gunasegaram, D. (2017). A desktop computer model of the arc, weld pool and workpiece in metal inert gas welding. *Applied Mathematical Modelling* 44(1), 91-106.
- Norrish, J. (1992). *Advanced Welding Processes*. Institute of Physics Publishing, Bristol, UK.
- Patankar, S.V. (1980). *Numerical Heat Transfer and Fluid Flow*. Hemisphere, Washington DC.