LRP 748/03 February 2003

Gradient evaluation on non-orthogonal meshes for the application of a plasma torch

L. Klinger, J.B. Vos, K. Appert

Submitted for Publication in
Computers and Fluids

ISSN 0458-5895

Gradient evaluation on non-orthogonal meshes for the application of a plasma torch

L. Klinger [a,*] J. B. Vos [b] K. Appert [c]

[a] *RTSE, Ancien Sierre 9, CH-3960 Sierre, Switzerland*

[b] *CFSE Engineering, PSE-B, CH-1015 Lausanne, Switzerland*

[c] *CRPP, EPFL, CH-1015 Lausanne, Switzerland*

Abstract

A numerical method is proposed to evaluate gradients on non-uniform, non-orthogonal 3D structured meshes of hexahedra, as commonly used by finite volume methods. The method uses isoparametric transforms on tetrahedra to evaluate the gradient on a regular mesh and transform it back to the general mesh. It provides second-order accuracy, even on highly non-orthogonal meshes. Results of stationary 3D numerical simulations of a DC plasma torch, making use of the proposed method, are presented.

Key words: finite volume method, isoparametric transforms, gradient evaluation, plasma torches

PACS: 02.70.Bf, 47.11.+j, 52.75.Hn

* Corresponding author. Fax.:+41 27 451 7329

Email address: laurent.klinger@alcan.com (L. Klinger).

1 Introduction

Finite volume methods (FVM) are widely used for solving conservation equations, such as the Navier-Stokes equations in CFD. Using FVM, one typically integrates in space the partial differential equation in conservative form, so that the divergence term is transformed into the surface integral of the field's flux on the cell sides. How the fluxes are calculated is unspecified by the FVM, as long as they satisfy two principles: conservation and consistency [1]. The former requires that the flux from cell C_i to neighbour cell C_j through their common side is opposite to the flux from C_j to C_i. The latter, roughly speaking, requires that the numerical error on the flux evaluation is $O(h)$, if h is the size of the largest cell in the mesh. While evaluation of the diffusive fluxes requires the gradients on the *cell sides*, any source terms, if themselves expressed as function of some gradients, may require a *cell-centered* evaluation of the gradients.

In the context of a plasma torch [2] simulation, we used an existing CFD code [3], to which an electromagnetic part was added. The code uses the Peyret-Taylor integral formula [4] to compute side-centered gradients and a finite difference formula for cell-centered gradients. For non-orthogonal meshes, it quickly appeared that a better gradient evaluation was needed to solve for the electrical potential and evaluate Ohmic heating, which appears as a source term in the energy equation and is proportional to the squared gradient of the electric potential. The difficulty with these quantities originates from strong non-linearities: the electric conductivity, which is a coefficient in a Poisson-like equation determining the electric potential, depends strongly (sometimes exponentially) on the temperature.

Several studies deal with this problem from scratch, see for example Hyman and Shashkov [5,6] (and references therein) or Hermeline [7]. To use the results of these studies for the gradient evaluation would have resulted in a major code rewrite, which we considered inacceptable, since we only sought to improve the method to compute the gradients. We later found out that another method (called the *path integral method*) was developed by Wesseling *et al.* [8]. This method considers the integral of the gradient on a straight line. On the one hand, this integral is equal to the difference of the field values at the path extremities (chosen to be cell centres around the point where the gradient is evaluated), and on the other hand, the integral can be approximated by the dot product of the gradient with the vector joining the path extremities. Thus, an equation for the component of the unknown gradient along the path is obtained, and by considering other paths, one builds a linear system for the complete gradient.

In the following, a numerical method based on isoparametric transforms to calculate cell-centered and surface-centered gradients on general structured meshes is presented. Results using this method are compared with those obtained with the Peyret-Taylor formula, and with the method developed by Wesseling *et al.*. Usage and benefits of the improved gradient evaluation method for the simulation of a 3D plasma torch are shown, and implementation issues are discussed.

2 Numerical method

A general (non-orthogonal, non-equidistant) 3D structured mesh of hexahedra is considered. In this case, the cells can be logically indexed by three indices

(i,j,k). Let $\vec{x}^{\mathrm{c}}_{i,j,k}$ be the centre of the cell (i,j,k). The discrete values $\phi_{i,j,k}$ are the numerical approximations, at the cell centres, of a continuous field Φ. Our objective is to find an expression for $\nabla\phi(\vec{x}^{\mathrm{c}}_{i,j,k})$ (cell-centered gradient) and $\nabla\phi(\vec{x}^{\mathrm{s}}_{i,j,k})$ (gradient at one of the cell surface centres).

It is well known [9] that using $\nabla\phi(\vec{x}^{\mathrm{c}}_{i,j,k})$ to compute $\nabla\phi(\vec{x}^{\mathrm{s}}_{i,j,k})$ (or vice-versa) employing some kind of interpolation may lead to numerical instabilities. It is thus necessary to perform the two calculations independently, and we will show how isoparametric transforms can be used for both calculations.

2.1 General statement of the problem

Whether cell-centered or surface-centered, the problem can be stated in the same way: given a 'central' point $\vec{x}_0$, six points $\vec{x}_1,\ldots,\vec{x}_6$ around $\vec{x}_0$, and the values ϕ_i, $i=0,\ldots,6$ of the discrete field ϕ at those points, find $\nabla\phi(\vec{x}_0)$. Cell-centered and surface-centered gradients differ in the choice of the $\vec{x}_i$'s.

In the case of a cell-centered gradient evaluation, the points considered are simply $\vec{x}^{\mathrm{c}}_{i,j,k}$ and its six nearest neighbour cell centres $(\vec{x}^{\mathrm{c}}_{i\pm1,j,k}, \vec{x}^{\mathrm{c}}_{i,j\pm1,k}, \vec{x}^{\mathrm{c}}_{i,j,k\pm1})$. The field values are directly known at those points. In the case of a surface-centered gradient, the 'central' point is a surface centre, while the other six points are the corners of that surface and the two centres of the cells sharing that surface. Nevertheless, one must first interpolate ϕ at the five points located on the surface, since ϕ is known only at cell centres. Figure 1 shows a cell and one of its neighbours, along with some of the points used for the gradient calculations.

Given this set of seven points and field values, we show how to use isopara-

metric transforms to estimate the gradient at the 'central' point.

2.2 Isoparametric transforms

Generally speaking, the isoparametric transform, $\vec{r}(u,v,w)$, that maps a (regular) polyhedron in (u,v,w) coordinates to its (irregular) counterpart in (x,y,z) coordinates may be written:

$$\vec{r}(u,v,w) = \sum_{\alpha=1}^{n} N_\alpha(u,v,w)\,\vec{r}_\alpha^{\,v} \tag{1}$$

where n is the number of vertices of the polyhedron and the $\{\vec{r}_\alpha^{\,v}\}$ its vertices in (x,y,z) coordinates [10,11]. The functions N_α are to be determined using some interpolation ensuring that the vertices of the regular polyhedron are transformed to $\{\vec{r}_\alpha^{\,v}\}$. As in the finite element method, (1) is also used to interpolate ϕ:

$$\phi(u,v,w) = \sum_{\alpha=1}^{n} N_\alpha(u,v,w)\,\phi(\vec{r}_\alpha^{\,v}). \tag{2}$$

Then, using (2), one relates $\nabla\phi$ in (u,v,w) coordinates and (x,y,z) coordinates through the transposed Jacobian matrix J. We have in fact:

$$\nabla\phi\Big|_{x,y,z} = J^{-1}\,\nabla\phi\Big|_{u,v,w}. \tag{3}$$

2.3 Gradient evaluation

While one cannot write an isoparametric transform mapping the octahedron defined by the seven points considered, it is possible to partition these points in two sets defining two tetrahedra. The procedure to evaluate $\nabla\phi(\vec{x}_0)$, given $(\vec{x}_0,\ldots,\vec{x}_6;\phi_0,\ldots,\phi_6)$, is then as follows. We consider a first tetrahedron defined by four points (including $\vec{x}_0$) from the seven given points, and the isopara-

metric transform it defines. Using (3), we obtain a first estimation of the gradient at $\vec{x}_0$, say $\nabla\phi^{(1)}$. We proceed similarly for a second tetrahedron, defined by $\vec{x}_0$ and the remaining three points not used for the first tetrahedron, yielding a second gradient estimation $\nabla\phi^{(2)}$. Finally, we average these two estimates:

$$\nabla\phi(\vec{x}_0) = \tfrac{1}{2}\big(\nabla\phi^{(1)} + \nabla\phi^{(2)}\big). \tag{4}$$

To define the first tetrahedron, in the case of a cell-centered gradient, one would choose the lower-indices neighbours, while for the surface-centered gradient, one would choose two consecutive side corners and one of the two cell centres.

Let us show in detail how the computation is performed for the first tetrahedron, which has vertices $\vec{r}_1^{\,v}, \ldots, \vec{r}_4^{\,v}$ (chosen from $(\vec{x}_0, \ldots, \vec{x}_6)$ as explained above), following the notation of (1). For the two tetrahedra, we take $\vec{r}_1^{\,v} = \vec{x}_0$. We assume that the isoparametric transform is of the form

$$\vec{x}(u,v,w) = \vec{a}_1 + \vec{a}_2 u + \vec{a}_3 v + \vec{a}_4 w \tag{5}$$

and is determined by the conditions:

$$\vec{x}(0,0,0) = \vec{r}_1^{\,v} \qquad \vec{x}(0,1,0) = \vec{r}_3^{\,v}$$
$$\vec{x}(1,0,0) = \vec{r}_2^{\,v} \qquad \vec{x}(0,0,1) = \vec{r}_4^{\,v}$$

meaning that, for example, the vertex $(u = 0, v = 0, w = 0)$ of the regular tetrahedron is mapped to $\vec{r}_1^{\,v}$. This yields:

$$\vec{a}_1 = \vec{r}_1^{\,v}, \qquad \vec{a}_2 = \vec{r}_2^{\,v} - \vec{r}_1^{\,v},$$
$$\vec{a}_3 = \vec{r}_3^{\,v} - \vec{r}_1^{\,v}, \qquad \vec{a}_4 = \vec{r}_4^{\,v} - \vec{r}_1^{\,v}.$$

which allows the determination of the N_α functions of (1):

$$N_1(u,v,w) = 1 - u - v - w \qquad N_3(u,v,w) = v$$
$$N_2(u,v,w) = u \qquad N_4(u,v,w) = w.$$

The Jacobian matrix of the isoparametric transform thus defined is given by

$$J = \left[\vec{a}_2 \; ; \; \vec{a}_3 \; ; \; \vec{a}_4\right]^T. \tag{6}$$

We now express the gradient of ϕ in the (u,v,w) coordinates, using (2) and its derivatives. Writing $\phi_i = \phi(\vec{r}_i^{\,\nu})$, $i \in \{1,\ldots,4\}$, we obtain:

$$\frac{\partial\phi}{\partial u} = \phi_2 - \phi_1, \qquad \frac{\partial\phi}{\partial v} = \phi_3 - \phi_1, \qquad \frac{\partial\phi}{\partial w} = \phi_4 - \phi_1. \tag{7}$$

Finally, recalling (3) and using (7), we have

$$\nabla\phi^{(1)}\Big|_{x,y,z} = J^{-1}\begin{bmatrix}\phi_2 - \phi_1\\ \phi_3 - \phi_1\\ \phi_4 - \phi_1\end{bmatrix}. \tag{8}$$

One proceeds similarly to compute $\nabla\phi^{(2)}$, the only difference being in the values of $\vec{r}_2^{\,\nu}, \vec{r}_3^{\,\nu}$ and $\vec{r}_4^{\,\nu}$, which are now the three points from $(\vec{x}_1,\ldots,\vec{x}_6)$ not used yet.

2.4 Test cases

The quality of the numerical gradient evaluation has been assessed by testing it on various given fields ϕ with analytically known gradient. The quality measures used are the mean absolute and relative errors for a cell-centered

gradient evaluation using the Peyret-Taylor method (PT) and the proposed isoparametric transform method (ITM), respectively.

For example, we considered a single block domain with a 'L' shaped mesh, for the field $\phi(x,y,z) = \exp\Big(-(x^2+y^2+z^2)\Big)$. This mesh is irregular, and strongly non-orthogonal at the L crease. To increase the difficulty, the mesh was 'shaken', the term designating a random displacement of the interior mesh nodes, see also Fig. 4.

Figure 2 shows the average relative and absolute errors of the gradient evaluation for a 3D L-shaped domain of unit extension in each direction. Figure 3 shows the same results obtained on the 'shaken' mesh. In each cell, the absolute and relative errors are defined by

$$e_{\text{rel}} = \left| 1 - \frac{(\nabla\phi)^2}{(\nabla\phi)^2_{\text{analytic}}} \right|, \tag{9}$$

$$e_{\text{abs}} = \left| (\nabla\phi)^2 - (\nabla\phi)^2_{\text{analytic}} \right|. \tag{10}$$

The Peyret-Taylor method shows linear convergence on the regular mesh, and fails to converge on the shaken mesh, whereas ITM exhibits quadratic convergence in both cases. Figures 4 shows a cut (at $z = 0.5$) of $(\nabla\phi)^2$ on the 'shaken' L mesh.

2.5 An electric arc test case

In this section, we present a numerical simulation of a plasma torch. A plasma torch is a device in which an electric arc is established between two co-axial electrodes. The main feature of the case considered here is the fact that the arc column is subject to a cross-flow, that can only be studied by 3D computations.

Due to the cross-flow and intense Ohmic heating, the gas at the torch exit can be used for thermal spraying by introducing a powder into the flow, where it melts before coating a surface facing the exit (see figure 5).

Under the assumption of local thermodynamic equilibrium, a hydrodynamic model is used to describe the flow inside the torch, and we solve the Navier-Stokes equations together with a Poisson-like equation for the electric potential, $\nabla \cdot (\sigma \nabla \phi) = 0$, where σ and ϕ stand for the electric conductivity and potential, respectively. Fluid and electric parts are coupled through Ohmic heating (energy source term of the form $\sigma(\nabla\phi)^2$), Lorentz force (momentum source term) and temperature (electric conductivity strongly depends on temperature) [12,13].

A current density profile, $\sigma\nabla\phi$, is given on the boundaries corresponding to the electrodes, while the electric potential is set to zero on the remaining boundaries. For the flow, subsonic inlet and outlet boundary conditions are specified, along with no-slip conditions on the walls. Temperature profiles are imposed on the solid walls.

Using the original Peyret-Taylor formula to evaluate gradients, we quickly ran into severe problems appearing first in the electric potential in regions with strong mesh non-orthogonality. The poor evaluation of the cell-centered gradient of the electric potential at these locations typically resulted in an overestimated Ohmic heating. This in turn had dramatic consequences on the Navier-Stokes part due to the strong non-linear dependence of the electric conductivity on the temperature. Once the evaluation of the gradient of the electric potential was improved, unrealistic temperatures were observed at the same locations due to an inadequate evaluation of the heat flux. All these fail-

ures were related to non-orthogonal meshes and the elliptic parts of the system (for which fluxes depend on gradients). It should be stressed that this inadequacy of the method cannot be overcome by just performing more iteration steps in the hope to bring the system to convergence; a poorly approximated Ohmic heating term renders any such attempt futile.

We present here a 600 A plasma torch simulation, with 30 standard litres per minutes Argon at inflow at atmospheric pressure. Figure 6 shows the squared electric current density near the cathode, after 200 iterations, using the Peyret-Taylor formula. The obviously wrong current density obtained is typical of the above-mentioned problems with gradient evaluation.

Using the proposed ITM method, we were able to converge the calculation, and to obtain a numerical solution which makes sense. Figure 8 displays the computed temperature field. For this current, one obtains temperature levels and velocities (around 400 m/s) at the torch exit that are consistent with measurements [14]. The high temperatures near the cathode tip, even though in agreement with experiments are somewhat questionable due to the the simplified physical model employed for the cathode tip region. In particular, radiation transport and non-equilibrium sheaths [15,16] are ignored.

The decribed numerical simulation was obtained on a mesh of 14 blocks containing in total half a million grid points. Using 14 processors on an Origin 3000 computer, the simulation took approximately 2 days of elapsed time. More details about the modeling and the peculiarities of such simulations can be found in [17].

3 Discussion

As illustrated in Section 2.4, on non-orthogonal meshes, the isoparametric transform method yields second-order accuracy, while the Peyret-Taylor method leads to first order only. On the corresponding shaken meshes, the isoparametric transform method retains almost second-order accuracy.

For a regular, orthogonal mesh, the isoparametric transform procedure reduces to a standard central difference formula. In fact, the two methods share the same approach: first, estimate the gradient in each direction around the central point (this is where the geometry, whether simply distances in 1D or a Jacobian matrix in 3D, is taken into account), and then average these estimates to remove first order errors. Although there are multiple ways to build two tetrahedra from the set of seven points considered (section 2.1), numerical experiments showed that no significant difference occurred from one choice or the other, as long as lower/higher indices points (e. g. $\vec{x}^{\mathrm{c}}_{i-1,j,k}$ and $\vec{x}^{\mathrm{c}}_{i+1,j,k}$) are not in the same tetrahedron.

To assess the efficiency of the numerical method, it is important to compare execution time and memory requirements. Since the isoparametric transform method uses two (inverted) Jacobians per gradient evaluation, it is desirable to precompute and store J^{-1} for each transform. This allows the determination of the gradients at the same computational costs as the standard central difference method, but at the price of increased memory requirements (18 reals per gradient evaluation per grid point).

Comparing the ITM method with the path integral method of Wesseling *et al.* [8] shows that the methods are equivalent in terms of numerical precision.

But the path integral method is computationally more efficient, since it only requires the evaluation of one 3×3 matrix per gradient calculation, instead of two 3×3 matrices for the ITM method.

The use of the path integral method becomes delicate at some very specific cells, as for example corner cells in multi block meshes, see Fig. 7. In these cells, one must be careful to use suitable values in the ghost cells, otherwise the conservativity principle will not be satisfied. In the example of Fig. 7 (a typical configuration of the mesh in the plasma torch), this may be problematic for the corner ghost cell, which is degenerate. For these cases, it may be easier to forget simply the contributions associated with these points in the path integral method, so that the unknown points are not used at all, at the expense of a slightly less precise formula.

The ITM method offers for these cases the elegant solution of using wedge-shaped or tetrahedral isoparametric elements for the interpolation. The ITM method thus never makes use of unknown values (or values generated to estimate unknown values). In the 2D schema of figure 7, this is represented by using the gray triangle on the right, instead of a quadrilateral.

To conclude, some remarks based on our experience are in order. For 3D configurations, the use of the path integral method to compute all gradients requires the storage of 36 reals per cell (3 side-centered gradients and 1 cell-centered). However, the use of this method is only required for some cells located at the most non-orthogonal parts of the mesh, often located near block interfaces where the mesh cannot always be made orthogonal. It is then computationally advantageous, and relatively simple to implement, to use the Peyret-Taylor formula for most cells of a block, and to use the path integral or

ITM method only for the outermost cells, each time recomputing the matrices used. This brings the benefits of the improved gradient evaluation where it is needed, at no memory costs and at marginally higher execution times.

4 Conclusion

We proposed a method using two isoparametric transforms on tetrahedra to evaluate the gradient of a scalar field, at the cell centre or centre of cell side, of a general (non-orthogonal) 3D structured mesh. The method provides second-order accuracy and is fast, at the expense of a higher memory cost. Its use for flows that require special care for the gradients evaluations is shown. The presented case is a stationary 3D simulation of a plasma torch operated a 600 A and atmospheric pressure, and for which the elliptical part of the equations used to model the flow is critical. While no solution could be obtained for this case using a simple gradient evaluation method, a converged solution was obtained with the isoparametric transform method.

Acknowledgements

We are obliged to Jean-Luc Dorier for valuable discussions on the experimental aspects of the plasma torch and to Laurent Sansonnens for sharing his insight into arc modeling with us. We are grateful to Tony Murphy from the CSIRO for providing thermodynamic data of Argon at high temperature. All the computations have been performed on the computers of the Service informatique central de l'École polytechnique fédérale de Lausanne, in particular on the Origin 3000 and the Swiss T-1. This work was partially supported by

the Swiss National Science Foundation.

References

[1] Morton KW, Numerical Solution of Partial Differential Equations, Cambridge University Press, 1994

[2] Pfender E, Electric arcs and arc gas heaters, in Gaseous electronics, Vol. I, Electric discharges, Academic Press, 1978

[3] Vos JB et al., Recent Advances in Aerodynamics inside the NSMB Consortium. 36th Aerospace Sciences Meeting & Exhibit, Reno NV, January 1998

[4] Peyret R and Taylor TD, Computational Methods for Fluid Flows, Springer, Berlin, 1985

[5] Hyman J and Shashkov M, Mimetic Discretizations for Maxwell's Equations, J. Comp. Phys 1999;151:881–909

[6] Shashkov M, Conservative Finite-Difference Methods on General Grids, CRC Press, Boca Raton, FL, 1995

[7] Hermeline F, A Finite Volume Method for the Approximation of Diffusion Operators on Distorted Meshes, J. Comp. Phys 2000;160:481–499

[8] Wesseling P, Segal A and Kassels CGM, Computing flows on general three-dimensional non-smooth staggered grids, J. Comp. Phys 1999;149:333-362

[9] Patankar SV, Numerical Heat Transfer and Fluid Flow, Washington: Hemisphere Publishing Corporation, 1980

[10] Hughes T, The finite element method, Englewood Cliffs: Prentice-Hall, 1987

[11] Morton KW, Basic course in finite element methods. Comput. Phys. Rep. 1987;6:1–72

[12] Cao M, Proulx P and Boulos MI, Mathematical modelling of high-power transferred arcs, J. Appl. Phys. 1994;76:7757

[13] Kelkar M and Heberlein J, Physics of an arc in cross flow, J. Phys. D: Appl. Phys. 2000;33:2172–2182

[14] Snyder SC, Reynolds LD, Lassahn GD, Fincke JR, Shaw CB and Kearney RJ, Determination of gas-temperature and velocity profiles in an argon thermal-plasma jet by laser-light scattering, Physical Review E 1993;47:1996–2005

[15] Menart J, Malik S and Lin L, Coupled radiative, flow and temperature-field analysis of a free burning arc, J. Phys. D: Appl. Phys. 2000;33:257–269

[16] Zhu P, Lowke JJ and Morrow R, A unified theory of free burning arcs, cathode sheats and cathode, J. Phys. D: Appl. Phys. 1992;25:1221–1230

[17] Klinger L, Simulation numérique 3D d'une torche à plasma par une méthode de volumes finis, Ph.D. thesis no. 2678, Swiss Federal Institue of Technology, Lausanne 2002

[18] Dorier JL, Gindrat M, Hollenstein C, Salito A, Loch M and Barbezat G, Time-resolved imaging of Anodic Arc Root Behavior During Fluctuations of a DC Plasma Spraying Torch, IEEE Transactions on plasma science 2001;29:494–501

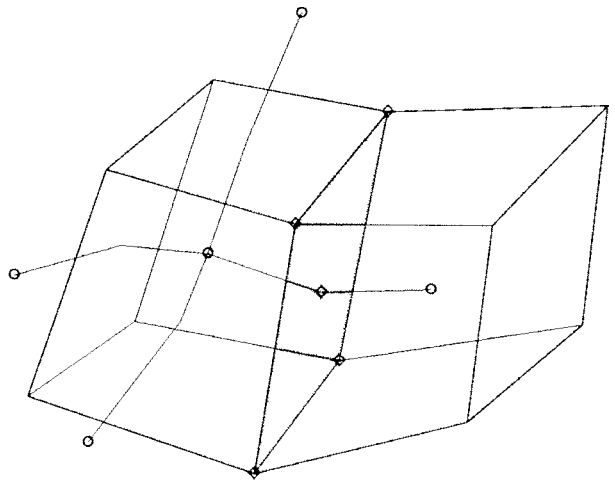

Fig. 1. Two cells of a mesh, showing cell centres $\circ$, where field value is known, and points where ϕ is interpolated $\diamond$ (used for surface-centered gradient). Not all nearest neighbour cell centres are shown, but all points used to estimate the side-centered gradient are shown (for the gray side).

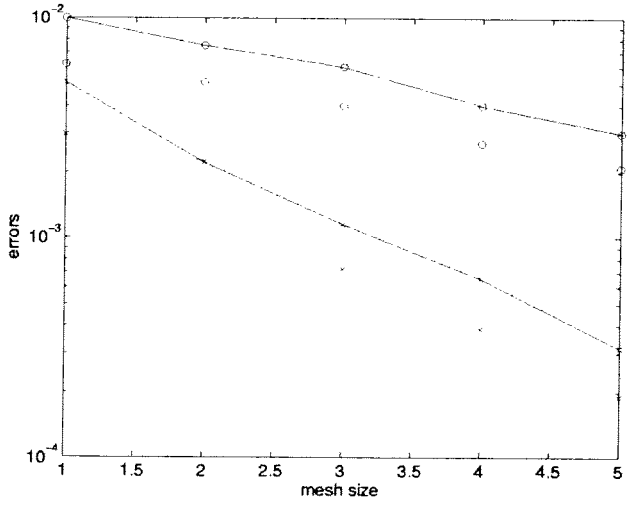

Fig. 2. Relative (solid line) and absolute (dotted line) errors on the gradient evaluation for ITM (crosses) and Peyret-Taylor (circles) formulas. The mesh used was L-shaped, having $24 \times 12 \times 12$, $36 \times 18 \times 18$, $48 \times 24 \times 24$, $72 \times 36 \times 36$ and $96 \times 48 \times 48$ cells.

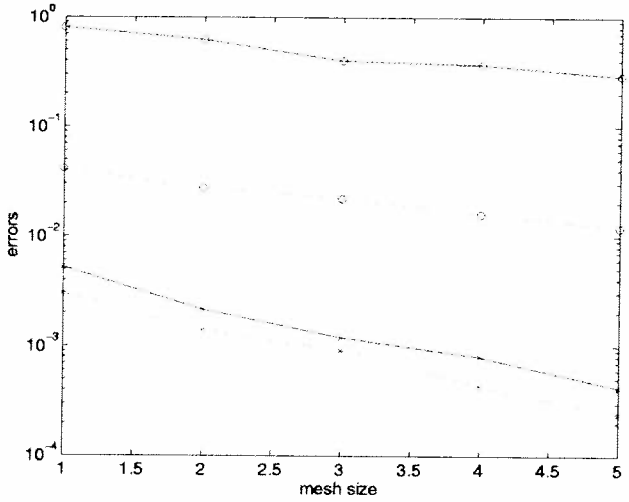

Fig. 3. Relative (solid line) and absolute (dotted line) errors on the gradient evaluation for ITM (crosses) and Peyret-Taylor (circles) formulas on a shaken L-shaped mesh.

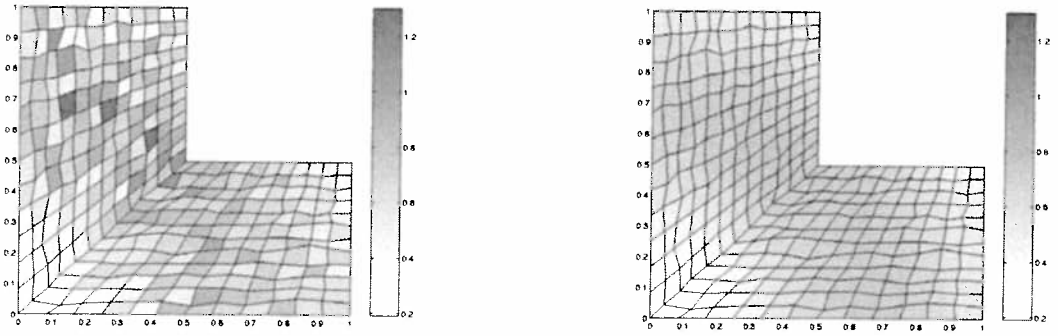

Fig. 4. Comparison between Peyret-Taylor (left) and ITM (right) evaluation of $(\nabla\phi)^2$ at cells centre, when $\phi(x,y,z) = \exp(-x^2 - y^2 - z^2)$. On such a plot, the ITM result cannot be distinguished from the analytical one.

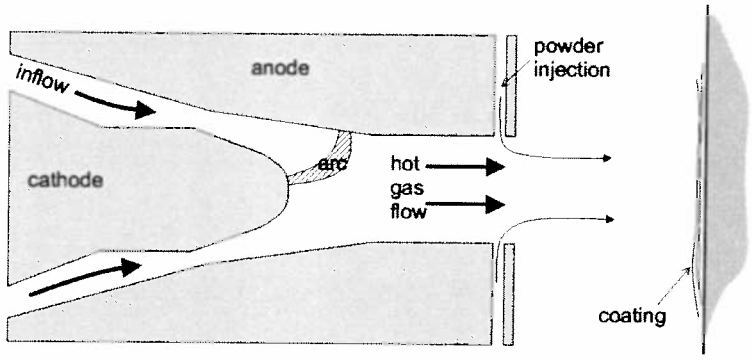

Fig. 5. Schema of a plasma torch used for thermal spraying; only the inside of the torch is modelled. Note that, while the electrodes have the cylindrical symmetry, this is not true for the arc column [18].

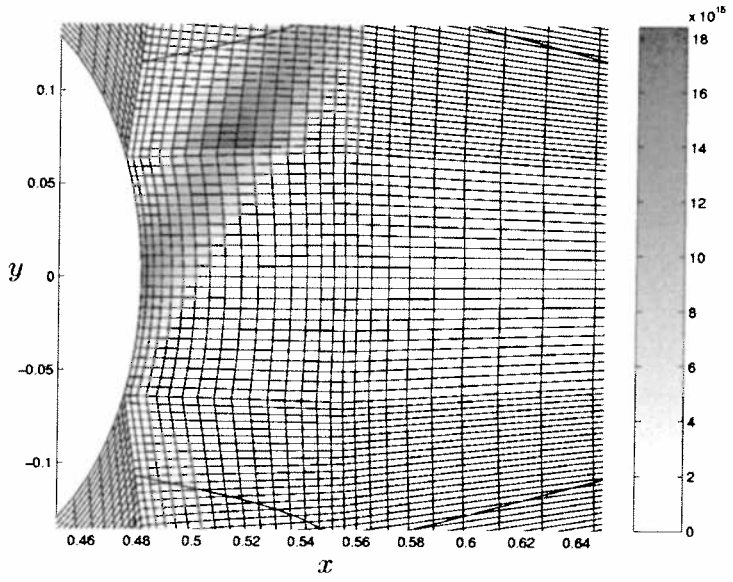

Fig. 6. Cut of the squared electric current density in the torch, after 200 iterations, using the Peyret-Taylor formula, showing problems where the mesh is non-orthogonal.

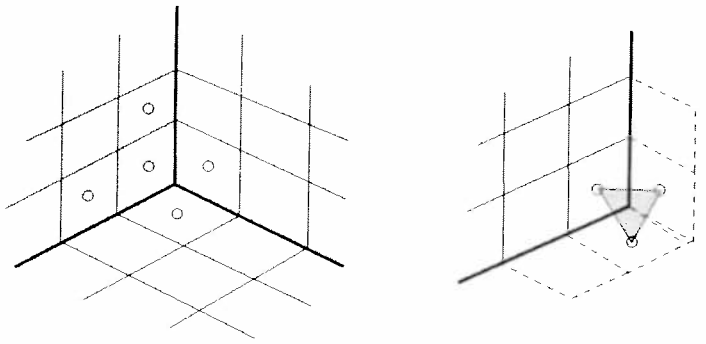

Fig. 7. On the left, a 2D case of three blocks having a point in common. Only the gray block is considered on the right, where the ghost cells are delimited by broken lines. The triangle represents the polygon used to interpolate the field value at the block corner.

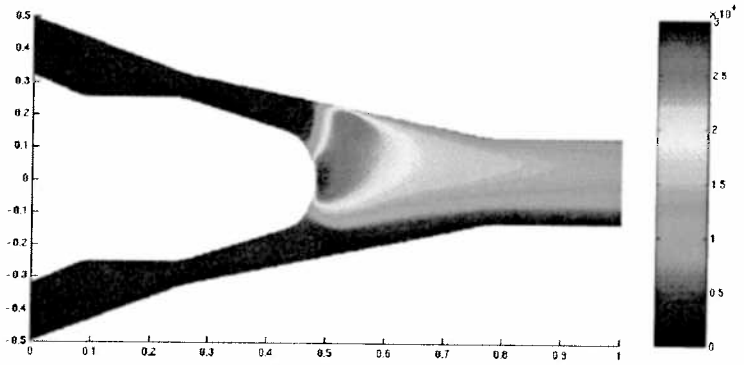

Fig. 8. Cut of the temperature, in Kelvins, in the symmetry plane of a 600 A torch, with 30 standard litres per minute Argon at inflow.