Supercomputing applications in materials engineering

G. Phanikumar*,‡, K. Chattopadhyay* and P. Dutta†

*Department of Metallurgy, and †Department of Mechanical Engineering, Indian Institute of Science, Bangalore 560 012, India

We review some of the materials processing problems that are being tackled by the scientific community today in the light of the high computing power available at affordable costs. Simulation challenges related to materials processing that once were thought of as impossible to tackle by computational methods are now feasible and are highlighted. As an illustration, computer simulation of materials joining is detailed.

1. Introduction

MANUFACTURING processes have become increasingly challenging in recent times. Stringent performance criteria and multi-disciplinary issues that arise during the processes, have rendered this area sophisticated. Most manufacturing processes involve heat transfer, melting, fluid flow, convection and diffusion of elements and solid–solid phase transformations that determine the mechanical properties, compositional variation/homogeneity, residual stresses and the overall performance of the final product. Thus materials design and manufacturing today is an inter-disciplinary science involving computational fluid dynamics, heat transfer, process design and materials science. Casting, joining, cladding, spray-forming, surface treatment and rapid-prototyping are some examples of such processes.

Insight into the physical processes that take place in these manufacturing methods and an understanding of the process as a function of input parameters such as the material properties, heat transfer conditions and the time scales involved will help us achieve a predictive capability about the final product and its performance. Complexity of product geometry, limited resources of material as well as man-hours, and complexity of the process itself do not always allow one to tune the input parameters to achieve desired end result via an experimental route. Often one would like to get the final product right with minimal experimental trials. Thus computer simulation becomes important in assisting product/process design which can often give insight into certain issues that could be expensive and sometimes impossible to achieve in experiments.

The length scales at which several of the processes that occur vary a lot and a simulation of the whole process with a single model could be CPU-intensive. For example, in case of laser processing of materials, the length scales for a specimen of a few centimeters in size would be millimeters for heat transfer, few tens of micrometers for convection and diffusion of elements and microns for solidification[1]. The physical processes that take place at these length scales vary significantly, rendering the computer simulation of the whole process a challenging task. It is not surprising that **CFD and Materials Design** is considered as one of the *grand challenges* for computational science[2].

With the increase in computational facilities, more problems are now amenable to solutions. Several problems give more insight by a mere increase in the resolution of the computational domain keeping the formulation almost the same. While a direct numerical simulation of turbulence modelling[3] is a standard example, there do exist several other successful applications in the recent literature that illustrate the same. Modelling of solidification and the resulting compositional variations have been successfully simulated, and today there could be a claim that a successful casting of almost any complexity could be made in one attempt. But most of the simulations which were two-dimensional with coarse mesh could not predict certain defects that were observed in the actual castings. These defects, called freckles, are minute channel formations in the cast ingot that are detrimental to the mechanical behaviour of the product. It was only recently that a three-dimensional simulation of the solidification process with a fine mesh led to a better understanding of the mechanism of formation of these defects[4].

2. Computing facilities

While the challenges and success stories encourage one to plunge into the area, any computational scientist would appreciate the fact that at the lower end the facilities and their affordability is also an important issue. High end computers like CRAY, IBM SP2, ONYX, SGI Power Challenger, etc. have been conventionally used for CPU-intensive applications. The high price of these systems has prohibited several moderately financed universities from

‡For correspondence. (e-mail: phani@metalrg.iisc.ernet.in)

venturing into computational science. The low cost of high (cycle) speed Intel processors and emergence of multiple processor computers have now provided an alternative to achieve high computing power with low costs. NASA's Beowulf project[5] is an example of success in this new alternative.

Most of the programs for parallel computers need some part of the code to instruct the processors about the various parts of the code that they execute and for communication with the other processors. There exist two major paradigms of parallel computer architectures: (a) Shared Memory Paradigm of computation, where all processors access one single chunk of memory eliminating most of the communication between the processors. Thus, the compiler itself can take care of almost all the instructions needed to distribute computation among the processors; (b) Distributed Memory Paradigm, where each processor can access only its own memory and any data needed from the memory of another processor can only be got by explicit communication with that processor. The latter class of systems is scalable, and is more popular. Here, the user has to write parts of the code to distribute the computation in an efficient manner. The Message Passing Interface (MPI) standard provides a high level layer over which the user can write this code. Once written, these programs can be ported to parallel computers of different kinds with minimal changes. This development has motivated growth of recent codes in this direction[6].

3. Illustration

As an illustration, we detail the usage of a parallel computer to simulate joining of materials. The problem involves determination of temperature field, velocity field in the liquid pool formed and composition variation in the sample that is heated with a source such as a laser or an electron beam. This problem, when applied to dissimilar materials with a moving heat source, will require one to use a three-dimensional domain with transient formulation. The governing equations include the Navier–Stokes equation for the solution of flow of liquid, diffusion–convection equation for determination of temperature and composition variations. The formulation, which is described elsewhere[7], consists of a set of highly nonlinear partial differential equations. The governing equations can be written in the canonical form as given below.

$$\frac{\partial(\rho\phi)}{\partial t}+\bar{\nabla}\cdot(\rho\bar{U}\phi)=\bar{\nabla}\cdot(\Gamma\bar{\nabla}\phi)+S. \tag{1}$$

ϕ is the variable we are solving for, such as temperature, velocity, pressure or composition, ρ is the density, $\bar{U}$ is the velocity field, Γ is the diffusivity and S is the source term. The solution procedure involves reduction of the governing equations into the following form amenable for solution by tridiagonal matrix algorithm (TDMA)[8].

$$a_i\phi_i = \Sigma_{j(\text{all neighbours})}\, a_i\phi_i + b_i. \tag{2}$$

Here a_i is the coefficient that determines the flux between the control volume in consideration and its neighbour, and b_i is the source term for the control volume. A typical problem involves a moderate number of grids (30 × 30 × 30), 6 variables, 1000 time-steps, 100 iterations per time-step and 6 rounds of TDMA leading to about 100 billion operations. With each operation taking several machine cycles, an IBM590 would take a couple of days to complete the solution. When the formulation for alloy solidification is added, the scale at which changes are expected to take place decreases and a much higher resolution is required for a meaningful solution. Equation (2) is inherently parallelizable because the solution of a variable at a grid point requires data only from its neighbours and data for the rest of the domain are not used. We exploit this feature of the problem to split the computational domain among the processors.

The paradigm used here is that of a distributed memory system. We use N processors to solve the problem. The domain is split into N parts as illustrated in Figure 1 so that each processor can compute for its subdomain with minimal communication with the neighbouring processor. The algorithm used is the same as that to solve Poisson equation with pipelining[9], i.e. each processor communicates with its left neighbour to exchange data, proceeds to solve the equations for its subdomain and then communicates with the right neighbour to exchange the solved data. Thus each processor lags behind its left neighbour by one iteration but is doing the computation simultaneously. Thus the parallelization facility is exploited. The solution is said to have been converged when the maximum of residuals of the parameter ϕ in the domain falls below a prescribed small number ε. Since, for every iteration of the solution, one communication needs to be done with the neighbour, we can bundle the residuals

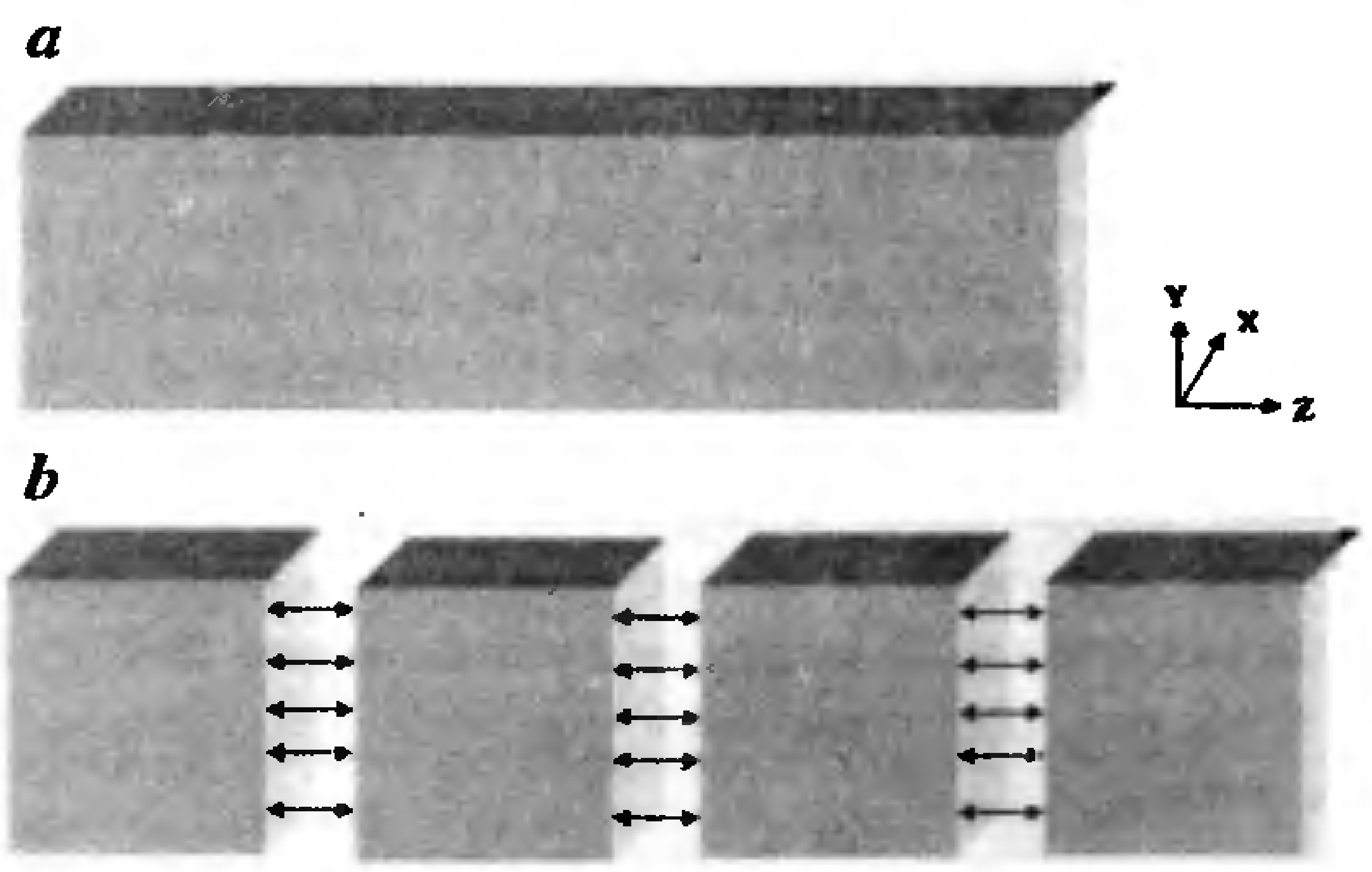

Figure 1. *a*, Computational domain; *b*, Domain decomposition. Communication indicated by the arrows is required only for the grid points on the inner faces of the sub-domains.

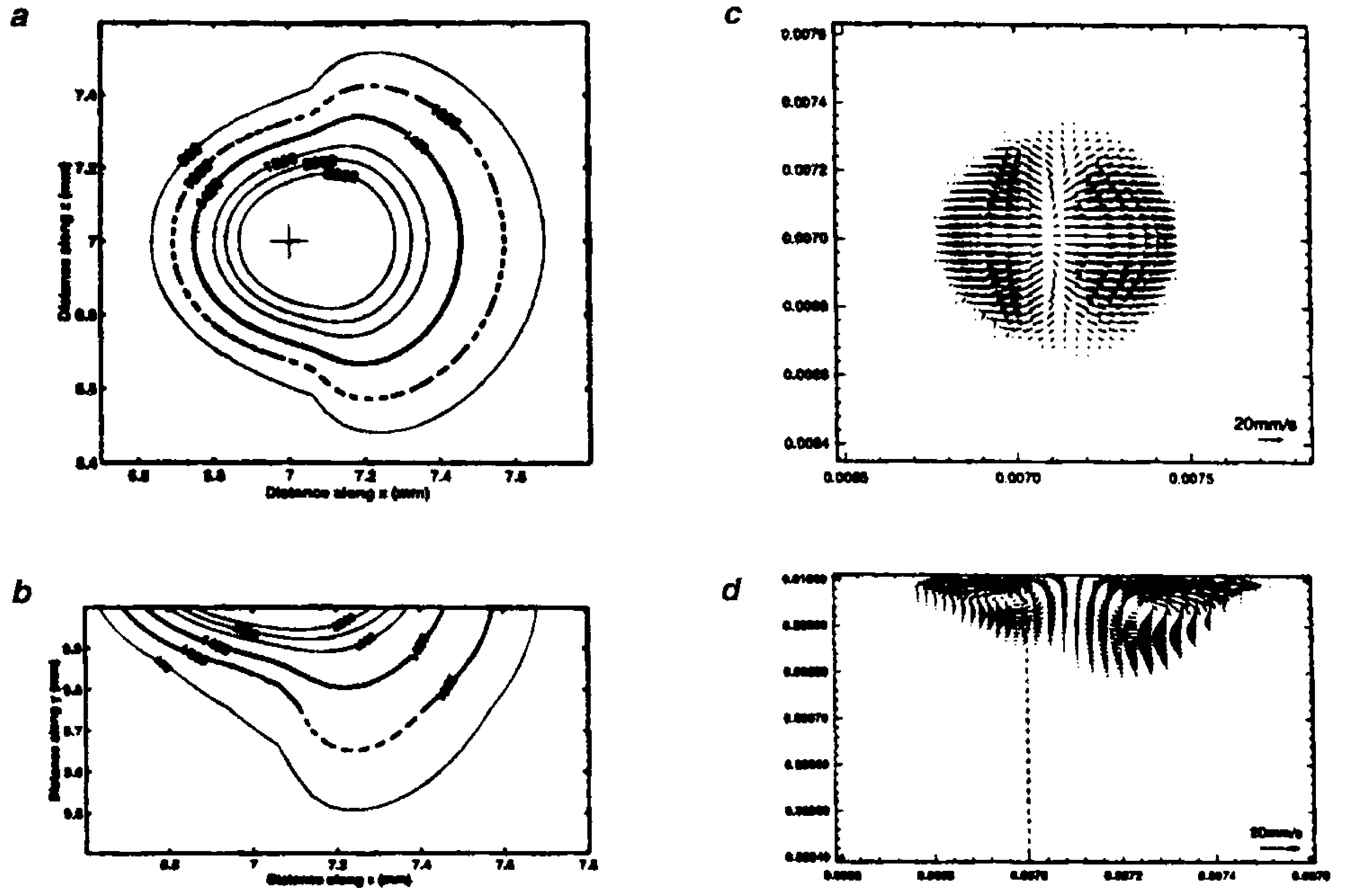

Figure 2. Temperature contours (*a*) Top view (*b*) Cross-sectional view and Velocity profiles (*c*) Top view and (*d*) Cross-sectional view of simulation of laser spot weld.

of previous iteration along with the ϕ values for exchange. Thus, within N–1 iterations, all the processors will know the residuals of all subdomains, and determination of a maximum, checking for the convergence criterion, and making a decision to move into the next time-step or continuing with the iterations will become trivial. Hence, with minimal communication, one can get a speedup of almost N times by using N processors. N has to be small compared to the number of grid points in the direction along which the domain is split into subdomains.

Figure 2 illustrates a typical computational simulation of laser spot welding of two dissimilar metals (copper and nickel, in this case). The figure which shows the temperature and velocity fields, clearly demonstrates the asymmetric weld pool formation as observed in experiments. This parallel program was written in FORTRAN90 and run on an IBMSP2. The program which would have taken 80 h on an IBM580 has completed execution in $\approx$ 12 h on the parallel machine using 8 nodes.

4. Conclusions

Numerous computational challenges in materials science are easier to tackle today, in the wake of high computing power. Several examples have brought forward more insights into the problem because of this development. An illustration of parallelization of a metal joining code is discussed.

1. Rappaz, M., *Int. Mater. Rev.*, 1989, **34**, 93–123.
2. Eugene Levine, *Commun. ACM*, 1989, **12**, 1456–1457.
3. Paul R. Woodward, *IEEE Comput. Sci. Eng.*, winter, 1994, 4–5.
4. Felicelli, S. D. *et al.*, *Metall. Mater. Trans. B*, 1998, **29**, 847.
5. Daniel Ridge *et al.*, Proc. IEEE Aerospace, 1997.
6. Cherri M. Pancake, *IEEE Comput. Sci. Eng.*, 1996, **3**, 18–37.
7. Dutta, P. *et al.*, *Numer. Heat Transfer A*, 1995, **27**, 499.
8. Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publications, New York, 1980, 2nd edn.
9. Michael J. Quinn, *Parallel Computing: Theory and Practice*, McGrawHill, 1994, 2nd edn, pp. 217–254.

ACKNOWLEDGEMENTS. We thank Supercomputer Education and Research Center (SERC), Indian Institute of Science (IISc), Bangalore for its facilities and Dr Siddhartha Shankar Ghosh, SERC, IISc for help and encouragement during parallelization of their program.