

Parallel Performance in Multi-Physics Simulation

Kevin McManus, Mark Cross, Chris Walshaw, Nick Croft, and Alison Williams

Centre for Numerical Modelling and Process Analysis, University of Greenwich,
London SE10 9LS,
k.mcmanus@gre.ac.uk,
WWW home page: <http://multi-physics.com>

Abstract. A comprehensive simulation of solidification/melting processes requires the simultaneous representation of free surface fluid flow, heat transfer, phase change, non-linear solid mechanics and, possibly, electromagnetics together with their interactions in what is now referred to as 'multi-physics' simulation. A 3D computational procedure and software tool, PHYSICA, embedding the above multi-physics models using finite volume methods on unstructured meshes (FV-UM) has been developed. Multi-physics simulations are extremely compute intensive and a strategy to parallelise such codes has, therefore, been developed. This strategy has been applied to PHYSICA and evaluated on a range of challenging multi-physics problems drawn from actual industrial cases.

1 Introduction

Typically, solidification/melting processes can involve free surface turbulent fluid flow, heat transfer, change of phase, non-linear solid mechanics, electromagnetic fields and their interaction, often in complex three dimensional geometries. The heritage of computational mechanics modelling is such that most Computer Aided Engineering (CAE) software tools have their focus upon one of 'fluids' (CFD)¹[11, 5], 'structures' (CSM)²[3, 1] or 'electromagnetics' (CEM)³[7]. As such, much of the work on phase change processes has addressed either thermofluid or thermomechanical aspects alone. However, as the customer demands ever higher levels of product integrity, then this has its impact upon the need for more comprehensive simulation of all the component phenomena and their interactions. Such 'multi-physics' simulation requires software technology that facilitates the closely coupled interaction that occur amongst the component phenomena. Experience over many years with closely coupled thermomechanical, thermofluid and magnetohydrodynamic applications, has demonstrated that to both capture accurately the loads, volume sources or boundary condition effects from one phenomena in another, in a computationally effective manner,

¹ Computational Fluid Dynamics

² Computational Structural Mechanics

³ Computational Electromagnetics

requires all the continuum phenomena to be solved within one software environment.

It is, of course, well recognised that single phenomena simulation based upon CFD, CSM or CEM is computationally intensive, especially for large complex meshes, and may involve many thousands of time-steps in transient problems. The implication here, is that if single phenomenon simulation is a computational challenge, then the challenge is more significant for multi-physics modelling. The authors and their colleagues have been involved in the development of computational procedures and software tools for multi-physics simulation for over a decade. The results of this research have led to the production of the PHYSICA [9, 19] software environment for the computational modelling of multi-physics processes in complex three dimensional geometries. One issue at the core of the development of these multi-physics tools has been the concern to ensure that the software will run effectively on high performance parallel systems. Multi-physics simulation of solidification/melting processes involves a computational load in each control volume of the mesh (representing the geometrical domain) that varies with the cocktail of physics active in the volume, and which may well change with time. As such, from a parallel operation perspective, multi-physics simulation represents a dynamically varying non-homogeneous load over the mesh. A generic approach to this challenge is described in [14].

2 Software Technology Overview

2.1 Generic models and computational procedures

In the context of phase change simulation the following continuum phenomena and their interactions are of key significance:

- free surface transient Navier Stokes fluid flow,
- heat transfer,
- solidification/melting phase change,
- non-linear solid mechanics and possibly
- electromagnetic forces.

It is useful to observe that all the above continuum phenomena can be written in a single form where Table 1 provides a summary of the terms required to represent the equation for each of the above phenomena [9]:

$$\frac{\partial}{\partial t} \int_v \rho A \phi dv = \int_s \Gamma_\phi \nabla \phi \mathbf{n} ds + \int_v Q_v dv - \int_s \mathbf{Q}_s \mathbf{n} ds \quad (1)$$

The suite of solution procedures chosen for this work is based upon an extension of finite volume (FV) techniques from structured to unstructured meshes. The fluid flow and heat transfer [6], phase change [8] and electromagnetics [18] procedures are based upon cell centred approximations, where the control volume is the element itself. The solid mechanics algorithms used in this work [4,

Phenomenon	ϕ	A	Γ_ϕ	Q_v	\mathbf{Q}_s
Continuity	1	1	0	S_{mass}	$\rho\mathbf{v}$
Velocity	\mathbf{v}	1	μ	$S + \mathbf{J} \times \mathbf{B} - \nabla p$	$\rho\mathbf{v}\mathbf{v}$
Heat Transfer	\mathbf{h}	1	k/c	S_h	$\rho\mathbf{v}h$
Electromagnetic Field	\mathbf{B}	1	η	$(\mathbf{B}\nabla)\mathbf{v}$	$\mathbf{v}\mathbf{B}$
Solid Mechanics	\mathbf{u}	$\frac{\partial}{\partial t}$	μ	$\rho\mathbf{f}_b$	$\mu(\nabla\mathbf{u})^T + \lambda(\nabla \cdot \mathbf{u} - (2\mu + 3\lambda)\alpha T)\mathbf{I}$

Table 1. Definition of terms in the generic transport equation

22] employ a vertex based approximation, so that the control volume is assembled from components of the neighbouring cells/elements to a vertex. The 'cell centred' phenomena are all solved using an extension of the conventional SIMPLE pressure correction procedures originated by Patankar and Spalding [17]. As this is a co-located flow scheme, the Rhie-Chow approximation is used to prevent checker-boarding of the pressure field [21]. The solid mechanics solution procedure involves a formulation as a linear system in displacement and solved in a similar manner to finite element methods [16]. Indeed, this enables the FV procedures to be readily extended to a range of nonlinear behaviours [22]. At this stage, a cautious approach to the solution strategy has been explored. A complete system matrix is constructed regardless of the physical state of each element or node. This approach is key for solidification/melting processes because, as phase change fronts move through the domain, then the local combination of physics changes with time and the transition between solid and liquid is not necessarily well defined. Hence, each phenomena is solved over the entire mesh for each time step.

2.2 The Parallelisation Strategy

Use of a single program multi-data (SPMD) strategy employing mesh partitioning is now standard for CFD and related codes. When the code uses an unstructured mesh, the mesh partitioning task is non-trivial. This work has used the JOSTLE graph partitioning and dynamic load-balancing tool [12, 23]. However, a key additional difficulty with respect to multi-physics simulation tools for phase change problems is that the computational workload per node/mesh element is not constant.

At this exploratory stage of multi-physics algorithm development, a cautious strategy has been followed, building upon established single discipline strategies (for flow, structures, etc) and representing the coupling through source terms, loads, etc [9]. Each discipline may of course involve the solution of one or more variables. A complication here is that separate physics procedures may use differing discretisation schemes. For example, in PHYSICA the flow procedure is cell centred, whereas the structure procedure is vertex centred. It follows that if a parallelisation strategy for closely coupled multi-physics simulation is to be effective, it must be able to achieve a load balance within each single discipline solution procedure.

It is the multi-phase nature of the problem that is a specific challenge here. The parallelisation strategy proposed is then essentially a two-stage process:

- a) The multi-physics application code is parallelised on the basis of the single mesh, using primary and secondary partitions for the varying discretisation schemes [14],
- b) Determination of a mesh partition that provides the required characteristics of good load balance with low inter-processor communication.

This strategy simplifies both the adaption of the code to run in parallel and the subsequent use of the code in parallel by others. However, it requires a mesh (graph) partitioning tool that has the following capabilities:

- produces load balanced partition for a single (possibly discontinuous) graph with a non-homogeneous workload per node,
- structures the sub-domain partitions so that they minimise inter-processor communication (i.e. the partitions respect the geometry of the problem),
- operates with a graph that is distributed across a plurality of processors (memory).

2.3 Load balancing

A key issue in multi-physics parallelisation is the use of primary and secondary partitions to cope with distinct discretisation techniques (ie. cell centred for flow and vertex centred for solid mechanics). In principle, one could use distinct (i.e. un-related) partitions for each discretisation technique, but this would seriously compromise any parallel scalability, because of the potential communication overheads. The approach is straightforward:

- a mesh entity type (e.g. element) associated with the greatest computational load is selected for a primary partition.
- secondary partitions for the other mesh entity types are derived from the primary partition to satisfy both load balance and minimisation of the overlap depth into neighbouring domains

2.4 Message passing approach

All data access that is not local to a processor (sub-domain) necessitates communication of the required data as a message from the processor that has the data to the processor that requires the data. The overhead of this communication limits parallel performance. The time required for message passing can be characterised in terms of the bandwidth (for long messages) and the communication start-up latency (for short messages). With current technology, processor speed is high in comparison to latency and this significantly restricts parallel performance.

PHYSICA uses a generic 'thin layer' message passing library, CAPLib [13], which provides a highly efficient portability layer that maps onto PVM [20],

MPI [15], shared memory and other native message passing systems. CAPLib is targeted at computational mechanics codes and provides a flexible and compact data model that is very straightforward to apply with no measurable performance overhead [13].

2.5 Portability, Transparency and Scalability

Parallel systems exhibit a diversity of characteristics that are of little or no concern to numerical modellers. The PHYSICA code must port without difficulty or compromise of performance onto a wide range of parallel platforms. Similarly it is essential that no system specific case configuration is required in moving between serial and parallel systems. Serial and parallel geometry and data files are therefore the same, although not necessarily identical. Provided that the user's model stays within the framework prescribed by the 'core' PHYSICA modules, then parallelism is straightforward. The intention is to reduce the difference between running in serial and parallel to be simply the executable instruction. Portability is achieved through the combination of a highly standardised programming language, Fortran77 and the portability layer, CAPLib. Transparency has been achieved by embedding JOSTLE into parallel PHYSICA to provide run-time partitioning.

Scalability may be seen as the extent to which either more processors will reduce run-time or model size may be increased by using more processors. As the number of processors, P , is increased with a fixed problem size the performance will encounter an Amdahl limit [2] because the problem size per processor becomes so small that inherently sequential operations dominate run-time.

2.6 Machine Characterisation

Run time on both the SMP cluster and the NUMA system is dependent upon several factors. Within each system there is competition for access to the memory bus. Execution times are consequently related to the overall load on the machine. In addition, performance of current generation high clock speed, superscalar pipelined processors is closely tied to cache success (hit) rate. Cache hit rates begin to suffer as vector lengths become large in comparison to the cache size and so single processor performance deteriorates. Inter node communication is affected by bottlenecks in the inter-processor communications and so run times vary with activity across the entire machine.

3 High Performance Computing Systems Used

The key objectives of this research programme were to evaluate the parallel scalability performance of the multi-physics simulation code, PHYSICA, for a range of problems on two standard high performance computing systems.

3.1 Compaq Alpha System

The Compaq Alpha System is a symmetric multi-processor (SMP) cluster consisting of three Compaq 4100 quad processor SMP nodes, providing a total of 12 processors with 3Gb of memory. Each Alpha processor is an EV5/6 running at 466Mhz. High speed, low latency inter-node communication is provided by Memory Channel MkI. The inter-processor communication characteristics of this system for MPI calls are approximately 5 nanosecond communication start up latency and 55 Mbytes per second bandwidth.

3.2 SGI Origin 2000 System

These systems are described as cache coherent non-uniform memory access (NUMA) systems. Each computer node has two MIPS R10000 processors with 1Gb of memory and there are up to 32 nodes clustered as a single address space within a cabinet. The overall MPI interprocessor communication characteristics within a box are approximately 5 nanosecond communication start-up latency and a bandwidth of 300 Mbytes per second.

4 Results and Discussion

To provide realistic measures of parallel performance, the test cases discussed here have been developed from actual PHYSICA applications. In order to encourage efficient use and provide meaningful performance figures, PHYSICA reports the run times split into file access time and calculation time. The sum of these two times provides overall run or wallclock time. The test cases aim to provide a realistic view of these times.

4.1 Case 1: Shape casting

In this case the metal-mould arrangement has a complicated three dimensional geometry resulting in a mesh of 82,944 elements and 4 materials illustrated in Figure 1. This case was investigated as a thermo-mechanical problem from which a thermal only parallel test case has been extracted. Run time and parallel speedup results for the Compaq and SGI systems are given in Tables 2 and 3

4.2 Case 2: Aluminium reduction cell

This process involves the interactions between the electric current density, magnetic field, temperature distribution and phase change resulting in a two-phase flow behaviour with thermally induced stresses and deformations. A simplified model of this process considers 'lumped' source terms for the Joule heating, electromagnetic force field and bubble movement. This reduces the modelling to a consideration of the Navier-Stokes fluid flow, heat transfer, phase change, and solid mechanics [10]. These are solved over a mesh of 23,408 elements of mixed

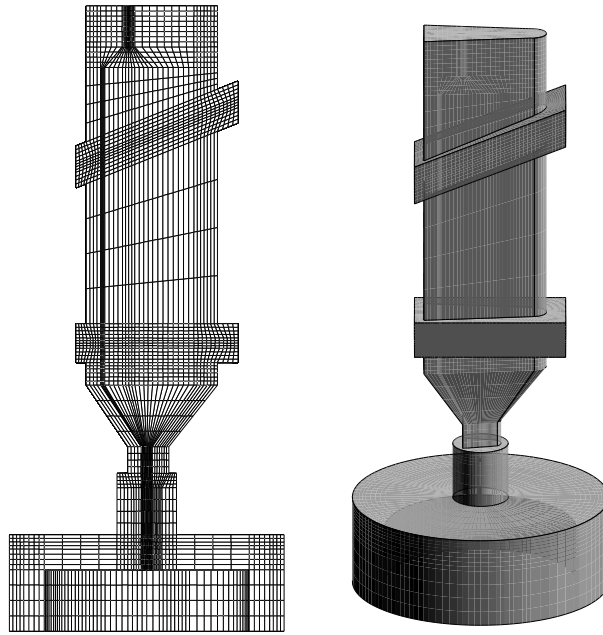


Fig. 1. Mesh and surface of the turbine blade test case

P	t calculation	t file	t overall	Sp calculation	Sp file	Sp overall
1	3644	45	3689	1	1	1
2	1950	72	2022	1.87	0.63	1.82
4	610	73	683	5.97	0.62	5.40
8	323	83	406	11.28	0.54	9.09
12	285	92	377	12.78	0.49	9.79

Table 2. Run times and speedup for the Turbine Blade test case on a Compaq 4100 SMP cluster

P	t overall	Sp overall
1	4177	1
2	1912	2.18
4	1051	3.97
6	761	5.49
8	812	5.14
12	676	6.17
16	657	6.36
32	547	7.64

Table 3. Run times and speedup for the Turbine Blade test case on a SGI Origin system

type with 6 materials illustrated in Figure 2. In this case the linear solvers run to convergence using JCG⁴ for heat and flow and BiCG⁵ for stress. Run time and parallel speedup results for the Compaq and SGI systems are given in Tables 2 and 3

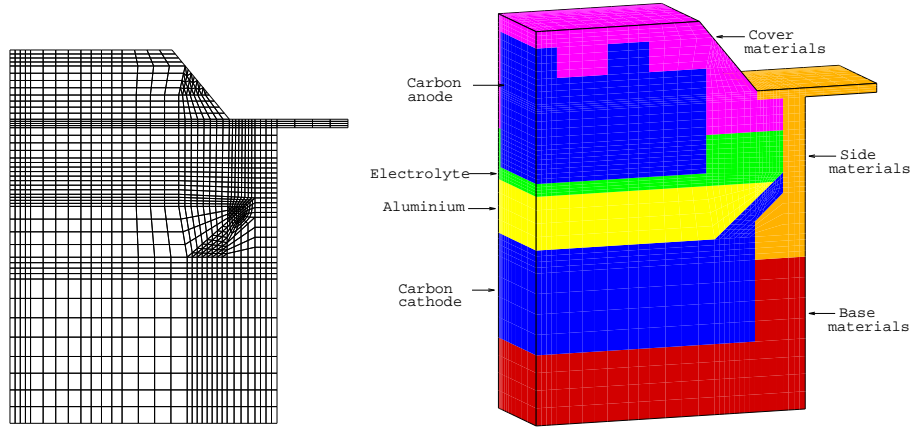


Fig. 2. Mesh and materials for the aluminium reduction cell test case

P	t calculation	t file	t overall	Sp calculation	Sp file	Sp overall
1	3860	52	3912	1	1	1
2	1936	73	2009	1.99	0.71	1.95
4	928	88	1016	4.16	0.59	3.85
8	470	92	562	8.21	0.57	6.96
12	346	120	466	11.16	0.43	8.39

Table 4. Run times and speedup for the Aluminium Reduction Cell with flow, heat transfer, solidification and stress on a Compaq 4100 SMP cluster

5 Conclusion

The work presented in this paper has described the performance results of an investigation into the parallelisation of an unstructured mesh code targeted at multi-physics simulation. Although the code uses finite volume methods, typical of CFD calculations, they are complicated by two factors:

⁴ Jacobi preconditioned conjugate gradient

⁵ Bi-directional conjugate gradient

P	t overall	Sp overall
1	6579	1
2	3707	1.77
4	1524	4.31
8	1310	5.02
16	833	7.90
32	808	8.14

Table 5. Run times and speedup for the Aluminium Reduction Cell simulation with flow, heat transfer, solidification and stress on a SGI Origin system

- the mixture of physics is much more complex than conventional CFD and varies both in time and space across the solution domain, and
- the simultaneous exploitation of a variety of discretisation methods

The load balancing task requires that the graph passed the to partitioner accurately represents the computational load of each mesh entity. This representation can only be an estimate but the results clearly demonstrate that the load balance is good, otherwise we would not see the significant superlinear speed-up.

Three factors dominate parallel performance:

- Load balance
- Communication latency
- File access

Although the test cases employ only static load balancing for dynamically inhomogeneous multi-physics problems the resulting load balance is remarkably good. This is in part due to the cautious solution strategy currently used in PHYSICA and in part due to the reasonably homogeneous mesh element shapes providing reasonable alignment between the primary and secondary partitions.

Latency has a marked effect on parallel performance. This partly why the first example performs so well by removing the norm calculations required to determine convergence in the linear solvers. This is fortunately consistent with SIMPLE type schemes. As computer technology develops the gap between latency and processor performance is increasing and latency effects are expected to become increasingly problematic.

Increasingly the results from modelling take the form of animated images and so a typical PHYSICA run will write significant quantities of data to file during execution. File access remains a problem for parallel performance. Certainly it is possible to stripe the result files across multiple hard drives but striped data has, at some point, to be reconstructed into global data in order to be visualised. Data striping is to a large extent postponing the parallel overhead.

What is clear from this work is that although the current parallelisation strategy is effective for modest numbers of processors, a more scalable strategy will eventually be required. This requires a less conservative sequential solution strategy together with dynamic load balancing parallel solution. Such an effort is now underway.

References

1. Abaqus. URL: <http://www.hks.com>.
2. G. M. Amdahl. Validity of the single-processor approach to achieving large scale computing capabilities. In *Proc AFIPS*, pages 483–485, 1967.
3. Ansys. URL: <http://www.ansys.com>.
4. C. Bailey and M. Cross. A finite volume procedure to solve elastic solid mechanics problems in three dimensions on an unstructured mesh. *Int. J Num Meth in Engg*, 38:1757–776, 1995.
5. CFX. URL: <http://www.software.aea.com/cfx>. AEA.
6. P. Chow, M. Cross, and K. Pericleous. A natural extension of standard control volume CFD procedures to polygonal unstructured meshes. *Appl. Math Modelling*, 20:170–183, 1995.
7. Concerto. URL: <http://www.vectorfields.co.uk>. Vector Fields.
8. N. Croft, K. Pericleous, and M. Cross. PHYSICA: a multiphysics environment for complex flow processes. *Numerical Methods in Laminar and Turbulent Flows*, Vol IX:1269 – 1280, 1995.
9. M. Cross. Computational issues in the modelling of materials based manufacturing processes. *Journal of Computationally Aided Materials*, 3:100–116, 1996.
10. M. Cross et al. Computational modelling of casting processes - a multi-physics challenge. In G. Irons and A. Cramb, editors, *Brimacombe Memorial Symposium Proceedings*, pages 439–450. Met Soc (Canada), 2000.
11. Fluent. URL: <http://www.fluent.com>.
12. JOSTLE. URL: <http://www.gre.ac.uk/jostle>. University of Greenwich.
13. P. Leggett, S. P. Johnson, and M. Cross. CAPLib - a 'thin layer' message processing library to support computational mechanics codes on distributed memory parallel systems. *Advances in Engineering Software*, 32:61–81, 2001.
14. K. McManus, M. Cross, C. Walshaw, S. Johnson, and P. Leggett. A scalable strategy for the parallelization of multiphysics unstructured-mesh iterative codes on distributed-memory systems. *Int Jnl of High Performance Computing Applications*, 14(2):137–174, 2000.
15. MPI: Message Passing Interface. URL: <http://www.mcs.anl.gov/mpi/index.html>. Argonne National Laboratory.
16. E. Onate, M. Ceiveia, and O. C. Zienkiewicz. A finite volume format for structural mechanics. *Int J Num Maths in Engg*, 37:181–201, 1994.
17. S. V. Patankar and D. B. Spalding. A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flows. *Int J Heat Mass Trans*, 15:1787–1806, 1972.
18. K. Pericleous, M. Cross, M. Hughes, and D. Cook. Mathematical modelling of the solidification of liquid tin with electromagnetic stirring. *Jnl of Magnetohydrodynamics*, 32(4):472–478, 1996.
19. Physica. URL: <http://www.multi-physics.com>. University of Greenwich.
20. PVM Parallel Virtual Machine. URL: <http://www.epm.ornl.gov/pvm/pvmhome.html>. Oak Ridge National Laboratory.
21. C. Rhie and W. Chow. A numerical study of the flow past an isolated airfoil with trailing edge separation. *JAIAA*, 21:1525–1532, 1982.
22. G. Taylor, C. Bailey, and M. Cross. Solution of elasto-visco-plastic constitutive equalities: a finite volume approach. *Appl Math Modelling*, 19:746–760, 1995.
23. C. Walshaw and M. Cross. Parallel optimisation algorithms for multi-level mesh partitioning. *Parallel Computing*, 26:1635–1660, 2000.