

# Virtual Manufacturing and Design in the Real World – Implementation and Scalability on HPPC Systems

K McManus, M Cross, C Walshaw, S Johnson, C Bailey,  
K Pericleous, A Slone and P Chow<sup>†</sup>

Centre for Numerical Modelling and Process Analysis  
University of Greenwich, London, UK.

<sup>†</sup>FECIT, Uxbridge, UK

Virtual manufacturing and design assessment increasingly involve the simulation of interacting phenomena, *sic.* multi-physics, an activity which is very computationally intensive. This paper describes one attempt to address the parallel issues associated with a multi-physics simulation approach based upon a range of compatible procedures operating on one mesh using a single database – the distinct physics solvers can operate separately or coupled on sub-domains of the whole geometric space. Moreover, the finite volume-unstructured mesh solvers use different discretisation schemes (and, particularly, different ‘nodal’ locations and control volumes). A two-level approach to the parallelisation of this simulation software is described: the code is restructured into parallel form on the basis of the mesh partitioning alone, i.e. without regard to the physics. However, at run time, the mesh is partitioned to achieve a load balance, by considering the load per node/element across the whole domain. The latter of course is determined by the problem specific physics at a particular location.

## 1. INTRODUCTION

As industry moves inexorably towards a simulation-based approach to manufacturing and design assessment, the tools required must be able to represent all the phenomena active together with their interactions, (increasingly referred to as multi-physics). Conventionally, most commercial tools focus upon one main phenomena, (typically ‘fluids’ or ‘structures’) with others supported in a secondary fashion – if at all. However, the demand for multi-physics has brought an emerging response from the CAE sector – the ‘structures’ tools ANSYS<sup>(1)</sup> and ADINA<sup>(2)</sup> have both recently introduced flow modules into their environments. However, these ‘flow’ modules are not readily compatible with their ‘structures’ modules, with regard to the numerical technology employed. Thus, although such enhancements facilitate simulations involving loosely coupled interactions amongst fluids and structures, closely coupled situations remain a challenge. A few tools are now emerging into the

community that have been specifically configured for closely coupled multi-physics simulation, see for example SPECTRUM<sup>(3)</sup>, PHYSICA<sup>(4)</sup> and TELLURIDE<sup>(5)</sup>. These tools have been designed to address multi-physics problems from the outset, rather than as a subsequent bolt-on.

Obviously, multi-physics simulation involving ‘complex physics’, such as CFD, structural response, thermal effects, electromagnetics and acoustics (not necessarily all simultaneously), is extremely computationally intensive and is a natural candidate to exploit high performance parallel computing systems. This paper highlights the issues that need to be addressed when parallelising multi-physics codes and provides an overview description of one approach to the problem.

## **2. THE CHALLENGES**

Three examples serve to illustrate the challenges in multi-physics parallelisation.

### **2.1. Dynamic fluid – structure interaction (DFSI)**

DFSI finds its key application in aeroelasticity and involves the strong coupling between a dynamically deforming structure (e.g. the wing) and the fluid flow past it. Separately, these problems are no mean computational challenge – however, coupled they involve the dynamic adaption of the flow mesh. Typically, only a part of the flow mesh is adapted; this may well be done by using the structures solver acting on a sub-domain with negligible mass and stiffness. Such a procedure is then three-phase<sup>(6-7)</sup>:

- The flow is solved for and pressure conditions loaded onto the structure.
- The structure responds dynamically.
- Part of the flow mesh (near to the structure) is adapted via the ‘structures’ solver and both the adapted mesh and mesh element velocities are passed to the flow solver.

Here we are dealing with 3 separate sub-domains (structure mesh, deformed flow mesh and the whole flow mesh) with three separate ‘physics’ procedures (dynamic structural response, static structural analysis and dynamic flow analysis). Note the deformed flow mesh is a subset of the whole flow mesh.

### **2.2. Flow in a cooling device**

The Scenario here is reasonably straight forward; hot fluid is passing through a vessel and loses heat through the walls. As this happens, the walls develop a thermally based stress distribution. Of course, the vessel walls may also deform and affect the geometry (i.e. mesh) of the flow domain. In the simplest case, part of the domain is subject to flow and heat transfer, whilst the other is subject to conjugate heat transfer and stress development. Of course, if the structural deformation is large, then the mesh of the flow domain will have to be adapted.

### **2.3. Metals casting**

In metals casting, hot liquid metal fills a mould, then cools and solidifies. The liquid metal may also be stirred by electromagnetic fields to control the metallic structure. This problem is complex from a load balancing perspective:

- The flow domain is initially full of air, and as the metal enters this is expelled; the air-metal free surface calculation is more computationally demanding than the rest of the flow field evaluation in either the air or metal sub-domains.
- The metal loses heat from the moment it enters the mould and eventually begins to solidify.
- The mould is being dynamically thermally loaded and the structure responds to this.
- The electromagnetic field, if present, is active over the whole domain.

Although the above examples are complex, they illustrate some of the key issues that must be addressed in any parallelisation strategy that sets out to achieve an effective load balance:

- In 2.1 the calculation has 3 phases (see Figure 1a); each phase is characterised by one set of physics operating on one sub-mesh of the whole domain; one of these sub-meshes is contained within another.
- In the simpler case in 2.2 the thermal behaviour affects the whole domain, whilst the flow and structural aspects affect distinct sub-meshes (see Figure 1b).
- In the more complex case in 2.2 the additional problem of adapting the flow sub-mesh (or part of it) re-emerges.
- In the casting problem in 2.3 the problem has three sub-domains which have their own ‘set of physics’ (see figure 1c) – however, ‘one set of physics’, the flow field, has a dynamically varying load throughout its sub-domain, and two of the sub-domains vary dynamically. Actually, if the solidified metal is elasto-visco-plastic, then its behaviour is non homogeneous too.

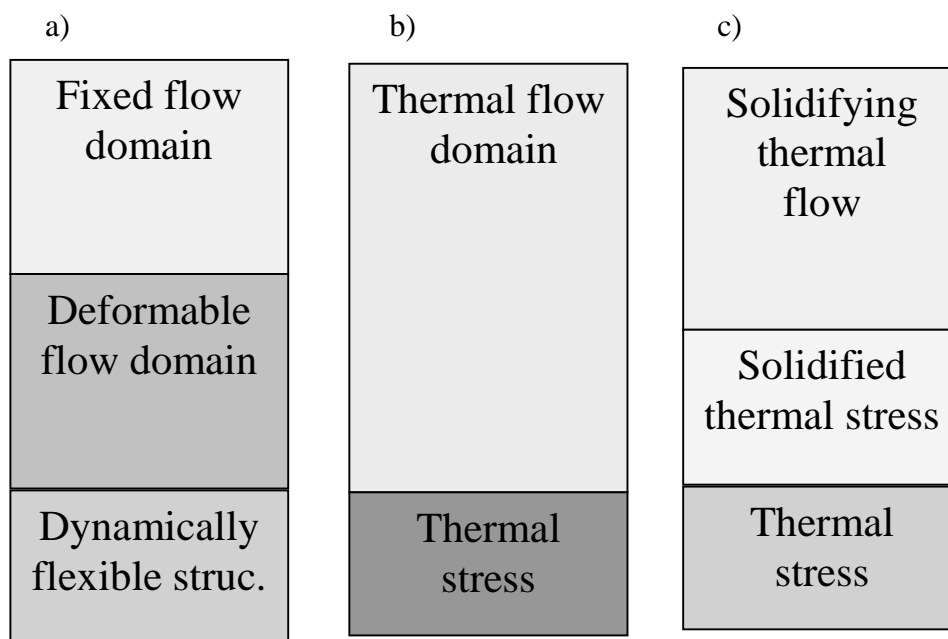


Figure 1 - Examples of differing mesh structures for various multi-physics solution procedures a)DFSI b) thermally driven structural loading and c) solidification.

The approach to the solution of the multi-physics problems, followed here, has used segregated procedures in the context of iterative loops. It is attractive to take the approach of formulating the numerical strategy so that the whole set of equations can be structured into one large non-linear matrix. However, at this exploratory stage of multi-physics algorithm development, a more cautious strategy has been followed, building upon tried and tested single discipline strategies (for flow, structures, etc.) and representing the coupling through source terms, loads, etc.<sup>(8)</sup>.

An added complication here is that separate physics procedures may use differing discretisation schemes, for example, the flow procedure may be cell centred, whilst the structure procedure will be vertex centred.

### 3. A PRACTICAL STRATEGY

Despite all the above complexity the multi-physics solution strategy may be perceived, in a generic fashion, as:

```
Do 10 i = 1 to number-of-subdomains
  Select subdomain (i)
  Grab data required from database for other subdomain solutions which share mesh space
  with subdomain (i)
  Solve for specified 'cocktail' of physics
10 Store solution to database.
```

The above 'multi-physics' solution strategy does at least mean that, although a domain may consist of a number of solution sub-domains (which may overlap or be subsets), each specified sub-domain has a prescribed 'cocktail' of physics. However, the sub-domains may be dynamic – they may grow or diminish, depending upon the model behaviour, as for example in the case of shape casting, where the flow domain diminishes as the structural domain grows.

Hence, for any parallelisation strategy for closely coupled multi-physics simulation to be effective, it must be able to cope with multiple sub-domains which

- a) have a specified 'cocktail' of physics that may be transiently non-homogeneous, and
- b) can change size dynamically during the calculation.

Within a single domain, of course, characteristic a) is not uncommon for CFD codes which have developing physics – a shock wave or combustion front, for example. It is the multi-phase nature of the problem which is a specific challenge.

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.
- b) the responsibility for determining a mesh partition to give a high quality load balance devolves to the load balance tool.

This strategy simplifies the code conversion task, but it requires a mesh partitioning tool that has the following capabilities:

- produces load balanced partition for a single (possibly discontinuous) sub-domain with a non-homogeneous workload per node or element,
- structures the sub-domain partitions so that they minimise inter-processor communication (e.g. the partitions respect the geography of the problem),
- dynamically re-load balances as the physics changes, and
- also tries to structure the partitions so that the usual inter-processor communication costs are minimised.

Of course, this problem is very challenging, as Hendrickson and Devine<sup>(9)</sup> point out in their review of mesh partitioning/dynamic load balancing techniques, from a computational mechanics perspective. From this review, currently it would appear that the only tools with the potential capability to address this conflicting set of challenges are parMETIS<sup>(10,11)</sup> and JOSTLE<sup>(12,13)</sup>. These tools follow a broadly similar strategy – they are implemented in parallel, are multi-level (in that they operate on contractions of the full graph) and employ a range of optimisation heuristics to produce well balanced partitions of weighted graphs against a number of constraints. The tool exploited in this work is JOSTLE<sup>(12)</sup>.

#### **4. PARALLEL PHYSICA – THE TARGET APPLICATION**

For the last decade, a group at Greenwich has focused upon the development of a consistent family of numerical procedures to facilitate multi-physics modelling<sup>(4,8)</sup>. These procedures are based upon a finite volume approach, but use an unstructured mesh. They now form the basis of a single software framework for closely coupled multi-physics simulation and enable the following key characteristics to be implemented:

- one consistent mesh for all phenomena,
- a measure of compatibility in the solution approaches to each of the phenomena,
- a single database and memory map so that no data transfer is involved and efficient memory use is facilitated between specific ‘physics’ modules,
- accurate exchange of boundary data and volume sources amongst phenomena.

This has been achieved in the PHYSICA framework which:

- uses a finite volume discretisation approach on a 3D unstructured mesh for tetrahedral, wedge and hexahedral elements,
- has fluid flow and heat transfer solvers based on an extension of the SIMPLE strategy, but using a cell centred discretisation with Rhie-Chow interpolation,
- uses melting/solidification solvers based on the approach of Voller and co-workers,
- exploits solid mechanics solution procedure based upon a vertex centred approximation with an iterative solution procedure and non-linear functionality.

This software has been used to model a wide range of materials and metals processing operations.

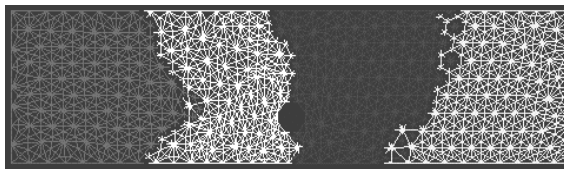
The parallelisation approach described in section 2 above has been incorporated within the PHYSICA software. This software has ~90000 lines of FORTRAN code and so the task of converting the code is immense. However, this has been considerably simplified by employing the strategy exemplified above.

A key issue has been the use of primary and secondary partitions to cope with distinct discretisation techniques – flow, etc. are cell centred and the solid mechanics is vertex centre based. Using these distinct partitions streamlines the task of code transformation, but leaves a debt for the partitioner/load balancing tool to ensure that the partitions are structured to maximise the co-location of neighbouring nodes and cells on the same processor. Some considerable attention was given to these issues in the last PCFD conference<sup>(16)</sup> and this functionality has now been implemented into JOSTLE<sup>(12)</sup> – see the example in Figure 2. The approach is straight forward:

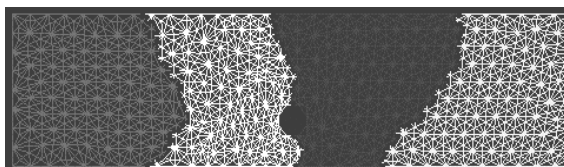
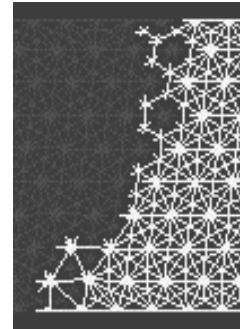
- a mesh entity type (e.g. cell) associated with the greatest workloads is selected,
- primary and secondary partitions are generated which both load balance and minimise penetration depth into neighbouring partitions,

though it should be noted that:

- an in-built overhead is associated with the additional information required for secondary partitions.



Poor secondary partition



Good partition from JOSTLE

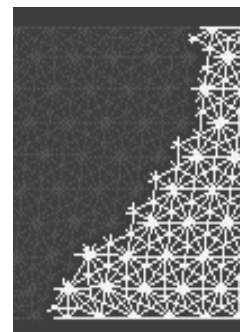


Figure 2 - Poor and good secondary partitions for two phase calculations; the good secondary partitions are generated by JOSTLE

The upper partition in Figure 2 is poor because, although it is load balanced, it causes rather more communication volume than is required in the lower partition. The impact of a

good secondary partitioning which is both load balanced and minimises penetration depth is illustrated in Figure 3, using the PHYSICA code on a CRAYT3E for a modest multi-physics problem with an ~8000 node mesh.

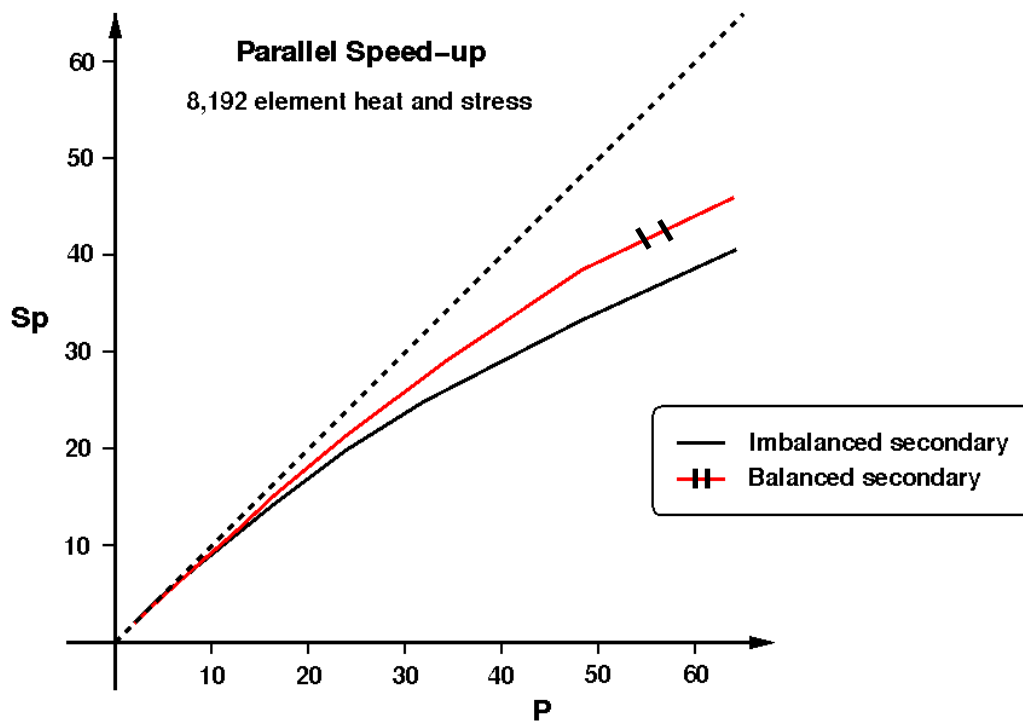
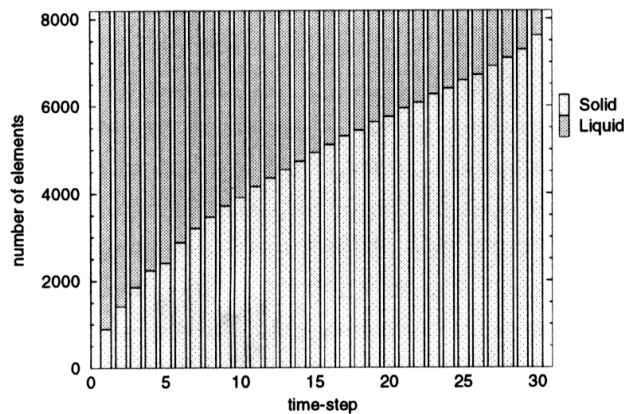


Figure 3 - Speed ups for a multi-physics calculation on a CRAY-T3E showing the impact of balanced secondary partitions

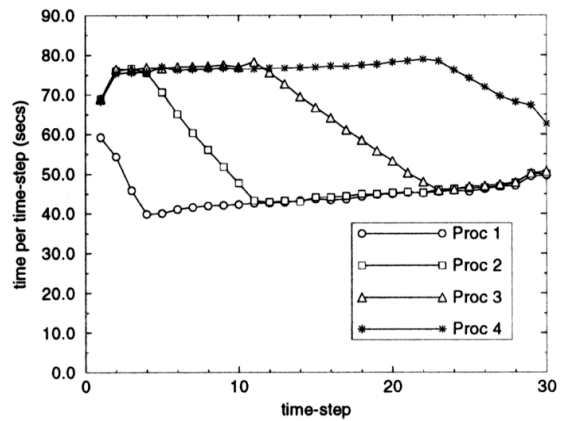
A final challenge to address here is the issue of dynamic load balancing – there are two key factors:

- the first occurs when the phenomena to be solved changes – as in the case of solidification where metal changes from the fluid to the solid state and activates different solvers, and
- the second occurs when the phenomena becomes more complex (and so computationally demanding) as, for example, where a solid mechanics analysis moves from elastic into visco-plastic mode for part of its sub-domain.

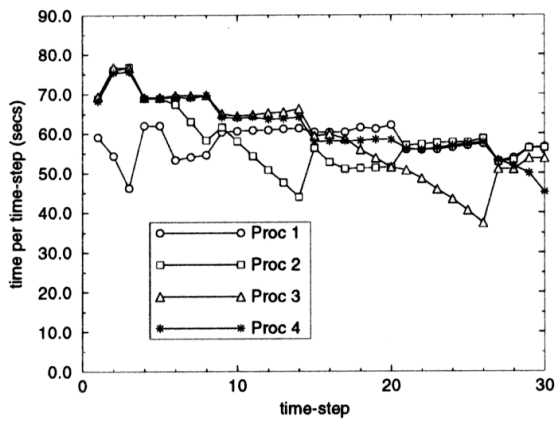
In the first case, this is addressed through a redefinition of the physics specific sub-domains and then re-load balanced. In the second case, the overall problem is re-load balanced because of the non-homogeneous dynamically varying workload in one sub-domain. A strategy to address these issues has been proposed and evaluated by Arulananthan et al.<sup>(16)</sup>. Figure 4 shows the effect of this dynamic load balancing strategy on the PHYSICA code.



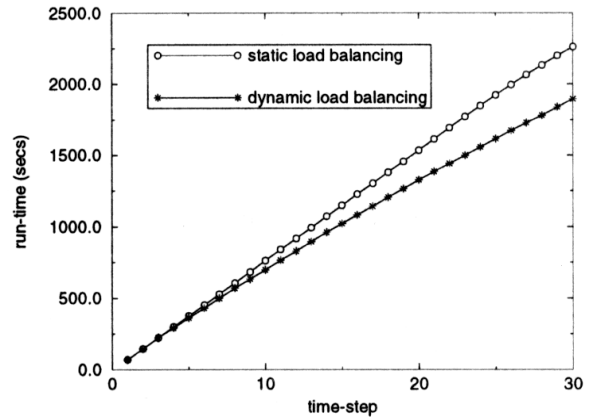
a) solidification of cooling bar



b) time per time-step without DLB



c) Time per time-step with DLB



d) Overall run time in parallel

Figure 4 - Sample results for dynamic load balancing tool running on Parallel PHYSICA

## 5. CONCLUSIONS

The move to multi-physics simulation brings its own challenges in the parallelisation of the associated software tools. In this paper we have highlighted the problems with regard to algorithmic structure and computational load balancing. Essentially, the multi-physics problem on a large domain mesh may be perceived as a series of sub-domains (and their meshes), each of which is associated with a specific 'set of physics'. The sub-domains may be distinct, overlap, or even be subsets of each other. What this means is that the code parallelisation tool can proceed on the basis of the large single mesh, with the partitioning/load balancing problem entirely devolved to the mesh partitioning and load



balancing tools. However, these tools need a range of optimisation techniques that can address non-homogeneous sub-domains which may vary in size dynamically. They must be able to produce high quality secondary as well as primary partitions and run in parallel, as well as deliver the usual array of performance characteristics (such as minimising data movement amongst processors).

In the work described here, the implementation focus has involved the multi-physics simulation tool, PHYSICA, and the mesh partitioning/dynamic load balancing tool, JOSTLE. Through these tools a range of challenges have been addressed that should lead to scalable performance for strongly inhomogeneous mesh based multi-physics applications, of which CFD is but one component.

## 6. REFERENCES

1. ANSYS, see <http://www.ansys.com>
2. ADINA, see <http://www.adina.com>
3. SPECTRUM, see <http://www.centric.com>
4. PHYSICA, see <http://physica.gre.ac.uk>
5. TELLURIDE, see [http://lune\\_mst.laul.gov/telluride](http://lune_mst.laul.gov/telluride)
6. C Farhat, M Lesoinne and N Maman, Mixed explicit/implicit time integration of coupled aeroelastic problems: Three field formulation, geometric conservation and distributed solution. *International Journal of Numerical Methods Fluids*, Vol. 21, 807-835 (1995).
7. A K Slone et al. Dynamic fluid-structure interactions using finite volume unstructured mesh procedures, in *CEAS Proceedings*, Vol. 2, 417-424 (1997).
8. M Cross, Computational issues in the modelling of materials based manufacturing processes. *Journal of Comp. Aided Mats Des.* 3, 100-116 (1996).
9. B Hendrickson and K Devine, Dynamic load balancing in computational mechanics. *Comp. Meth. Appl. Mech. Engg.* (in press).
10. G. Karypis and V Kumar, Multi-level algorithms for multi-constraint graph partitioning, University of Minnesota, Dept of Computer Science. Tech. Report 98-019 (1998).
11. ParMETIS, see <http://www.cs.umn.edu/~metis>
12. JOSTLE, see <http://www.gre.ac.uk/jostle>
13. C Walshaw and M Cross, Parallel optimisation algorithms for multi-level mesh partitioning, *Parallel Computing* (in press).
14. K McManus et al., Partition alignment in three dimensional unstructured mesh multi-physics modelling, *Parallel Computational Fluid Dynamics – Development and Applications of Parallel Technology* (Eds C A Lin et al.). Pub. Elsevier, 459-466 (1999).
15. C Walshaw, M Cross and K McManus, Multi-phase mesh partitioning. (in preparation).

16. A Arulananthan et al. A generic strategy for dynamic load balancing of distributed memory parallel computational mechanics using unstructured meshed. *Parallel Computational Fluid Dynamics – Development and Application of Parallel Technology* (Eds C A Lin et al.). Pub. Elsevier, 43-50 (1999).