

Finite Volume Methodology for Contact Problems of Linear Elastic Solids

H. Jasak *

*Computational Dynamics Ltd.
Hythe House
200 Shepherds Bush Road
London W6 7NY, England
E-mail: h.jasak@cd.co.uk*

H.G. Weller

*Department of Mechanical Engineering
Imperial College of Science,
Technology and Medicine
Exhibition Road
London SW7 2BX, England*

Abstract

This paper presents the coupling algorithm for contact stress problems of linear elastic solids using the Finite Volume Method. The method applies second-order accurate discretisation and unstructured meshes in a segregated framework with explicit update of the contact condition, which allows for geometric flexibility and efficient treatment of non-linearity. Special attention is given to contact detection and interpolation algorithms on arbitrary polygonal patches. The method is tested on a 2-D contact problem of a curved and a planar surface.

1 Introduction

The driving force behind the development of segregated Finite Volume (FV) stress analysis algorithms is the potential for dealing with non-linear problems with only a marginal increase in computational cost. This class of algorithms originates from Computational Fluid Dynamics (CFD) and has several characteristics not typically associated with the more popular Finite Element (FE) algorithms. The key features of the solution method described here are similar to the well-established fluid-flow FV solvers.

- The computational domain is discretised in an arbitrarily unstructured manner, where the Control Volumes (CV) can be arbitrary polyhedra in 3-D. The “shape function” used in equation discretisation is independent of the shape of the CV and does not need to be known in advance, providing high flexibility in mesh

*Corresponding author, Tel: +44 (0)20 7471 6200, Fax: +44 (0)20 7471 6201

generation. The vector properties are defined in the global Cartesian coordinate system and the solution points are located in the centroids of the polyhedra (colocated variable arrangement).

- The equations are discretised in the integral form over the CV and solved in a segregated manner, where each component of a vector and/or any additional transport equations are solved sequentially and the cross-coupling is lagged. Discretisation is second-order accurate in space, which is achieved by prescribing a linear variation of the variable across the CV and creates sparse diagonally dominant matrices which are largely identical between the components of the displacement vector.
- Equation segregation allows us to use memory-efficient iterative solvers and partial convergence. The intermediate solution is used to update the cross-coupling terms, as well as any non-linear interaction. The problem is solved when the residual on all equations drops below a prescribed level.

As a consequence of equation segregation, even a linear model, such as that for a linear elastic solid, is effectively “non-linearised” within the solution procedure. Previous studies [1–3] show that a combination of smaller matrices, iterative solvers and partial convergence sufficiently compensates for this effect. The advantage of segregation becomes obvious when the problem under consideration becomes non-linear, as is the case here: linearisation and explicit updates naturally slot into the algorithm, which is most definitely not the case for linear block solvers typically encountered in the Finite Element Method (FEM).

This paper describes the solution method for contact problems in linear elasticity as the simplest example of non-linearity introduced through boundary conditions. As the basic linear elastic solver has been described elsewhere [3], we shall concentrate on problems associated with contact boundaries. The objective of this study is to lay a foundation for the FV contact stress solver, which will subsequently be extended to include other non-linear phenomena: large deformations and moving meshes, non-linear constitutive laws and crack propagation. The use of unstructured meshes and consistency with the rest of the setup is paramount – in some cases, simpler practices may exist, but precedence is given to more general, albeit less accurate algorithms, which fit into the unstructured mesh framework.

The rest of the paper is organised as follows: Section 2 summarises the mathematical model, equation segregation and boundary conditions, which is followed by the description of the contact boundary treatment in Section 3. Our main concern is the accurate detection of contact area between two polygonal surfaces and an appropriate interpolation procedure. The algorithm is tested on a simple 2-D test case in Section 4 and the paper is closed with a short summary.

2 Summary of Governing Equations

The steady-state force balance for a solid body element in its differential form states:

$$-\nabla \cdot \boldsymbol{\sigma} = \rho \mathbf{f}, \quad (1)$$

where ρ is the density, \mathbf{f} is the body force and $\boldsymbol{\sigma}$ is the stress tensor. The strain tensor $\boldsymbol{\varepsilon}$ is defined in terms of the displacement vector \mathbf{u} :

$$\boldsymbol{\varepsilon} = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T] \quad (2)$$

and the Hooke's law is used to close the system:

$$\boldsymbol{\sigma} = 2\mu\boldsymbol{\varepsilon} + \lambda \text{tr}(\boldsymbol{\varepsilon}) \mathbf{I}. \quad (3)$$

Here, \mathbf{I} is the unit tensor and μ and λ are the Lamé's coefficients, relating to the Young's modulus of elasticity E and the Poisson's ratio ν as:

$$\mu = \frac{E}{2(1+\nu)}, \quad \lambda = \begin{cases} \frac{\nu E}{(1+\nu)(1-\nu)} & \text{plane stress} \\ \frac{\nu E}{(1+\nu)(1-2\nu)} & \text{plane strain and 3-D.} \end{cases} \quad (4)$$

For convenience, the governing equation is rewritten with \mathbf{u} as the primitive variable:

$$-\nabla \cdot [\mu \nabla \mathbf{u} + \mu (\nabla \mathbf{u})^T] - \nabla (\lambda \nabla \cdot \mathbf{u}) = \rho \mathbf{f}. \quad (5)$$

Efficient segregated discretisation [3] is achieved by decomposing Eqn. (5) into maximum implicit coupling and pure rotation, which is lagged:

$$-\underbrace{\nabla \cdot [(2\mu + \lambda) \nabla \mathbf{u}]}_{\text{implicit}} - \underbrace{\nabla \cdot [\mu (\nabla \mathbf{u})^T - (\mu + \lambda) \nabla \mathbf{u} + \lambda \mathbf{I} \text{tr}(\nabla \mathbf{u})]}_{\text{explicit}} = \rho \mathbf{f}. \quad (6)$$

Boundary conditions used in this study can be classified as follows:

1. Fixed displacement boundaries, where all components of \mathbf{u} are specified,
2. Traction boundaries, where the surface traction $\mathbf{t} = \hat{\mathbf{n}} \cdot \boldsymbol{\sigma}$ is given. This can include the specification of both the pressure p and a tangential force (eg. friction); a special case of this condition is a free surface ($\mathbf{t} = 0$). In numerical terms, a traction boundary specifies a fixed gradient on \mathbf{u} :

$$\hat{\mathbf{n}} \cdot \nabla \mathbf{u} = \frac{\mathbf{t} - \hat{\mathbf{n}} \cdot [\mu (\nabla \mathbf{u})^T - (\mu + \lambda) \nabla \mathbf{u}] - \hat{\mathbf{n}} \lambda \nabla \cdot \mathbf{u}}{2\mu + \lambda}, \quad (7)$$

where $\hat{\mathbf{n}}$ is the outward-pointing unit normal.

3. Planes of symmetry are discretised by imagining a "ghost" CV next to the symmetry plane as a mirror image of the solution inside.

4. A contact surface represents a special type of boundary condition, where two situations can occur. In the region of contact, the normal component of traction is transferred and the relative displacement is adjusted to reflect the force balance; for non-zero friction, the additional friction force is calculated from the relative motion of the two sides and the contact pressure. Outside of contact, the boundary is treated as a free surface.

3 Treatment of Contact Boundaries

The complexity of the contact boundary treatment is two-fold. Firstly, the boundary condition on the contact surface is a combination of the Dirichlet condition for the normal component of \mathbf{u} and the fixed gradient condition for the tangential component in contact, and a fixed gradient condition outside of contact. Secondly, the extent of the contact is not given and represents a part of the solution: it is calculated from the relative displacement which in turn depends on the force balance. The non-linearity of the system is caused by the fact that the boundary condition depends on the solution itself.

Mixed fixed value – fixed gradient conditions are not unusual in the FVM and their implementation poses no major problems. The only curiosity here is the fact that the split is based on the local normal and further complicated by the possibility of partial contact. This “direction-mixed” boundary condition is therefore specified in three parts: the “boundary value”, the normal component of which is prescribed for the contact part, the “boundary gradient”, specifying the gradient condition and the “value fraction”, indicating which part of the surface is in contact.

The case setup for a contact problem specifies the pair of potential contact surfaces. Numerical stability is achieved by prescribing the direction-mixed condition on one side and the fixed gradient condition on the other; the two boundary conditions are updated in unison.

Typically, the iterative procedure causes the two sides to overlap, triggering the boundary condition update. There remains to be established which part of the surface overlaps, based on the geometry and the current displacement field. In the simplest case, the number of faces on both surfaces is identical and the contact area could be calculated from the relative displacement and the initial distance. This is, however, unsatisfactory for two reasons: it imposes a strict limitation on mesh generation and returns a binary result, *i.e.* a partial contact of two faces cannot be detected. We are therefore forced to adopt a more general approach.

Consider a pair of surfaces in partial contact, Fig. 1. In the first phase, an “equivalent” point is found for each vertex of side A, *eg.* P and Q . The dot-product of \overline{PQ} and the local normal \mathbf{n}_P gives the indication and contact distance for every vertex of A.

Assembly of equivalent point pairs can be done in several ways, the simplest of which is normal projection, Fig. 1. In order to improve the monotonicity of interpolation for curved surfaces with variable gaps, the implemented method constructs

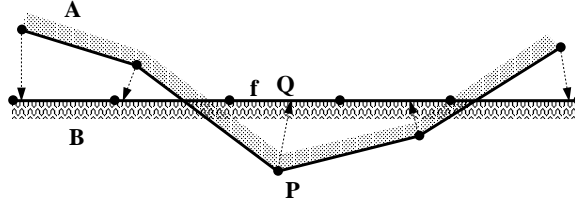


Figure 1: Surfaces in contact.

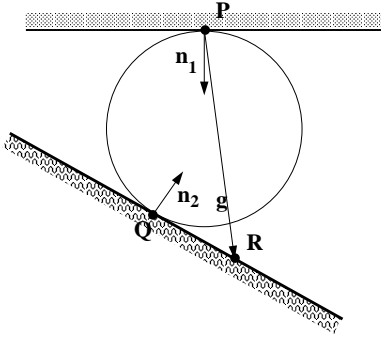


Figure 2: Contact sphere.

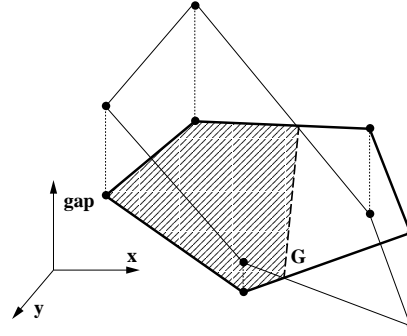


Figure 3: Polygonal area in contact.

the sphere of minimum radius which passes through P and is tangential to both surfaces, Fig. 2. For a point P , the unit normal \mathbf{n}_1 and the opposite surface defined by an arbitrary point R and unit normal \mathbf{n}_2 , the radius of the sphere is given as:

$$r = \frac{\mathbf{g} \cdot \hat{\mathbf{n}}_2}{\hat{\mathbf{n}}_1 \cdot \hat{\mathbf{n}}_2 - 1}, \quad (8)$$

where $\mathbf{g} = \overline{PR}$. The equivalent point on the opposite surface is:

$$\mathbf{Q} = \mathbf{P} + r(\hat{\mathbf{n}}_1 - \hat{\mathbf{n}}_2). \quad (9)$$

Negative r denotes overlapping surfaces.

It finally remains to determine the contact area for a general polygon, given the surface gap in all its vertices, Fig. 3. The contact area fraction is the area ratio of the “contact polygon” (shaded) and the complete face. If all the gaps are of the same sign, no further calculation is necessary. For partial contact, a sub-polygon is constructed from all vertices with negative gap values and edge intersections where the gap changes the sign along the edge, *eg.* G in Fig. 3. The surface area of both polygons is calculated using triangular decomposition.

The remaining problem is the transfer of face-based data between the patches. For generality, we shall use a two-step inverse distance weighting, where the data is first interpolated into the vertices of the master patch, then transferred via the equivalent

point pairs onto the slave vertices, where the face values are reassembled. The area in contact is calculated separately for each direction. Although this practice potentially introduces an inconsistency in contact area, it is more general and the level of distortion is acceptable, as will be illustrated below.

4 Numerical Example

The numerical example represents the contact of a cylindrical segment with a plane, schematically shown in Fig. 4.

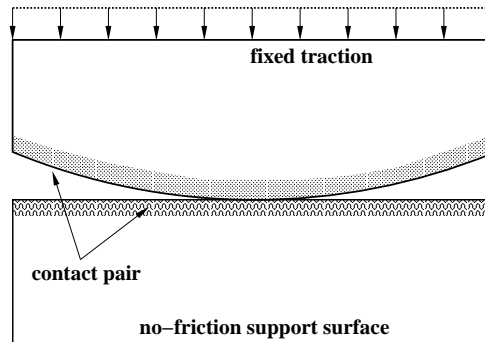


Figure 4: Cylindrical contact surface test setup.

A small domain has been chosen deliberately to capture the effects of finite dimensions; a comparison with the analytical solution for a larger domain will be presented elsewhere. Calculations are performed using the Field Operation and Manipulation (FOAM) C++ Computational Continuum Mechanics library [4], developed by the authors.

The problem is solved on two meshes with and without the mismatch on the contact surface. The result for the mesh with 2400 CVs is shown in Fig. 5. The most notable consistency check is the continuity of the σ_{yy} iso-lines across the contact boundary, which is satisfactory. Also, full symmetry of the solution is preserved although it was not enforced. The calculation converged by six orders of magnitude in 241 iterations, or 24.7 s on an SGI Origin 200 workstation when boundaries were matched and 329 iterations (41.2 s) for mismatched boundaries.

Distribution of the normal stress along the contact surface is given in Fig. 6 and shows adequate accuracy of force transfer. The wiggles caused by interpolation for mismatched boundaries are relatively small and the size of contact surface is realistic when compared with the analytical solution for a large domain (5.5 mm vs. 5.7 mm).

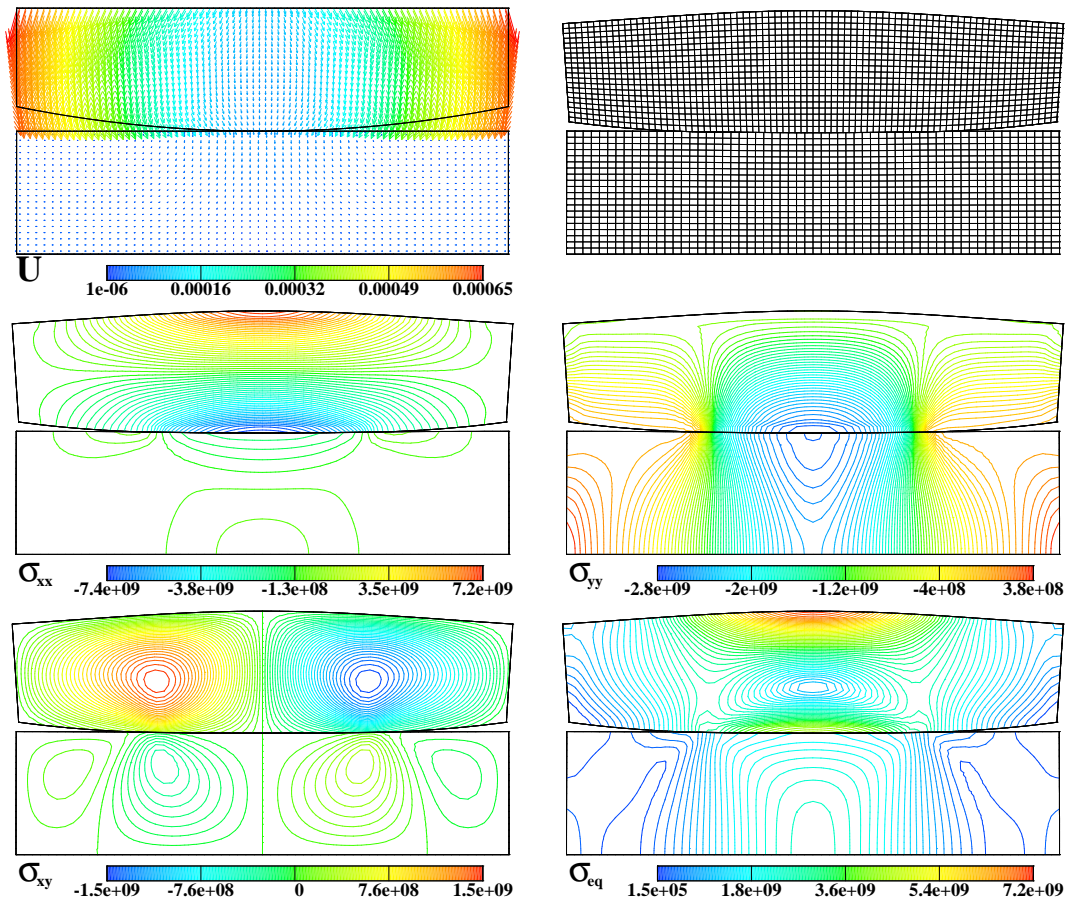


Figure 5: Cylindrical body in contact: solution.

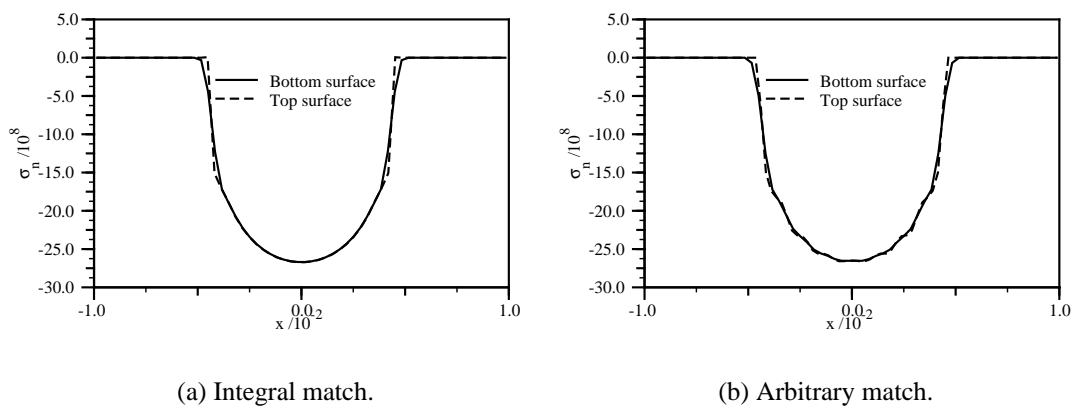


Figure 6: Normal stress on contact surface.

5 Summary and Future Work

This paper presents the solution algorithm for the contact problem of linear elastic solids in the framework of the second-order Finite Volume discretisation. The method relies on equation segregation and iterative solvers, which allows for natural treatment of non-linearities. The performance of the method is considered satisfactory, both in terms of accuracy and computational efficiency. In future, the method will be extended to deal with non-zero friction and large deformations.

Acknowledgement

The authors would like to thank Drs. C.J. Marooney and V. Tropsa for their contribution in the development of the methodology.

References

- [1] Demirdžić, I., Muzaferija, S., and Perić, M.: “Benchmark solutions of some structural analysis problems using finite-volume method and multigrid acceleration”, *Int. J. Numer. Meth. Engineering*, 40(10):1893–1908, 1997.
- [2] Demirdžić, I., Muzaferija, S., and Perić, M.: “Advances in computation of heat transfer, fluid flow and solid body deformation using finite volume approaches”, In Minkowycz, W.J. and Sparrow, E.M., editors, *Advances in Numerical Heat Transfer*, volume 1, chapter 2. Taylor & Francis, 1997.
- [3] Jasak, H. and Weller, H.G.: “Application of the Finite Volume Method and Unstructured Meshes to Linear Elasticity”, *Comp. Meth. Appl. Mech. Engineering*, 2000: in print.
- [4] Weller, H.G., Tabor, G., Jasak, H., and Fureby, C.: “A tensorial approach to computational continuum mechanics using object orientated techniques”, *Computers in Physics*, 12(6):620 – 631, 1998.