

Hybrid finite element–finite volume discretization of complex geologic structures and a new simulation workflow demonstrated on fractured rocks

A. PALUSZNY¹, S. K. MATTHÄI¹ AND M. HOHMEYER²

¹*Department of Earth Sciences and Engineering, Imperial College London, South Kensington Campus, London, UK;*

²*Ansys Corporation, Berkeley, CA, USA*

ABSTRACT

The generation of computational meshes of complex geological objects is a challenge: their shape needs to be retained, resolution has to adapt to local detail, and variations of material properties in the objects have to be represented. Also mesh refinement and adaptation must be sufficient to resolve variations in the computed variable(s). Here, we present an unstructured hybrid finite element, node-centred finite-volume discretization suitable for solving fluid flow, reactive transport, and mechanical partial differential equations on a complex geometry with inhomogeneous material domains. We show that resulting meshes accurately capture free-form material interfaces as defined by non-uniform rational B-spline curves and surfaces. The mesh discretization error is analysed for the elliptic pressure equation and an error metric is introduced to guide mesh refinement. Finite elements and finite volumes are represented in parametric space and integrations are conducted numerically. Subsequently, integral properties are mapped to physical space using Jacobian transformations. This method even retains its validity when the mesh is deformed. The resulting generic formulation is demonstrated for a transport calculation performed on a complex discrete fracture model.

Key words: discretization, FE-FVM, finite elements, finite volumes, fluid flow, fractures, meshing, modelling, NURBS, simulation

Received 9 September 2006; accepted 8 February 2007

Corresponding author: Adriana Paluszny, Department of Earth Sciences and Engineering, Imperial College London, South Kensington Campus, London SW72AZ, UK.

Email: a.paluszny-rodriguez@imperial.ac.uk. Tel: +44 20 759 41350. Fax: +44 20 7594 7444.

Geofluids (2007) 7, 186–208

INTRODUCTION

Recent advances in computer software and hardware capabilities allow hydro-mechanical computations with an unprecedented degree of physical realism. The definition of model geometry in three dimensions remains a challenge. Due to the irregular shape of material domains and their internal heterogeneity, the governing partial differential equations cannot be solved analytically and numerical methods are used to find approximate solutions. These methods approximate unknown functions of multiple variables by integration of simpler sub-equations for the coupling among discrete points. The discretization derives from the spatial subdivision of the domain of interest into a mesh of geometrically primitive cells. Contributions from

the sub-equations combine into a system of algebraic equations constraining the domain solution.

The first step of any simulation workflow is the adequate geometrical representation of the model. The introduction of graphical tools suitable for the geologic interpretation of seismic data, maps, and cross-sections from mines has facilitated the generation of rigorous three-dimensional structural geologic models. The migration of these complex interpretations into internally consistent simulation models, without loss of crucial geological detail, constitutes an important new field of applied research. The representation of faults and fractures is especially challenging as they have a large aspect ratio, displace layers relative to one-another, and display complex cross-cutting relationships. Figure 1A shows an example of a complexly fractured fold exposed at

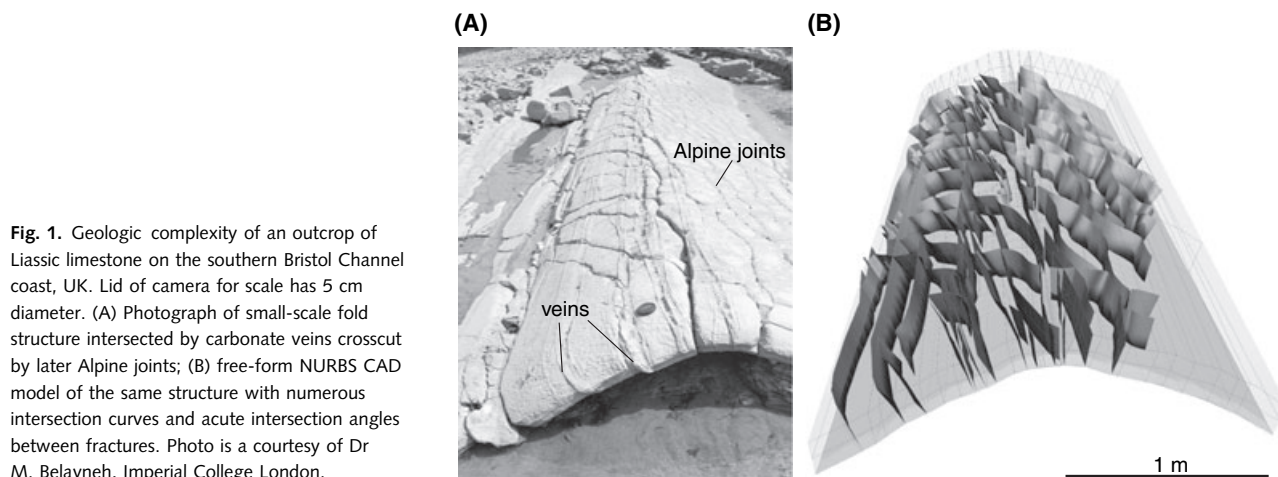


Fig. 1. Geologic complexity of an outcrop of Liassic limestone on the southern Bristol Channel coast, UK. Lid of camera for scale has 5 cm diameter. (A) Photograph of small-scale fold structure intersected by carbonate veins crosscut by later Alpine joints; (B) free-form NURBS CAD model of the same structure with numerous intersection curves and acute intersection angles between fractures. Photo is a courtesy of Dr M. Belayneh, Imperial College London.

Kilve beach, UK. Typically, fractures have a variable aperture, intersect at small angles, and range in size over several orders of magnitude.

Commonly used polyhedral representations of geological models are sufficient for visual display, but unnecessarily constrain volume discretization. Severe discretization problems arise at the intersections of multiple polyhedral surfaces (Bogdanov *et al.* 2003). In contrast to differentiable curves and surfaces, polygons and polyhedra force meshing tools to honour predefined node points. This constrains model resolution. Additionally, such models are often characterized by a lack of topological information: geometric entities are not properly connected, intersection curves are often absent, and models contain occluded gaps, holes and overlaps. Meshing algorithms, however, require water-tight models with internally consistent space subdivisions. For these a mesh can be generated, and simulation may begin.

In this paper, we will introduce a new rigorous workflow for the simulation of flow and transport in deformable fractured rocks using a hybrid finite element (FEM) finite volume method (FVM). The three obstacles to overcome before this FE-FVM can be applied are: (1) representation of the geometry by an internally consistent computer-aided design (CAD) model; (2) efficient, at least semi-automatic, finite element discretization of this model; and (3) construction of a complementary and internally consistent finite-volume mesh.

A growing body of simulation results highlights the potential of unstructured mesh-based simulations for understanding subsurface processes (Helmig 1997; Kim & Deo 2000; Bogdanov *et al.* 2003; Karimi-Fard *et al.* 2004; Matthäi *et al.* 2005a). Finite element methods dominate heat transfer, mechanical and fluid dynamics modelling (e.g. Chung 2002). They operate on unstructured meshes by default. Finite-volume methods are applied to solve transport problems on both regular grids and unstructured meshes. Geometrically complex fluid flow problems have

been solved by the FE-FVM, in an operator splitting approach where the elliptic parts of a partial differential equation are solved with the FEM and the hyperbolic ones with FVM (Schneider *et al.* 1992; Durlofsky 1994; Helmig 1997; Bogdanov *et al.* 2003; Karimi-Fard *et al.* 2004; Leveque 2003; Matthäi *et al.* 2005a; Reichenberger *et al.* 2006). FE and FV methods have much in common: they are integral formulations, use shape functions for interpolation and are applicable to unstructured meshes. Our paper contributes a suite of new generic finite element-based finite-volume stencils defined in parametric space for tetrahedral, prism, pyramid, hexahedral, quadrilateral, triangular and bar elements. A representation of the finite-volume mesh in physical space is not required. Jacobian transformations are used to map properties from parametric to physical space. Well established integration techniques are employed for the efficient processing of the stencils. The FE-FVM derived and employed in this paper is an extension of Baliga & Patankar's (1980) original formulation for the solution of the advection–diffusion equation. This method is embedded into a novel simulation workflow that begins with CAD modelling and ends with the efficient numeric solution of the governing equations via algebraic multigrid methods (Ruge & Stüben 1987; Stüben 2001).

METHODS

Modelling and discretization of geological structures are intricate procedures by which approximations and simplifications are introduced (Ewing & Spagnuolo 2003). Power-law fracture aperture and length distributions (e.g. Segall & Pollard 1983), complicated cross-cutting relationships, and curved layer boundaries require a spatially variable model resolution. To assign rock properties material domains are necessary. The first part of this method section addresses such model building issues. Once the model has been constructed, numerical solution methods for the

governing equations demand its spatial discretization. This is the subject of the second part of this section. We first discuss the generation of a static mesh concerned primarily with the model's accurate geometric representation. Then we introduce the finite element–finite volume stencils and apply them to the first-order discretization of the advection–diffusion equation.

Individual steps of our simulation workflow facilitated by these methods are summarized as follows:

1. Representation of the geologic inventory by non-uniform rational B-splines (NURBS). This boundary representation (BREP) is explained further below. Fractures and faults are represented using equal (3D) or lower dimensional (2D embedded in 3D) geometrical BREP approximations.
2. Discretization of the model with a hybrid finite element mesh composed of triangles, quadrilaterals, tetrahedra, hexahedra, prisms and pyramids. High aspect ratio features like thin layers, faults or fractures are discretized with large aspect ratio prism and hexahedral elements.
3. Spatial discretization of the governing equations using the FE-FVM. Formulation of the finite element finite-volume stencils. Temporal discretization via finite differences.
4. Trial simulation.
5. Adaptation of the geometrically conforming mesh to the shape of the solution of the governing equation(s), such that it can adequately represent variations in the value of the computed variable(s).

Topological information, such as the grouping and facing directions of surfaces delimiting a volume, is mapped onto the mesh and retained throughout the simulation process. Material domains persist through the calculations and can be visualized individually to gain insights into the behaviour of the system.

Step 1: Building the geometrical model with CAD

Geologic modelling relies on accurate representations of the subsurface at various levels of detail, including features that may differ in size over several orders of magnitude. For instance, a sand-shale sequence may contain hundreds of large aspect ratio layers truncated and offset along fault discontinuities. This calls for a framework that supports scalable geometrical entities such as parametric curves and surfaces (De Boor 1978).

For 40 years, smooth parametric curves and surfaces have been used to define boundaries of free-form objects in CAD (Farin 2001). NURBS curves and surfaces allow accurate representation of simple polylines, conic sections and free-form objects (Farin 2001). Storage requirements are modest and NURBS allow local control during generation and modification. This means that point editing only affects the immediate neighbourhood of a point so that surfaces which simultaneously match multiple boundaries can still be reshaped. Therefore, NURBS meet the two most important criteria of geometrical representation: flexibility – the ability to initially represent a feature and fidelity – the ability to represent the feature throughout modifications (Ellens 1997). NURBS are able to represent geological objects with a tolerance-based spatially adaptive level of detail, independent of scale. In contrast to faceted representations, they do not prescribe a specific resolution (Fig. 2).

Faceted representations based on polylines and surface triangulations are flexible and easy to use, but they introduce discontinuities into the model that are not present in the original geometry, see Fig. 2. By contrast, NURBS are differentiable, smoothly representing shapes as continuous features without the polyline disadvantages. Meshing of fracture polygons, for example, will produce an element

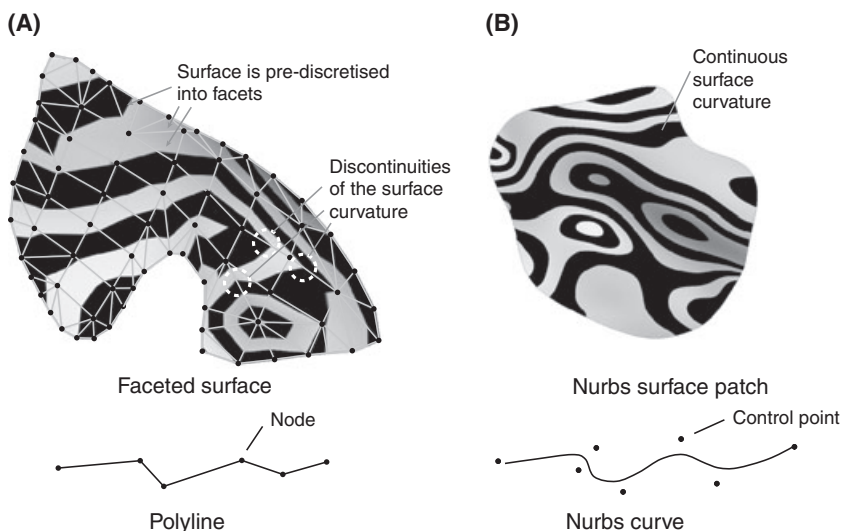


Fig. 2. Faceted versus NURBS representation of a curved surface in full and in cross-sectional view. (A) Polyhedral surface composed of disjoint triangles and quadrilaterals and node points imposing a fixed resolution. The zebra stripes are a visualization technique for discontinuities (e.g. white dashed circles), showing that surface (A) is not differentiable. (B) Continuous differentiable NURBS surface without prescribed resolution; points in cross-section define the shape of this NURBS.

node at each vertex. With NURBS, however, a purpose-dependent adaptation of the mesh to the smooth tipline is possible.

Using NURBS, volumetric objects are defined by grouping curve-delimited surfaces together to define the boundary of a volume. The resulting BREP is a hierarchical tree structure of points (nodes), holes and surfaces (loops), and surface-enclosed volumes (bodies), recording their relations to each other. It defines the topology of each body, facilitating boolean operations, such as union, intersection and difference (Fig. 3).

Figure 1 illustrates the NURBS–BREP representation of a small fold with axis parallel veins being cross-cut by later joints within an analogue of a fractured hydrocarbon reservoir (Al-Mahruqi 2001; Belayneh & Cosgrove 2005). Fractures were traced with NURBS curves and extruded into surfaces orthogonal to the layering. Rock matrix blocks are delimited by NURBS surfaces forming a compo-

site boundary enclosing material domains. This set of objects represents the macroscopic geometry accurately.

Preparation and repair of CAD models for meshing is very time consuming. Volume overlaps make a unique discretization of space impossible such that labour intensive geometric model healing is required (Beall *et al.* 2003). Artefacts, including gaps, hamper the mesh generation process and must be avoided at the CAD model construction stage. BREP significantly reduces potential inconsistencies within a model and is an effective tool in producing boundary-conforming, watertight and topology compliant models (Requicha & Rossignac 1992; Beall & Shephard 1997; Caumon *et al.* 2004).

We advocate that CAD models are built already at the geological interpretation stage, geometry editing is standardized, and meshing is done directly on the CAD model, precluding errors introduced by its conversion into another format. NURBS and BREP are supported by many CAD packages.¹

Step 2: Generating the hybrid finite element mesh

CAD geometry can be discretized with three types of meshes: rectilinear, curvilinear and unstructured (Fig. 4A). Rectilinear grids have a fixed resolution. The refinement they require in order to track material interfaces that are not aligned with the coordinate axes is prohibitively high, especially if these interfaces are curved. Yet, their accurate resolution is very important because material properties, like the permeability, can vary by several orders of magnitude across them. Here, curvilinear and unstructured grids can provide the necessary adaptive resolution. Structured curvilinear grids, also known as O-grids, are able to capture free-form objects by mapping curves and surfaces to topologically cubic blocks in parametric space. However, even for geometrically simple models, this subdivision requires significant manual intervention. Therefore, curvilinear grids are ill-suited for the discretization of complex geological models (cf. Owen 1998).

We advocate unstructured grids because they can track free-form geometrical entities, such as NURBS with spatially variable refinement, and can also be generated automatically. The disadvantage of unstructured grids is that mesh coordinates cannot be calculated from indices. They therefore must be stored. However, in practice, this does not significantly increase memory requirements because the majority of storage is taken up by the discretized physical variables.

Traditional unstructured mesh generation approaches, such as Delaunay and advancing front, require water-tight geometric models because they anchor any volume on the

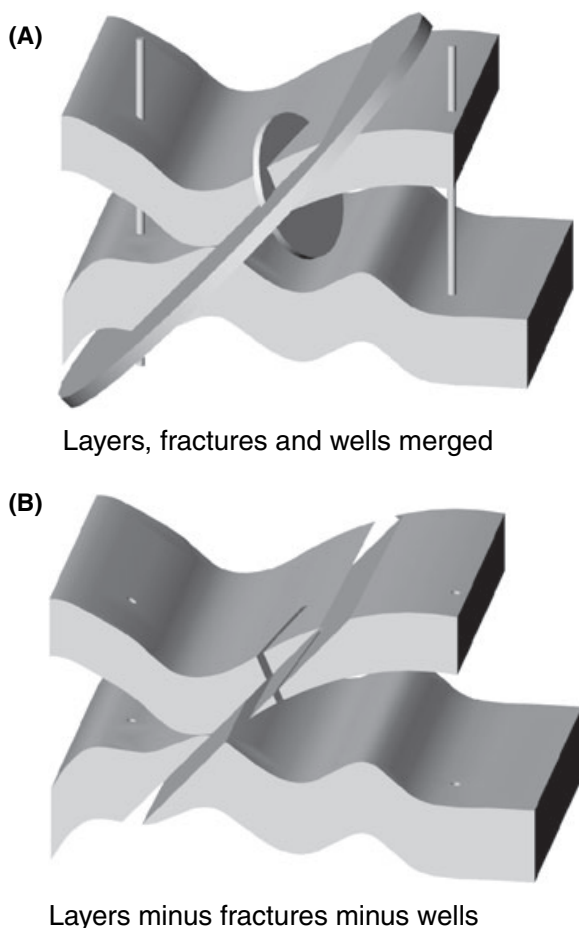


Fig. 3. Boolean operations applied to BREP solids illustrating union, difference and intersection. (A) Idealized model with two layers, faults and wells. (B) Result of applying the boolean difference between layers and wells and faults.

¹All free-form geometries presented in this paper were created with Rhinoceros v3.0, a NURBS BREP-based CAD geometric modelling tool.

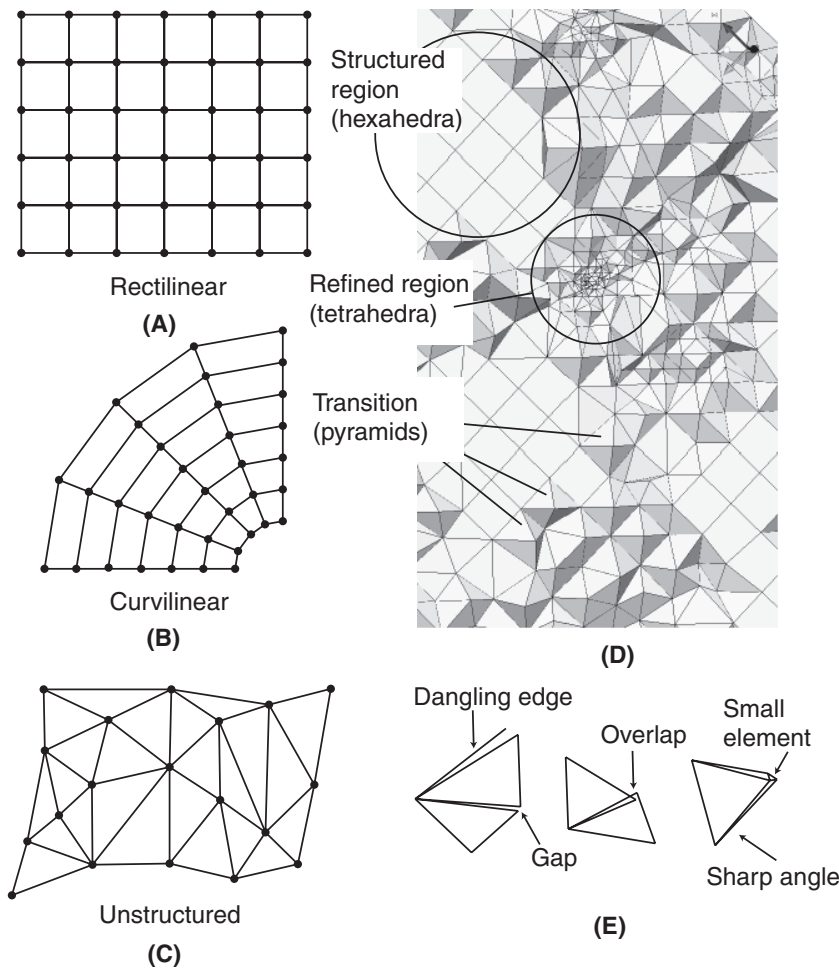


Fig. 4. Three common types of meshes: (A) rectilinear, (B) curvilinear and (C) unstructured. (D) Hybrid element mesh: featureless regions consist of hexahedra, constrained ones of tetrahedral, and transitions are covered by pyramid and prism elements interfacing tetrahedra with hexahedra.

surface mesh forming its boundary. Alternative octree-based mesh generation algorithms perform well on slightly imperfect models (McMorris & Kallinderis 1997). Factors which still compromise mesh quality are gaps and overlaps, faces with narrow angles or small areas, and dangling edges (Fig. 4E). We refer the interested reader to Owen (1998) and Lo (2002).

Pure hexahedral meshes tend to be more compact and have better computational properties than tetrahedral meshes of the same order (Benzley *et al.* 1995; Blacker 2000). However, automatic generation of pure hexahedral unstructured meshes for free-form geometry is extremely difficult. While a number of new robust algorithms for hexahedral meshing have been proposed (e.g. Owen 1998), these methods cannot handle arbitrary geometry without intensive manual intervention. As a good compromise between the complexity of all-hexahedral meshing and the simplicity of automatic tetrahedralization, hexahedra-dominated hybrid element meshes with tetrahedra, prisms and pyramids have become increasingly popular in the finite element community (Huber & Helmig 2000; Khawaja & Kallinderis 2000; Matthäi *et al.* 2005a;

Reichenberger *et al.* 2006). A typical hybrid mesh contains hexahedral elements in geometrically unconstrained regions. More shape-adaptive tetrahedral elements are used to capture geometric complexities and intentional refinement variations (Fig. 4D). Pyramid and prism elements are introduced to connect hexahedra with tetrahedra.

Hybrid element meshes can be generated automatically from tetrahedral meshes. This indirect approach starts with the generation of a pure tetrahedral mesh. It is partially converted to hexahedra by merging and splitting of elements and the introduction of prisms and pyramids (Zgainski *et al.* 1996; Thompson *et al.* 1998). To evaluate the quality of the resulting mesh, the element to node ratio can be used. For realistic hybrid meshes of free-form geometry we will show in the Results section that a ratio close to 2 can be obtained for hybrid meshes when compared with 5–6 for pure tetrahedral meshes (Bogdanov *et al.* 2003).

Fractures and faults in three-dimensional models may be represented either by surfaces (e.g. Juanes *et al.* 2002) or by volumes (e.g. Matthäi *et al.* 2005a). Our workflow supports both approaches. Using a lower dimensional representation,

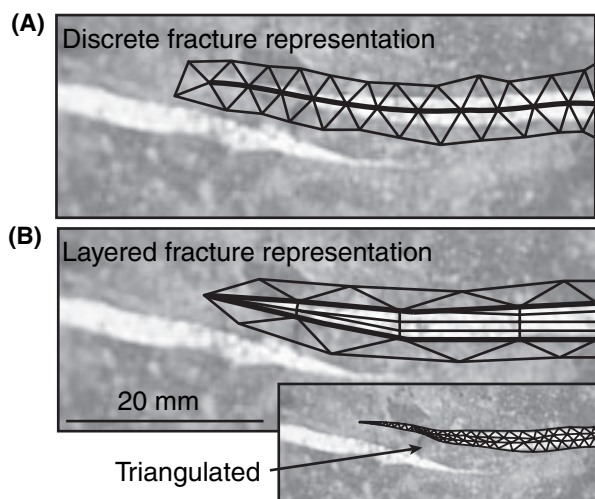


Fig. 5. Photograph of a carbonate vein in Jurassic limestone at Kilve, UK. (A) 2D NURBS model using a curve with a thickness attribute discretized by line elements surrounded by triangular elements representing the rock matrix. (B) Areal representation of vein by triangles and quadrilaterals. Inset shows comparatively inefficient triangle mesh.

such as 2D fracture elements embedded into 3D models, increases efficiency and is flexible enough to represent the characteristics of the fractures implicitly (Kim & Deo 2000; Monteagudo & Firoozabadi 2004). However, it is permissible only when the fractures are more permeable than the rock. This excludes sealed fractures and faults (Fig. 5A). Moreover, numerical simulation of capillary pressure-driven fluid exchange between fractures and rock matrix requires multiple degrees of freedom across the fracture in order to represent sharp-gradational or discontinuous saturation variations. Here volumetric meshes come into play.

While tetrahedral automatic meshing performs well for isometric material domains, large aspect-ratio geological structures like thin strata, faults or fractures, attract a large number of very small elements, see inset Fig. 5B. Some incremental meshing algorithms produce highly distorted tetrahedra leading to large finite element interpolation errors (Thompson *et al.* 1998). Others avoid these errors by subdividing elements into smaller, better shaped tetrahedra based on geometrical measures such as aspect ratio and minimum internal angle (Liu & Joe 1994). This leads to the prohibitively large number of elements. We circumvent these problems using fracture-aligned high aspect ratio prism and hexahedral elements. These perform well even if they have a large aspect ratio (Khawaja & Kallinderis 2000). This dramatically reduces the number of elements required to represent fractures. We achieve this discretization by extruding surface meshes into volumes composed of high aspect ratio prism elements. Figure 5B displays a cross-section of a fracture that has been discretized with prisms and hexahedra. Multiple element layers provide the desired internal nodal degrees of freedom.

Tapering of prism layers near fracture tips elegantly models pointy terminations.

All hybrid meshes displayed in this paper were generated using a meshing algorithm based on Moore & Warren (1990, 1991). It has been extensively modified, optimized and embedded into the Ansys ICEM CFD 5.1² Tetra mesh generator.

Step 3: Spatial FE–FV discretization of governing equations

The main idea behind the finite element method (Courant 1942; Zienkiewicz & Cheung 1965) is that an unknown continuous function, $\Psi(x, y, z, t)$, is modelled by the interpolation functions, N_i , defined in a piecewise fashion inside each finite element. To capture incremental changes of Ψ , integrals over their spatial derivatives, ∇N_i , are accumulated over the domain of interest – element by element – into a system of algebraic equations, **A**. The solution of this matrix equation gives approximate values of $\hat{\Psi}$ but only at the finite element nodes. Unique values of $\hat{\Psi}$ can be found only if an integration constant is specified. To this end, $\hat{\Psi}$ is prescribed at some node points in the boundary condition vector, **b**. Figure 6 shows a one-dimensional mesh where each finite element is represented by a line connecting two nodes. Two linear interpolation functions, N_i and N_j , model Ψ at any point, X , inside each finite element:

$$\hat{\Psi}^{(e)}(X) = \sum_{i=1}^{i \leq 2} N_i(X) \hat{\Psi}_i \quad X \in [n_1, n_2] \quad (1)$$

As they are linear, their derivatives are piecewise constant functions and are most accurate at the element centre, but discontinuous across, and therefore undefined at the nodes (Fig. 6). This also applies to fluid pressure gradients, $\nabla \hat{p}$, computed to calculate Darcy velocity, \mathbf{v} . As will be shown, flux computations for transport modelling require integration of the discontinuous velocity field. This is accomplished with node-centred finite volumes bounded by faceted surfaces. Vector quantities, including \mathbf{v} , are projected onto facet normals, \mathbf{n} . These dot products are multiplied by facet area and summed over all finite volume facets to complete the surface integral. As Green's theorem establishes the equivalence of volume and surface integrals of the form

$$\oint_{FV} \langle \mathbf{v}, \mathbf{n} \rangle dS \equiv \int_{FE} \nabla \cdot \mathbf{v} dV \quad (2)$$

where \mathbf{n} and \mathbf{v} represent the unit normal field and the Darcy velocity field, respectively, this approach elegantly

²The use of product names is for descriptive purposes only and does not constitute endorsement by *Geofluids*.

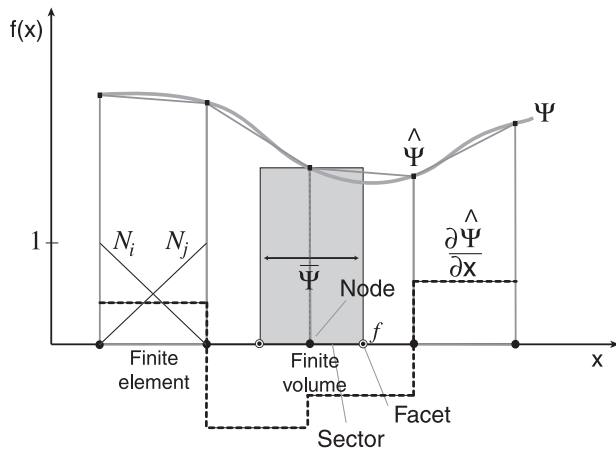


Fig. 6. Finite element node-centred finite volume discretization in one dimension. The variation of the solution variable $\Psi(x)$ is approximated by $\hat{\Psi}$ using piecewise linear interpolation functions N_i and N_j associated with the nodes of each finite element; $\partial\hat{\Psi}/\partial x$ and $\hat{\Psi}$ are the piecewise constant derivative of this solution and finite volume representation of Ψ .

avoids using the discontinuous ∇N derivatives at the finite element boundaries (Fig. 6). In stead, surface integrations are performed inside the finite elements where ∇N is continuous.

For single element-type meshes this approach was pioneered by Baliga & Patankar (1980, 1983), Cordes & Kinzelbach (1992), Durlofsky (1993, 1994) and Huber & Helmig (2000). More recently, it has also been applied to combinations of triangular and tetrahedral elements by Bogdanov *et al.* (2003), Monteagudo & Firoozabadi (2004) and Reichenberger *et al.* (2006). This paper further extends it to arbitrary combinations of the element types: bar (line) element, triangle and quadrilateral (surface) elements, tetrahedron, hexahedron, pyramid and prism (volume) elements. This is accomplished by element partitioning into finite volumes in parametric space. As a result, there is no storage required for the finite volume mesh. The concrete implementation of this scheme is described below, including integral calculations and the mapping of variables from parametric to physical space. The notation is summarized in section ‘Nomenclature’. Multi-phase flow simulations with the hybrid-element FE-FVM are presented by Matthäi *et al.* (2005a).

Finite element–finite volume stencils

Each finite element (FE) contributes to as many finite volumes (FV) as it has nodes. We call the resulting FE partitions *sectors* and the set of equations ensuing from each element, *FV stencil*. Within each FE, sectors and therefore FVs are bounded by *facets*. Sectors are volumes in 3D, surfaces in 2D, and lines in 1D. Corresponding facets are surfaces, lines and points. Figure 7 illustrates these basic concepts for triangles and quadrilaterals, showing how adjacent FEs

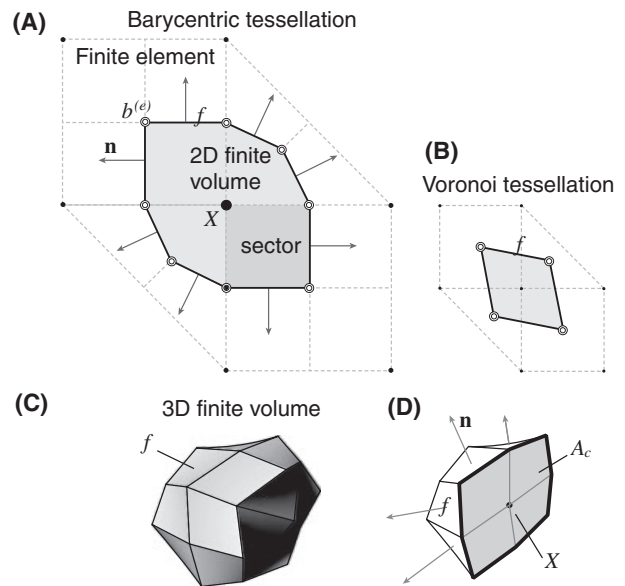


Fig. 7. Node-centred finite volumes discretizing surface and volume elements. (A) Four neighbouring triangles and quadrilaterals share node X around which finite volume is built using FE barycentres $b^{(e)}$ and midpoints of faces. FE are subdivided into sectors delimited by FV facets, f , with outward pointing normals, \mathbf{n} , and the boundaries of finite elements. (B) Voronoi tessellation of the same geometry where FVs are bounded by orthogonal bisectors cutting adjacent finite element faces. (C) 3D finite volume composed of six pyramid finite elements. (D) Cross-section revealing node at FV centre.

contribute to an FV surrounding a shared node. Importantly, facets connect at element barycentres and are subdivided in the middle of element faces (=the outer surfaces of the finite elements). This approach is referred to as barycentric tessellation and guarantees a robust decomposition of the FE mesh into non-degenerate FVs. It also permits the integration of material properties which vary from element to element. The alternative Voronoi tessellation (Voronoi 1908; Cai *et al.* 1997) is simpler, requiring only half the number of facet-related computations (Fig. 7b). However, material properties varying from element to element cannot be integrated. An even bigger disadvantage is that Voronoi cells cannot be represented in parametric space. This rules out the Voronoi tessellation for our purposes.

New barycentric FV tessellations of the bar, triangle, quadrilateral, tetrahedron, hexahedron, prism and pyramid elements in parametric space are shown in Figs 7 and 8. Taking $B(\dots)$ as a function which returns the coordinates of barycentre points of elements, e , element faces, a , and element edges, $\overline{X_i X_j}$, facet corner points F_{ij} are generated from element node coordinates as follows:

$$F_{ij} \supset \{B(X_i, X_j), B(a_1), B(e), B(a_2)\} \quad (3)$$

Here, a_1 and a_2 are the two element faces bordering $\overline{X_i X_j}$, numbered following the right-hand rule using the

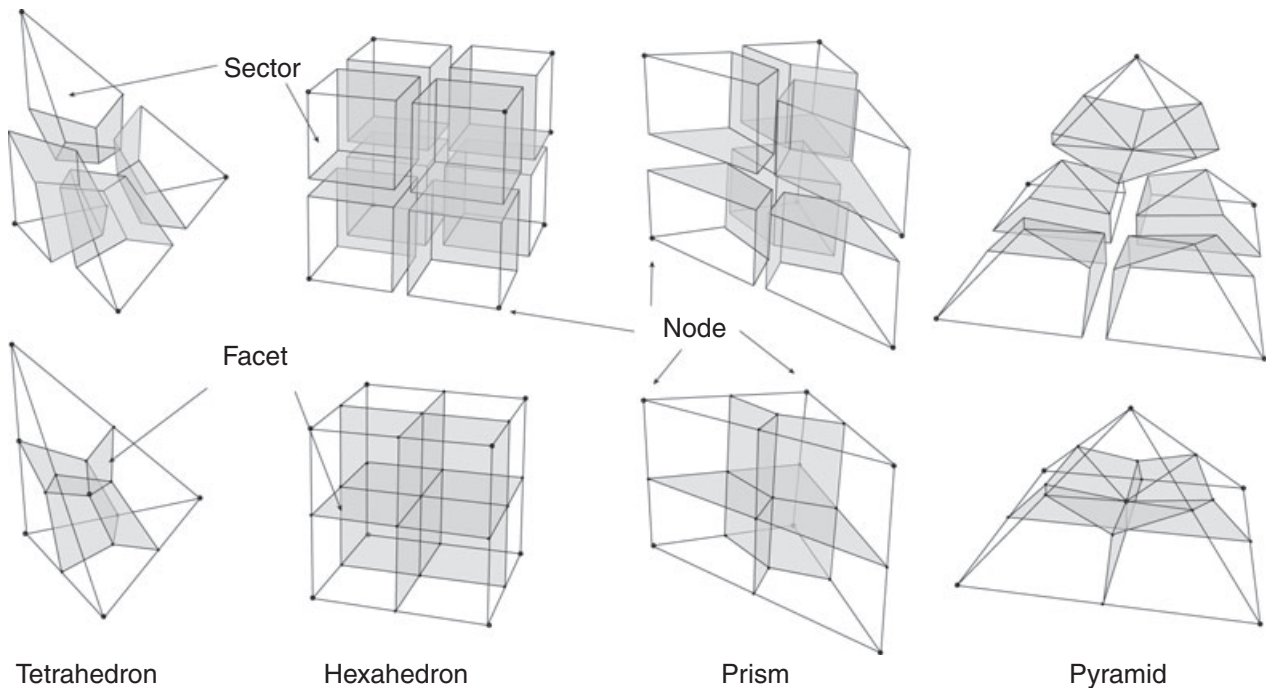


Fig. 8. Finite volume stencils for isoparametric linear tetrahedron, pyramid, prism and hexahedron. The apical sector of the pyramid on the right is an octahedron with warped basal facets making surface integrations inexact. To circumvent this problem the warped quadrilaterals are replaced by two planar triangular facets each.

direction of this edge as one of the axes. This guarantees that facet corner points are ordered counter-clockwise, the facet is outward facing relative to node i , and the scalar projection of the facet normal onto the corresponding element edge is positive:

$$(B(a_1) - B(X_i, X_j)) \times (B(e) - B(a_1)) \geq 0 \quad (4)$$

Equation (4) uses the cross-product of two interconnected facet edges to find \mathbf{n}_{ij} . To reduce the amount of floating point operations, point locations, $X_i^{(rst)}$ in parametric space can be memorized and interpolated to physical space on demand:

$$X_i^{(xyz)} = \sum_{j=1}^{j \leq n} N_j(X_i^{(rst)}) X_j^{(N,xyz)} \quad (5)$$

As the FE-FVM is based on isoparametric elements, i.e. their interpolation function order is the same as that of so-called shape functions describing element shape (Taig 1961), equation (5) guarantees a projection of X which is consistent with the spatial discretization. This point is revisited in the discussion of discretization errors.

In three-dimensional FEs, sectors are hexahedra and facets are planar quadrilaterals. The pyramid is an interesting exception: it has four non-planar facets which delimit an octahedral sector below its apex (Fig. 8b). As is alluded to in Eaton *et al.* (2003) special provisions need to be made

to perform integrations on this element. This is discussed in more detail in the section on parametric-to-physical transformations.

Volume and surface integrations for finite volumes

We apply Gaussian quadrature (e.g. Burden *et al.* 1978), to obtain surface and volume integrals from the FVs. Integration is performed for specific integration points, X_i^G with corresponding weights, w_i , chosen to give the best approximation of the unknown function. In contrast with linear FEM which integrates ‘weak’ approximations of the field variable via the linear N functions, FVM integrates constant approximations. These averages are products of the integrand and a single M basis function which is unity inside the FV and zero elsewhere. In our node-centred FE-FVM, the presence of the N functions permits to interpolate integrand values inside of FV sectors to locations where an approximation by a constant leads to an exact volume integral. The specific location in a sector where the value of the trilinearly interpolating N functions is an exact average of the integrand, is its barycentre. More integration points are warranted only, if higher-order N functions are used by the FEM. The volume integration weight for the sector barycentre, w_1 , is set to the unit volume fraction of the FE which the sector occupies in parametric space.

Equation (1) shows that surface integration of a velocity field yields the divergence of flow inside an FV. This implies, for example, that the change in the concentration, c , of a solute in an FV can be computed as a flux balance (flux, $q = t c A |\mathbf{v}|$, in kg m^{-3}) of incoming versus outgoing fluid with spatially variable concentration. The total flux, q , is accumulated by summing the fluxes across the facets of each sector of the FV:

$$q = t \oint_{FV} \langle \mathbf{n}, \mathbf{v}, c \rangle dS = t \sum_{i=1}^{i \leq s} \sum_{j=1}^{j \leq f^{(s)}} \tilde{c}_i A_i \mathbf{n}_i \cdot \mathbf{v}_j^{(e)} \quad (6)$$

Here, t refers to the duration of the flow period and, \tilde{c} is the concentration interpolated from the FE nodes (equation 1). Again choices of number and position of integration points for the scalar quantity concentration and normals for projection of the vector quantity velocity are required. Consistency demands their location to be the same. For the nodal quantity, c , interpolated by N , the same arguments apply as stated for volume integration over FV sectors: Only one integration point located at the facet barycentre is warranted. Velocity, as it is computed from $\text{grad } N$, is constant inside each finite element and across individual facets. It varies only between facets located in different elements. For planar facets, therefore, the location of the normal does not matter and it can be placed at the facet barycentre. Facets delimiting the octahedral sector in the pyramid, however, are non-planar. Here, experiments show that while the accuracy of integrated facet area can be improved with multiple integration points, internal consistency of flux computations is obtained only for a single facet normal coinciding with an integration point at the facet barycentre. To achieve both, consistency and accuracy, we divide each non-planar quadrilateral facet into two planar triangular facets. This is discussed further in the context of the FV integration error.

In summary, we compute surface integrals of vector and scalar quantities and their products using a single facet normal and integration point located at the barycentre of the facet. The individual surface integration weights, w_i , assigned to f_i , are set equivalent to their area, A_i , in parametric space. Weights, integration point locations and facet normals have to be computed only once for each finite element type, see Table 1.

Importantly, all computations described thus far are performed in parametric space. The next section explains how the obtained integrals are scaled to their actual values in physical space.

Vector and integral transformations

Sector volume is mapped from parametric to physical space by a Jacobian transformation (Barr 1984). The standard square Jacobian matrix, \mathbf{J} , for a specific FE type and

integration point therein is obtained by premultiplying the element's node coordinate vector with the shape function derivative matrix evaluated in parametric space at the integration point $X_i^{G(rst)}$ of interest. Thus, for the bar element (Fig. 6) in a one-dimensional space

$$\mathbf{J}|_{X_i^G} = \begin{bmatrix} \frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial r} \end{bmatrix} \begin{bmatrix} X_1^{(x)} \\ X_2^{(x)} \end{bmatrix} \quad (7)$$

where X_i refers to its node coordinates.

This procedure is extensively documented (e.g. Zienkiewicz & Taylor 2000). However, FE interpolation functions and their derivatives are rarely spelt out in full. To aid the reproduction of our hybrid element FE-FVM, we provide a complete list of shape functions, their derivatives, and corresponding node, face, and edge numbering conventions in the Appendix.

Facet area

Facets of surface elements are lines. To map their length from parametric to global space we use the line's Jacobian vector \mathbf{J}_m . Its determinant multiplied by the weight, w_i , yields the length of the facet in physical space. In this case, a single integration point and a unit length of the facet line in parametric space are assumed:

$$\mathbf{J}_n|_{X_i^G} = \begin{bmatrix} X_1^{(x)} & X_1^{(y)} \\ X_2^{(x)} & X_2^{(y)} \end{bmatrix} \begin{bmatrix} \frac{\partial N_1}{\partial r} = -\frac{1}{2} & \frac{\partial N_2}{\partial r} = \frac{1}{2} \end{bmatrix} \quad (8)$$

$$\Delta X^{(xyz)} = w_i |\mathbf{J}_m| \quad (9)$$

$$|\mathbf{J}_m| = \sqrt{|\mathbf{J}_n \mathbf{J}_n^T|} \quad (10)$$

$$w_i = 1$$

In equation (8) X_i refers to the coordinates of the end-points of the facet line interpolated from parametric space to physical space (equation 5).

For this type of transformations from a lower to a higher dimensional space, \mathbf{J} is not square and calculating its determinant involves transformation techniques given by Juanes *et al.* (2002). The essence of this procedure is to add (a) row(s) to \mathbf{J} to make it square and therefore invertible. The dimensions of the Jacobian of a surface element embedded in three-dimensional space are $n \times m$ (number of dimensions in physical space by number of dimensions in parametric space). This matrix is made square by multiplication with its transpose. Its determinant is the square root of the covariant metric tensor $\sqrt{\mathbf{G}}$, where $\mathbf{G} = \mathbf{J}_{m \neq n}^T \mathbf{J}_{m \neq n}$, see equation (10).

Table 1 Integration points and weights in parametric space of isoparametric linear finite elements.

	Nodes	Integration point	Weight	Normal
Isoparametric linear bar (two nodes)				
Facet				
1	(1,2)	(0,0,0)	1	(1,0,0)
Sector				
1	(1)	(-1/2,0,0)	1	
2	(2)	(1/2,0,0)	1	
Isoparametric linear triangle (three nodes)				
Facet				
1	(1,2)	(5/12,1/6,0)	$\sqrt{5}/6$	(2/ $\sqrt{5}$,1/ $\sqrt{5}$,0)
2	(2,3)	(5/12,5/12,0)	$\sqrt{2}/6$	(-1/ $\sqrt{2}$,1/ $\sqrt{2}$,0)
3	(3,1)	(1/6,5/12,0)	$\sqrt{5}/6$	(-1/ $\sqrt{5}$, -2/ $\sqrt{5}$,0)
Sector				
1	(1)	(7/36,7/36,0)	1/6	
2	(2)	(11/18,7/36,0)	1/6	
3	(3)	(7/36,11/18,0)	1/6	
Isoparametric linear quadrilateral (four nodes)				
Facet				
1	(1,2)	(0,-1/2,0)	1	(1,0,0)
2	(2,3)	(1/2,0,0)	1	(0,1,0)
3	(3,4)	(0,1/2,0)	1	(-1,0,0)
4	(4,1)	(-1/2,0,0)	1	(0,-1,0)
Sector				
1	(1)	(-1/2,-1/2,0)	1	
2	(2)	(1/2,-1/2,0)	1	
3	(3)	(1/2,1/2,0)	1	
4	(4)	(-1/2,1/2,0)	1	
Isoparametric linear tetrahedron (four nodes)				
Facet				
1	(1,2)	(13/36,5/36,5/36)	$\sqrt{6}/24$	($\sqrt{6}/3$, $\sqrt{6}/6$, $\sqrt{6}/6$)
2	(2,3)	(13/36,13/36,5/36)	$\sqrt{2}/24$	(- $\sqrt{2}/2$, $\sqrt{2}/2$,0)
3	(3,1)	(5/36,13/36,5/36)	$\sqrt{6}/24$	(- $\sqrt{6}/6$, - $\sqrt{6}/3$, - $\sqrt{6}/6$)
4	(1,4)	(5/36,5/36,13/36)	$\sqrt{6}/24$	($\sqrt{6}/6$, $\sqrt{6}/6$, $\sqrt{6}/3$)
5	(2,4)	(13/36,5/36,13/36)	$\sqrt{2}/24$	(- $\sqrt{2}/2$,0, $\sqrt{2}/2$)
6	(3,4)	(5/36,13/36,13/36)	$\sqrt{2}/24$	(0,- $\sqrt{2}/2$, $\sqrt{2}/2$)
Sector				
1	(1)	(23/144,23/144,23/144)	1/24	
2	(2)	(25/48,23/144,23/144)	1/24	
3	(3)	(23/144,25/48,23/144)	1/24	
4	(4)	(23/144,23/144,25/48)	1/24	
Isoparametric linear prism (six nodes)				
Facet				
1	(1,2)	(5/12,1/6,-1/2)	$\sqrt{5}/6$	(2/ $\sqrt{5}$,1/ $\sqrt{5}$,0)
2	(2,3)	(5/12,5/12,-1/2)	$\sqrt{2}/6$	(- $\sqrt{2}/2$, $\sqrt{2}/2$,0)
3	(3,1)	(1/6,5/12,-1/2)	$\sqrt{5}/6$	(-1/ $\sqrt{5}$, -2/ $\sqrt{5}$,0)
4	(1,4)	(7/36,7/36,0)	1/6	(0,0,1)
5	(2,5)	(11/18,7/36,0)	1/6	(0,0,1)
6	(3,6)	(7/36,11/18,0)	1/6	(0,0,1)
7	(4,5)	(5/12,1/6,1/2)	$\sqrt{5}/6$	(2/ $\sqrt{5}$,1/ $\sqrt{5}$,0)
8	(5,6)	(5/12,5/12,1/2)	$\sqrt{2}/6$	(- $\sqrt{2}/2$, $\sqrt{2}/2$,0)
9	(6,4)	(1/6,5/12,1/2)	$\sqrt{5}/6$	(-1/ $\sqrt{5}$, -2/ $\sqrt{5}$,0)
Sector				
1	(1)	(5/24,5/24,-1/2)	1/6	
2	(2)	(7/12,5/24,-1/2)	1/6	
3	(3)	(5/24,7/12,-1/2)	1/6	
4	(4)	(5/24,5/24,1/2)	1/6	
5	(5)	(7/12,5/24,1/2)	1/6	
6	(6)	(5/24,7/12,1/2)	1/6	

Table 1 (continued)

	Nodes	Integration point	Weight	Normal
Isoparametric linear pyramid (five nodes)				
Facet				
1	(1,2)	(0,-4/9,5/36)	1/4	(1,0,0)
2	(2,3)	(4/9,0,5/36)	1/4	(0,1,0)
3	(3,4)	(0,4/9,5/36)	1/4	(-1,0,0)
4	(4,1)	(-4/9,0,5/36)	1/4	(0,-1,0)
Non-planar upper facets				
5	(1,5)	(-7/24,-7/24,17/48)	$\sqrt{2}/4$	($\sqrt{8}/12, \sqrt{8}/12, \sqrt{8}/3$)
6	(2,5)	(7/24,-7/24,17/48)	$\sqrt{2}/4$	($-\sqrt{8}/12, \sqrt{8}/12, \sqrt{8}/3$)
7	(3,5)	(7/24,7/24,17/48)	$\sqrt{2}/4$	($-\sqrt{8}/12, -\sqrt{8}/12, \sqrt{8}/3$)
8	(4,5)	(-7/24,7/24,17/48)	$\sqrt{2}/4$	($\sqrt{8}/12, -\sqrt{8}/12, \sqrt{8}/3$)
Planar triangular upper facets				
5	(1,5)	(-7/18,-1/6,13/36)	$\sqrt{74}/48$	($\sqrt{74}/74, 3\sqrt{74}/74, 4\sqrt{74}/37$)
6	(1,5)	(-1/6,-7/18,13/36)	$\sqrt{74}/48$	($3\sqrt{74}/74, \sqrt{74}/74, 4\sqrt{74}/37$)
7	(2,5)	(1/6,-7/18,13/36)	$\sqrt{74}/48$	($-3\sqrt{74}/74, \sqrt{74}/74, 4\sqrt{74}/37$)
8	(2,5)	(7/18,-1/6,13/36)	$\sqrt{74}/48$	($-\sqrt{74}/74, 3\sqrt{74}/74, 4\sqrt{74}/37$)
9	(3,5)	(7/18,1/6,13/36)	$\sqrt{74}/48$	($-\sqrt{74}/74, -3\sqrt{74}/74, 4\sqrt{74}/37$)
10	(3,5)	(1/6,7/18,13/36)	$\sqrt{74}/48$	($-3\sqrt{74}/74, -\sqrt{74}/74, 4\sqrt{74}/37$)
11	(4,5)	(-1/6,7/18,13/36)	$\sqrt{74}/48$	($3\sqrt{74}/74, -\sqrt{74}/74, 4\sqrt{74}/37$)
12	(4,5)	(-7/18,1/6,13/36)	$\sqrt{74}/48$	($\sqrt{74}/74, -3\sqrt{74}/74, 4\sqrt{74}/37$)
Sector				
1	(1)	(-31/72,-31/72,23/144)	1/4	
2	(2)	(31/72,-31/72,23/144)	1/4	
3	(3)	(31/72,31/72,23/144)	1/4	
4	(4)	(-31/72,31/72,23/144)	1/4	
5	(5)	(0,0,25/48)	1/3	
Isoparametric linear hexahedron (eight nodes)				
Facet				
1	(1,2)	(0,-0.5,-0.5)	1	(1,0,0)
2	(2,3)	(0.5,0,-0.5)	1	(0,1,0)
3	(3,4)	(0,0.5,-0.5)	1	(-1,0,0)
4	(4,1)	(-0.5,0,-0.5)	1	(0,-1,0)
5	(1,5)	(-0.5,-0.5,0)	1	(0,0,1)
6	(2,6)	(0.5,-0.5,0)	1	(0,0,1)
7	(3,7)	(0.5,0.5,0)	1	(0,0,1)
8	(4,8)	(-0.5,0.5,0)	1	(0,0,1)
9	(5,6)	(0,-0.5,0.5)	1	(1,0,0)
10	(6,7)	(0.5,0,0.5)	1	(0,1,0)
11	(7,8)	(0,0.5,0.5)	1	(-1,0,0)
12	(8,5)	(-0.5,0,0.5)	1	(0,-1,0)
Sector				
1	(1)	(-0.5,-0.5,-0.5)	1	
2	(2)	(0.5,-0.5,-0.5)	1	
3	(3)	(0.5,0.5,-0.5)	1	
4	(4)	(-0.5,0.5,-0.5)	1	
5	(5)	(-0.5,-0.5,0.5)	1	
6	(6)	(0.5,-0.5,0.5)	1	
7	(7)	(0.5,0.5,0.5)	1	
8	(8)	(-0.5,0.5,0.5)	1	

The facets of volume elements are planar quadrilaterals or triangles. The determinants of their non-square Jacobians relate area in parametric space to that in physical space. A shape and integration point location as well as number-dependent weighting factor, w_i , is required so that the mapping yields the correct facet area in physical space (Table 1). For quadrilateral facets:

$$\mathbf{J}_{m \neq n}|_{X_i^G} = \begin{bmatrix} \frac{\partial N_1}{\partial r}|_{X_i^G} & \cdots & \frac{\partial N_4}{\partial r}|_{X_i^G} \\ \frac{\partial N_1}{\partial s}|_{X_i^G} & \cdots & \frac{\partial N_4}{\partial s}|_{X_i^G} \end{bmatrix} \begin{bmatrix} X_1^{(x)} & X_1^{(y)} & X_1^{(z)} \\ X_2^{(x)} & X_2^{(y)} & X_2^{(z)} \\ X_3^{(x)} & X_3^{(y)} & X_3^{(z)} \\ X_4^{(x)} & X_4^{(y)} & X_4^{(z)} \end{bmatrix} \quad (11)$$

and for the triangular facets in the apex of the pyramid:

$$\mathbf{J}_{m \neq n}|_{X_i^G} = \begin{bmatrix} \frac{\partial N_1}{\partial r}|_{X_i^G} & \dots & \frac{\partial N_3}{\partial r}|_{X_i^G} \\ \frac{\partial N_1}{\partial s}|_{X_i^G} & \dots & \frac{\partial N_3}{\partial s}|_{X_i^G} \end{bmatrix} \begin{bmatrix} X_1^{(x)} & X_1^{(y)} & X_1^{(z)} \\ X_2^{(x)} & X_2^{(y)} & X_2^{(z)} \\ X_3^{(x)} & X_3^{(y)} & X_3^{(z)} \end{bmatrix} \quad (12)$$

The coordinates X_i mark corner points of the facet mapped from parametric to physical space. For internal consistency, mapped facet area must be used together with mapped normal orientation unless velocity projections are also carried out in parametric space.

Normal orientation

Facet normals are mapped from parametric space to physical space by using the standard Jacobian inverse relationship described by Barr (1984), where:

$$\mathbf{n}^{xyz} = (\mathbf{J}_{m \neq n}^{-1})^T \mathbf{n}^{rst} \quad (13)$$

and \mathbf{J} is the Jacobian of the parent element formed at the facet integration point of interest. Once computed, the Euclidean length of the normal is scaled to 1. Facets of line elements have a normal parallel to the element pointing from node $n - 1$ to node n .

Integral computations in physical space

The mapping of facet normals from parametric to physical space is more computationally expensive than their calculation in physical space. For our optimized C++ template-based implementation the extra number of clock cycles required varies with parent element type between 5% for the tetrahedron and 90% for the quadrilateral element in three dimensions. However, this disadvantage is outweighed by the internal consistency delivered by the mapped approach: quadrilateral facets in physical space are rarely planar so that a distortion-dependent error is incurred in the area computation and it is difficult to find a representative facet normal. A remedy would be to replace all quadrilateral facets by triangular facets because these are planar by default. This is how we treat the apical facets in the pyramid element because these are distorted already in parametric space. As expected, triangular facets are faster to compute and map than quadrilateral facets (approximately 8%). However, twice as many facets, normals and projections would be required if such a discretization was used for all facets. Computational cost could only partially be offset by storing mapped normals and facet areas for repeated use.

For linear line elements, the facet normal is simply an element-parallel unit vector pointing from node to node. For surface elements, facet normals lie in the plane of the element and are calculated by:

$$\mathbf{n}_{ij} = \left(\frac{X_i + X_j}{2} - B \right) \times \mathbf{n}^{(e)} \quad (14)$$

where $\mathbf{n}^{(e)}$ is the normal to the parent surface element in physical space, i and j are its node points on either side of the facet, and B is its barycentre.

For volume elements, each facet is a quadrilateral or triangular isoparametric FE with a surface orientation defined by the two counter-clockwise edge vectors originating in B . Their cross product is the facet outward pointing normal. A single Jacobian transformation is needed to find facet area in physical space. For quadrilateral facets:

$$\mathbf{J}_{m \neq n}|_B = \begin{bmatrix} -1/4 & 1/4 & 1/4 & -1/4 \\ -1/4 & -1/4 & 1/4 & 1/4 \end{bmatrix} \begin{bmatrix} X_1^{(x)} & X_1^{(y)} & X_1^{(z)} \\ X_2^{(x)} & X_2^{(y)} & X_2^{(z)} \\ X_3^{(x)} & X_3^{(y)} & X_3^{(z)} \\ X_4^{(x)} & X_4^{(y)} & X_4^{(z)} \end{bmatrix} \quad (15)$$

$$w_1 = 4$$

and for triangular facets:

$$\mathbf{J}_m = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} \quad (16)$$

$$w_1 = 1/2$$

The facet corner point coordinates, X_i , in equations (3–5) are calculated directly in physical space. To obtain unit normals, all facet normals are divided by their length.

We favour the parametric space approach combined with Jacobian mapping over the direct computation of sector volume, facet area and facet normals in physical space. Noteworthy is that for the transport modelling described later, computation speed can be improved if the transport velocity is also computed in parametric space, eliminating among other steps the mapping of the facet normal. The presented numerical method is suitable also for quadratic or cubic finite elements. The main idea of finite volumes surrounding finite element nodes remains the same. Notice, however, that the improved continuity for such elements permits to treat them as finite volumes. This constitutes another discretization approach, an element-centred FE-FVM.

Step 4: Tracer advection in fractured porous rock

In this section we discretize the governing equations for fluid flow and tracer advection with the FE-FVM for a demonstration on a model of the intensely fractured San

Andreas formation in the Results section. We will assume that transport occurs in a time invariant velocity field without feedback between tracer concentration and velocity. Huyakorn & Pinder (1983) solved this problem by coupling the solution of two partial differential equations for pressure and tracer concentration, respectively. As the pressure equation is elliptic and the transport equation is hyperbolic, we apply a simple operator splitting technique solving for pressure with the FEM and for concentration with the FVM. Note, however, that our FE-FVM is completely general, forming the basis of a higher order accurate, semi-implicit transport method for two-phase flow (Matthäi *et al.* 2005b), and calculations of thermohaline convection including the effects of phase separation of highly compressible steam (Geiger *et al.* 2006a,b). The discussion presented here serves the sole purpose of illustration of how such schemes can be constructed with the data structures which we provide.

Governing pressure equation

Assuming that the divergence of the velocity field is zero, i.e. that there are no fluid sources or sinks in the model domain Ω ,

$$\nabla \cdot \mathbf{u} = 0 \text{ (conservation of volume)} \quad (17)$$

If the volumetric flow, \mathbf{u} , obeys Darcy's law,

$$\mathbf{u} = A \frac{\mathbf{k}}{\mu} \nabla p \quad (18)$$

it follows that,

$$A \frac{\mathbf{k}}{\mu} \nabla^2 p = 0 \quad (19)$$

where p is pressure, A is the area of the flow cross-section, \mathbf{k} is the piecewise-constant permeability tensor, and μ is a piecewise-constant dynamic fluid viscosity. FE integration of PDE (19) yields the element contributions to the domain integral $\int \Omega$ for the model:

$$\int dN^T A \frac{\mathbf{k}}{\mu} dN dV = 0 \times \int N^T N dV \quad (20)$$

where dN represents the interpolation function matrix of spatial derivatives (see Appendix). This integration requires a value of the integration constant (fluid pressure) fixed at, at least, one node in the model.

Governing advection–diffusion equation

Advection and diffusion of a non-reactive tracer through a fractured porous medium with discrete fracture representations can be described by

$$\phi \frac{\partial c}{\partial t} + \mathbf{D} \nabla^2 c + \nabla \cdot \mathbf{v} c - q = 0 \quad (21)$$

The variables in this continuity equation are the porosity, ϕ , tracer concentration, c , macro dispersivity, \mathbf{D} of the tracer, and \mathbf{v} the Darcy velocity; q refers to sources or sinks of the tracer. In many practical cases it is not possible to parameterize \mathbf{D} due to small-scale geometric uncertainty but we retain \mathbf{D} as a diagonal tensor to illustrate how diffusion and advection can be solved for simultaneously using the FE-FVM.

Equation (15) is discretized spatially using the FV framework and temporally using a fully implicit (Backward-Euler) finite difference scheme. For each FV, this yields the volume and surface integrals listed below. These are accumulated into a sparse solution matrix, \mathbf{A} , of linear algebraic equations i with coefficients j and a right-hand vector, \mathbf{b} , indexed with i :

$$\begin{aligned} & \int_{V_\Phi} M_i \frac{\partial c_i}{\partial t} dV + \int_{\Gamma_{\text{out}}^{\text{FV}}(\bar{n}_f \cdot \bar{u}(c) > 0)} \mathbf{n} \cdot \mathbf{u} c_c + D_n \frac{\partial c}{\partial \mathbf{n}} c_c dS \\ & + \int_{\Gamma_{\text{in}}^{\text{FV}}(\bar{n}_f \cdot \bar{u}(c) < 0)} \mathbf{n} \cdot \mathbf{u} c_u + D_n \frac{\partial c}{\partial \mathbf{n}} c_u dS - \int_{V_\Phi} M_i q_i dV = 0 \\ & \mathbf{A}_{(i,i)} = \frac{V\Phi}{\Delta t} + \sum_{j=0}^{j \leq n_f^{(\text{FV})}} A_j \mathbf{n}_j \cdot \mathbf{u}_j + k \frac{\partial c}{\partial \mathbf{n}_j} H \left(\mathbf{n}_j \cdot \mathbf{u}_j + D_n \frac{\partial c}{\partial \mathbf{n}_j} \right) \\ & \mathbf{A}_{(i,j)} = \sum_{j=0}^{j \leq n_f^{(\text{FV})}} A_j \mathbf{n}_j \cdot \mathbf{u}_j + k \frac{\partial c}{\partial \mathbf{n}_j} H \left(1 - \left(\mathbf{n}_j \cdot \mathbf{u}_j + D_n \frac{\partial c}{\partial \mathbf{n}_j} \right) \right) \\ & \rightarrow H = \begin{cases} 1 & \text{if } \mathbf{n}_f \cdot \mathbf{u}_f + D_n \frac{\partial c}{\partial \mathbf{n}_j} < 0 \\ 0 & \text{otherwise} \end{cases} \\ & \mathbf{b}_{(i)} = \frac{c_i^t V\Phi}{\Delta t} + q_i^{t+1} \end{aligned} \quad (22)$$

Concentration, c , is discretized on the FVs by the piecewise constant interpolation functions M_i . The subscripts, u and c refer to the upstream and current finite volumes relative to a facet, f , across which the contribution to the total flux is calculated. Each FV facet has the unit normal \mathbf{n} and area A (Fig. 6). The Heaviside function $H(\cdot)$, is used to discard couplings between FVs which would arise from fluxes in the downstream direction. As H evaluates to zero in this case, only incoming fluxes create couplings between adjacent FVs, i.e. diagonal and off-diagonal terms in the solution matrix. This technique is known as upstream weighting and is unconditionally stable for first-order schemes (Baliga & Patankar 1980).

Diffusion of the tracer is modelled simultaneously with advection projecting concentration gradients onto the facet normals, see Equation (6).

At the domain boundaries

$$\begin{aligned} \mathbf{A}_{ii} &= \mathbf{A}_{ii} + \left| \sum_{j=1}^{j \leq n_f} A_j \mathbf{n}_j \cdot \mathbf{u}_j \right| & \text{if } \sum_{j=1}^{j \leq n_f} A_j \mathbf{n}_j \cdot \mathbf{u}_j < 0 \\ \mathbf{b}_i &= \mathbf{b}_i + \bar{c}_i \sum_{j=1}^{j \leq n_f} A_j \mathbf{n}_j \cdot \mathbf{u}_j & \text{if } \sum_{j=1}^{j \leq n_f} A_j \mathbf{n}_j \cdot \mathbf{u}_j > 0 \end{aligned} \quad (23)$$

Thus, a left-hand-side implicit (meaning that c does not need to be specified) compensation is made for finite volumes split by outflow boundaries and an explicit one for inflow boundaries.

The resulting non-symmetric but diagonally dominant and positive definite matrix of linear algebraic equations, $\mathbf{A} \mathbf{x} = \mathbf{b}$, as well as the symmetric matrix arising from the pressure equation, is solved with the algebraic multigrid method for systems, SAMG (Stüben 2001). The results are stored at the finite element nodes.

The new hybrid element FE-FVM is implemented in the Complex System Modelling Platform (CSMP++), an object-oriented application programmer interface (API), designed for the simulation of complex geological processes and their interactions (formerly CSP, cf. Matthäi *et al.* 2001).

RESULTS

We evaluate the new hybrid FE-FVM framework in four ways: (1) we measure the impact of increasing geometric complexity on the quality of the generated hybrid element meshes, (2) we measure the FE discretization error for the pressure equation, (3) we evaluate the FV integration error by integrating prescribed and computed non-divergent

velocity fields over the model, and (4) we estimate speed and memory requirements of the implicit first-order transport scheme applied to the intensely fractured San Andreas model.

Mesh refinement and increasing geometric complexity

Three fracture models built using FracMan Reservoir Edition (FRED) (Golder Associates, Seattle, WA, USA), are used to illustrate the effects of increasing geometric complexity on mesh quality (Fig. 9). These models are populated with randomly oriented disc-shaped fractures with a power law length distribution. The power law exponent is 1.5 and fracture diameter ranges from 5 to 50 m (box size $100 \times 100 \times 100$ m). The ratio between the fracture surface area and the volume of the model, also known as p32 (cf. Dershowitz 1984), is used as a proxy for geometric complexity. In contrast with the polygonal representation in FracMan, however, we represent each fracture by a circular NURBS surface.

Each model was meshed at three levels of refinement decreasing target element size from 10 to 5 to 2 m. The resulting tetrahedral meshes were converted into hybrid elements by transforming groups of 12 tetrahedra into single hexahedra. This procedure succeeds in regions of the CAD model that are far from confining surfaces. Conversion rate depends on feature density relative to target element size.

In all cases, the mesh was adaptively refined down to one-tenth of the reference size to match the curved tip lines of the fractures. The least complex model, model A, has a p32 of 0.05, and contains 66 fractures. Model B has 108 fractures and a p32 of 0.1, and model C, the most

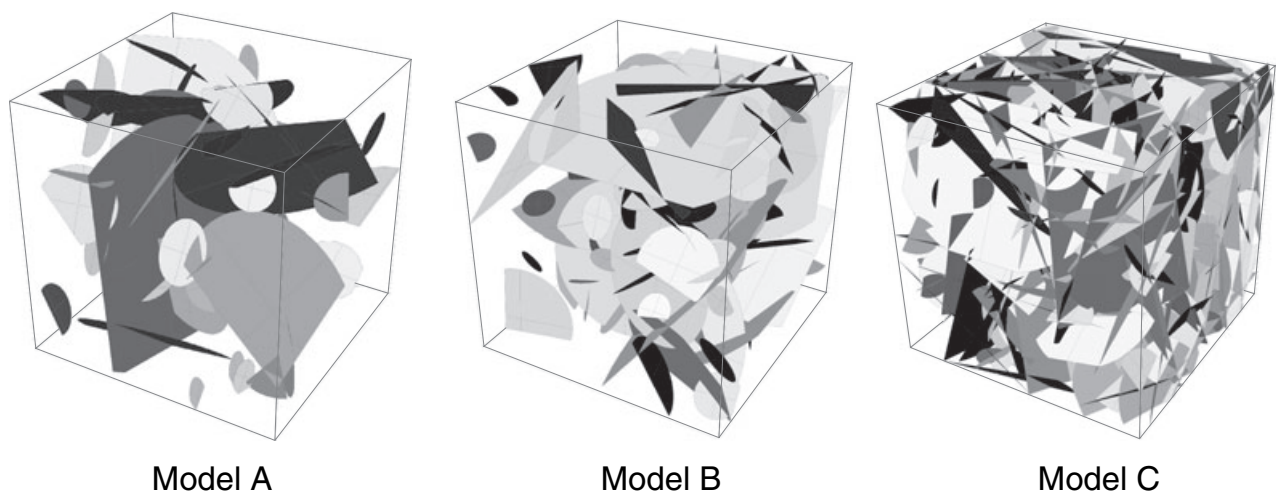


Fig. 9. Discrete fracture models of increasing geometric complexity generated with FracMan Reservoir Edition (FRED). (A) 66 fractures and 117 intersection curves, (B) 108 fractures, 313 intersection curves, and (C) 334 fractures, 2262 intersection curves.

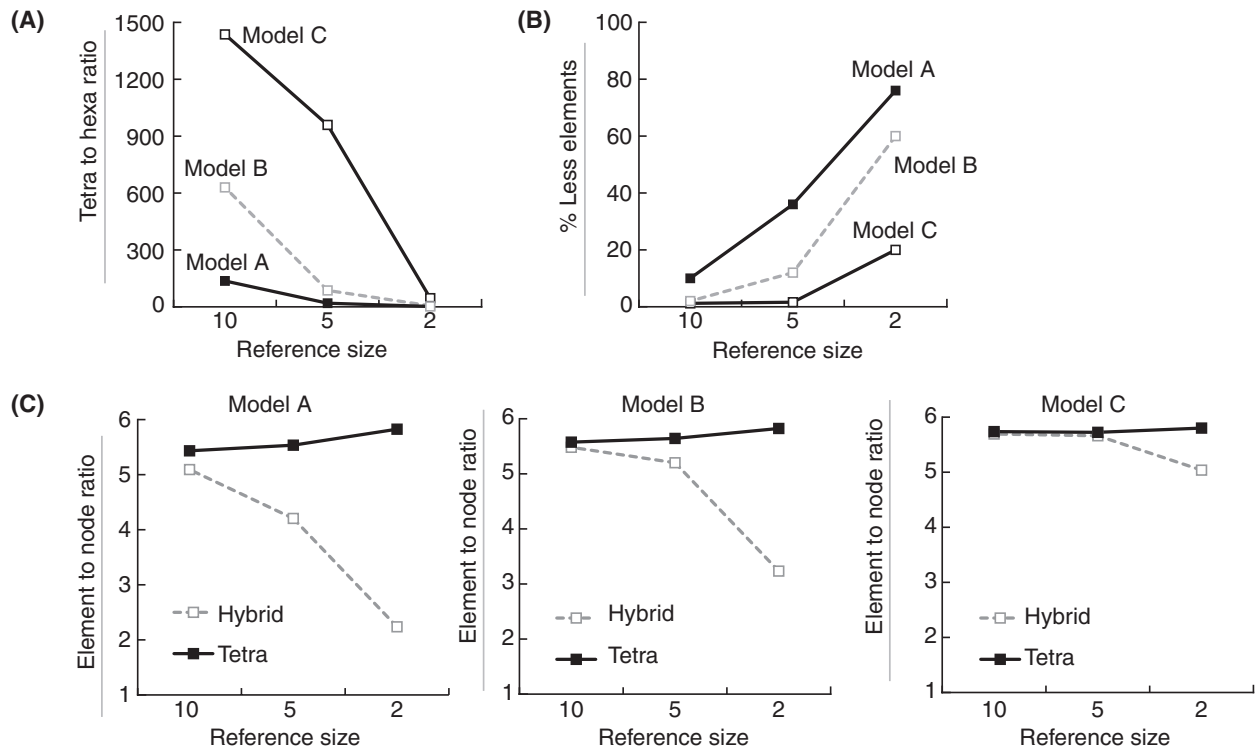


Fig. 10. Mesh properties as a function of geometric complexity and refinement. (A) Ratio of tetrahedral to hexahedral elements as a function of p32 and element reference size. (B) Reduction of total element number upon conversion to hybrid mesh, and (C) element to node ratio as a function of mesh refinement.

complex model, contains 334 fractures with a p32 of 0.3. In total, 18 meshes (3 models \times 3 refinement levels \times 2 mesh types) were generated to evaluate mesh characteristics such as the impact of refinement on the number of nodes, number of elements, element to node ratio, and hexahedral to tetrahedral element ratio (Fig. 10). This ratio is of interest because it influences the sparseness of the solution matrix as well as the complexity of the FV cells. Our detailed analysis shows that:

- As refinement increases more and more tetrahedra are converted to hexahedra. Feature (=fracture) poor regions are discretized exclusively with hexahedra while fracture-rich regions remain tetrahedra. Tetrahedra dominate near inclined fractures. With increasing refinement, tip line curvature becomes less influential on the conversion rate and larger volume fractions are converted to hexahedra.
- In model A, tetrahedral to hexahedral element conversion reduces the number of nodes by 3.48–37%. The number of volume elements is reduced by 9.59–76%, respectively. The element to node ratio decreases from an average of 5.82 for pure tetrahedral meshes (consistent with the observations of Bogdanov *et al.* 2003), to an average of 2.23 elements in the most refined hybrid meshes.

- In model B conversion decreases the number of nodes by 0.95–29%. The number of volume elements is reduced by 2.61–60%. Element to node ratio changes from an average of 5.68 in the tetrahedral meshes to 4.63 in the hybrid ones.
- In the intensely fractured model C, differences between tetrahedral and hybrid meshes are not as pronounced as in A and B. The hybrid mesh only has up to 10% less nodes and elements, and the element to node ratio stays around 5.46. Thus, as model complexity increases, less and less elements are converted into hexahedra. However, this should not be interpreted as a weakness of the method because conversion percentage relates directly to element size in relation to feature density. As feature density increases, the domain is more intensively subdivided and less tetrahedral elements are converted into hexahedra. As accurate resolution of the curved pressure field between fractures requires multiple elements (see Step 5 of workflow), a poor conversion ratio almost always indicates that the mesh is too coarse to resolve the input geometry.

The fractures in models A, B and C are represented by surfaces. For a volumetric discretization by high aspect ratio prism elements, the number of additional elements required scales with the number of element layers. The

representation of a fracture by multiple layers of prism elements with a large aspect ratio of 150 requires several orders of magnitude less elements than a volumetric representation by tetrahedra. For models A–C and a target element size of 2 m, a conversion of fractures from triangulated surfaces to volumes with four prism layers increases the total element number only by 39, 43 and 53%, respectively.

Hybrid element meshes share the advantages of unstructured and structured grids: they are flexible enough to capture complex geometry but use the minimum number of elements required to discretize regions without geometric constraints. Fast transitions between highly refined and coarse regions are possible and the better performing hexahedra are used wherever the geometry allows this.

Step 5: *A posteriori* mesh adaptation guided by finite element discretization error

During the discretization phase the mesh is optimized to accurately represent the complex geometry. This match can be quantified, for example, in terms of the largest deviation of an element face from a material interface represented by an NURBS curve. The finite element discretization error has a similar role as it quantifies the ability of the mesh to represent a specific solution for a specific order of the element interpolation functions. For polynomials of n th degree the discretization error is of the order $O(h^{n+1})$ (e.g. Zienkiewicz & Taylor 2000), where h is the element size. For linear elements, used in this work, the error is of order $O(h^2)$. They can capture linear variations in the solution variable exactly, but any nonlinear variations produce a solution curvature dependent error which scales with element size. As the mesh is refined, this error decreases proportionally. While it cannot be eliminated completely, adaptive refinement can be used to distribute the discretization error uniformly over the mesh. To guide such refinement a measure for the discretization error is needed. By contrast with first-order accurate linear elements, quadratic elements are second-order accurate ($O(h^3)$) and the normalized difference between quadratic and linear FEM solutions

$$\varepsilon^2(\Psi) \approx \frac{\|\hat{\Psi}^2\| - \|\hat{\Psi}^1\|}{\|\hat{\Psi}^2\|} \quad (24)$$

therefore quantifies the $O(h^2)$ error term. In equation (24) $\hat{\Psi}^1$ and $\hat{\Psi}^2$ are the linear and quadratic FEM solutions, respectively.

To illustrate ε^2 we have computed it for a uniformly refined idealized model of three intersecting disc-shaped fractures (Fig. 12). This model also allows an assessment of the accuracy of a discretization error estimate $\tilde{\varepsilon}^2$ made in

the absence of a second-order accurate solution which is usually too expensive to compute.

Cook *et al.* (2002) show that a linear FEM solution contains enough information to estimate the discretization error. Here, we compute an element centred error metric using a Hessian matrix-based *a posteriori* error estimate that takes into account the size and shape of the finite elements as well as the level of alignment of the mesh with the solution (Pain *et al.* 2001),

$$\tilde{\varepsilon}^2 = \gamma \mathbf{h}^T |\mathbf{H}| \mathbf{h}, \quad \mathbf{H} = \nabla N^T \hat{\Psi} \nabla N \quad (25)$$

where γ is a scale factor needed to arrive at a target finite element size and \mathbf{H} is the characteristic Hessian matrix obtained by an eigen decomposition of the squared interpolation function derivative matrix multiplied with the first-order FEM solution. Each row of $|\mathbf{H}|$ contains an eigenvector λ of \mathbf{H} , scaled in length by the corresponding eigen value. The vector \mathbf{h} contains the length of the analysed element in the direction of this eigenvector.

Figure 11 illustrates how \mathbf{h} is computed by measuring element length in the directions of λ_1 – λ_3 in terms of the height of an oriented bounding box around the element. This computation is performed as follows:

$$\mathbf{h}_i = \max_{j=1}^{j \leq n_n} \left\{ \frac{X_j^{(xyz)} \cdot \lambda_i}{|\lambda_i|} \right\} - \min_{j=1}^{j \leq n_n} \left\{ \frac{X_j^{(xyz)} \cdot \lambda_i}{|\lambda_i|} \right\} \quad (26)$$

where X_j is the node coordinate vector for node j of the element in physical space. Figure 12A plots the actual and estimated discretization errors which have a very similar distribution. Their magnitude also matches reasonably well (within a factor of 3). Five peaks are localized in the rock matrix next to up- and downstream fracture terminations (cf. Fig. 12B). In these locations fluid pressure gradients and their spatial variation are maximized and the error indicator suggests that they should be highly refined ($\geq 5\times$). Within the fractures the error is small because their high permeability minimizes the pressure gradient. Thus, large aspect ratio prism or hexahedral elements are a suitable discretization for these.

Alongside with Pain *et al.* (2001) and Lipnikov & Vassilevski (2004), we conclude that a Hessian-based error metric provides a good estimate of the discretization error and can be used to control adaptive mesh refinement. Within the hybrid FE-FVM framework, the computation of $\tilde{\varepsilon}^2$ for the pressure equation is possible because the N functions are differentiable. By contrast, pressure computations based on a finite-volume approach only, as are common in reservoir simulators, preclude this. While the FVM is mass conservative by default, flow velocities computed on a coarse model with large aspect ratio heterogeneities will have little in common with ones obtained from an appropriately refined FE-FVM discretization.

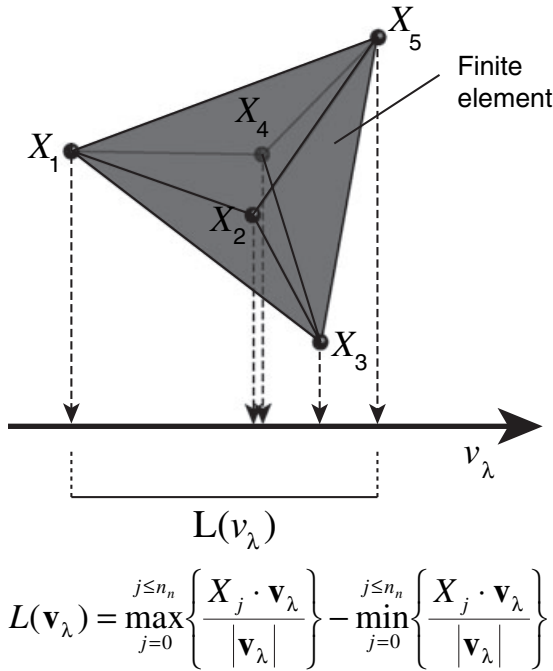


Fig. 11. Computation of element length, L , in direction \mathbf{v}_λ illustrated for a tetrahedron. Corners of bounding box are projected onto direction vector. L is the maximum separation of projected points; approach is equivalent to measuring height of bounding box in direction of \mathbf{v}_λ .

Finite-volume integration error

To validate the internal consistency of the stencils, mappings and finite volume surface integrations with a hybrid element model we compare the integrated volumetric inflow and outflow from each FV for a prescribed constant velocity with a magnitude of 1 and a computed divergence free velocity field of spatially variable magnitude. For the constant velocity, the measured differences, Δu , should be numerically zero (`numeric_limits<double>::epsilon()`, see C++ standard). For the spatially variable velocity, Δu will depend on the divergence of the computed field, i.e. numerical accuracy of the linear FE solution, and on the field's curl in relation to FE size. In any case, the volume integration error scales with FV flow cross-sectional area, A_T :

$$A_T = \frac{1}{2} \sum_{j=1}^{j \leq n_f^{(FV)}} A_j \left| \mathbf{n}_j \cdot \frac{\mathbf{v}}{|\mathbf{v}|} \right| \quad (27)$$

To account for this size effect Δu is scaled by A_T , as follows:

$$\varepsilon_i^{(FV)} = \left| \frac{\Delta u_i}{A_{T,i}} \right| \quad (28)$$

where $\varepsilon_i^{(FV)}$ is the volume integration error for finite volume i , reflecting the accuracy of facet area calculations and pro-

jections of velocities. For the $10 \times 10 \times 7$ m hybrid element model (Fig. 12B) with 19% tetrahedra, 18% pyramids, and 63% prisms and finite element volumes ranging from 1.88×10^{-5} to 0.0708 m^3 , we obtain the following results:

1. Prescribed unit velocity: $\varepsilon_{\max}^{(FV)}$ is less than or equal to $9.379 \times 10^{-14} \text{ m}^3 \text{ sec}^{-1}$. For the entire model the difference between inflow and outflow is $6.82 \times 10^{-13} \text{ m}^3 \text{ sec}^{-1}$ at a total cross-sectional flow of $70 \text{ m}^3 \text{ sec}^{-1}$.
2. For the computed velocity field with magnitudes ranging from 1.695×10^{-5} to $6.454 \times 10^{-3} \text{ m sec}^{-1}$, $\varepsilon_{\max}^{(FV)}$ is less than or equal to $1.16 \times 10^{-14} \text{ m}^3 \text{ sec}^{-1}$. The total cross-sectional flow through this model is $0.0038 \text{ m}^3 \text{ sec}^{-1}$.

As the maximum velocity in the computed velocity field is orders of magnitude smaller than the prescribed velocity (case 1), the second set of results implies a much larger $\varepsilon_{\max}^{(FV)}$ for these actual application conditions. However, the maximum $\varepsilon_{\max}^{(FV)}$ value still is so small that the corresponding violation of volume conservation does not lead to a drift in the values of the advected variable. In the development of the method and experimentation with a pyramid stencil with all quadrilateral facets, three orders of magnitude larger $\varepsilon_{\max}^{(FV)}$ occurred wherever pyramid stencils contributed to finite volumes. This result prompted us to replace the warped quadrilateral facets in the apex of the pyramid by planar triangular ones.

Application of implicit first-order transport scheme to discrete fracture model

Figure 13 is a planview of the $1 \times 1 \times 0.2$ km-sized model FRACS2000 of a fractured sandstone reservoir in the San Andreas formation, southern California, USA (Matthäi *et al.* 2005b). Two sets of subvertical fractures partially intersect at angles between 70 and 80° . The total fracture-rock matrix interface area is 12.3 km^2 . Sandstone porosity (0.25) and permeability (10 mD) are treated as uniform, and fracture aperture as a linear function of fracture diameter leading to values ranging from 0.5 to 3.5 mm. The parallel plate model (e.g. Witherspoon *et al.* 1980) was used to compute fracture permeability for these aperture values. We treat the macro-dispersivity, \mathbf{D} as an isotropic diagonal tensor valued 10^{-30} . This means that any diffusion which arises in the computation is of numerical nature.

Model FRACS2000 is discretized with 1 113 580 hybrid elements (60% tetrahedra, 15% hexahedra, 13% triangles, 11% pyramids, 1% quadrilaterals and 0.01% prisms) and 223 705 nodes (=degrees of freedom) using a surface-mesh representation of the fractures.

Figure 13 displays a snapshot of the concentration of a non-reactive tracer injected continuously for 4 months through the lower boundary into a time-invariant flow field

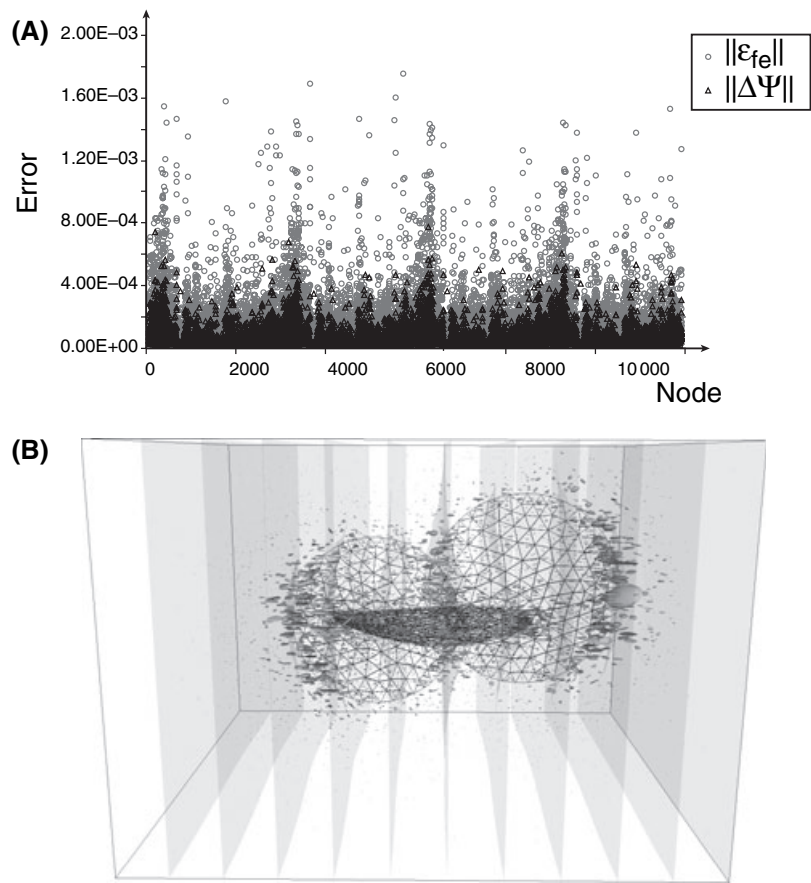


Fig. 12. Estimation of spatial discretization error for *a posteriori* mesh refinement. (A) Normalized difference between predicted maximum of $O(h^2)$ error ε , and its exact value. Error is plotted as a function of node number for model shown in (B). (B) Model of three intersecting highly permeable discs. Isosurfaces of fluid pressure and anisotropic Hessian-based discretization error are visualized by tensor glyphs rendered with the visualization toolkit (VTK). The spatial discretization error is largest where the curvature in the fluid pressure gradient is maximized.

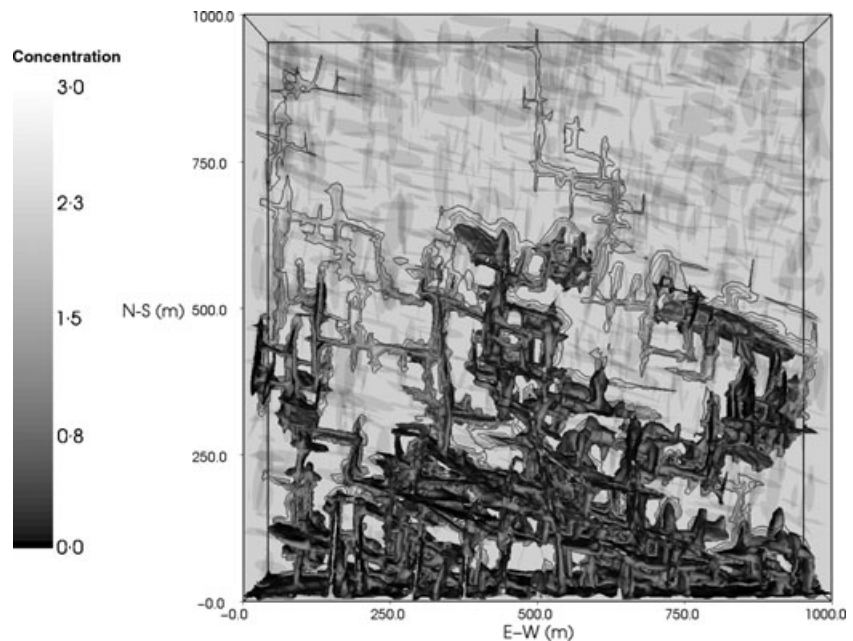


Fig. 13. Planview of three-dimensional flow and transport model FRACS2000 (Fig 8, 28–32 of Matthäi *et al.* 2005a). Translucent fractures belong to two approximately orthogonal sets, 1000 fractures per set, each with a power-law diameter-frequency distribution. Flow is from the bottom to the top. After 3.5 months of injection through the lower boundary the tracer front visualized by a dark grey isosurface through a tracer concentration of 1.5 has advanced through two-third of the model. Contour lines on a horizontal cut plane mark the tracer dispersion across the front.

arising from a hydrostatic far-field fluid pressure gradient aligned with the model edge. This gradient was created by applying uniform pressures of approximately 20 and

0 MPa at the two opposite model boundaries, respectively. The pressure equation (19) was solved with the FEM and the transport equation (21) with the first-order implicit

FVM (equation 22). The run on a 2-GHz M-processor notebook took 6.5 min. Most of this time was used to write output files in text format. This result reflects a logarithmic scaling of solution time with number of unknowns as the decisive characteristic of SAMG which can be applied in spite of the non-symmetric shape of the sparse solution matrix.

The snapshot (Fig. 13) reveals a highly embayed fracture-controlled tracer front partially penetrating the sandstone. In the fractures, the tracer has advanced over two-thirds of the model's length equating to an average speed of 0.2 m h^{-1} . Examination of the finite volumes in the different CAD domains reveals that only 14% of the fractures have come in contact with the tracer thus far. Averaging tracer concentration in front-parallel slabs along the direction of flow yields a long leading edge typical for 'anomalous' non-Fickian dispersion seen in actual fractured rock (e.g. Berkowitz & Scher 1995).

The steep fracture-associated concentration gradients perpendicular to the tracer front show that even for the first-order transport scheme used, numerical diffusion is subordinate to mechanical dispersion caused by the fractures. While no physical importance should be attributed to this specific set of results as they lack any meaningful parameterization of \mathbf{D} , the example serves to illustrate the potential capabilities of FE-FVM combined with SAMG.

DISCUSSION

The characteristics of the geometrically complex simulation with model FRACS2000 highlight the decisive advantages of our hybrid FE-FVM scheme:

- Indirect unstructured hybrid-element meshing facilitates efficient and semi-automatic discretization of complex geological models, even permitting a realistic representation of the power law length–frequency relationships which typify some natural fractured rocks. For comparison, a structured regular grid model of FRACS2000 with the same amount of detail would have required 4^{11} cells $((1000 \times 1000 \times 200)/0.0005 = \text{aperture of smallest fracture})$, i.e. 5 orders of magnitude more cells than the unstructured model.
- In spite of the cheap linear FEM derived piecewise-constant velocity field, our node-centred FVM conserves the advected tracer during migration across strong discontinuities in permeability and porosity.
- Although it is advocated primarily for elliptic problems, the algebraic multigrid solver SAMG performs fast and reliably on the non-symmetric hyperbolic sets of linear algebraic equations arising from the upstream-weighted advection–dispersion PDE.
- The effort to generate the internally consistent BREP with topology, i.e. the CAD volume decomposition of the geological system also pays off in the analysis of the

results because the model subvolumes distinguished by the CAD can be analysed individually so that their specific behaviour and interplay can be determined.

CONCLUSIONS

We introduce a new workflow for the simulation of structurally complex geologic models starting with geological interpretation and NURBS BREP of structures followed by indirect unstructured hybrid finite element meshing aimed at producing an unstructured mesh which consists primarily of hexahedra. This second step includes the volumetric discretization of fractures and faults by multiple layers of large aspect ratio prism and hexahedral elements. We present new isoparametric finite element-based finite volume stencils which eliminate the storage of the finite-volume mesh while facilitating an efficient accumulation of the solution matrix. The resulting discretization is suited for the efficient solution of flow and transport equations in highly heterogeneous geological models. Due to the application of algebraic multigrid, solution time scales logarithmically with the number of unknowns. We analyse our workflow for models of variable complexity arriving at error estimates for the finite element spatial discretization as a guide for mesh refinement and the accuracy of the finite volume integrations. We also demonstrate it on a field data-based model of intensely fractured rock.

ACKNOWLEDGEMENTS

The development of this discretization scheme has been supported by the sponsors of the UK's industry technology facilitator (itf) ISF project: BG, BP, ConocoPhillips, Shell, Statoil and Total. We thank F. Schneider and A. M. Shapiro for their careful reviews which helped to improve the manuscript.

NOMENCLATURE:

A	area
B	barycentre of finite element
\mathbf{h}	element length vector
D	spatial dimension, i.e. 1D, 2D or 3D
G	integration point
n_e	number of finite elements
n_n	number of finite element nodes
n_f	total number of facets
$n_f^{(e)}$	total number of facets of the element e
n_G	number of integration points
\mathbf{I}	identity matrix
\mathbf{J}	Jacobian matrix
h	element dimension length in 1D (m), area in 2D (m^2), volume in 3D (m^3)

H Heaviside function
H Hessian matrix
 $L(v)$ length of an element in the direction of a vector v
 M_i finite volume interpolation function i
 N_i finite element interpolation function i
n unit normal vector (w.r.t., FV facet in most cases)
 rst (superscript) = in reference space
s sector
S surface
V volume
w integration weight
X point
 xyz (superscript) = in global space
 ε finite element discretization error
 λ eigenvalue
 Ψ physical variable
 $\hat{\Psi}$ finite element approximation of Ψ
 $\tilde{\Psi}$ interpolated value of $\hat{\Psi}$ inside of finite element
 $\bar{\Psi}$ finite-volume approximation of Ψ

Physical variables from governing equations

c tracer concentration (kg m^{-3})
D dispersivity (m)
k permeability (m^2)
 μ dynamic fluid viscosity (Pa sec^{-1})
 ϕ porosity (X)
p fluid pressure (Pa)
q fluid source/sink ($\text{m}^{-3} \text{sec}^{-1}$)
u Darcy flux vector ($\text{m}^3 \text{sec}^{-1}$).

REFERENCES

- Al-Mahruqi SAS (2001) *Fracture Pattern and Fracture Propagation as a Function of Lithology*. PhD Thesis, Imperial College, London, 288 pp.
- Baliga BR, Patankar SV (1980) A new finite element formulation for convection diffusion problems. *Numerical Heat Transfer*, **3**, 393–409.
- Baliga BR, Patankar SV (1983) A control volume finite-element method for two dimensional fluid flow and heat transfer. *Numerical Heat Transfer*, **6**, 245–261.
- Barr AH (1984) Global and local deformations of solid primitives. *Siggraph Computer Graphics*, **18**, pp. 21–30.
- Beall MW, Shephard MS (1997) A general topology-based mesh data structure. *International Journal for Numerical Methods in Engineering*, **40**, 1573–1596.
- Beall MW, Walsh J, Shephard MS (2003) Accessing CAD geometry for mesh generation. *Proceedings, 12th International Meshing Roundtable*, Sandia National Laboratories, NM, USA, pp. 33–42.
- Belayneh M, Cosgrove JW (2005) Fracture pattern variation around a major field and their implications regarding fracture prediction using limited data: an example from the British Channel Basin. In: *The Initiation, Propagation and Arrest of Joints and Other Fractures* (eds Cosgrove JW, Engelder T), *Geological Society, London, Special Publication*, **231**, 89–102.
- Benzley SE, Perry E, Merkley K, Clark B (1995) A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elastic–plastic analysis. *Proceedings of the 4th International Meshing Roundtable*, Albuquerque, NM, USA, pp. 179–191.
- Berkowitz B, Scher H (1995) On characterization of anomalous dispersion in porous and fractured media. *Water Resources Research*, **31**, 1461–1466.
- Blackner T (2000) Meeting the challenge for automated conformal hexahedral meshing. *Proceedings of the 9th International Meshing Roundtable*, Sandia National Laboratories, NM, USA, pp. 11–19.
- Bogdanov, Mourzenko VV, Thovert JF, Adler PM (2003) Effective permeability of fractured porous media in steady state flow. *Water Resources Research*, **39**, 1023–1029.
- Burden RL, Faires JD, Reynolds AC (1978) *Numerical Analysis*. Prentice Hall, Englewood Cliffs, NJ.
- Cai Z, Jones JE, McCormick SF, Russell TF (1997) Control-volume mixed finite element methods. *Computational Geosciences*, **1**, 289–315.
- Caumon G, Lepage F, Sword CH, Mallet J-L (2004) Building and editing a sealed geological model. *Mathematical Geology*, **36**, 405–424.
- Chung TJ (2002) *Computational Fluid Dynamics*. Cambridge University Press, Cambridge, UK, 1036 pp.
- Cook RD, Malkus DS, Plesha ME, Witt RJ (2002) *Concepts and Applications of Finite Element Analysis*, 4th edn. John Wiley and Sons, Inc., New York, 240 pp.
- Cordes C and Kinzelbach W (1992) Continuous groundwater velocity fields and path lines in linear, bilinear, and trilinear finite elements. *Water Resources Research*, **28**, 2903–2911.
- Courant R (1942) Variational methods for the solution of problems of equilibrium and vibrations. *Transactions of the American Mathematical Society*, **49**, 1–23.
- De Boor C (1978) *A Practical Guide to Splines*. Springer-Verlag, Berlin.
- Dershowitz WS (1984) *Rock Joint Systems*, PhD Thesis, Massachusetts Institute of Technology, 987 pp.
- Durlofsky LJ (1993) A triangle based mixed finite element–finite volume technique for modelling two phase flow through porous media. *Journal of Computational Physics*, **105**, 252–266.
- Durlofsky LJ (1994) Accuracy of mixed and control volume finite element approximations to Darcy velocity and related quantities. *Water Resources Research*, **30**, 965–973.
- Eaton MD, Pain CC, De Oliveira CRE, Goddard AJH (2003) A high-order Riemann method for the Boltzmann transport equation. In: *Nuclear Mathematical and Computational Sciences: A Century in Review*. American Nuclear Society, Gatlinburg, TN, April 6–11.
- Ellens MS (1997) *Representation of and Modeling with Arbitrary Discontinuity Curves in Sculptured Surfaces*. PhD Thesis, Department of Computer Science, University of Utah.
- Ewing RE, Spagnuolo AM (2003) Difficulties and uncertainty in mathematical/numerical modeling fluid flow in fractured media. *Fracture and In-situ Stress Characterization of Hydrocarbon Reservoirs*, *Geological Society of London*, **209**, 187–200.
- Farin G (2001) *Curves and Surfaces for Computer Aided Geometric Design*, 5th edn. Morgan Kaufmann Editors, San Francisco, CA, 384 pp.
- Geiger S, Driesner T, Heinrich CA, Matthäi SK (2006a) Multiphase thermohaline convection in the earth's crust: I. A new finite element–finite volume solution technique combined with a new equation of state for NaCl–H₂O. *Transport in Porous Media*, **63**, 399–434.

- Geiger S, Driesner T, Heinrich CA, Matthäi SK (2006b) Multi-phase thermohaline convection in the earth's crust: II. Benchmarking and application of a finite element-finite volume solution technique with a NaCl-H₂O equation of state. *Transport in Porous Media*, **63**, 435–461.
- Helmig R (1997) *Multiphase Flow and Transport Processes in the Subsurface: a Contribution to the Modelling of Hydrosystems*, 1st edn. Springer-Verlag, Berlin, 367 pp.
- Huber R, Helmig R (2000) Node-centered finite volume discretizations for the numerical simulation of multi-phase flow in heterogeneous porous media. *Computational Geosciences*, **4**, 141–164.
- Huyakorn PS, Pinder GF (1983) *Computational Methods in Subsurface Flow*. Academic Press, New York, 473 pp.
- Juanes R, Samper J, Molinero, J (2002) A general and efficient formulation of fractures and boundary conditions in the finite element method. *International Journal for Numerical Methods in Engineering*, **54**, 1751–1774.
- Karimi-Fard M, Durlinsky LJ, Aziz K (2004) An efficient discrete fracture model applicable for general purpose reservoir simulators, SPE 79699. SPE Reservoir Simulation Symposium. *SPE Journal*, **9**, 227–236.
- Khawaja A, Kallinderis Y (2000) Hybrid grid generation for turbomachinery and aerospace applications. *International Journal for Numerical Methods in Engineering*, **49**, 145–166.
- Kim J-G, Deo MD (2000) Finite element, discrete-fracture model for multiphase flow in porous media. *AIChE Journal*, **46**, 1120–1130.
- Leveque RJ (2003) *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, Cambridge, 559 pp.
- Lipnikov K, Vassilevski Y (2004) Error estimates for Hessian-based mesh adaptation algorithms with control of adaptivity. *Proceedings of 13th International Meshing Roundtable*, Williamsburg, VA, USA, September 19–22, pp. 345–351.
- Liu A, Joe B (1994) Relationship between tetrahedron shape measures. *BIT*, **34**, 268–287.
- Lo SH (2002) Finite element mesh generation and adaptive meshing. *Progress in Structural Engineering and Materials*, **4**, 381–399.
- Matthäi, SK, Geiger S, Roberts SG (2001) *The Complex Systems Platform CSP3D3.0: User's Guide*, 3rd edn. ETH Research Reports, Zürich, Switzerland, 150 pp.
- Matthäi SK, Mezentsev A, Belayneh M (2005a) Control-volume finite-element two-phase flow experiments with fractured rock represented by unstructured 3D hybrid meshes. SPE93341. *Proceedings of SPE Reservoir Simulation Symposium*, Houston, TX, USA, 21 pp.
- Matthäi SK, Mezentsev A, Pain CC, Eaton MD (2005b) A high-order TVD transport method for hybrid meshes on complex geological geometry. *International Journal for Numerical Methods in Fluids*, **47**, 1181–1187.
- McMorris H, Kallinderis Y (1997) Octree-advancing front method for generation of unstructured surface and volume meshes. *American Institute of Aeronautics and Astronautics Journal*, **35**, 976–984.
- Monteagudo JEP, Firoozabadi A (2004) Control-volume method for numerical simulation of two-phase immiscible flow in two- and three-dimensional discrete-fractured media. *Water Resources Research*, **40**, 1–20. doi: 10.1029/2003WR002996.
- Moore D, Warren, J (1990) *Adaptive Mesh Generation I: Packing Space, Technical Report*. Department of Computer Science, Rice University, Houston, TX, USA.
- Moore D, Warren, J (1991) Bounded aspect ratio triangulation of smooth solids. *Proceedings: Symposium on Solid Modeling Foundations and CAD/CAM Applications*, Austin, TX, USA, 5–7 June, pp. 455–464.
- Owen SJ (1998) A survey of unstructured mesh generation technology. *Proceedings, 7th International Meshing Roundtable*, Dearborn, MI, USA, pp. 239–267.
- Pain CC, Umpheby AP, De Oliveira CRE, Goddard AJH (2001) Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. *Computational Methods Applied Mechanical Engineering*, **90**, 3771–3796.
- Reichenberger V, Jakobs H, Bastian P, Helmig R (2006) A mixed-dimensional finite volume method for two-phase flow in fractured porous media. *Advances in Water Resources*, **29**, 1020–1036.
- Requicha AAG, Rossignac JR (1992) Solid modelling and beyond. *IEEE Computer Graphics and Applications*, **12**, 31–44.
- Ruge J, Stüben K (1987) Algebraic multigrid. In: *Multigrid Methods, Vol. 3 of Frontiers in Applied Mathematics* (ed. McCormick SF), pp. 73–130. SIAM, Philadelphia.
- Schneider F, Wolf S, Fäille I, Gallouët T, Choueiri W (1992) Hydrocarbons migration in basin modelling: is the combined use of finite element and control volume possible? *3rd European Conference on the Mathematics of Oil Recovery*, Delft, Netherlands, pp. 17–19.
- Segall P, Pollard DD (1983) Joint formation in granitic rock of the Sierra Nevada. *Geologic Society of America Bulletin*, **94**, 563–575.
- Stüben K (2001) A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, **128**, 281–309.
- Taig IC (1961) Structural analysis by the matrix displacement method. Report S017, English Electric Aviation Report, UK.
- Thompson JF, Soni B, Weatherill N (1998) *Handbook of Grid Generation*. CRC Press, BocaRaton, FL, USA.
- Voronoi MG (1908) Nouvelles applications des parametres continues a la theorie des formes quadratiques. *Journal für die Reine und Angewandte Mathematik*, **134**, 198–287.
- Witherspoon, PA, Wang JSY, Iwai K, Gale JE (1980) Validity of cubic law for fluid flow in a deformable rock fracture. *Water Resources Research*, **16**, 1016–1024.
- Zgainski FX, Coulomb JL, Marechal J (1996) A new family of finite elements: the pyramidal elements. *IEEE Transactions on Magnetics*, **32**, 1393–1396.
- Zienkiewicz OC, Cheung YK (1965) Finite elements in the solution of field problems. *The Engineer*, **220**, 507–510.
- Zienkiewicz OC, Taylor RL (2000) *The Finite Element Method, Volume 1, The Basis*, 5th edn. Butterworth-Heinemann, Oxford, UK, 689 pp.

APPENDIX

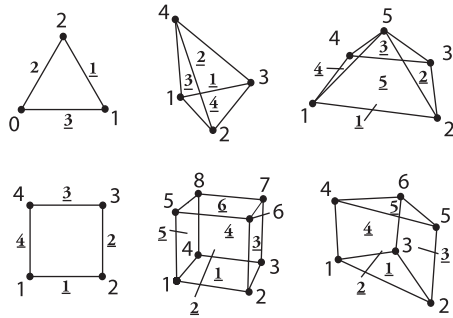


Fig. A. Face and node numbering for the isoparametric linear elements used in the hybrid element discretization. Face numbers are underlined.

Table A1 Node coordinates in parametric space. Node number are followed by their coordinates in brackets.

Element	Coordinates
Isoparametric linear bar	1 (−1); 2 (1)
Isoparametric linear triangle	1 (0,0); 2 (1,0); 3 (0,1)
Isoparametric linear quadrilateral	1 (−1, −1); 2 (1, −1); 3 (1,1); 4 (−1,1)
Isoparametric linear tetrahedron	1 (0,0,0); 2 (1,0,0); 3 (0,1,0); 4 (0,0,1)
Isoparametric linear prism	1 (0,0, −1); 2 (1,0, −1); 3 (0,1, −1); 4 (0,0,1); 5 (1,0,1); 6 (0,1,1)
Isoparametric linear pyramid	1 (−1, −1,0); 2 (1, −1,0); 3 (1,1,0); 4 (−1,1,0); 5 (0,0,1)
Isoparametric linear hexahedron	1 (−1, −1, −1); 2 (1, −1, −1); 3 (1,1, −1); 4 (−1,1, −1); 5 (−1, −1,1); 6 (1, −1,1); 7 (1,1,1); 8 (−1,1,1)

Table A2 Element interpolation functions and their first derivatives.

Element	Interpolation functions/derivatives
Isoparametric linear bar	$N = \begin{cases} (1-r)/2 \\ (1+r)/2 \end{cases} \quad \frac{dN}{dr} = \begin{cases} -1/2 \\ 1/2 \end{cases}$
Isoparametric linear triangle	$N = \begin{cases} 1-r-s \\ r \\ s \end{cases} \quad \frac{dN}{dr} = \begin{cases} -1 \\ 1 \\ 0 \end{cases} \quad \frac{dN}{ds} = \begin{cases} -1 \\ 0 \\ 1 \end{cases}$
Isoparametric linear quadrilateral	$N = \begin{cases} (1-r)(1-s)/4 \\ (1+r)(1-s)/4 \\ (1+r)(1+s)/4 \\ (1-r)(1+s)/4 \end{cases} \quad \frac{dN}{dr} = \begin{cases} -(1-s)/4 \\ (1-s)/4 \\ (1+s)/4 \\ -(1+s)/4 \end{cases} \quad \frac{dN}{ds} = \begin{cases} -(1-r)/4 \\ -(1+r)/4 \\ (1+r)/4 \\ (1-r)/4 \end{cases}$
Isoparametric linear tetrahedron	$N = \begin{cases} 1-r-s-t \\ r \\ s \\ t \end{cases} \quad \frac{dN}{dr} = \begin{cases} -1 \\ 1 \\ 0 \\ 0 \end{cases} \quad \frac{dN}{ds} = \begin{cases} -1 \\ 0 \\ 1 \\ 0 \end{cases} \quad \frac{dN}{dt} = \begin{cases} -1 \\ 0 \\ 0 \\ 1 \end{cases}$
Isoparametric linear prism	$N = \begin{cases} (1-r-s)(1-t)/2 \\ r(1-t)/2 \\ s(1-t)/2 \\ (1-r-s)(1+t)/2 \\ r(1+t)/2 \\ s(1+t)/2 \end{cases} \quad \frac{dN}{dr} = \begin{cases} -(1-t)/2 \\ (1-t)/2 \\ 0 \\ -(1+t)/2 \\ (1+t)/2 \\ 0 \end{cases} \quad \frac{dN}{ds} = \begin{cases} -(1-t)/2 \\ (1-t)/2 \\ 0 \\ -(1+t)/2 \\ (1+t)/2 \\ 0 \end{cases} \quad \frac{dN}{dt} = \begin{cases} -(1-r-s)/2 \\ -r/2 \\ -s/2 \\ (1-r-s)/2 \\ r/2 \\ s/2 \end{cases}$
Isoparametric linear pyramid ($t \neq 1$)	$N = \frac{1}{4} \begin{cases} (1-r)(1-s)-t+rst/(1-t) \\ (1+r)(1-s)-t-rst/(1-t) \\ (1+r)(1+s)-t+rst/(1-t) \\ (1-r)(1+s)-t-rst/(1-t) \end{cases} \quad \frac{dN}{dr} = -\frac{1}{4} \begin{cases} -(-1+t+s)/(1-t) \\ (-1+t+s)/(1-t) \\ -(1-t+s)/(1-t) \\ (1-t+s)/(1-t) \end{cases} \quad \frac{dN}{ds} = -\frac{1}{4} \begin{cases} -(-1+t+r)/(1-t) \\ (1-t+r)/(1-t) \\ -(1-t+r)/(1-t) \\ (-1+t+r)/(1-t) \end{cases} \quad \frac{dN}{dt} = \frac{1}{4} \begin{cases} -1+rs/(1-t)+rst/(1-t)^2 \\ -1-rs/(1-t)-rst/(1-t)^2 \\ -1+rs/(1-t)+rst/(1-t)^2 \\ -1-rs/(1-t)-rst/(1-t)^2 \end{cases}$

Isoparametric linear pyramid ($t = 1$)

$$N = \frac{1}{4} \begin{pmatrix} (1-r)(1-s)-t \\ (1+r)(1-s)-t \\ (1+r)(1+s)-t \\ (1-r)(1+s)-t \\ 4t \end{pmatrix} \frac{dN}{dr} = \frac{1}{4} \begin{pmatrix} -1 \\ 1 \\ -1 \\ 1 \\ 0 \end{pmatrix} \frac{dN}{ds} = \frac{1}{4} \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \frac{dN}{dt} = \frac{1}{4} \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 4 \end{pmatrix}$$

Isoparametric linear hexahedron

$$N = \frac{1}{8} \begin{pmatrix} (1-r)(1-s)(1-t) \\ (1+r)(1-s)(1-t) \\ (1+r)(1+s)(1-t) \\ (1-r)(1+s)(1-t) \\ (1-r)(1-s)(1+t) \\ (1+r)(1-s)(1+t) \\ (1+r)(1+s)(1+t) \\ (1-r)(1+s)(1+t) \end{pmatrix} \frac{dN}{dr} = \frac{1}{8} \begin{pmatrix} -(1-s)(1-t) \\ (1-s)(1-t) \\ (1+s)(1-t) \\ -(1+s)(1-t) \\ -(1-s)(1+t) \\ (1-s)(1+t) \\ (1+s)(1+t) \\ -(1+s)(1+t) \end{pmatrix}$$

$$\frac{dN}{ds} = \frac{1}{8} \begin{pmatrix} -(1-r)(1-t) \\ -(1+r)(1-t) \\ (1+r)(1-t) \\ (1-r)(1-t) \\ -(1-r)(1+t) \\ -(1+r)(1+t) \\ (1+r)(1+t) \\ (1-r)(1+t) \end{pmatrix} \frac{dN}{dt} = \frac{1}{8} \begin{pmatrix} -(1-r)(1-s) \\ -(1+r)(1-s) \\ -(1+r)(1+s) \\ -(1-r)(1+s) \\ (1-r)(1-s) \\ (1+r)(1-s) \\ (1+r)(1+s) \\ (1-r)(1+s) \end{pmatrix}$$