N

Numerical Methods for Flow in Fractured Porous Media

Luca Formaggia[1], Alessio Fumagalli[2] and Anna Scotti[2]
[1]MOX, Department of Mathematics, Politecnico di Milano, Milan, Italy
[2]Department of Mathematics, Politecnico di Milano, Milan, Italy

Definition

Porous medium: A material containing voids (pores), whose size is small compared to the size of the sample.
Fracture: a break in a material with a small thickness compared to its global extension.
Fracture network: a network composed of several intersecting fractures.

Introduction

Fractures are ubiquitous in porous media. Here, with the term fracture we denote a void in the porous material that has the following characteristics:

(i) One of its dimensions, the aperture, is orders of magnitude smaller than the other dimensions and the size of the domain of interest, but still large compared to pore size. We will indicate with extension the size of a fracture in the directions orthogonal to the aperture. The extension of fractures in a network has a distribution that is usually assumed to be governed by a power law, which implies the presence of a large variation of space scales. See Fig. 1.
(ii) Fractures may be either open or infilled by a material, whose physical characteristics may be strongly different from those of the surrounding material.
(iii) Fractures usually form networks, called fracture networks, often highly connected.

With the previous definitions one can consider as fractures, depending on the scale of interest, different objects such as: tectonic faults at the scale of sedimentary basins, cracks in glaciers, as well as fractures in concrete or in rocks.

When they are empty or filled with highly permeable materials, fractures may provide a preferential path to fluid flow, but in some cases the deposits inside fractures can become nearly impermeable. We refer to the latter situation as blocking fracture.

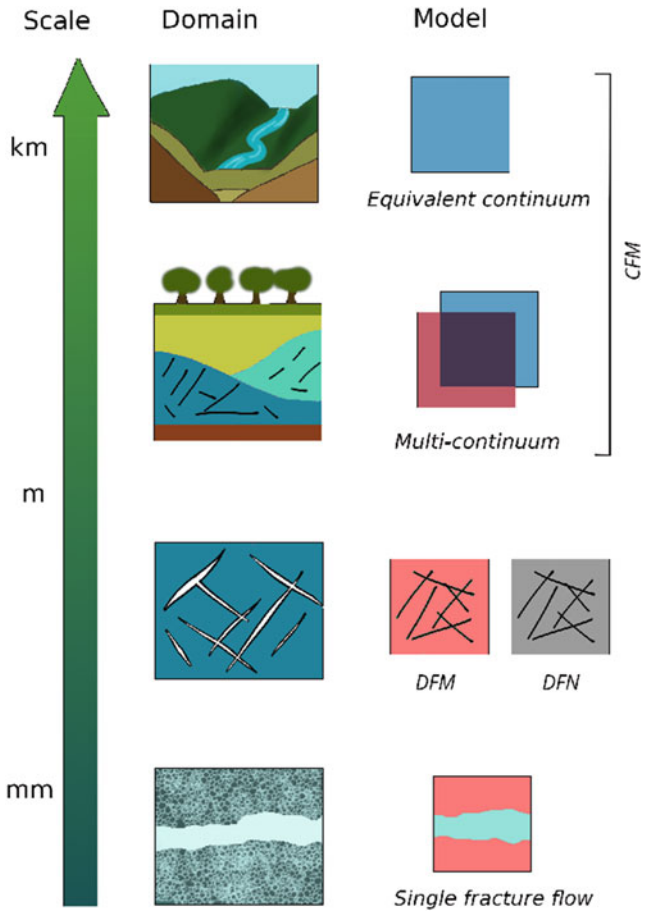
The presence of fractures greatly alters the macroscopic properties of the material in a complex way, in particular its mechanical and flow characteristics. We will be here concerned with the second aspect, and specifically on the mathematical modeling and computing techniques that may be adopted in the presence of fractured porous media. For a more general treatment of fractures in porous media the reader may refer to the book "Fractured Porous Media" (Adler et al. 2012).

The irregular spatial distribution of fractures and the presence of multiple scales make it difficult and often impossible to account for fractures by deriving effective upscaled parameters, permeability for instance, by volume averaging or homogenization techniques. Indeed, such procedures assume a strong separation of scales. Therefore, methods have been developed that model fractures explicitly, at least those crucial for the flow. These models are based on the assumption that in the porous medium flow is governed by Darcy's law, and often a similar model is adopted for the flow taking place in the fractures as well.

We recall the main characteristic of the Darcy's model, considering, for simplicity, just the case of single-phase flow. In this mathematical framework the two main variables are the pressure p and the macroscopic velocity field u, also called Darcy's velocity. The two quantities are related by Darcy's law (Bear 1972; Helmig 1997),

$$\mu \boldsymbol{K}^{-1} u + \nabla p = \mathbf{0} \quad \text{in } \Omega, \tag{1a}$$

H. K. Gupta (ed.), *Encyclopedia of Solid Earth Geophysics*,
https://doi.org/10.1007/978-3-030-58631-7



Numerical Methods for Flow in Fractured Porous Media, Fig. 1 Choice of the numerical model for domains at different space scales

where $\Omega \subset \mathbb{R}^d$ represents the domain occupied by the porous material and ∇ indicates the gradient. In the case where gravity effect are relevant, Eq. (1a) may be modified by replacing the pressure term with $p - \rho g z$, where ρ is the fluid density, g the magnitude of the gravity acceleration, and z is the vertical coordinate pointing upwards from the Earth surface. The main hypotheses behind the model are that fluid velocity is small, so we can neglect inertial effects, and the main model parameters are: μ, the fluid viscosity, and $\boldsymbol{K}$, the permeability tensor of the porous medium, which is a symmetric and positive definite tensor. Permeability may be heterogeneous in space and often with high variations.

The second equation expresses continuity of mass by the following differential equation (Bear 1972; Helmig 1997),

$$c\phi\partial_t p + \nabla \cdot u = q \quad \text{in } \Omega, \tag{1b}$$

where $\nabla\cdot$ is the divergence operator, q a source/sink term, c accounts for the medium and fluid compressibility, and ϕ is the porosity. Sometimes one is interested in the steady-state solution or the compressibility can be neglected, in which case $c\phi\partial_t p = 0$.

Equations (1a) and (1b) form a system of partial differential equations which, complemented by appropriate boundary and initial conditions, allows to describe the evolution of (u, p) in the porous medium.

Numerical Models for Fractured Porous Media

Numerical models for fractured porous media may be subdivided into two main categories: continuum fracture models (CFMs) and discrete fracture models (DFMs).

Continuum Fracture Models

CFMs are early models introduced in the 1960s (Warren and Root 1963), later justified mathematically in (Arbogast et al. 1990), and currently implemented in many industrial software. They assume a highly permeable and interconnected fracture network so that it can be modeled as a continuum superimposed to that of the porous medium.

A commonly adopted method is the dual-porosity/dual-permeability scheme. It assumes that at each point of the domain Ω we may use the Darcy's equations for flow in the fractures and in the porous medium, respectively, with a term representing the interchange of mass between them. A basic model of this type may be written as:

$$\begin{aligned} c_m\phi_m\partial_t p_m + \nabla \cdot u_m + \alpha(p_m - p_f) &= q_m \\ c_f\phi_f\partial_t p_f + \nabla \cdot u_f - \alpha(p_m - p_f) &= q_f \end{aligned} \quad \text{in } \Omega, \tag{2a}$$

where suffixes m and f refer to quantities related to the porous medium and fractures, respectively. The term $\alpha(p_m - p_f)$ represents the mass exchange between the two components, with α a rate parameter. The Darcy's velocities u_m and u_f are given by

$$\begin{aligned} \mu\boldsymbol{K}_m^{-1}u_m + \nabla p_m &= 0 \\ \mu\boldsymbol{K}_f^{-1}u_f + \nabla p_f &= 0 \end{aligned} \quad \text{in } \Omega. \tag{2b}$$

A simpler model, called dual-porosity/single-permeability, may be obtained formally by setting $\boldsymbol{K_m} = \boldsymbol{0}$. It assumes that the porous medium acts as storage volume for flow occurring only along fractures. A further simplification, appropriate only for highly connected networks of fracture of small extension relative to the domain size, consists in the use of a single equivalent continuum with upscaled properties that account for the combined effect of porous medium and fractures.

Discrete Fracture Models

DFMs represent fractures explicitly, modeled as a network of (typically planar) surfaces Γ, immersed in the porous medium. In the fractures we typically use a Darcy-type model where

some special source terms are added to account for the fluid exchange with the porous medium. The Darcy equations in the latter are also modified, with terms that act as interface conditions.

DFMs are computationally more demanding that CFMs, but also more accurate, particularly when fracture of relative large extension are present. For this reason the choice between CFM and DFM may depend on the spatial scale of interest, as illustrated in Fig. 1.

On each fracture we may identify a unit normal vector n_f, and thus a positive and a negative side of Γ, see Fig. 2. We indicate with $[\![f]\!] = f^+ - f^-$ the jump of a quantity f across the fracture.

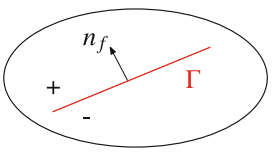
A commonly used model is derived in Martin et al. (2005), in which the fracture permeability is split into a normal, $K_{f,n}$, and tangential, $\boldsymbol{K}_{f,t}$, components to account for the fact that those quantities may be different and scale differently with the fracture aperture ϵ. In the porous medium $\Omega\backslash\Gamma$ we consider Eq. (1), while in the fractures we have, neglecting the compressibility term,

$$\begin{aligned} \mu \boldsymbol{K}_{f,t}^{-1} u_f + \epsilon \nabla_\tau p_f &= 0 \\ \nabla_\tau \cdot u_f - [\![u_m \cdot n_f]\!] &= q_f \end{aligned} \quad \text{in } \Gamma, \tag{3a}$$

where $\nabla_\tau \cdot$ and ∇_τ are the divergence and gradient operating on the tangent plane of the fractures, respectively. The jump of normal velocity across the fracture acts as a source term in the mass conservation equation and represent the net flux entering (or leaving) the fracture. This model must be complemented by appropriate boundary conditions. In the portion of the boundary of Γ that touches the boundary of Ω, the conditions are determined by the specific problem at hand: either pressure or mass flux is prescribed. In the case of a fracture tip that ends inside Ω, a zero mass flux condition is generally adopted.

A network normally exhibits intersection between fractures. A common approach is to assume that the pressure is continuous at the intersection and the net sum of fluxes is zero. More sophisticated models are available, to account for fractures with different hydraulic properties and flow along fracture intersections.

To close the problem we need also to specify interface conditions to couple Γ to the porous medium. In Martin et al. (2005) a family of models is presented, including the following, often adopted in practice



Numerical Methods for Flow in Fractured Porous Media, Fig. 2 Positive and negative side of Γ

$$\begin{aligned} \mu \epsilon K_{f,n}^{-1} u_m^+ \cdot n_f + 2(p_f - p_m^+) &= 0 \\ \mu \epsilon K_{f,n}^{-1} u_m^- \cdot n_f + 2(p_m^- - p_f) &= 0 \end{aligned} \quad \text{on } \Gamma. \tag{3b}$$

These conditions can be interpreted as the application of a discrete Darcy law across the fracture, and indeed they link the flux across the fracture to pressure differences.

Remark 1 If the fractures are highly permeable in the normal direction, continuity of pressure across Γ is often assumed, that is, $p_m^+ = p_m^-$. This induces a certain simplification in the model, since it does not account for a net mass flow across the fracture, but only for porous medium-fracture exchanges.

Remark 2 For nearly impermeable porous media, a different simplification of these models, called discrete fracture networks (DFNs), consists in neglecting the effect of the porous material and simulates flow just in the fracture network. They can be used in the presence of highly connected and permeable fractures.

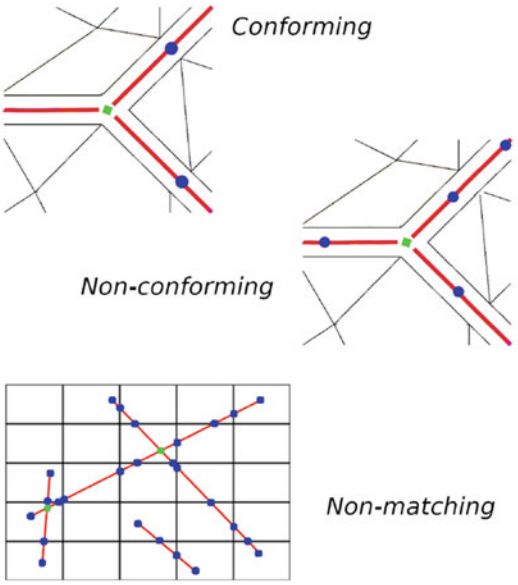
Discretization Schemes for DFM

Numerical schemes for the approximation of the equations presented in the previous section are based on partitioning the domain Ω into a mesh of polyhedral elements $\mathcal{T}_h$. The unknowns are then discretized by assuming a given variation inside each element, for instance constant or linear. The continuous solution is then replaced by the discrete values at the mesh nodes.

In the case of DFM we need to mesh both the porous medium and fracture domains and construct suitable ways to couple the two via (3b). Many discretization techniques available in the literature may be roughly subdivided into three categories, depending on the relation between porous medium and fracture grids, see Fig. 3.

(i) Conforming methods. The mesh for the porous medium conforms to that adopted for the fracture network. It means that a fracture mesh element coincides (geometrically) with the faces of two porous medium mesh elements, on the positive and negative side of the fracture. Consequently, no porous medium mesh elements are cut by the fracture. This requirement poses strong constraints on the mesh generation process, which indeed can be the most time-consuming part of the simulation, particularly in the presence of complex networks. On the other hand, the implementation of (3b) is rather straightforward.

(ii) Non-conforming methods. Still no elements in the porous medium grid are cut by the fractures: however, the grid at the two sides of the fracture and the fracture



Numerical Methods for Flow in Fractured Porous Media, Fig. 3 Porous medium and fracture grids

grid are independent. The implementation of (3b) involves the setup of suitable operators to transfer the discrete solution in the fractures to the porous medium grid and vice versa. The so-called mortar technique, which is based on the set up of additional variables at the interface, is sometimes used to simplify the construction of the transfer operators. The process of grid generation is eased and it is simpler to avoid the generation of highly distorted elements: however, it is still rather demanding in complex situations.

(iii) Non-matching methods. The mesh for the porous medium and the fracture network are completely independent and the fracture grids can cut the porous medium mesh elements in an arbitrary way. This simplifies mesh generation greatly, since porous medium and fracture grids can be generated independently. It is still necessary to find the intersections, but this is simpler than generating a conforming mesh and can be done with standard geometric search tools. However, implementation of conditions (3b) is more complex.

For every category, many different numerical schemes are at disposal. For the cases (i) and (ii), it is beneficial to use techniques able to operate on arbitrary polyhedral grids, like finite volumes or mimetic finite differences, to mention the more established one. The research in this field is, however, very active, and we mention also gradient schemes, the hybrid high order (HHO) method, and the virtual elements Method (VEM). A reference containing examples of numerical schemes applied in a DFM context is Fumagalli et al. (2019).

In the case (iii), as already stated, the main difficulty is how to impose the interface conditions. In that respect we have two classes of procedures. The first is to represent the possible jumps in the solution across the elements cut by a fracture explicitly. This is what is done in eXtended Finite Elements (XFEM), where the finite element basis functions are locally enriched to allow discontinuous solutions that can satisfy the coupling conditions (3b). Contrarily, if one accepts a less accurate representation of the solution, some manipulations of the interface conditions are possible to transform them into source terms acting both on the fracture elements and on the elements in the porous medium that are cut by the fracture. The resulting source terms have some similarity with those present in CFM techniques. The embedded discrete fracture model (EDFM), usually coupled with a simple finite volumes approximation, falls into this second category of methods. A reference on EDFM techniques is in Lee et al. (2000).

The performances of many classes of methods are presented and discussed in Flemisch et al. (2018) for bi-dimensional problems and Berre et al. (2020) for three-dimensional problems.

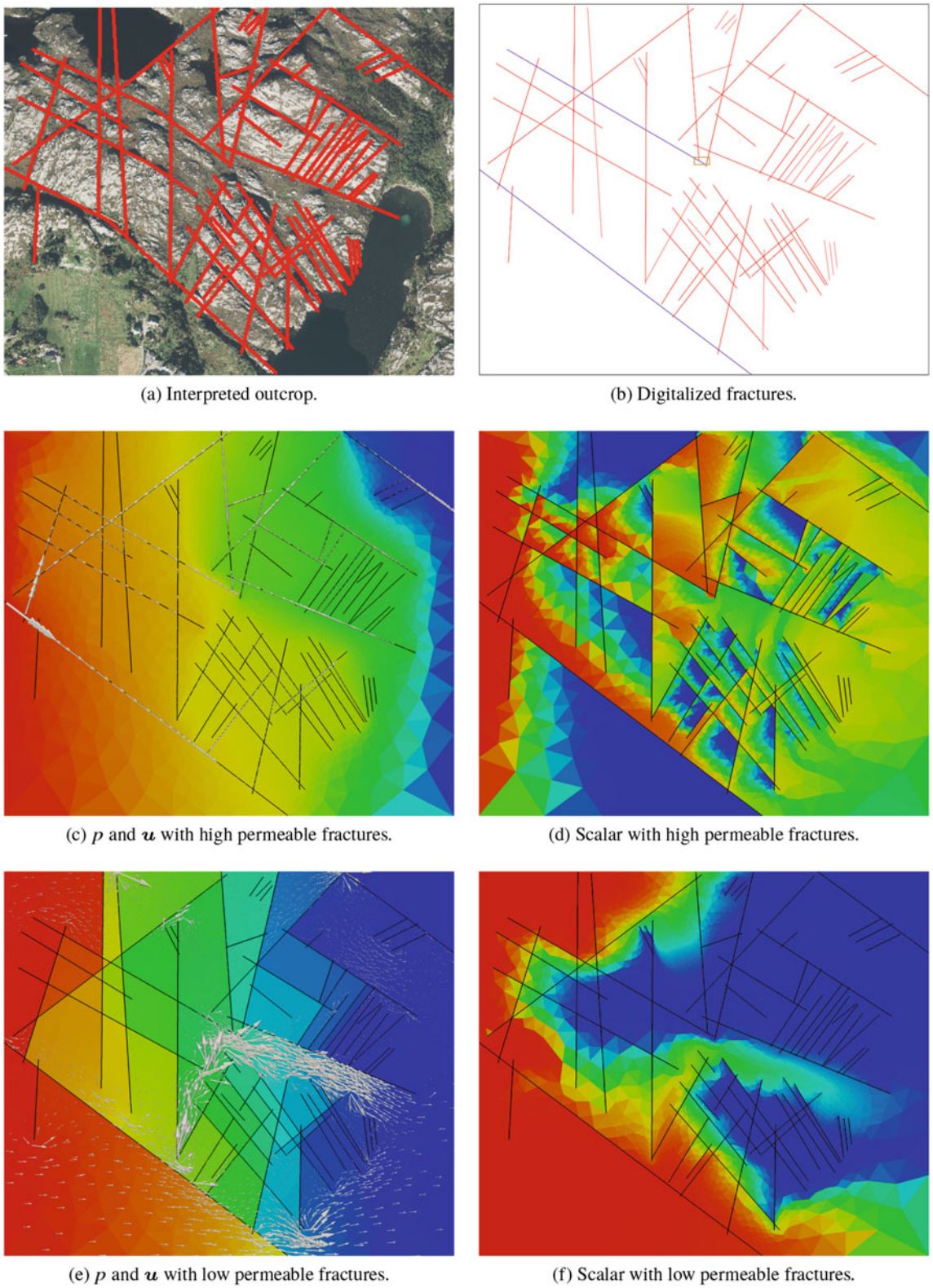
An Example of Computational Workflow

Numerical simulation of flow in fractured porous media is challenging due to the intrinsic geometrical complexity of the fractures, as well as the measurement of real fractures and their properties buried deep in the underground. These data are difficult to obtain and usually affected by large uncertainty which compromises the reliability of the numerical solutions.

Several approaches can be considered to detect fractures in the underground, from seismic inversion to outcrop interpretation, the former effective to detect big fractures few kilometers below the surface while the latter normally used as an analogue of the underground. Once the fractures are collected and digitalized by means of one of these methods, a suitable mathematical model can be adopted to perform the simulations.

We focus our attention on the case of outcrop interpretation. From highly detailed photographs of the interested region, all the fractures are collected and interpreted up to a minimum sampling scale determined by the quality of the data or by computational constraints. Smaller fractures can be accounted for by a suitable change of the porous medium properties and seen as upscaled or homogenized. What quantity defines a fracture "small" and thus not explicitly represented is still an open question, many authors consider the fracture extension a good proxy for its importance. After this operation, the digitalized version of the outcrop is thus available, see Fig. 4 on the top as an example where natural and human factors limit its exposure. The outcrop is a portion of Sotra Island, near Bergen in Norway.

Another challenging aspect is the collection of the physical data, again affected by uncertainty, that are needed to run the



(a) Interpreted outcrop. (b) Digitalized fractures.

(c) p and $\boldsymbol{u}$ with high permeable fractures. (d) Scalar with high permeable fractures.

(e) p and $\boldsymbol{u}$ with low permeable fractures. (f) Scalar with low permeable fractures.

Numerical Methods for Flow in Fractured Porous Media, Fig. 4 On the top the interpreted outcrop and digitalized fractures. The blue fractures are geometrically simplified due to software constraint. Images are taken from Flemisch et al. (2018). The others represent pressure, velocity field, and a scalar tracer at a specific time. The color scheme spans from the lowest in blue to the highest in red. (Images are taken from Fumagalli and Keilegavlen (2019). Under CC BY 4.0 licence)

simulation. If some data are not available, it is possible to fill the gap considering models from the literature to, for example, relate the permeability with the aperture and the latter with the fracture extension. The model can, at this point, be discretized numerically by means of one of the methods mentioned in the previous sections.

Figure 4 reports numerical results of a simulation, in particular the pressure and concentration of a tracer (for instance a contaminant) in the interpreted outcrop. Two extreme cases are considered, if the fractures are more or less permeable than the surrounding porous medium. Data are homogeneous and a left to right pressure gradient is imposed. The solutions are obtained with the library PorePy, see https://github.com/pmgbergen/porepy.

Conclusion

Modeling flow in porous media in presence of fractures and fracture networks is a challenging task. Several approaches are available in literature, and a few of them implemented in specialized software. The choice depends on the scales at which the phenomenon has to be considered, on the connectivity of the fracture network and on the level of accuracy desired. Simpler continuum fracture models are suitable for highly connected and dense networks of fracture, while in presence of fracture with a larger extension and more sparse, discrete fracture models provide more accurate results. Clearly, it is also possible to use a combination of the two approaches. In this entry we gave a rapid review of the different strategies and, for the sake of simplicity, we focused the attention on single-phase flow. The general conclusions can however be extended to the more complex situation of multiphase flows.

Cross-References

- ▶ Lithosphere, Mechanical Properties
- ▶ Numerical Methods, Finite Element
- ▶ Sedimentary Basins

Acknowledgments The authors acknowledge Davide Losa-pio for his help in the research underlying this manuscript.

Bibliography

Adler PM, Thovert JF, Mourzenko VV (2012) Fractured porous media. Oxford University Press, Oxford

Arbogast T, Douglas J Jr, Hornung U (1990) Derivation of the double porosity model of single phase flow via homogenization theory. SIAM J Math Anal 21(4):823–836. https://doi.org/10.1137/0521046

Bear J (1972) Dynamics of fluids in porous media. American Elsevier, New York

Berre I, Boon WM, Flemisch B, Fumagalli A, Gläser D, Keilegavlen E, Scotti A, Stefansson I, Tatomir A, Brenner K, Burbulla S, Devloo P, Duran O, Favino M, Hennicker J, Lee IH, Lipnikov K, Masson R, Mosthaf K, Nestola MGC, Ni CF, Nikitin K, Schädle P, Svyatskiy D, Yanbarisov R, Zu-lian P (2020) Verification benchmarks for single-phase flow in three-dimensional fractured porous media. Tech Rep. arXiv:2002.07005 [math.NA]. https://arxiv.org/abs/2002.07005

Flemisch B, Berre I, Boon W, Fumagalli A, Schwenck N, Scotti A, Stefansson I, Tatomir A (2018) Benchmarks for single-phase flow in fractured porous media. Adv Water Resour 111:239–258

Fumagalli A, Keilegavlen E (2019) Dual virtual element methods for discrete fracture matrix models. Oil Gas Sci Technol – Revue d'IFP Energies nouvelles 74(41):1–17

Fumagalli A, Berre I, Formaggia L, Keilegavlen E, Scotti A (eds) (2019) Numerical methods for processes in fractured porous media. Lecture notes in geosystems mathematics and computing. Birkhäuser, Basel

Helmig R (1997) Multiphase flow and transport processes in the subsurface: a contribution to the modeling of hydrosystems. Springer, Berlin

Lee S, Jensen C, Lough M (2000) Efficient finite-difference model for flow in a reservoir through discrete fracture networks and homogenized media. SPE Reserv Eval Eng 3:268–275

Martin V, Jaffré J, Roberts JE (2005) Modeling fractures and barriers as interfaces for flow in porous media. SIAM J Sci Comput 26(5):1667–1691

Warren EJ, Root PJ (1963) The behaviour of naturally fractured reservoir. Soc Petr Eng 3:245

Numerical Methods, Boundary Element

Michele Cooke
Geosciences Department, University of Massachusetts at Amherst, Amherst, MA, USA

Definition

The boundary element method is a numerical technique for investigating deformation within the Earth. The method is particularly adept at investigations of discontinuities in the crust, such as cracks, faults, and dikes. The numerical method is able to analyze complex problems by discretizing the surfaces of the discontinuities into small elements and solving the problem piecewise.

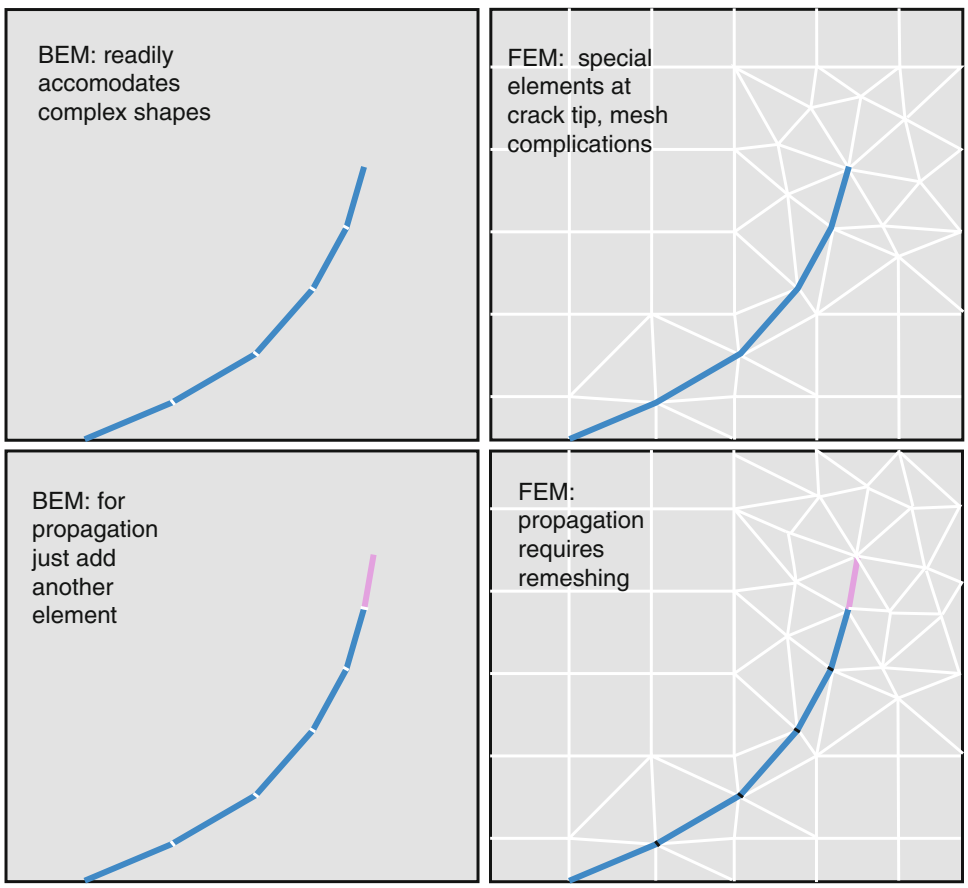
Introduction

The boundary element method (BEM) is one of several numerical methods used to investigate deformation within the Earth. The principal difference between boundary element

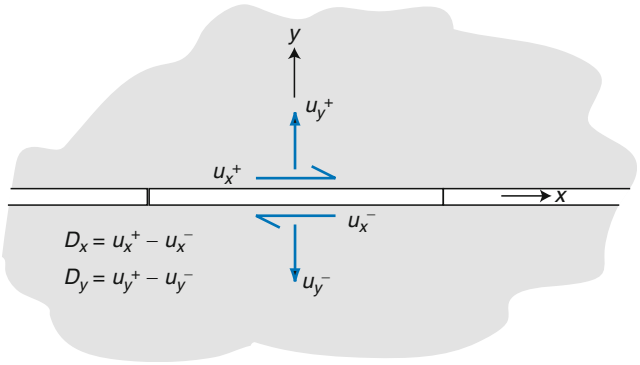
method and most other numerical methods of continuum mechanics, such as the finite element method (FEM), is that only the boundaries of the deforming body need to be meshed (Fig. 1). While other methods require meshing the entire volume of the deforming body into two-dimensional (2D) or three-dimensional (3D) elements, the BEM only requires discretization of the boundaries (e.g., fractures, bedding planes, or external boundaries) into one-dimensional (1D) or 2D elements. Minimizing discretization decreases model building and run time as well as errors due to discretization (Crouch and Starfield 1990). The greatest advantages of BEM modeling in geophysics are that (1) very complicated surface geometries, such as undulating fault surfaces, can be incorporated and (2) the method can readily accommodate fracture propagation by addition of new elements (Fig. 1). The disadvantages of BEM are (1) that the stiffness matrix relating stresses to strains on the elements is non-sparse so that inverting the matrix to find stresses can be very CPU intensive and (2) heterogeneous material properties are more complex to incorporate in BEM than methods that utilize volume discretization, such as FEM (Becker 1992).

How Does BEM Work?

BEM models are based on the solution to Kelvin's problem for the deformation due to a point load within a body. The solution provides the stresses and displacements everywhere due to the point load. We can calculate the effects of distributed loads by integrating the point force. When the loads are applied over discrete elements, the relationships between the displacements at one element due to loads at another are used to build an influence coefficient matrix. When either the displacements or tractions are prescribed to all elements that comprise the BEM model, the method will solve for the unknown element tractions or displacements via the influence coefficient matrix. Once all the displacements and tractions along the boundary elements are known, the stresses and strains at any point within the body can be calculated using the solution of Kelvin's problem. So unlike FEM, where solutions are calculated at each node within the element mesh, within BEM, problems displacements and stresses can be calculated at any point within the body, once all the tractions and displacements along the elements are known.



Numerical Methods, Boundary Element, Fig. 1 Contrasting discretization approaches of the boundary element method and finite element method. BEM only discretizes the boundaries and/or internal surfaces (*blue line*) rather than the entire volume. Fracture propagation is facilitated by addition of elements (*red line*)



Numerical Methods, Boundary Element, Fig. 2 Formulation of the displacement discontinuity along each element within a boundary element method model. The normal and shear displacement discontinuity (D_x and D_y, respectively) are the difference between the displacements along the top and bottom of the element

Two variations of BEMs are commonly used, the *fictitious stress* and *displacement discontinuity* methods. The fictitious stress method is based on Kelvin's problem and applies uniform tractions along each boundary element. In contrast, the displacement discontinuity method applies displacements along each element such that the difference in displacement form one side of the element to the other is constant along the boundary element (Fig. 2). While fictitious stress method works well for homogeneous bodies under external loads, the displacement discontinuity method is better suited for problems of solid bodies containing cracks (Crouch and Starfield 1990). For this reason, the displacement discontinuity method is more common for geophysical analyses, which investigate deformation associated with dikes, sills, microcracks, joints, and faults within the Earth.

What Can BEM Do?

Boundary element method codes using the displacement discontinuity formulation have been developed to solve both two-dimensional (e.g., Crouch and Starfield 1990 and three-dimensional problems (Okada 1992; Thomas 1993; Meade 2007). Two-dimensional formulations use a linear dislocation for each element while threedimensional formulations utilize either rectangular dislocations (Okada 1992) or angular dislocations (Cominou and Dundurs 1975) that can be assembled to form triangular elements of constant displacement discontinuity (Thomas 1993; Meade 2007). The advantage of triangular elements over rectangular elements is that complex nonplanar surfaces can be meshed without creating locations of overlaps and gaps.

Important refinements to the basic displacement discontinuity method for investigation of problems in geophysics include incorporation of frictional slip along crack elements and prevention of interpenetration of crack walls during compressive loading of cracks. These refinements facilitate investigation of deformation associated with subsurface faults and joints. Algorithms for preventing interpenetration include the penalty method (Crouch and Starfield 1990; Cooke and Pollard 1997) and the complementarity method (Maerten et al. 2010). The penalty method employs normal and shear stiffness to each element to prevent interpenetration. The complementarity method uses an iterative solver to solve the inequality that opening should be equal to or greater than zero along each element (Maerten et al. 2010).

Heterogeneous properties can be incorporated within BEM formulations by way of a contact boundary. Stresses and displacements are prescribed to be uniform across the boundary between otherwise homogeneous portions of the model. With this constraint along the boundary, the solution for each homogeneous section is found separately. This approach has not been widely applied because it is difficult to implement; finite element methods are generally believed to be better suited for heterogeneous properties.

Applications of the Boundary Element Method

In most BEM implementations, gravitational body forces are superposed onto the solution to the applied loading. This precludes direct investigation of problems related to topographic loading. However, Martel and Muller (2000) developed a formulation for analyzing topographic stresses using a long stress-free crack to simulate the topography.

Investigations of fluid flow through fracture networks has benefited from the surface discretization approach of the boundary element method. Models of subsurface flow through fracture networks utilize a boundary element method model that incorporates fluid flow along each element (e.g., Dershowitz and Fidelibus 1999). Such approaches can permit the full coupling of mechanical deformation and fluid flow.

The boundary element method has found widespread use in the subfield of structural geology for solving problems of deformation and propagation of opening mode and sliding mode fractures (e.g., joints and faults). Both forward and inverse modeling techniques have utilized the boundary element method. Forward models apply loading to the crack system to determine the resulting deformation of the system. Such investigations may explore the development of secondary cracks around a fault or the aperture of a hydraulic fracture. In contrast, inverse models start with the observed deformation and invert to find the slip on the fault or opening on the crack that must have produced the observations. For example, deformation observed along the surfaces of the earth via GPS stations velocities or InSAR imagery can be inverted to find the slip distribution on the underlying fault (e.g., Maerten et al. 2005).

Cross-References

- ▶ Numerical Methods, Finite Difference
- ▶ Numerical Methods, Finite Element

Bibliography

Becker AA (1992) The boundary element method in engineering: a complete course. McGraw-Hill, New York

Cominou MA, Dundurs J (1975) The angular dislocation in a half space. J Elast 5:203–216

Cooke ML, Pollard DD (1997) Bedding plane slip in initial stages of fault-related folding. J Struct Geol 19:567–581. Special Issue on Fault-Related Folding

Crouch SL, Starfield AM (1990) Boundary element method in solid mechanics with applications in rock mechanics and geological engineering. Unwin Hyman, Boston, 322 p

Dershowitz W, Fidelibus C (1999) Derivation of equivalent pipe network analogues for three-dimensional discrete fracture networks by the boundary element method. Water Resour Res 35(9):2685–2691

Maerten F, Resor P, Pollard D, Maerten L (2005) Inverting for slip on three-dimensional fault surfaces using angular dislocations. Bull Seismol Soc Am 95(5):1654–1665

Maerten F, Maerten L, Cooke ML (2010) Solving 3D boundary element problems using constrained iterative approach. Comput Geosci. https://doi.org/10.1007/s10596-009-9170-x [in press]

Martel S, Muller J (2000) A two-dimensional boundary element method for calculating elastic gravitational stresses in slopes. Pure Appl Geophys 157:989–1007. https://doi.org/10.1007/s000240050014

Meade BJ (2007) Algorithms for the calculation of exact displacements, strains, and stresses for triangular dislocation elements in a uniform elastic half space. Comput Geosci 33:1064–1075. https://doi.org/10.1016/j.cageo.2006.12.003

Okada Y (1992) Internal deformation due to shear and tensile faults in a half-space. Bull Seismol Soc Am 82:1018–1040

Thomas AL (1993) Poly3D: a three-dimensional polygonal-element displacement discontinuity boundary element computer program with applications to fractures, faults and cavities in the earth's crust. MS thesis, Stanford University, California

Numerical Methods, Domain Decomposition

Alfio Quarteroni[1,2] and Luca Formaggia[1]

[1]MOX, Department of Mathematics, Politecnico di Milano, Milan, Italy

[2]CMCS-MATHICSE, EPFL, Lausanne, Switzerland

Definition

Computational domain	A bounded spatial region where a partial differential equation is solved.
Dirichlet boundary condition	A condition imposed directly on the unknown of the differential problem, also called essential boundary condition.
Neumann boundary condition	A condition imposed on the fluxes associated to the differential problem at hand, that is, stresses, heat flux, etc. Also called natural boundary condition.
Parallel computer	A computer with more than one processing unit capable of computing operations concurrently.
Parallel computation	A numerical procedure executed on a parallel computer.
Preconditioner	In the context of the solution of a linear system by an iterative method, a preconditioner is an easily invertible matrix spectrally similar to the matrix governing the problem.

Introduction

Domain decomposition (DD) method is a technique for the solution of partial differential equations which can be instrumental to the development of *parallel computations*. It can be used in the framework of discretization methods, for example, finite elements, finite volumes, finite differences, or spectral element methods.

It is based on the reformulation of the given boundary-value problem on a partition of the computational domain Ω into M subdomains Ω_i, with $i = 1,\ldots, M$. Typically M is also the number of processors at disposal, even if it is possible to have more subdomains per processor.

The DD method also provides a convenient framework for the solution of heterogeneous or multi-physics problems, that is, those that are governed by differential equations of different kinds in different subregions of the computational domain. In this case, the subdomains conform with the subregions. In this work, however, we will address only homogeneous DD methods, yet some of the concepts presented here may be readily adapted for the nonhomogeneous case.

We will refer to a partial differential equation of the form

$$-div(\boldsymbol{T}(\boldsymbol{u})) = \boldsymbol{f} \text{ in } \Omega \tag{1}$$

with suitable boundary conditions on $\partial\Omega$.

For instance, in an elastostatic problem, $\boldsymbol{u}$ is the displacement and $\boldsymbol{T}(\boldsymbol{u})$ the stress tensor; in a heat conduction problem, $\boldsymbol{u}$ is a scalar variable, the temperature, and $\boldsymbol{T}$ the Laplace operator. With a rather similar formalism we may also account for the Stokes problem. The method is in fact applicable also to time-dependent problems, since in most cases the time advancing scheme required by the numerical simulation leads eventually to a sequence of problems of type (1). For instance, if we consider the evolution equation

$$\partial_t \boldsymbol{u} - div(\boldsymbol{T}(\boldsymbol{u})) = \boldsymbol{f},$$

a time discretization by the implicit Euler method leads to solve for the unknown $\boldsymbol{u}^n$ at each time step t^n a differential problem of the type

$$\boldsymbol{u}^n - \delta t div(\boldsymbol{T}(\boldsymbol{u}^n)) = \delta t \boldsymbol{f}^n + \boldsymbol{u}^{n-1}.$$

It is still of the form (1) with $div(\boldsymbol{T})$ replaced by $\delta t\ div\ (\boldsymbol{T}) - \boldsymbol{I}$, δt being the time step, and $\boldsymbol{I}$ the identity operator. A similar form is obtained also for the elastodynamic equations, discretized in time, for instance, by a Newmark method (Quarteroni and Valli 1994).

Typically, and for evident practical reasons, the partition into subdomains is made after having triangulated Ω into a finite element mesh τ_h, each subdomain being in fact formed by a set of elements of the original grid (see Fig. 1). The partition is often made automatically, using libraries like METIS (Karypis and Kumar 1998b) or PARMETIS (Karypis and Kumar 1998a), the latter able to exploit a parallel architecture also for this preprocessing stage.

More in particular, there are two ways of subdividing the computational domain into subdomains. The first uses disjoint subdomains, where $\Omega_i \cap \Omega_j = \emptyset$ for $i \neq j$. Here, the interface between subdomains reduces to surfaces (in 3D) and lines (in 2D). The second strategy adopts *overlapping subdomains*, usually built by letting the subdomains of an initial nonoverlapping partition grow of a certain factor δ. In most of the cases δ is dictated by the number of "layers" of grid elements that are added to the original partition. The minimal overlap is of one element, like that shown in Fig. 1. For a more complete review of DD methods one may refer to Smith et al. (1996), Quarteroni and Valli (1999), Wohlmuth (2001), Toselli and Widlund (2005), and Mathew (2008); several examples with emphasis on parallel computations are reported, for instance, in Bruaset and Tveito (2006).

Algorithms Without Overlap

We indicate with Γ_{ij} the interface between subdomains Ω_i and Ω_j and we set $\Gamma = \cup_{ij}\Gamma_{ij}$. We exploit the fact that problem (1) is equivalent to solving the following M coupled problems for $i = 1,\ldots, M$,

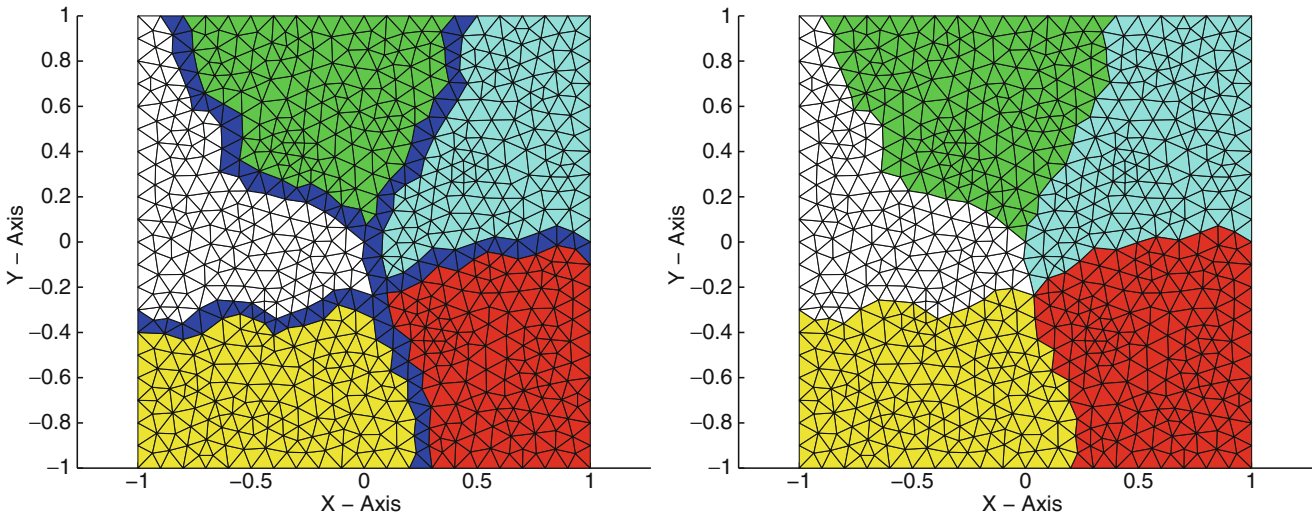
$$-div(\boldsymbol{T}(\boldsymbol{u}_i)) = \boldsymbol{f}_i \text{ in } \Omega_i \tag{2}$$

with the same boundary conditions of the original problem applied to $\partial\Omega_i \cap \partial\Omega$, while on each Γ_{ij} we set the continuity of the local solutions $\boldsymbol{u}_i$ and of the fluxes, that is,

$$\boldsymbol{u}_i = \boldsymbol{u}_j \qquad \boldsymbol{T}(\boldsymbol{u}_i)\cdot\boldsymbol{n}_{ij} + \boldsymbol{T}(\boldsymbol{u}_j)\cdot\boldsymbol{n}_{ji} = 0, \tag{3}$$

where $\boldsymbol{n}_{ij}$ is the normal to Γ_{ij} outward oriented w.r.t. Ω_i. It may be proved that $\boldsymbol{u}_i = \boldsymbol{u}_{|\Omega i}$ (Quarteroni and Valli 1999).

Problems (2) are coupled because of relations (3). A parallel algorithm may be obtained by an iterative procedure where conditions (3) are enforced in such a way to generate at each iteration decoupled problems that can be run on different processes, with only a small amount of communication needed at the beginning of each iteration. For the sake of space we give an example of just one of these procedures, called *Dirichlet-Neumann* for the case of two subdomains Ω_1 and Ω_2. Starting from a guess for $u_1^{(0)}$ and $u_2^{(0)}$, the algorithm solves for $k = 1, 2, \ldots$ the following sequence of independent problems,



Numerical Methods, Domain Decomposition, Fig. 1 Partition of a computational domain starting from a given triangulation. *Left:* A partition into subdomains with an overlap equal to one layer of elements. *Right:* Partition of a domain with disjoint subdomains

$$\begin{cases} -div\left(\boldsymbol{T}\left(\boldsymbol{u}_1^{(k+1)}\right)\right) = \boldsymbol{f}_1, & \text{in } \Omega_1, \\ \boldsymbol{u}_1^{(k+1)} = \theta u_2^{(k)} + (1-\theta)\boldsymbol{u}_1^{(k)}, & \text{on } \Gamma_{12} \end{cases} \text{ and}$$

$$\begin{cases} -div\left(\boldsymbol{T}\left(\boldsymbol{u}_2^{(k+1)}\right)\right) = \boldsymbol{f}_2, & \text{in } \Omega_2 \\ \boldsymbol{T}\left(\boldsymbol{u}_2^{(k+1)}\right)\cdot \boldsymbol{n}_{21} = \boldsymbol{T}\left(\boldsymbol{u}_1^{(k)}\right)\cdot \boldsymbol{n}_{21} & \text{on } \Gamma_{12} \end{cases}.$$

until a measure of the difference $u_1^{(k+1)} - u_2^{(k+1)}$ on Γ_{12} is below a given tolerance; θ here is a convenient relaxation factor. The Dirichlet-Neumann technique is not easily extendable to an arbitrary number of subdomains (unless suitable *coloring techniques* are used) and its convergence characteristics strongly depend on the geometry of the subdomains as well as on the possible jump of characteristic coefficients (for instance, the viscosity of different rocks forming a sedimentary basin). Other more favorable techniques like the Neumann-Neumann, Robin-Robin, and FETI methods are described in Quarteroni and Valli (1999), Wohlmuth (2001) and Toselli and Widlund (2005).

The DD method may be also set from an algebraic viewpoint. Indeed when discretized, for instance, by a finite element method, (2) and (3) reduce to a system of linear equations, which may be written in a block form as

$$\begin{pmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_I \\ \boldsymbol{u}_\Gamma \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_I \\ \boldsymbol{f}_\Gamma \end{pmatrix},$$

where A_{II} is a *block diagonal* matrix with M blocks of dimension equal to the number of unknowns internal to each subdomain, the latter being collected in $\boldsymbol{u}_I$, while $\boldsymbol{u}_\Gamma$ is the vector of unknowns on the interface Γ. If A_{II} is invertible (and normally this is the case), we may obtain a problem for the $\boldsymbol{u}_\Gamma$ only (*Schur complement system*), $\Sigma_\Gamma \boldsymbol{u}_\Gamma = \boldsymbol{\chi}_\Gamma$, where $\sum_\Gamma = A_{\Gamma\Gamma} - A_{\Gamma I}A_{II}^{-1}A_{I\Gamma}$ is the so-called *Schur complement matrix* w.r.t. $A_{\Gamma\Gamma}$. Having solved for $\boldsymbol{u}_\Gamma$ the computation of $\boldsymbol{u}_I$ can be done in a perfect parallel fashion by solving the block diagonal problem $A_{II}\boldsymbol{u}_I = \boldsymbol{f}_I - A_{I\Gamma}\boldsymbol{u}_\Gamma$.

A DD scheme with no overlap may be interpreted as *preconditioned iterative scheme* (Quarteroni and Valli 1994; Quarteroni 2009) for the Schur complement system, where the preconditioner can be efficiently applied in a parallel setting. A crucial issue for parallel computing is that of *scalability*. In the DD setting, an algorithm is said to be scalable if its convergence properties do not depend on the number of subdomains M, and in particular does not degrade if we keep the ratio M/N between number of subdomains and the total number N of unknowns of our problem fixed. Indeed in this case (if we neglect communication overheads), we may solve in the same time a problem twice as large by doubling the number of processors.

A scalable parallel preconditioner cannot be built using only local (i.e., at the subdomain level) approximations of the Schur matrix; we need to add also a *coarse operator* that has the role of transferring information among far away subdomains. The typical form of the preconditioner is (we write directly the inverse operator since it is the one actually required by the iterative procedure (Quarteroni et al. 2007))

$$P_\Sigma^{-1} = \sum_{i=1}^{M} R_i^T \sum_i^* R_i + R_0^T \sum_0^* R_0, \tag{4}$$

where R_i is a *restriction operator* that selects from all the $\boldsymbol{u}_\Gamma$ those local to the i-th subdomain, and $\sum_i^*$ is a local approximation of the inverse Schur matrix, which typically can be built using just data related to the i-th subdomain (and thus in parallel). Finally, $\sum_0^*$ is the coarse operator, of small dimension (typically of the order of M), whose role is to guarantee the coupling among all subdomains and it is necessary for scalability. The application of the preconditioner (i.e., the computation of $P_\Sigma^{-1}\boldsymbol{x}$, where $\boldsymbol{x}$ is any vector of the right length) can be done in parallel a part from the coarse operator, which however, being small in size, is irrelevant in the computational cost balance. Several scalable preconditioners are available, see Toselli and Widlund (2005) and Canuto et al. (2007).

Methods with Overlap

A typical DD scheme with overlapping subdomains is the Schwarz method. In its basic form, it is an iterative algorithm for the solution of (1) that for $k = 1, 2, \ldots$ solves a series of local problems on each subdomain Ω_i, where on the nonempty interfaces $\Gamma_{ij} = \partial\Omega_i \cap \Omega_j$, we apply Dirichlet boundary conditions using the latest values available from Ω_j, that is,

$$\begin{cases} -div\left(\boldsymbol{T}\left(\boldsymbol{u}_i^{(k+1)}\right)\right) = \boldsymbol{f}_i, & \text{in } \Omega_i, \\ \boldsymbol{u}_i^{(k+1)} = \boldsymbol{u}_j^{(k)}, & \text{on } \Gamma_{ij}. \end{cases}$$

The iteration continues until the difference between two successive iterations is sufficiently small. The convergence analysis of the Schwarz method may be found in Smith et al. (1996); Quarteroni and Valli (1999); Toselli and Widlund (2005), and Canuto et al. (2007) in the context of spectral element discretizations.

The method is seldom used in this form, however. Yet, it is probably the most adopted method to build parallel preconditioners for an iterative solver of the global problem, giving rise the so-called *Krylov-Schwarz* (for linear problems) and *Newton-Krylov-Schwarz* methods.

A Schwarz parallel preconditioner P_S may be written similarly to (4), that is,

$$P_S^{-1} = \sum_{i=1}^{M} R_i^T A_i^{-1} R_i + R_0^T A_0^{-1} R_0,$$

where here the restriction operator R_i extracts from a vector of length N the elements corresponding to the unknowns internal to the extended subdomain Ω_i, $A_i = R_i A R_i^T$ is the local matrix extracted from the matrix A of our problem, and finally A_0 is again a global coarse operator of size of the order M needed for scalability, and R_0 being the corresponding restriction matrix.

Again, apart from the coarse operator, the computation of $P_S^{-1}\boldsymbol{x}$ can be done in parallel since the matrices A_i are local and can be handled by each processor.

Application to Geophysical and Geological Problems

DD methods have been applied successfully in the context of geophysical problems, like acoustic and elastodynamic wave propagation (Faccioli et al. 1997), or full-waveform tomography (Sourbier et al. 2009).

It has been also applied to speed up the simulation of the evolution of sedimentary basins. Here, different types of sediments, such as gravel, sand, rocks, and biological remains that have been transported by the wind, the rivers, and, sometimes, by the sea, accumulate, are buried deeper and deeper, and are transformed eventually into rocks by a complex process of compaction and diagenesis.

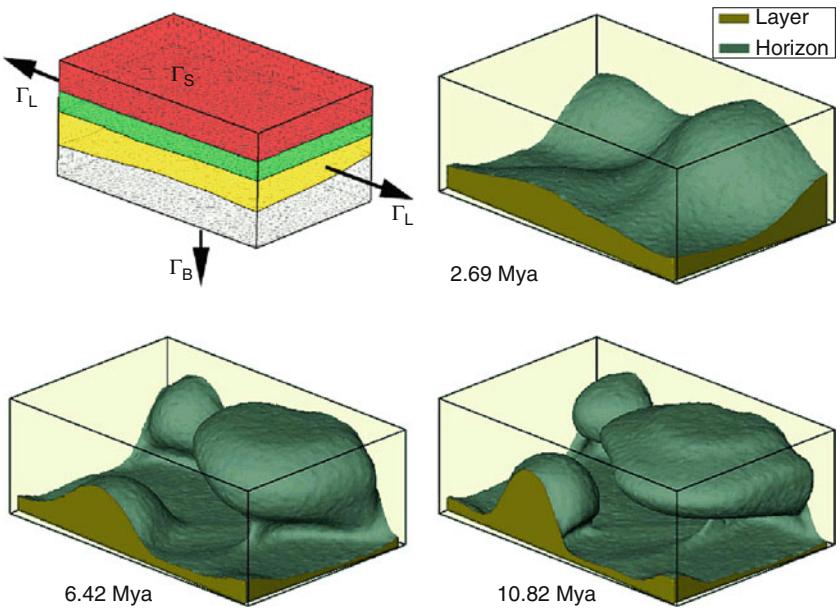
On geological scales, the evolution of a sedimentary basin may be tackled by a fluid approach where each rock layer is modeled as a viscous fluid. This is particularly convenient in the presence of salt tectonics (Massimi et al. 2007).

The computational domain Ω is split in several subdomains Ω_i, which usually correspond to the rock layers, as shown in Fig. 2. At each time step, we have to solve in the domain Ω a Stokes problem, possibly with non-Newtonian rheology, to compute the instantaneous velocity field,

$$\begin{cases} -div(\boldsymbol{T}(\boldsymbol{v})) + \nabla p = \boldsymbol{f}, \\ div(\boldsymbol{v}) = \phi, \\ \dfrac{\partial \rho}{\partial t} + (\boldsymbol{v}\nabla)\rho = 0. \end{cases} \tag{5}$$

Here, $\boldsymbol{v}$ is the velocity, p the pressure, $\boldsymbol{f}$ the external gravitational field, and ρ the density. The stress tensor $\boldsymbol{T}$ depends on characteristics of the rocks, such as viscosity and density, which may be discontinuous across layers. The function ϕ may account for compaction processes or may be simply set to zero when the hypothesis of isochoric flow is acceptable. The last equation describes the evolution of the density, which is simply advected by the velocity field. The presence of faults is accounted for by appropriately reducing the rock viscosity in the vicinity of the faults.

The movement of the layers has been tracked using a level set technique. A finite element scheme has been adopted for the discretization of the Stokes problem, while a conservative finite volume scheme has been used for the tracking of the layer interfaces. The parallel implementation has been carried out with the help of the TRILINOS library (Heroux et al.



Numerical Methods, Domain Decomposition, Fig. 2 The simulated evolution of a salt diapir. In the first graphic, the computational domain showing the subdivision of the layers. The other pictures show snapshots of the evolution of the salt layer with the formation of a diapir

2005), using a Schwarz algorithm with a coarse operator built by aggregation (Sala 2004).

Figure 2 shows the evolution of a salt dome. Salt is less compressible than the surrounding rock, so during the sedimentation process it ends having a smaller density than the overburden. We are then facing a Rayleigh-Taylor instability and any small perturbation will cause the salt to rise, with large movements that cause the so-called salt diapirism.

Cross-References

- ▶ Numerical Methods, Finite Element
- ▶ Numerical Methods, Multigrid
- ▶ Sedimentary Basins

Bibliography

Bruaset A, Tveito A (eds) (2006) Numerical solution of partial differential equations on parallel computers. Lecture notes in computational science and Engineering, vol 51. Springer, New York

Canuto C, Hussaini MY, Quarteroni A, Zang TA (2007) Spectral methods. Evolution to complex geometries and application to fluid dynamics. Springer, Berlin/Heidelberg

Faccioli E, Maggio F, Paolucci R, Quarteroni A (1997) 2D and 3D elastic wave propagation by a pseudo-spectral domain decomposition method. J Seismol 1(3):237–251

Heroux M et al (2005) An overview of the TRILINOS project. ACM Trans Math Softw 31(3):397–423. (TOMS)

Karypis G, Kumar V (1998a) A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J Sci Comp 20:359–392

Karypis G, Kumar V (1998b) METIS: unstructured graph partitining and sparse matrix ordering system. Techical report 98-036. University of Minnesota, Department of Computer Science

Massimi P, Quarteroni A, Saleri F, Scrofani G (2007) Modeling of salt tectonics. Comput Methods Appl Mech Eng 197(1–4):281–293

Mathew T (2008) Domain decomposition methods for the numerical solution of partial differential equations. Lecture notes in computational science and engineering. Springer, New York

Quarteroni A (2009) Numerical models for differential problems. MS&A, vol 2. Springer, Berlin

Quarteroni A, Valli A (1994) Numerical approximation of partial differential equations. Springer, Berlin/Heidelberg

Quarteroni A, Valli A (1999) Domain decomposition methods for partial differential equations. Oxford Science, Oxford

Quarteroni A, Sacco R, Saleri F (2007) Numerical mathematics, 2nd edn. Springer, Berlin/Heidelberg

Sala M (2004) Analysis of two-level domain decomposition preconditioners based on aggregation. Math Model Numer Anal 38(5):765–780

Smith B, Bjorstad P, Gropp W (1996) Domain decomposition, parallel multilevel methods for elliptic partial differential equations. Cambridge University Press, New York

Sourbier F, Operto S, Virieux J, Amestoy P, L'Excellent JY (2009) FWT2D: a massively parallel program for frequency-domain full-waveform tomography of wide-aperture seismic data–part 2: numerical examples and scalability analysis. Comput Geosci 35(3):496–514

Toselli A, Widlund O (2005) Domain decomposition methods – algorithms and theory. Springer series in computational mathematics, vol 34. Springer, New York

Wohlmuth B (2001) Discretization methods and iterative solvers based on domain decomposition. Lecture notes in computational science and engineering. Springer, Berlin

Numerical Methods, Finite Difference

Johan O. A. Robertsson[1] and Joakim O. Blanch[2]
[1]Institute of Geophysics, Department of Earth Sciences, ETH-Zürich, Zürich, Switzerland
[2]BHP, Houston, TX, USA

Definition

Finite-difference method. A method to approximate derivatives between neighboring points in a grid. The method can be applied to solve partial-differential equations, such as the wave equation.

Introduction

The finite-difference (FD) method is among the most commonly used methods for simulating wave propagation in a heterogeneous Earth. In this article, we describe the FD method for modeling wave propagation on Cartesian grids in acoustic, elastic isotropic, elastic anisotropic, as well as viscoacoustic/elastic media. The basic equations for wave propagation can be formulated in various formally equivalent ways. We will restrict our description to systems of first-order partial-differential equations with pressure/stress and particle velocities as wavefield variables. Due to its versatility, attractive stability, and dispersive properties, particularly for modeling wave propagation in elastic media, this formulation has been by far the most popular formulation over the last several decades. A more extensive review of the FD method is given by Moczo et al. (2007).

We start with a brief review of wave propagation theory and introduce some fundamental concepts including the basic structure (staggering) of the FD grid. In the following two sections, we discuss the choice of FD approximation and introduce two basic numerical properties of the FD method: numerical dispersion and stability. The last two sections are devoted to different boundary conditions and source implementations.

Theory of Wave Propagation and Fundamental Concepts

We focus on wave propagation in 3D Cartesian coordinates with coordinate axes (x, y, z). The special cases of 1D or 2D can easily be generalized from the expressions presented here, for example, in 2D, omitting terms or expressions containing components in the y-direction. The particle velocity wavefield is denoted by $\vec{v} = (v_x, v_y, v_z)$. In acoustic media, the second wavefield variable is the scalar quantity pressure, p, whereas

in elastic media it is the (symmetric) stress tensor, S. We will often be representing stresses and particle velocities using stress and strain vectors defined as (the so-called Voigt notation):

$$\vec{\sigma} = \left(\sigma_{xx} \;\; \sigma_{yy} \;\; \sigma_{zz} \;\; \sigma_{yz} \;\; \sigma_{xz} \;\; \sigma_{xy}\right)^T,$$

and

$$\vec{\varepsilon} = \left(\varepsilon_{xx} \quad \varepsilon_{yy} \quad \varepsilon_{zz} \quad 2\varepsilon_{yz} \quad 2\varepsilon_{xz} \quad 2\varepsilon_{xy}\right)^T,$$

where the strain vector is related to the particle velocity vector components through:

$$\frac{\partial \vec{\varepsilon}}{\partial t} = \left(\frac{\partial v_x}{\partial x} \;\; \frac{\partial v_y}{\partial y} \;\; \frac{\partial v_z}{\partial z} \;\; \frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y} \;\; \frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \;\; \frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y}\right)^T.$$

The system of equations for wave propagation consists of two (dependent) sets of equations. First, the equation of motion (or Newton's second law):

$$\rho \frac{\partial \vec{v}}{\partial t} = \nabla \cdot \mathrm{S} + \vec{f}, \tag{1}$$

where ρ is the density of the medium, $\nabla \cdot \mathrm{S}$ denotes divergence of the stress tensor and $\vec{f}$ is a point-force source (source formulations driving the FD simulations will be described in detail later). In an acoustic medium, only the diagonal elements of the stress tensor are non-zero and equal to the negative pressure $-p$.

The second set of equations for wave propagation is the constitutive stress–strain relation:

$$\frac{\partial}{\partial t}\vec{\sigma} = C \frac{\partial}{\partial t}\vec{\epsilon}, \tag{2}$$

where C is the 6-by-6 stiffness matrix. For an elastic isotropic medium, the stiffness matrix takes the form:

$$C = \begin{pmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix}, \tag{3}$$

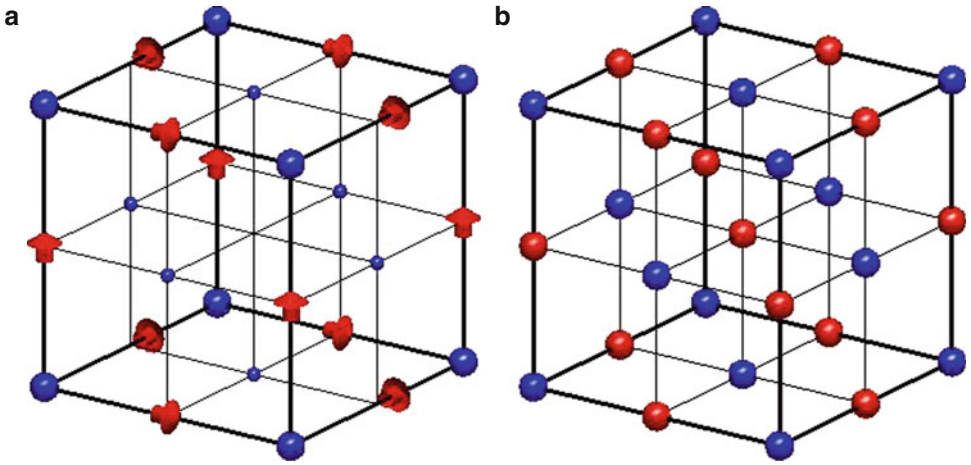
whereas in a general anisotropic medium, all components of the stiffness matrix may be non-zero. As we shall see, this has profound implications on the choice of grid geometry to discretize the constitutive relation. Finally, although redundant, the acoustic constitutive relation can also be written using the notation of the stiffness matrix and the stress–strain constitutive relation. As noted above, instead of a stress vector with three independent components, we obtain:

$$\vec{\sigma} = -\left(p \quad p \quad p \quad 0 \quad 0 \quad 0\right)^T. \tag{4}$$

In an acoustic medium, only the (3-by-3) upper left quadrant of the stiffness matrix is non-zero, with all elements equal and identical to the bulk modulus of the acoustic medium κ. Clearly this results in three identical equations – the familiar Hooke's law: $\frac{\partial}{\partial t} p = -\kappa \nabla \cdot \vec{v}$.

The FD method comprises solving Eqs. 1 and 2 by discretizing them in time and space and stepping forward in small incremental time steps. In particular, the exact choice of discretization in space turns out to be of fundamental importance. The so-called staggered grid was introduced for modeling isotropic elastic wave propagation by Virieux (1986) and Levander (1988, 1989). In the following, we will be referring to this as the Virieux grid. The choice of this grid results in schemes with attractive stability and dispersion properties, which other non-staggered grids do not necessarily possess. FD approximations to first-order derivatives naturally result in outputting the derivative of the wavefield at a location in between the discretized wavefield. By shifting the exact location of individual wavefield components by half a grid step in certain directions within each grid cell, it is possible to ensure that spatial derivatives can be computed exactly at the locations where they are needed to advance canonical wavefield quantities in time using Eqs. 1 and 2. The structure of the Virieux staggered grid is illustrated in Fig. 1a. The Virieux grid is also a good choice for modeling wave propagation in acoustic media. The resulting sparsely populated staggered-grid cell is equivalent to Fig. 1a but without the shear stresses populating the grid cell (small blue balls in Fig. 1a). In many situations, it is desirable to have a computational model that may contain both acoustic and elastic regions (e.g., modeling a marine seismic experiment or the seismic response of magma chambers). This is straightforward using a Virieux grid, by simply setting $\lambda = \kappa$ and $\mu = 0$ in Eq. 3 in the acoustic regions. Updating the equations will guarantee that the "acoustic stress vector" in Eq. 4 is satisfied to within-machine precision.

The Virieux grid is the natural choice for modeling wave propagation in elastic media as long as the stiffness matrix, C, in Eq. 2 belongs to a class of anisotropic materials referred to as orthorhombic media and as long as the symmetry axes of the medium are aligned with the Cartesian grid (isotropic media is a special case that belongs in this family of media). For this class of anisotropic materials, all elements that are zero in Eq. 3 remain zero. However, as soon as we introduce more complex anisotropic symmetry classes or rotate the symmetry axes with respect to the grid, the elements that are zero in Eq. 3 will become non-zero. As a consequence, Igel et al. (1995) noted that spatial derivatives are no longer



Numerical Methods, Finite Difference, Fig. 1 Staggered-grid geometries. (**a**) (*left*): Virieux staggered grid (1986). Large blue ball: normal stress components; *small blue ball*: shear stress components; red arrows: particle velocity components in the directions indicated. (**b**) (*Right*): Lebedev staggered grid (Lisitsa and Vishnevskiy 2010; Bernth and Chapman 2010). *Large blue ball*: all stress components; large red ball: all particle velocity components. The authors acknowledge Chris Chapman and Henrik Bernth (Schlumberger Cambridge Research) for providing the figure

available at all locations where they are needed. Igel et al. (1995) solved this by interpolating strains between their natural locations. However, the scheme is both expensive and results in fairly complex implementations. Saenger (2000) solved the problem in 2D by observing that a rotation of the staggered grid by 45° results in a natural choice for staggering the wavefield quantities so that no interpolation is necessary. The so-called rotated staggered scheme can also be generalized to 3D, although it can no longer be seen as a simple rotation of the Virieux grid. Another staggered grid that avoids the necessity to interpolate wavefield quantities in anisotropic media is the Lebedev grid (Lisitsa and Vishnevskiy 2010), which is illustrated in Fig. 1b. Bernth and Chapman (2010) analyzed and compared the different staggered-grid geometries and concluded that whereas the rotated staggered grid and the Lebedev grid are equivalent in 2D, the Lebedev grid is different and a better choice compared to the rotated staggered grid in 3D. The Lebedev grid is both computationally more efficient and also lends itself to a simpler implementation as it can be regarded as a combination of four independent Virieux grids that decouple in isotropic media.

So far, the discussion in this section has concerned lossless media. In order to account for attenuation, a viscoacoustic or viscoelastic model is introduced by using a different constitutive stress–strain relation compared to Eq. 2. Equation 2 can be thought of as a model of springs connecting particles in a lattice. When compressing the springs, energy is stored for an infinite time or until the compressed springs are released after which the particles return to their original positions. In a viscoelastic medium, dashpots are introduced in parallel or in series with the springs causing energy to dissipate after compression. The stiffness matrix becomes time-dependent and the multiplication with the time derivative of the strains is replaced by a convolution in time. The new viscoelastic constitutive relation becomes:

$$\frac{\partial}{\partial t}\vec{\sigma} = G(t) * \frac{\partial}{\partial t}\vec{\varepsilon}, \tag{5}$$

where $G(t)$ is the new relaxation stiffness matrix, which contains elements that are functions of time. Such a particularly useful function corresponds to a spring/dashpot configuration called a standard linear solid (Robertsson et al. 1994). First, due to the exponential kernel (in time) in the relaxation function of a standard linear solid, the convolution in Eq. 5 can be eliminated at the expense of introducing a new set of differential equations to be solved at each time step and spatial location. No additional complications due to various grid geometries arise after introducing the new differential equations solving for the so-called viscoelastic memory variables as these equations are fairly simple ordinary differential equations in time (as opposed to partial-differential equations with additional spatial derivatives). Second, Robertsson et al. (1994) and Blanch et al. (1995) showed how arrays of standard linear solid elements can advantageously be used to model constant quality factor Q versus frequency and also how to model attenuation of P and S waves in elastic isotropic media separately. However, exactly what attenuation to model for different wave types in an anisotropic medium is unclear and remains a topic for research (there are no pure P and S waves in an anisotropic medium).

Finite-Difference Approximations

The FD method approximates derivatives by combining neighboring function values on a grid, where the particular

combination is commonly derived using the Taylor expansion of the function at the different sample points. The simplest example of a FD approximation is the first derivative of a function p at x_0 using only two samples:

$$\frac{\partial p(x_0)}{\partial x} = \frac{1}{\Delta x}\left(p\left(x_0 + \frac{\Delta x}{2}\right) - p\left(x_0 - \frac{\Delta x}{2}\right)\right) + C_1 \frac{\partial^3 p(x_0)}{\partial x^3}(\Delta x)^2 + O\left((\Delta x)^4\right), \quad (6)$$

where Δx is the sampling interval and C_1 is a constant. The lowest-order error term in this expansion is proportional to the square of the sampling interval. Hence, as we decrease Δx, the FD approximation will become increasingly accurate (the increase in accuracy proportional to the square of Δx) and in the limit be equivalent to the first derivative of p. The approximation is therefore considered second-order accurate since the lowest-order error term depends on the square of the sampling interval. Higher-order approximations of the first derivative can be expressed as a weighted sum of additional adjacent sample points:

$$\frac{\partial p(x_0)}{\partial x} = \frac{1}{\Delta x}\sum_{i=1}^{M} \alpha_i \left(p\left(x_0 + \frac{(2i-1)\Delta x}{2}\right) - p\left(x_0 - \frac{(2i-1)\Delta x}{2}\right)\right) + O\left((\Delta x)^{2M}\right). \quad (7)$$

The coefficients α_i describe what is often called the FD stencil and are chosen such that terms containing lower-order error factors are canceled (Table 1).

The accuracy of the FD approximations for the temporal derivatives and the spatial derivatives are often different and schemes are accordingly named with two accuracy orders. An O(2,8) accurate scheme would be second-order accurate in time and eighth-order accurate in space.

From Eq. 6, we note that the derivative estimate of a spatial derivative is computed in between sample points of the wavefield quantity in question. As described in the previous section, this is the reason why the staggered-grid formulation is particularly well suited for FD modeling of the system of first-order differential equations describing wave propagation. The grid staggering is also applied in time, typically utilizing the simple second-order accurate approximation in Eq. 6, resulting in a time-stepping method referred to as a "leap-frog" scheme. The leap-frog scheme is in turn referred to as an explicit FD scheme since the differential equations only contain one unknown after discretization in space and time, and which, therefore, can be computed directly without solving further systems of equations. For example, if we are in the process of updating particle velocities at a new time step $t_0 + \Delta t$ using Eq. 1, we can compute the particle velocities directly from the known particle velocities at time t_0 and through spatial FD approximations of the known stresses at time $t_0 + \Delta t/2$. We then proceed to update stresses at time $t_0 = 3\Delta t/2$ using Eq. 2 explicitly from the stresses at $t_0 + \Delta t/2$ and particle velocities at $t_0 + \Delta t$. In this fashion, we march through the FD simulation until we reach the desired maximum time of the simulation. Clearly, we need to initiate the simulation with so-called initial conditions for particle velocities at time 0 and stresses at time $\Delta t/2$.

Numerical Methods, Finite Difference, Table 1 Examples of α_i for different accuracies

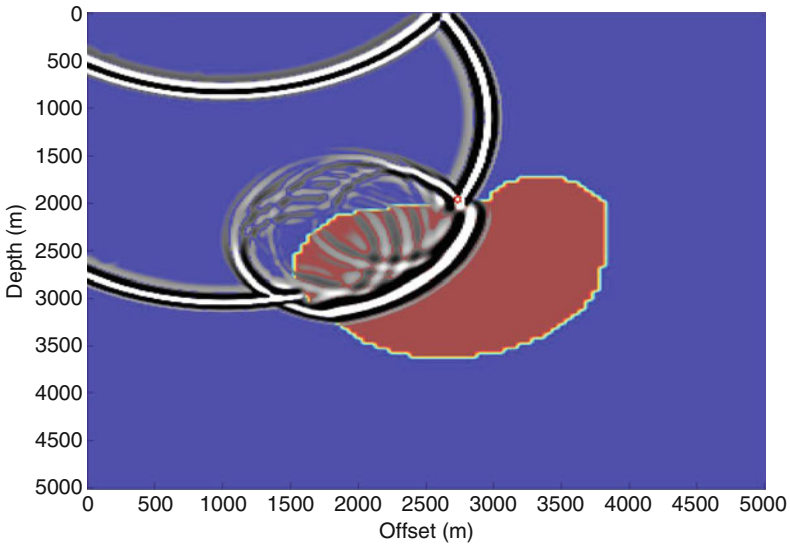
	α_i		
Order	$i = 1$	$i = 2$	$i = 3$
2	1	N/A	N/A
4	9/8	−1/24	N/A
6	75/64	−25/384	3/640

Accuracy and Numerical Dispersion

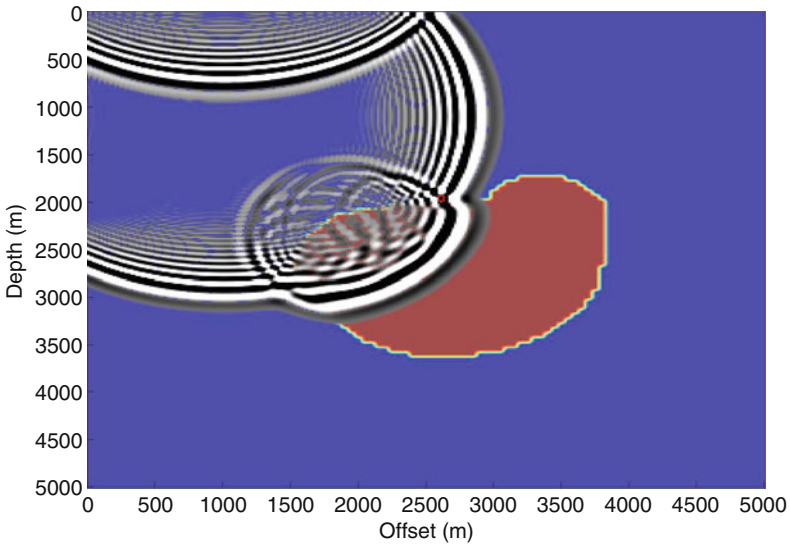
The most visible error introduced through the FD approximation is called numerical dispersion and appears when either the sampling interval is too large or the approximation is not of sufficiently high order. Numerical dispersion is defined as the effect when different frequencies of the wavefields propagate slower or faster compared to the correct speed of wave propagation in the medium. Figures 2 and 3 show wave propagation simulations with two different order schemes using the same spatial sampling and clearly demonstrate severe effects of numerical dispersion of the lower-order scheme.

The numerical dispersion can be predicted by Fourier transforming the FD expressions in time and space. Using the Fourier-transformed expressions, we obtain a relationship that shows how the propagation velocity in the medium is affected by numerical dispersion. It is a function of several parameters including the spatial grid step and the so-called Courant number (see stability section below). Figure 4 shows several such dispersion curves (phase velocity error as a function of the number of wavelengths per gridpoint) for different choices of time steps (resulting in different Courant numbers, see below). The numerical dispersion is dominated by the slowest wave speed in the medium, where the wavelength is the shortest. The number of sampling/gridpoints per wavelength necessary for a certain level of accuracy is thus determined by the dispersion curve and the slowest wave speed in a medium.

In practice, it is still difficult to define exactly how fine the wavefields must be sampled to avoid numerical dispersion as the severity of the numerical dispersion depends on the accuracy of the approximation and how far the wave has to propagate. However, as general rules of thumb, in order to roughly limit effects of numerical dispersion within a 2.5% error in propagation velocity, an O(2,4) scheme requires at



Numerical Methods, Finite Difference, Fig. 2 Snapshot in time of a simulation using the acoustic wave equation. The black-and-white wave front indicates the location of a travelling wave. The wave has been reflected in the upper free surface and changed polarity. The left side shows the radiating boundary condition, which in this case is a PML. The blue area has slower propagation velocity than the red area. The solution has been computed using an O(2,16) scheme



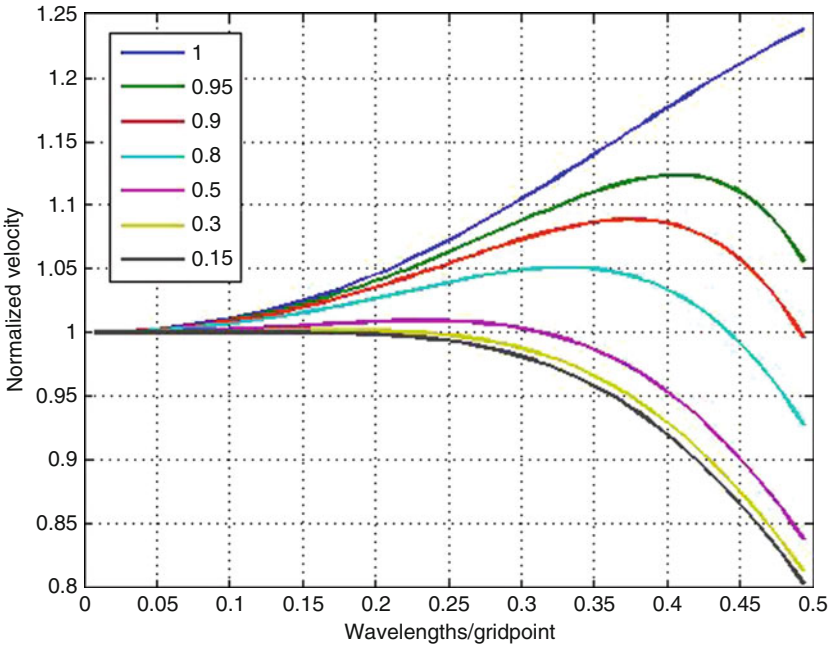
Numerical Methods, Finite Difference, Fig. 3 Similar snapshot in time to that shown in Fig. 2. The spatial sampling and medium is the same as in Fig. 2, but in this case the simulation has been performed with an O(2,2) scheme. The trailing ripples of the wave front are typical for numerical dispersion

least eight grid points per minimum wavelength, whereas an O(2,6) scheme requires six grid points per minimum wavelength.

It is obvious from Eq. 8 that a higher-order derivative approximation requires more computations than a lower-order approximation. However, in general higher-order approximations (up to some limit) are preferred as the computational requirements increase linearly with accuracy order, whereas a finer grid increases the computational requirement to the fourth power of the refinement in 3D. The fourth power dependence originates from the fact that the temporal step must be proportional to the spatial step (see Time Stepping and Stability below). Thus, if the spatial step is reduced by a factor n, then a factor n^3 more gridpoints are required in 3D in addition to another factor n more time steps, therefore resulting in an increase of a factor n^4 in the number of computations required.

Lax–Wendroff Corrections, Optimally Accurate FD Schemes and Time Dispersion Correction

The higher-order FD approximations are usually applied to spatial derivatives. To construct a functional scheme that has a higher order of accuracy in time, it is often necessary to use implicit time stepping, which results in a substantial increase in computational requirements. A solution to increase the formal order of accuracy (in a Taylor sense) without resorting to implicit schemes or saving several time levels is to apply a so-called Lax–Wendroff correction (Dablain 1986). The Lax–Wendroff correction works by using the system of equations to express higher-order derivatives in time as spatial derivatives. The approximations of the higher-order time derivatives can be used to construct correction terms to cancel errors of successively higher order. Using the system



Numerical Methods, Finite Difference, Fig. 4 Dispersion curves for a O(2,6) FD Scheme. A perfect solution should be equal to one. The dispersion curves vary depending on the ratio of the velocity of the medium and the maximum velocity used to determine the time step, that is, the (normalized) Courant number displayed in the legend. To yield a sufficiently accurate solution, this scheme should probably not be run with less than five to seven gridpoints per minimum wavelength (0.14–0.2 wavelengths/gridpoint)

$$\begin{cases} \dfrac{\partial p}{\partial t} = -\kappa \dfrac{\partial v}{\partial x} \\ \dfrac{\partial v}{\partial t} = -\dfrac{1}{\rho} \dfrac{\partial p}{\partial x} \end{cases}, \tag{8}$$

the corresponding Lax–Wendroff correction for a third-order time derivative is:

$$\frac{\partial^3 p}{\partial t^3} = \kappa \frac{\partial^3 v}{\partial x \partial t^2} = \kappa \frac{\partial}{\partial x} \frac{1}{\rho} \frac{\partial^2 p}{\partial x \partial t} = \kappa \frac{\partial}{\partial x} \frac{1}{\rho} \frac{\partial}{\partial x} \kappa \frac{\partial v}{\partial x}. \tag{9}$$

However, the method is somewhat cumbersome in multi-dimensions as it requires mixed derivatives and also derivatives of the material parameters, that is, ρ and κ in Eqs. 8 and 9 (see for instance Blanch and Robertsson (1997) for applications of the Lax–Wendroff correction).

The FD stencil can also be adjusted such that it is optimal for a certain frequency/wavelength range and thus achieves a higher accuracy than a standard scheme (Holberg 1987; Geller and Takeuchi 1995; Zingg 2000). Note that in a strict (and in this case misleading) Taylor sense, these schemes do not necessarily appear to be high-order schemes.

Fornberg (1987) noted that spatial and temporal dispersion can be considered as separate issues that can be minimized separately. Dispersion associated with the approximation of the time derivatives is not a function of the propagation path or type (e.g., P or S) of the synthesized data but only of the number of time-steps in the simulation since the source was introduced and can therefore be completely removed from FD computed data with a pre- and post-processing filtering procedure (Stork 2013; Koene et al. 2018). The computational cost of obtaining this spectral accuracy in time is negligible. It can also be used in combination with Lax-Wendroff related schemes (e.g., Amundsen and Pedersen 2017) which generally also allow for larger time-steps without violating the stability condition of the FD scheme. Thus, stability may be increased without increasing time dispersion.

Time Stepping and Stability

As discussed above, the time stepping is most commonly explicit using a second-order approximation resulting in a leap-frog scheme. The explicit time stepping introduces a so-called CFL (Courant–Friedrich–Levy) stability condition prescribing the maximum size of the time step Δt. The CFL condition can be derived from the same Fourier transformation as described above (in the context of numerical dispersion) and requiring the resulting difference equation to have roots of absolute value less than or equal to one. As a result, the maximum time step is determined by the fastest wave speed in a medium, the spatial step, and a constant depending on the particular FD approximations used:

$$\Delta t \leq \frac{K}{\sqrt{D}} \frac{\Delta x}{c_{\max}}, \tag{10}$$

where the constant K depends on the spatial accuracy and is slowly decreasing with increasing accuracy of the scheme from a value of 1 for an O(2,2) scheme, D is the dimension of the simulation (e.g., D = 2 for 2D) and $c_{\max}$ is the maximum wave propagation speed in the medium. Rapid changes

in material parameters may require an even smaller time step, but do not, in general, cause problems (Haney 2007). Using a temporal step larger than determined by the CFL condition yields an unstable simulation, that is, parasitic solutions will grow at an exponential rate and swamp the physical solution.

The Courant number is often used in the context of discussing stability and numerical dispersion. It is defined as

$$\gamma = c_0 \frac{\Delta t}{\Delta x}, \tag{11}$$

where c_0 is the local velocity of the medium. The Courant number can be interpreted physically as the fraction of the spatial increment that a wave can advance in a time step. From Eq. 11, we see that the Courant number controls the stability of the numerical scheme. In a heterogeneous medium, there exists a large range of velocities and it is thus important to study a FD scheme's behavior (i.e., numerical dispersion) for a range of Courant numbers (e.g., see Fig. 4). In the literature, the normalized Courant number is often used:

$$\gamma_{norm} = c_0 \frac{\Delta t}{\Delta x} \frac{\sqrt{D}}{K}, \tag{12}$$

which needs to be less than or equal to 1 for a stable simulation.

Amundsen and Pedersen (2017) showed that through a Lax-Wendroff related procedure it is possible to choose time steps that can be doubled, tripled, or generally, n-tupled beyond the classical CFL limit. However, like the Lax-Wendroff procedure, they require more computational effort per time step.

Boundary Conditions

The most important boundary conditions for wave simulations are radiating and free-surface boundary conditions. Radiating boundary conditions are applied at the edge of the finite computational domain to annihilate undesired reflections from the edge of the domain. The radiating boundary conditions can be implemented as "one-way" wave equation propagators using special operators at the boundaries. A first-order simple version is to implement the following equation at the boundary:

$$\frac{\partial p}{\partial t} = \pm c_0 \frac{\partial p}{\partial x}, \tag{13}$$

where the sign depends on which boundary (maximum or minimum x coordinate) and c_0 is the (local) propagation velocity. For a single propagation velocity, Eq. 13 is fairly straightforward to implement. However, significant complications arise for elastic or anisotropic media and for higher dimensions than 1D where it is the apparent velocity perpendicular to the boundary that is the relevant parameter (Clayton and Engquist 1977; Higdon 1986, 1990).

The radiating boundary conditions are more commonly implemented as absorbing (or attenuating) boundary conditions applied in a finite region just inside the computational domain (Cerjan et al. 1985). Absorbing boundaries do, however, need to have a transition zone, where the attenuation of the wavefield is gradually increased, since a rapid change in attenuation will cause reflections as well. A particularly efficient absorbing boundary condition is the "Perfectly Matched Layer" (PML) (Bérenger 1994; Collino and Tsogka 2001). The PML allows the attenuation to be increased extremely rapidly without causing reflections. The PML boundary conditions can be thought of as introducing artificial anisotropy to the boundary and matching the reflection coefficient in different propagation directions. The most straightforward realization of a PML boundary is through the split-field technique, where the wavefield is split depending on which spatial derivatives are used to update it. For the system in Eq. 8 in two dimensions, an implementation for the p variable would be

$$\begin{cases} \frac{\partial p_x}{\partial t} + \beta_x p = -K \frac{\partial v_x}{\partial x} \\ \frac{\partial p_z}{\partial t} + \beta_z p = -K \frac{\partial v_z}{\partial z} \\ p = p_x + p_z \end{cases} \tag{14}$$

The coefficients β_i control the attenuation in either the x or z directions independently.

The straightforward split-field implementation is equivalent to a fully anisotropic implementation (Teixeira et al. 2002) but suffers from a mild instability for waves impinging on the boundary at an incidence angle higher than 45°. A slightly more complex implementation avoids this instability (Bécache et al. 2002).

The free-surface boundary condition is used to approximate the interface between a water/solid material and air. Since the acoustic impedance of air is significantly different from that of water or the Earth, explicit modeling results in a computationally much more expensive simulation. In fact, it is possible to use a (pressure release) free-surface condition instead (water or Earth in contact with vacuum). For a water/air contact it is simply implemented through mirroring,

$$\begin{cases} p(-z) = -p(z) \\ v_z(-z) = v_z(z) \end{cases}, \tag{15}$$

assuming the boundary is located at $z = 0$. The upper boundary in Fig. 2 is implemented using a free-surface condition. The free-surface condition for a solid elastic material is more complicated and cannot be achieved solely through mirroring. Care must be taken to ensure that all wavefield components satisfy the wave equation and the free-surface condition in

order to accurately model wave phenomena such as Rayleigh waves at the free surface (see Robertsson 1996, for details).

Topography and Conformal Mapping of Grids

Topography can either be implemented explicitly into a FD scheme (Robertsson 1996) or by stretching the grid through conformal mapping of the regular computational grid onto a grid with continuously varying grid spacings where the top of the grid follows the topography (Fornberg 1988; Hestholm and Ruud 1994). The explicit method will lose some of its flexibility to implement rapidly varying topography if a higher-order scheme is used for the solution. If a conformal mapping is used, all derivatives will depend on the mapping function through the chain-rule. This may lead to stricter stability conditions and stricter requirements on spatial sampling.

Source Implementations

Seismic sources can be introduced in terms of body forces as indicated in Eq. 1 or as moment-tensor sources using an equivalent source term in the constitutive equation (e.g., Eqs. 2 or 5). A straightforward approach is to simply drive a body force at one point by feeding Eq. 1 with the value of a desired source wavelet in a single spatial grid point at each time step. Although this approach tends to work fairly well, researchers have sometimes reported on high-frequency noise being introduced. The origin of this noise has been attributed to the intrinsic difficulty in representing delta functions on a discretely sampled model with continuous wavefields.

Practitioners in exploration seismology often mimic explosive sources by equal excitation of the diagonal elements of the stress tensor at the source location or alternatively model compressional or shear vibroseis source by either exciting vertical or horizontal body forces. Earthquake seismologists who wish to model a moment-tensor source can do this either through a stress or body-force representation as described by Moczo et al. (2007).

An alternative way to introduce a source wavefield on a FD grid is by means of a boundary condition on an artificial surface interior to the grid. The method was first described by Alterman and Karal (1968) and followed by many authors clarifying the concept (Kelly et al. 1976; Levander 1989; Robertsson and Chapman 2000; Moczo et al. 2007). The technique relies on two prerequisites. First, that the spatial FD operators have a limited extent in space and second, that the (isolated) source wavefield can be computed in a region around the source location without reverberations from the rest of the model at the (simulated) time of source insertion.

The method of introducing a source wavefield as a boundary condition along an internal surface can also be used in seismic data processing and imaging applications (e.g., Amundsen and Robertsson 2014) and in so-called hybrid methods where two different numerical solutions are coupled together (e.g., Robertsson et al. 1996; Zahradník and Moczo 1996; Robertsson and Chapman 2000; van Manen et al. 2007). However, the wavefield on opposite sides of the artificial surface in the technique above is reversed so that the source wavefield is present inside the artificial surface only.

Finally, Robertsson et al. (2015) showed that by introducing a source wavefield as a boundary condition, it is possible to output up- and down-going as well as P- and S-wave constituents to machine precision accuracy. This enables the use of FD methods in research where such outputs are required instead of relying on so-called propagator matrix methods that are limited to 1D media.

Conclusions

We have outlined the basic principles of FD approximations to the systems of first-order partial-differential equations describing acoustic, isotropic elastic, anisotropic elastic, or viscoacoustic/elastic wave propagation. The method is based on explicit leap-frog time-stepping and staggered-grid representations. Most commonly Taylor-series-derived stencils are used to approximate the spatial derivatives, although other means to optimize FD stencils have also been described.

Two types of boundary conditions exist. First, radiating or absorbing boundary conditions (e.g., PML) are used to truncate the model while avoiding boundary reflections. Second, a free-surface boundary condition can be used to simulate the surface of the Earth with or without topography.

Sources are introduced either as point-force or stress sources, or along closed (artificial) surfaces inside the FD grid.

The maximum possible spatial grid size in order to limit numerical error is computed from the slowest propagation velocity in the model (e.g., a shear-wave velocity in the near-surface) using the so-called numerical dispersion relation for the FD scheme. Once the grid spacing has been chosen, the maximum possible time step is computed from the CFL-stability condition. The maximum possible time step is proportional to the spatial grid size and inversely proportional to the maximum propagation velocity in the grid. Since the computational cost of a 3D FD simulation increases to the power of four with a linear reduction of grid size, high-order accurate FD stencils are usually preferred.

Cross-References

- ▶ Earthquake Source Theory
- ▶ Magnetic Anisotropy
- ▶ Propagation of Elastic Waves: Fundamentals
- ▶ Seismic Imaging, Overview
- ▶ Seismic Viscoelastic Attenuation

► Seismic, Migration
► Seismic, Waveform Modeling and Tomography

Bibliography

Alterman Z, Karal FC (1968) Propagation of elastic waves in layered media by finite-difference methods. Bull Seismol Soc Am 58:367–398
Amundsen L, Pedersen Ø (2017) Time step n-tupling for wave equations. Geophysics 82:T249–T254
Amundsen L, Robertsson JOA (2014) Wave equation processing using finite-difference propagators, part 1: wavefield dissection and imaging of marine multicomponent seismic data. Geophysics 79:T287–T300
Bécache E, Petropoulos P, Gedney S (2002) On the long-time behavior of unsplit perfectly matched layers. INRIA report de Recherche, No 4538. ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-4538.pdf
Bérenger JP (1994) A perfectly matched layer for the absorption of electromagnetic waves. J Comput Phys 114:185–200
Bernth H, Chapman CH (2010) A comparison of Lebedev and rotatated staggered grids for anisotropic finite difference modeling. In: Extended abstracts from the 72nd EAGE conference and exhibition
Blanch JO, Robertsson JOA (1997) A modified Lax–Wendroff correction for wave propagation in media described by Zener elements. Geophys J Int 131:381–386
Blanch JO, Robertsson JOA, Symes WW (1995) Modeling of constant Q: methodology and algorithm for an efficient and optimally inexpensive viscoelastic technique. Geophysics 60:176–184
Cerjan C, Kosloff D, Kosloff R, Reshef M (1985) A non-reflecting boundary condition for discrete acoustic and elastic wave equations. Geophysics 50:705–708
Clayton RW, Engquist B (1977) Absorbing boundary conditions for acoustic and elastic wave equations. Bull Seismol Soc Am 6:1529–1540
Collino F, Tsogka C (2001) Application of the PML absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media. Geophysics 66:294–1307
Dablain MA (1986) The application of high-order differencing to the scalar wave equation. Geophysics 51:54–66
Fornberg B (1987) The pseudospectral method: comparison with finite differences for the elastic wave equation. Geophysics 52:483–501
Fornberg B (1988) Generation of finite difference formulas on arbitrary spaced grids. Math Comput 51:699–706
Geller RJ, Takeuchi N (1995) A new method for computing highly accurate DSM synthetic seismograms. Geophys J Int 123:449–470
Haney M (2007) Generalization of von Neumann analysis for a model of two discrete half-spaces: the acoustic case. Geophysics 72: SM35–SM46
Hestholm S, Ruud BO (1994) 2-D finite difference elastic wave modeling including surface topography. Geophys Prospect 42:371–390
Higdon RL (1986) Absorbing boundary conditions for difference approximations to the multi-dimensional wave equation. Math Comput 47:437–459
Higdon RL (1990) Radiating boundary conditions for elastic wave propagation. SIAM J Numer Anal 27:831–870
Holberg O (1987) Computational aspects of the choice of operator and sampling interval for numerical differentiation in large-scale simulation of wave phenomena. Geophys Prospect 35:629–655
Igel H, Mora P, Riollet B (1995) Anisotropic wave propagation through finite-difference grids. Geophysics 60:1203–1216
Kelly KR, Ward RW, Treitel S, Alford RM (1976) Synthetic seismograms: a finite-difference approach. Geophysics 41:2–27
Koene EFM, Robertsson JOA, Broggini F, Andersson F (2018) Eliminating time dispersion from seismic wave modeling. Geophys J Int 213:169–180
Levander AR (1988) Fourth-order finite-difference P-SV seismograms. Geophysics 53:1425–1435
Levander AR (1989) Finite-difference forward modeling in seismology. In: James DE (ed) The encyclopedia of solid earth geophysics. Van Nostrand Reinhold, New York, pp 410–431
Lisitsa V, Vishnevskiy D (2010) Lebedev scheme for the numerical simulation of wave propagation in 3D anisotropic elasticity. Geophys Prospect 58:619–645
Moczo P, Robertsson JOA, Eisner L (2007) The finite-difference time-domain method for modelling of seismic wave propagation. In: Wu RS, Maupin V (eds) Advances in wave propagation in heterogeneous Earth, vol 48, Advances in geophysics (ed Dmowska R). Elsevier-Pergamon, Oxford, pp 421–516
Robertsson JOA (1996) A numerical free-surface condition for elastic/viscoelastic finite-difference modeling in the presence of topography. Geophysics 61:1921–1934
Robertsson JOA, Chapman CH (2000) An efficient method for calculating finite-difference seismograms after model alterations. Geophysics 65:907–918
Robertsson JOA, Blanch JO, Symes WW (1994) Viscoelastic finite-difference modeling. Geophysics 59:1444–1456
Robertsson JOA, Levander A, Holliger K (1996) Modeling of the Acoustic Reverberation Special Research Program deep ocean seafloor scattering experiments using a hybrid wave propagation simulation technique. J Geophys Res 101:3085–3101
Robertsson JOA, van Manen DJ, Schmelzbach C, van Renterghem C, Amundsen L (2015) Finite-difference modelling of wavefield constituents. Geophys J Int 203:1334–1342
Saenger EH (2000) Wave propagation in fracture media: theory and applications of the rotated staggered finite-difference grid. PhD thesis, Karlsruhe University
Stork C (2013) Eliminating nearly all dispersion error from FD modeling and RTM with minimal cost increase. In Extended abstracts from the 75th EAGE conference and exhibition
Teixeira FL, Moss CD, Chew WC, Kong JA (2002) Split-field and anisotropic medium PML-FDTD implementations for inhomogeneous media. IEEE Trans Microwave Theory 50:30–35
van Manen DJ, Robertsson JOA, Curtis A (2007) Exact wave field simulation for finite-volume scattering problems. J Acoust Soc Am 122:EL115–EL121
Virieux J (1986) P-SV wave propagation in heterogeneous media: velocity-stress finite-difference method. Geophysics 51:889–901
Zahradník J, Moczo P (1996) Hybrid seismic modeling based on discrete-wavenumber and finite-difference methods. Pure Appl Geophys 148:21–38
Zingg DW (2000) Comparison of high-accuracy finite-difference methods for linear wave propagation. SIAM J Sci Comput 22:476–502

Numerical Methods, Finite Element

J. N. Reddy
Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA

Definition

The finite element method is a numerical method, like the finite difference method, for solving differential equations arising in the study of physical phenomena. In the finite

element method, a given domain is viewed as a set of nonintersecting subdomains, called *finite elements,* and over each element the governing equation is approximated by any of the traditional variational methods (e.g., Ritz, Galerkin, and least-squares methods). The main reason behind seeking approximate solution on a set of elements is the fact that it is easier to represent a complicated function as a sum of simple polynomials. Of course, each individual segment of the solution should fit with its neighbors in the sense that the function and possibly its derivatives up to a chosen order are continuous along the interface between elements. This entry is a brief introduction to the finite element modeling of diffusion processes and the Navier–Stokes equations governing viscous incompressible fluids. Both phenomena play crucial role in modeling a variety of geological or geomechanics processes.

General Introduction

Scientists and engineers model (i.e., develop mathematical models and numerically simulate) natural phenomena with the objective of understanding it. Virtually every phenomenon in nature, whether aerospace, biological, chemical, geological, or mechanical, can be described in terms of algebraic, differential, and/or integral equations relating various quantities of interest. Determining the stress distribution in a mechanical structure with oddly shaped members and numerous stiffeners and subjected to mechanical, thermal, and/or aerodynamic loads, finding the concentration of pollutants in lakes and estuaries or in the atmosphere, predicting geophysical and geological events, and simulating weather in an attempt to predict the formation of tornadoes and thunderstorms are a few examples of many important practical problems that scientists and engineers are occupied with.

In the last four decades, the finite element method has emerged as a powerful computational tool for solving nonlinear partial differential equations over irregular domains with complex domain properties (Reddy 2015, 2019; Reddy and Gartling 2010; Bathe 1996; Belytschko et al. 2000). Applications of the method to geology, geophysics, and geomechanics problems are numerous (see, e.g., Reddy et al. 1982; Wickham et al. 1982; Bird 1989; Parsons 2002; Dyksterhuis et al. 2005 and references therein). This brief introduction is meant to provide some insight into the workings of the finite element method as applied to the Poisson equation as well as the Navier–Stokes equations that are used to model certain geomechanics processes in two dimensions.

Finite Element Model of 2-D Poisson Equation

Consider the problem of finding $u(x, y)$ such that the following partial differential equation (a generalized Poisson's equation) is satisfied

$$-\left[\frac{\partial}{\partial x}\left(a_{xx}\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(a_{yy}\frac{\partial u}{\partial y}\right)\right]=f(x,y)\text{ in }\Omega \quad (1)$$

where Ω is a two-dimensional domain with boundary Γ. Here a_{xx} and a_{yy} are material coefficients in the x and y directions, respectively, and $f(x, y)$ is the known source. For example, in a ground water flow problem u denotes the water head (i.e., velocity potential), a_{xx} and a_{yy} are the permeabilities in the x and y directions, respectively, and $f(x, y)$ is distributed water source. Equation 1 also arises in many areas of science and engineering.

In the finite element method, the domain $\overline{\Omega}=\Omega\cup\Gamma$ is divided into a set of subdomains $\overline{\Omega}^e=\Omega^e\cup\Gamma^e$, called finite elements. Any geometric shape qualifies as an element, provided that the approximation functions ψ_i^e can be derived uniquely for the shape.

Suppose that the dependent unknown u is approximated over a typical finite element $\overline{\Omega}^e$ by the expression

$$u(\mathbf{x})\approx u_h^e(\mathbf{x})=\sum_{j=1}^{n}u_j^e\psi_j^e(\mathbf{x}),\mathbf{x}=(x,y)\in\overline{\Omega}^e \quad (2)$$

where $u_h^e(\mathbf{x})$ represents an approximation of $u(\mathbf{x})$ over the element $\overline{\Omega}^e$, parameters u_j^e denote the values of the function $u_h^e(\mathbf{x})$ at a selected number of points (i.e., element nodes) in the element $\overline{\Omega}^e$, and ψ_j^e are the Lagrange interpolation functions associated with the element.

We seek to satisfy the governing differential equation in a weak-form sense, with the weight functions being the same as the approximation functions. The resulting finite element model, i.e., set of algebraic equations) is called the *weak-form Galerkin finite element model* (Reddy 2019). The weak form is

$$\begin{aligned}0=&\int_{\Omega^e}\left(a_{xx}\frac{\partial w_i^e}{\partial x}\frac{\partial u_h^e}{\partial x}+a_{yy}\frac{\partial w_i^e}{\partial y}\frac{\partial u_h^e}{\partial y}-w_i^e f\right)\mathrm{d}x\mathrm{d}y\\&-\oint_{\Gamma^e}w_i^e q_n\mathrm{d}s\end{aligned} \quad (3)$$

Suppose that u_h^e is represented over a typical finite element Ω^e by expression of the form 2. Substituting the finite element approximation 2 into the weak form 3, we obtain

$$\mathbf{K}^e\mathbf{u}^e=\mathbf{f}^e+\mathbf{q}^e\equiv\mathbf{F}^e \quad (4)$$

where the coefficients K_{ij}^e, f_i^e and q_i^e are defined by

$$K_{ij}^e = \int_{\Omega^e} \left(a_{xx} \frac{\partial \psi_i^e}{\partial x} \frac{\partial \psi_j^e}{\partial x} + a_{yy} \frac{\partial \psi_i^e}{\partial y} \frac{\partial \psi_j^e}{\partial y} \right) \mathrm{d}A + \oint_{\Gamma^e} h_c \psi_i^e \psi_j^e \mathrm{d}s$$
$$f_i^e = \int_{\Omega^e} f \psi_i^e dA, q_i^e = \int_{\Gamma^e} (\widehat{q}_n + h_c u_c) \psi_i^e \mathrm{d}s \tag{5}$$

We note that $K_{ij}^e = K_{ji}^e$ (i.e., $\mathbf{K}^e$ is symmetric).

Finite Element Models of the 2-D Navier–Stokes Equations

In this section, we develop the finite element models of steady flows of viscous, incompressible fluids in two-dimensional domains. The governing equations are the conservation of linear momentum and conservation of mass, expressed in terms of the Cartesian components

$$\rho \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) - \frac{\partial \sigma_{xx}}{\partial x} - \frac{\partial \sigma_{xy}}{\partial y} - f_x = 0 \tag{6}$$

$$\rho \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) - \frac{\partial \sigma_{xy}}{\partial x} - \frac{\partial \sigma_{yy}}{\partial y} - f_y = 0 \tag{7}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{8}$$

$$\sigma_{xx} = 2\mu \frac{\partial u}{\partial x} - P, \sigma_{yy} = 2\mu \frac{\partial v}{\partial y} - P,$$
$$\sigma_{xy} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \tag{9}$$

Here we present two different finite element models associated with Eqs. 6, 7, 8, and 9. The first one is a direct formulation in which the three equations in (*u, v, P*) are used in their original form. This formulation is known as the *velocity–pressure formulation*. The other formulation is based on the interpretation that the continuity equation (Eq. 8) is a constraint on *u* and *v*, and the constraint is satisfied in a least-squares (i.e., approximate) sense. This particular method of including the constraint in the formulation is known as the penalty function method, and the model is termed as the *penalty-finite element model*. It is informative to note that the velocity–pressure formulation is the same as the Lagrange multiplier formulation, wherein the constraint is included by means of the Lagrange multiplier. The Lagrange multiplier turns out to be the negative of the pressure.

Velocity–Pressure (Mixed) Model

The weak forms of Eqs. 6, 7, and 8 over an element Ω^e can be constructed following the standard procedure (Reddy 2015). The weight functions have the following physical interpretation:

$$w_1 \sim u, w_2 \sim v \text{ and } w_3 \sim -P \tag{10}$$

Assuming approximations of the form

$$u = \sum_{j=1}^{m} v_j \psi_j^e, \quad v = \sum_{j=1}^{m} v_j \psi_j^e, \quad P = \sum_{j=1}^{n} P_j \phi_j^e \tag{11}$$

where ψ_j^e and ϕ_j^e denote the Lagrange type interpolation functions ($n < m; n + 1 = m$) and substituting into the weak forms of Eqs. 6, 7, and 8, we obtain the following finite element model:

$$\begin{bmatrix} \mathbf{K}^{11} & \mathbf{K}^{12} & \mathbf{K}^{13} \\ \mathbf{K}^{21} & \mathbf{K}^{22} & \mathbf{K}^{23} \\ \mathbf{K}^{31} & \mathbf{K}^{32} & \mathbf{K}^{33} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{P} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}^1 \\ \mathbf{F}^2 \\ \mathbf{F}^3 \end{Bmatrix} \tag{12}$$

where $F_i^3 = 0$

$$K_{ij}^{11} = \int_{\Omega^e} \left[\mu \left(2 \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right) + \rho \psi_i \left(u \frac{\partial \psi_j}{\partial x} + v \frac{\partial \psi_j}{\partial y} \right) \right] \mathrm{d}x\mathrm{d}y$$
$$K_{ij}^{12} = \int_{\Omega^e} \mu \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial x} \mathrm{d}x\mathrm{d}y,$$
$$K_{ij}^{13} = -\int_{\Omega^e} \frac{\partial \psi_i}{\partial x} \psi_j \, \mathrm{d}x\mathrm{d}y, \quad K_{ij}^{33} = 0$$
$$K_{ij}^{22} = \int_{\Omega^e} \left[\mu \left(\frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + 2 \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right) + \rho \psi_i \left(u \frac{\partial \psi_j}{\partial x} + v \frac{\partial \psi_j}{\partial y} \right) \right] \mathrm{d}x\mathrm{d}y \tag{13}$$
$$K_{ij}^{23} = -\int_{\Omega^e} \frac{\partial \psi_i}{\partial y} \psi_j \mathrm{d}x\mathrm{d}y, \quad K_{ij}^{21} = K_{ji}^{21},$$
$$K_{ij}^{31} = K_{ji}^{13}, \quad K_{ij}^{32} = K_{ji}^{23}$$
$$F_i^1 = \int_{\Omega^e} f_x \psi_i \mathrm{d}x\mathrm{d}y + \oint_{\Gamma^e} t_x \psi_i \mathrm{d}s,$$
$$F_i^2 = \int_{\Omega^e} f_y \psi_i \mathrm{d}x\mathrm{d}y + \oint_{\Gamma^e} t_y \psi_i \mathrm{d}s$$

Here (t_x, t_y) denote the components of the stress vector $\mathbf{t}$ on the boundary

$$t_x = \sigma_{xx} n_x + \sigma_{xy} n_y, \quad t_y = \sigma_{xy} n_x + \sigma_{yy} n_y \tag{14}$$

and (n_x, n_y) are the components of the unit normal vector $\widehat{\mathbf{n}}$.

Penalty-Finite Element Model

Use of the penalty function method amounts to replacing the pressure with

$$P = -\gamma\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \tag{15}$$

where γ is known as the penalty parameter. For complete details, the reader may consult (Reddy 2015, 2019; Reddy and Gartling 2010). Note that Eq. 15 is used to post-compute P once the velocity field is available.

Assuming interpolation of the form

$$u = \sum_{j=1}^{n} u_j^e \psi_j^e, \quad v = \sum_{j=1}^{n} v_j^e \psi_j^e \tag{16}$$

where ψ_i^e are Lagrange interpolation functions. Substituting Eq. 16 into the weak forms of Eqs. 6 and 7, we obtain the finite element model

$$\begin{bmatrix} \mathbf{K}^{11} & \mathbf{K}^{12} \\ \mathbf{K}^{21} & \mathbf{K}^{22} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{v} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}^1 \\ \mathbf{F}^2 \end{Bmatrix} \tag{17}$$

where

$$\begin{aligned}
K_{ij}^{11} &= \int_{\Omega^e} \left[\mu \left(2\frac{\partial \psi_i^e}{\partial x}\frac{\partial \psi_j^e}{\partial x} + \frac{\partial \psi_i^e}{\partial y}\frac{\partial \psi_j^e}{\partial y} \right) \right. \\
&\left. + \rho \psi_i^e \left(u\frac{\partial \psi_j^e}{\partial x} + v\frac{\partial \psi_j^e}{\partial y} \right) \right] \mathrm{d}x\mathrm{d}y + \int_{\Omega^e} \gamma \frac{\partial \psi_i^e}{\partial x}\frac{\partial \psi_j^e}{\partial x}\, \mathrm{d}x\mathrm{d}y \\
K_{ij}^{12} &= \int_{\Omega^e} \mu \frac{\partial \psi_i}{\partial y}\frac{\partial \psi_j}{\partial x}\mathrm{d}x\mathrm{d}y + \int_{\Omega^e} \gamma \frac{\partial \psi_i^e}{\partial x}\frac{\partial \psi_j^e}{\partial y}\, \mathrm{d}x\mathrm{d}y \\
K_{ij}^{22} &= \int_{\Omega^e} \left[\mu \left(\frac{\partial \psi_i}{\partial x}\frac{\partial \psi_j}{\partial x} + 2\frac{\partial \psi_i}{\partial y}\frac{\partial \psi_j}{\partial y} \right) \right. \\
&\left. + \rho \psi_i \left(u\frac{\partial \psi_j}{\partial x} + v\frac{\partial \psi_j}{\partial y} \right) \right] \mathrm{d}x\mathrm{d}y + \int_{\Omega^e} \gamma \frac{\partial \psi_i^e}{\partial y}\frac{\partial \psi_j^e}{\partial y}\, \mathrm{d}x\mathrm{d}y \\
F_i^1 &= \int_{\Omega^e} f_x \psi_i^e \mathrm{d}x\mathrm{d}y + \oint_{\Gamma^e} t_x \psi_i^e \mathrm{d}s, \\
F_i^2 &= \int_{\Omega^e} f_y \psi_i^e \mathrm{d}x\mathrm{d}y + \oint_{\Gamma^e} t_y \psi_i^e \mathrm{d}s
\end{aligned} \tag{18}$$

The numerical evaluation of the coefficient matrices appearing in Eq. 17 requires special consideration. Eq. 17 is of the general form

$$(\mathbf{K}^{\mu} + \mathbf{K}^{\rho} + \mathbf{K}^{\gamma})\Delta = \mathbf{F} \tag{19}$$

where $\mathbf{K}^{\mu}$, $\mathbf{K}^{\rho}$, and $\mathbf{K}^{\gamma}$ denote the contributions from the viscous, inertia, and penalty terms, respectively. In theory, as we increase the value of the penalty parameter γ, the conservation of mass is satisfied more exactly. However, in practice, for some large value of γ, the contribution from the viscous and inertia terms would be negligibly small compared to the penalty terms in the computer. Thus, if $\mathbf{K}^3$ is a nonsingular (i.e., invertible) matrix, the solution of the final equations associated with Eq. 19 for a large value of γ is trivial

$$\lim \gamma \to 0(\mathbf{K}^{\mu} + \mathbf{K}^{\rho} + \mathbf{K}^{\gamma})\Delta = \mathbf{F} \to \mathbf{K}^{\gamma}\Delta = \frac{1}{\gamma}\mathbf{F} \tag{20}$$

which will yield $\Delta = 0$. While the trivial solution satisfies the continuity equation, it does not satisfy the momentum equations for nontrivial boundary data or body forces. In this case, the discrete problem (Eq. 17) is said to be over-constrained or "locked." If $\mathbf{K}^{\gamma}$ is singular, then the sum $\mathbf{K}^{\mu}$ + $\mathbf{K}^{\rho}$ + $\mathbf{K}^{\gamma}$ is nonsingular (because $\mathbf{K}^{\mu}$ is nonsingular after the imposition of proper boundary conditions), and a nontrivial solution to the problem may be obtained.

The numerical problem described above is eliminated by proper evaluation of the integrals in $\mathbf{K}^{\gamma}$. It is found that if the coefficients of $\mathbf{K}^{\gamma}$ (i.e., penalty terms) are evaluated using a numerical integration rule of one order less than that required to integrate them exactly, the finite element equations (Eq. 17) will give acceptable solutions for the velocity field. This technique of under-integrating the penalty terms is known in the literature as *reduced* (*order*) *integration*. For example, if a linear rectangular element is used to approximate the velocity field in a two-dimensional problem, the matrix coefficients $\mathbf{K}^{\mu}$ as well as $\mathbf{K}^{\rho}$ are evaluated using the 2×2 Gauss quadrature, and $\mathbf{K}^{\gamma}$ are evaluated using the one-point 1×1) Gauss quadrature. The one-point quadrature yields a singular $\mathbf{K}^{\gamma}$. Therefore, Eq. 17 cannot be inverted, whereas $\mathbf{K}^{\mu} + \mathbf{K}^{\rho} + \mathbf{K}^{\gamma}$ is nonsingular and can be inverted (after assembly and imposition of boundary conditions) to obtain a good finite element solution of the original problem. When a quadratic rectangular element is used, the 3×3 Gauss quadrature is used to evaluate $\mathbf{K}^{\mu}$ and $\mathbf{K}^{\rho}$, and the 2×2 Gauss quadrature is used to evaluate $\mathbf{K}^{\gamma}$.

The choice of the penalty parameter is largely dictated by the ratio of the magnitude of penalty terms to the viscous and convective terms (or compared to the Reynolds number, *Re*), the mesh, and the word length in the computer. The following range of γ is suggested in computations

$$\gamma = 10^4 R_e \text{ to } \gamma = 10^{12} R_e.$$

Summary

Numerical simulation of geomechanical processes requires a good understanding of computational fluid mechanics, heat

transfer, and solid mechanics and their couplings. The increase in computing power in both single processor and parallel environments has allowed realistic geomechanics problems of significant complexity and fidelity to be routinely solved and utilized in technological advances. Commercial software has made rapid progress in providing a broad spectrum of analysis capabilities to a variety of industries. Though software is increasingly robust, accurate simulations still require a knowledgeable user, with a background in both mechanics and numerical methods. This entry only provides an introduction to an individual who is interested in the use of the finite element method as a numerical simulation tool for the study and understanding of geomechanical phenomena. The Poisson equation and the Navier–Stokes equations visited here provide the necessary background for the study of diffusion processes and viscous flow problems. Interested readers may consult the references listed.

Cross-References

- ▶ Numerical Methods, Boundary Element
- ▶ Numerical Methods, Finite Difference
- ▶ Numerical Methods, Multigrid

Acknowledgments The author gratefully acknowledges the support provided by the Oscar S. Wyatt Endowed Chair.

Bibliography

Bathe KJ (1996) Finite element procedures. Prentice Hall, Englewood Cliffs

Belytschko T, Liu WK, Moran B (2000) Nonlinear finite elements for continua and structures. Wiley, Chichester

Bird P (1989) New finite element techniques for modeling deformation histories of continents with stratified temperature-dependent rheology. J Geophys Res 94(B4):3967–3990

Dyksterhuis S et al (2005) Finite-element modelling of contemporary and palaeo-intraplate stress using ABAQUS. Comput Geosci 31(3):297–307

Parsons T (2002) Post-1906 stress recovery of the San Andreas fault system calculated from three-dimensional finite element analysis. J Geophys Res (B8). https://doi.org/10.1029/2001JB001051

Reddy JN (2015) An introduction to nonlinear finite element analysis, 2nd ed. Oxford University Press, Oxford

Reddy JN (2019) An introduction to the finite element method, 4th edn. McGraw-Hill, New York

Reddy JN, Gartling DK (2010) The finite element method in heat transfer and fluid dynamics, 3rd edn. CRC Press, Boca Raton

Reddy JN, Stein RJ, Wickham JS (1982) Finite-element modeling of folding and faulting. Int J Numer Anal Methods Geomech 6:425–440

Wickham JS, Tapp GS, Reddy JN (1982) Finite-element modeling of fracture density in single layer folds. Int J Numer Anal Methods Geomech 6:441–459

Numerical Methods, Multigrid

Wim A. Mulder
Faculty of Civil Engineering and Geosciences, Department of Geoscience and Engineering, Delft University of Technology, Delft, The Netherlands
Shell Global Solutions International BV, Amsterdam, The Netherlands

Definition

A multigrid method is an algorithm for the iterative solution of partial differential equations using a sequence of discretizations on multiple scales.

Introduction

The numerical solution of a partial differential equation (PDE) requires its discretization and a method to solve the resulting large system of algebraic equations. For linear equations, the resulting system is often a sparse matrix, and a direct solution method suffices if the size of the problem is modest. In three space dimensions, the computational cost of a direct solver may be too high. An iterative method that improves the accuracy of an approximate solution step by step can be a good alternative. Multigrid is an example.

The multigrid method has optimal complexity: the amount of work required to solve a problem with N unknowns is $O(N)$, meaning that it scales with the problem size N. It achieves its efficiency by employing several discretization grids for the same problem.

History

Fedorenko (1964) introduced the multigrid method as an iterative scheme for solving Poisson's equation on a square and showed that the number of computations required to determine a solution with a prescribed accuracy is proportional to the number of unknowns, N. Therefore, the method has an optimal computational complexity. Brandt (1973) found that the actual computational cost for a sufficiently accurate result was about 10 work units, where a work unit is the cost of evaluating the discretized equations. He connected the method to local adaptive grid refinement and introduced nonlinear multigrid. Hackbush (1976, 1985) discovered the method independently and provided a mathematical foundation. Since then, the method was developed further to handle PDEs other than Poisson's, which is elliptic, and to go beyond PDEs.

Textbooks include those by Hackbush (1976), who includes convergence proofs; by Wesseling (1992), with chapters on computational fluid dynamics; by Briggs et al. (2000), an easy to read introduction; and by Trottenberg et al. (2001), among others.

Two-Grid Scheme

A discrete representation of a PDE provides an algebraic system of equations with relations between solution values at neighbouring grid points. It is fairly easy to make local corrections that reduce the solution error on a short range but much harder to correct the long-range or long-wavelength components of the solution that have a more global character. By projecting the solution onto a coarser grid, the long wavelengths become shorter and can effectively be solved for. Combining the corrections to the solution from coarser and finer grids yields an efficient solver.

A simple one-dimensional example can help to understand the fundamentals of the multigrid method. The PDE is $Lu = f$, with $u(x)$ the unknown solution as a function of position x on a finite interval $[x_{\min}, x_{\max}]$ on the real axis, $f(x)$ a source term or forcing function, and L a linear differential operator, for instance, minus the Laplacian, which in 1D is $-\frac{d^2}{dx^2}$. Dirichlet boundary conditions let the solution be zero at the endpoints of the interval. To obtain a discrete representation of the problem, an equidistant one-dimensional grid is defined with grid points $x_k = x_{\min} + kh$, index $k = 0, \ldots, N+1$, where $N + 1 = 2^M$ and M is a positive integer. The grid spacing is $h = (x_{\max} - x_{\min})/(N + 1)$. A standard second-order finite-difference scheme leads to

$$\frac{-u_{k-1} + 2u_k - u_{k+1}}{h^2} = f_k, \qquad k = 1, \ldots, N,$$

where u_k approximates $u(x_k)$ and $f_k = f(x_k)$. At the boundaries, $u_0 = u_{N+1} = 0$. The discrete equations represent the problem $L^h u^h = f^h$, where L^h is a $N \times N$ sparse matrix and u^h and f^h are vectors of length N. The residual is defined as $r^h = f^h - L^h u^h$ and should vanish once the numerical solution has been found.

A simple iterative method is Jacobi relaxation, in which the matrix L^h is replaced by its diagonal $D^h = 2/h^2$. One step of Jacobi relaxation amounts to

$$u^h := u^h + \omega \left(D^h\right)^{-1} r^h.$$

The symbol ":=" indicates that the solution values are replaced by the expression on the right-hand side. The factor ω controls the amount of damping. Convergence requires $0 < \omega \leq 1$. Fourier analysis (Hackbush 1985) shows that the convergence rate, the factor by which the norm of the difference between the current and the exact numerical solution is reduced, is $1 - O(h^2)$ for Jacobi relaxation. Convergence slows down on increasingly finer grids.

The Fourier analysis reveals that the slow convergence is caused by the long-wave components of the solution. Because long waves can be represented on coarser grids, it makes sense to have look at a two-grid scheme, outlined in Fig. 1 and its caption. The coarse grid consists of every other grid point of the fine grid: $x_K = x_0 + KH$, with $H = 2h$ and $K = 0, \ldots, \frac{1}{2}(N + 1)$. A restriction operator $\widetilde{I}_h^H$ maps the current fine-grid residual r^h to the coarser grid: $r^H = \widetilde{I}_h^H r^h$. The simplest restriction operator is injection: $r_K^H = r_{2K}^h$, $K = 1, \ldots, \frac{1}{2}(N - 1)$. Full weighting, which lets $r_K^H = \frac{1}{4} r_{2K-1}^h + \frac{1}{2} r_{2K}^h + \frac{1}{4} r_{2K+1}^h$, applies some smoothing to the residual and is more common. The exact solution of the coarse-grid problem yields the coarse-grid correction $v^H = (L^H)^{-1} r^H$, which should be interpolated back to the fine grid and added as a correction to the latest fine-grid solution. The interpolation or prolongation operator is denoted by I_H^h and lets $u^h := u^h + I_H^h v^H$.

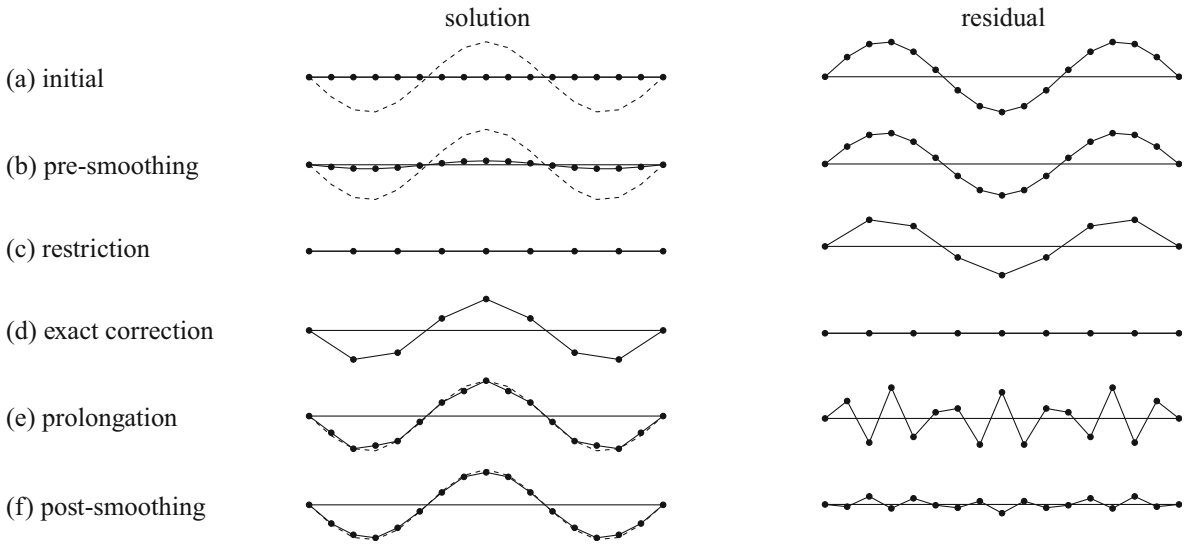
If linear interpolation is used for prolongation, this becomes

$$u_{2K}^h := u_{2K}^h + v_K^H, \qquad K = 1, \ldots, \frac{1}{2}(N - 1),$$

$$u_{2K-1}^h := u_{2K-1}^h + \frac{1}{2}\left(v_{K-1}^H + v_K^H\right), \qquad K = 1, \ldots, \frac{1}{2}(N + 1).$$

Here, it is assumed that $v_0^H = v_{(N+1)/2}^H = 0$. After prolongation, an additional relaxation step with damped Jacobi further removes oscillatory error components of the solution. Jacobi relaxation inside this two-grid scheme has a different purpose than when used as an iterative solver by itself. Instead of removing both the short- and long-wave components of the numerical solution error, it only has to deal with those components that cannot be represented on the coarser grid without aliasing. Therefore, a damped version with $\omega < 1$ can be more effective as it can be optimized to remove the short-wave or oscillatory components. For Fig. 1, $\omega = 2/3$ was used. The optimal choice requires a more detailed analysis (Hackbush 1985).

A relaxation scheme geared towards removing oscillatory components is called a smoother. Another popular choice is Gauss-Seidel relaxation. The operator is then approximated by a lower or upper triangular matrix, which is easy to invert. The implementation of the scheme is similar to Jacobi, but the residual is evaluated with the most recent solution available. The result will depend on the order in which the grid is traversed. With lexicographic Gauss-Seidel, one follows the natural index k in increasing order or in the opposite direction. Symmetric Gauss-Seidel performs both these smoothing steps in sequence. An alternative is red-black Gauss-Seidel, where first the points with an odd and then those with an even



Numerical Methods, Multigrid, Fig. 1 Steps in a two-grid cycle. (**a**) The initial solution on the fine grid is set to zero and the residual equals the forcing function. The dash line represents the exact numerical solution. (**b**) After one step of pre-smoothing with damped Jacobi relaxation, the solution error is still large, and the residual has hardly changed in this example. Its restriction to the coarser grid (**c**) is solved exactly, using a coarse-grid version of the discrete differential operator. The resulting correction to the fine-grid solution (**d**) is interpolated or prolongated back to the fine grid and added to the fine-grid solution (**e**). The error in the solution is now dominated by the short wavelengths and appears as an oscillatory function, which is reflected in the corresponding residual. A post-smoothing step (**f**) removes most of the solution error. Repeating the whole cycle will further reduce the error

index are updated, always using the latest solution values for the residual. In 2D, this would follow a checkerboard pattern.

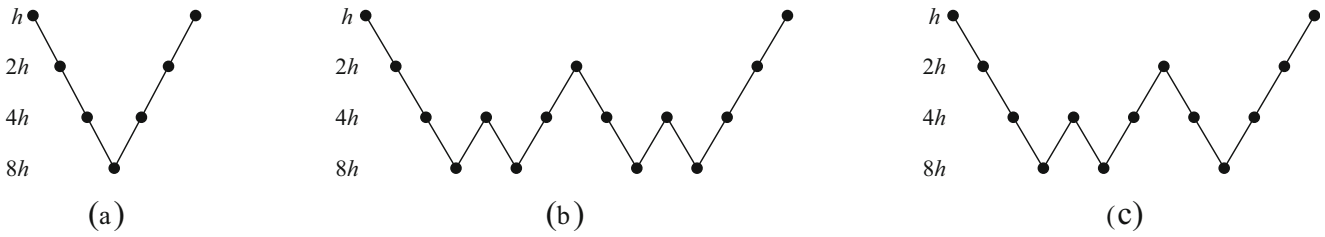
The grid transfer operator, restriction and prolongation, should obey $m_p + m_r > 2m$ (Hackbush 1985) where $2m$ is the order of the differential equation and $m_p - 1$ is the highest degree of the polynomial that is interpolated exactly. The scaled adjoint of the restriction operator is an interpolation operator for which m_r can be defined in the same way as m_p. In the example above, $2m = 2$, full-weighting has $m_r = 2$, and prolongation based on linear interpolation has $m_p = 2$. More advanced grid transfer operators are based on the differential operator L^h, leading to operator-weighted restriction and prolongation.

The coarse-grid operator L^H can be based on the same discretization as the fine-grid operator L^h. An alternative is the Galerkin coarse-grid approximation $\widetilde{I}_h^H L^h I_H^h$. In the current example with full weighting, these happen to be the same. Operator-weighted grid transfer operators can accelerate the convergence by using the coefficients of the differential operator in the construction of restriction and prolongation operators, which in turn will affect the Galerkin coarse-grid approximation of the differential operator.

Multigrid

Instead of using the exact numerical solution on the coarser of the two grids, a two-grid scheme can be applied to the coarse-grid problem. Extending this recursively to the coarsest grid with three points and one unknown, a multigrid solver is obtained. All these grids can be visited in different orders. The standard approach is to start at the finest grid, perform a number of pre-smoothing steps, then move to the coarser grid, again perform pre-smoothing, and continue this until the coarsest grid is reached. There, the exact numerical solution is computed, and the resulting coarse-grid correction is prolongated back to the finer level, followed by a number of post-smoothing steps. This is repeated up to the finest grid. Such a sequence is called a V-cycle. A single V-cycle may be insufficient to obtain a convergence rate on the coarser grid that is similar to the smoothing rate on the finer grid. Convergence on coarser grids can be improved by performing more than one cycle. In the W-cycle, the number of cycles doubles on each coarser grid. Figure 2 illustrates the order in which grids are visited for a V-, W-, and F-cycle. The last one is less expensive than a W-cycle, as the number of cycles increases by only one for increasingly coarser grids. Another option, adaptive cycling, decides on the sequence of grids by monitoring the decrease of the residuals.

With a proper choice of relaxation scheme or smoother grid transfer operators, the multigrid method can reach a grid-independent or h-independent convergence rate. This does not, however, mean that the amount of work required to reach a sufficiently accurate solution is proportional to the number of grid points N. The exact numerical solution differs from the true solution of the PDE because of the numerical error due to the discretization, typically with some power p of h. In the earlier example, the discretization is second-order



Numerical Methods, Multigrid, Fig. 2 (**a**) V-cycle, (**b**) W-cycle, and (**c**) F-cycle

accurate, so $p = 2$. Given this numerical error, convergence to machine precision is more than needed. An iteration error somewhat smaller than the discretization error should suffice. Therefore, more iterations should be carried out if a more accurate solution on a finer grid has to be computed. This leads to an overall complexity of $O(N \log N)$ rather than $O(N)$. Successive grid refinement, in which one first computes a solution on a coarse grid and uses that as an initial guess for the next finer grid and so on, enables removal of the factor log N, assuming that a fixed number of multigrid cycles are used at each level. The combination of successive grid refinement with a multigrid solver is called full multigrid (FMG).

Nonlinear Multigrid

One way to apply the multigrid method to nonlinear PDEs is the use of Newton's method. A multigrid solver is then applied to a linearization of the discretized nonlinear problem. Another approach is the use of the discretized nonlinear PDE inside the multigrid algorithm. This requires the full solution to be available on the coarser grids. In the example above, only corrections to the solution were represented on coarser grids. The full approximation scheme (FAS) is a reformulation of the multigrid method that includes the full solution (Brandt 1982). It requires a restriction of the full solution, $u^H = I_h^H u^h$, and of the coarse-grid forcing function, $f^H = L^H\left(I_h^H u^h\right) + \widetilde{I}_h^H r^h$. The prolongation needs to be changed to $u^h := u^h + I_H^h\left(u^H - I_h^H u^h\right)$, where u^H is the sufficiently converged solution on the coarser grid and $I_h^H u^h$ the restriction of the latest fine-grid solution. The restriction operators for the residuals, $\widetilde{I}_h^H$, and for the solution, I_h^H, do not have to be the same. Note that the coarse-grid problem, $L^H u^H = f^H$, can be interpreted in a different way: $L^H u^H = f^H = \widetilde{I}_h^H f^h + \tau_h^H$. The fine-to-coarse defect correction, $\tau_h^H = L^H\left(I_h^H u^h\right) - \widetilde{I}_h^H\left(L^h u^h\right)$, ensures that the coarse-grid equations maintain the accuracy of the fine-grid equations.

Nonlinear equations of the form $L^h(u^h) = f^h$ fit into this scheme. For $L^H(u^H)$, the same discretization scheme as on the fine grid can be adopted. Smoothing operators can be based on a local linearization of the residuals.

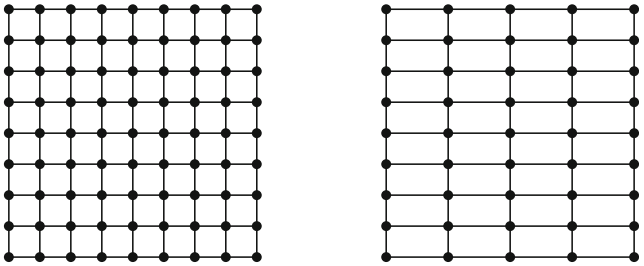
Generalizations

Multigrid is highly efficient for the solution of Poisson's equation on the square, discretized by the finite-difference method on a Cartesian grid. For other PDEs, describing convection, diffusion, waves, and flow, the method may be less easy to apply. The same is true for unstructured grids, for instance, based on triangles or tetrahedra. With suitable modifications and at the expense of additional code complexity, the multigrid method can still be the optimal approach. With unstructured grids, the coarser and finer grid are generally not nested, leading to grid transfer operations that are less easy to code than on a Cartesian grid.

Anisotropy in the PDE can degrade performance. Consider the equation:

$$-\frac{\partial^2 u}{\partial x^2} - a\frac{\partial^2 u}{\partial y^2} = f(x, y).$$

For $0 < a \ll 1$, smoothing operators may have difficulty with the y-direction where there is only weak coupling between neigbouring unknowns. Effective anisotropy in the discretized equation can also happen with grids that have widely different spacings in the various coordinate directions. A more powerful relaxation scheme may repair the poor performance of a simple smoother. For 2D problems, line relaxation will smooth in one direction and solve in the other. An example is line Jacobi relaxation, which approximates the matrix L^h by dropping the off-diagonals in one coordinate direction, while keeping them in the other. The dropped off-diagonals can be subtracted from the main diagonal with a suitable weight factor to obtain better smoothing properties. For 3D problems, plane relaxation is required, which may be too expensive. An alternative is semi-coarsening, where the coarse grid has the same number of points in the direction of weak coupling and is coarsened only in the direction in which the smoothing effectively removes the shorter wavelengths. Figure 3 shows an example of a grid before and after semi-coarsening in the horizontal direction. In the more general case where semi-coarsening is required in all coordinates, it can be applied in alternating directions.



Numerical Methods, Multigrid, Fig. 3 Semi-coarsening of grid (**a**) in the horizontal direction provides grid (**b**)

A more costly but powerful approach is simultaneous semi-coarsening in all coordinates (Mulder 1989), which still retains optimal complexity.

Beyond Partial Differential Equations

The application of the multigrid method is not restricted to the solution of PDEs. There are generalizations to eigenvalue problems, integral equations, optimization and optimal control, statistical physics, image processing, image segmentation, and edge detection. In all cases, the simultaneous use of different scales overcomes the problem of slowly converging solution components.

Algebraic multigrid (AMG), or, better, the algebraic multilevel method, refers to a class of solvers that construct sets of coarse-level equations without referring to grids or geometrical properties (Trottenberg et al. 2001; Shapira 2008). Coarsening is applied to subsets of unknowns for which smoothing is effective. If the problem is specified by a large matrix, these subsets can be determined by examining the coefficients of the matrix. Strongly coupled unknowns, say u_i and u_j, are related by a matrix coefficient a_{ij} that is relatively large. In that case, smoothing is effective, and one of the unknowns can be removed from the equations on the coarser level. The coarser level will then be mainly populated by weakly coupled variables. Operator-weighted grid transfer operators are natural in this context, as well as the Galerkin approach for the construction of the coarse-level operator. Algebraic multilevel methods are a popular choice for finite-element discretizations on unstructured grids, as they can be used as a black-box solver and do not require the tedious coding of grid transfer operations between non-nested elements.

The multigrid method bears some resemblance to other techniques such as wavelets, hierarchical-basis finite elements, multi-scale techniques, cyclic reduction, fast multipole methods, and other divide-and-conquer methods.

Geophysical Applications

Multigrid has been applied successfully to a wide range of partial differential equations, describing static and dynamic problems, diffusion, convection, and flow problems. Examples in geophysics include elliptic problems in potential theory, such as gravity (Kusche 2002), magnetostatics (De Gersem and Hameyer 2001), electrostatics (Ascher and Haber 2003), and Darcy's law in porous media flow (Mulder and Gmelig Meyling 1993). Two-dimensional phase unwrapping for satellite interferometric synthetic aperture radar (Pritt 1996) can also be stated as an elliptic problem. Controlled-source electromagnetics (Werthmüller et al. 2019), magnetotellurics (Li et al. 2015), mantle convection, and geodynamics (Gerya 2019) are examples of parabolic problems.

The wave equation for seismic applications is a hyperbolic problem. Iterative solution of its frequency-domain formulation, the Helmholtz equation, was a notoriously difficult numerical problem but can nowadays be accomplished by using a damped version as preconditioner in a conjugate-gradient-type iterative scheme. Multigrid efficiently deals with the approximate inversion of the preconditioner (Riyanti et al. 2006). In the generalization of the method to the elastic wave equation, simple Point-Jacobi smoothing turns out to be insufficient for the secondary or shear waves, and a more powerful smoother is required (Rizzuti and Mulder 2016).

Summary

The multigrid method provides an optimal iterative solution method for a wide class of problems governed by partial differential equations. Its application to a finite-difference discretization of Poisson's equation on a square is the easiest. For other problems, a bit more effort may be required.

Cross-References

▶ Numerical Methods, Finite Difference
▶ Numerical Methods, Finite Element

Bibliography

Ascher UM, Haber E (2003) A multigrid method for distributed parameter estimation problems. Electron Trans Numer Anal 15:1–17

Brandt A (1973) Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems. In: Cabannes H, Temam R (eds) Proceedings of the third international conference on numerical methods in fluid mechanics. Springer Berlin Heidelberg, Berlin/Heidelberg, pp 82–89. https://doi.org/10.1007/BFb0118663

Brandt A (1982) Guide to multigrid development. In: Hackbusch W, Trottenberg U (eds) Multigrid methods. Springer Berlin Heidelberg, Berlin/Heidelberg, pp 220–312. https://doi.org/10.1007/BFb0069930

Briggs WL, Henson VE, McCormick SF (2000) A multigrid tutorial, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia. https://doi.org/10.1137/1.9780898719505

De Gersem H, Hameyer K (2001) Full multigrid for magnetostatics using unstructured and non-nested meshes. IEEE Trans Magn 37(5):3460–3464. https://doi.org/10.1109/20.952637

Fedorenko RP (1964) The speed of convergence of one iterative process. USSR Comput Math Math Phys 4(3):227–235. https://doi.org/10.1016/0041-5553(64)90253-8

Gerya TV (2019) Introduction to numerical geodynamic modelling, 2nd edn. Cambridge University Press, Cambridge. https://doi.org/10.1017/9781316534243

Hackbush W (1976) Ein iteratives Verfahren zur schnellen Auflösung elliptischer Randwertprobleme. Report 76-12. Mathematisches Institut, Universität zu Köln, Köln

Hackbush W (1985) Multi-grid methods and applications. Springer series in computational mathematics, vol 4. Springer, Berlin/Heidelberg. https://doi.org/10.1007/978-3-662-02427-0

Kusche J (2002) On fast multigrid iteration techniques for the solution of normal equations in satellite gravity recovery. J Geodyn 33(1):173–186. https://doi.org/10.1016/S0264-3707(01)00062-X

Li G, Hao T, Zhang L (2015) Performance of preconditioned iterative and multigrid solvers in solving the three-dimensional magneto-telluric modeling problem using the staggered finite-difference method: a comparative study. J Geophys Eng 13(1):1–10, 12. https://doi.org/10.1088/1742-2132/13/1/1

Mulder WA (1989) A new multigrid approach to convection problems. J Comput Phys 83(2):303–323. https://doi.org/10.1016/0021-9991(89)90121-6

Mulder WA, Gmelig Meyling RHJ (1993) Numerical simulation of two-phase flow using locally refined grids in three space dimensions. SPE Adv Technol Ser 1:36. https://doi.org/10.2118/21230-PA

Pritt MD (1996) Phase unwrapping by means of multigrid techniques for interferometric SAR. IEEE Trans Geosci Remote Sens 34(3):728–738. https://doi.org/10.1109/36.499752

Riyanti CD, Erlangga YA, Plessix R-É, Mulder WA, Vuik C, Oosterlee C (2006) A new iterative solver for the time-harmonic wave equation. Geophysics 71(5):E57–E63. https://doi.org/10.1190/1.2231109

Rizzuti G, Mulder WA (2016) Multigrid-based 'shifted-Laplacian' preconditioning for the time-harmonic elastic wave equation. J Comput Phys 317:47–65. https://doi.org/10.1016/j.jcp.2016.04.049

Shapira Y (2008) Matrix-based multigrid: theory and applications. Numerical methods and algorithms, vol 2, 2nd edn. Springer, Boston. https://doi.org/10.1007/978-0-387-49765-5

Trottenberg U, Oosterlee CW, Schüller A (2001) Multigrid. Academic, London. ISBN 9780127010700

Werthmüller D, Mulder WA, Slob EC (2019) emg3d: a multigrid solver for 3D electromagnetic diffusion. J Open Source Softw 4(37):1463. https://doi.org/10.21105/joss.01463

Wesseling P (1992) An introduction to multigrid methods. Wiley, Chichester