# Finite element method

The **finite element method** (**FEM**) is the most widely used method for solving problems of engineering and mathematical models. Typical problem areas of interest include the traditional fields of structural analysis, heat transfer, fluid flow, mass transport, and electromagnetic potential. The FEM is a particular numerical method for solving partial differential equations in two or three space variables (i.e., some boundary value problems). To solve a problem, the FEM subdivides a large system into smaller, simpler parts that are called **finite elements**. This is achieved by a particular space discretisation in the space dimensions, which is implemented by the construction of a mesh of the object: the numerical domain for the solution, which has a finite number of points. The finite element method formulation of a boundary value problem finally results in a system of algebraic equations. The method approximates the unknown function over the domain.[1] The simple equations that model these finite elements are then assembled into a larger system of equations that models the entire problem. The FEM then uses variational methods from the calculus of variations to approximate a solution by minimizing an associated error function.

Studying or analyzing a phenomenon with FEM is often referred to as **finite element analysis (FEA)**.

## Contents

## Basic concepts

The subdivision of a whole domain into simpler parts has several advantages:[2]

- Accurate representation of complex geometry
- Inclusion of dissimilar material properties
- Easy representation of the total solution
- Capture of local effects.

A typical work out of the method involves (1) dividing the domain of the problem into a collection of subdomains, with each subdomain represented by a set of element equations to the original problem, followed by (2) systematically recombining all sets of element equations into a global system of equations for the final calculation. The global system of equations has known solution techniques, and can be calculated from the initial values of the original problem to obtain a numerical answer.

In the first step above, the element equations are simple equations that locally approximate the original complex equations to be studied, where the original equations are often partial differential equations (PDE). To explain the approximation in this process, FEM is commonly introduced as a special case of Galerkin method. The process, in mathematical language, is to construct an integral of the inner product of the residual and the weight functions and set the integral to zero. In simple terms, it is a procedure that minimizes the error of approximation by fitting trial functions into the PDE. The residual is the error caused by the trial functions, and the weight functions are polynomial approximation functions that project the residual. The process eliminates all the spatial derivatives from the PDE, thus approximating the PDE locally with

- a set of algebraic equations for steady state problems,

- a set of ordinary differential equations for transient problems.

These equation sets are the element equations. They are linear if the underlying PDE is linear, and vice versa. Algebraic equation sets that arise in the steady state problems are solved using numerical linear algebra methods, while ordinary differential equation sets that arise in the transient problems are solved by numerical integration using standard techniques such as Euler's method or the Runge-Kutta method.

In step (2) above, a global system of equations is generated from the element equations through a transformation of coordinates from the subdomains' local nodes to the domain's global nodes. This spatial transformation includes appropriate orientation adjustments as applied in relation to the reference coordinate system. The process is often carried out by FEM software using coordinate data generated from the subdomains.

FEM is best understood from its practical application, known as **finite element analysis (FEA)**. FEA as applied in engineering is a computational tool for performing engineering analysis. It includes the use of mesh generation techniques for dividing a complex problem into small elements, as well as the use of software program coded with FEM algorithm. In applying FEA, the complex problem is usually a physical system with the underlying physics such as the Euler-Bernoulli beam equation, the heat equation, or the Navier-Stokes equations expressed in either PDE or integral equations, while the divided small elements of the complex problem represent different areas in the physical system.

FEA is a good choice for analyzing problems over complicated domains (like cars and oil pipelines), when the domain changes (as during a solid state reaction with a moving boundary), when the desired precision varies over the entire domain, or when the solution lacks smoothness. FEA simulations provide a valuable resource as they remove multiple instances of creation and testing of hard prototypes for various high fidelity situations.[3] For instance, in a frontal crash simulation it is possible to increase prediction accuracy in "important" areas like the front of the car and reduce it in its rear (thus reducing cost of the simulation). Another example would be in numerical weather prediction, where it is more important to have accurate predictions over developing highly nonlinear phenomena (such as tropical cyclones in the atmosphere, or eddies in the ocean) rather than relatively calm areas.

# History

While it is difficult to quote a date of the invention of the finite element method, the method originated from the need to solve complex elasticity and structural analysis problems in civil and aeronautical engineering. Its development can be traced back to the work by A. Hrennikoff[4] and R. Courant[5] in the early 1940s. Another pioneer was Ioannis Argyris. In the USSR, the introduction of the practical application of the method is usually connected with name of Leonard Oganesyan.[6] In China, in the later 1950s and early 1960s, based on the computations of dam constructions, K. Feng proposed a systematic numerical method for solving partial differential equations. The method was called the *finite difference method based on variation principle*, which was another independent invention of the finite element method. Although the approaches used by these pioneers are different, they share one essential characteristic: mesh discretization of a continuous domain into a set of discrete sub-domains, usually called elements.

Hrennikoff's work discretizes the domain by using a lattice analogy, while Courant's approach divides the domain into finite triangular subregions to solve second order elliptic partial differential equations (PDEs) that arise from the problem of torsion of a cylinder. Courant's contribution was evolutionary, drawing on a

large body of earlier results for PDEs developed by Rayleigh, Ritz, and Galerkin.
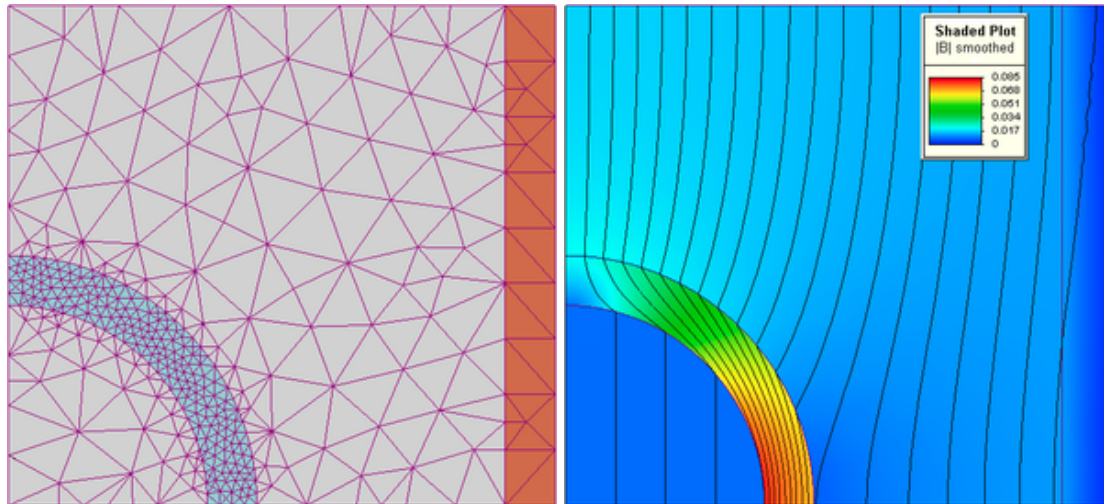
FEM mesh created by an analyst prior to finding a solution to a magnetic problem using FEM software. Colours indicate that the analyst has set material properties for each zone, in this case a conducting wire coil in orange; a ferromagnetic component (perhaps iron) in light blue; and air in grey. Although the geometry may seem simple, it would be very challenging to calculate the magnetic field for this setup without FEM software, using equations alone.

FEM solution to the problem at left, involving a cylindrically shaped magnetic shield. The ferromagnetic cylindrical part is shielding the area inside the cylinder by diverting the magnetic field created by the coil (rectangular area on the right). The color represents the amplitude of the magnetic flux density, as indicated by the scale in the inset legend, red being high amplitude. The area inside the cylinder is low amplitude (dark blue, with widely spaced lines of magnetic flux), which suggests that the shield is performing as it was designed to.

The finite element method obtained its real impetus in the 1960s and 1970s by the developments of J. H. Argyris with co-workers at the University of Stuttgart, R. W. Clough with co-workers at UC Berkeley, O. C. Zienkiewicz with co-workers Ernest Hinton, Bruce Irons[7] and others at Swansea University, Philippe G. Ciarlet at the University of Paris 6 and Richard Gallagher with co-workers at Cornell University. Further impetus was provided in these years by available open source finite element software programs. NASA sponsored the original version of NASTRAN, and UC Berkeley made the finite element program SAP IV[8] widely available. In Norway the ship classification society Det Norske Veritas (now DNV GL) developed Sesam in 1969 for use in analysis of ships.[9] A rigorous mathematical basis to the finite element method was provided in 1973 with the publication by Strang and Fix.[10] The method has since been generalized for the numerical modeling of physical systems in a wide variety of engineering disciplines, e.g., electromagnetism, heat transfer, and fluid dynamics.[11][12]

# Technical discussion

## The structure of finite element methods

A finite element method is characterized by a variational formulation, a discretization strategy, one or more

solution algorithms and post-processing procedures.

Examples of variational formulation are the Galerkin method, the discontinuous Galerkin method, mixed methods, etc.

A discretization strategy is understood to mean a clearly defined set of procedures that cover (a) the creation of finite element meshes, (b) the definition of basis function on reference elements (also called shape functions) and (c) the mapping of reference elements onto the elements of the mesh. Examples of discretization strategies are the h-version, p-version, hp-version, x-FEM, isogeometric analysis, etc. Each discretization strategy has certain advantages and disadvantages. A reasonable criterion in selecting a discretization strategy is to realize nearly optimal performance for the broadest set of mathematical models in a particular model class.

There are various numerical solution algorithms that can be classified into two broad categories; direct and iterative solvers. These algorithms are designed to exploit the sparsity of matrices that depend on the choices of variational formulation and discretization strategy.

Postprocessing procedures are designed for the extraction of the data of interest from a finite element solution. In order to meet the requirements of solution verification, postprocessors need to provide for *a posteriori* error estimation in terms of the quantities of interest. When the errors of approximation are larger than what is considered acceptable then the discretization has to be changed either by an automated adaptive process or by action of the analyst. There are some very efficient postprocessors that provide for the realization of superconvergence.

## Illustrative problems P1 and P2

We will demonstrate the finite element method using two sample problems from which the general method can be extrapolated. It is assumed that the reader is familiar with calculus and linear algebra.

P1 is a **one-dimensional** problem

$$\text{P1} \ : \begin{cases} u''(x) = f(x) \text{ in } (0,1), \\ u(0) = u(1) = 0, \end{cases}$$

where $f$ is given, $u$ is an unknown function of $x$, and $u''$ is the second derivative of $u$ with respect to $x$.

P2 is a **two-dimensional** problem (Dirichlet problem)

$$\text{P2} \ : \begin{cases} u_{xx}(x,y) + u_{yy}(x,y) = f(x,y) & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases}$$

where $\Omega$ is a connected open region in the $(x,y)$ plane whose boundary $\partial\Omega$ is nice (e.g., a smooth manifold or a polygon), and $u_{xx}$ and $u_{yy}$ denote the second derivatives with respect to $x$ and $y$, respectively.

The problem P1 can be solved directly by computing antiderivatives. However, this method of solving the boundary value problem (BVP) works only when there is one spatial dimension and does not generalize to higher-dimensional problems or to problems like $u + u'' = f$. For this reason, we will develop the finite

element method for P1 and outline its generalization to P2.

Our explanation will proceed in two steps, which mirror two essential steps one must take to solve a boundary value problem (BVP) using the FEM.

- In the first step, one rephrases the original BVP in its weak form. Little to no computation is usually required for this step. The transformation is done by hand on paper.
- The second step is the discretization, where the weak form is discretized in a finite-dimensional space.

After this second step, we have concrete formulae for a large but finite-dimensional linear problem whose solution will approximately solve the original BVP. This finite-dimensional problem is then implemented on a computer.

## Weak formulation

The first step is to convert P1 and P2 into their equivalent weak formulations.

### The weak form of P1

If $u$ solves P1, then for any smooth function $v$ that satisfies the displacement boundary conditions, i.e. $v = 0$ at $x = 0$ and $x = 1$, we have

$$(1) \quad \int_0^1 f(x)v(x)\, dx = \int_0^1 u''(x)v(x)\, dx.$$

Conversely, if $u$ with $u(0) = u(1) = 0$ satisfies (1) for every smooth function $v(x)$ then one may show that this $u$ will solve P1. The proof is easier for twice continuously differentiable $u$ (mean value theorem), but may be proved in a distributional sense as well.

We define a new operator or map $\phi(u, v)$ by using integration by parts on the right-hand-side of (1):

$$
\begin{aligned}
\int_0^1 f(x)v(x)\, dx &= \int_0^1 u''(x)v(x)\, dx \\
(2) \qquad &= u'(x)v(x)\big|_0^1 - \int_0^1 u'(x)v'(x)\, dx \\
&= -\int_0^1 u'(x)v'(x)\, dx \equiv -\phi(u, v),
\end{aligned}
$$

where we have used the assumption that $v(0) = v(1) = 0$.

### The weak form of P2

If we integrate by parts using a form of Green's identities, we see that if $u$ solves P2, then we may define $\phi(u, v)$ for any $v$ by

$$\int_\Omega fv\,ds = -\int_\Omega \nabla u \cdot \nabla v\,ds \equiv -\phi(u, v),$$

where $\nabla$ denotes the gradient and $\cdot$ denotes the dot product in the two-dimensional plane. Once more $\phi$ can be turned into an inner product on a suitable space $H_0^1(\Omega)$ of once differentiable functions of $\Omega$ that are zero on $\partial\Omega$. We have also assumed that $v \in H_0^1(\Omega)$ (see Sobolev spaces). Existence and uniqueness of the solution can also be shown.

### A proof outline of existence and uniqueness of the solution

We can loosely think of $H_0^1(0, 1)$ to be the absolutely continuous functions of $(0, 1)$ that are $0$ at $x = 0$ and $x = 1$ (see Sobolev spaces). Such functions are (weakly) once differentiable and it turns out that the symmetric bilinear map $\phi$ then defines an inner product which turns $H_0^1(0, 1)$ into a Hilbert space (a detailed proof is nontrivial). On the other hand, the left-hand-side $\int_0^1 f(x)v(x)dx$ is also an inner product, this time on the Lp space $L^2(0, 1)$. An application of the Riesz representation theorem for Hilbert spaces shows that there is a unique $u$ solving (2) and therefore P1. This solution is a-priori only a member of $H_0^1(0, 1)$, but using elliptic regularity, will be smooth if $f$ is.

# Discretization

P1 and P2 are ready to be discretized which leads to a common sub-problem (3). The basic idea is to replace the infinite-dimensional linear problem:

Find $u \in H_0^1$ such that
$$\forall v \in H_0^1, \ -\phi(u, v) = \int fv$$

with a finite-dimensional version:

(3) Find $u \in V$ such that
$$\forall v \in V, \ -\phi(u, v) = \int fv$$



A function in $H_0^1$, with zero values at the endpoints (blue), and a piecewise linear approximation (red)

where $V$ is a finite-dimensional subspace of $H_0^1$. There are many possible choices for $V$ (one possibility leads to the spectral method). However, for the finite element method we take $V$ to be a space of piecewise polynomial functions.

### For problem P1

We take the interval $(0, 1)$, choose $n$ values of $x$ with $0 = x_0 < x_1 < \cdots < x_n < x_{n+1} = 1$ and we define $V$ by:

$$V = \{v : [0, 1] \to \mathbb{R} \ : v \text{ is continuous}, v|_{[x_k, x_{k+1}]} \text{ is linear for } k = 0, \ldots, n, \text{ and } v(0) = v(1) = 0\}$$
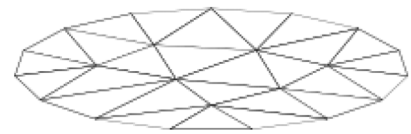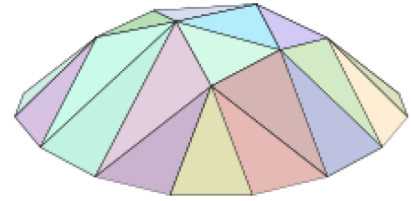
where we define $x_0 = 0$ and $x_{n+1} = 1$. Observe that functions in $V$ are not differentiable according to the elementary definition of calculus. Indeed, if $v \in V$ then the derivative is typically not defined at any $x = x_k$, $k = 1, \ldots, n$. However, the derivative exists at every other value of $x$ and one can use this derivative for the purpose of integration by parts.

## For problem P2

We need $V$ to be a set of functions of $\Omega$. In the figure on the right, we have illustrated a triangulation of a 15 sided polygonal region $\Omega$ in the plane (below), and a piecewise linear function (above, in color) of this polygon which is linear on each triangle of the triangulation; the space $V$ would consist of functions that are linear on each triangle of the chosen triangulation.

One hopes that as the underlying triangular mesh becomes finer and finer, the solution of the discrete problem (3) will in some sense converge to the solution of the original boundary value problem P2. To measure this mesh fineness, the triangulation is indexed by a real valued parameter $h > 0$ which one takes to be very small. This parameter will be related to the size of the largest or average triangle



A piecewise linear function in two dimensions

in the triangulation. As we refine the triangulation, the space of piecewise linear functions $V$ must also change with $h$. For this reason, one often reads $V_h$ instead of $V$ in the literature. Since we do not perform such an analysis, we will not use this notation.

## Choosing a basis

To complete the discretization, we must select a basis of $V$. In the one-dimensional case, for each control point $x_k$ we will choose the piecewise linear function $v_k$ in $V$ whose value is $1$ at $x_k$ and zero at every $x_j$, $j \neq k$, i.e.,

$$v_k(x) = \begin{cases} \frac{x - x_{k-1}}{x_k - x_{k-1}} & \text{if } x \in [x_{k-1}, x_k], \\ \frac{x_{k+1} - x}{x_{k+1} - x_k} & \text{if } x \in [x_k, x_{k+1}], \\ 0 & \text{otherwise}, \end{cases}$$

for $k = 1, \ldots, n$; this basis is a shifted and scaled tent function. For the two-dimensional case, we choose again one basis function $v_k$ per vertex $x_k$ of the triangulation of the planar region $\Omega$. The function $v_k$ is the unique function of $V$ whose value is $1$ at $x_k$ and zero at every $x_j$, $j \neq k$.

Depending on the author, the word "element" in "finite element method" refers either to the triangles in the domain, the piecewise linear basis function, or both. So for instance, an author interested in curved domains might replace the triangles with curved primitives, and so might describe the elements as being curvilinear. On the other hand, some authors replace "piecewise linear" by "piecewise quadratic" or even "piecewise polynomial". The author might then say "higher order element" instead of "higher degree polynomial". Finite element method is not restricted to triangles (or tetrahedra in 3-d, or higher order simplexes in multidimensional spaces), but can be defined on quadrilateral subdomains (hexahedra, prisms, or pyramids

in 3-d, and so on). Higher order shapes (curvilinear elements) can be defined with polynomial and even non-polynomial shapes (e.g. ellipse or circle).

Examples of methods that use higher degree piecewise polynomial basis functions are the hp-FEM and spectral FEM.

More advanced implementations (adaptive finite element methods) utilize a method to assess the quality of the results (based on error estimation theory) and modify the mesh during the solution aiming to achieve approximate solution within some bounds from the exact solution of the continuum problem. Mesh adaptivity may utilize various techniques, the most popular are:

- moving nodes (r-adaptivity)
- refining (and unrefining) elements (h-adaptivity)
- changing order of base functions (p-adaptivity)
- combinations of the above (hp-adaptivity).

## Small support of the basis

The primary advantage of this choice of basis is that the inner products

$$\langle v_j, v_k \rangle = \int_0^1 v_j v_k \, dx$$

and

$$\phi(v_j, v_k) = \int_0^1 v_j' v_k' \, dx$$

will be zero for almost all $j, k$. (The matrix containing $\langle v_j, v_k \rangle$ in the $(j, k)$ location is known as the Gramian matrix.) In the one dimensional case, the support of $v_k$ is the interval $[x_{k-1}, x_{k+1}]$. Hence, the integrands of $\langle v_j, v_k \rangle$ and $\phi(v_j, v_k)$ are identically zero whenever $|j - k| > 1$.
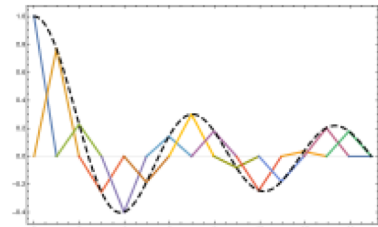
Similarly, in the planar case, if $x_j$ and $x_k$ do not share an edge of the triangulation, then the integrals
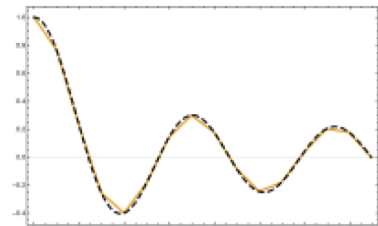
$$\int_\Omega v_j v_k \, ds$$

and

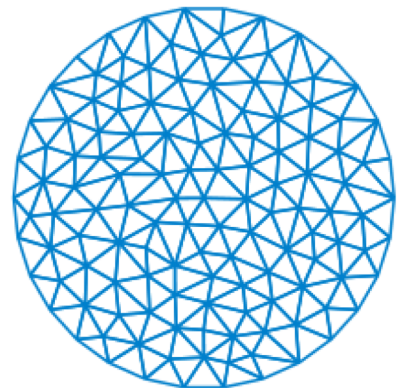$$\int_\Omega \nabla v_j \cdot \nabla v_k \, ds$$



**Interpolation of a Bessel function**

16 scaled and shifted triangular basis functions (colors) used to reconstruct a zeroeth order Bessel function $J_0$ (black).



The linear combination of basis functions (yellow) reproduces $J_0$ (blue) to any desired accuracy.



Solving the two-dimensional problem $u_{xx} + u_{yy} = -4$ in the disk centered at the origin and radius 1, with zero boundary conditions.
(a) The triangulation.

are both zero.

## Matrix form of the problem

If we write $u(x) = \sum_{k=1}^{n} u_k v_k(x)$ and $f(x) = \sum_{k=1}^{n} f_k v_k(x)$ then problem (3), taking $v(x) = v_j(x)$ for $j = 1, \ldots, n$, becomes

$$-\sum_{k=1}^{n} u_k \phi(v_k, v_j) = \sum_{k=1}^{n} f_k \int v_k v_j dx \text{ for } j = 1, \ldots, n. \quad (4)$$

If we denote by $\mathbf{u}$ and $\mathbf{f}$ the column vectors $(u_1, \ldots, u_n)^t$ and $(f_1, \ldots, f_n)^t$, and if we let

$$L = (L_{ij})$$

and

$$M = (M_{ij})$$

be matrices whose entries are

$$L_{ij} = \phi(v_i, v_j)$$

and

$$M_{ij} = \int v_i v_j dx$$

(b) The sparse matrix $L$ of the discretized linear system

(c) The computed solution, $u(x, y) = 1 - x^2 - y^2$.

then we may rephrase (4) as

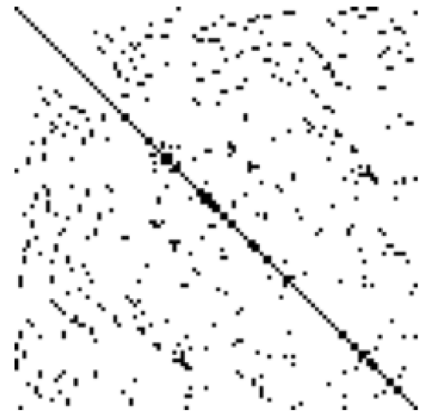$$-L\mathbf{u} = M\mathbf{f}. \quad (5)$$

It is not necessary to assume $f(x) = \sum_{k=1}^{n} f_k v_k(x)$. For a general function $f(x)$, problem (3) with $v(x) = v_j(x)$ for $j = 1, \ldots, n$ becomes actually simpler, since no matrix $M$ is used,
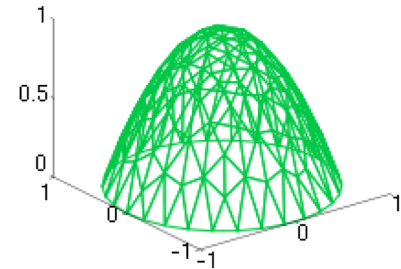
$$-L\mathbf{u} = \mathbf{b}, \quad (6)$$

where $\mathbf{b} = (b_1, \ldots, b_n)^t$ and $b_j = \int f v_j dx$ for $j = 1, \ldots, n$.

As we have discussed before, most of the entries of $L$ and $M$ are zero because the basis functions $v_k$ have small support. So we now have to solve a linear system in the unknown $\mathbf{u}$ where most of the entries of the matrix $L$, which we need to invert, are zero.

Such matrices are known as sparse matrices, and there are efficient solvers for such problems (much more efficient than actually inverting the matrix.) In addition, $L$ is symmetric and positive definite, so a technique

such as the conjugate gradient method is favored. For problems that are not too large, sparse LU decompositions and Cholesky decompositions still work well. For instance, MATLAB's backslash operator (which uses sparse LU, sparse Cholesky, and other factorization methods) can be sufficient for meshes with a hundred thousand vertices.

The matrix $L$ is usually referred to as the stiffness matrix, while the matrix $M$ is dubbed the mass matrix.

## General form of the finite element method

In general, the finite element method is characterized by the following process.

- One chooses a grid for $\Omega$. In the preceding treatment, the grid consisted of triangles, but one can also use squares or curvilinear polygons.
- Then, one chooses basis functions. In our discussion, we used piecewise linear basis functions, but it is also common to use piecewise polynomial basis functions.

A separate consideration is the smoothness of the basis functions. For second order elliptic boundary value problems, piecewise polynomial basis function that are merely continuous suffice (i.e., the derivatives are discontinuous.) For higher order partial differential equations, one must use smoother basis functions. For instance, for a fourth order problem such as $u_{xxxx} + u_{yyyy} = f$, one may use piecewise quadratic basis functions that are $C^1$.

Another consideration is the relation of the finite-dimensional space $V$ to its infinite-dimensional counterpart, in the examples above $H_0^1$. A conforming element method is one in which the space $V$ is a subspace of the element space for the continuous problem. The example above is such a method. If this condition is not satisfied, we obtain a nonconforming element method, an example of which is the space of piecewise linear functions over the mesh which are continuous at each edge midpoint. Since these functions are in general discontinuous along the edges, this finite-dimensional space is not a subspace of the original $H_0^1$.

Typically, one has an algorithm for taking a given mesh and subdividing it. If the main method for increasing precision is to subdivide the mesh, one has an *h*-method (*h* is customarily the diameter of the largest element in the mesh.) In this manner, if one shows that the error with a grid $h$ is bounded above by $Ch^p$, for some $C < \infty$ and $p > 0$, then one has an order $p$ method. Under certain hypotheses (for instance, if the domain is convex), a piecewise polynomial of order $d$ method will have an error of order $p = d + 1$.

If instead of making *h* smaller, one increases the degree of the polynomials used in the basis function, one has a *p*-method. If one combines these two refinement types, one obtains an *hp*-method (hp-FEM). In the hp-FEM, the polynomial degrees can vary from element to element. High order methods with large uniform *p* are called spectral finite element methods (SFEM). These are not to be confused with spectral methods.

For vector partial differential equations, the basis functions may take values in $\mathbb{R}^n$.

# Various types of finite element methods

### AEM

The Applied Element Method, or AEM combines features of both FEM and Discrete element method, or (DEM).

## Generalized finite element method

The generalized finite element method (GFEM) uses local spaces consisting of functions, not necessarily polynomials, that reflect the available information on the unknown solution and thus ensure good local approximation. Then a partition of unity is used to "bond" these spaces together to form the approximating subspace. The effectiveness of GFEM has been shown when applied to problems with domains having complicated boundaries, problems with micro-scales, and problems with boundary layers.[13]

## Mixed finite element method

The mixed finite element method is a type of finite element method in which extra independent variables are introduced as nodal variables during the discretization of a partial differential equation problem.

## Variable – polynomial

The hp-FEM combines adaptively, elements with variable size $h$ and polynomial degree $p$ in order to achieve exceptionally fast, exponential convergence rates.[14]

## hpk-FEM

The hpk-FEM combines adaptively, elements with variable size $h$, polynomial degree of the local approximations $p$ and global differentiability of the local approximations $(k-1)$ in order to achieve best convergence rates.

## XFEM

The extended finite element method (XFEM) is a numerical technique based on the generalized finite element method (GFEM) and the partition of unity method (PUM). It extends the classical finite element method by enriching the solution space for solutions to differential equations with discontinuous functions. Extended finite element methods enrich the approximation space so that it is able to naturally reproduce the challenging feature associated with the problem of interest: the discontinuity, singularity, boundary layer, etc. It was shown that for some problems, such an embedding of the problem's feature into the approximation space can significantly improve convergence rates and accuracy. Moreover, treating problems with discontinuities with XFEMs suppresses the need to mesh and remesh the discontinuity surfaces, thus alleviating the computational costs and projection errors associated with conventional finite element methods, at the cost of restricting the discontinuities to mesh edges.

Several research codes implement this technique to various degrees: 1. GetFEM++ 2. xfem++ 3. openxfem++

XFEM has also been implemented in codes like Altair Radioss, ASTER, Morfeo and Abaqus. It is increasingly being adopted by other commercial finite element software, with a few plugins and actual core implementations available (ANSYS, SAMCEF, OOFELIE, etc.).

## Scaled boundary finite element method (SBFEM)

The introduction of the scaled boundary finite element method (SBFEM) came from Song and Wolf (1997).[15] The SBFEM has been one of the most profitable contributions in the area of numerical analysis of fracture mechanics problems. It is a semi-analytical fundamental-solutionless method which combines the advantages of both the finite element formulations and procedures, and the boundary element discretization. However, unlike the boundary element method, no fundamental differential solution is required.

## S-FEM

The S-FEM, Smoothed Finite Element Methods, are a particular class of numerical simulation algorithms for the simulation of physical phenomena. It was developed by combining meshfree methods with the finite element method.

## Spectral element method

Spectral element methods combine the geometric flexibility of finite elements and the acute accuracy of spectral methods. Spectral methods are the approximate solution of weak form partial equations that are based on high-order Lagragian interpolants and used only with certain quadrature rules.[16]

## Meshfree methods

## Discontinuous Galerkin methods

## Finite element limit analysis

## Stretched grid method

## Loubignac iteration

Loubignac iteration is an iterative method in finite element methods.

# Link with the gradient discretisation method

Some types of finite element methods (conforming, nonconforming, mixed finite element methods) are particular cases of the gradient discretisation method (GDM). Hence the convergence properties of the GDM, which are established for a series of problems (linear and non linear elliptic problems, linear, nonlinear and degenerate parabolic problems), hold as well for these particular finite element methods.

# Comparison to the finite difference method

The finite difference method (FDM) is an alternative way of approximating solutions of PDEs. The differences between FEM and FDM are:
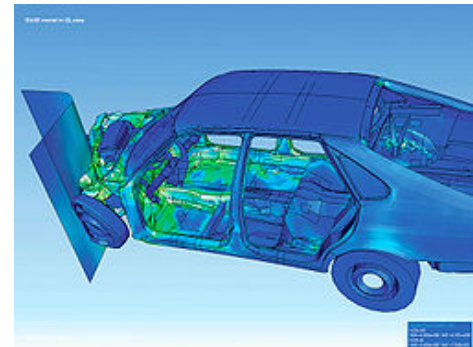
- The most attractive feature of the FEM is its ability to handle complicated geometries (and boundaries) with relative ease. While FDM in its basic form is restricted to handle rectangular shapes and simple alterations thereof, the handling of geometries in FEM is theoretically straightforward.
- FDM is not usually used for irregular CAD geometries but more often rectangular or block shaped models.[17]
- The most attractive feature of finite differences is that it is very easy to implement.
- There are several ways one could consider the FDM a special case of the FEM approach. E.g., first order FEM is identical to FDM for Poisson's equation, if the problem is discretized by a regular rectangular mesh with each rectangle divided into two triangles.
- There are reasons to consider the mathematical foundation of the finite element approximation more sound, for instance, because the quality of the approximation between grid points is poor in FDM.
- The quality of a FEM approximation is often higher than in the corresponding FDM approach, but this is extremely problem-dependent and several examples to the contrary can be provided.

Generally, FEM is the method of choice in all types of analysis in structural mechanics (i.e. solving for deformation and stresses in solid bodies or dynamics of structures) while computational fluid dynamics (CFD) tends to use FDM or other methods like finite volume method (FVM). CFD problems usually require discretization of the problem into a large number of cells/gridpoints (millions and more), therefore cost of the solution favors simpler, lower order approximation within each cell. This is especially true for 'external flow' problems, like air flow around the car or airplane, or weather simulation.

## Application

A variety of specializations under the umbrella of the mechanical engineering discipline (such as aeronautical, biomechanical, and automotive industries) commonly use integrated FEM in design and development of their products. Several modern FEM packages include specific components such as thermal, electromagnetic, fluid, and structural working environments. In a structural simulation, FEM helps tremendously in producing stiffness and strength visualizations and also in minimizing weight, materials, and costs.[18]

FEM allows detailed visualization of where structures bend or twist, and indicates the distribution of stresses and displacements. FEM software provides a wide range of simulation options for controlling the complexity of both modeling and analysis of a system. Similarly, the desired level of accuracy



Visualization of how a car deforms in an asymmetrical crash using finite element analysis.[1] (http://impact.sourceforge.net)

required and associated computational time requirements can be managed simultaneously to address most engineering applications. FEM allows entire designs to be constructed, refined, and optimized before the design is manufactured. The mesh is an integral part of the model and it must be controlled carefully to give the best results. Generally the higher the number of elements in a mesh, the more accurate the solution of the discretised problem. However, there is a value at which the results converge and further mesh refinement

does not increase accuracy.[19]

This powerful design tool has significantly improved both the standard of engineering designs and the methodology of the design process in many industrial applications.[21] The introduction of FEM has substantially decreased the time to take products from concept to the production line.[21] It is primarily through improved initial prototype designs using FEM that testing and development have been accelerated.[22] In summary, benefits of FEM include increased accuracy, enhanced design and better insight into critical design parameters, virtual prototyping, fewer hardware prototypes, a faster and less expensive design cycle, increased productivity, and increased revenue.[21]
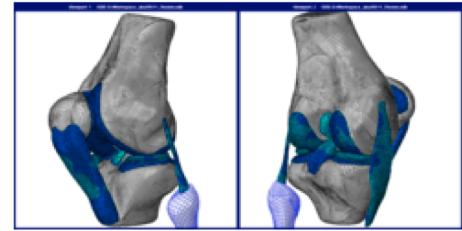


Finite Element Model of a human knee joint.[20]

In the 1990s FEA was proposed for use in stochastic modelling for numerically solving probability models[23] and later for reliability assessment.[24]

## See also

- Applied element method
- Boundary element method
- Céa's lemma
- Computer experiment
- Direct stiffness method
- Discontinuity layout optimization
- Discrete element method
- Finite difference method
- Finite element machine
- Finite element method in structural mechanics
- Finite volume method
- Finite volume method for unsteady flow
- Infinite element method
- Interval finite element
- Isogeometric analysis
- Lattice Boltzmann methods
- List of finite element software packages
- Meshfree methods
- Movable cellular automaton
- Multidisciplinary design optimization
- Multiphysics
- Patch test
- Rayleigh–Ritz method
- Space mapping
- Tessellation (computer graphics)

- Weakened weak form

# References

1. Daryl L. Logan (2011). *A first course in the finite element method*. Cengage Learning. ISBN 978-0495668251.
2. Reddy, J. N. (2006). *An Introduction to the Finite Element Method* (Third ed.). McGraw-Hill. ISBN 9780071267618.
3. "Finite Elements Analysis (FEA)" (http://www.manortool.com/finite-element-analysis.html). *www.manortool.com*. Retrieved 2017-07-28.
4. Hrennikoff, Alexander (1941). "Solution of problems of elasticity by the framework method". *Journal of Applied Mechanics*. **8** (4): 169–175.
5. Courant, R. (1943). "Variational methods for the solution of problems of equilibrium and vibrations". *Bulletin of the American Mathematical Society*. **49**: 1–23. doi:10.1090/s0002-9904-1943-07818-4 (https://doi.org/10.1090%2Fs0002-9904-1943-07818-4).
6. "СПб ЭМИ РАН" (https://web.archive.org/web/20150930001741/http://emi.nw.ru/INDEX.html?0%2Fresume%2Foganesan.htm). *emi.nw.ru*. Archived from the original (http://emi.nw.ru/INDEX.html?0/resume/oganesan.htm) on 30 September 2015. Retrieved 17 March 2018.
7. Hinton, Ernest; Irons, Bruce (July 1968). "Least squares smoothing of experimental data using finite elements". *Strain*. **4** (3): 24–27. doi:10.1111/j.1475-1305.1968.tb01368.x (https://doi.org/10.1111%2Fj.1475-1305.1968.tb01368.x).
8. "SAP-IV Software and Manuals" (http://nisee.berkeley.edu/elibrary/getpkg?id=SAP4). NISEE e-Library, The Earthquake Engineering Online Archive.
9. Gard Paulsen; Håkon With Andersen; John Petter Collett; Iver Tangen Stensrud (2014). *Building Trust, The history of DNV 1864-2014*. Lysaker, Norway: Dinamo Forlag A/S. pp. 121, 436. ISBN 978-82-8071-256-1.
10. Strang, Gilbert; Fix, George (1973). *An Analysis of The Finite Element Method* (https://archive.org/details/analysisoffinite0000stra). Prentice Hall. ISBN 978-0-13-032946-2.
11. Olek C Zienkiewicz; Robert L Taylor; J.Z. Zhu (31 August 2013). *The Finite Element Method: Its Basis and Fundamentals* (https://books.google.com/books?id=7UL5Ls9hOF8C&printsec=frontcover#v=onepage&q&f=false). Butterworth-Heinemann. ISBN 978-0-08-095135-5.
12. Bathe, K.J. (2006). *Finite Element Procedures*. Cambridge, MA: Klaus-Jürgen Bathe. ISBN 978-0979004902.
13. Babuška, Ivo; Banerjee, Uday; Osborn, John E. (June 2004). "Generalized Finite Element Methods: Main Ideas, Results, and Perspective". *International Journal of Computational Methods*. **1** (1): 67–103. doi:10.1142/S0219876204000083 (https://doi.org/10.1142%2FS0219876204000083).
14. P. Solin, K. Segeth, I. Dolezel: Higher-Order Finite Element Methods, Chapman & Hall/CRC Press, 2003

15. Song, Chongmin; Wolf, John P. (5 August 1997). "The scaled boundary finite-element method – alias consistent infinitesimal finite-element cell method – for elastodynamics". *Computer Methods in Applied Mechanics and Engineering*. **147** (3–4): 329–355. Bibcode:1997CMAME.147..329S (https://ui.adsabs.harvard.edu /abs/1997CMAME.147..329S). doi:10.1016/S0045-7825(97)00021-2 (https://doi.org /10.1016%2FS0045-7825%2897%2900021-2).

16. "Spectral Element Methods" (http://lsec.cc.ac.cn/~cjxu/SEM_mem.html). *State Key Laboratory of Scientific and Engineering Computing*. Retrieved 2017-07-28.

17. "What's The Difference Between FEM, FDM, and FVM?" (http://www.machinedesign.com/fea-and-simulation/what-s-difference-between-fem-fdm-and-fvm). *Machine Design*. 2016-04-18. Retrieved 2017-07-28.

18. Kiritsis, D.; Eemmanouilidis, Ch.; Koronios, A.; Mathew, J. (2009). "Engineering Asset Management". *Proceedings of the 4th World Congress on Engineering Asset Management (WCEAM)*: 591–592.

19. "Finite Element Analysis: How to create a great model" (https://coventivecomposites.com/explainers/finite-element-analysis-how-to-create-a-great-model/). *Coventive Composites*. 2019-03-18. Retrieved 2019-04-05.

20. Naghibi Beidokhti, Hamid; Janssen, Dennis; Khoshgoftar, Mehdi; Sprengers, Andre; Perdahcioglu, Emin Semih; Boogaard, Ton Van den; Verdonschot, Nico (2016). "A comparison between dynamic implicit and explicit finite element simulations of the native knee joint" (https://ris.utwente.nl/ws/files/6153316/CMBBE2014-Hamid-Submitted.pdf) (PDF). *Medical Engineering & Physics*. **38** (10): 1123–1130. doi:10.1016/j.medengphy.2016.06.001 (https://doi.org /10.1016%2Fj.medengphy.2016.06.001). PMID 27349493 (https://pubmed.ncbi.nlm.nih.gov/27349493).

21. Hastings, J. K., Juds, M. A., Brauer, J. R., *Accuracy and Economy of Finite Element Magnetic Analysis*, 33rd Annual National Relay Conference, April 1985.

22. McLaren-Mercedes (2006). "McLaren Mercedes: Feature - Stress to impress" (https://web.archive.org/web/20061030200423/http://www.mclaren.com/features /technical/stress_to_impress.php). Archived from the original (http://www.mclaren.com/features/technical/stress_to_impress.php) on 2006-10-30. Retrieved 2006-10-03.

23. Peng Long; Wang Jinliang; Zhu Qiding (19 May 1995). "Methods with high accuracy for finite element probability computing". *Journal of Computational and Applied Mathematics*. **59** (2): 181–189. doi:10.1016/0377-0427(94)00027-X (https://doi.org /10.1016%2F0377-0427%2894%2900027-X).

24. Haldar, Achintya; Mahadevan, Sankaran (2000). *Reliability Assessment Using Stochastic Finite Element Analysis*. John Wiley & Sons. ISBN 978-0471369615.

## Further reading

- G. Allaire and A. Craig: *Numerical Analysis and Optimization: An Introduction to Mathematical Modelling and Numerical Simulation (https://books.google.com /books?id=HIwSDAAAQBAJ&printsec=frontcover#v=onepage& q=%22finite%20element%22&f=false)*.
- K. J. Bathe: *Numerical methods in finite element analysis*, Prentice-Hall (1976).
- Thomas J.R. Hughes: *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis,* Prentice-Hall (1987).

- J. Chaskalovic: *Finite Elements Methods for Engineering Sciences*, Springer Verlag, (2008).
- Endre Süli: *Finite Element Methods for Partial Differential Equations* (http://people.maths.ox.ac.uk/suli/fem.pdf).
- O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu : *The Finite Element Method: Its Basis and Fundamentals*, Butterworth-Heinemann (2005).

## External links

- IFER – Internet Finite Element Resources (http://homepage.usask.ca/~ijm451/finite/fe_resources/) – describes and provides access to finite element analysis software via the Internet
- NAFEMS – International Association Engineering Modelling (http://www.nafems.org)
- Mathematics of the Finite Element Method (http://math.nist.gov/mcsd/savg/tutorial/ansys/FEM/)