IC/88/63
INTERNAL REPORT
(Limited distribution)

International Atomic Energy Agency
and
United Nations Educational Scientific and Cultural Organization

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS

FINITE ELEMENTS FOR PARTIAL DIFFERENTIAL EQUATIONS: AN INTRODUCTORY SURVEY *

Sauro Succi **
International Centre for Theoretical Physics, Trieste, Italy.

MIRAMARE – TRIESTE
March 1988

* Lecture Notes delivered at the II School on Advanced Techniques in Computational Physics, ICTP, Trieste, 18 January-12 February 1988.

** Permanent address: IBM/ECSEC, European Centre for Scientific and Engineering Computing, Via del Giorgione 159, 00147 Rome, Italy.

Introduction

The *Finite Element Method* (FEM) was invented to solve the complicated equations of elasticity and structural mechanics, two areas in which it has rapidly gained the role of leading computational technique. The merits of the method become particularly evident whenever the geometry plays an important role and the power of the computer is needed not only to *solve* a given problem but also to *formulate* and *assemble* it in a systematic and well organized way. This motivates the popularity of this method among engineers. However, the Finite Element Method is a powerful tool in general and consequently we believe that a knowledge of its main features deserves some attention also among the physicists community. In these lecture notes we present the basic features of the Finite Element Method with specific regard to the solution of ordinary and partial differential equations.
These notes are meant to serve as an introductory survey for those who wish to get acquainted with the subject without having any prior knowledge of it. The reader interested in a deeper and complete treatment is referred to the huge literature available on the subject. In particular, the following references are recommended.

1. A.J. Davies, *The Finite Element Method*, Oxford University Press, London, 1980.

2. G. Strang and J. Fix, *An analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, 1973

3. O.C. Zienkiewicz, *The Finite Element Method*, 3rd ed., Mc Graw Hill, New-York 1977.

These notes are subdivided in two main sections: the first one is essentially dealt with the exposition of the basic ideas behind the theory of the Finite Element Method. In the second part, we discuss in some detail the application of the method to three equations of particular interest in Physics and Engineering, notably a one-dimensional Sturm-Liouville problem, a two-dimensional (linear) Fokker-Planck equation and a two-dimensional (nonlinear) Navier-Stokes equation.

Part I- Theory

Let us consider the problem of finding a function $u(\bar{x})$ obeying the following partial differential equation:

$$\mathcal{D}u \equiv \sum_{ij=1}^{D} \partial_i A_{ij}(\bar{x},u)\partial_j u = f \qquad in\,\Omega \tag{1}$$

where Ω is a D-dimensional domain in R^D (D = 1,2,3), $\partial\Omega$ its frontier on which the boundary condition

$$\lambda u + \mu\partial_n u = g \tag{2}$$

is imposed. In the above equation ∂_n is the outward normal derivative along $\partial\Omega$ and λ, μ and g functions *defined in* $\partial\Omega$.

The ultimate goal of any numerical technique aiming to solve the eqs.(1-2) is to transform the differential problem into an algebraic one and then demand its solution to the computer. To this end, several approaches may be conceived; let us review the most important ones.

Global Projection Method

If the domain Ω is particularly regular (a square, a sphere ...) and the function $A_{ij}(\bar{x},u)$ are simple enough (for example constants), the equations (1,2) can be attacked by a series-expansion of the form:

$$u^N(\bar{x}) = \sum_{n=1}^{N} u_n^N B_n(\bar{x}) \tag{3}$$

where $B_n(\bar{x})$ is a suitable set of complete, orthogonal and orthonormal basis functions.

By plugging the expression (3) into equation (1), and projecting systematically again onto the set of functions B_m, $m = 1,...N$ we obtain a corresponding set of algebraic equations

$$\sum_{m=1}^{N} D_{nm} u_m = f_n \qquad , \tag{4}$$

where $D_{nm} = (B_n, \mathcal{D}\, B_m)$ and $f_m = (f, B_m)$, $(. , .)$ denoting the scalar product in the subspace H_N spanned by the "vectors" $B_1, B_2 ... B_N$.

The eq.(4) can be solved numerically to yield the sequence of values u_n^N(moments) and consequently the sought approximate solution $u^N(x)$ via eq.(3). Of course one expects that by rising the number of moments the approximate solution will get closer and closer to the exact one. Unfortunately, this is not always true in practice because of numerical problems which inhibit the progressive refinement of the solution predicted by the theory. The numerical "illness" of the method traces back to the *global* nature of the basis functions. To see this let us remind that global basis functions such as Hermite, Laguerre, Legendre polynomials are orthogonal, i.e. their Graham matrix is diagonal:

$$(B_n, B_m) = \delta_{nm} \tag{5}$$

This is a very important property because, were it not like that, the matrix D_{nm} would be full and very expensive to be solved. On the other hand, in order to ensure orthogonality, since these function are global, they must change sign in their set of definition so that their scalar product includes many contributions of opposite sign. Analytically these contributions sum exactly to zero, but in practice they never do it exactly because of the cancellation errors. This may quench the effective refinement of the solution one would expect by rising N.

We now see both faces of the medal: globality is good because it may allow it to capture the essential physics with only a few degrees of freedom. However, globality is also "bad" because in order to keep orthogonality it may force the equations to be numerically ill conditioned.

Finite Difference Method (FD)

A quite different approach is adopted in the Finite Difference Method. Here, one gives up with the idea of selecting a few "good" degrees of freedom at the outset and looks instead for a collection of a high number of ***local*** degrees of freedom. These are the values of the unknown on a discrete set of points $(\bar{x}_1, ...\bar{x}_N)$. Consistently with this discretization, the differential operators are replaced by the corresponding discrete version. As an example, in a one-dimensional lattice we have

$$\frac{du}{dx} \rightarrow \Delta^+ u \equiv \frac{(u_{i+1} - u_i)}{h} \qquad \frac{d^2u}{dx^2} \rightarrow \Delta^2 u \equiv \frac{(u_{i+1} - 2u_i + u_{i-1})}{h^2} \tag{6}$$

Consequently, the eq.(1) will transform to a difference equation of the form

$$\sum_{i=1}^{N} D_{ij} u_j = f_i \tag{7}$$

where $u_j \equiv u(x_j)$ and D_{ij} is a matrix of the same type as Δ^+ and Δ^2. Note that these matrices are very sparse (only a few non-zero elements per row); for example, for $N = 5$ the matrix generated by d/dx reads as:

$$(1/h) \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Again, one expects the solution of eq.(7) to reproduce the exact one in the limit $h \rightarrow 0$ since the error done in replacing $\frac{d}{dx}$ with Δ^+ is of order $O(h)$. In fact, this error is proportional to the departure of the function $u(x)$ from a linear behaviour in the interval $(x, x + h)$. Higher accuracy can be achieved by adopting centered differencing; for example the replacement

$$\frac{d}{dx} \rightarrow \frac{(u_{i+1} - u_{i-1})}{2h} \tag{8}$$

is readily verified to give second order accuracy if the mesh is ***uniform***. In any case, it is clear that in order for the FD method to work successfully the mesh discretization must be fine enough to prevent the solution from suffering too rapid changes within a mesh cell size.

FD is a powerful technique but it suffers of some drawbacks.

- Mesh clustering, requiring uncentered differencing, implies loss of accuracy
- Boundary Conditions are not easily handled if the boundary of the domain does not fall on a coordinate line

These problems can indeed be satisfactorily circumvented by the so called "Finite Volume" technique which is however already a way to go to the Finite Element Method philosophy.

The Finite Element Method (FEM)

The Finite Element method may loosely be viewed as an intermediate technique between the two ones previously mentioned. In fact, in FEM the two notions of globality and locality are both retained, albeit in a peculiar sense, as we are going to discuss. Let us start with globality. As is known, many problems in Physics and Engineering can be formulated in terms of **variational principles**.

Mathematically one defines a functional of the unknown field $u(x,y)$ as

$$I(u) = \int_\Omega L(u, u_x, u_y ...) dxdy \tag{9}$$

and shows that in order for I to attain an extremum, the field must obey the Euler-Lagrange differential equation:

$$F_u - (F_{u_x})_x - (F_{u_y})_y = 0 \tag{10}$$

where subcripts denote partial derivatives. As an example, by taking $L = (1/2)(u_x^2 + u_y^2)$ we obtain the Poisson's equation $u_{xx} + u_{yy} = 0$. This correspondence suggests that there is a way to look for approximate solutions of the problem (9) by working directly on the functional (10), that is on a quantity which sensitive to the *global* behaviour of the function $u(x,y)$.

An immediate advantage of this strategy is that while the solution of the Euler-Lagrange equation must be continuous up to the second derivative the minimization of the functional I can be achieved by requiring only the square-integrability of *first* order derivatives. Thus, the solution we are looking for is somehow "weaker" in the sense that the convergence is defined in the sense of distributions, i.e. in a broader functional space.

Ritz Method.

When the functional I is positive definite its minimization constitutes a well-posed problem which can be solved by expanding the unknown function onto a set of *basis* functions, the coefficients of the expansion being the parameters to be adjusted for the minimization. However, even when the differential operator is not positive definite and self-adjoint, one can resort to a similar strategy which is known as

Weak Formulation (WF).

This reads as follows;
Let v be an element of a certain functional space H_t (v is usually referred to as *test* function) the WF of the problem (1) is:

Find u in H such that for any v in H_t

$$b(u,v) \equiv (v, \mathcal{D}u - f) = 0 \qquad (11)$$

Here the notation $b(u,v)$ stresses the idea that the scalar product is a bilinear functional, (which needs not be positive definite). It should be noted that the above equation is 'weaker' than its classical counterpart eq.(1) in the sense that any solution of (1) is also a solution of (11) but not viceversa. In fact eq.(11) replaces a requirement of equality between Du and f with a statement of *orthogonality* between the test function v and the residual $r \equiv Du - f$. In a geometrical sense, r is forced to lie in the functional space $\overline{H}_N$, the orthogonal complement to $H_N \equiv span(v_1, ...v_N)$ $(H_N + \overline{H}_N = H)$. Thus, by rising N, r is forced to lie in a narrower and narrower functional space which, in the limit $N \to \infty$ becomes empty. Obviously, the residual r, belonging to an empty set, is forced to vanish!

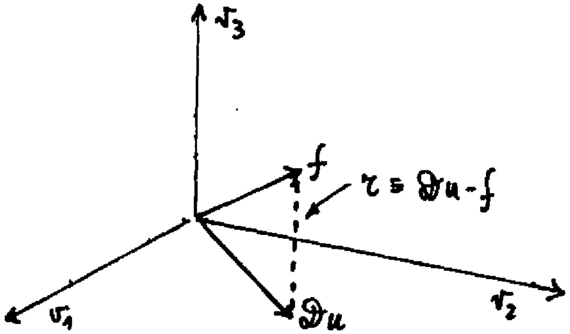


Fig.1; Geometrical representation of convergence

The weak formulation has a general significance and is not tied to any specific choice concerning the functional space H to be adopted for the minimization of the functional b. The FEM represents indeed *one* possible way to make such a choice.
The method consists basically of two steps

- Geometrical approximation
- Polynomial approximation

Geometrical Approximation

The first step consists of replacing the domain Ω with a corresponding computational domain Ω_h obtained by partitioning Ω into a set of N_e disjoint subdomains (*elements*) Ω_i such that

$$\Omega_i \cap \Omega_j = O, \quad \bigcup_{i=1}^{N_e} \Omega_i = \Omega_h.$$

Note that Ω and Ω_h need not coincide: they certainly don't if Ω is unbounded or if its boundaries are not conformal to the geometrical shape of the subdomains Ω_i.

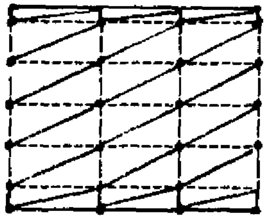


Fig.2; FEM Triangulation of the computational domain

On each vertex (*node*) $\bar{x}_i$ we place a function $\Psi_i(\bar{x})$ which is equal to one at $\bar{x} = \bar{x}_i$ and zero outside the simplex S_i defined by the union of all the elements Ω_c that share $\bar{x}_i$ as a common vertex (See Fig. below)

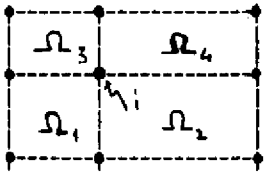


Fig.3; A rectangular support and its surrounding elements

In the current terminology, S_i is called the **support** of the **finite element** Ψ_i.
In general

$$S_i = \bigcup_{c=1}^{C} \Omega_c$$

where C is the coordination number of the lattice (i.e. the number of links out of a nodal point). For example, in the case of figure above, we have $C = 4$.

Piecewise Polynomial Approximation

Inside its support the function Ψ_i is taken in the form of a piece-wise polynomial of degree p

$$\Psi_i(\bar{x}) = \Pi_i(\bar{x}), \quad \bar{x} \in S_i \tag{12}$$

$$\Psi_i(\bar{x}) = 0 \qquad \textit{elsewhere} \tag{13}$$

with the constraint that $\Pi_i(x_j) = \delta_{ij}$, that is Π_i is one on the centroid of its support and zero on its vertices. Consequently, Π_i can be expressed as a superposition of C polynomials of degree p each of which is non-zero only outside $\Omega_c \in S_i$. These polynomials are often called "spline" functions. Thus each finite element Ψ is the sum of the spline functions associated with the ensemble of the elements which form its support.

We are already able to appreciate the flexibility of this approach with respect to the geometry: a proper choice of the shape of the subdomains Ω_i, for example triangles, allows it to accurately approximate very complicated geometrical shapes. This is the main motivation for the success of FEM in engineering applications.

Each finite element can be regarded as a "unit vector" in the functional space $H_N = Span(\Psi_1, ...\Psi_N)$ generated by paving the computational domain with the Ω_i's. Note that, contrary to the elements Ω_i, the supports S_i in general **do overlap**. This means that in general the set of approximating functions is only "nearly" orthogonal in the sense that the scalar product (Ψ_i, Ψ_j) is different from zero only for those values of j associated with the neigborhood of i. Obviously this neighborhood includes S finite elements, S being the number of vertices of the support S_i.
As a result, the notion of locality is again restored and the matrices generated by FEM are sparse as in the FD method.

Having defined the approximating subspace, it is now natural to look for a solution $u(x,y)$ in the form

$$u(x,y) = \sum_{i=1}^{N} u_i \Psi_i(x,y) \tag{14}$$

Given the fact that $\Psi_i(\bar{x}_j) = \delta_{ij}$ we recognize that the coefficients u_i are the values of the function $u(x,y)$ at the nodal points. On the other hand, it is intuitively clear that as the degree of the interpolating polynomials is raised, the FEM representation should give more information than simply the nodal values of u. To fix this point concretely, let us consider the case of triangles $T_l \equiv \Omega_l$. The restriction of $u(x,y)$ to each triangle, say $u_l(x,y)$ reads as follows:

$$u_l(x,y) = \sum_{q+r}^{p} u_{qrl} x^q y^r \tag{15}$$

We see that in each triangle we have $p(p + 1)/2$ coefficients u_{qrl}, i.e. $DF = p(p + 1)/2$ degrees of freedom. These can be exploited to specify

- The values of the function on nodal and internodal points (Lagrange interpolation)

- The values of the function and its derivatives at the nodal points (Hermite Interpolation)

With $p = 1$ we have DF=3 which means we can only compute the value of the function at the three nodal points. With $p = 2$, DF=6 so that we can additionally compute the values of the function at the mid-points of the sides. With $p = 3$, DF = 10 and can either compute the tri-section nodal values (Lagrange) plus the value at the center or the function together with its gradient on the nodal points plus the central value (Hermite interpolation).

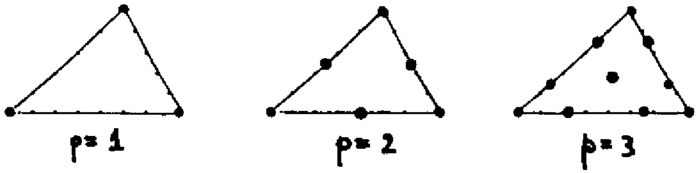


Fig.4; Lagrangian interpolation with triangular elements

We see that rising the value of p is tantamount to refining the original grid and should therefore increase the accuracy of the method.

Convergence

These heuristic considerations have a mathematical counterpart in a series of theorems stating the dependence of the FEM error as a function of the upper size h_M (the diameter) of the elements Ω_i.
To this purpose, one introduces the Sobolev space H_k defined as the space of the functions which are square-integrable up to the $k - th$ derivative.

$$H_k = (u, \int (u^2 + u'^2 + ...u^{(k)2})dx < \infty) \qquad (16)$$

Note that the sequence H_k is mutually inclusive, i. e. $H_0 \supset H_1 \supset H_2 ... \supset H_k$.
Typically, the error measured in H_k scales like

$$E(h_M) \sim h_M^{(p+1-k)} \qquad (17)$$

which shows that in the "largest" subspace H_0 (the familiar Hilbert space) the method has $p + 1$-th order accuracy.

A question arises on which value should one choose for p. The trade-off is clear: by rising p one enhances the convergence rates but obviously it also increases the computational work. The choice is by no means universal and must be selected case by case. In principle, a "professional" code should contemplate several options for different types of elements, especially when the problem is so complicated that no underlying convergence theory exists.

In any case, it is worth stressing that the error scaling with h is independent of the uniformity of the mesh. This means that one can accumulate and rarefy nodes in different regions of the computational domain without loosing the overall accuracy of the method. The mesh-densification strategies, usually suggested by the physical phenomenology, may be very effective in saving computing time and storage. In engineering applications this is so important that the task of generating the mesh is entirely demanded to a separate program (pre-processor) to be run interactively several times before the submission of the "physical" job.

Boundary Conditions

Boundary conditions are usually divided into two basic classes

- Natural

- Essential

By natural one refers to the fact that they can be automatically built-in in the variational formulation without introducing any constraint on the approximating functional subspace. Typical examples are Neumann boundary conditions in which the surface contribution arising from the integration by parts is discarded at the outset and consequently does not appear at all in the functional $b(u, v)$. Essential boundary conditions are those for which such a simplification does not occurr. Typically, Dirichlet boundary conditions are essential.

We are now in a position to summarize the main merits of the method

1. The notion of globality is retained through the weak-formulation

2. The locality, i. e., the sparsity of the matrices is ensured by the very definition of the finite elements that are non-zero only inside their support.

3. By a suitable choice of the elements complicated geometrical domains can be approximated accurately.

4. Boundary conditions are handled in a systematic way

Point 1. is important because it guarantees that the method is optimal in the sense of the weak formulation, i.e. from a global point of view. Point 2. ensures the sparsity of the matrices without paying the price of "ill-conditioning" that affects the projection methods based on global functions. Point 3. and partly point 4. are the keys of the success of FEM in all those applications where the geometry plays a dominant role.

Part II- Applications

In this second part we offer some examples of the application of FEM to three equations of Physics:

- A one-dimensional Sturm-Liouville problem
- A two-dimensional Fokker-Planck equation
- A two-dimensional Navier-Stokes equation

Application N.1: Sturm-Liouville in one dimension.

Let us consider the following one-dimensional Sturm-Liouville problem:

$$-(p(x)y')' + q(x)y = 0, \quad (p(x),\ q(x) > 0) \tag{18}$$

(prime stands for d/dx) with Dirichlet boundary conditions:

$$u(0) = a, u(1) = b \tag{19}$$

Equations (18-19) constitute the **strong formulation** of the problem; for any regular function $f(x)$ one looks for a solution $u(x)$ at least differentiable twice. The differential equation (18), however, is the Euler equation for the following extremum principle:

$$I = \int_0^1 (py'^2 + qy^2)dx \tag{20}$$

Since only the function and its derivative are involved, the functional space over which the minimization is to be carried on, can be made up with piecewise linear elements ("hat" functions):

$$e_i(x) = \frac{x - x_i}{x_i - x_{i-1}} \quad x_{i-1} < x < x_i \tag{21}$$

$$e_i(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i} \quad x_i < x < x_{i+1} \tag{22}$$

$$e_i(x) = 0 \qquad \textit{elsewhere} \tag{23}$$

Thus, we look for an approximate solution in the form:

$$y(x) = ae_0(x) + \sum_{i=1}^{N-1} y_i e_i(x) + be_N(x) \tag{24}$$

where the boundary conditions have manifestly been forced in by imposing $y_0 = a$ and $y_N = b$. Clearly, this solution is continuous but not differentiable.

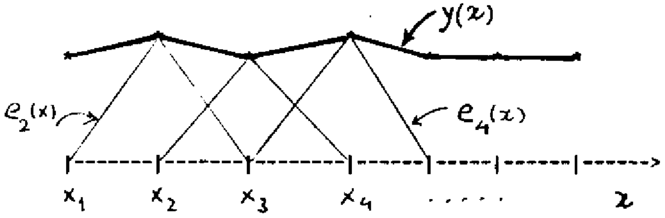


Fig.5; FEM representation with hat functions

By inserting the expression (24) in (20), the functional I becomes a quadratic function of the array $u_1 ... u_N$ whose minimum is determined by the condition :

$$\frac{\partial I}{\partial u_i} = 0 \tag{25}$$

It easy to verify that the above condition translates into the following set of algebraic equations:

$$\sum_{j=1}^{N-1} (P_{ij} + Q_{ij})y_j = f_i \tag{26}$$

where

$$P_{ij} = \int_0^1 e_i'(x)p(x)e_j'(x)dx \tag{27}$$

$$Q_{ij} = \int_0^1 e_i(x)q(x)e_j(x)dx \tag{28}$$

$$f_i = \int_0^1 f(x)e_i(x)dx - a\delta_{01}(P_{ij} + Q_{ij}) - b\delta_{N,N-1}(P_{ij} + Q_{ij}) \tag{29}$$

A number of important properties of the matrices P and Q can be established independently of the specific values of their elements; these are:

- symmetry

- positive definitiness

- sparsity

Symmetry, which is readily verified by exchanging l and j in the above expressions is a direct consequence of the self-adjointness of the Sturm-Liouville operator. Positive definiteness stems from the inequality

$$\int_a^b (py'^2 + qy^2)dx > 0 \tag{30}$$

and sparsity is due to the fact that the basis functions are localized. More precisely, since each basis function overlaps only with its nearest neighbours, only the elements $(l, l-1), (l, l)$ and $(l, l+1)$ are non-zero. Therefore P and Q are tri-banded matrices, as depicted below.

```
------------
|DR         |
| LDR       |
|  LDR      |
|   LDR     |              L <--- D ---> R
|    LDR    |
|     LDR   |
|      LDR  |              L: Left
|       LDR|               D: Diagonal
|        LD|               R: Right
------------
```

Fig. 6; Tridiagonal matrices generated by the Sturm-Liouville problem

This properties make the system (26) particularly well-behaved and cheap to solve on a computer. In terms of storage we only need three arrays, that is $3N$ computer words (in fact just $2N$ because of symmetry). The best solution algorithm is the LU decomposition which in this particular case reduces to a double recurrence known as the Thomas alghoritm, requiring roughly $6N$ floating point operations.

Computing the matrix element

According to the definition of the hat functions e_l, the matrix elements become:

$$P_{ij} = \int_{x_{l-1}}^{x_{l+1}} e_i'(x)p(x)e_j'(x)dx \quad , \quad Q_{ij} = \int_{x_{l-1}}^{x_{l+1}} e_i(x)q(x)e_j(x)dx \tag{31}$$

These integrals may rarely evaluated exactly on analytical grounds. Even when this is the case, it is preferable to compute them by a numerical quadrature technique which has the advantage of being more flexible and general. The most commonly method adopted is the Gauss quadrature in which the integrals are replaced by weigthed sums

$$\int_a^b f(x)dx = \sum_{g=1}^{G} f(x_g)w_g \tag{32}$$

where x_g, w_g are the Gauss nodes and weights respectively. The G-point formula is exact for polynomials up to degree $(2G - 1)$. In practice it makes no sense to push G to higher values than those required to ensure the same accuracy of the overall discretization procedure. So, for linear finite elements which yield second order accuracy $G = 2$ is perfectly in order.

From the practical point of view, the matrix coefficients are best evaluated on a element-by-element ground. For each interval $I_l \equiv (x_l, x_{l+1})$ we count four contributions

- $d_l(A) \equiv A_{ll}$

- $r_l(A) \equiv A_{l,l+1}$

- $l_{l+1}(A) \equiv A_{l+1,l}$

- $d_{l+1}(A) \equiv A_{l+1,l+1}$

where l, d, r stand for "left,diagonal,right" respectively and A is a generic matrix. The matrices P and Q can thus be assembled once forever with a single pass through the mesh, (a single DO-LOOP) in a fairly sound and systematic way. It is now instructive to examine the case in which $p(x)$ and $q(x)$ are both constant, say p_0 and q_0. The matrix elements take the simple form:

$$P_{ij} = p_0 \frac{1}{2h}(-1 \quad 0 \quad 1) \equiv p_0(K_{i,j-1}, K_{ij}, K_{i,j+1}) \tag{33}$$

$$Q_{ij} = q_0 \frac{h^1}{6}(1 \quad 4 \quad 1) \equiv q_0(M_{i,j-1}, M_{ij}, M_{i,j+1}) \tag{34}$$

The matrices M and K are usually referred to as the "mass" and "stiffness" matrices respectively. These denominations become clear if one considers that the finite element discretization (always with 'hat' functions) of the wave equation (think of u as to the vertical displacement of a one-dimensional string)

$$u_{tt} - u_{xx} = 0 \tag{35}$$

reads precisely:

$$\sum_{j=1}^{N} M_{ij}\ddot{u}_j = \sum_{j=1}^{N} K_{ij}u_j \tag{36}$$

It is also worth comparing eq.(36) with the corresponding finite difference version:

$$\ddot{u}_j = h^{-2}(u_{j,j+1} - 2u_{j,j} + u_{j,j-1}) \tag{37}$$

We see that, apart from a common factor h, the stiffness matrix is exactly the same, while the mass matrix is not. In particular we have:

$$M_{ij}^{FE} = \frac{h}{6}(\delta_{i,j+1} + 4\delta_{i,j} + \delta_{i,j-1}) \tag{38}$$

to be compared (after the division by a common factor h) with

$$M_{ij}^{FD} = \delta_{ij} \tag{39}$$

The latter expression shows that in FEM each node i "gives" part of its mass=inertia to its nearest neighbours, in such a way that even **local** operators (i.e. operators with no differential structure) give rise to non-diagonal matrices. Note however that the FD and the FE matrices are equal *in average*;

$$\sum_{i=1}^{3} M_{ij}^{FD} = \sum_{i=1}^{3} M_{ij}^{FE} \tag{40}$$

This way of averaging is associated with the non-orthogonality of the basis functions. In fact the mass matrix $M_{ij}^{FE} = (\Psi_i, \Psi_j)$ is diagonal only if the basis functions are piecewise constants.

Differently restated, the eq.(36) is the Finite Difference version of the following integro-differential equation

$$\int_{x-h}^{x+h} u_{tt}\,dx - u_{xx} = 0 \tag{41}$$

where the integral is evaluated with the Sympson rule. The FD discretization of u_{xx} is second order accurate: we then realize that the averaging is somehow a "trick" to ensure the same accuracy also for the u_{tt} term.

As a general remark, we can say that any FE discretization has a FD counterpart on a corresponding "averaged" equation. The point is that with FD a certain experience is needed to find out the appropriate scheme ensuring the desired level of accuracy, whereas with FEM this comes along in a much more systematic and sound fashion through the initial choice of the approximating subspace.

Application N.2: Fokker-Planck Equation

In this section we consider a much more difficult problem with respect to one examined in the previous case.
The problem is to solve the following advection-diffusion equation in two dimensions:

$$\partial_t f + div\bar{J} = 0, \quad \bar{J} = -(\bar{R}f + \bar{\bar{D}} grad f) \ , \ in\ R \tag{42}$$

with boundary conditions

$$\bar{J} \bullet \hat{n} = 0 \ \ along\ \partial R \tag{43}$$

R being a two dimensional rectangular domain, $\hat{n}$ the outward normal along the boundary. Such an equation is often encountered in physics; in particular it is very popular in Plasma Physics to describe the kinetic evolution of the electron (ion) distribution function.
This problem shows the following difficulties with respect to the Sturm-Lioville problem:

- Time dependence
- Two dimensionality
- Non self-adjointness

In particular, non self-adjointness stems from the presence of the advective term $\bar{R}f$, which prevents the possibility of finding a functional associated with eq.(42) endowed with the property of positive definiteness. This forces us to abandon the Ritz minimum principle in favour of the more general weak-formulation outlined in the first part of this lectures.

If g is a test function in the Hilbert space $H(R)$, the weak form of the eq.(42) reads as follows: "For each g belonging to $H(R)$, find f such that"

$$(g, \partial_t f + div\bar{J}) = 0 \tag{44}$$

After integrating by parts and discarding the boundary contribution in force of the Neumann boundary condition, this becomes

$$(g\,, \partial_t f - grad\, g \bullet \bar{J}) = 0 \tag{45}$$

We now look for a time dependent solution in the form

$$f(u,v,t) = \sum_{l=1}^{N} f_l(t)\Psi_l(u,v) \tag{46}$$

where l is a two dimensional index and u, v are the components of the velocity along x and y. By plugging the eq.(46) into eq.(45) and identifying g with Ψ_l for $l = 1, N$ respectively we obtain a system of N ordinary differential equation, which are the equations of motion of the time dependent amplitudes $f_l(t)$.
These read as:

$$\sum_{j=1}^{N} A_{lj}\dot{f}_j = \sum_{j=1}^{N} B_{lj}f_j \tag{47}$$

where

$$A_{lj} = (\Psi_l, \Psi_j) \tag{48}$$

$$B_{lj} = (-grad\,\Psi_l, \bar{R}\Psi_j + \overline{\overline{D}}grad\Psi_j) \tag{49}$$

The time variable is usually treated by a **Finite Difference** technique:

$$df/dt \rightarrow (f^{(n+1)} - f^n)\Delta t^{-1} \tag{50}$$

$$f \rightarrow (\theta f^{(n+1)} + (1-\theta)f^n) \tag{51}$$

where θ is an interpolation parameter in the range (0,1). In particular, the choice $\theta = 0$ corresponds to a fully *explicit* (forward Euler) and $\theta = 1$ to a fully *implicit* time integration. The intermediate case $\theta = 0.5$, known as *Crank-Nicolson* method is also frequently adopted.

Rearranging for f^{n+1} in terms of f^n we obtain

$$(A - B\Delta t\theta)f^{n+1} = (A + B\Delta t(1-\theta))f^n \tag{52}$$

where we have omitted the spatial indices. This formally inverts to

$$f^{n+1} \equiv P^{n \rightarrow n+1} f^n = (A - B\theta\Delta t)^{-1}(A + B(1-\theta)\Delta t)f^n \tag{53}$$

where P is the "time propagator" .

Obviously, at each time-step the practical construction of the propagator requires the solution of the algebraic system, eq.(52). In performing this task, one must ensure the following properties of the propagator:

- Consistency

- Accuracy

- Stability

Consistency is the requirement that P reduces to the identity in the limit $\Delta t = 0$. It is readily cheked that this is the case for our scheme since $P = A^{-1}A \equiv 1$ for $\Delta t = 0$. Accuracy is related to the discretization error introduced when replacing the continuous derivative with a discrete difference.
Finally, stability is the requirement that the discretization error does not amplify from step to step. Mathematically, one requires:

$$||P|| < 1 \tag{54}$$

where $||P||$ is some norm of the matrix P. We can get a qualitative insight into the role of the parameter θ by considering the function

$$F(x,\theta) = \left|\frac{1 + x(1-\theta)}{1 - x\theta}\right| \tag{55}$$

where x stands for some representative element of the matrix $A^{-1}B\Delta t$. The two extreme cases of fully explicit and fully implicit time integration yield respectively:

$$\theta = 0 \rightarrow F = |1 + x| \qquad \theta = 1 \rightarrow F = \frac{1}{|(1-x)|} \tag{56}$$

Since A is positive defined, if we assume that B is negative defined (as it is case for a purely diffusion process), x is a negative number. This means that for x sufficiently large one has $P(\theta = 1) < P(\theta = 0)$.
This highlights the fact that implicit methods allow it to maintain larger values of Δt which is very helpful in long time simulations involving disparate time-scales.

The structure of the matrices

We have already remarked the FEM gives rise to sparse matrices; the degree of sparseness depending on the order of the interpolant polynomials. In the present case, the "minimal" set of basis functions is given by **bilinear** "hat" functions of the form:

$$\Psi_l(u,v) = e_{l1}(u) \times e_{l2}(v) \tag{57}$$

where e_l are the linear basis functions introduced in the previous section. With bilinear elements we put 4 unknowns per element = quadrilateral, which means that again we guarantee only the continuity of the *approximated solution.*

It is easy to verify that the corresponding matrices are block-tridiagonal, each block having exactly the *one-dimensional structure already encountered in the previous application.*

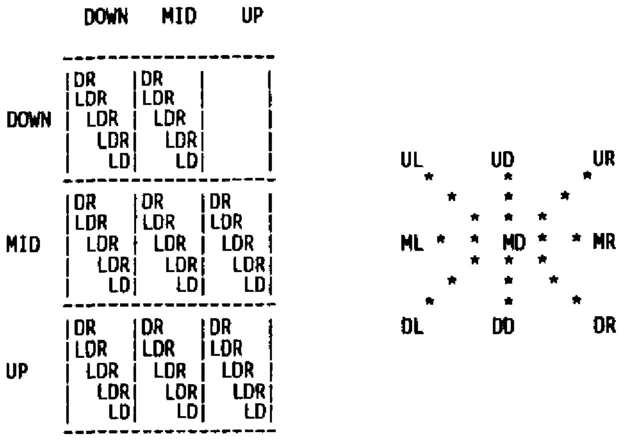


Fig.7; The block-tridiagonal matrices generated by bilinear elements

Assuming the coefficient functions R and D constants, the matrices M, $R = RX + RY$ and $D = DXX + DXY + DYY$ take the simple form (element-wise):

$$M = \frac{h^2}{36} \qquad \begin{matrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{matrix}$$

$$RX = \frac{h}{12} \qquad \begin{matrix} -1 & 0 & 1 \\ -4 & 0 & 4 \\ -1 & 0 & 1 \end{matrix}$$

$$RY = \frac{h}{12} \qquad \begin{matrix} -1 & -4 & -1 \\ 0 & 0 & 0 \\ -1 & -4 & -1 \end{matrix}$$

$$DXX = \frac{1}{4} \qquad \begin{matrix} 1 & -2 & 1 \\ -2 & 4 & 2 \\ 1 & -2 & 1 \end{matrix}$$

and $DXY = DYY = DXX$ if the mesh is uniform. One recognizes that owing to the reducibility of the bilinear basis functions, these matrices can be decomposed into a direct product of two one-dimensional matrices. The matrices M and D are symmetric and positive definite, and consequently well-behaved. R is anti-symmetric and ill-behaved since the diagonal elements are zero.

In particular, the eigenvalues of R are purely imaginary and give rise to spurious oscillations in the numerical solution (overshoots) which break the requirement of positivity of the solution.
These spurious oscillations are a manifestation of **Numerical Dispersion**, i.e., the fact that the discrete mesh alters the dispersion relation governing the propagation of independent modes. To understand this, let us refer to the simple hyperbolic problem

$$u_t + cu_x = 0 \tag{58}$$

Analytically, we know that at any time the solution has the form $u(x, t) = u_0(x - ct)$, $u_0(x)$ being the initial profile. The initial profile does not get distorted because all the independent modes contributing to u_0 travel at the same constant speed c. In fact, by Fourier-Laplace transforming the above equation one obtains the well-known dispersion relation:

$$D(\omega, k) \equiv = \omega - ck = 0 \tag{59}$$

The effect of the discrete lattice is to distort this dispersion relation which takes approximately the form

$$\tilde{D}(\omega, k, \Delta t, \Delta x) \equiv \frac{\cos\omega\Delta t - 1}{\Delta t} - c\frac{\cos k\Delta x - 1}{\Delta x} = 0 \tag{60}$$

It is worth noting that the condition $D = 0$ reflects the invariance of the eq.(58) with respect to the continuous group of traslations in space and time (Galilean invariance). In the lattice D is replaced by $\tilde{D}$ because the Galilean invariance holds only if the traslations amplitude is a multiple integer of the mesh spacing ($\tilde{D} = 0$ for $\omega = 2n\pi$, $k = 2m\pi$). The numerical dispersion relation is consistent, in the sense that in the continuum limit $k\Delta x \to 0$ and $\omega\Delta t \to 0$ the exact relation is recovered. This indicates that the breaking of Galilean invariance is especially caused by high-frequency short-wavelength modes. These modes are not well resolved by the lattice and consequently "see" and interact with its discrete nature thereby getting distorted. A widely used remedy against this problem consists of introducing an artificial diffusion term in the equation of the form δu_{xx}. This new term changes the dispersion relation to

$$D(\omega, k) \equiv = \omega - ck + i\delta k^2 = 0 \tag{61}$$

where we see the emergence of a damping term which is particularly effective for short-wavelength modes. It is easy to show that, in the FE formalism, the artificial diffusion method (called "Up-winding") is equivalent to adopt two distinct sets for the basis and the test functions respectively. In particular, if Ψ is the test function, the basis function can be chosen in the form $\Psi + \Delta x\bar{R} \bullet grad\,\Psi$.
The parameter which governs the amplitude of the spurious oscillations is the Mesh Reynolds Number, defined as

$$Re = \frac{Rh}{D} \tag{62}$$

This parameter, measuring the ratio convection/dissipation, arises naturally by requiring that the physical information should not travel "too fast" in the grid. This translates to the following inequalities:

$$Physical\,Speed \equiv R < Numerical\,Speed \equiv \frac{h}{\Delta t} \tag{63}$$

$$Physical\ Diffusion \equiv D < Numerical\ Diffusion \equiv \frac{h^2}{\Delta t} \qquad (64)$$

whence the condition $Re < 1$.

To follow the evolution of the function f one has to solve a linear algebraic system at each time-step. Depending on the size of the problem, one can resort to a direct (Gauss elimination) or to an iterative solver (Gauss-Seidel, Conjugate Gradient). When using the direct Gauss solver, care has to be taken to number the unknowns in such a way as to minimize the matrix bandwidth: for rectangular domain this reduces to number the nodes row by row (XY) or column by column (YX) according whether $NY > NX$ or $NX > NY$ respectively. In principle, the iterative solvers may get troubles because the matrix to be inverted is not symmetric-positive-definite. However, recent work with modified CG schemes has proven quite successfull for this type of problem. A typical cost on a present-day high-speed computer (some tens of Megaflops) is of the order of 0.1 milliseconds per grid-point.

Application N.3: Navier Stokes

The Navier-Stokes equation is of fundamental importance in applied and theoretical fluid-dynamics.
For an incompressible fluid ($div\bar{u} = 0$) *it reads*

$$\bar{u}_t + (\bar{u} \bullet grad)\bar{u} = \nu\Delta\bar{u} - grad p \qquad (65)$$

where $\bar{u}$ is the fluid velocity, ν the kinematic viscosity and p the scalar pressure field.
Peculiar difficulties associated with the treatment of this equation are:

- Non-linearity

- Incompressibility

- The unknown is vector valued

The non-linear term is difficult to treat because it tends to produce short wavelengths in the flow by quadratic mode-mode coupling. These short-wavelength modes become particular dangerous whenever they "hide" themselves in the grid, which is whenever their wavelength becomes smaller than the grid spacing. It can be shown that if L denotes the typical macroscopic length scale of the flow, the shortest scale which needs to be represented adequately in the mesh (dissipative scale), is given by $l = LR^{-0.5}$ (in two *dimensions*).
This shows that as the Reynolds number increases, it becomes more and more necessary to increase the mesh resolution.

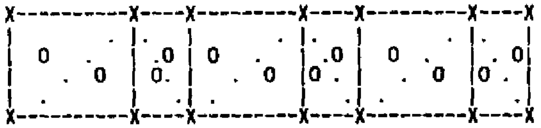
The compressibility condition, $div\bar{u} = 0$, puts some constraints on the class of finite elements.
To see this, let us consider the following FE expansion of the unknowns

$$u_1 = \sum_{i=1}^{N} u_{1i}(t)\Psi_i(x,y) \qquad (66)$$

$$u_2 = \sum_{i=1}^{N} u_{2i}(t)\Psi_i(x,y) \qquad (67)$$

$$p = \sum_{i=1}^{M} p_i(t)\Phi_i(x,y) \qquad (68)$$

By plugging these expression in the eq.(65) we obtain $2N$ equations for $2N + M$ unknowns, *in such a* way that the *compressibility* equation needs to be *imposed* in the remaining M nodes. In order for the velocity field not to be completely fixed by the incompressibility we require $M < 2N$. More precisely, if we assume Dirichlet boundary conditions and recall that p is defined up to an additive constant, we may replace the above inequality by $M - 1 < 2NI$ where NI is the number of interior velocity nodes. The above inequality limits the choice of the functions Ψ and Φ. Let us see this with an example. Consider *a rectangular domain with triangular elements and suppose to place the nodes for the velocity and pressure* fields on the corners and on the centers of the triangles respectively.The basis functions are piecewise linear for $\bar{u}$ and piecewise constants for p.



X Velocity Node
O Pressure Node

Fig. 8; A non-admissible triangulation for the incompressible Navier-Stokes equation

The count of the nodes is as follows:

- Total Number NT = NX*NY + M- 1

- Velocity NI = NX*NY - 2(NX + NY- 2)

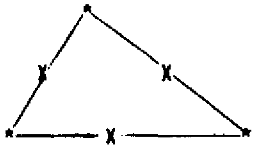
- Pressure M-1 = 2*(NX - 1)*(NY - 1)

The inequality reads now $NX + NY < 3$ which is manifestly never met. This problem can be overcome in several ways: it also exists a theoretical condition (Brezzi-Babuska) which fixes the requisites of allowable basis functions. In general the admissible functions are classified into two main families

- Taylor-Hood (the pressure is continuous)

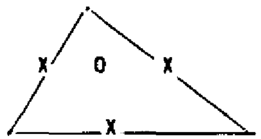
- Crouzeix-Raviart (the pressure is discontinuous)

Taylor-Hood (*u* quadratic , *p* linear)



```
X   Velocity Node
*   Pressure &Vel. Node
```

Crouzeix-Raviart (*u* linear-discontinuous , *p* constant)



```
X   V. Node (double)
0   P. Node
```

It is easy to verify that in both cases the count of the nodes fulfills the condition $M - 1 < 2NI$.

It is a common practice to decouple the incompressibility from nonlinearity by means of the so called *Operator Splitting Technique*.

To see qualitatively how this works let us rewrite the NS equation in abstract form as follows

$$\bar{u}_t = L\bar{u} + N(\bar{u})\bar{u} - grad\,p \tag{69}$$

where L and N denote the linear and non-linear part of the NS operator ($\nu\Delta - grad p$ and $\bar{u} \bullet grad\bar{u}$ respectively). The idea is to split the time step in two parts, say $I_{ab} \equiv (t_a, t_b)$ and $I_{bc} \equiv (t_b, t_c)$ and then discretize L implicitely in I_{ab} and explicitely in I_{bc} and viceversa for N. We obtain

$$\frac{\bar{u}_b - \bar{u}_a}{t_b - t_a} = L\bar{u}_b + N(\bar{u}_a)\bar{u}_b - grad p_b \tag{70}$$

and

$$\frac{\bar{u}_c - \bar{u}_b}{t_c - t_b} = L\bar{u}_b + N(\bar{u}_c)\bar{u}_b - grad p_b \tag{71}$$

We see that first equation is linear in $\bar{u}_b$ while the second is non-linear for u_c The incompressibility is imposed only to u_b in such a way that the non-linear problem is freed from the incompressibility constraint. After the space discretization, the two equations above yield an linear algebraic problem to march in time from t_a to t_b and a non-linear one from t_b to t_c. The matrices arising from this problem look as follows:

$$M_{ij}^{lm} = \int \Psi_i^l \Psi_j^m \, dxdy \tag{72}$$

$$L_{ij}^{lm} = \int \Psi_i^l L \Psi_j^m \, dxdy \tag{73}$$

$$N_{ij}^{lm} = \int \Psi_i^l N(\sum_{k,p} u_k^p \Psi_k^p) \Psi_j^m \, dxdy \tag{74}$$

where $l, m, p = 1, 3$, are the "internal" indices for each scalar field and $i, j, k = 1, N^2$ are the spatial indices. The block-structure of the problem is depicted here below:

```
--------------------   ----
|      |      |      |  |  |
| M11  |      |      |  |U1|
|      |      |      |  |  |
--------------------   |--|
|      |      |      |  |  |
|      | M22  |      |  |U2|
|      |      |      |  |  |
--------------------   |--|
|      |      |      |  |  |
|      |      | M33  |  |P |
|      |      |      |  |  |
--------------------   ----
```

for the mass matrix and

```
--------------------   ----
|      |      |      |  |  |
| F11  | F12  | F13  |  |U1|
|      |      |      |  |  |
--------------------   |--|
|      |      |      |  |  |
| F21  | F22  | F23  |  |U2|
|      |      |      |  |  |
--------------------   |--|
|      |      |      |  |  |
| F31  | F32  |  0   |  |P |
|      |      |      |  |  |
--------------------   ----
```

for the force matrix. Obviously, the matrices are again very sparse, in the sense that on each sub-block only nearest neighbors interaction contribute to the matrix elements.

The linear step is usually handled by an iterative solver. The non-linear problem is first linearized via either a Newton method or Least Squares method and subsequently solved by a linear scheme.

The solution of the NS equation in three-dimensional complicated geometrical conditions is feasible only on the most advanced present-day supercomputer generation.

FEM for Quantum Field Theory ?

It has been recently argued that the FE discretization of the operator field equations arising in Quantum Field Theory exhibits a number of appealing properties. The most crucial point is to show that FEM is consistent with the operator nature of the equations, that is to prove that Equal Time Commutation Relations (ETCR) are preserved in the course of time. To be specific, one expands an operator field $\Psi(x, t)$ as

$$\Psi = \sum \psi_i^n \varphi_i(x)\varphi^n(t) \tag{75}$$

where now the coefficients ψ_i^n are operators instead of numerical coefficients. The key point is to show that if the commutator $C_{ij}^n(\psi) \equiv \psi_i^n \psi_j^n - \psi_j^n \psi_i^n$ is a *c*-number at $t = 0$ it will stay such for any subsequent time.

For a free-massive fermion in a 2D Minkowsky lattice, Bender and coworkers have proved that a number of important properties are preserved in the course of the time-stepping. These are:

- ETCR
- Unitarity
- Chiral Symmetry (No fermion doubling)

It would be interesting to learn whether these properties still hold for more complicated theories such as Quantum Chromodynamics. In any event, there is probably a long way to go before one can really regard FEM as a realistic alternative to the Montecarlo method for the investigation of non-perturbative effects in Quantum Field Theory.

ACKNOWLEDGMENTS

The author would like to thank Professor Abdus Salam, the International Atomic Energy Agency and UNESCO for hospitality at the International Centre for Theoretical Physics, Trieste. He would also like to thank Dr. F. Piccolo and Dr. P. Carnevali for discussions and useful suggestions.

References

Part I

1. K. W. Morton, "A basic course in finite element methods" in Ref. 1, Part-II.
2. The references quoted in the Introduction

Part II

1. *Finite Elements in Physics*, proc. of the first European Graduate Summer Course on Computational Physics, Computer Physics Reports, 6(1987).
2. S. Succi, K. Appert, H. Hamnen, T. Hellsten and J. Vaclavik, Computer Physics Communications,40,137 (1986).
3. V. Girault and P. A. Raviart, *Finite Element Methods for the Navier-Stokes Equations*, Springer Verlag, Berlin 1986.
4. C. M. Bender et al, Phys. Rev. Letters, 50,1535 (1983) and 51, 1815 (1983).