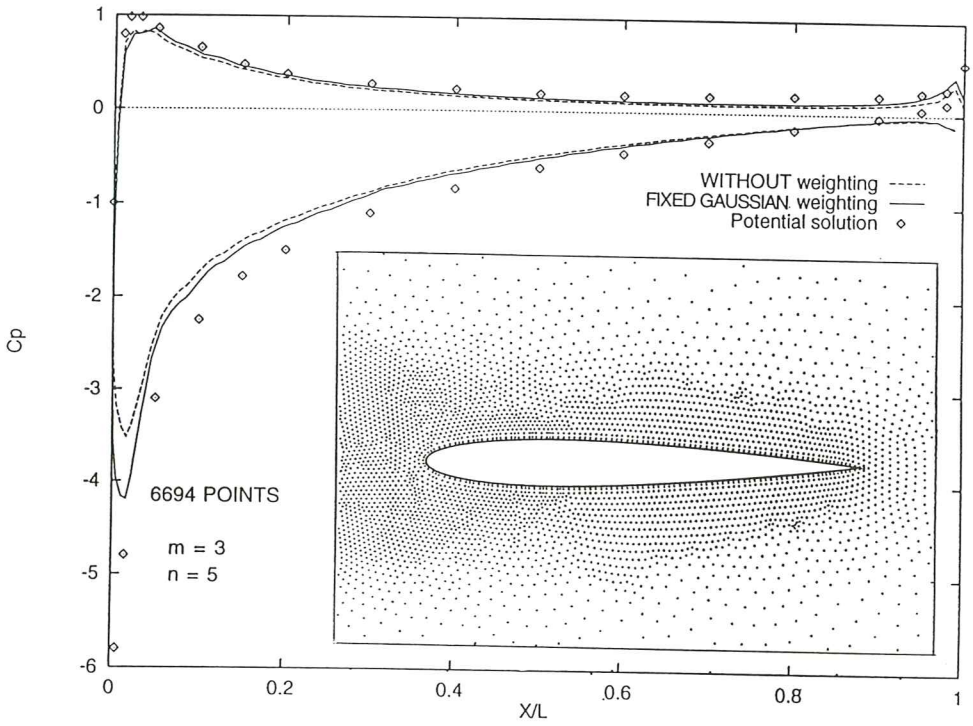
Finite Point Methods in Computational Mechanics

E. Oñate

S. Idelsohn

O.C. Zienkiewicz



Publication CIMNE Nº 67, July 1995

Finite Point Methods in Computational Mechanics

E. Oñate
S. Idelsohn
O.C. Zienkiewicz

Publication CIMNE Nº 67, July 1995

International Center for Numerical Methods in Engineering
Gran Capitán s/n, 08034 Barcelona, Spain

FINITE POINT METHODS IN COMPUTATIONAL MECHANICS

E. Oñate
S. Idelsohn
O.C. Zienkiewicz

International Center for Numerical Methods in Engineering
Universitat Politècnica de Catalunya
Edificio C1, Gran Capitán s/n,
08034, Barcelona, Spain

1. INTRODUCTION

The Finite Element Method (FEM) [1] and its subclass the Finite Volume Method (FVM) [2-4] are well established numerical techniques whose main advantage is their ability to deal with complicated domains in a simple manner while keeping a local character in the approximation. Both methods seek to divide the total domain volume into a finite number of subdomains on which a volume integration is performed. The subdomains are constrained by some geometrical regularity conditions such as having a positive volume or a limit aspect ratio between the element dimensions and angles, etc.. Although this poses no serious difficulties for 2D situations, the lack of efficient 3D mesh generators makes the solution of 3D problems a difficult task.

It is widely acknowledged that 3D mesh generation remains one of the big challenges in both FE and FV computations. Thus, given enough computer power even the most complex problems in computational mechanics, such as the 3D solution of Navier-Stokes equations in fluid flow can be tackled accurately nowadays providing an acceptable mesh is available. The generation of 3D meshes, however despite major recent advances in this field, is certainly the bottle neck in most industrial CFD computations and, in many cases, it can absorb far more time and cost than the numerical solution itself.

Considerable effort has been devoted during last decades to the development of so called mesh-free methods. The first attempts were reported by some finite difference (FD) practitioners deriving FD schemes in arbitrary irregular grids [5-12]. Here typically the concept of "star" of nodes was introduced to derive FD approximations for each central node by means of local Taylor series expansions using the information provided by the number and position of nodes contained in each star [12].

A similar class of methods named Smooth Particle Hydrodynamics (SPH), sometimes called the Free Lagrange method, depend only on a set of disordered point or particles and have enjoied considerable popularity in computational physics [13-17] and astrophysics to model the motion and collision of stars. These methods work well in the absence of boundaries, although they are not as accurate

as the regular finite element methods [18].

Different authors have recently investigated the possibility of deriving numerical methods where meshes are unnecessary. Nayroles *et al.* [19] proposed a technique which they call the Diffuse Element (DE) method, where only a mesh of nodes and a boundary description is needed to formulate the Galerkin equations. The interpolating fuctions are polynomials fitted to the nodal values by a weighted least squares approximation. Although no finite element mesh is explicitely required in this method, still some kind of "auxiliary grid" was used in [19] in order to compute numericaly the integral expressions derived from the Galerkin approach, thus eliminating many of the advantages of the original mesh-free phylosophy.

Belytschko *et al.* [20,21] have proposed an extension of the DE approach which they call the element-free Galerkin (EFG) method. Here, generalized moving least-squares (MLS) interpolants, typically exploited in curve and surface fitting, are used to define the local approximation. This provides additional terms in the derivatives of the interpolant omitted by Nayroles *et al.* [20]. In addition, a regular cell structure is chosen as the "auxiliary grid" to compute the integrals by means of high order quadratures. Duarte and Oden [22] and Babustka and Melenk [23] have recently formalized the MLS interpolants as a subclass of the so called "partition of unity" (PU) functions and they propose meshless procedures similar to the EFG approach using a number of PU interpolations.

MLS interpolants have been recently used by Sulsky *et al.* [24–26] to develop the so called particle-in-cell method. In this approach a collection of material points are followed throughout the deformation of a solid and provide a Lagrangian description that is not subject to mesh tangling. MLS interpolation is used to transfer information (velocities) from the moving points to the nodes of an auxiliary grid where the equations of motion are solved by the standard Galerkin method. The method can thus be interpreted as a generalization of the EFG approach to transient dynamic problems in solid mechanics. A particular interesting feature of this approach is that no special contact algorithm is needed to simulate impact and penetration problems.

Liu *et al.* [18,27,28,29] have developed a different class of "gridless" multiple scale methods based on reproducing kernel and wavelet analysis. This technique termed Reproducing Kernel Particle (RKP) method allows to develop a new type of shape function through an integrals windows transform. The window function can be *translated* and *dilated* around the domain thus replacing the need to define elements and providing refinement. An interesting comparative study of RKP, SPH, DE and EFG methods can be found in [30].

This paper presents an overview of some point data interpolation based procedures (termed here "finite point methods") and their possibilities in computational mechanics. Particular emphasis is put in distinguishing the specific merits and drawbacks of the different procedures such as DE, EFG, RKP and SPH methods. The paper analyses some solutions for adjoint and non-selfadjoint equations using finite point methods and shows results for a compressible fluid mechanics problem.

2. BASIC CONCEPTS ON MESH FREE TECHNIQUES

Let us assume a scalar problem governed by a differential equation

$$A(u) = b \quad \text{in} \quad \Omega \tag{1}$$

with boundary conditions

$$\begin{aligned} B(u) &= t \quad \text{in} \quad \Gamma_t \\ u - u_p &= 0 \quad \text{in} \quad \Gamma_u \end{aligned} \tag{2}$$

to be satisfied in a domain Ω with boundary $\Gamma = \Gamma_t \cup \Gamma_u$. In above A and B are appropriate differential operators, u is the problem unknown and b and t represent external forces or sources acting over the domain Ω and along the boundary Γ_t, respectively. Finally u_p is the prescribed value of u over the boundary Γ_u.

The most general procedure of solving numerically the above system of differential equations is the weighted residual method in which the unknown function u is approximated by some trial value $\hat{u}$ and eqs.(1) and (2) are replaced by

$$\int_\Omega W_i[A\hat{u} - b]\, d\Omega + \int_{\Gamma_t} \bar{W}_i[B\hat{u} - t]\, d\Gamma + \int_{\Gamma_u} \bar{\bar{W}}_i[\hat{u} - u_p] d\Gamma = 0 \tag{3}$$

with the weighting functions W_i, $\bar{W}_i$ and $\bar{\bar{W}}_i$ defined in different ways FE, FV and FD methods can be considered as particular cases of (3).

In order to keep a local character of the problem (leading to a banded matrix), function u must be approximated by a combination of locally defined functions as

$$u(x) \cong \hat{u}(x) = \sum_{i=1}^{n_p} N_i(x) u_i^h = \mathbf{N}^T(x)\mathbf{u}^h \tag{4}$$

with n_p being the total number of points in the domain and the interpolation functions $N_i(x)$ satisfy

$$\begin{aligned} N_i(x) &\neq 0 \quad \text{if} \quad x \in \Omega_i \\ N_i(x) &= 0 \quad \text{if} \quad x \notin \Omega_i \end{aligned} \tag{5}$$

where Ω_i is a subdomain of Ω containing n points, $n \ll n_p$. In (4) u_i^h is the *approximate value* of u at point i such that $u(x_i) \simeq u_i^h$.

In FE and FV methods the Ω_i subdomains, also termed "interpolating domains" or "clouds" in the mesh-free literature [22], are divided into elements and the N_i function may have some discontinuities (in the function itself or in its derivatives) in the element interfaces. In the FE method the weighting functions N_i are defined in "weigthing domains" which precisely coincide with the interpolating domains Ω_i. In cell vertex FV the interpolation and integration domains also coincide, however in the cell centered case they are different (see Figure 1).

A common feature of FE and FV methods is that they both require a mesh for interpolation purposes and also to compute the integrals in eq.(3).

On the basis of above remarks, a mesh-free numerical procedure should satisfy the following three conditions:

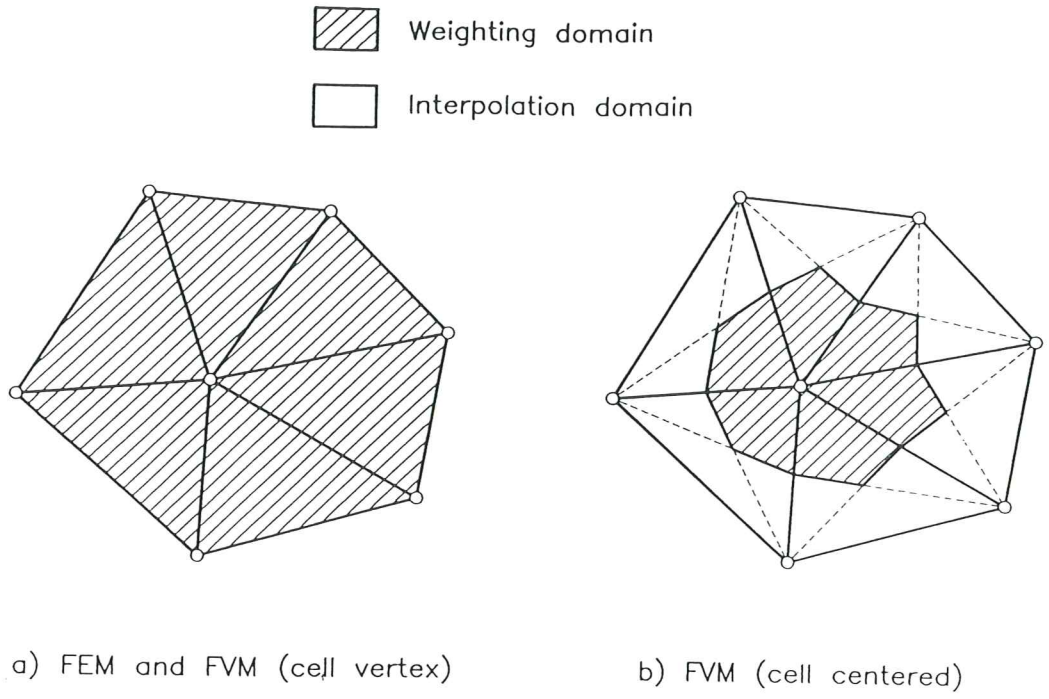


Figure 1. Interpolation and weighting domains in finite element (FE) and finite volume (FV) methods

I. The discretization of the unknown function and its derivatives must be defined *only* by the position of points located within the analysis domain.
II. a) No volume or surface integration is needed, or
b) Any volume or surface integration should be independent of the interpolation procedure chosen.
III. The weighting function and its derivatives must be defined *only* by the position of points located within the analysis domain.

Condition IIa is satisfied by finite difference and local collocation procedures. Other methods like domain collocation, or those based on the use of auxiliary background grids for integration purposes (i.e. EFG [20], RPK [18]) satisfy condition IIb.

Condition I is satisfied by the Rayleigh-Ritz method in which the N_i functions are defined over the whole domain Ω, thus precluding the local character of the aproximation. FE methods define the shape functions N_i over non overlapping region (elements) which assembly constitues the domain Ω_i [2]. Different interpolations are therefore possible for a given number of points simply by changing the orientation or the form of these regions, thus violating the necessary requirement of condition I. Although FV techniques do not explicitely define an interpolation of the form (4), it is well known that they are equivalent to using linear shape functions over domains Ω_i defined in the same manner as in the FE method [2-4].

In next section some of the more popular approximations used to build interpolations based on a finite number of points (hereafter termed generically "finite point method") are briefly reviewed.

3. LEAST SQUARE, DIFFUSE LEAST SQUARE, MOVING LEAST SQUARE AND REPRODUCING PARTICLE KERNEL APPROXIMATIONS

Let Ω_i be the interpolating domain (cloud) of a function $u(x)$ and let s_i with $i = 1, 2, \cdots, n$ be a set of n points with coordinates $x_i \in \Omega_i$. The unknown function u may be approximated in Ω_i by

$$u(x) \cong \hat{u}(x) = \sum_{i=1}^{m} p_i(x)\alpha_i = \mathbf{p}(x)^T \boldsymbol{\alpha} \tag{6}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots \alpha_n]^T$ and vector $\mathbf{p}(x)$ contains the monomials, hereafter termed "base interpolating functions", in the space coordinates $\mathbf{x}$ ensuring that the basis is complete. For a 2D problem

$$\mathbf{p} = [1, x, y]^T \quad \text{for} \quad m = 3 \tag{7a}$$

and

$$\mathbf{p} = [1, x, y, x^2, xy, y^2]^T \quad \text{for} \quad m = 6 \quad , \quad \text{etc} \tag{7b}$$

Function $u(x)$ can now be sampled at the n points belonging to Ω_i

$$\mathbf{u}^h = \begin{Bmatrix} u_1^h \\ u_2^h \\ \vdots \\ u_n^h \end{Bmatrix} \cong \begin{Bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_n \end{Bmatrix} = \begin{Bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \vdots \\ \mathbf{p}_n^T \end{Bmatrix} \boldsymbol{\alpha} = \mathbf{C}\boldsymbol{\alpha} \tag{8}$$

where $u_i^h = u(x_i)$ are the unknown but sought for values of function u at point i, $\hat{u}_i = \hat{u}(x_i)$ are the approximate values, and $\mathbf{p}_i = \mathbf{p}(x_i)$.

In the FE approximation the number of points is chosen so that $m = n$. In this case $\mathbf{C}$ is a square matrix and we can obtain after *equaling* u^h *with* $\mathbf{C}\boldsymbol{\alpha}$ in (8)

$$\boldsymbol{\alpha} = \mathbf{C}^{-1}\mathbf{u}^h \tag{9}$$

and

$$u \cong \hat{u} = \mathbf{p}^T\mathbf{C}^{-1}\mathbf{u}^h = \mathbf{N}^T\mathbf{u}^h \tag{10}$$

with $\mathbf{N}^T = [N_1, \cdots, N_n] = \mathbf{p}^T\mathbf{C}^{-1}$ and $N_i = \sum_{j=1}^{m} p_j(x)C_{ji}^{-1}$ (11)

The shape functions $N_i(x)$ satisfy the standard condition [1]

$$\begin{aligned} N_i(x_j) &= 1 \quad i = j \\ &= 0 \quad i \neq j \qquad i, j = 1, \cdots, m \end{aligned} \tag{12}$$

3.1 Least Square (LSQ) Approximation

If $n > m$, $\mathbf{C}$ is not longer a square matrix and the approximation can not fit all the u_i^h values. This problem can be simply overcome by approximating the u values minimizing the square distance

$$J = \sum_{j=1}^{n} (u_j^h - \hat{u}(x_j))^2 = \sum_{j=1}^{n} (u_j^h - \mathbf{p}_j^T\boldsymbol{\alpha})^2 \tag{13}$$

with respect to the α_i parameters (least square fit).

Standard minimization gives

$$\boldsymbol{\alpha} = \bar{\mathbf{C}}^{-1}\mathbf{u}^h \quad \text{with} \quad \bar{\mathbf{C}}^{-1} = \mathbf{A}^{-1}\mathbf{B} \tag{14}$$

where

$$\mathbf{A} = \sum_{j=1}^{n} \mathbf{p}(x_j)\mathbf{p}^T(x_j) \tag{15a}$$

$$\mathbf{B} = [\mathbf{p}(x_1), \mathbf{p}(x_2), \cdots \mathbf{p}(x_n)] \tag{15b}$$

The final approximation is still given precisely by eq.(10) now however substituting matrix $\mathbf{C}$ by $\bar{\mathbf{C}}$ of eq.(14). The new shape functions are therefore

$$N_i = \sum_{j=1}^{m} p_j(x)\bar{C}^{-1}_{ji} \tag{16}$$

It must be noted that accordingly to (13)

$$u_j = u^h(x_j) \neq \hat{u}(x_j) \tag{17}$$

i.e. the local values of the approximating function do not fit the nodal unknown values (Figure 2).

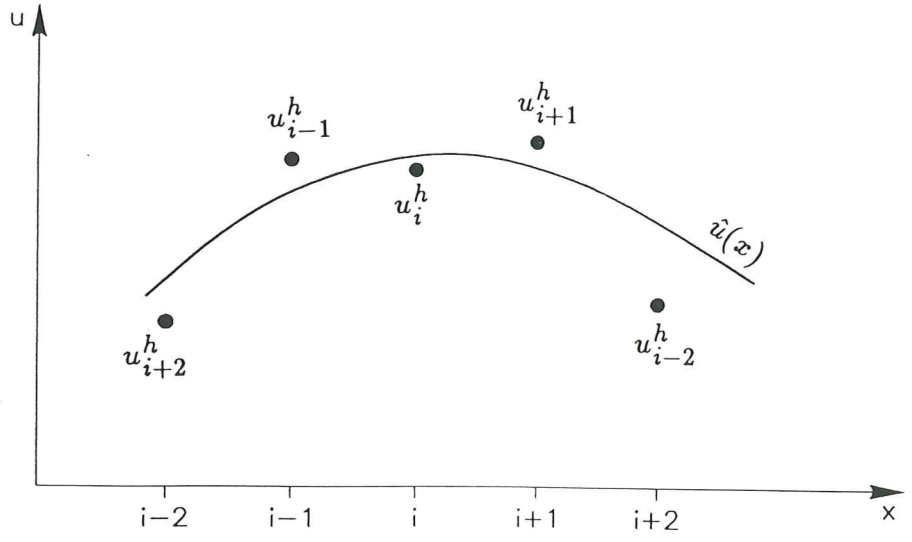


Figure 2. Unknown function $\hat{u}(x)$ and unknown parameters u^h_i

Also, note that when $n = m$ the FEM approximation is recovered and then $\hat{u}(x_j) = u^h(x_j)$.

LSQ approximation has enjoied some popularity in deriving point data interpolations for numerical computation in solid and fluid mechanics. Nay and Utku [31] used quadratic LSQ interpolations to fit the deflection field giving

constant curvatures for thin plate bending analysis. More recently, Batina [32] has used linear LSQ fits to approximate the fluxes in the solution of high speed compressible flows.

The main drawback of the LSQ approach is that the approximation rapidly deteriorates if the number of points used, n, largely exceeds that of the m polynomial terms in $\mathbf{p}$. Some examples of this kind are given in a next section. This deficiency can be overcome by using a weighted least square interpolation as described next.

3.2 Diffuse Least Square (DLS) Approximation

The LSQ approximation can be enhanced in a region where, for instance, the derivatives of the unknown are evaluated, by weighting the squared distances with a function $\varphi(x)$ so that we minimize

$$J = \sum_{j=1}^{n} \varphi(x_j)(u_j^h - \hat{u}(x_j))^2 = \sum_{j=1}^{n} \varphi(x_j)(u_j^h - \mathbf{p}_j^T \alpha)^2 \tag{18}$$

Function $\varphi(x)$ is usually built in such a way that it takes a unit value in the vicinity of the node where the function (or its derivatives) are to be computed and vanishes outside a region Ω_φ surrounding the node (Figure 3). The region Ω_φ defines the number of sampling points n in the interpolating region. A typical choice for $\varphi(x)$ is the normalized Gaussian function. Of course $n \geq m$ is always required and if equality occurs no effect of weighting is present.

This approach was used in [19] under the name of "diffuse" approximation. Standard minimization of eq.(18) with respect to α_i gives the same form of the shape functions of (16) with matrices $\mathbf{A}$ and $\mathbf{B}$ given now by

$$\mathbf{A} = \sum_{j=1}^{n} \varphi(x_j)\mathbf{p}(x_j)\mathbf{p}^T(x_j) \tag{19a}$$

$$\mathbf{B} = [\varphi(x_1)\mathbf{p}(x_1), \varphi(x_2)\mathbf{p}(x_2), \cdots, \varphi(x_n)\mathbf{p}(x_n)] \tag{19b}$$

3.3 Moving Least Square (MLS) Approximation

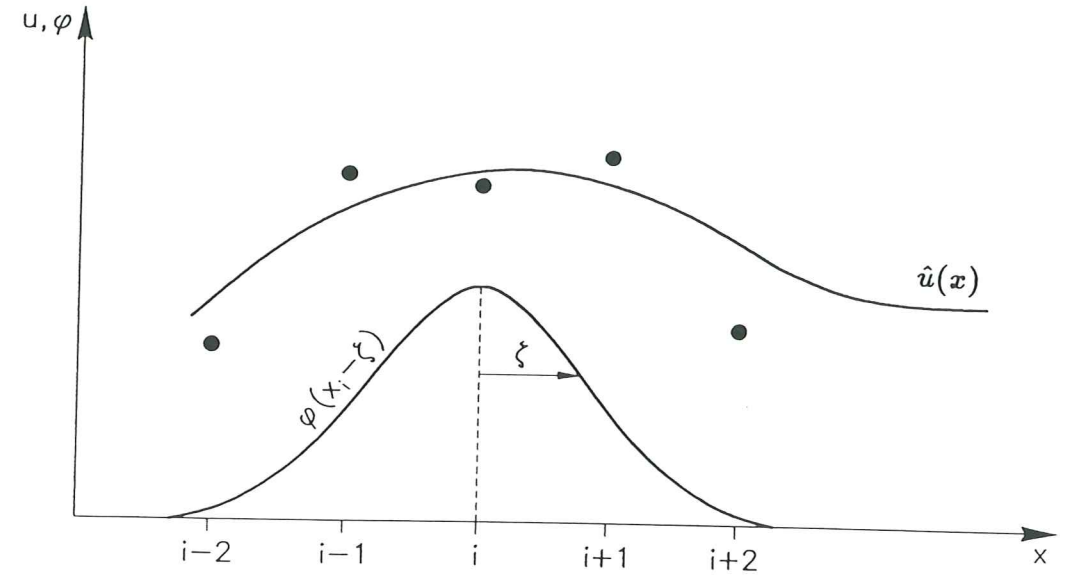
In the moving least square (MLS) approach function $\varphi(x)$ is translated over the domain so that it takes the maximum value over the point ,x, where the unknown function u is evaluated. From Figure 3 it is deduced that $\varphi(x - x_i)$ and matrices $\mathbf{A}$ and $\mathbf{B}$ of eq.(19) are given now by

$$\mathbf{A} = \sum_{j=1}^{n} \varphi(x - x_j)\mathbf{p}(x_j)\mathbf{p}^T(x_j) \tag{20a}$$

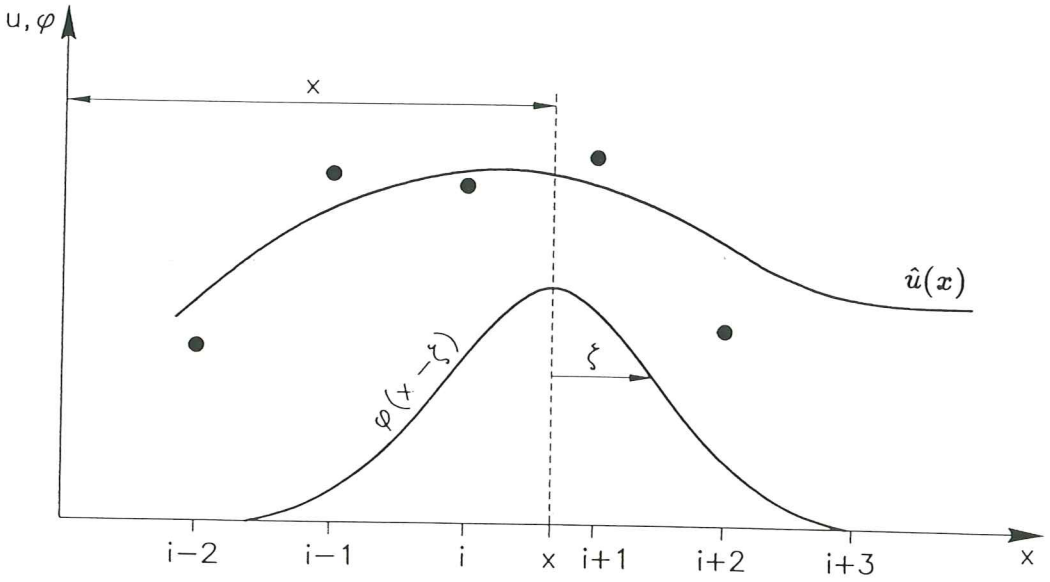
$$\mathbf{B} = [\varphi(x - x_1)\mathbf{p}(x_1), \varphi(x - x_2)\mathbf{p}(x_2), \cdots, \varphi(x - x_n)\mathbf{p}(x_n)] \tag{20b}$$

Note that in this case the parameters α_i are not larger constants and that inversion of matrices is required at every point where $\hat{u}$ is to be evaluated.

This method has been successfully used by Belytschko *et al.* [20] in the context of the so called Element Free Galerkin (EFG) method. In order to avoid



a) Fixed φ corresponding to node x_i



b) Moving φ

Figure 3. Diffuse and moving weighting functions

the need of inverting matrix **A** for each quadrature point where the discrete equations are assembled, Lu *et al.* [21] proposed to use a weighted orthogonal basis functions using a Schmidt orthogonalization technique. A generalization of the MLS approach using the concept of partition of unity functions can be found in [22].

3.4 Reproducing Kernel Particle (RPK) Methods

An alternative procedure to derive point data interpolation of the form (6) is to use multiple scale methods based on reproducing kernel and wavelet analysis. This approach has been successfully exploted by Liu *et al.* [18,27,28,29] under the name of reproducing kernel particle (RKP) methods for the solution of some solid and fluid flow problems.

A reproducing kernel is a class of operators that reproduces the function itself by integrating through the domain. The Fourier Transform is a typical example of a reproducing kernel. In general form we can write

$$f(x) = \int_{\Omega} \phi(x-y) f(y)\, dy \tag{21}$$

where $\phi(x)$ is an appropriate window functions defined with a compact support so that it can be *translated* around the domain. A *dilation* parameter is also used to provide *refinement.* The parameters α_i in (6) can be obtained now as follows. The variable $\mathbf{x}$ is changed to $\mathbf{s}$ in eq.(6) and then both sides are pre-multiplied by $\mathbf{p}(s)$. The integral window transform is then applied as

$$\mathbf{p}\hat{u} = \int_{\Omega} \mathbf{p}(s)\hat{u}(s)\phi(s)\, ds = \Big(\int_{\Omega} \mathbf{p}(s)\mathbf{p}(s)^T \phi(s)\, ds \Big) \boldsymbol{\alpha} = \hat{\mathbf{C}}(x)\boldsymbol{\alpha} \tag{22}$$

where Ω is the analysis domain.

Substituting the values of $\boldsymbol{\alpha}$ obtained from (22) into (6) and applying again the integral window transform gives finally

$$\hat{u}(x) = \int_{\Omega} \mathbf{p}^T(x)\hat{\mathbf{C}}^{-1}(\mathbf{x})\mathbf{p}(s)\phi(s)\hat{u}(s)\, ds = \int_{\Omega} E(x,s)\phi(s)\hat{u}(s)\, ds \tag{23}$$

Eq.(23) can be now discretized to provide a form identical to eq.(4) with

$$N_j = E(x, x_j)\phi(x_j) \tag{24}$$

Comparing the reconstruction equation (23) to the Smooth Particle Hydrodynamic (SPH) method [16] reveals that the only difference is the appearence of the term $E(x,s) = \mathbf{p}^T(x)\mathbf{C}^{-1}(\mathbf{x})\mathbf{p}(s)$ which is equal to unity in SPH procedures and these can be considered a particular case of the RPK methodology. This term, defined as a correction function, provides an accurately solution near the boundary and plays no role in the interior domain. A comparison of RPK, DLS, MLS and SPH methods can be found in [18,30].

3.5 Comparison of FE, LSQ, DLS, MLS and RPK Approximations

Independently of the method chosen to solve the global problem (eq.(3)), the accuracy of the solution will very much depend on the shape functions used in each approximation. One of the main differences between FE and LSQ, DLS, MLS and

RPK methods is that in the FE approach the local value of the approximating function is coincident with the unknown parameter, i.e.

$$\hat{u}(x_i) = u_i^h \tag{25}$$

whereas in finite point based methods (FPM) such as LSQ, DLS, MLS and RPK methods

$$\hat{u}(x_i) \neq u_i^h \tag{26}$$

Summarizing the "degrees of freedom" for choosing the approximation in the different methods considered we can list

a) In FE methods

- the mesh
- the polynomial order (m in eq.(6))

b) In Finite Point Methods (FPM)

- the node position (in LSQ, DLS, MLS and RPK methods)
- the polynomial orden (in LSQ, DLS, MLS and RPK methods)
- the shape of the weighting least square function $\varphi(x)$ (in DLS and MLS methods)
- the size of $\varphi(x)$, i.e. more or less points included in Ω_φ (in DLS and MLS methods)
- the choice of fixed or moving least square weighting functions $\varphi(x)$ (in DLS and MLS methods)
- the translation and dilation parameters of the window function $\phi(x)$ (RPK method)

In order to better understand these differences let us plot some of the 1D shape functions for these methods.

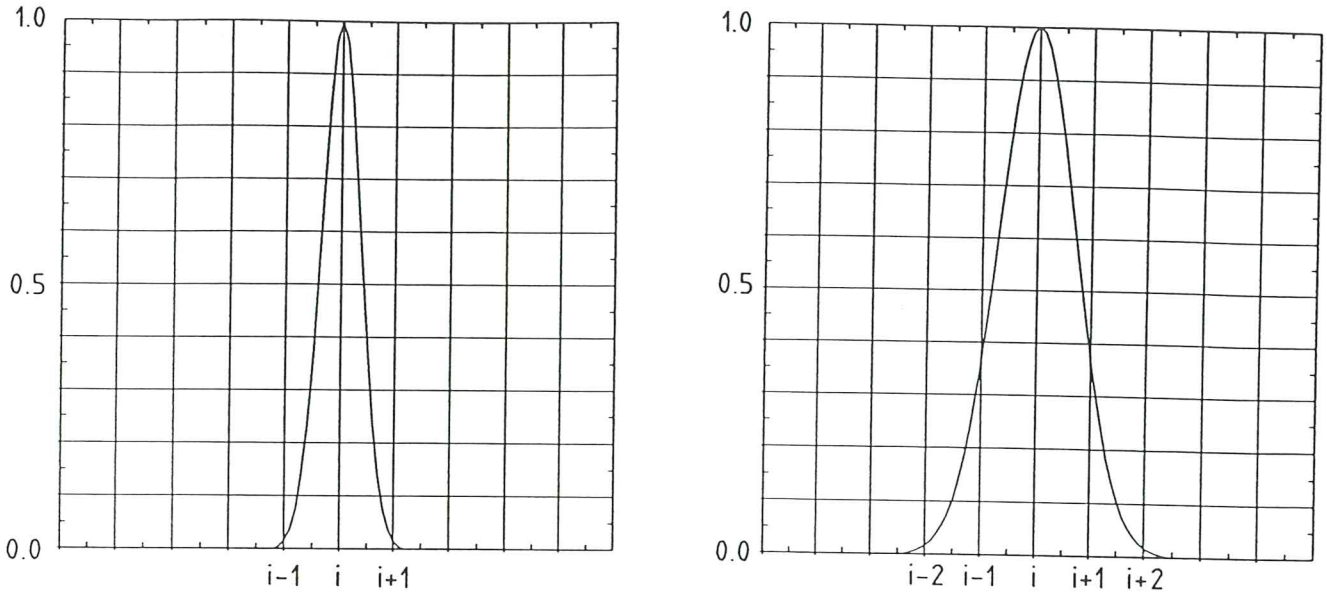
Figure 4 shows the weighting least square function $\phi(x)$ given by the gaussian expression

$$\varphi(x) = \frac{e^{-(\frac{x}{c})^2} - e^{-(\frac{x_m}{c})^2}}{1 - e^{(\frac{x_m}{c})^2}} \tag{27}$$

where x_m is the size of the support and c is a parameter determining the shape. We will choose $c = x_{m/2}$.

DLS and MLS methods will use the same ϕ function with $x_m = 1.2h$ for a three point cloud ($n = 3$) and $x_m = 2.5h$ for a cloud with five points ($n = 5$). All points are taken equally spaced and h is the distance between 2 adjacent points.

Figure 5 shows the shape functions $N_i(x)$ for a quadratic base function polynomial ($m = 3$) and clouds of three and five points, respectively. Note that with 3 points clouds LSQ and DLS methods yield interpolating functions identical to the standard FE quadratic shape functions as expected. The MLS method gives interpolating functions of higher order than quadratic. When $n = 5$ (five points cloud) none of the methods gives $N_i(x) = 1$ at $x = x_i$ neither $N_i(x) = 0$ at



a) $n = 3$ b) $n = 5$

Figure 4. Gaussian weighting functions for 3 and 5 points clouds

$x = x_j$, $j \neq i$. In particular, the LSQ method yields very unacurate interpolating functions in this case with the value $N_i(x_i)$ equal to less than 0.5 everywhere.

Similar results can be obtained with linear base functions ($m = 2$) as shown in Figure 6. Here again, the shape functions provided by the LSQ method deteriorate when the number of points within the cloud n is changed from 3 to 5. This deterioration is also important in DLS and MLS methods where the $N_i(x_i)$ value shoots down from 0.97 to 0.57.

It can be concluded that the selection of the more adequate approximation for point data interpolation should be based on its *insensivity* to the number of points chosen within the approximating region (cloud). In order to preserve the freedom of adding, moving or removing points for a given order of interpolation, the approximating functions should be as insensitive as possible to the number of points within the cloud. We have found that the LSQ method is very sensitive to the number of cloud points chosen and the approximation rapidly deteriorates as the number of points increases. The MLS and DLS approximations with linear base polynomials seem to be also quite sensitive to the number of cloud points. Conversely the MLS and DLS methods with quadratic base functions seem to be best suited for point data interpolation using blendings of approximations based on 3 and 5 points for each cloud (see Table 1).

Another important property of a good Finite Point Method (FPM) is the possibility to introduce (or to translate) new points independently of the distance existing between existing points. In the FE method, two nodes that are very close together, generate a gradient in the shape function wich introduces large numerical error. This adds severe limitations in the mesh generator and adaptivity criteria. In a good FPM, a close distance between two points should not affect the numerical results.

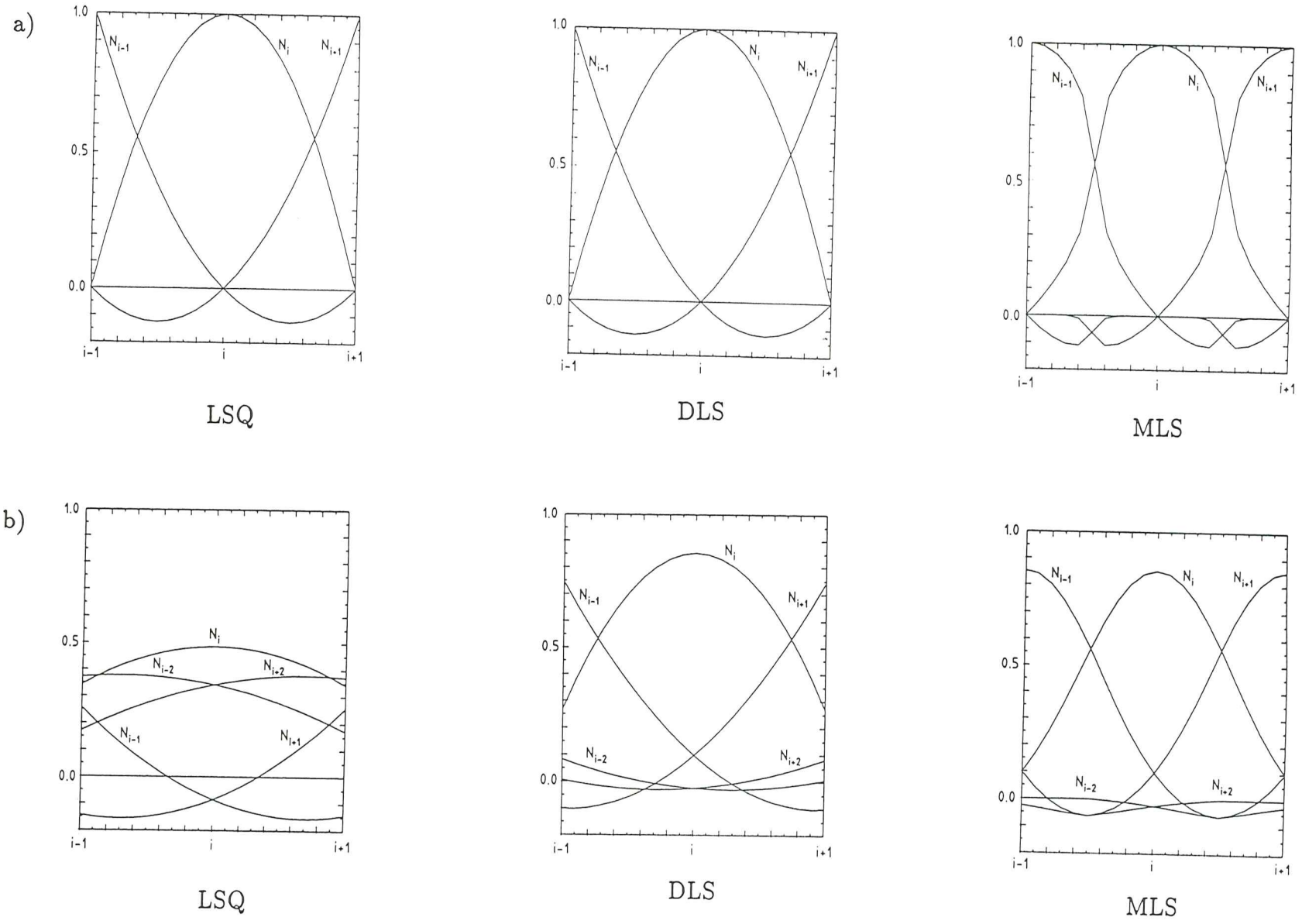


Figure 5. Shape functions $N_i(x)$ for $\mathbf{p} = [1, x, x^2]$. a) Three points clouds ($n = 3$); b) Five points clouds ($n = 5$)

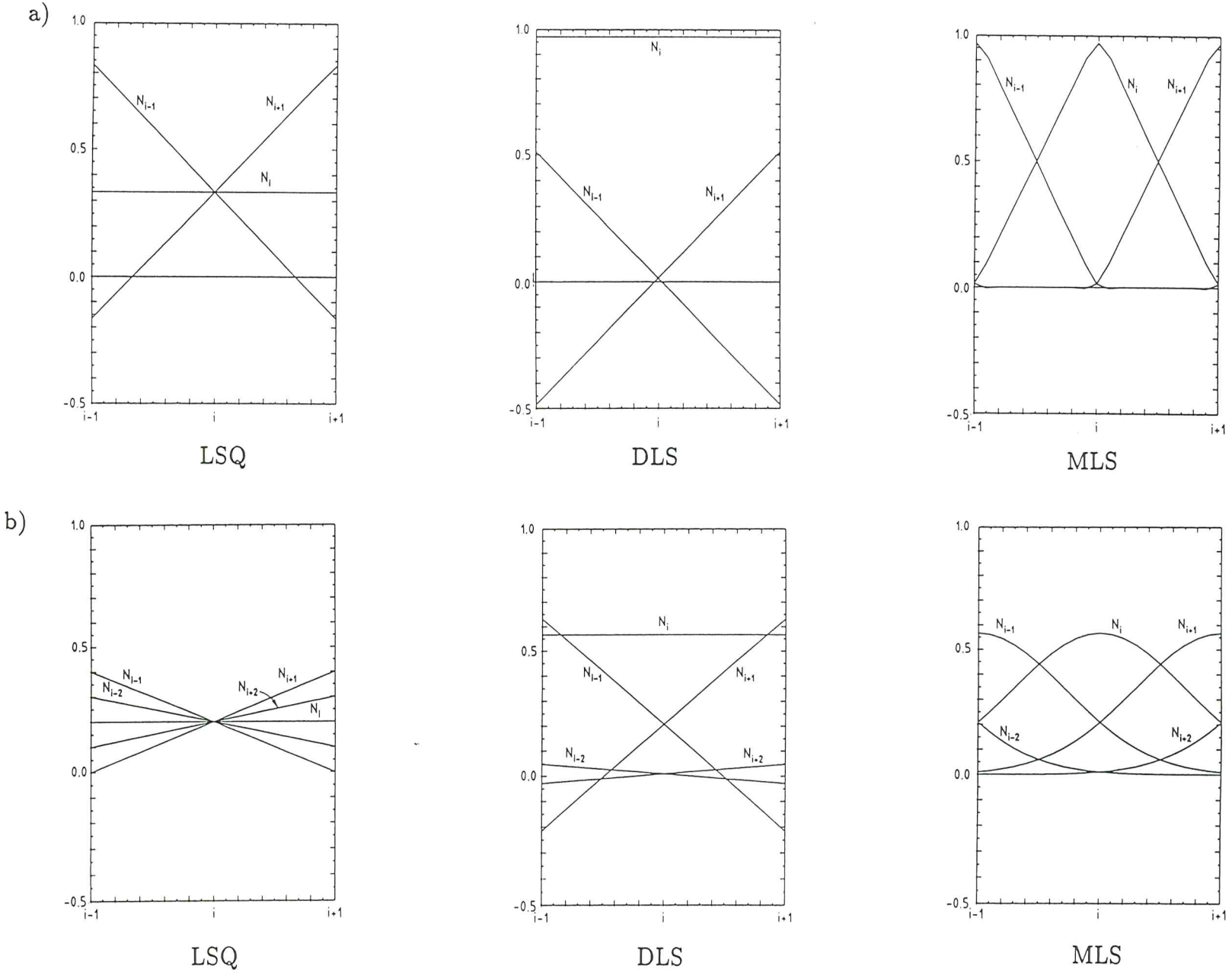


Figure 6. Shape functions $N_i(x)$ for $\mathbf{p} = [1, x]$. a) Three points clouds ($n = 3$); b) Five points clouds ($n = 5$)

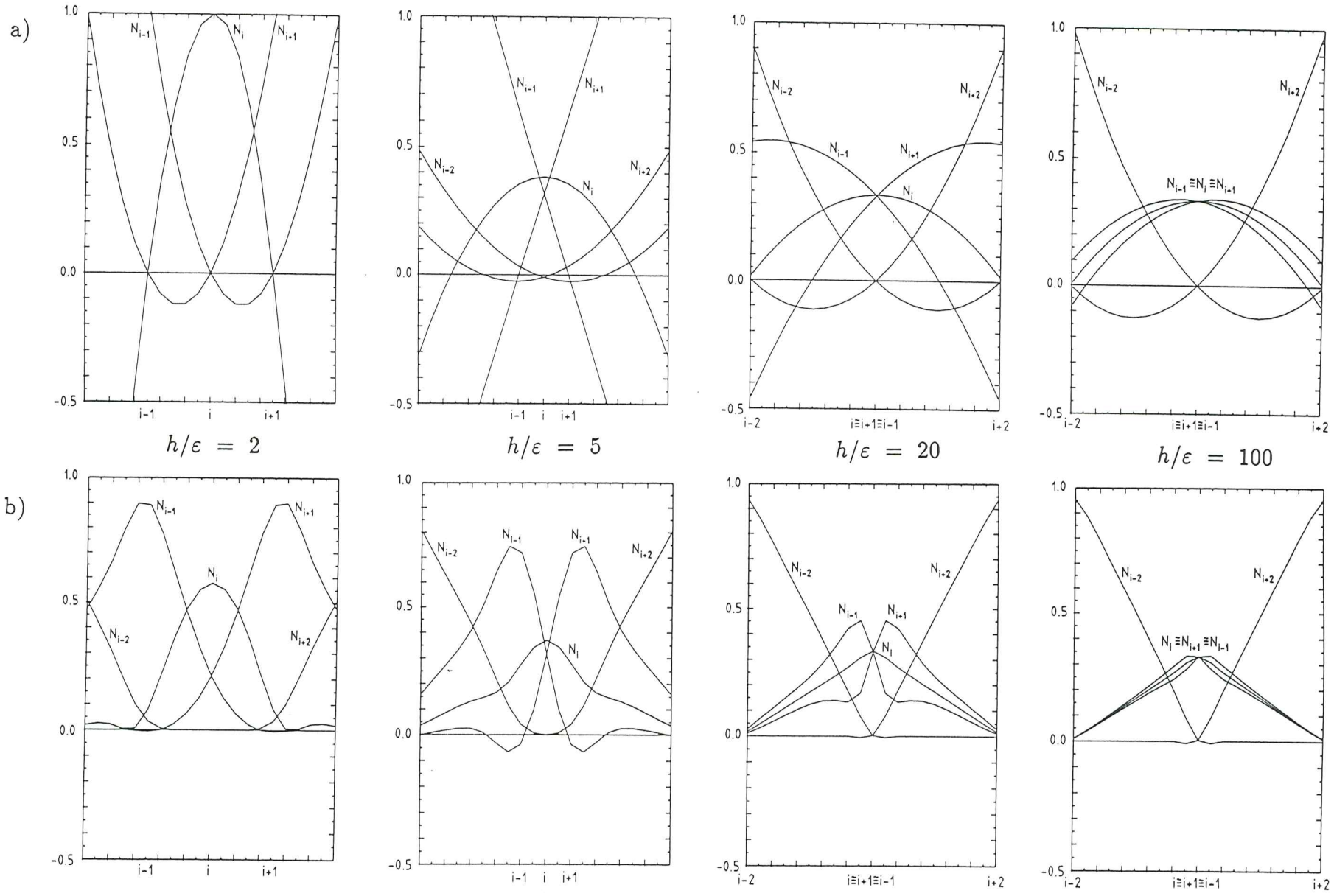


Figure 8. Shape functions for 3 very close points. a) DLS method, ($m = 3$); b) MLS method, ($m = 2$)

4.1 Point Collocation

The simplest choice that satisfies the mesh-free condition is making $W_i = \bar{W}_i = \bar{\bar{W}}_i = \delta_i$ where δ_i is the Dirac delta. This gives the set of equations

$$[A(\hat{u})]_i - b_i = 0 \quad \text{in } \Omega \tag{28}$$

$$[B(\hat{u})]_i - t_i = 0 \quad \text{in } \Gamma_t \tag{29}$$

$$\hat{u}_i - u_p = 0 \quad \text{in } \Gamma_u \tag{30}$$

Any of the previous shape functions may be used to approximate $\hat{u}$ leading in all cases to the system of equations

$$\mathbf{K}\mathbf{u}^h = \mathbf{f} \tag{31}$$

with $K_{ij} = [A(N_j)]_i + B(N_j)]_i$ and where the symmetry of the stiffness matrix $\mathbf{K}$ is not necessarily preserved. Vector $\mathbf{u}^h$ contains the problem unknowns, u_i^h, and $\mathbf{f}$ is a vector containing the contributions from the force terms b and t and the prescribed values u_p.

Taking a particular set of nodes and shape functions, this method is coincident with the standard Finite Difference Method. However the approach proposed here is more general and any of the interpolation techniques described in Section 3 can be used.

4.2 Galerkin Weighting

For self-adjoint problems, the Galerkin method is the most popular weighting procedure because it has optimal convergence properties and when correctly applied leads to a symmetric system of equations. Using any of the shape functions described in Section 3, the method remains gridless, as these functions were defined on a gridless way. Nevertheless, special care is to be taken if the symmetry of the system must be preserved.

For instance, in LSM and DLS methods, there is not a unique definition of the unknown field at each point. Let us take, for instance, two points of coordinates x_p and x_q. The definition of $\hat{u}$ in the vicinity of x_p and x_q will be

$$u(x) \cong \hat{u}(x) = \sum_{i=1}^{N} N_i^p u_i^h \qquad \text{for } x \text{ near } p \tag{32}$$

$$u(x) \cong \hat{u}(x) = \sum_{i=1}^{N} N_i^q u_i^h \qquad \text{for } x \text{ near } q \tag{33}$$

where superindexes p and q have been used to distinguish the interpolations around p and q respectively. In FE, MLS and RPK approaches, the definition of the unknown is unique and them $N_i^p = N_i^q$ for $i = 1, \cdots, n_p$.

The standard weighted equations around points x_p and x_q are written as

$$\int_\Omega W_p \left[A\left(\sum_{i=1}^{n_p} N_i^p u_i^h \right) \right] d\Omega + b.t. = 0 \tag{34}$$

$$\int_\Omega W_q \left[A\left(\sum_{i=1}^{n_p} N_i^q u_i^h \right) \right] d\Omega + b.t. = 0 \tag{35}$$

and the Galerkin method stands

$$W_p = N_p^p \quad ; \quad W_q = N_q^q \tag{36}$$

giving

$$K_{ij} = \int_\Omega N_i^i [A(N_j^i)] d\Omega \tag{37}$$

which, after integration by parts, yields a symmetric matrix for self-adjoint operators $A(\hat{u})$ and providing that $N_p^q = N_q^p$. This is not the case in LSQ and DLS procedures and hence $\mathbf{K}$ is invariably non-symmetric in these situations.

The integration by parts may also introduce a non-symmetric term if the weighting functions are not zero at the boundary. This is the case of most gridless approaches in which equation (25) is not satisfied and some alternatives will be discussed in a next section.

4.3 Constant Weighting Functions

Choising a subdomain collocation procedure with

$$\begin{aligned} W_i &= 1 \quad x \in \Omega_i \\ W_i &= 0 \quad x \notin \Omega_i \end{aligned} \tag{38}$$

leads to a set of equations similar to those based on Finite Volume methods [2-4].

After integration by parts of equation (3) all the volume terms connected with the derivatives of W_i are zero, remaining only the integration of the normal fluxes around the boundary of Ω_i. In order to preserve the gridless character, the Ω_i region must be defined independently of the node position. Simple figures around each node as spheres, elipsoides or hexahedra may be used.

4.4 Petrov-Galerkin Weighting

The case of non self adjoint operators $A(u)$ in equation (3) is usually treated via Petrov-Galerkin weighting ($W_i \neq N_i$) in order to avoid numerical instabilities in the solution.

For convective transport problems equivalent forms to standard Galerkin Least Square or SUPG approaches [33] may be used in which the weighting function around a node of coordinates x_p becomes:

$$W_p = N_p^p + \nabla N_p^p \frac{h}{2} \alpha \tag{39}$$

where α is the optimal "upwinding" parameter ∇ is the divergence operator and h is some normalized distance between neighbouring points. Other stabilization techniques based on Taylor-Galerkin, Characteristic Galerkin or Fractional Step

algorithms [1] are also equivalent to using a Petrov-Galerkin weighting and will be discussed in a later section when dealing with advection-diffusion problems.

5. ESSENTIAL BOUNDARY CONDITIONS

One important difference between FE and FP methods lays in the introduction of the essential boundary conditions

$$u - u_p = 0 \quad \text{in} \quad \Gamma_u$$

In the FEM this is easily done by enforcing at the equation solution level that:

$$u^h(x_i) = u_p(x_i) \qquad \text{for} \quad x_i \in \Gamma_u$$

In the FPM this procedure does not lead to satisfaction of the essential boundary conditions since

$$u^h(x_j) \neq \hat{u}(x_j)$$

Different alternatives to impose the essential boundary conditions in the FPM are described next.

5.1 Least Square Approximation

The simplest approach is to enforce pointwise (as in FEM)

$$u^h(x_j) = u_p(x_j) \qquad \text{for} \quad x_j \in \Gamma_u \tag{40}$$

Note that in this case the essential boundary conditions are satisfied in a least square sense since the FPM minimizes the square distance

$$\varphi(x_i)[u^h(x_j) - u_p(x_j)]^2 \tag{41}$$

Now the total system of equations to be solved involves $n_p - n_u$ points where n_p is the total number of points and n_u is the number of boundary points on Γ_u.

5.2 Lagrangian Multipliers

The weighted residual equation (3) is extended by adding the integrals

$$\int_{\Gamma_u} \lambda(u - u_p)\, d\Gamma + \int_{\Gamma_u} W_i(\lambda - t)\, d\Gamma \tag{42}$$

where λ are Lagrangian multipliers linked to the surface tractions t.

Discretizing the λ parameters as

$$\lambda(x) = \sum_{i=1}^{n_f} N_i^{\lambda} \lambda_i \tag{43}$$

and after integration by parts of eq.(3) a system of $n_\lambda = n_p + n_u$ equations is obtained as

$$\begin{bmatrix} \mathbf{K} & \mathbf{Q} \\ \mathbf{Q} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^h \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \tag{44}$$

with

$$Q_{ij} = -\int_{\Gamma_u} N_i N_j^\lambda \, d\Gamma \tag{45}$$

$$g_i = -\int_{\Gamma_u} N_i u_p \, d\Gamma \tag{46}$$

This approach was used in the context of the EFG method in [20].

5.3 Satisfaction in a Weak Form

The Lagrangian multipliers can be "a priori" identified as the surface tractions, i.e.

$$\lambda = t = B(u) \tag{47}$$

After integration by parts of equation (3) a system of n_p equation is obtained in this case as

$$[\mathbf{K} + \mathbf{K}^*]\mathbf{u}^h = [\mathbf{f} + \mathbf{f}^*] \tag{48}$$

with

$$\begin{aligned} \mathbf{K}^* &= -\int_{\Gamma_u} B(\mathbf{N})\mathbf{N}^T \, d\Gamma - \int_{\Gamma_u} \mathbf{N} B(\mathbf{N}^T) \, d\Gamma \\ \mathbf{f}^* &= -\int_{\Gamma_u} B(\mathbf{N}) u_p d\Gamma \end{aligned} \tag{49}$$

This method was suggested in [21].

5.4 Satisfaction in Exact Form

In FPM, the essential boundary conditions may be satisfied in a "exact" form as follows. Supose the local values of the unknown are splitted into those belonging to Γ_u and the remaining n_r, such that

$$\begin{aligned} u_i^\Gamma &= u(x_i) \qquad \text{for} \quad x_i \in \Gamma_u \\ u_i^\Gamma &= u(x_i) \qquad \text{for} \quad x_i \notin \Gamma_u \quad (i = 1, n_r) \end{aligned} \tag{50}$$

Equation (8) may be splitted into:

$$\begin{bmatrix} \mathbf{u}^\Gamma \\ \mathbf{u}^r \end{bmatrix} = \mathbf{C}\boldsymbol{\alpha} = \begin{bmatrix} \mathbf{C}_\Gamma \\ \mathbf{C}_r \end{bmatrix} \boldsymbol{\alpha} \tag{51}$$

and the least square approach may be applied to the n_r points

$$J = \sum_{j=1}^{n_r} (u_j^r - \hat{u}(x_j))^2 = \sum_{j=1}^{n_r} (u_j^r - \mathbf{C}_r \boldsymbol{\alpha})^2 \tag{52}$$

To minimize J for arbitrary $\boldsymbol{\alpha}$ with the restrictions

$$\mathbf{C}_\Gamma \boldsymbol{\alpha} = \mathbf{u}^\Gamma \tag{53}$$

is a classical minimization problem.

Let as assume without lose of generality that $u_i^\Gamma = 0$. Then the restriction (53) becomes

$$\mathbf{C}_\Gamma \boldsymbol{\alpha} = 0 \tag{54}$$

Vector $\boldsymbol{\alpha}$ can now be splitted in two parts giving

$$\mathbf{C}_{\Gamma_1} \boldsymbol{\alpha}_1 + \mathbf{C}_{\Gamma_2} \boldsymbol{\alpha}_2 = 0 \tag{55}$$

where $\mathbf{C}_{\Gamma_1}$ is as square matrix. From (55) we have

$$\boldsymbol{\alpha}_1 = -(\mathbf{C}_{\Gamma_1}^{-1} \mathbf{C}_{\Gamma_2}) \boldsymbol{\alpha}_2 = \mathbf{D} \boldsymbol{\alpha}_2 \tag{56}$$

Substituting eq.(56) into (52) yields

$$\min[\mathbf{u}^r - (\mathbf{C}_{\Gamma_1} \boldsymbol{\alpha}_1 + \mathbf{C}_{\Gamma_2} \boldsymbol{\alpha}_2)]^2 = \min[\mathbf{u}^r - \mathbf{H} \boldsymbol{\alpha}_2]^2 \quad \text{with} \quad \mathbf{H} = \mathbf{C}_{\Gamma_1} \mathbf{D} + \mathbf{C}_{\Gamma_2} \tag{57}$$

which gives after minimization

$$\boldsymbol{\alpha}_2 = [\mathbf{H}^T \mathbf{H}]^{-1} \mathbf{H}^T \mathbf{u}^r = \mathbf{G} \mathbf{u}^r \tag{58}$$

The new relationship between $\boldsymbol{\alpha}$ and $\mathbf{u}^r$ is finally

$$\boldsymbol{\alpha} = \mathbf{C}^{-1} \mathbf{u}^r \quad \text{with} \quad \mathbf{C}^{-1} = \begin{bmatrix} \mathbf{DG} \\ \mathbf{G} \end{bmatrix} \tag{59}$$

In this case, the final system of equations involves $n_p - n_u$ nodes.

6. THE FINITE POINT METHOD IN CONVECTION-DIFFUSION PROBLEMS

A FPM will be developed next for solving the non self adjoint convection-diffusion equation

$$\begin{aligned} & c\phi_{,t} + \mathbf{u} \cdot \nabla \phi - \nabla \cdot (k \nabla \phi) - Q = 0 \quad \text{in} \quad \Omega \\ & \mathbf{n} \cdot k \nabla \phi + \bar{q}_n = 0 \quad \text{in} \quad \Gamma_t \\ & \phi - \phi_p = 0 \quad \text{in} \quad \Gamma_u \end{aligned} \tag{60}$$

with the initial condition

$$\phi = \phi_0 \qquad \text{for} \quad t = t_0$$

where ∇ is the gradient operator, c, u and k are known physical parameters, ϕ the unknown field and Q a source term. $\bar{q}_n$ and $\bar{\phi}_p$ are known values of the flux and the unknown at the boundaries Γ_t and Γ_u, respectively.

It is well known that the numerical solution of non-self adjoint equations must be stabilized in order to avoid oscilations. Upwind finite difference derivatives, anisotropic balancing diffusion, Petrov-Galerkin weighting functions or characteristic time integration are some of the standard techniques used to stabilize FD, FE and FV methods [1,35]. We will test some of these approaches in the context of a FPM using a DLS approximation and point collocation.

6.1 The One Dimensional Problem

Let us consider for simplicity the case of the stationary 1D equation

$$\begin{aligned} & u\frac{\partial \phi}{\partial x} - k\frac{\partial^2 \phi}{\partial x^2} = 0 \\ & \phi = 0 \qquad \text{in} \quad x = 0 \\ & \phi = 1 \qquad \text{in} \quad x = L \end{aligned} \tag{61}$$

The FD and FV method stabilize the numerical solution of this equation by evaluating the first derivative upwind each point as

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_i - \phi_{i-1}}{h} \tag{62}$$

in which h is the distance between two points.

The FE method with linear approximation use Petrov-Galerkin weigthing functions defined as [1]

$$W = N + \frac{h}{2}\alpha\frac{\partial N}{\partial x} \tag{63}$$

where N are the linear shape function. Exact nodal solution are obtained if

$$\alpha = \coth|Pe| - \frac{1}{|Pe|} \tag{64}$$

with the Peclet number defined as

$$Pe = \frac{uh}{2k} \tag{65}$$

Both upwind derivative and Petrov-Galerkin weighting procedures can be interpreted as the addition of a balancing diffusion term to the original differential equation [1].

Quadratic finite elements typically require the definition of two upwind parameters. In ref. [34] a single parameter is proposed for quadratic elements with an expression identical to eq.(64) and h given now by half the element lenght. Exact nodal values are not obtained in this case, but the superconvergence is preserved and the method is very simple to use.

Let us try to generalize these concepts for a FPM using point collocation.

We propose as in the FD procedure to evaluate the first derivative upwind each point. Taking advantage of the continuity of the field around each node the upwind distance ξ from point x_i can be evaluated in order to obtain exact nodal values.

For quadratic base interpolating functions ($m = 3$)

$$\mathbf{p}^T = [1, x, x^2]$$

and choosing 3 point clouds ($n = 3$), exact nodal values are obtained if

$$\xi = \frac{h}{2}\alpha \tag{67}$$

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \left(\frac{\partial \phi}{\partial x}\right)_{x_i - \xi}$$

where h is defined as

$$h = (x_{i+1} - x_{i-1})/2 \tag{68}$$

Using quadratic base interpolating functions and 5 point clouds ($n = 5$), acceptable results are obtained as in quadratic finite elements (also linking five nodes) using

$$\xi = \frac{h}{2}\alpha \tag{69}$$

and evaluating now α by eqs.(64) and (65) with the distance h defined as

$$h = (x_{i+2} - x_{i-2})/2 \tag{70}$$

Figure 9 shows numerical results for eq.(61) with a source term

$$Q = \text{sen}\pi x \tag{71}$$

and boundary conditions

$$\phi_{(x=0)} = \phi_{(x=1)} = 0 \tag{72}$$

Twenty one equaly spaced points have been used in this case giving

$$Pe = \frac{uh}{2k} = 2.5 \tag{73}$$

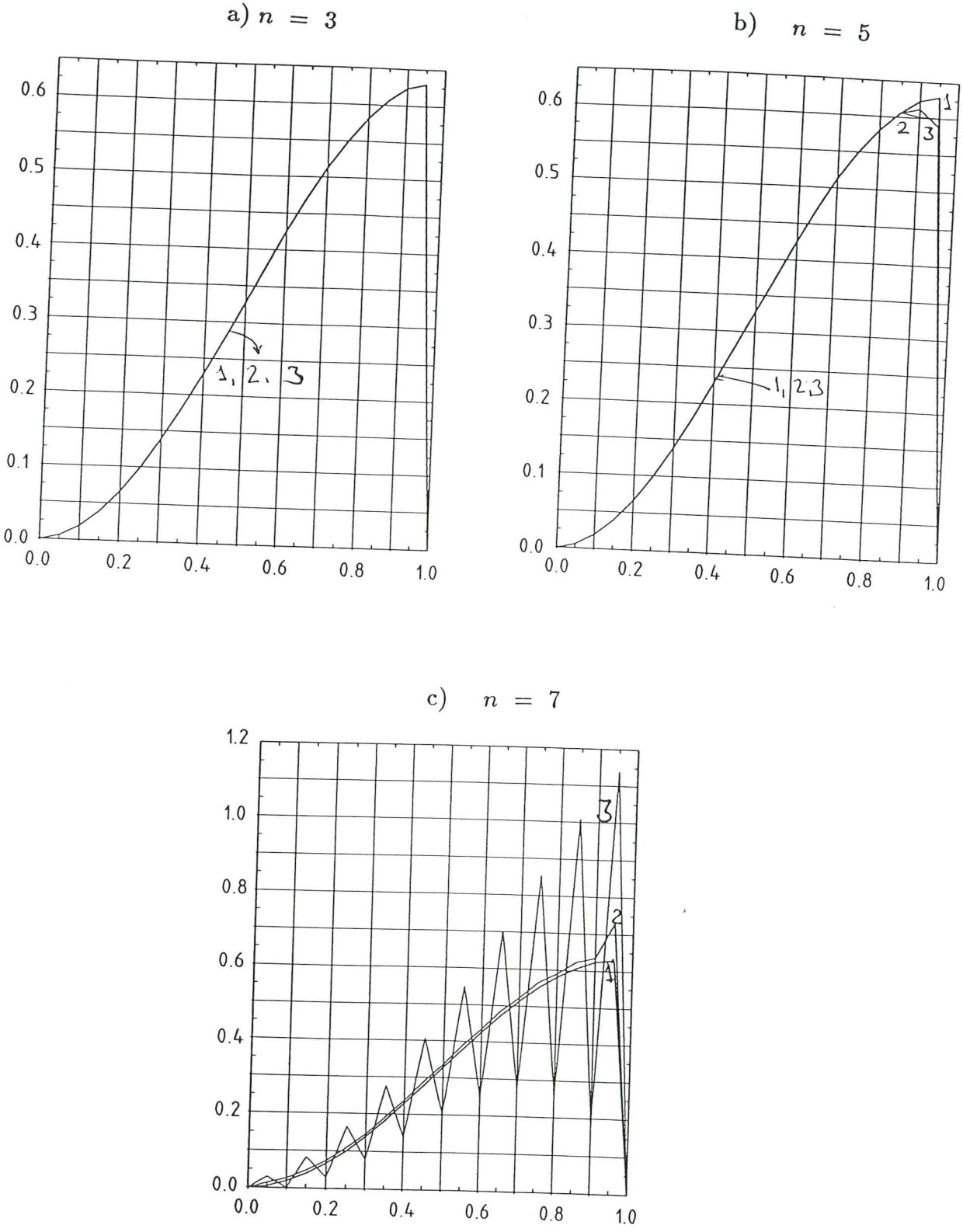


Figure 9. One dimensional convection-diffusion problem with sinusoidal source term analyzed with differents clouds. a) $n = 3$; b) $n = 5$; c) $n = 7$. Quadratic base interpolation ($m = 3$) are used in all cases. Curve 1: Analytical curve. Curve 2: Unknown function values $\hat{\phi}(x_i)$. Curve 3: Unknown parameters curve ϕ_i^h

Figures 9 a, b, c show the numerical results using clouds with $n = 3, 5$ and 7 points, respectively. On each figure, the exact results, the unknown function $\hat{\phi}(x_i)$ and the unknown parameters ϕ_i^h are shown. For $n = 3$ both solutions are coincident with the analytical one. A small over diffusion appeas for $n = 5$. Finally for $n = 7$ the unknown parameters ϕ_i^h are wrong but the values obtained for the unknown function $\hat{\phi}$ are acceptable.

6.2 The n-Dimensional Problem

For the n-dimensional problem, ξ is evaluated in the **u** direction using

$$Pe = \frac{|\mathbf{u}|h}{2k} \tag{74}$$

where the critical distance h is defined now as one half of the average projection of the major and minor distances $|\mathbf{x}_i - \mathbf{x}_j|$ in the velocity direction (see Figure 10).

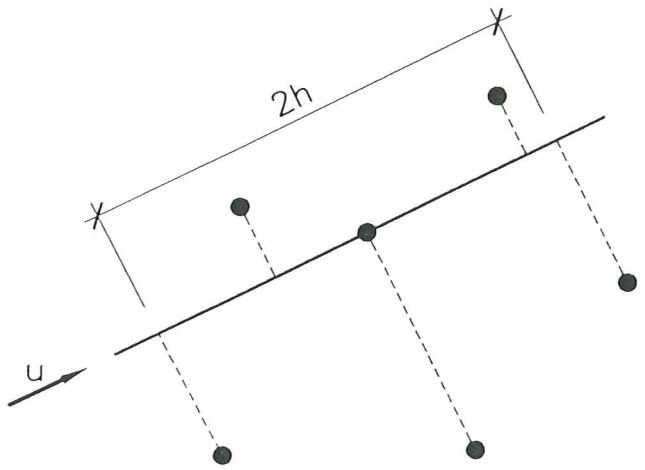


Figure 10. Definition of the critical distance h

Figure 11 shows the results of a pure thermal diffusion problem in a square domain under a uniform heat source $Q = 10$ using a regular square grid of 7×7 points. A prescribed zero value of the temperature at the boundary has been taken. A quadratic base interpolation ($m = 6$) and clouds containing 10 points ($n = 10$) have been chosen. This test shows the importance of using a weigthing least square approach with φ functions as described in eq.(27) rather than the standard Least Square method ($\varphi = 1$). Figures 11a and 11b show results for the unknown function $\hat{\phi}(x)$ and the unknown parameters ϕ_i^h when φ_i is taken equal to a fixed Gaussian function (eq. (27)). Figures 11c and d display the same results for $\varphi = 1$ (LSQ method). Note the deterioration of the solution giving non-physical results for the unknown parameters ϕ_i^h for the second case ($\varphi = 1$).

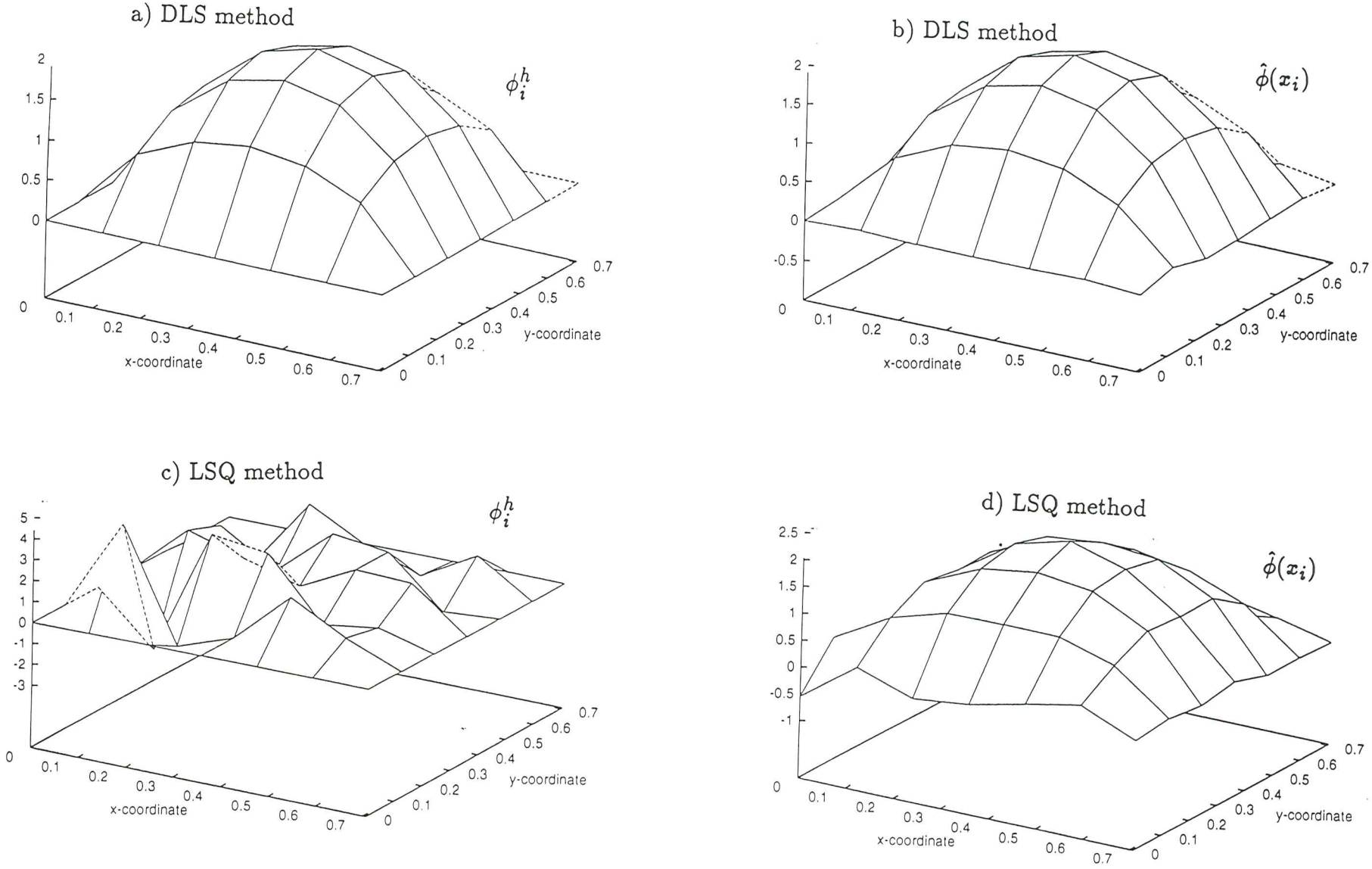


Figure 11. Pure thermal diffusion problem under constant heat source. 7×7 points grid. Quadratic base interpolation ($m = 6$). Ten points clouds ($n = 10$). Results for ϕ_i^h and $\hat{\phi}(x_i)$ with DLS and LSQ methods

Figure 12 shows results for the same example taking into account convection effects. A diagonal oriented velocity field has been chosen giving $Pe = 10$. Quadratic base interpolating polynomials are again used ($m = 6$) and each cloud contains now 9 points ($n = 9$). Figure 12a shows the unknown function and Figure 12b the unknown parameters. The numerical solution is free of oscillations and coincides with the expected result [34].

Further evidence of the importance of using a weighted approximation for solving convection-diffusion problems with the FP method can be found in [36,37].

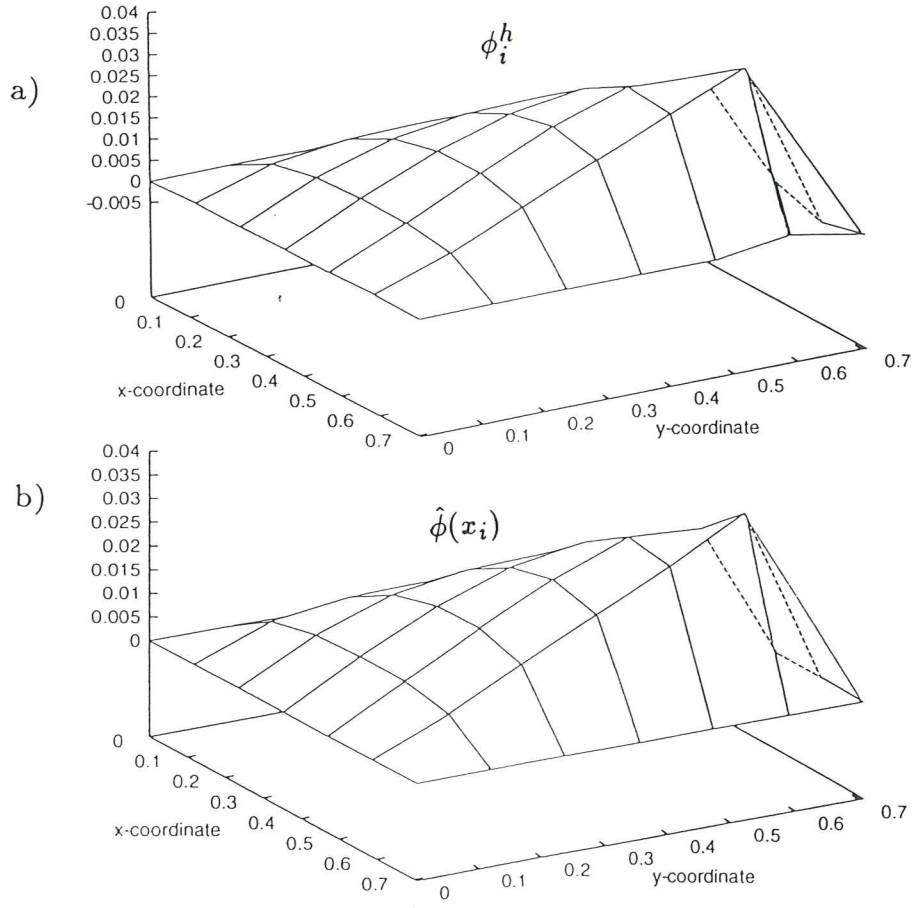


Figure 12. Convection-diffusion problem with diagonal velocity field. 7×7 points grid. Quadratic base interpolation ($m = 6$). Nine points clouds ($n = 9$). Results for ϕ_i^h and $\hat{\phi}(x_i)$ obtained with DLS method

6.3 Characteristic Approximation

Other possibility to solve equation (60) is eliminating the convective term using a lagrangian description. In this way, the operator becomes selfadjoint, and a central difference scheme may be used. This method is also known as characteristic approach [1,38].

Let

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \frac{d\phi}{dt} \tag{75}$$

Equation (60) becomes now

$$\frac{d\phi}{dt} - \nabla \cdot (k\nabla\phi) - Q = 0 \tag{76}$$

Considering $\phi^n(\mathbf{x})$ the unknown function in a coordinate $(\mathbf{x})$ at time t_n and $\phi^n(\mathbf{x} - \boldsymbol{\delta})$ the same function at the coordinate $\mathbf{x} - \boldsymbol{\delta}$ where $\boldsymbol{\delta} = \mathbf{u}\Delta t$ is a distance along the characteristic. Then

$$\phi^{n+1}(\mathbf{x}) - \phi^n(\mathbf{x} - \boldsymbol{\delta}) = \Delta t[\nabla \cdot k\nabla\phi + Q]^{n+\frac{1}{2}} \tag{77}$$

The following Taylor expansion can now be written in component form [33]

$$\phi^n(x_i - u_i\Delta t) = \phi^n(x_i) - \Delta t u_j \frac{\partial\phi^n}{\partial x_j} + \frac{\Delta t^2}{2} u_i \frac{\partial}{\partial x_i}(u_j \frac{\partial\phi}{\partial x_j})^n \tag{78}$$

$$Q^{n+\frac{1}{2}} = Q(x - \frac{\delta}{2}) = Q^n - u_i \frac{\Delta t}{2} \frac{\partial Q^n}{\partial x_i} \tag{79}$$

Substituting (78) and (79) into (77) gives finally [38]

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + \mathbf{u} \cdot \nabla\phi^n - \nabla \cdot (k\nabla\phi)^n - Q^n - \frac{\Delta t}{2}\mathbf{u}^T\nabla[\mathbf{u} \cdot \nabla\phi + Q]^n = 0 \tag{80}$$

Note that the last term may be interpreted as an artificial diffusion in which the term $\Delta t\ \mathbf{u}$ is a characteristic distance. An interesting particular case arises for $\Delta t\mathbf{u} = \mathbf{h}$ with $\mathbf{h}$ defined (in component form) as shown in Figure 10. The algorithm is now similar to the upwinding approach described in previous section and identical results are obtained for the example shown in Figure 12.

7. COMPRESSIBLE FLUID FLOWS

The FPM will be used now to solve fluid mechanics problems governed by the generalized Navier-Stokes equation

$$\frac{\partial\mathbf{u}}{\partial t} + \mathbf{A}_i \frac{\partial\mathbf{u}}{\partial x_i} + \mathbf{K}_{ij} \frac{\partial^2\mathbf{u}}{\partial x_i \partial x_j} + \mathbf{Q} = 0 \tag{81}$$

where $\mathbf{A}_i$ and $\mathbf{K}_{ij}$ denote the standard convective and diffusive matrix operators [1,35].

The well known Lax-Wendroff scheme [34] has been used giving after same algebra [1]

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathbf{A}_i \frac{\partial\mathbf{u}^n}{\partial x_i} + \mathbf{K}_{ij} \frac{\partial^2\mathbf{u}^n}{\partial x_i \partial x_j} + \mathbf{Q}^n + \frac{\Delta t}{2} \frac{\partial}{\partial x_i} \left[\mathbf{A}_i \frac{\partial}{\partial x_j}(\mathbf{A}_j\mathbf{u})\right]^n$$

$$- \frac{\Delta t}{2} \mathbf{A}_i \frac{\partial}{\partial x_i} \mathbf{Q}^n = 0 \tag{82}$$

This approach has been followed to solve the steady state Euler equation [$\mathbf{K}_{ij} = \mathbf{0}$] around a NACA 0012 profile with a Mach number at infinity of 0.3 and an angle of attack=10°.

Both a Finite Point Method with point collocation and a DLS method with a gaussian weighting function have been used for comparison. The initial distribution of 6694 points was generated with a standard unstructured advancing front triangular mesh generator [39]. The essential boundary conditions around the profile and in the incoming flow were imposed in a least square sense. A linear basic interpolating polynomial ($m = 3$) and clouds of a minimum of 5 points ($n = 5$) have been chosen.

In order to select the nodes involved in each cloud a technique of quadrants has been used consisting in defining a system of ortogonal axes in each node and taking the closest node to the origin of each quadrant. The first set of 5 nodes have been selected with this criteria (the "star" node and one node in each quadrant). New nodes must be added to the cloud if the **A** matrix defined in (15a) is singular or near singular. The new nodes were added using the same quadrant criteria.

Figure 13 shows the distribution of the pressure coefficient C_p around the profile obtained with the FPM with gaussian weighting (DLS) and constant weighting (LSQ). The potential solution is also plotted in the figure for comparison purposes. Note the higher accuracy of the DLS method as expected. This difference becomes greater if the number of points is increased. It was found that for $n \geq 10$ the results become unstable for the LSQ approximation.

Figure 14 shows a more accurate solution using a cloud of 5436 points obtained after an adaptivity criteria based on the curvature of the solution as suggested in [1]. Figure 15 shows the pressure distribution which also agrees with the expected result [35].

Further information on the solution of this problem using the FPM can be found in [36,37]

CONCLUDING REMARKS

The weighted least square interpolation combined with a simple point collocation technique is a promisory Finite Point Method for the numerical solution of computational mechanics problems. The advantage of the method compared with standard FEM is to avoid the necesity of mesh generation and compared with classical FDM is the facility to handle the boundary conditions and the non-structured distribution of points.

The method proposed seems to be as accurate as other numerical methods and the computing time is of the same order than for methods using non-structured grids.

Another interesting conclussion is the comparison with other FPM presented in the literature. Firstly, the use of a Gaussian weighting function improves considerably the results with respect to the standard LSQ approach [32,36,37]. Secondly, the sensitivity of a FPM to a variable number of points in each cloud must be low enough to preserve the freedom of adding, moving or removing points. This sensitivity is very high in FPM using the LSQ approximation, it is too large in DLS and MLS methods with linear base interpolations and it is acceptable in DLS and MLS methods using quadratic base interpolations.

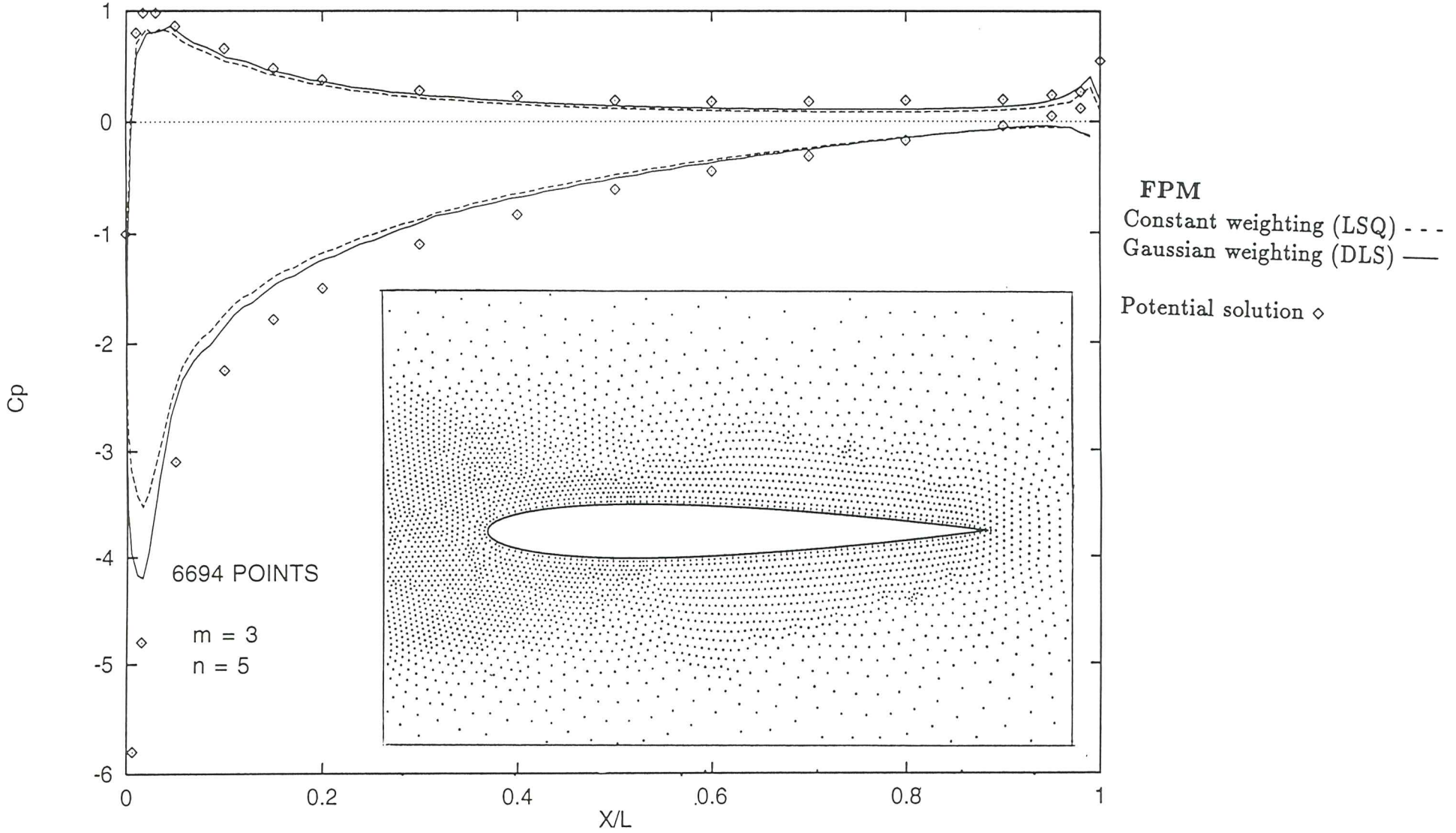


Figure 13. NACA 0012 profile, Mach=0,3, $\alpha = 10^o$. C_p distribution

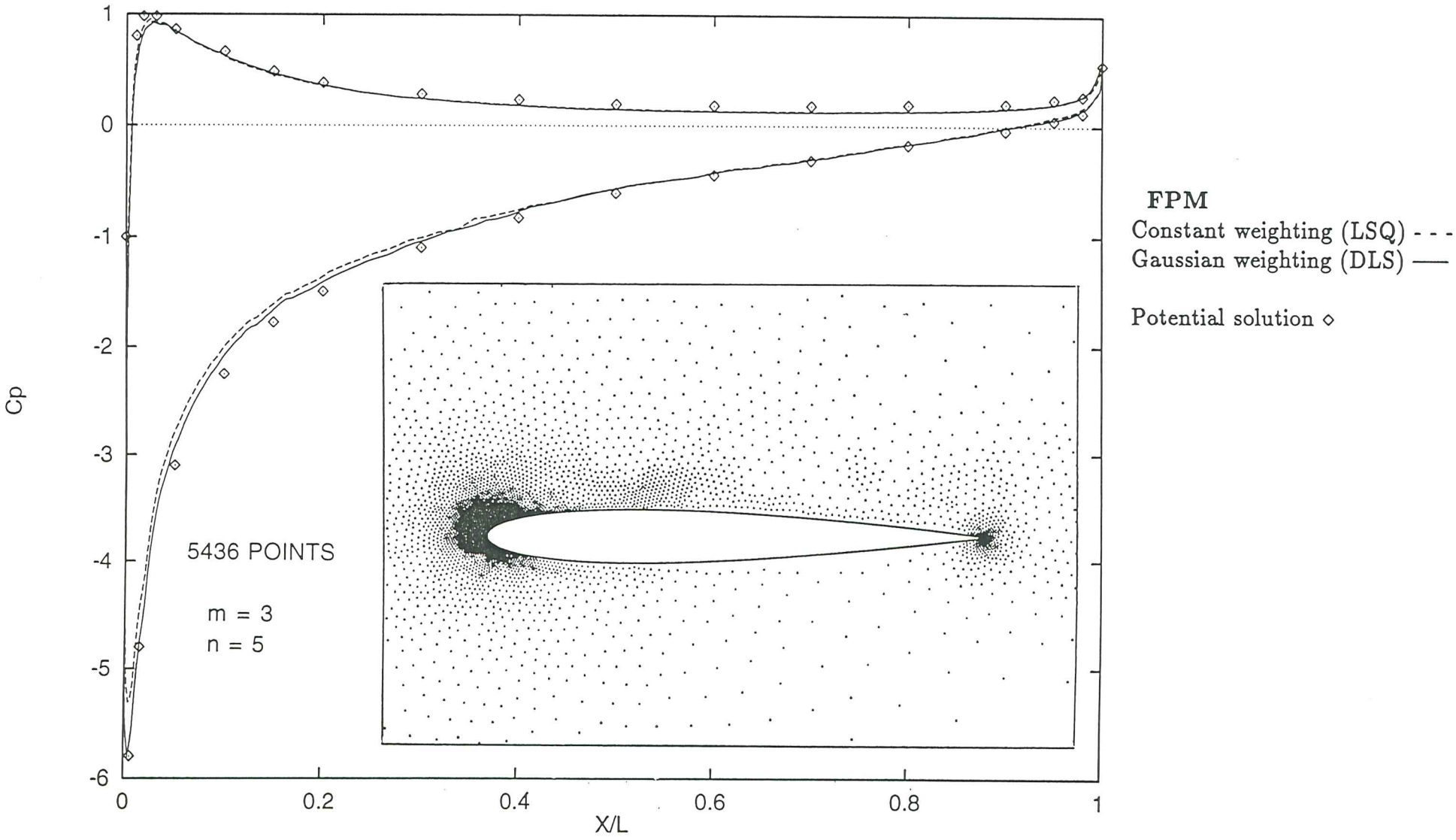


Figure 14. NACA 0012 profile, Mach=0,3, $\alpha = 10°$. C_p distribution for an adaptive distribution of points

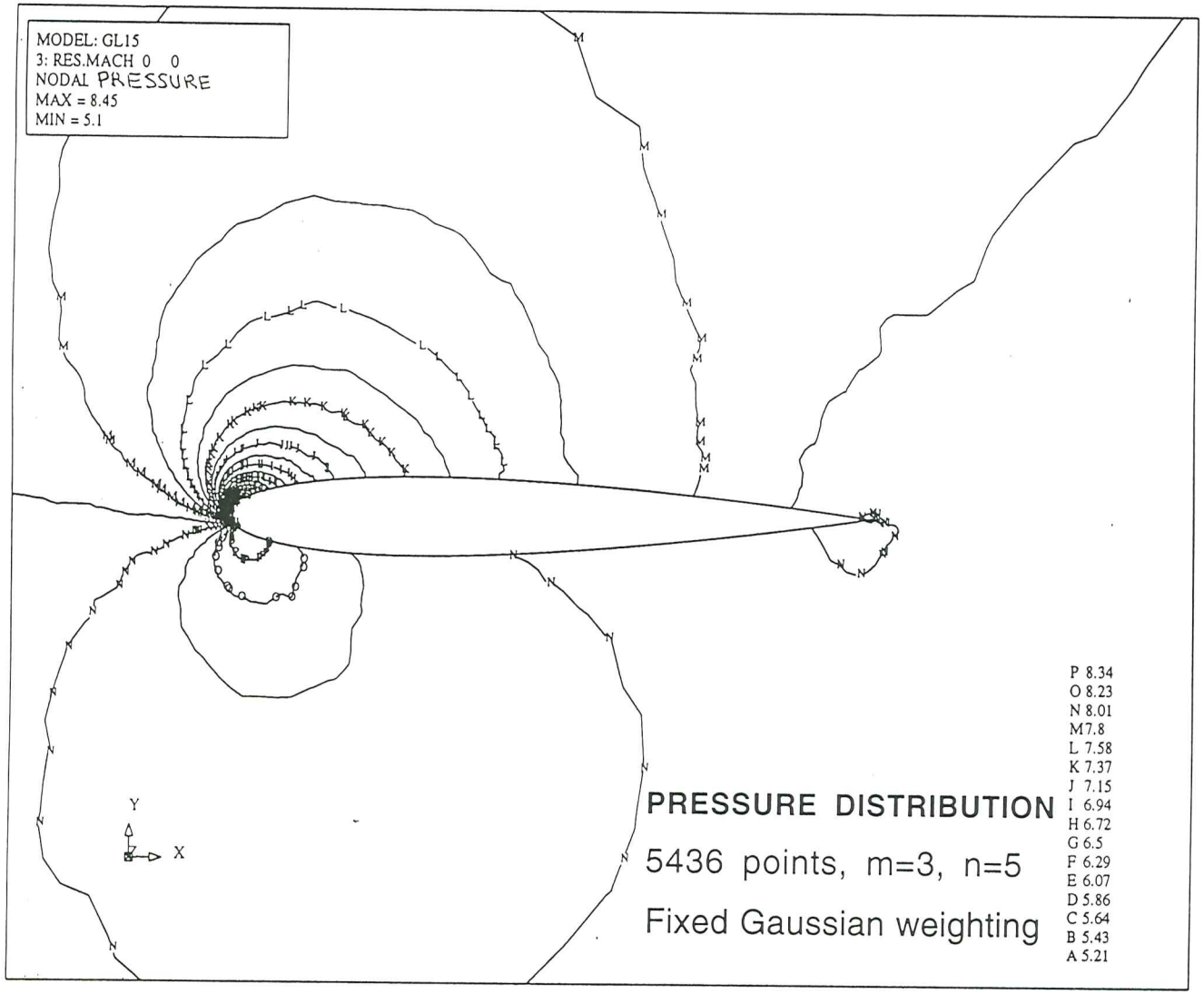


Figure 15. NACA 0012 profile, Mach=0,3. Pressure distribution

ACKNOWLEDGEMENTS

The second and third author acknowledge the Universidad Politécnica de Cataluña for supporting their visit in CIMNE, Barcelona, Spain.

Results for the last example were obtained by Mr. T. Fischer. This contribution is gratefully acknowledged.

REFERENCES

1. Zienkiewicz, O.C. and Taylor, R.L., "*The finite element method*", Mc.Graw Hill, Vol. I., 1989, Vol. 2, 1991.
2. Idelsohn, S. and Oñate, E., "Finite element and finite volumes. Two good friends." *Int. J. Num. Meth. Engng.*, **37**, pp. 3323-3341, 1994.
3. Oñate, E., Cervera, M. and Zienkiewicz, O.C., "A finite volume format for structural mechanics", *Int. J. Num. Meth. Engng.*, **37**, pp. 181-201, 1994.
4. Zienkiewicz, O.C. and Oñate, E., "Finite elements versus finite volumes. Is there a choice?", in *Non Linear Computational Mechanics. State of the Art*, P. Wriggers and W. Wagner (Eds.), Springer-Verlag, 1991.
5. MacNeal, R.H., "An asymmetrical finite difference network", *Q. Appl. Math.*, **11**, 295-310, 1953.
6. Forshythe, G.E. and Wasow, W.R., "*Finite Difference Methods for Partial Differential Equations*", Wiley, New York, 1960.
7. Jensen, P.S., "Finite difference techniques for variable grids", *Comp. Struct.*, **2**, 17-29, 1972.
8. Perrone, N. and Kao, R., "A general finite difference method for arbitrary meshes", *Comp. Struct.*, **5**, 45-47, 1975.
9. Frey, W.H., "Flexible finite difference stencils from isoparametric finite elements", *Int. J. Num. Meth. Engng.*, **11**, 1653-65, 1977.
10. Pavlin, V. and Perrone, N., "Finite difference energy techniques for arbitrary meshes applied to linear plate problems", *Int. J. Num. Meth. Engng.*, **14**, 647-64, 1979.
11. Wilkins, M.L., Blum, R.E., Cranshagen and Granthan, P., "A method for computer simulation of problems in solid mechanics and gas dynamics in three dimensions and time", Lawrence Livermore Lab., California, 1979.
12. Liska, T. and Orkisz, J., "The finite difference method at arbitrary irregular grids and its application in applied mechanics", *Comp. Struct.*, **11**, 83-95, 1980.
13. Lucy, L., "A numerical approach to testing the fission hypothesis", *A.J.*, **82**, 1013-24, 1977.
14. Gingold, R.A. and Moraghan, J.J., "Smoothed Particle Hydrodynamics: Theory and applications to non spherical stars", *Man. Not. Roy. Astron. Soc.*, **181**, 375-89, 1977.
15. Moraghan, J.J., "Why particles methods work", *SIAM J. Sci., Stat. Comput.*, **3**, 422-33, 1982.
16. Moraghan, J.J., "An introduction to SPH.", *Comp. Phys. Comm.*, **48**, 89-96, 1988.
17. Trease, H.E., Fritts, M.J. and Crowley, W.P. (Eds.), "*Advances in the Free*

Lagrange Method", Lecture Notes in Physics, Springer-Verlag, 1990.
18. Liu, W.K., Jun, S. and Belytschko, T., "Reproducing Kernel particle methods", *Int. J. Num. Meth. Fluids*, **20**, 1081–1106, 1995.
19. Nayroles, B., Touzot, G. and Villon, P., "Generalizing the FEM: Diffuse approximation and diffuse elements", *Comput. Mechanics*, **10**, 307-18, 1992.
20. Belytschko, T., Lu, Y. and Gu, L., "Element free Galerkin methods", *Int. J. Num. Meth. Engng.*, **37**, 229-56, 1994.
21. Lu, Y.Y., Belytschko, T. and Gu, L., "A new implementation of the element free Galerkin method", *Comput. Meth. Appl. Mech. Eng.*, **113**, 397–414, 1994.
22. Duarte, C.A. and Oden, J.T., "H_p clouds-A meshless method to solve boundary-value problems" TICAM Repport 95-05, May 1995.
23. Babuska, I. and Melenk, J.M., "The partition of unity finite element method", Technical Note EN-1185, Institute for Physical Science and Technology, Univ. Maryland, April 1995.
24. Sulsky, D. and Schreyer, H.L., "The Particle-in-Cell Method as a natural impact algorithm", *Advanced Computational Methods for Material Modeling*, The American Society for Mechanical Engineers, AMD-Vol. **180**, PVP- Vol. **268**, 219–229, 1993b.
25. Sulsky, D., Chen, Z. and Schreyer, H.L., "A particle method for history-dependent materials", *Computer Methods in Applied Mechanics and Engng.*, **118**, 179–96, 1994.
26. Sulsky, D., Zhon, S.J. and Schreyer, H.L., "Application of a particle-in-cell method to solid mechanics", to appear in *Comput. Phys. Comm.*
27. Liu, W.K., Jun, S., Li, S., Adee, J. and Belytschko, T., "Reproducing Kernel particle methods for structural dynamics", *Int. J. Num. Meth. Engng.*, **38**, 1655–1679, 1195..
28. Liu, W.K. and Chen, Y., "Wavelet and multiple scale reproducing Kernel methods", *Int. J. Num. Meth. Fluids*, in print, 1995.
29. Liu, W.K., Chen, Y., Jun, S., Chen, J. S., Belytschko, T., Pan, C. Uras, R.A. and Chang, C.T., "Overview and applications of the Reproducing Kernel particle methods", *Archives of Comput. Meth. in Engng.*, in print, 1996.
30. Duarte, C.A.M. "A review of some meshless methos to solve partial differential equations", TICAM Report 95-06, The Univ. of Texas, Austin, May 1995..
31. Nay, R.A. and Utku, S., "An alternative for the finite element method", *Variational Methods in Engineering*, Vol. **1**, Univ. of Southampton, 1972.
32. Batina, J., "A Gridless Euler/Navier-Stokes solution algorithm for complex aircraft aplications", AIAA 93-0333, Reno NV, January 11-14, 1993.
33. Brooks, A.N. and Hughes, T.J.R., "Streamline upwind/Petrov-Galerkin formulaltions for convective dominated flows with particular emphasis on the incompressible Navier-Stokes equations", *Comp. Meth. in Applied Mechanics and Engineering*, Vol. **32**, 1982.
34. Codina, R., Oñate, E. and Cervera, M., "The intrinsic time for the streamline upwind/Petrov-Galerkin formulations using quadratic elements", *Comp. Meth. in Applied Mechanics and Engineering*, Vol. **94**, 1992.
35. Hirsch, C., *Numerical computations of internal and external flows*, Vol. 2, J. Wiley, 1990.

36. Fisher, T., Idelsohn, S. and Oñate, E., "A meshless method for analysis of high speed flows". AGARD Meeting, Seville, October 1995.
37. Oñate, E., Idelsohn, S., Zienkiewicz, O.C. and Fisher, T., "A finite point method for analysis of fluid flow problems", Proceedings of the 9th *Int. Conference on Finite Element Methods in Fluids*, Venize, Italy, 15-21, October 1995.
38. Zienkiewicz, O.C. and Codina, R., "A general algorithm for compressible and incompressible flow. Part I: The split characteristic based scheme", *Finite Elements in Fluids.*
39. Peraire, J., Peiró, J., Formaggia, L. Morgan, K. and Zienkiewicz, O.C., "Finite element Euler computation in three dimensions", *Int. J. Num. Meth. Engng.*, **26**, 2135–59, 1988.