INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 26, 1597–1613 (1988)

ON A NON-LINEAR FORMULATION FOR CURVED TIMOSHENKO BEAM ELEMENTS CONSIDERING LARGE DISPLACEMENT/ROTATION INCREMENTS

EDUARDO N. DVORKIN

Instituto de Materiales y Estructuras, Facultad de Ingeniería, Universidad de Buenos Aires, Buenos Aires, Argentina

EUGENIO OÑATE AND JAVIER OLIVER

E.T.S. Ingenieros de Caminos, Canales y Puertos, Universidad Politécnica de Catalunya, Barcelona, Spain

SUMMARY

An incremental Total Lagrangian Formulation for curved beam elements that includes the effect of large rotation increments is developed. A complete and symmetric tangent stiffness matrix is obtained and the numerical results show, in general, an improvement over the standard formulation where the assumption of infinitesimal rotation increments is made in the derivation of the tangent stiffness matrix.

1. INTRODUCTION

The development of structural finite elements (beam and shell elements) for non-linear analysis and the development of solution methods for non-linear problems is a very active research field because the non-linear analysis of structures is an engineering application continuously demanding more efficient, robust and above all reliable numerical tools.

The elements with C^0 continuity, derived from the Ahmad, Irons and Zienkiewicz shell element[1] appear to be the most suitable ones and the majority of the latest developments in the field take this element as a starting point (e.g. References 2–8).

In the development of elements for geometrically non-linear analysis, the consideration of large rotations introduces additional difficulties due to the non-vectorial nature of finite rotations.

In this paper we concentrate in the geometrically non-linear formulation for C^0 curved beam elements (also isoparametric beam elements or Timoshenko beam elements).

A derivation of the C^0 beam element is presented by Bathe in Reference 9, and an extensive derivation of the kinematics of large rotations is presented by Argyris in Reference 10.

In the standard geometrically non-linear formulation for C^0 beam elements the tangent stiffness matrix is derived assuming infinitesimal rotation increments (rotation increments linearization) and the effect of large rotation increments is considered only during the equilibrium iterations, when calculating the stresses.

Different formulations that take into account the effect of finite rotation increments on the resulting stiffness matrices have been presented by Surana ,[11] by Simo,[12] Simo and Vu Quoc[13] and by Oñate.[14]

In this paper, we develop an incremental Total Lagrangian Formulation[9] for C^0 curved beam elements with finite incremental rotations. This formulation,

0029–5981/88/071597–17$08.50

Received 15 May 1987
Revised 17 November 1987

(a) includes in the linearization of the equations of motion all the terms that can be considered in a tangent formulation, providing therefore a complete tangent stiffness matrix;
(b) leads to symmetric stiffness matrices.

Numerical results show that, in general, the complete tangent stiffness matrix provides a faster convergence during equilibrium iterations than the stiffness matrix obtained with the standard formulation.

2. KINEMATICS OF THE TIMOSHENKO BEAM ELEMENT

In this Section we review the kinematics of C^0 beam elements with rectangular cross section and N nodes along its axis (see Figure 1).

For the initial geometry of the beam element ($t=0$) we define its axis by means of N nodes of global co-ordinates ${}^0x_i^k$; $k=1, \ldots, N$ and $i=1, 2, 3$.

We make use of Bathe's notation, therefore a superscript 0 indicates that the quantity is measured in the configuration at $t=0$.[9] Also, at each node we define the orthonormal system $({}^0\mathbf{V}_r, {}^0\mathbf{V}_s, {}^0\mathbf{V}_t)$, where ${}^0\mathbf{V}_r$ is tangent to the element axis ($t=0$).

Defining the natural co-ordinate system (r, s, t),[9] the position vector of any point inside the beam element is

$$ {}^0\mathbf{x}(r, s, t) = \sum_{k=1}^{N} h_k(r) {}^0\mathbf{x}_k + \frac{s}{2} \sum_{k=1}^{N} h_k(r) a_k {}^0\mathbf{V}_s^k + \frac{t}{2} \sum_{k=1}^{N} h_k(r) b_k {}^0\mathbf{V}_t^k \tag{1} $$

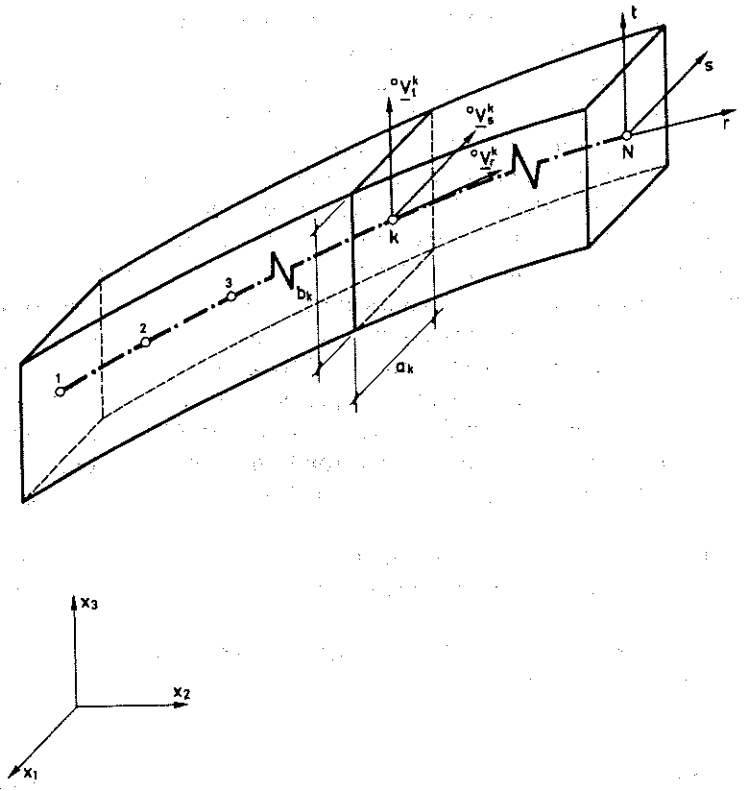


Figure 1. N-node curved beam element

where the $h_k(r)$ are the 1D interpolation functions.[9]

The deformation hypothesis is that the orthonormal systems $({}^0\mathbf{V}_r^k, {}^0\mathbf{V}_s^k, {}^0\mathbf{V}_t^k)$ rotate without deformation, hence

(a) a straight line normal to the beam axis at $t=0$ remains straight during the beam deformation but not necessarily normal to the deformed axis (shear deformations are considered);
(b) the cross section of the beam is not deformed (therefore this formulation cannot model large strain situations).

At any time t, the position vector of the (r, s, t) point in the isoparametric beam element is

$$ {}^t\mathbf{x}(r,s,t)=\sum_{k=1}^{N} h_k(r)\,{}^t\mathbf{x}_k+\frac{s}{2}\sum_{k=1}^{N} h_k(r)a_k\,{}^t\mathbf{V}_s^k+\frac{t}{2}\sum_{k=1}^{N} h_k(r)b_k\,{}^t\mathbf{V}_t^k \tag{2}$$

The displacement vector of the same point, corresponding to the configuration at time t is,

$$ {}^t\mathbf{u}={}^t\mathbf{x}-{}^0\mathbf{x} \tag{3}$$

Using (1) and (2) in (3) we obtain

$$ {}^t\mathbf{u}=\sum_{k=1}^{N} h_k\,{}^t\mathbf{u}_k+\frac{s}{2}\sum_{k=1}^{N} h_k a_k({}^t\mathbf{V}_s^k-{}^0\mathbf{V}_s^k)+\frac{t}{2}\sum_{k=1}^{N} h_k b_k({}^t\mathbf{V}_t^k-{}^0\mathbf{V}_t^k) \tag{4}$$

where ${}^t\mathbf{u}_k$ is the displacement vector of node k, at time t.

Since the orthonormal system at node k rotates,

$$ \begin{aligned} {}^t\mathbf{V}_s^k&={}_0^t\mathbf{R}^k\,{}^0\mathbf{V}_s^k \\ {}^t\mathbf{V}_t^k&={}_0^t\mathbf{R}^k\,{}^0\mathbf{V}_t^k \end{aligned} \tag{5}$$

where ${}_0^t\mathbf{R}^k$ is the rotation matrix corresponding to node k, at time t and referred to the initial configuration.

The rotation of the orthonormal system at node k can be described by a vector[10]

$$ {}_0^t\boldsymbol{\theta}^k={}_0^t\theta^k\,{}^t\mathbf{e}^k \tag{6a}$$

$$ {}_0^t\theta^k=[({}_0^t\theta_1^k)^2+({}_0^t\theta_2^k)^2+({}_0^t\theta_3^k)^2]^{1/2} \tag{6b}$$

where ${}^t\mathbf{e}^k$ is a unit vector in the direction of the rotation axis.

Studying this rotation, Argyris arrived at (Reference 10, equation (16))

$$ {}_0^t\mathbf{R}^k=\mathbf{I}_3+\frac{\sin {}_0^t\theta^k}{{}_0^t\theta^k}\,{}_0^t\boldsymbol{\Theta}^k+\frac{1}{2}\left[\frac{\sin({}_0^t\theta^k/2)}{({}_0^t\theta^k/2)}\right]^2({}_0^t\boldsymbol{\Theta}^k)^2 \tag{7}$$

where

$$ {}_0^t\boldsymbol{\Theta}^k=\begin{bmatrix} 0 & -{}_0^t\theta_3^k & {}_0^t\theta_2^k \\ {}_0^t\theta_3^k & 0 & -{}_0^t\theta_1^k \\ -{}_0^t\theta_2^k & {}_0^t\theta_1^k & 0 \end{bmatrix} \tag{8}$$

Note that the ${}_0^t\theta_i^k$ are not independent rotations around the global axes but are the components of the matrix defined in equation (8), which characterizes a rotation around the axis ${}^t\mathbf{e}^k$.

In a very elegant way, Argyris proved that equation (7) can be rewritten as (Reference 10, equation (31))

$$ {}^{t}_{0}\mathbf{R}^{k} = \mathbf{I}_3 + {}^{t}_{0}\boldsymbol{\Theta}^{k} + \frac{1}{2!}({}^{t}_{0}\boldsymbol{\Theta}^{k})^2 + \frac{1}{3!}({}^{t}_{0}\boldsymbol{\Theta}^{k})^3 + \ldots \tag{9} $$

where $\mathbf{I}_3$ is a (3×3) unit matrix.

The incremental displacement from the configuration at time t to the configuration at time $t+\Delta t$ is,

$$ \mathbf{u} = {}^{t+\Delta t}\mathbf{x} - {}^{t}\mathbf{x} \tag{10} $$

Therefore,

$$ \mathbf{u} = \sum_{k=1}^{N} h_k \mathbf{u}^k + \frac{S}{2} \sum_{k=1}^{N} h_k a_k ({}^{t+\Delta t}\mathbf{V}_s^k - {}^{t}\mathbf{V}_s^k) + \frac{t}{2} \sum_{k=1}^{N} h_k b_k ({}^{t+\Delta t}\mathbf{V}_t^k - {}^{t}\mathbf{V}_t^k) \tag{11} $$

Since to go from the configuration at time t to the configuration at time $t+\Delta t$ the orthonormal system at node k is only rotated,

$$ \begin{aligned} {}^{t+\Delta t}\mathbf{V}_s^k &= {}^{t+\Delta t}_{t}\mathbf{R}^k \, {}^{t}\mathbf{V}_s^k \\ {}^{t+\Delta t}\mathbf{V}_t^k &= {}^{t+\Delta t}_{t}\mathbf{R}^k \, {}^{t}\mathbf{V}_t^k \end{aligned} \tag{12} $$

Using equation (9),

$$ {}^{t+\Delta t}_{t}\mathbf{R}^k = \mathbf{I}_3 + \boldsymbol{\Theta}^k + \frac{1}{2!}(\boldsymbol{\Theta}^k)^2 + \frac{1}{3!}(\boldsymbol{\Theta}^k)^3 + \ldots \tag{13} $$

where

$$ \boldsymbol{\Theta}^k = \begin{bmatrix} 0 & -\theta_3^k & \theta_2^k \\ \theta_3^k & 0 & -\theta_1^k \\ -\theta_2^k & \theta_1^k & 0 \end{bmatrix} \tag{14} $$

again, the θ_i^k are not independent incremental rotations around the global axes but are the components of the matrix defined in equation (14), which characterizes via equation (13) the incremental rotation at node k.

Note that, if the incremental rotation is infinitesimal, equations (12) to (14) produce, keeping in equation (13) only the linear terms

$$ \begin{aligned} {}^{t+\Delta t}\mathbf{V}_s^k - {}^{t}\mathbf{V}_s^k &= \boldsymbol{\theta}^k \times {}^{t}\mathbf{V}_s^k \\ {}^{t+\Delta t}\mathbf{V}_t^k - {}^{t}\mathbf{V}_t^k &= \boldsymbol{\theta}^k \times {}^{t}\mathbf{V}_t^k \end{aligned} \tag{15} $$

where $[\boldsymbol{\theta}^k]^{\mathrm{T}} = [\theta_1^k \quad \theta_2^k \quad \theta_3^k]$. In this case the θ_i^k are independent infinitesimal incremental rotations around the three global axes.

Because the incremental rotations are finite we keep in (13), in a first attempt, only the linear and quadratic terms.

Therefore,

$$ \begin{aligned} {}^{t+\Delta t}\mathbf{V}_s^k - {}^{t}\mathbf{V}_s^k &= \boldsymbol{\Theta}^k \, {}^{t}\mathbf{V}_s^k + \tfrac{1}{2}\boldsymbol{\Theta}^k \boldsymbol{\Theta}^k \, {}^{t}\mathbf{V}_s^k \\ {}^{t+\Delta t}\mathbf{V}_t^k - {}^{t}\mathbf{V}_t^k &= \boldsymbol{\Theta}^k \, {}^{t}\mathbf{V}_t^k + \tfrac{1}{2}\boldsymbol{\Theta}^k \boldsymbol{\Theta}^k \, {}^{t}\mathbf{V}_t^k \end{aligned} \tag{16} $$

which can be rewritten as

$$ \begin{aligned} {}^{t+\Delta t}\mathbf{V}_s^k - {}^{t}\mathbf{V}_s^k &= \boldsymbol{\theta}^k \times {}^{t}\mathbf{V}_s^k + \tfrac{1}{2}\boldsymbol{\theta}^k \times (\boldsymbol{\theta}^k \times {}^{t}\mathbf{V}_s^k) \\ {}^{t+\Delta t}\mathbf{V}_t^k - {}^{t}\mathbf{V}_t^k &= \boldsymbol{\theta}^k \times {}^{t}\mathbf{V}_t^k + \tfrac{1}{2}\boldsymbol{\theta}^k \times (\boldsymbol{\theta}^k \times {}^{t}\mathbf{V}_t^k) \end{aligned} \tag{17} $$

$$ {}^{t}\mathbf{g}_r \cdot \frac{\partial \mathbf{u}_{R2}}{\partial s} = \mathbf{U}^{\mathrm{T}}\, {}^{t}_{0}\tilde{\mathbf{B}}^{2}_{rs}\mathbf{U} \tag{41b} $$

$$ {}^{t}\mathbf{g}_r \cdot \frac{\partial \mathbf{u}_{R2}}{\partial t} = \mathbf{U}^{\mathrm{T}}\, {}^{t}_{0}\tilde{\mathbf{B}}^{2}_{rt}\mathbf{U} \tag{41c} $$

$$ {}^{t}\mathbf{g}_s \cdot \frac{\partial \mathbf{u}_{R2}}{\partial r} = \mathbf{U}^{\mathrm{T}}\, {}^{t}_{0}\tilde{\mathbf{B}}^{2}_{sr}\mathbf{U} \tag{41d} $$

$$ {}^{t}\mathbf{g}_t \cdot \frac{\partial \mathbf{u}_{R2}}{\partial r} = \mathbf{U}^{\mathrm{T}}\, {}^{t}_{0}\tilde{\mathbf{B}}^{2}_{tr}\mathbf{U} \tag{41e} $$

Matrices $\tilde{\mathbf{B}}_r$, $\tilde{\mathbf{B}}_s$, $\tilde{\mathbf{B}}_t$ can be easily obtained using the kinematic relations presented in Section 2. Matrices ${}^{t}_{0}\tilde{\mathbf{B}}^{2}_{rr}, \ldots, {}^{t}_{0}\tilde{\mathbf{B}}^{2}_{rt}$ arise from the underlined terms in equations (27).

Being k_i and k_l the degrees of freedom corresponding to θ^k_i and θ^k_l; $k = 1, \ldots, N$ and $i, l = 1, 2, 3$, the only non-zero terms in those matrices are

$$ {}^{t}_{0}\tilde{B}^{2}_{rr}(k_i, k_l) = \frac{s}{8} h_{k,r} a_k [{}^{t}V^{k}_{s(l)}\, {}^{t}g_{r(i)} + {}^{t}V^{k}_{s(i)}\, {}^{t}g_{r(l)} - ({}^{t}\mathbf{V}^{k}_{s} \cdot {}^{t}\mathbf{g}_r)(\delta_{il} + \delta_{li})] $$

$$ + \frac{t}{8} h_{k,r} b_k [{}^{t}V^{k}_{t(l)}\, {}^{t}g_{r(i)} + {}^{t}V^{k}_{t(i)}\, {}^{t}g_{r(l)} - ({}^{t}\mathbf{V}^{k}_{t} \cdot {}^{t}\mathbf{g}_r)(\delta_{il} + \delta_{li})] \tag{42a} $$

$$ {}^{t}_{0}\tilde{B}^{2}_{rs}(k_i, k_l) = \tfrac{1}{8} h_k a_k [{}^{t}V^{k}_{s(l)}\, {}^{t}g_{r(i)} + {}^{t}V^{k}_{s(i)}\, {}^{t}g_{r(l)} - ({}^{t}\mathbf{V}^{k}_{s} \cdot {}^{t}\mathbf{g}_r)(\delta_{il} + \delta_{li})] \tag{42b} $$

$$ {}^{t}_{0}\tilde{B}^{2}_{rt}(k_i, k_l) = \tfrac{1}{8} h_k b_k [{}^{t}V^{k}_{t(l)}\, {}^{t}g_{r(i)} + {}^{t}V^{k}_{t(i)}\, {}^{t}g_{r(l)} - ({}^{t}\mathbf{V}^{k}_{t} \cdot {}^{t}\mathbf{g}_r)(\delta_{il} + \delta_{li})] \tag{42c} $$

$$ {}^{t}_{0}\tilde{B}^{2}_{sr}(k_i, k_l) = \frac{s}{8} h_{k,r} a_k [{}^{t}V^{k}_{s(l)}\, {}^{t}g_{s(i)} + {}^{t}V^{k}_{s(i)}\, {}^{t}g_{s(l)} - ({}^{t}\mathbf{V}^{k}_{s} \cdot {}^{t}\mathbf{g}_s)(\delta_{il} + \delta_{li})] $$

$$ + \frac{t}{8} h_{k,r} b_k [{}^{t}V^{k}_{t(l)}\, {}^{t}g_{s(i)} + {}^{t}V^{k}_{t(i)}\, {}^{t}g_{s(l)} - ({}^{t}\mathbf{V}^{k}_{t} \cdot {}^{t}\mathbf{g}_s)(\delta_{il} + \delta_{li})] \tag{42d} $$

$$ {}^{t}_{0}\tilde{B}^{2}_{tr}(k_i, k_l) = \frac{s}{8} h_{k,r} a_k [{}^{t}V^{k}_{s(l)}\, {}^{t}g_{t(i)} + {}^{t}V^{k}_{s(i)}\, {}^{t}g_{t(l)} - ({}^{t}\mathbf{V}^{k}_{s} \cdot {}^{t}\mathbf{g}_t)(\delta_{il} + \delta_{li})] $$

$$ + \frac{t}{8} h_{k,r} b_k [{}^{t}V^{k}_{t(l)}\, {}^{t}g_{t(i)} + {}^{t}V^{k}_{t(i)}\, {}^{t}g_{t(l)} - ({}^{t}\mathbf{V}^{k}_{t} \cdot {}^{t}\mathbf{g}_t)(\delta_{il} + \delta_{li})] \tag{42e} $$

where δ_{ij} is the Kronecker delta.

It is important to point out that the above defined matrices are symmetric. From equations (39) to (42)

$$ {}^{t}_{0}\mathbf{K}_{\mathrm{NL}} = \int_{{}^{0}V} [{}^{t}_{0}\tilde{S}^{rr}\tilde{\mathbf{B}}^{\mathrm{T}}_{r}\tilde{\mathbf{B}}_{r} + {}^{t}_{0}\tilde{S}^{rs}(\tilde{\mathbf{B}}^{\mathrm{T}}_{r}\tilde{\mathbf{B}}_{s} + \tilde{\mathbf{B}}^{\mathrm{T}}_{s}\tilde{\mathbf{B}}_{r}) + {}^{t}_{0}\tilde{S}^{rt}(\tilde{\mathbf{B}}^{\mathrm{T}}_{r}\tilde{\mathbf{B}}_{t} + \tilde{\mathbf{B}}^{\mathrm{T}}_{t}\tilde{\mathbf{B}}_{r})]^{0}\mathrm{d}V $$

$$ + \int_{{}^{0}V} 2[{}^{t}_{0}\tilde{S}^{rr}\, {}^{t}_{0}\tilde{\mathbf{B}}^{2}_{rr} + {}^{t}_{0}\tilde{S}^{rs}({}^{t}_{0}\tilde{\mathbf{B}}^{2}_{rs} + {}^{t}_{0}\tilde{\mathbf{B}}^{2}_{sr}) + {}^{t}_{0}\tilde{S}^{rt}({}^{t}_{0}\tilde{\mathbf{B}}^{2}_{rt} + {}^{t}_{0}\tilde{\mathbf{B}}^{2}_{tr})]^{0}\mathrm{d}V \tag{43} $$

Note that

(a) The tangent stiffness matrix, ${}^{t}_{0}\mathbf{K} = {}^{t}_{0}\mathbf{K}_{\mathrm{L}} + {}^{t}_{0}\mathbf{K}_{\mathrm{NL}}$ is symmetric.

(b) The second integral in equation (43) represents the difference with the tangent matrix as obtained with the standard formulation.

4.2. Internal forces

The vector of generalized internal nodal forces equivalent in the virtual work sense to the element stresses is[9]

$$ {}_0^t\mathbf{F} = \int_{{}_0V} {}_0^t\tilde{\mathbf{B}}_L^T \, {}_0^t\tilde{\mathbf{S}} \, {}^0dV \tag{44} $$

For calculating the stresses, the vectors ${}^t\mathbf{V}_s^k$ and ${}^t\mathbf{V}_t^k$ are updated using equation (7).

5. NUMERICAL EXPERIMENTATION

In this section we will compare for some simple examples the results obtained using the standard formulation and using the formulation presented in this paper. In order to avoid the locking problem,[19] reduced numerical integration will be used along the r-direction.

It is important to point out that in beam elements reduced integration does not produce spurious zero energy modes,[20] and therefore does not raise objections from the reliability view point.[4]

Full Newton–Raphson iterations are used in all the examples and the energy criterion[9, 21] is used to test for convergence. Therefore, we stop the iterations when

$$ [\Delta \mathbf{U}^{(i)}]^T ({}^{t+\Delta t}\mathbf{P} - {}_0^{t+\Delta t}\mathbf{F}^{(i-1)}) \leqslant \mathrm{ETOL}\,[\Delta \mathbf{U}^{(1)}]^T ({}^{t+\Delta t}\mathbf{P} - {}_0^t\mathbf{F}) \tag{45} $$

where ETOL is an error tolerance defined for each case.

5.1. Cantilever beam under constant moment

5.1.1. Analysis using 2-node elements. The cantilever is analysed using four 2-node elements, as shown in Figure 2. The total moment of 2π, which for our case ($EI/L = 2$) corresponds to a total tip rotation of π, is applied in 10 equal steps. Our formulation needs a total of 33 iterations to converge, against 92 iterations of the standard formulation. The coding differences for both formulations are very minor, therefore the number of iterations can be considered as an approximate indicator of the computational efficiency.

5.1.2. Analysis using 3-node elements. In the example shown in Figure 3 four 3-node elements are used. A total moment of $1{\cdot}8\pi\ EI/L$ is applied in 10 steps, as indicated in the figure. Surana[11] also analysed this problem.

In Figure 3 we display our results and the results reported in Reference 11; it is worth pointing out that, since different convergence criteria were used, the comparison of the number of iterations used by each formulation is not necessarily indicative of the effectiveness of each formulation.

5.2. Simply supported beam under constant moment

The beam is analysed using five 3-node elements, as shown in Figure 4. A total moment of 2·0 (which corresponds to a relative rotation of the beam ends of $2{\cdot}55\pi$) is applied in 10 equal steps. Our formulation needs a total of 44 iterations to converge against 151 iterations of the standard formulation.

Surana[11] and Oliver[22] also analysed this problem. In Figure 4 we display their results and the present results. Again, different convergence criteria were used in the three cases and the comparison is not straightforward.

where ${}^0V$ is the volume in the initial configuration ($t=0$), ${}^{t+\Delta t}_0\tilde{S}^{ij}$ are the contravariant components measured in the convected system of the 2nd Piola–Kirchhoff stress tensor[9, 16] and ${}^{t+\Delta t}\mathscr{R}$ is the virtual work of the external loads acting on the configuration at time $t+\Delta t$.

When calculating ${}^{t+\Delta t}\mathscr{R}$ it is important to notice that, although we are considering finite incremental rotations, the $\delta\theta_i^k$ are infinitesimal, therefore the virtual work of the applied moments is directly given by $\sum_{k=1}^{NN} M_i^k \delta\theta_i^k$ where NN is the total number of nodes in the model.

Working out equation (29) the linearized equations of motion are obtained (Reference 9, Chapter 6).

4. INCREMENTAL FORMULATION FOR THE TIMOSHENKO BEAM ELEMENT

Using the kinematic equations presented in Section 2, we develop in this section the incremental Total Lagrangian Formulation[9] for the Timoshenko beam element.

With the Newton–Raphson iteration scheme[9] the equations for the ith iteration in a finite elements model are

$$({}^{t+\Delta t}_0\mathbf{K}_L + {}^{t+\Delta t}_0\mathbf{K}_{NL})^{(i-1)}\Delta\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{P} - {}^{t+\Delta t}_0\mathbf{F}^{(i-1)} \tag{30a}$$

For the displacements,

$$\mathbf{U}^{(i)} = \mathbf{U}^{(i-1)} + \Delta\mathbf{U}^{(i)} \tag{30b}$$

and for the rotations

$$({}^{t+\Delta t}_t\mathbf{R}^k)^{(i)} = (\Delta\,{}^{t+\Delta t}_t\mathbf{R}^k)^{(i)}({}^{t+\Delta t}_t\mathbf{R}^k)^{(i-1)} \tag{30c}$$

In the above

${}^{t+\Delta t}_0\mathbf{K}_L$ is the linear part of the tangent stiffness matrix,
${}^{t+\Delta t}_0\mathbf{K}_{NL}$ the non-linear part of the tangent stiffness matrix,
$\mathbf{U}$ the vector of generalized nodal incremental displacements,
${}^{t+\Delta t}\mathbf{P}$ the vector of generalized external nodal loads acting at $t+\Delta t$

and

${}^{t+\Delta t}_0\mathbf{F}$ the vector of generalized internal nodal loads acting at $t+\Delta t$, equivalent (in the virtual work sense) to the element stresses.

4.1. Tangent stiffness matrix

We define a vector

$${}_0\tilde{\mathbf{e}}^T = [{}_0\tilde{e}_{rr} \quad 2\,{}_0\tilde{e}_{rs} \quad 2\,{}_0\tilde{e}_{rt}] \tag{31}$$

and the usual relation[9]

$${}_0\tilde{\mathbf{e}} = {}^t_0\tilde{\mathbf{B}}_L\mathbf{U} \tag{32}$$

The matrix ${}^t_0\tilde{\mathbf{B}}_L$ is derived using equations (27) and (28).

From the linearized equations of motion (Reference 9, Chapter 6) we obtain,

$${}^t_0\mathbf{K}_L = \int_{{}_0V} {}^t_0\tilde{\mathbf{B}}_L^T\,{}_0\tilde{\mathbf{C}}\;{}^t_0\tilde{\mathbf{B}}_L\,{}^0\mathrm{d}V \tag{33}$$

where ${}_0\tilde{\mathbf{C}}$ is a constitutive matrix formed with the contravariant components ${}_0\tilde{C}^{ijkl}$ of the

constitutive tensor that relates the increments of the contravariant components of the 2nd Piola–Kirchhoff stress tensor, measured in the convected system (${}_0\tilde{S}^{ij}$), with the increments of the covariant components of the Green–Lagrange strain tensor also measured in the convected system (${}_0\tilde{\varepsilon}_{ij}$). The incremental constitutive equation is, in matrix notation,

$$ {}_0\tilde{\mathbf{S}} = {}_0\tilde{\mathbf{C}}\,{}_0\tilde{\boldsymbol{\varepsilon}} \tag{34} $$

where

$$ {}_0\tilde{\mathbf{S}}^{\mathrm{T}} = [{}_0\tilde{S}^{rr} \quad {}_0\tilde{S}^{rs} \quad {}_0\tilde{S}^{rt}] \tag{35} $$

and

$$ {}_0\tilde{\boldsymbol{\varepsilon}}^{\mathrm{T}} = [{}_0\tilde{\varepsilon}_{rr} \quad 2\,{}_0\tilde{\varepsilon}_{rs} \quad 2\,{}_0\tilde{\varepsilon}_{rt}] \tag{36} $$

The curvilinear components ${}_0\tilde{C}^{ijkl}$ are calculated from the components in an orthonormal system $(\hat{e}_i, \hat{e}_j, \hat{e}_k)$:[2]

$$ {}_0\tilde{C}^{ijkl} = {}_0\hat{C}_{mnpq}({}^0\mathbf{g}^i \cdot \hat{\mathbf{e}}_m)({}^0\mathbf{g}^j \cdot \hat{\mathbf{e}}_n)({}^0\mathbf{g}^k \cdot \hat{\mathbf{e}}_p)({}^0\mathbf{g}^l \cdot \hat{\mathbf{e}}_q) \tag{37} $$

For elements of constant (a_k, b_k) we can define an orthonormal system $\hat{\mathbf{e}}_i = {}^0\mathbf{g}_i / \| {}^0\mathbf{g}_i \|$, and

$$ {}_0\hat{\mathbf{C}} = \begin{bmatrix} E & 0 & 0 \\ 0 & \kappa G & 0 \\ 0 & 0 & \kappa G \end{bmatrix} \tag{38} $$

where E is the Young's modulus, G is the shear modulus and κ is the shear correction factor.[9]

The linear part of the tangent stiffness matrix, as defined by equation (33), is the same for our formulation and for the standard formulation. The difference will appear only in the non-linear part of the tangent stiffness matrix.

We derive the non-linear part of the tangent stiffness matrix from the equality (Reference 9, Chapter 6)

$$ \delta\mathbf{U}^{\mathrm{T}}\,{}^t_0\mathbf{K}_{\mathrm{NL}}\mathbf{U} = \int_{{}^0V} {}^t_0\tilde{S}^{ij}\,\delta_0\tilde{\eta}_{ij}\,{}^0\mathrm{d}V \tag{39} $$

We define now the following matrices:

$$ \frac{\partial \mathbf{u}_{\mathrm{S}}}{\partial r} = \begin{bmatrix} \partial u_{\mathrm{S}(1)}/\partial r \\ \partial u_{\mathrm{S}(2)}/\partial r \\ \partial u_{\mathrm{S}(3)}/\partial r \end{bmatrix} = \tilde{\mathbf{B}}_r\mathbf{U} \tag{40a} $$

$$ \frac{\partial \mathbf{u}_{\mathrm{S}}}{\partial s} = \begin{bmatrix} \partial u_{\mathrm{S}(1)}/\partial s \\ \partial u_{\mathrm{S}(2)}/\partial s \\ \partial u_{\mathrm{S}(3)}/\partial s \end{bmatrix} = \tilde{\mathbf{B}}_s\mathbf{U} \tag{40b} $$

$$ \frac{\partial \mathbf{u}_{\mathrm{S}}}{\partial t} = \begin{bmatrix} \partial u_{\mathrm{S}(1)}/\partial t \\ \partial u_{\mathrm{S}(2)}/\partial t \\ \partial u_{\mathrm{S}(3)}/\partial t \end{bmatrix} = \tilde{\mathbf{B}}_t\mathbf{U} \tag{40c} $$

and also,

$$ {}^t\mathbf{g}_r \cdot \frac{\partial \mathbf{u}_{\mathrm{R}2}}{\partial r} = \mathbf{U}^{\mathrm{T}}\,{}^t_0\tilde{\mathbf{B}}^2_{rr}\mathbf{U} \tag{41a} $$

The incremental displacement of any point (r, s, t) inside the beam element is

$$\mathbf{u}=\sum_{k=1}^{N} h_k\mathbf{u}^k+\frac{s}{2}\sum_{k=1}^{N} h_k a_k[\boldsymbol{\theta}^k\times{}^t\mathbf{V}_s^k+\tfrac{1}{2}\boldsymbol{\theta}^k\times(\boldsymbol{\theta}^k\times{}^t\mathbf{V}_s^k)]$$

$$+\frac{t}{2}\sum_{k=1}^{N} h_k b_k[\boldsymbol{\theta}^k\times{}^t\mathbf{V}_t^k+\tfrac{1}{2}\boldsymbol{\theta}^k\times(\boldsymbol{\theta}^k\times{}^t\mathbf{V}_t^k)] \tag{18}$$

We can write,

$$\mathbf{u}=\mathbf{u}_S+\mathbf{u}_{R2} \tag{19}$$

where $\mathbf{u}_S$ are the terms obtained considering only infinitesimal rotation increments, i.e. using equation (15) (standard linearization), and $\mathbf{u}_{R2}$ are the extra terms obtained using equation (17).

Hence,

$$\mathbf{u}_S=\sum_{k=1}^{N} h_k\mathbf{u}^k+\frac{s}{2}\sum_{k=1}^{N} h_k a_k(\boldsymbol{\theta}^k\times{}^t\mathbf{V}_s^k)+\frac{t}{2}\sum_{k=1}^{N} h_k b_k(\boldsymbol{\theta}^k\times{}^t\mathbf{V}_t^k) \tag{20}$$

and

$$\mathbf{u}_{R2}=\frac{s}{4}\sum_{k=1}^{N} h_k a_k[\boldsymbol{\theta}^k\times(\boldsymbol{\theta}^k\times{}^t\mathbf{V}_s^k)]+\frac{t}{4}\sum_{k=1}^{N} h_k b_k[\boldsymbol{\theta}^k\times(\boldsymbol{\theta}^k\times{}^t\mathbf{V}_t^k)] \tag{21}$$

At any time t, the covariant basis of the convected system (r, s, t) are[15]

$${}^t\mathbf{g}_r=\frac{\partial{}^t\mathbf{x}}{\partial r}={}^0\mathbf{g}_r+\frac{\partial{}^t\mathbf{u}}{\partial r} \tag{22a}$$

$${}^t\mathbf{g}_s=\frac{\partial{}^t\mathbf{x}}{\partial s}={}^0\mathbf{g}_s+\frac{\partial{}^t\mathbf{u}}{\partial s} \tag{22b}$$

$${}^t\mathbf{g}_t=\frac{\partial{}^t\mathbf{x}}{\partial t}={}^0\mathbf{g}_t+\frac{\partial{}^t\mathbf{u}}{\partial t} \tag{22c}$$

The covariant components of the Green–Lagrange deformation tensor in the configuration at time t, referred to the configuration at $t=0$ and measured in the convected system are[15]

$${}^t_0\tilde{\varepsilon}_{rr}=\tfrac{1}{2}[{}^t\mathbf{g}_r\cdot{}^t\mathbf{g}_r-{}^0\mathbf{g}_r\cdot{}^0\mathbf{g}_r] \tag{23a}$$

$${}^t_0\tilde{\varepsilon}_{rs}=\tfrac{1}{2}[{}^t\mathbf{g}_r\cdot{}^t\mathbf{g}_s-{}^0\mathbf{g}_r\cdot{}^0\mathbf{g}_s] \tag{23b}$$

$${}^t_0\tilde{\varepsilon}_{rt}=\tfrac{1}{2}[{}^t\mathbf{g}_r\cdot{}^t\mathbf{g}_t-{}^0\mathbf{g}_r\cdot{}^0\mathbf{g}_t] \tag{23c}$$

${}^t_0\tilde{\varepsilon}_{tt}$; ${}^t_0\tilde{\varepsilon}_{ss}$ and ${}^t_0\tilde{\varepsilon}_{ts}$ are zero because of our deformation hypotheses.

Using the contravariant base vectors[15] we represent the Green–Lagrange strain tensor as[2–4, 16]

$${}^t_0\boldsymbol{\varepsilon}={}^t_0\tilde{\varepsilon}_{rr}\,{}^0\mathbf{g}^r\,{}^0\mathbf{g}^r+{}^t_0\tilde{\varepsilon}_{rs}[{}^0\mathbf{g}^r\,{}^0\mathbf{g}^s+{}^0\mathbf{g}^s\,{}^0\mathbf{g}^r]+{}^t_0\tilde{\varepsilon}_{rt}[{}^0\mathbf{g}^r\,{}^0\mathbf{g}^t+{}^0\mathbf{g}^t\,{}^0\mathbf{g}^r] \tag{24}$$

Note that, for a beam element with constant (a_k, b_k), the vectors $({}^0\mathbf{g}_r, {}^0\mathbf{g}_s, {}^0\mathbf{g}_t)$, form an orthogonal basis for any point inside the beam.

In the incremental step from t to $t+\Delta t$,

$${}^{t+\Delta t}\mathbf{g}_r={}^t\mathbf{g}_r+\frac{\partial\mathbf{u}}{\partial r} \tag{25a}$$

$${}^{t+\Delta t}\mathbf{g}_s={}^t\mathbf{g}_s+\frac{\partial\mathbf{u}}{\partial s} \tag{25b}$$

$${}^{t+\Delta t}\mathbf{g}_t={}^t\mathbf{g}_t+\frac{\partial\mathbf{u}}{\partial t} \tag{25c}$$

Therefore, using equations (23) and (25),

$$ {}^{t+\Delta t}_{0}\tilde{\varepsilon}_{rr} = {}^{t}_{0}\tilde{\varepsilon}_{rr} + {}^{t}\mathbf{g}_r \cdot \frac{\partial \mathbf{u}}{\partial r} + \frac{1}{2}\frac{\partial \mathbf{u}}{\partial r}\cdot\frac{\partial \mathbf{u}}{\partial r} \tag{26a} $$

$$ {}^{t+\Delta t}_{0}\tilde{\varepsilon}_{rs} = {}^{t}_{0}\tilde{\varepsilon}_{rs} + \frac{1}{2}\left[{}^{t}\mathbf{g}_r \cdot \frac{\partial \mathbf{u}}{\partial s} + {}^{t}\mathbf{g}_s \cdot \frac{\partial \mathbf{u}}{\partial r}\right] + \frac{1}{2}\frac{\partial \mathbf{u}}{\partial r}\cdot\frac{\partial \mathbf{u}}{\partial s} \tag{26b} $$

$$ {}^{t+\Delta t}_{0}\tilde{\varepsilon}_{rt} = {}^{t}_{0}\tilde{\varepsilon}_{rt} + \frac{1}{2}\left[{}^{t}\mathbf{g}_r \cdot \frac{\partial \mathbf{u}}{\partial t} + {}^{t}\mathbf{g}_t \cdot \frac{\partial \mathbf{u}}{\partial r}\right] + \frac{1}{2}\frac{\partial \mathbf{u}}{\partial r}\cdot\frac{\partial \mathbf{u}}{\partial t} \tag{26c} $$

Using equation (19) we rewrite equations (26), keeping only up to the quadratic terms in generalized incremental displacements.

$$ {}_{0}\tilde{\varepsilon}_{rr} = {}^{t+\Delta t}_{0}\tilde{\varepsilon}_{rr} - {}^{t}_{0}\tilde{\varepsilon}_{rr} = {}^{t}\mathbf{g}_r \cdot \frac{\partial \mathbf{u}_S}{\partial r} + \frac{1}{2}\frac{\partial \mathbf{u}_S}{\partial r}\cdot\frac{\partial \mathbf{u}_S}{\partial r} + \underline{{}^{t}\mathbf{g}_r \cdot \frac{\partial \mathbf{u}_{R2}}{\partial r}} \tag{27a} $$

$$ {}_{0}\tilde{\varepsilon}_{rs} = {}^{t+\Delta t}_{0}\tilde{\varepsilon}_{rs} - {}^{t}_{0}\tilde{\varepsilon}_{rs} = \frac{1}{2}\left[{}^{t}\mathbf{g}_r \cdot \frac{\partial \mathbf{u}_S}{\partial s} + {}^{t}\mathbf{g}_s \cdot \frac{\partial \mathbf{u}_S}{\partial r}\right] + \frac{1}{2}\frac{\partial \mathbf{u}_S}{\partial r}\cdot\frac{\partial \mathbf{u}_S}{\partial s} + \underline{\frac{1}{2}\left[{}^{t}\mathbf{g}_r \cdot \frac{\partial \mathbf{u}_{R2}}{\partial s} + {}^{t}\mathbf{g}_s \cdot \frac{\partial \mathbf{u}_{R2}}{\partial r}\right]} \tag{27b} $$

$$ {}_{0}\tilde{\varepsilon}_{rt} = {}^{t+\Delta t}_{0}\tilde{\varepsilon}_{rt} - {}^{t}_{0}\tilde{\varepsilon}_{rt} = \frac{1}{2}\left[{}^{t}\mathbf{g}_r \cdot \frac{\partial \mathbf{u}_S}{\partial t} + {}^{t}\mathbf{g}_t \cdot \frac{\partial \mathbf{u}_S}{\partial r}\right] + \frac{1}{2}\frac{\partial \mathbf{u}_S}{\partial r}\cdot\frac{\partial \mathbf{u}_S}{\partial t} + \underline{\frac{1}{2}\left[{}^{t}\mathbf{g}_r \cdot \frac{\partial \mathbf{u}_{R2}}{\partial t} + {}^{t}\mathbf{g}_t \cdot \frac{\partial \mathbf{u}_{R2}}{\partial r}\right]} \tag{27c} $$

The extra terms with respect to the standard formulation are the underlined ones in equations (27).

We can decompose the strain increment (${}_{o}\tilde{e}_{ij}$) in two parts; one part has all the linear terms in generalized displacements (${}_{0}\tilde{e}_{ij}$), the other part has all the quadratic terms in generalized displacements (${}_{0}\tilde{\eta}_{ij}$).[9] Therefore,

$$ {}_{0}\tilde{\varepsilon}_{ij} = {}_{0}\tilde{e}_{ij} + {}_{0}\tilde{\eta}_{ij} \tag{28} $$

Note that

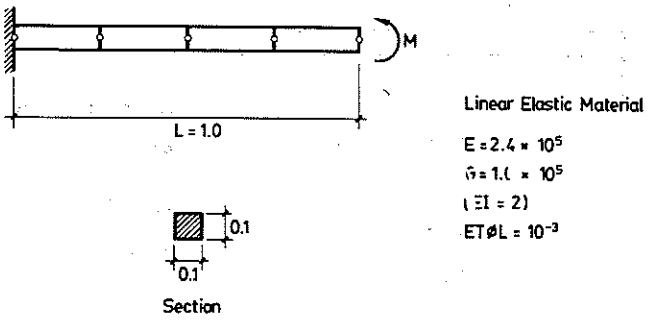
(a) In elements with no rotational degrees of freedom (e.g. 2D and 3D continuum elements) equation (28) represents exactly the total strain increments. In our case equation (28) represents only an approximation to the strain increments, because in the derivation of equation (27) we neglect the terms of order higher than two in generalized displacement increments.[17]

(b) Equations (27) contain all the terms up to the second order in generalized displacement increments. This guarantees a complete quadratic form of the incremental energy, leading therefore to a complete expression of the tangent stiffness matrix.

3. PRINCIPLE OF VIRTUAL WORK

For the equilibrium configuration at time $t+\Delta t$ (the one being sought) the principle of virtual work[9, 18] states

$$ \int_{{}_{0}V} {}^{t+\Delta t}_{0}\tilde{S}^{ij}\,\delta\,{}^{t+\Delta t}_{0}\tilde{\varepsilon}_{ij}\,{}^{0}\mathrm{d}V = {}^{t+\Delta t}\mathscr{R} \tag{29} $$



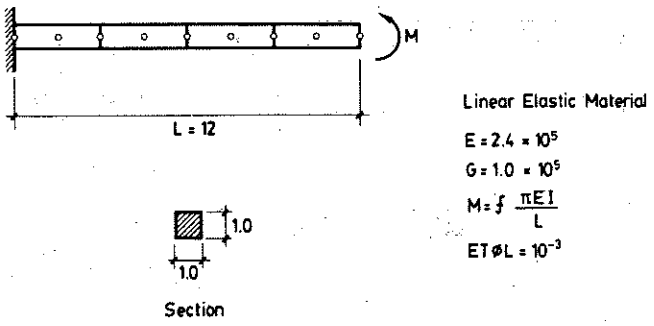
Step	M	Number of Iterations: Standard formulation	Number of Iterations: This work
1	0·2π	3	3
2	0·4π	3	3
3	0·6π	4	3
4	0·8π	5	3
5	1·0π	6	3
6	1·2π	8	3
7	1·4π	10	3
8	1·6π	13	4
9	1·8π	17	4
10	2·0π	23	4
Total Iterations		92	33

Figure 2. Analysis of a cantilever under constant moment using 2-node elements

5.3. Bend (45-degrees) under concentrated load

The curved beam loaded normal to its plane is analysed using five 3-node elements, as shown in Figure 5. The total load of 600·0 is applied in 10 equal steps. The results are compared with those obtained by Bathe and Bolourchi.[23]

In this case the standard formulation and our formulation both use approximately the same number of iterations.



f	Total tip disp. (Surana) [11] / Total tip disp. (Analit·)	Total tip disp. (This work) / Total tip disp. (Analit.)	Iterations (Surana) [11]	Iterations (This work)
0·2	0·99	1·00	7	4
0·4	1·00	1·01	7	5
0·6	1·00	1·01	7	4
0·8	1·00	1·02	7	5
1·0	1·01	1·02	7	4
1·2	1·01	1·01	8	5
1.4	0·99	1·00	8	5
1.6	0·96	0·97	8	6
1.7	0·94	0·94	6	5
1·8	0·91	0·92	7	5

Figure 3. Analysis of a cantilever under constant moment using 3-node elements

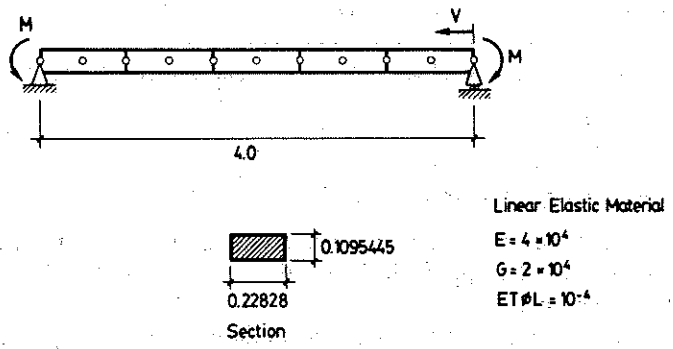
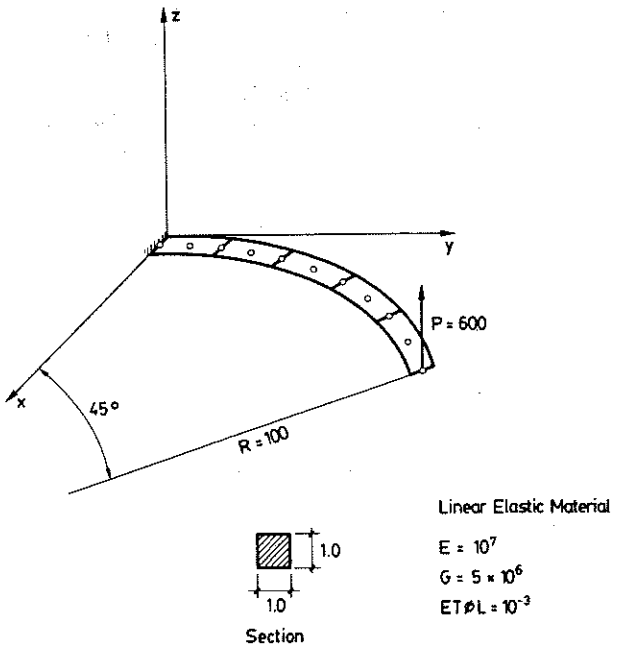


Figure 4

M	Number of Iterations	
	Standard formulation	This work
0·2	4	4
0·4	5	4
0·6	7	5
0·8	9	4
1·0	11	4
1·2	14	5
1·4	18	4
1·6	22	5
1·8	27	4
2·0	34	5
Total Iterations	151	44

M	V(Surana) [11] / V(Analit.)	Iterations (Surana) [11]	V(Oliver) [22] / V(Analit.)	Iterations (Oliver) [22]	V(This work) / V(Analit.)	Iterations (This work)
0·2	1·00	6	1·00	4	1·00	4
0·4	1·00	6	1·00	4	1·00	4
0·6	1·00	6	1·00	4	1·01	5
0·8	1·00	6	1·00	5	1·01	4
1·0	1·00	7	1·01	4	1·01	4
1·2	1·01	7	1·01	4	1·01	5
1·4	1·01	7	1·01	5	1·01	4
1·6	1·01	7	1·02	4	1·02	5
1·8	1·02	7	1·02	5	1·02	4
2·0	1·01	7	1·01	4	1·01	5

Figure 4. Analysis of a simply supported beam under constant moment



	Bathe-Bolourchi [23]	Standard formulation	This work
U/TIP	−13·4	−13·3	−13·6
V/TIP	−23·5	−23·7	−23·5
W/TIP	53·4	53·2	53·3

Total number of iterations for the standard formulation = 35
Total number of iterations for the present formulation = 34

Figure 5. Analysis of a 45-degree bend

5.4. *Closed frame under concentrated loads*

The frame shown in Figure 6 is analysed, modelling one quarter of it with twenty equal 3-node elements. The total load is applied in 5 and 10 equal steps. The displacements corresponding to the total load are compared with the results obtained by Wood.[24]

In this case, again, the standard formulation and the new one both need approximately the same number of iterations to converge.

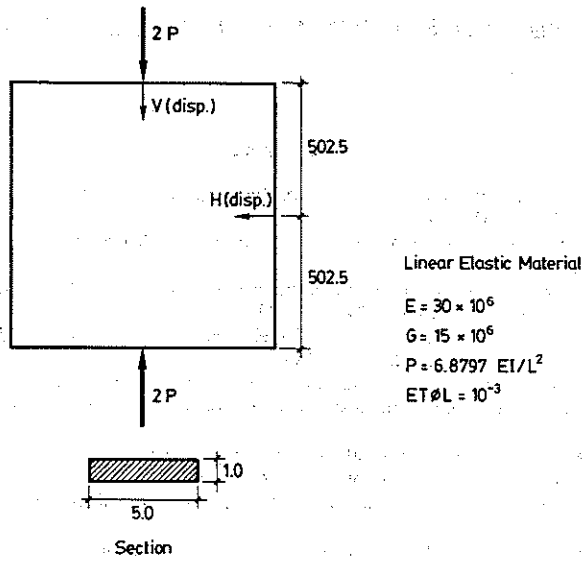
6. CONCLUSIONS

An incremental Total Lagrangian Formulation for curved Timoshenko beam elements that includes the effect of large rotation increments was developed. A symmetric tangent stiffness

matrix was obtained because, as stated above, our nodal variables are three numbers that define a rotation rather than independent rotations around fixed axes. Therefore the commutativity problem which leads to non-symmetric tangent stiffness matrices[25] does not arise. It is important to notice, however, that our nodal variables are coincident with independent rotations around fixed axes in the case of infinitesimal rotations.

The advantages of the new formulation over the standard formulation, that assumes infinitesimal rotation increments in the tangent stiffness matrix derivation, are as follows.

(a) The tangent stiffness matrix obtained with the new formulation derives from a complete linearization of the virtual work equations, therefore it does not introduce errors in stability analyses (linearized buckling analyses.[26])



Number of steps	Total number of iterations	
	Standard formulation	This work
5	23	24
10	35	32

	Wood [24]	This work	
		5 steps	10 steps
V	300·70	299·96	302·73
H	−241·00	−235·19	−240·23

Figure 6. Analysis of a closed frame

(b) The complete tangent stiffness matrix assures quadratic convergence in the displacements norm when using full Newton–Raphson iterations,[27] therefore, in many cases when using the new formulation convergence is achieved with fewer iterations than when using the standard formulation. It is well known, however, that this should not always be the case, because in some problems iterating with a 'non-exact' tangent matrix may lead to a faster convergence. Also, some type of secant formulation[14] could be developed to improve the efficiency of the analyses.

The same kind of formulation we presented for beam elements can be developed for C^0 shell elements.

ACKNOWLEDGEMENT

E. N. Dvorkin acknowledges Prof. K. J. Bathe from M.I.T. for many important discussions on this topic.

REFERENCES

1. S. Ahmad, B. M. Irons and O. C. Zienkiewicz, 'Analysis of thick and thin shell structures by curved finite elements', *Int. j. numer. methods eng.*, **2**, 419–451 (1970).
2. E. N. Dvorkin and K. J. Bathe, 'A continuum mechanics based four-node shell element for general nonlinear analysis', *Eng. Comp.*, **1**, 77–88 (1984).
3. K. J. Bathe and E. N. Dvorkin, 'A four-node plate bending element based on Mindlin/Reissner plate theory and a mixed interpolation', *Int. j. numer. methods eng.*, **21**, 367–383 (1985).
4. K. J. Bathe and E. N. Dvorkin, 'A formulation of general shell elements—the use of mixed interpolation of tensorial components', *Int. j. numer. methods eng.*, **22**, 697–722 (1986).
5. J. Oliver and E. Oñate, 'A total Lagrangian formulation for the geometrically nonlinear analysis of structures using finite elements. Part I. Two-dimensional problems: shell and plate structures', *Int. j. numer. methods eng.*, **20**, 2253–2281 (1984).
6. J. Oliver and E. Oñate, 'A total Lagrangian formulation for the geometrically nonlinear analysis of structures using finite elements. Part II: arches, frames and axisymmetric shells', *Int. j. numer. methods eng.*, **23**, 253–274 (1986).
7. K. C. Park and G. M. Stanley, 'A curved C^0 shell element based on assumed natural-coordinate strains', *J. Appl. Mech. ASME*, **53**, 278–290 (1986).
8. E. Hinton and H. C. Huang, 'A family of quadrilateral Mindlin plate elements with substitute shear strain fields', *Comp. Struct.*, **23**, 409–431 (1986).
9. K. J. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
10. J. Argyris, 'An excursion into large rotations', *Comp. Methods Appl. Mech. Eng.*, **32**, 85–155 (1982).
11. K. S. Surana, 'Geometrically nonlinear formulation for the curved shell elements', *Int. j. numer. methods eng.*, **19**, 581–615 (1983).
12. J. C. Simo, 'A finite strain beam formulation. The three-dimensional dynamic problem. Part I', *Comp. Methods Appl. Mech. Eng.*, **49**, 55–70 (1985).
13. J. C. Simo and L. Vu Quoc, 'A three dimensional finite strain rod model. Part II: computational aspects', *Comp. Methods Appl. Mech. Eng.*, **58**, 79–116 (1986).
14. E. Oñate, 'Una formulación incremental para análisis de problemas de no linealidad geométrica por el método de los elementos finitos' (in Spanish), Internal Report E.T.S. Ingenieros de Caminos, Universidad Politécnica de Catalunya, Spain, 1986.
15. A. E. Green and W. Zerna, *Theoretical Elasticity*, 2nd edn, Oxford University Press, 1968.
16. L. E. Malvern, *Introduction to the Mechanics of a Continuous Medium*, Prentice-Hall, Englewood Cliffs, New Jersey, 1969.
17. K. J. Bathe, 'Finite element procedures for solids and structures—Nonlinear analysis', *Video Course Study Guide*, M.I.T. Center for Advanced Engineering Study, 1986.
18. K. Washizu, *Variational Methods in Elasticity and Plasticity*, 3rd edn, Pergamon Press, London, 1982.
19. K. J. Bathe, E. N. Dvorkin and L. W. Ho, 'Our discrete-Kirchhoff and isoparametric shell elements for nonlinear analysis—an assessment', *Comp. Struct.*, **16**, 89–98 (1983).
20. A. K. Noor and J. M. Peters, 'Mixed models and reduced/selective integration displacement models for nonlinear analysis of curved beams', *Int. j. numer. methods eng.*, **17**, 615–631 (1981).
21. K. J. Bathe and A. P. Cimento, 'Some practical procedures for the solution of nonlinear finite element equations', *Comp. Methods Appl. Mech. Eng.*, **22**, 59–85 (1980).

22. J. Oliver, 'Una formulación cuasi-intrínseca para el estudio, por el métodó de lós elementos finitos, de vigas, arcos, placas y láminas sometidas a grandes corrimientos en régimen elastoplástico' (in Spanish), *Doctoral Thesis*, E.T.S. Ingenieros de Caminos, Universidad Politécnica de Catalunya, Spain, 1982.
23. K. J. Bathe and S. Bolourchi, 'Large displacement analysis of three-dimensional beam structures', *Int. j. numer. methods eng.*, **14**, 961–986 (1979).
24. R. D. Wood, 'The application of finite element methods for geometrically nonlinear structural analysis', *Ph.D. Thesis*, University College of Swansea, Wales, U.K., 1973.
25. J. H. Argyris, H. Balmer, J. St. Doltsinis, P. C. Dunne, M. Haase, M. Kleiber, G. A. Malejannakis, H. P. Mlejnek, M. Müller and D. W. Scharpf, 'Finite element method—the natural approach', *Comp. Methods Appl. Mech. Eng.*, **17/18**, 1–106 (1979).
26. K. J. Bathe and E. N. Dvorkin, 'On the automatic solution of nonlinear finite element equations', *Comp. Struct.*, **17**, 871–879 (1983).
27. E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, Wiley, New York, 1966.

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 39, 3839–3866 (1996)

A FINITE POINT METHOD IN COMPUTATIONAL MECHANICS. APPLICATIONS TO CONVECTIVE TRANSPORT AND FLUID FLOW

E. OÑATE, S. IDELSOHN*, O. C. ZIENKIEWICZ† AND R. L. TAYLOR‡

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

SUMMARY

The paper presents a fully meshless procedure for solving partial differential equations. The approach termed generically the 'finite point method' is based on a weighted least square interpolation of point data and point collocation for evaluating the approximation integrals. Some examples showing the accuracy of the method for solution of adjoint and non-self adjoint equations typical of convective-diffusive transport and also to the analysis of a compressible fluid mechanics problem are presented.

KEY WORDS: least squares; finite point method; mesh free techniques

1. INTRODUCTION

The Finite Element Method (FEM)[1] and its subclass the Finite Volume Method (FVM)[2–4] are well-established numerical techniques in computational mechanics (though the latter is not used widely in solid mechanics) 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 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 robust and efficient 3D mesh generators makes the solution of 3D problems a more 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 providing an acceptable mesh is available. The generation of 3D meshes, however, is despite major recent advances in this field, certainly the bottle neck in most industrial FE and FV computations and, in many cases, it can absorb far more time and cost than the numerical solution itself.

* Professor at Universidad del Litoral, Santa Fe, Argentina. Visiting Professor, CIMNE
† Professor of Civil Engineering, University College, Swansea, UK. UNESCO Professor, UPC
‡ Professor of Civil Engineering, University of California, Berkeley. Visiting Professor, CIMNE

CCC 0029–5981/96/223839–28 *Received 18 December 1995*

Considerable effort has been devoted during recent years to the development of the 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] Similar procedure was developed for thin plate problems solved by a variational approach by Nay and Utku[13] using a least-squares fit to nodal values and contributing domain concepts for deriving the approximation integrals.

An alternative class of methods named Smooth Particle Hydrodynamics (SPH), sometimes called the Free Lagrange methods, depend only on a set of disordered point or particles and has enjoyed considerable popularity in computational physics and astrophysics to model the motion and collision of stars.[14–17] 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 collection of nodes and a boundary description is needed to formulate the Galerkin equations. The interpolating functions 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 Reference 19 in order to compute numerically 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. This provides additional terms in the derivatives of the interpolant considered unnecessary by Nayroles *et al.*[19] In addition, a regular cell structure is chosen as the 'auxiliary grid' to compute the integrals by means of high-rder quadratures. Duarte and Oden[22] and Babuska and Melenk[23] have recently formalized this type of approximation as a subclass of the so-called 'Partition of Unity' (PU) methods and they propose meshless and enhanced FE procedures using hierarchical PU interpolations.

Liu *et al.*[18,24–26] 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 one to develop a new type of shape functions using an integral window transform. The window function can be *translated* and *dilated* around the domain thus replacing the need to define elements and providing refinement. Comparative study of RKP, SPH, DE and EFG methods can be found in Reference 27.

The objective of this paper is two fold. First, a brief overview of some point data interpolation-based procedures termed here generically 'finite point methods'[28] is presented. Then, a particularly simple finite point method based on

(a) weighted least-squares interpolation using a 'fixed' Gaussian weighting function, and
(b) point collocation for evaluating the approximation integrals

is proposed. Some examples showing the accuracy of the method for solution of adjoint and non-self adjoint equations typical of convective–diffusive transport and also to the analysis of a compressible fluid mechanics problem are then presented.

2. BASIC CONCEPTS OF MESH FREE TECHNIQUES

Let us assume a scalar problem governed by a differential equation

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

with boundary conditions

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

to be satisfied in a domain Ω with boundary $\Gamma = \Gamma_t \cup \Gamma_u$. In the 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 approximation $\hat{u}$ and equations (1) and (2) are replaced by[1]

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

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) and indeed so can all the meshes approximation procedures.

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

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

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

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

Here 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 are divided into elements and the N_i function may have some discontinuities (in the function itself or in its derivatives) on the element interfaces. In the FE method the weighting functions W_i are defined in 'weighting domains' which usually coincide precisely 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[2-4] (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 equation (3).

In meshless (or element free) methods the subdomains Ω_i are frequently termed interpolation domains or 'clouds'.[2,2]

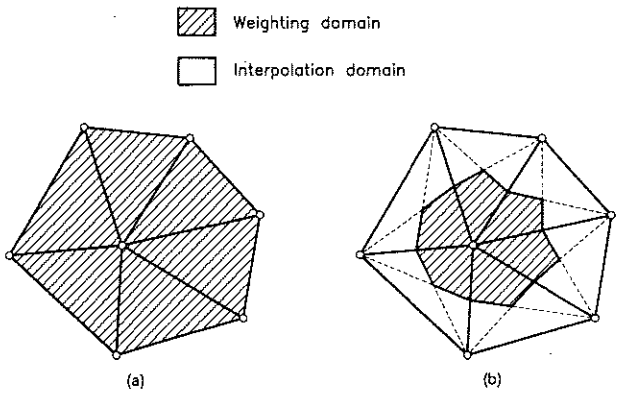


Figure 1. Interpolation and weighting domains in finite element (FE) and finite volume (FV) methods: (a) FEM and FVM (cell vertex); (b) FVM (cell centered)

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

I. (a) The discretization of the unknown function and its derivatives must be defined *only* by the position of points located within the analysis domain and parameters specified at these.
(b) The weighting 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.

Condition II(a) is satisfied by finite difference and point collocation procedures. Other methods like subdomain collocation, or meshless methods which are based on the use of auxiliary background grids for integration purposes (i.e. DEM,[19] EFG,[20] RPK[18]) satisfy condition II(b).

Condition I(a) is satisfied by classical use of Rayleigh–Ritz method in which the N_i functions are defined over the whole domain Ω, thus precluding the local character of the approximation. FE methods define the shape functions N_i over non-overlapping regions (elements) the assembly of which constitutes the domain Ω_i.[2–4] 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(a). 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 methods') are briefly reviewed.

3. LEAST-SQUARES, DIFFUSE LEAST-SQUARES, MOVING LEAST-SQUARES AND REPRODUCING PARTICLE KERNEL APPROXIMATIONS

Let Ω_k be the interpolation domain (cloud) of a function $u(x)$ and let s_j with $j = 1, 2, \ldots, n$ be a collection of n points with coordinates $x_j \in \Omega_k$. The unknown function u may be approximated within Ω_k by

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

where $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_m]^T$ and vector $\mathbf{p}(x)$ contains typically monomials, hereafter termed 'base interpolating functions', in the space co-ordinates ensuring that the basis is complete. For a 2D problem we can specify

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

and

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

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

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

where $u_j^h = u(x_j)$ are the unknown but sought for values of function u at point j, $\hat{u}_j = \hat{u}(x_j)$ are the approximate values, and $\mathbf{p}_j = \mathbf{p}(x_j)$.

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* $\mathbf{C}\alpha$ in (8)

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

and

$$u \cong \hat{u} = \mathbf{p}^T\mathbf{C}^{-1}\mathbf{u}^h = \mathbf{N}^T\mathbf{u}^h = \sum_{j=1}^{n} N_j u_j^h \tag{10}$$

with

$$\mathbf{N}^T = [N_1, \ldots, N_n] = \mathbf{p}^T\mathbf{C}^{-1} \text{ and } N_j = \sum_{i=1}^{m} p_i(x) C_{ij}^{-1} \tag{11}$$

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

$$\begin{aligned} N_j(x) &= 1 \quad j = i \\ &= 0 \quad j \neq i, \quad i, j = 1, \ldots, n \end{aligned} \tag{12}$$

The development of the N_j can often be performed directly using interpolation methods and/or isoparametric concepts.[1]

3.1. *Least Squares (LSQ) approximation*

If $n > m$, $\mathbf{C}$ is no longer a square matrix and the approximation cannot fit all the u_j^h values. This problem can simply be overcome by determining the $\hat{u}$ values by minimizing the sum of the square distances of the error at each point

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

with respect to the α_i parameters (least-squares fit).

Standard minimization gives

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

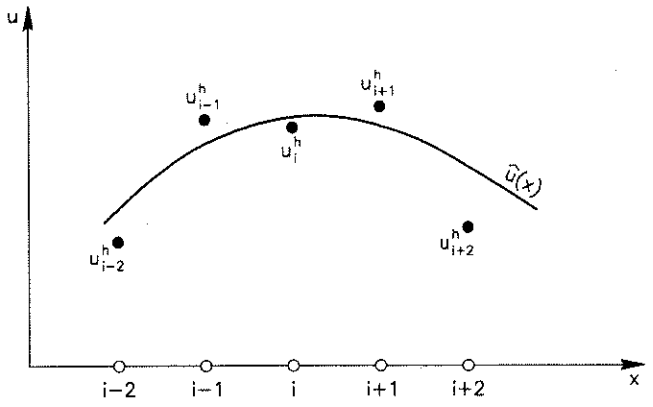


Figure 2. Unknown function $\hat{u}(x)$ and unknown parameters u_i^h

where

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

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

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

$$N_j^k = \sum_{i=1}^{m} p_i(x)\bar{C}_{ij}^{-1} \tag{16}$$

where superindex k emphasizes that the shape function N_j^k can now be defined differently for each cloud Ω_k.

It must be noted that accordingly to the least-squares character of the approximation

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

i.e. the local values of the approximating function do not fit the nodal unknown values (Figure 2). Indeed $\hat{u}$ is the true approximation for which we shall seek the satisfaction of the differential equation and boundary conditions and u_j^h are simply the unknown parameters sought!

However, if $n = m$ the FEM-type approximation is recovered. Then $\hat{u}(x_j) = u^h(x_j)$ and once again conditions (12) are satisfied.

Note again that according to (6), the approximate function $\hat{u}(x)$ is defined in each interpolation domain Ω_k. In fact, different interpolation domains can yield different shape functions N_j^k. As a consequence a point belonging to two or more overlapping interpolation domains has different values of the shape functions. The interpolation is now multivalued within Ω_k and, therefore, for any useful approximation a decision must be taken limiting the choice to a single value.

Indeed, the approximate function $\hat{u}(x)$ will be typically used to provide the value of the unknown function $u(x)$ and its derivatives in only specific regions within each interpolation domain. For instance by using point collocation we may limit the validity of the interpolation to a single point i.

LSQ approximation has enjoyed some popularity in deriving point data interpolations for numerical computation in solid and fluid mechanics. Nay and Utku[13] used quadratic LSQ interpolations to fit the deflection field giving constant curvature for thin plate bending analysis.

More recently, Batina[29] has used LSQ fits based on linear polynomial 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 later section. This deficiency can be overcome by using a weighted least-squares interpolation as described next.

3.2. *Weighted least-squares (WLS) approximation*

The LSQ approximation can be enhanced in a region in vicinity of point i where, for instance, the derivatives of the unknown are to be evaluated, by weighting the squared distances with a function φ_i so that, we minimize

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

Function $\varphi_i(x - x_i)$ is usually built in such a way that it takes a unit value in the vicinity of the point i (typically called 'star node'[12]) where the function (or its derivatives) are to be computed and vanishes outside a region Ω_i surrounding the point (Figure 3). The region Ω_i can be used to define the number of sampling points n in the interpolation region (i.e. $\Omega_i = \Omega_k$ is this case). A typical choice for $\varphi_i(x - x_i)$ is the normalized Gaussian function. Of course, $n \geqslant m$ is always required in the sampling region and if equality occurs no effect of weighting is present and the interpolation is the same as in the LSQ scheme.

Standard minimization of equation (18) with respect to α_i gives the same form of the shape functions as defined in equations (14)–(16) but with matrices $\mathbf{A}$ and $\mathbf{B}$ given now by

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

$$\mathbf{B} = [\varphi_i(x_1 - x_i)\mathbf{p}(x_1), \varphi_i(x_2 - x_i)\mathbf{p}(x_2), \ldots, \varphi_i(x_n - x_i)\mathbf{p}(x_n)] \tag{19b}$$

Note that also again a multivalued interpolation is obtained in this case and some definition of Ω_k is required as in the LSQ approach.

3.3. *Moving Least-Squares (MLS) approximation*

In the Moving Least-Squares (MLS) approach the weighting function φ is defined in shape and size and is translated over the domain so that it takes the maximum value over the point k identified by the co-ordinate x_k where the unknown function $\hat{u}$ is to be evaluated.

As shown in Figure 3 we now minimize for every point k the following functional:

$$J = \sum_{j=1}^{n} \varphi_k(x_j - x_k)(u_j^h - \mathbf{p}_j^{\mathrm{T}}\alpha) \tag{20}$$

where φ_k can in general change its shape and span depending on the position of point k. Note that x_k is now an arbitrary coordinate position and it can be simply replaced by the global co-ordinate x. We will however retain the form $\varphi_k(x_j - x_k)$ to emphazise the possibility of changing function φ at each position within the approximation domain. In the simplest case of constant grid spacing it is possible to take

$$\varphi_k(x_j - x_k) = \varphi(x_j - x) \tag{21}$$

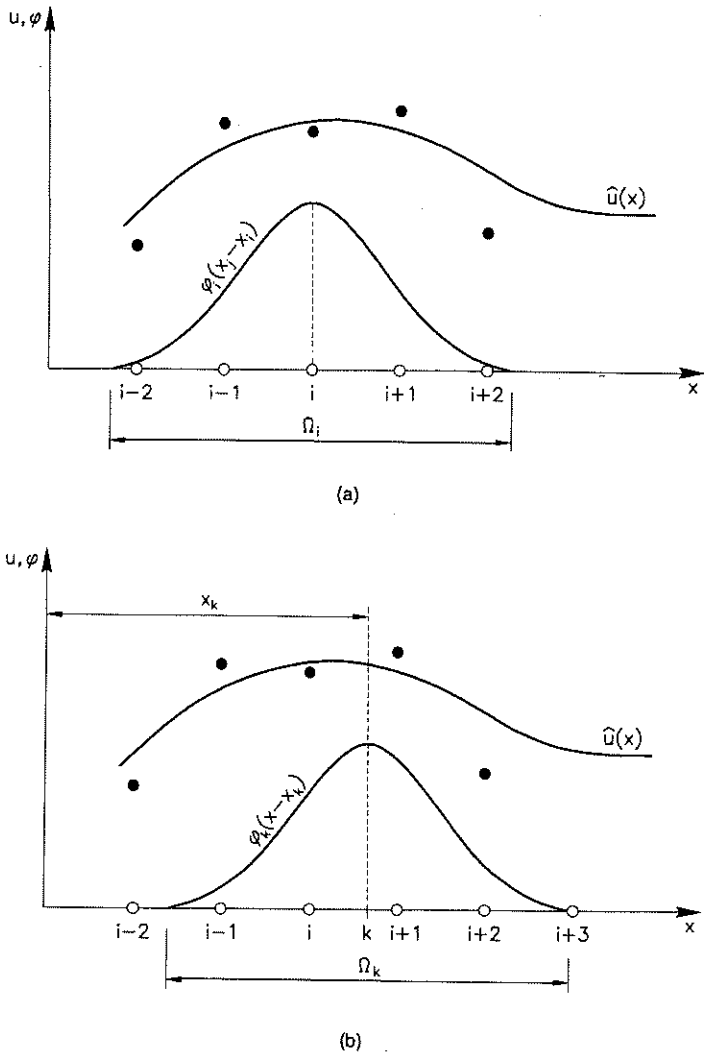


Figure 3. Fixed and moving weighting functions: (a) fixed φ corresponding to point x_i; (b) moving φ

and assume the shape of the weighting function *invariant*. It is precisely for this case that the name MLS is derived.

In general, with an arbitrary position of points the problem of specifying φ_k at every position x is very difficult and presents an infinite number of possibilities. It is convenient instead to specify only the weighting function at the finite number of points chosen and derive similar to the WLS, weightings of the type $\varphi_i(x_j - x_i)$. With this definition, a possible and computationally useful procedure is to define

$$\varphi_k(x_j - x_k) = \varphi_j(x_k - x_j) \tag{22}$$

Note again that k denotes an arbitrary location (i.e. $x_k = x$) whereas j is a fixed point in the domain. With this assumption the function to be minimized in place of that of equation (20) is

now the following:

$$J(x) = \sum_{j=1}^{n} \varphi_j(x - x_j)[u_j^h - \mathbf{p}_j^{\mathrm{T}}\boldsymbol{\alpha}(x)] \tag{23}$$

This gives immediately

$$\mathbf{A} = \sum_{j=1}^{n} \varphi_j(x - x_j)\mathbf{p}(x_j)\mathbf{p}^{\mathrm{T}}(x_j) \tag{24a}$$

$$\mathbf{B} = [\varphi_1(x - x_1)\mathbf{p}(x_1), \varphi_2(x - x_2)\mathbf{p}(x_2), \ldots, \varphi_n(x - x_n)\mathbf{p}(x_n)] \tag{24b}$$

Observe that in this case the parameters α_i are no longer constants but vary continuously with position x and that inversion of matrices is required at every point where $\hat{u}$ is to be evaluated.

Furthermore, a unique global definition of the shape functions can be now obtained provided:

(a) the weighting function φ_i is continuous and differentiable in Ω_i,
(b) the weighting function φ_i vanishes on the boundary of Ω_i and outside,
(c) the number of points n within Ω_i is equal or greater than the parameters m at all points in Ω_i.

This method has been successfully used by Nayroles *et al.*[19] with the name of Diffuse Finite Element (DFE) method and later by Belytschko *et al.*[20] in the context of the so-called Element Free Galerkin (EFG) method. In the DFE method reported in Reference 19, the derivatives of φ_i where omitted when computing the derivatives of the unknown function, whereas the full expression was used in the EFG method. In order to avoid 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 though the computational advantage seems to be small. A generalization of the MLS approach using the concept of partition of unity functions can be found in Reference 22. Indeed that paper introduces a very simple possibility of increasing the polynomial order of the shape functions hierarchically as illustrated in Reference 37.

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 exploited by Liu *et al.*[18,24–26] under the name of 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

$$f(x) = \int_{\Omega} \phi(x - y) f(y)\, \mathrm{d}y \tag{25}$$

where $\phi(x)$ is an appropriate window function defined with a compact support and which can be *translated* around the domain. A *dilation* parameter is also used to provide *refinement*. The parameters α_i in (6) can be obtained by applying the integral window transform to equation (6) and after discretization an approximation identical to equation (4) can be obtained.[18,28] A comparison of RPK, SPH, MLS and WLS methods can be found in References 18 and 27.

3.5. Comparison of FE, LSQ, WLS and MLS approximations

Independently of the method chosen to solve the global problem (equation (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, WLS and MLS methods is that in the FE approach the local value of the approximating function is coincident with the unknown parameters, i.e.

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

whereas in finite point-based methods (FPM) such as LSQ, WLS and MLS methods

$$\hat{u}(x_i) \neq u_i^h \tag{27}$$

Summarizing the 'freedom' for choosing the approximation in the different methods considered we can list

(a) In FE methods
  (1) the mesh,
  (2) the polynomial order (m in equation (6))
(b) In Finite Point Methods (FPM)
  (1) the point position (in LSQ, WLS and MLS methods),
  (2) the polynomial order (in LSQ, WLS and MLS methods),
  (3) the shape of the weighted least square function $\varphi_i(x)$ (in WLS and MLS methods),
  (4) the span of $\varphi_i(x)$ (in WLS and MLS methods), and
  (5) the choice of fixed or moving least-squares weighting functions $\varphi_i(x)$ (in WLS and MLS methods).

To understand better these differences we will plot some of the shape functions resulting from these methods in one dimension.

We choose for simplicity a collection of equally spaced points at a distance h from each other. Figure 4 shows the weighting least-squares function $\varphi_i(x)$ used for WLS and MLS methods given by the Gaussian expression

$$\varphi_i(x) = \frac{e^{-(x/c)^2} - e^{-(xm/c)^2}}{1 - e^{-(xm/c)^2}} \tag{28}$$

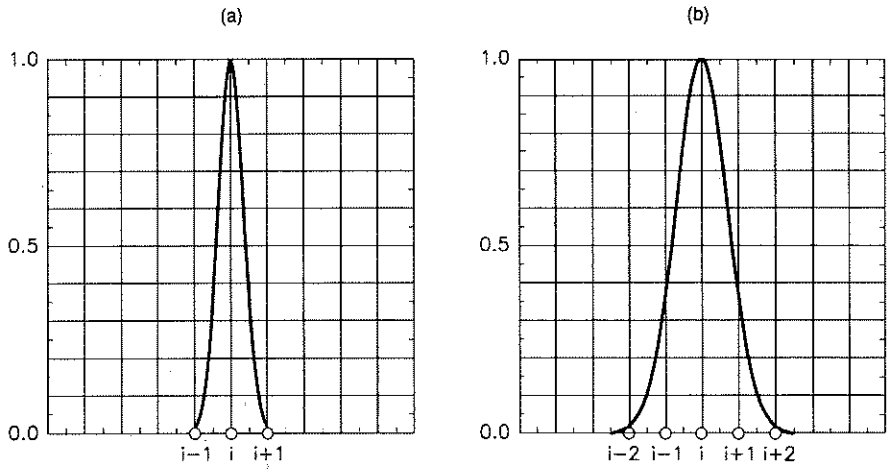


Figure 4. Gaussian weighting functions for three and five points clouds: (a) $n = 3$; (b) $n = 5$

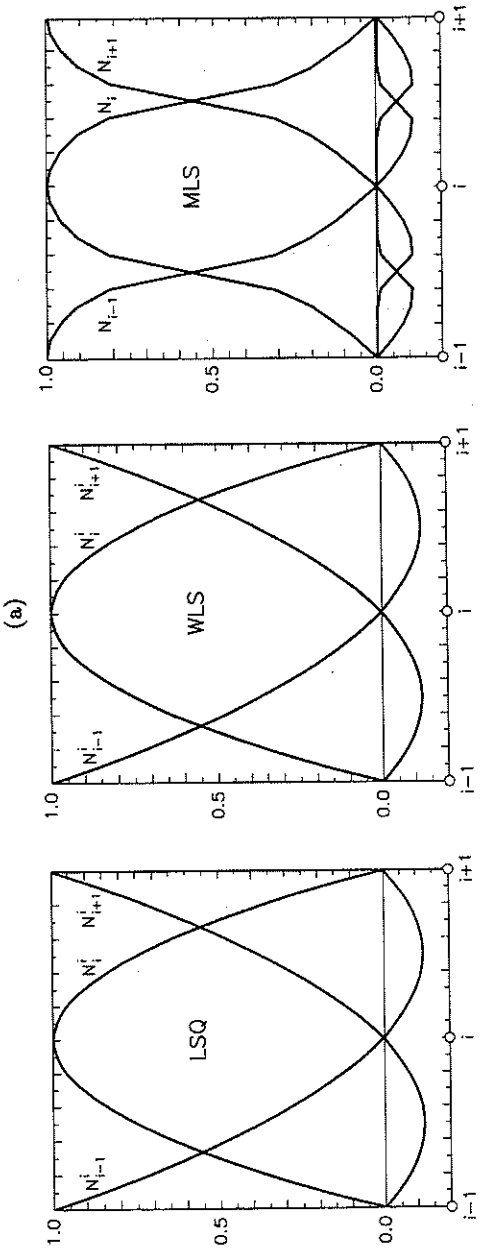


Figure 5. Shape functions $N_i(x)$ for $\mathbf{p} = [1, x, x^2]$: (a) three-points clouds ($m = 3$) ($n = 3$); (b) five-points clouds ($m = 3$) ($n = 5$). $\Omega_k = \Omega_i$ is the domain from $i - 1$ to $i + 1$ in (a) and $i - 2$ to $i + 2$ in (b)

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

The following weightings will be considered:

(a) $x_m = 1{\cdot}6h$. This defines three-point clouds ($n = 3$) for both LSQ and WLS methods. It also defines a *minimum* number of three-point clouds for the MLS method, although four points occur for some particular positions. For consistency the name *three-point cloud* will be given to this option in all cases.

(b) $x_m = 2{\cdot}5h$. This generally defines a five point cloud ($n = 5$). Note that now four points may occur for one position in the MLS method.

Figure 5 shows the shape functions for a quadratic base function polynomial ($m = 3$) and clouds of three and five points, respectively. Note that with three-points clouds LSQ and WLS methods yield interpolating functions similar to the standard FE quadratic shape functions as expected. The MLS method gives global interpolating functions of higher order than quadratic. When $n = 5$ (five-points cloud) none of the methods gives $N_i(x) = 1$ at $x = x_i$ nor is $N_i(x) = 0$ at $x = x_j, j \neq i$. In particular, the LSQ method yields very inaccurate interpolating functions in this case with the value $N_i(x_i)$ equal to less than 0·5 everywhere.

Note that, as previously mentioned, the global interpolation is multivalued in the LSQ and WLS approaches. This is illustrated in Figure 6(a) where the global expressions of the shape functions are plotted for the case $m = 3, n = 3$. A mid-point rule has been chosen in Figure 6(b) to preserve a single value of the shape function in the vicinity of each point. Note also that the first derivative is undeterminate at $x = (x_i + x_{i-1})/2$ which limits the validity of LSQ and WLS methods to point or subdomain collocation procedures.[33,34]

Similar results can be obtained with linear base functions ($m = 2$) as shown in Figure 7. Here again, the shape functions provided by the LSQ method attenuate when the number of points within the cloud n is changed from 3 to 5. This attenuation is also noted in WLS and MLS methods where the $N_i(x_i)$ value decreases from 0·97 to 0·57.

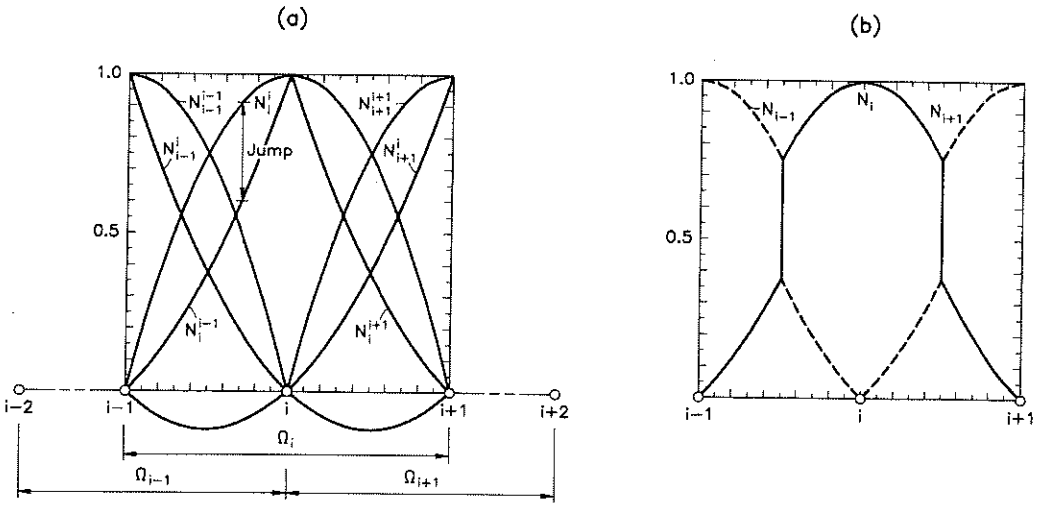


Figure 6. (a) Multivalue definition of the shape function in LSQ and WLS methods for $m = 3$, $n = 3$, and (b) global definition of the interpolation using a mid-point rule

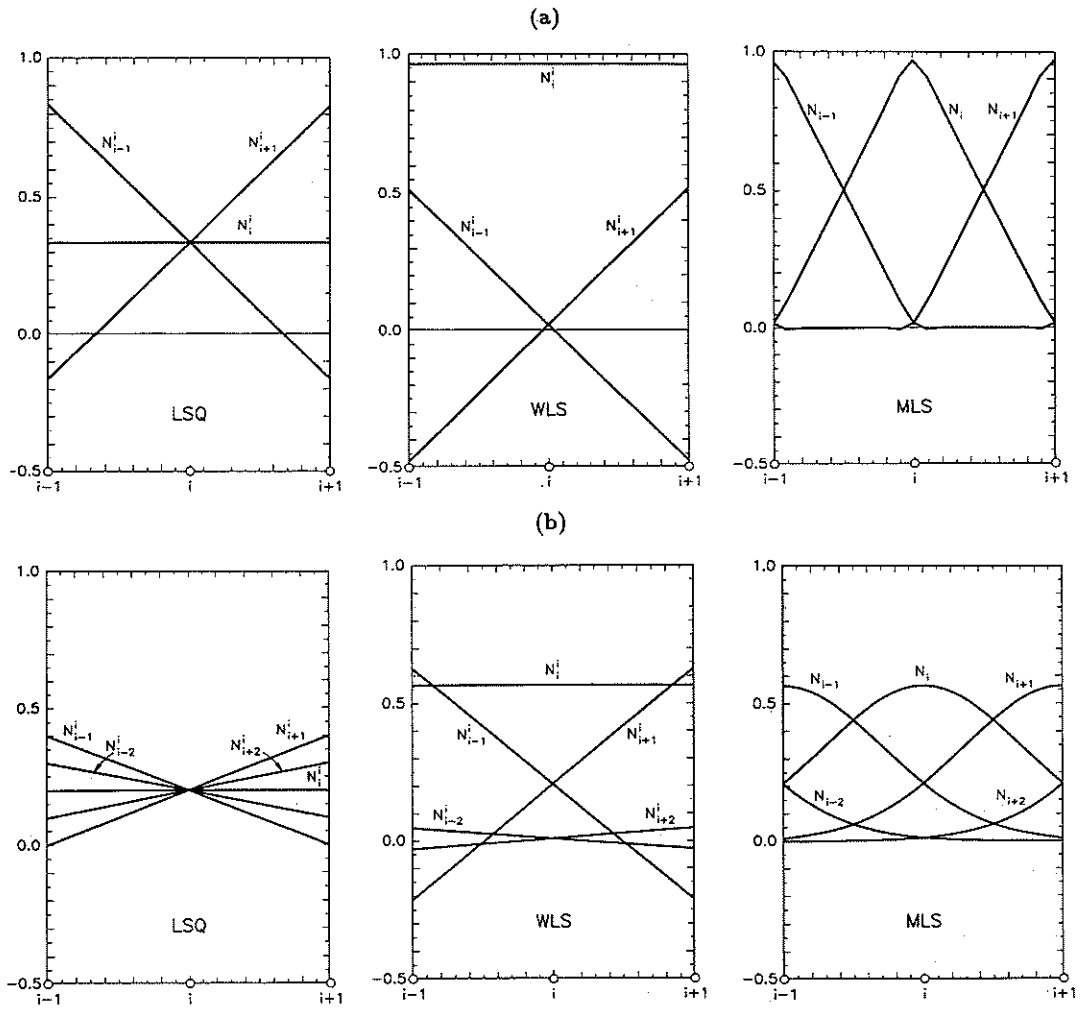


Figure 7. Shape functions $N_i(x)$ for $\mathbf{p} = [1, x]$; (a) three-points clouds ($m = 2$) ($n = 3$); (b) five-points clouds ($m = 2$) ($n = 5$). $\Omega_k = \Omega_i$ is the domain from $i - 1$ to $i + 1$ in (a) and $i - 2$ to $i + 2$ in (b)

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 WLS approximations with linear base polynomials seem to also be quite sensitive to the number of cloud points. Conversely, the MLS and WLS methods with quadratic base functions seem to be best suited for point data interpolation using blendings of approximations based on three and five points for each cloud (see Table I).

Table I. Sensitivity of the quality of the approximation when the number of cloud points n is changed from 3 to 5

Method	Order of the base interpolating polynomial (m)	Sensitivity of the approximation
LSQ	Linear ($m = 2$)	Very large
LSQ	Quadratic ($m = 3$)	Very large
WLS	Linear ($m = 2$)	Large
WLS	Quadratic ($m = 3$)	Low
MLS	Linear ($m = 2$)	Large
MLS	Quadratic ($m = 3$)	Low

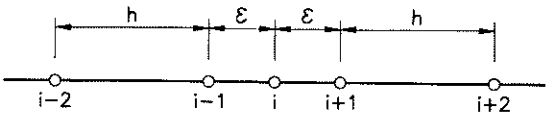


Figure 8. Point distribution with variable distance

Another important property of a good Finite Point Method (FPM) is the possibility to introduce (or to translate) new points independently of the distance between existing points. In the FE method, two nodes that are very close together, generate a gradient in the shape function which can introduce 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 8 shows a set of points with two points placed at a variable distance ε from a central point i. Figure 9 shows the shape functions when this distance ε tends to zero for two different cases: (a) WLS interpolation with quadratic base functions ($m = 3$) and (b) MLS interpolations function with a linear base function ($m = 2$). Again a truncated Gaussian weighting (equation (28)) was chosen in both cases. For $h/\varepsilon = 2$ the shape functions in the WLS approach are coincident with the quadratic FE shape functions. Alternatively, when the three points $i - 1$, i and i tend to coincide, the three corresponding shape functions tend to take the same expression in both WLS and MLS cases, thus avoiding the sharp gradients typical in FE approximation.

4. DERIVATION OF THE DISCRETIZED EQUATIONS

The selection of different weighting functions in the general weighted residual form of equation (3) yields different sets of discretized equations. In order to preserve the mesh-free character of the method, the weighting function must satisfy Condition II(b) of Section 2 and the weighting domain must be defined independently of any mesh. All approximation methods of integral type (i.e. Galerkin, area collocation, etc.) are costly and necessitate the introduction of complex procedures for integration (i.e. background grid, etc.[18–21]). Some of these procedures are reviewed in Reference 28. In this paper we shall therefore limit the choice to *point collocation* methods where we feel the advantages of Finite Point Methods are best realized.

Such point collocation has recently been used with success by Batina,[29] who however limited his work to the use of LSQ and linear approximation. We shall show that considerable improvement can be gained using WLS or MLS procedures.

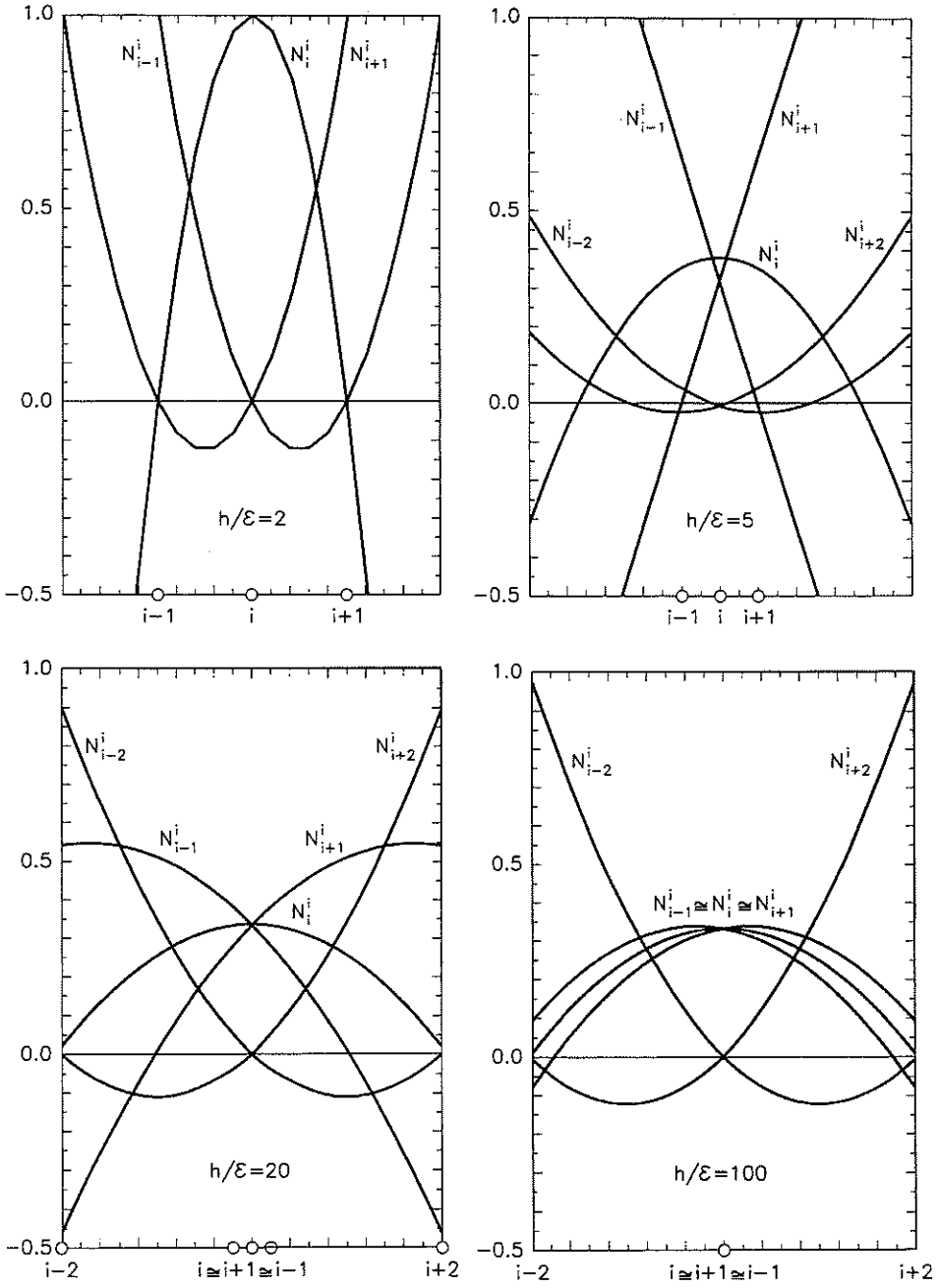


Figure 9a. Shape functions for 3 very close points, WLS method ($m = 3$)

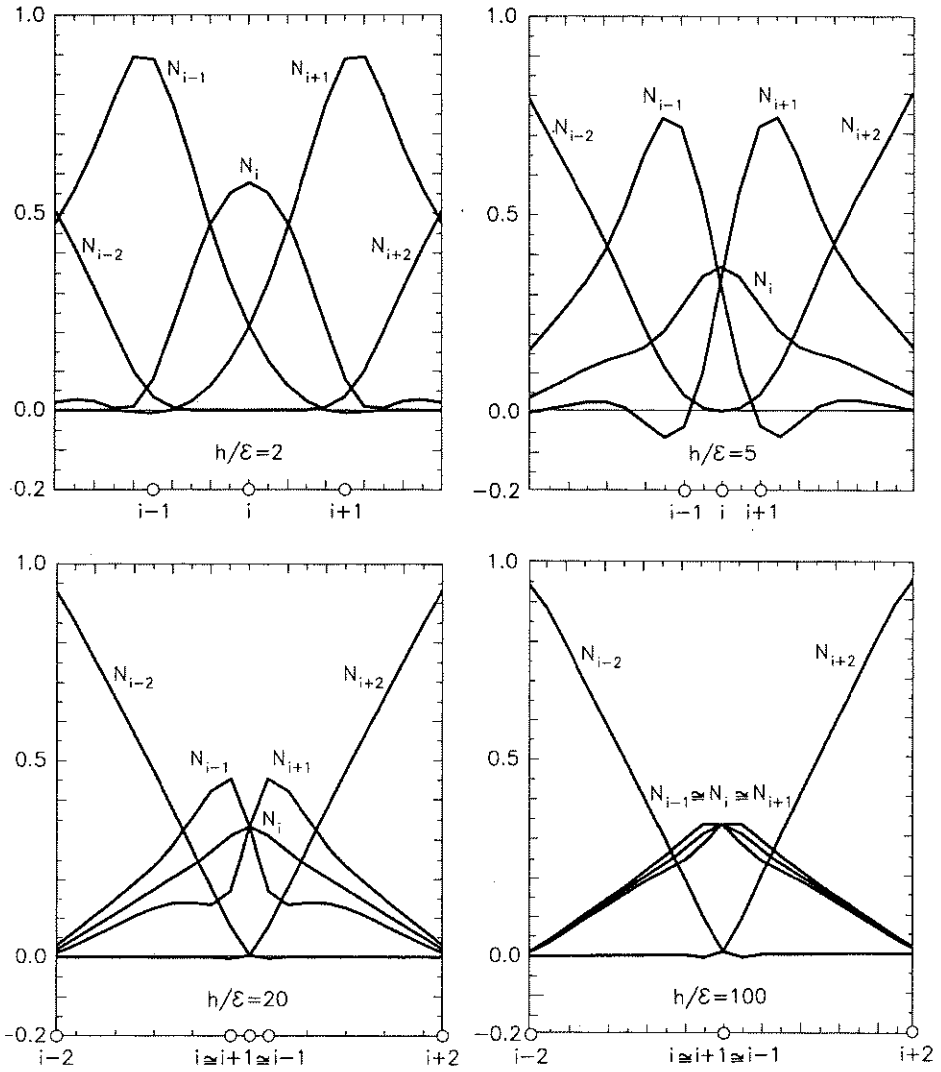


Figure 9b. Shape functions for 3 very close points, MLS method ($m = 2$)

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

$$[A(\hat{u})]_i - b_i = 0 \quad \text{in } \Omega \tag{29}$$

$$[B(\hat{u})]_i - t_i = 0 \quad \text{in } \Gamma_t \tag{30}$$

$$\hat{u}_i - u_p = 0 \quad \text{in } \Gamma_u \tag{31}$$

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

$$\mathbf{K}\mathbf{u}^h = \mathbf{f} \tag{32}$$

with $K_{ij} = [A(N_j)]_i + B(N_j)]_i$ and where the symmetry of the 'coefficient' matrix $\mathbf{K}$ is not generally achieved. 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 generalized Finite Difference Methods of the type described in References 7–12. However, the approach proposed here offers more possibilities. Indeed any of the interpolation techniques described in Section 3 can be used.

5. BOUNDARY CONDITIONS

The boundary conditions given by (2) are imposed by the second and third integral of (3). If a *unique* solution to (3) is to be achieved using approximations of the type given by (4), the total number of *independent* W_i, $\bar{W}_i$ and $\bar{\bar{W}}_i$ functions must be exactly n_p (the number of parameters defining $\hat{u}(x)$). In point collocation methods each W is a Dirac delta function and the above can be achieved by ensuring that the number of point used to approximate the differential equations is $n_p - n_b$, where n_b is the total number of points at which the boundary conditions Γ_t and Γ_u are constructed.

The satisfaction of the essential boundary conditions

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

can be approximated for points x_i placed on Γ_u as

$$\hat{u}(x_i) = u_p(x_i) \quad x_i \quad \text{on } \Gamma_u \tag{33}$$

Similarly, the other boundary conditions on Γ_t may be approximated as

$$B(\hat{u}(x_i)) = t(x_i) \quad x_i \quad \text{on } \Gamma_t \tag{34}$$

If n_u is the number of points where (29) is given and n_t the number of points where (34) is given, then

$$n_b = n_u + n_t \tag{35}$$

and the differential equation is approximated (collocated) at only $n_p - n_b$ points.

The approximation of $\hat{u}(x_i)$ on Γ_u can be performed using different least-squares approximations than that used for the differential equation. For example, approximating the essential boundary conditions

$$u(x_i) = u_p(x_i) \quad \text{on } \Gamma_u \tag{36}$$

by

$$u(x) \simeq \hat{u}(x) = 1 \cdot \alpha_1 \tag{37}$$

and using a cloud which includes only x_i (i.e. one point) gives

$$u(x_i) \simeq \hat{u}(x_i) = u_i^h. \tag{38}$$

Thus, an approximation using

$$u_i^h = u_p(x_i) \quad x_i \quad \text{on } \Gamma_u \tag{39}$$

may be used. In this case, however, the approximation chosen for the domain will *not* be identical to the interpolation given above at the boundary points.

Further details on the treatment of the essential boundary conditions can be found in Reference 28.

6. THE FINITE POINT METHOD IN CONVECTION–DIFFUSION PROBLEMS

For non-self-adjoint problems such as those which occur in fluid mechanics special treatment is needed to stabilize the numerical approximation. As a typical example, we shall outline the special feature on the convection–diffusion equation given by

$$\begin{aligned} c\phi,t + \mathbf{u}\cdot\nabla\phi - \nabla\cdot(k\nabla\phi) - Q = 0 \quad &\text{in } \Omega \\ \mathbf{n}\cdot k\nabla\phi + \bar{q}_n = 0 \quad &\text{in } \Gamma_t \\ \phi - \bar{\phi}_p = 0 \quad &\text{in } \Gamma_u \end{aligned} \tag{40}$$

with the initial condition

$$\phi = \phi_0(\mathbf{x}) \quad \text{for } t = t_0$$

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 function 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 oscillations.[1] 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.[1,30–35] We will test some of these approaches in the context of a FPM *using a WLS approximation and point collocation.*

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

$$\begin{aligned} u\frac{\partial\phi}{\partial x} - k\frac{\partial^2\phi}{\partial x^2} = 0 \\ \phi = 0 \quad \text{in } x = 0 \\ \phi = 1 \quad \text{in } x = L \end{aligned} \tag{41}$$

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

$$\left(\frac{\partial\phi}{\partial x}\right)_i = \frac{\phi_i - \phi_{i-1}}{h} \tag{42}$$

in which h is the distance between two points.

The FE method with linear approximation uses Petrov–Galerkin weighting functions defined as[1]

$$W = N + \frac{h}{2}\alpha\frac{\partial N}{\partial x} \tag{43}$$

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

$$\alpha = \coth|Pe| - \frac{1}{|Pe|} \tag{44}$$

with the Peclet number defined as

$$Pe = \frac{uh}{2k} \tag{45}$$

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 Reference 31, a single parameter is proposed for quadratic elements with an expression identical to equation (44) and h given now by half the element length. 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 the FPM using point collocation.

6.1. *Upwind computation of first derivative*

A simple stabilization procedure in the FPM can be derived by evaluating the first derivative upwind of each approximating point. This is in fact similar to some standard FD stabilization procedures.[32] Taking advantage of the continuity of the field around each point the upwind distance ξ from point x_k can be evaluated in order to obtain exact nodal values (Figure 9).

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

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

and choosing three point clouds ($n = 3$), we have found that *exact* nodal values are obtained if

$$\xi = \frac{h}{2}\alpha \tag{46}$$

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

where h is defined as

$$h = (x_{i+1} - x_{i-1})/2 \tag{47}$$

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

$$\xi = \frac{h}{4}\alpha \tag{48}$$

and evaluating now α by equation (44) with the distance h defined as

$$h = (x_{i+2} - x_{i-2})/2 \tag{49}$$

An identical approach can be followed in two and three-dimensional problems. In two-dimensional problems the critical distance h is defined now as shown in Figure 10.

6.2. *Characteristic approximation*

Other possibility to solve equation (40) is eliminating the convective term using a Lagrangian description. In this way, the operator becomes self-adjoint, and a central difference scheme may be used. This method is also known as characteristic approach.[1,35]

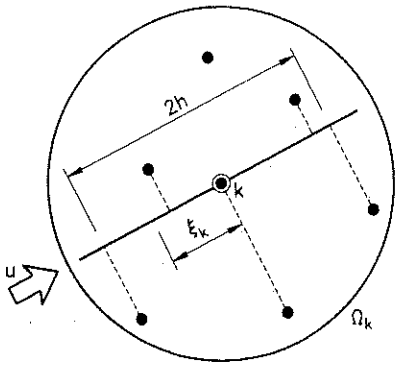


Figure 10. Definition of the critical distance h

Let

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \frac{\mathrm{d}\phi}{\mathrm{d}t} \tag{50}$$

where the right-hand side represents the temporal variation along the particle path (or characteristic).

Along this path equation (40) now becomes

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} - \nabla \cdot (k \nabla \phi) - Q = 0 \tag{51}$$

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

$$\phi^{n+1}(\mathbf{x}) - \phi^n(\mathbf{x} - \boldsymbol{\delta}) = \Delta t \, [\nabla \cdot k \nabla \phi + Q]^{n+1/2} \tag{52}$$

The following Taylor expansion can now be written in component form[1,35]

$$\phi^n(x_i - u_i \Delta t) = \phi^n(x_i) - \Delta t u_j \frac{\partial \phi^n}{\partial x_j} + \frac{\Delta t^2}{2} u_i \frac{\partial}{\partial x_i} \left(u_j \frac{\partial \phi}{\partial x_j} \right)^n \tag{53}$$

$$Q^{n+1/2} = Q\left(x - \frac{\delta}{2}\right) = Q^n - u_i \frac{\Delta t}{2} \frac{\partial Q^n}{\partial x_i} \tag{54}$$

Substituting (53) and (54) into (52) finally gives[35]

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + \mathbf{u} \cdot \nabla \phi^n - \nabla \cdot (k \nabla \phi)^n - Q^n - \frac{\Delta t}{2} \mathbf{u}^{\mathrm{T}} \nabla [\mathbf{u} \cdot \nabla \phi + Q]^n = 0 \tag{55}$$

Note that the last term may be interpreted as an artificial diffusion in *which the term* Δt $\mathbf{u}$ *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 stabilization algorithm is now similar to the upwinding approach described in previous section and identical results are obtained for the examples shown next if Δt is taken to be equal to the critical time step.[1,35] A generalization of this procedure can be found in Reference 38.

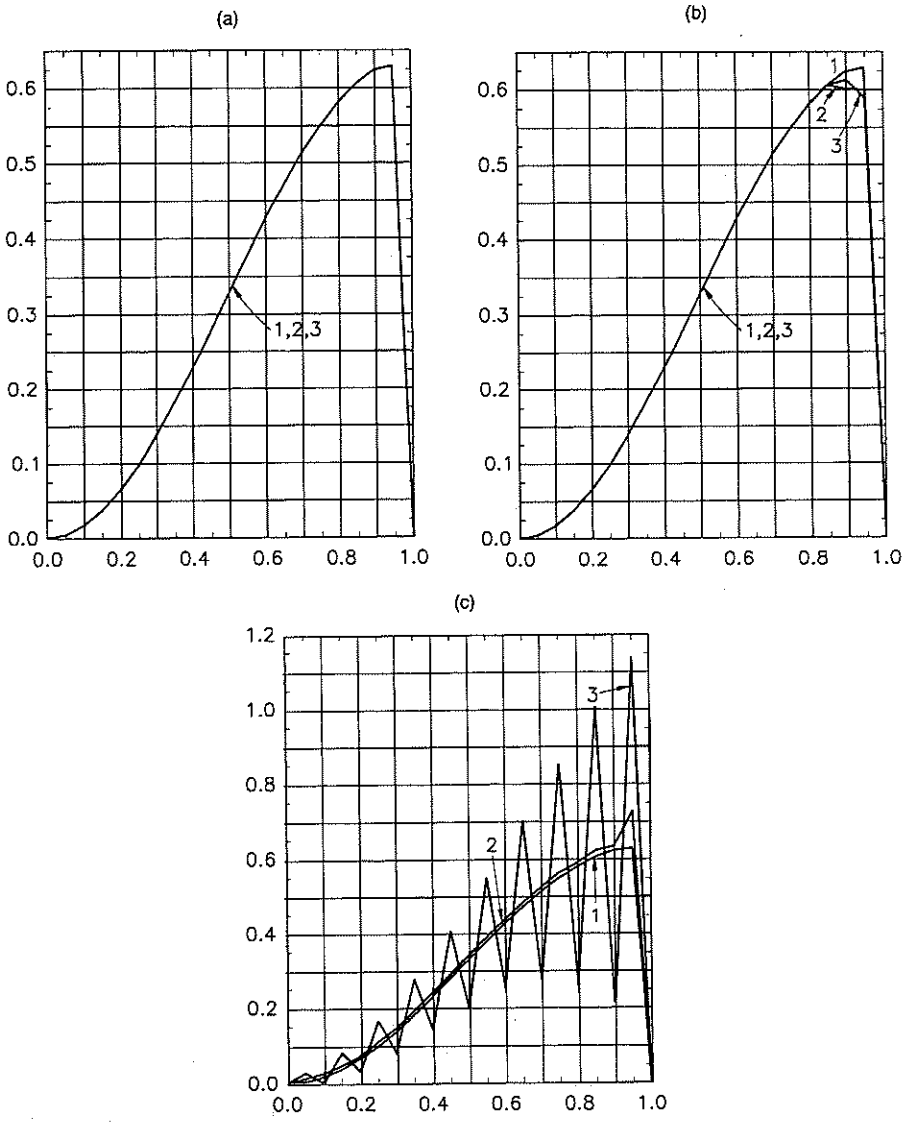


Figure 11. One-dimensional convection–diffusion problem with sinusoidal source team analysed with differents clouds: (a) WLS, $n = 3$; (b) WLS, $n = 5$; (c) WLS, $n = 7$. Quadratic base interpolation ($m = 3$) is used in all cases. Curve 1: Analytical curve. Curve 2: Unknown function values $\hat{\phi}(x_i)$. Curve 3: Unknown parameters curve ϕ_i^h

6.3. Numerical examples

Numerical solutions presented here onwards have been obtained with a FPM based on a WLS interpolation using a fixed Gaussian weighting and point collocation. The essential boundary condition have been imposed by using equation (39)[28]. Numerical results obtained with the standard LSQ approach (and point collocation) are shown in some cases for comparison purposes.

Figure 11 shows steady-state results for equation (40) with a source term

$$Q = \sin \pi x \tag{56}$$

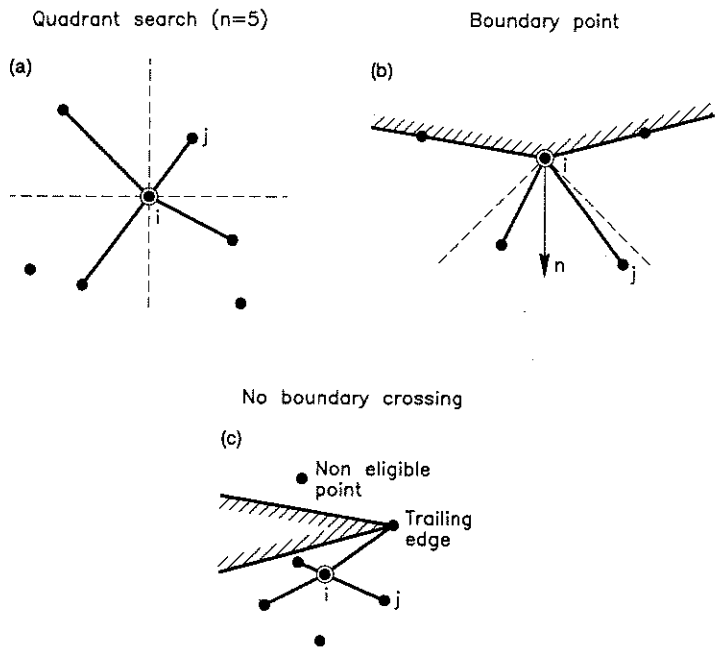


Figure 12. Quadrant technique for selection of points within a cloud: (a) quadrant search ($n = 5$); (b) boundary point; (c) no boundary crossing

and boundary conditions

$$\phi_{(x=0)} = \phi_{(x=1)} = 0 \tag{57}$$

Twenty-one equally spaced points have been used in this case giving

$$Pe = \frac{uh}{2k} = 2{\cdot}5 \tag{58}$$

Figures 11(a)–11(c) show the numerical results using quadratic interpolation ($m = 3$) and clouds with $n = 3$, 5 and 7 points, respectively. In each figure, the exact results, the approximate function $\hat{\phi}(x_i)$ and the parameters ϕ_i^h are shown. For $n = 3$ both solutions are coincident with the analytical one. A small overdiffusion appears for $n = 5$. Finally, for $n = 7$, the unknown parameters ϕ_i^h are inaccurate but the values obtained for the unknown function $\hat{\phi}$ are acceptable.

The next example is the analysis of a pure two-dimensional 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 WLS approach with ϕ_i functions as described in equation (28) rather than the standard LSQ method ($\varphi_i = 1$). In order to select the points involved in each cloud a technique of quadrants has been used consisting in defining a system of ortogonal axes in each point and taking the closest point to the origin of each quadrant. The first set of n points are selected with this criteria (the 'star' node and one in each quadrant for $n = 5$). New points must be added to the cloud if the A matrix defined in (19a) is singular or near singular. The new points were added using the same quadrant criteria for the next nearest points (Figure 12). Figures 13(a)

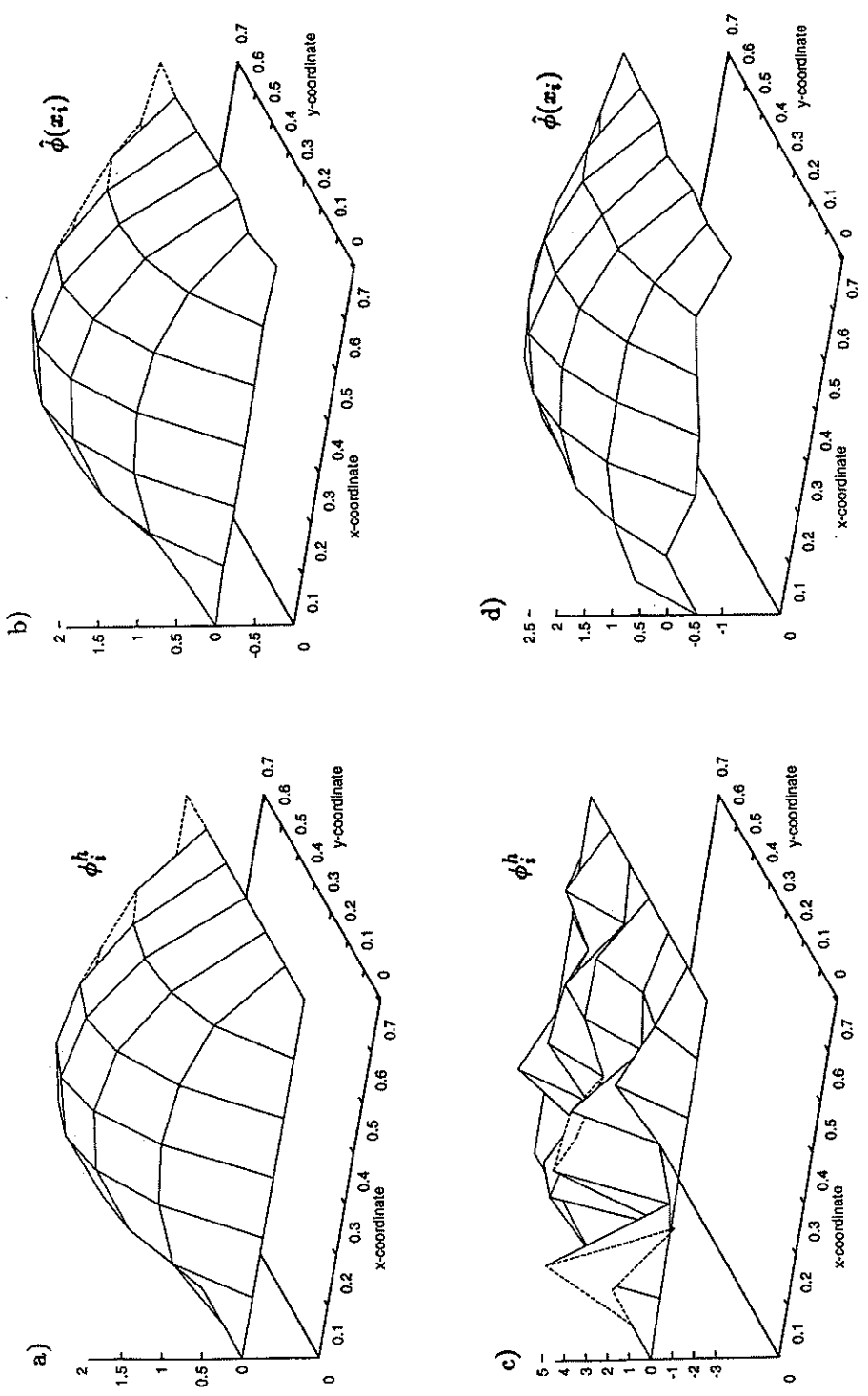


Figure 13. Pure thermal diffusion under constant heat source. 7×7 points grid. Quadratic base interpolation ($m = 6$). Ten points clouds ($n = 10$). Results for ϕ_i^h and $\hat{\phi}(x_i)$ with WLS and LSQ methods: (a) WLS method; (b) WLS method; (c) LSQ method; (d) LSQ method

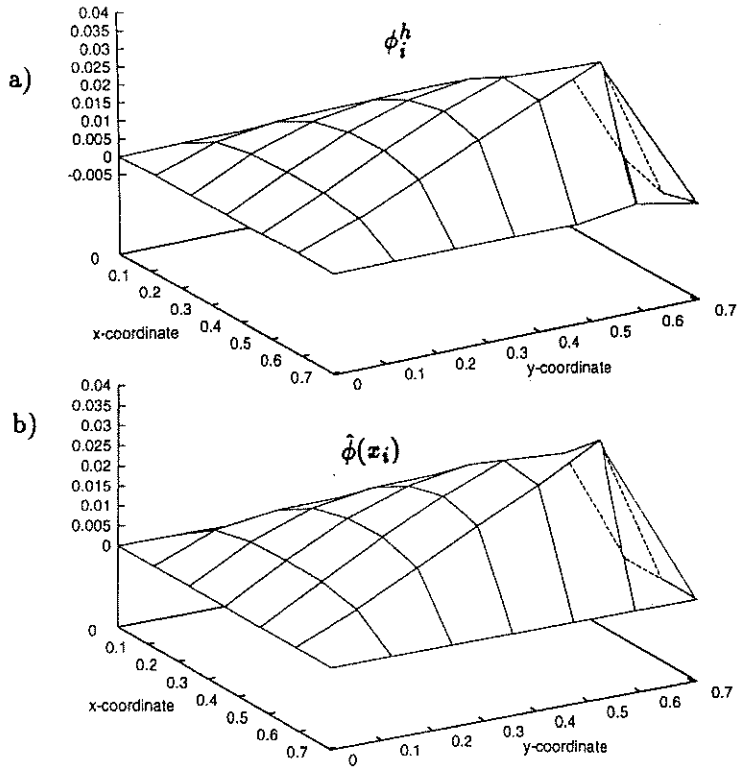


Figure 14. Convection–diffusion problem with diagonal velocity field. 7×7 points grid. Quadratic base interpolation ($m = 6$). Nine points clouds ($n = 9$). Results for ϕ_i^h and $\hat{\phi}(x_i)$ obtained with WLS interpolations and point collocation

and 13(b) 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 (equation (28)). Figures 13(c) and 13(d) display the same results for $\varphi_i = 1$ (LSQ method). Note the deterioration of the solution giving non-physical results for the unknown parameters ϕ_i^h for the second case ($\varphi_i = 1$).

Figure 14 shows results for the same example taking into account convection effects. A diagonally oriented field has been chosen giving $Pe = 10$. Quadratic base interpolating polynomials are again used ($m = 6$) and each cloud contains now nine points ($n = 9$). Figure 14(a) shows the unknown function and Figure 14(b) 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 least-squares approximation for solving convection–diffusion problems with this FP method can be found in References 33, 34 and 38.

7. COMPRESSIBLE FLUID FLOW

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

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{F}_i \frac{\partial \mathbf{U}}{\partial x_i} + \mathbf{K}_{ij} \frac{\partial^2 \mathbf{U}}{\partial x_i \partial x_j} + \mathbf{Q} = 0 \tag{59}$$

with $\mathbf{U} = [\rho, \rho\mathbf{u}^T]^T$, where ρ and $\mathbf{u}$ are density and the velocity vector, respectively and $\mathbf{F}_i$ and $\mathbf{K}_{ij}$ denote the standard convective and diffusive matrix operators.[1,35]

The well-known Lax–Wendroff scheme[32] has been used giving now after same algebra,[1]

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} + \mathbf{F}_i^n \frac{\partial \mathbf{U}^n}{\partial x_i} + \mathbf{K}_{ij}^n \frac{\partial^2 \mathbf{U}^n}{\partial x_i \partial x_j} + \mathbf{Q}^n + \underline{(\Delta t/2)(\partial/\partial x_i)[\mathbf{F}_i(\partial/\partial x_j)(\mathbf{F}_j \mathbf{U})]^n} + \underline{(\Delta t/2)\mathbf{F}_i^n(\partial/\partial x_i)\mathbf{Q}^n = 0} \tag{60}$$

where the the underline identifies the stabilization terms which have some similarity with those emerging from the characteristic approach (see equation (55)).[1,35]

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°.

The problem has again been solved with the FPM and point collocation using both LSQ and WLS (fixed Gaussian weighting) interpolation procedures. The initial distribution of 6694 points was generated with a standard unstructured advancing front triangular mesh generator.[36] The essential boundary conditions around the profile and in the incoming flow were imposed by equation (39). A linear base interpolating polynomial ($m = 3$) and clouds of a minimum of five points ($n = 5$) have been chosen.

Second derivatives corresponding to the stabilization diffusion terms in equation (60) were computed by constructing first a linear interpolation of the obtained solution gradients using the same FP procedure as

$$\frac{\partial \hat{\mathbf{U}}}{\partial x_k} = \sum_{i=1}^{n} N_i \left(\frac{\partial \hat{\mathbf{U}}}{\partial x_k}\right)_i^h \tag{61}$$

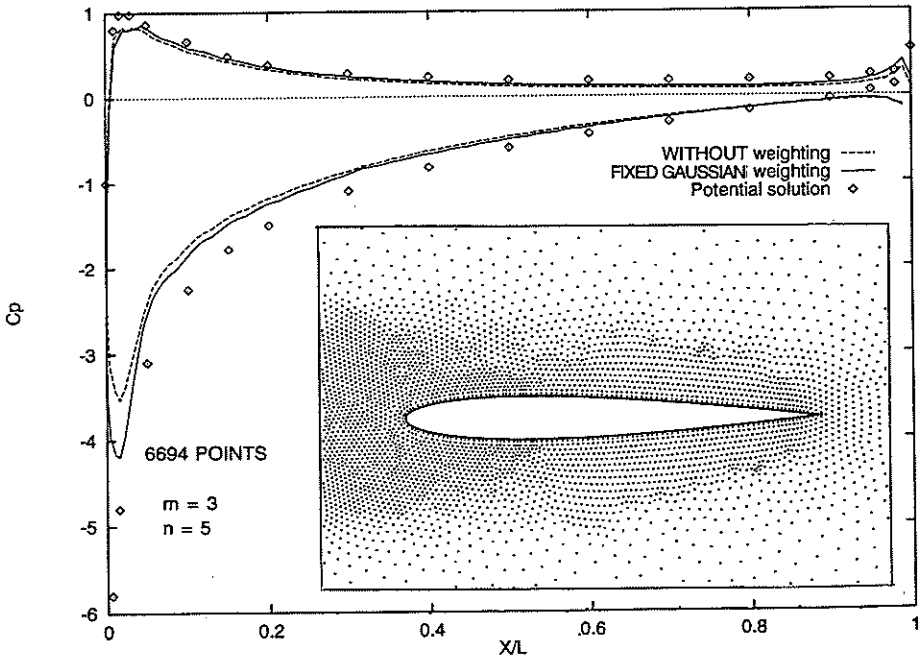


Figure 15. NACA 0012 profile, Mach = 0, 3, $\alpha = 10°$. C_p distribution: (- - -) Constant weighting (LSQ), (———) fixed Gaussian weighting (WLS), (◇) potential solution

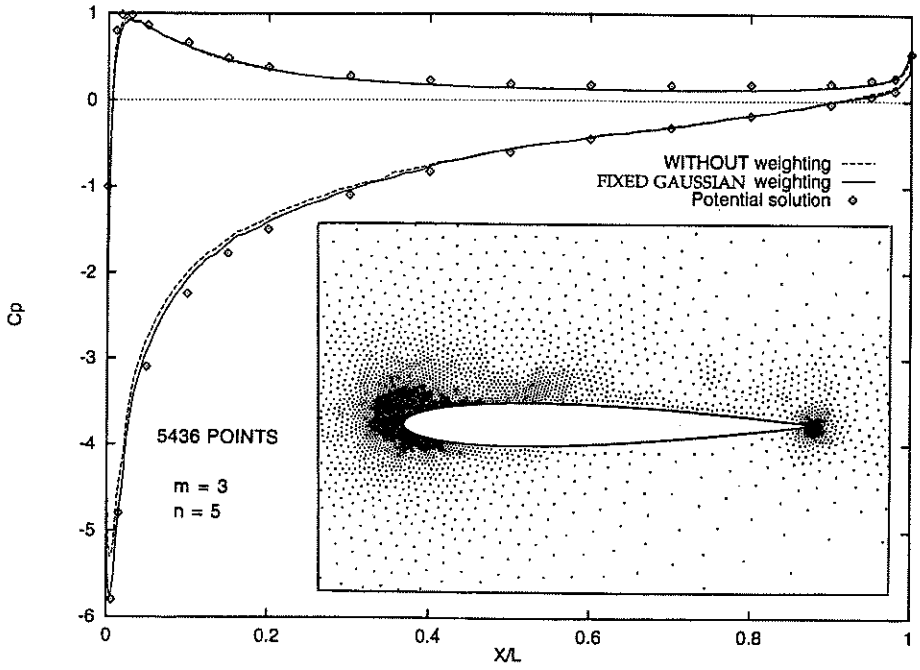


Figure 16. NACA 0012 profile, Mach = 0, 3, $\alpha = 10°$. C_p distribution for an adaptive distribution of points (- - -) Constant weighting (LSQ), (———) fixed Gaussian weighting (WLS), (◇) potential solution

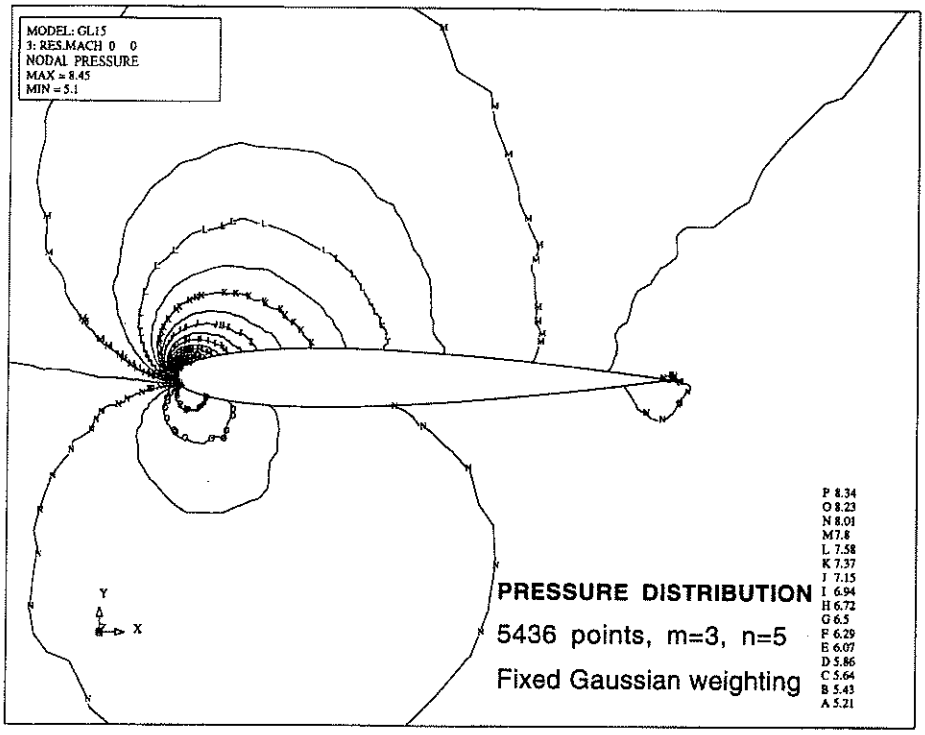


Figure 17. NACA 0012 profile, Mach = 0·3. Pressure distribution

which finally gives

$$\frac{\partial}{\partial x_j}\left(\frac{\partial \hat{U}}{\partial x_k}\right) = \sum_{i=1}^{n} \frac{\partial N_i}{\partial x_j}\left(\frac{\partial \hat{U}}{\partial x_k}\right)_i^h \tag{62}$$

Figure 15 shows the distribution of the pressure coefficient C_p around the profile obtained with the FPM with Gaussian weighting (WLS) and constant weighting (LSQ). The potential solution is also plotted in the figure for comparison purposes. Note the higher accuracy of the WLS method as expected. This difference becomes greater if the number of points is increased. It was found that for $n \geqslant 10$ the results become unstable for the LSQ approximation.

Figure 16 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 Reference 1. Figure 17 shows the pressure distribution which also agrees with the expected result.[32]

Further examples of the solution of fluid flow problems using the FPM can be found in References 33, 34 and 38.

8. CONCLUDING REMARKS

The weighted least-squares interpolation combined with a simple point collocation technique is a promising Finite Point Method for the numerical solution of computational mechanics problems. The advantage of the method compared with standard FEM is to avoid the necessity 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 for convective transport and fluid flow problems and the computing time to solve the differential equation is of the same order as for methods using non-structured grids.

Another interesting conclusion 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.[29,33] 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 large in WLS and MLS methods with linear base interpolations and it quite low in WLS and MLS methods using quadratic base interpolations.

ACKNOWLEDGEMENTS

The second, third and fourth 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. O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*, McGraw-Hill, New York, Vol. I., 1989, Vol. 2, 1991.
2. S. Idelsohn and E. Oñate, 'Finite element and finite volumes. Two good friends', *Int. j. numer. methods eng.*, **37**, 3323–3341 (1994).
3. E. Oñate, M. Cervera and O. C. Zienkiewicz, 'A finite volume format for structural mechanics', *Int. j. numer. methods eng.*, **37**, 181–201 (1994).
4. O. C. Zienkiewicz and E. Oñate, '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, Berlin, 1991.
5. R. H. MacNeal, 'An asymmetrical finite difference network', *Q. Appl. Math.*, **11**, 295–310 (1953).

6. G. E. Forsythe and W. R. Wasow, *Finite Difference Methods for Partial Differential Equations*, Wiley, New York, 1960.
7. P. S. Jensen, 'Finite difference techniques for variable grids', *Comput. Struct.*, **2**, 17–29 (1972).
8. N. Perrone and R. Kao, 'A general finite difference method for arbitrary meshes', *Comput. Struct.*, **5**, 45–47 (1975).
9. W. H. Frey, 'Flexible finite difference stencils from isoparametric finite elements', *Int. j. numer. methods eng.*, **11**, 1653–1665 (1977).
10. V. Pavlin and N. Perrone, 'Finite difference energy techniques for arbitrary meshes applied to linear plate problems', *Int. j. numer. methods eng.*, **14**, 647–664 (1979).
11. T. Liszka and J. Orkisz, 'The finite difference method at arbitrary irregular grids and its application in applied mechanics', *Comput. Struct.*, **11**, 83–95 (1980).
12. T. Liszka, 'An interpolation method for an irregular set of nodes', *Int. j. numer. methods eng.*, **20**, 1594–1612 (1984).
13. R. A. Nay and S. Utku, 'An alternative for the finite element method', *Variational Methods Eng.*, **1**, (1972).
14. R. A. Gingold and J. J. Moraghan, 'Smoothed Particle Hydrodynamics: theory and applications to non spherical stars', *Man. Not. Roy. Astron. Soc.*, **181**, 375–389 (1977).
15. J. J. Moraghan, 'Why particles methods work', *SIAM J. Sci., Statist. Comput.* **3**, 422–433 (1982).
16. J. J. Moraghan, 'An introduction to SPH', *Comput. Phys. Commun.*, **48**, 89–96 (1988).
17. H. E. Trease, M. J. Fritts and W. P. Crowley, (eds.), *Advances in the Free Lagrange Method*, Lecture Notes in Physics, Springer, Berlin, 1990.
18. W. K. Liu, S. Jun and T. Belytschko, 'Reproducing Kernel particle methods', *Int. j. numer. methods fluids*, **20**, 1081–1106 (1995).
19. B. Nayroles, G. Touzot and P. Villon, 'Generalizing the FEM: diffuse approximation and diffuse elements', *Comput. Mech.*, **10**, 307–318 (1992).
20. T. Belytschko, Y. Lu and L. Gu, 'Element free Galerkin methods', *Int. j. numer. methods eng.*, **37**, 229–256 (1994).
21. Y. Y. Lu, T. Belytschko and L. Gu, 'A new implementation of the element free Galerkin method', *Comput. Methods Appl. Mech. Eng.*, **113**, 397–414 (194).
22. C. A. Duarte and J. T. Oden, 'H_p clouds-A meshless method to solve boundary-value problems', *TICAM Repport 95-05*, May 1995.
23. I. Babuska and J. M. Melenk, 'The partition of unity finite element method', *Technical Note EN-1185*, Institute for Physical Science and Technology, Univ. Maryland, April 1995.
24. W. K. Liu, S. Jun, S. Li, J. Adee and T. Belytschko, 'Reproducing Kernel particle methods for structural dynamics', *Int. j. numer. methods eng.*, **38**, 1655–1679 (1195).
25. W. K. Liu and Y. Chen, 'Wavelet and multiple scale producing Kernel methods', *Int. j. numer. methods fluids*, (1995) in print.
26. W. K. Liu, Y. Chen, S. Jun, J. S. Chen, T. Belytschko, C. Pan, R. A. Uras and C. T. Chang, 'Overview and applications of the Reproducing Kernel particle methods', *Arch. Comput. Methods Eng.*, in print, 1996.
27. C. A. M. Duarte, 'A review some meshless methods to solve partial differetial equations', *TICAM Report 95-06*, The Univ. of Texas, Austin, May 1995.
28. E. Oñate, S. Idelsohn and O. C. Zienkiewicz, 'Finite point methods in computational mechanics', *Research Report* No. 67, CIMNE, Barcelona, July 1995.
29. J. Batina, 'A gridless Euler/Navier–Stokes solution algorithm for complex aircraft applications', *AIAA 93-0333*, Reno NV, January 11–14, 1993.
30. A. N. Brooks and T. J. R. Hughes, 'Streamline upwind/Petrov–Galerkin formulations for convective dominated flows with particular emphasis on the incompressible Navier–Stokes equations', *Comput. Methods Appl. Mech. Eng.*, **32**, (1982).
31. R. Codina, E. Oñate and M. Cervera, 'The intrinsic time for the streamline upwind/Petrov–Galerkin formulations using quadratic elements', *Comput. Methods Appl. Mech. Eng.*, **94**, (1992).
32. C. Hirsch, *Numerical Computations of Internal and External Flows*, Vol. 2, Wiley, New York, 1990.
33. T. Fisher, S. Idelsohn and E. Oñate, 'A meshless method for analysis of high speed flows', *AGARD Meeting*, Seville, October 1995.
34. E. Oñate, S. Idelsohn, O. C. Zienkiewicz and T. Fisher, 'A finite point method for analysis of fluid flow problems', *Proc. 9th Int. Conf. on Finite Element Methods in Fluids*, Venize, Italy, 15–21, October 1995.
35. O. C. Zienkiewicz and R. Codina, 'A general algorithm for compressible and incompressible flow. Part I. The split characteristic based scheme', *Int. j. numer. methods fluids*, **20**, 869–885 (1995).
36. J. Peraire, J. Peiró, L. Formaggia, K. Morgan and O. C. Zienkiewicz, 'Finite element Euler computation in three dimensions', *Int. j. numer. methods eng.*, **26**, 2135–2159 (1988).
37. R. L. Taylor, O. C. Zienkiewicz, S. Idelsohn and E. Oñate, 'Moving least square approximations for solution of differential equations', *Research Report no 74*, CIMNE, Barcelona, December 1995.
38. E. Oñate, S. Idelsohn, O. C. Zienkiewicz, R. L. Taylor and C. Sacco, 'A Stabilized finite point method for analysis of fluid mechanics problems', to appear in *Comp. Meth. Appl. Mech. Eng.*