JOURNAL OF COMPUTATIONAL PHYSICS 79, 1–11 (1988)

Algorithms for Interpolation and Localization in Irregular 2D Meshes

DAVID SELDNER* AND THOMAS WESTERMANN

Kernforschungszentrum Karlsruhe GmbH, Institut für Datenverarbeitung in der Technik, P.O. Box 3640, 7500 Karlsruhe, Federal Republic of Germany

Received August 3, 1987; revised November 30, 1987

In numerical simulations of power systems, discrete models and mesh-free models are very often used simultaneously, for example, in particle-in-cell (PIC) codes. The aim of this paper is to introduce new techniques for interpolation as well as for localization in irregular four-point meshes. These algorithms extend the existing methods for regular grids to grids consisting of non-equidistant convex four-point meshes. They were developed in order to obtain short CPU times and possible vectorization. © 1988 Academic Press, Inc.

I. INTRODUCTION

In order to describe the behavior of charged particles in electro-magnetic fields with a PIC code, a grid is introduced in order to compute the forces acting on the particles. The fields are determined at the mesh points of this grid, whereas the movement of the particles takes place in a mesh-free model. The fields at the particle position are determined by an interpolation from the grid (discrete model) onto the particle positions (continuous model). The particles are pushed by means of the forces acting upon them. In order to compute new fields at the mesh points in a self-consistent manner, the particle information must be assigned to the grid. In Fig. 1 such a time-step is outlined. As long as the used grid consists of equidistant cells, the applied algorithms for interpolation and localization are based on simple models [4].

Within the past few years, however, irregular grids such as the boundary-fitted coordinates have been developed in order to treat more complicated geometries, especially with curved boundaries [2, 6]. The cells originating from this approach are not equidistant, but in general arbitrary convex quadrangles. In particular, the existing methods can no longer be applied directly. In the following, new algorithms for interpolation as well as for localization will be introduced.

* The work described in this report is being carried out as part of the German SUPRENUM Supercomputer Project and is being funded by the Federal Ministry for Research and Technology (BMFT), F.R. of Germany, under Grant ITR8502K/4. The author assumes full responsibility for the contents of this report.

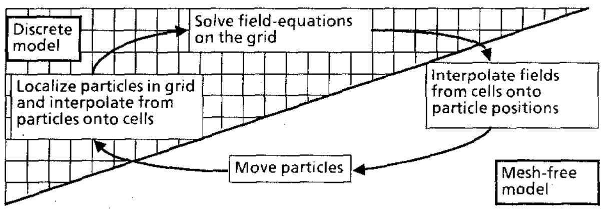
Discrete model
Solve field-equations on the grid
Localize particles in grid and interpolate from particles onto cells
Interpolate fields from cells onto particle positions
Move particles
Mesh-free model

FIGURE 1

II. INTERPOLATION

The Area-Weighting Method

Consider a particle in a certain cell. Usually only the information given at the four corners of the cell is taken into account for calculation of the force at the particle position. If the cell is rectangular, mostly the area-weighting method [3, 5] is applied for interpolation of the functional values $f_{00}, f_{10}, f_{11}, f_{01}$ (see Fig. 2) onto the particle position P.

Let us suppose for the sake of simplicity that the rectangular cell is the unit square. Then the functional value f in point $P(\alpha_1, \alpha_2)$ is calculated as the sum of the weighted functional values f_{ij} $(i, j = 0, 1)$: $f = g_{00}f_{00} + g_{10}f_{10} + g_{11}f_{11} + g_{01}f_{01}$, where

$$\begin{aligned} g_{00} &= (1-\alpha_1)(1-\alpha_2) \qquad & g_{11} &= \alpha_1\alpha_2 \\ g_{10} &= \alpha_1(1-\alpha_2) & g_{01} &= (1-\alpha_1)\,\alpha_2. \end{aligned} \tag{1}$$

The area-weighting method can easily be illustrated, as Fig. 3 shows. The areas A_{00}, A_{10}, A_{11}, A_{01} correspond to the weights g_{00}, g_{10}, g_{11}, g_{01}. This method possesses the following properties:

(P1) If point P lies on the edge of the cell, i.e., on a line $P_{ij}P_{kl}$ $(i, j, k, l \in \{0, 1\}$, $|i-j| + |k-l| = 1)$, only g_{ij} and g_{kl} are taken into account for the evaluation of the value in P. (This property assures continuity of the weights when a particle changes from one cell to the next.)

(P2) $0 \leqslant g_{ij} \leqslant 1$ $(i, j = 0, 1)$ and $g_{00} + g_{10} + g_{11} + g_{01} = 1$ (physical quantities like charge and momentum are conserved).

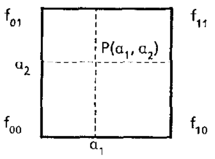

FIGURE 2

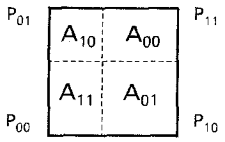

FIGURE 3

(P3) Bilinear functions $f\colon \mathbb{R}^2 \to \mathbb{R}$ with

$$f(x, y) = ax + by + cxy + d, \qquad a, b, c, d \in \mathbb{R},$$

are approximated exactly in $[0, 1] \times [0, 1]$.

Property P3 means that, if the values of a bilinear function are given on the corners of the unit square, and these values are interpolated onto a point inside the unit square, this point is assigned the exact value.

The area-weighting method cannot be applied in arbitrary convex four-point meshes, and therefore generalization is needed.

Formulation of the Problem for Generalization

We sum up the problem as follows: Given is a convex quadrangle G (see Fig. 4a) with the corners (x_{00}, y_{00}), (x_{10}, y_{10}), (x_{11}, y_{11}), and (x_{01}, y_{01}), and the unit square $I^2 = [0, 1] \times [0, 1]$.

Needed is a function F that assigns values $\alpha_1, \alpha_2 \in [0, 1]$ to every point $(x, y) \in G$:

$$F\colon G \to I^2, \qquad (x, y) \to F(x, y) = (\alpha_1, \alpha_2)$$

with the property that for all linear functions

$$f\colon \mathbb{R}^2 \to \mathbb{R} \qquad \text{with} \quad f(x, y) = ax + by + c \qquad (a, b, c \in \mathbb{R})$$

it holds that

$$\begin{aligned} f(x, y) = {} & (1 - \alpha_1)(1 - \alpha_2) f(x_{00}, y_{00}) + \alpha_1(1 - \alpha_2) f(x_{10}, y_{10}) + \alpha_1 \alpha_2 f(x_{11}, y_{11}) \\ & + (1 - \alpha_1) \alpha_2 f(x_{01}, y_{01}) \qquad \text{for all} \quad (x, y) \in G. \end{aligned} \tag{2}$$

$a, b, c \in \mathbb{R}$ being arbitrary, Eq. (2) assures exact approximation of every linear function. Restriction to linear functions is necessary, because—contrary to the interpolation in rectangular cells—arbitrary bilinear functions cannot be approximated exactly. Counterexamples can be constructed easily.

Necessary Conditions for Generalization

In order to find necessary conditions for function F, we insert an arbitrary linear function

$$f(x, y) = ax + by + c \qquad (a, b, c \in \mathbb{R})$$

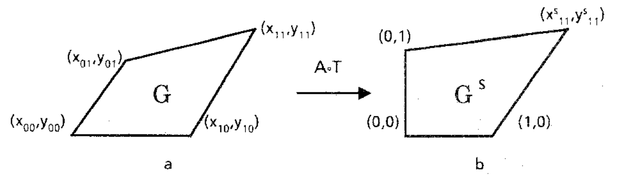

FIGURE 4

into formula (2):

$$ax+by+c=(1-\alpha_1)(1-\alpha_2)(ax_{00}+by_{00}+c)+\alpha_1(1-\alpha_2)(ax_{10}+by_{10}+c)$$
$$+\alpha_1\alpha_2(ax_{11}+by_{11}+c)+(1-\alpha_1)\,\alpha_2(ax_{01}+by_{01}+c).$$

The terms are arranged with respect to *a*, *b*, and *c*, and the coefficients are compared. Because this equation is to be satisfied for arbitrary linear functions, and, hence, for every $a, b, c \in \mathbb{R}$, we obtain

$$\alpha_1\alpha_2(x_{11}-x_{10}-x_{01}+x_{00})+\alpha_1(x_{10}-x_{00})+\alpha_2(x_{01}-x_{00})=x-x_{00}$$
$$\alpha_1\alpha_2(y_{11}-y_{10}-y_{01}+y_{00})+\alpha_1(y_{10}-y_{00})+\alpha_2(y_{01}-y_{00})=y-y_{00}.$$

As can be seen from this system of equations, it is invariant to a translation T by the vector (x_{00}, y_{00}). Thus, without loss of generality, we can assume that $(x_{00}, y_{00})=(0, 0)$. The resulting system of equations

$$\alpha_1\alpha_2\left[\begin{pmatrix}x_{11}\\y_{11}\end{pmatrix}-\begin{pmatrix}x_{10}\\y_{10}\end{pmatrix}-\begin{pmatrix}x_{01}\\y_{01}\end{pmatrix}\right]+\alpha_1\begin{pmatrix}x_{10}\\y_{10}\end{pmatrix}+\alpha_2\begin{pmatrix}x_{01}\\y_{01}\end{pmatrix}=\begin{pmatrix}x\\y\end{pmatrix}$$

is invariant to the multiplication by an invertible matrix A. If we define

$$A=\begin{pmatrix}x_{10} & x_{01}\\ y_{10} & y_{01}\end{pmatrix}^{-1},$$

and perform a coordinate transformation,

$$\begin{pmatrix}x^s\\y^s\end{pmatrix}:=A\begin{pmatrix}x\\y\end{pmatrix},$$

we obtain a new system of equations

$$\alpha_1\alpha_2\begin{pmatrix}x^s_{11}-1\\y^s_{11}-1\end{pmatrix}+\alpha_1\begin{pmatrix}1\\0\end{pmatrix}+\alpha_2\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}x^s\\y^s\end{pmatrix}.$$

Geometrically, this transformation together with the translation T corresponds to a mapping of G into a quadrangle G^s of form in Fig. 4b.

Hence, α_1 and α_2 can be determined by the nonlinear system of equations

$$\alpha_1(1+\alpha_2(x^s_{11}-1))=x^s$$
$$\alpha_2(1+\alpha_1(y^s_{11}-1))=y^s.$$

Explicit Formulae

The solution (α_1, α_2) of the system can be calculated explicitly by the formulae

$$\alpha_2=\frac{-p+(p^2+q)^{1/2}}{(x^s_{11}-1)}\quad\text{for } x^s_{11}\neq 1,\qquad \alpha_2=\frac{y^s}{1+x^s(y^s_{11}-1)}\quad\text{for } x^s_{11}=1$$

and

$$\alpha_1 = \frac{x^s}{1 + \alpha_2(x_{11}^s - 1)} \tag{3}$$

with $p = \frac{1}{2}(1 + x^s(y_{11}^s - 1) - y^s(x_{11}^s - 1))$ and $q = y^s(x_{11}^s - 1)$.

Discussion. It is easy to show that

$$(\alpha_1(x, y), \alpha_2(x, y)) \in I^2 \qquad \text{for all} \quad (x, y) \in G,$$

and function $F: G \to I^2$, defined by

$$F(x, y) := (\alpha_1(x, y), \alpha_2(x, y)),$$

satisfies formula (2) for arbitrary linear functions. Besides, F is continuous (also with respect to x_{11}^s, y_{11}^s) and properties P1 and P2 of the area-weighting method are satisfied. Furthermore, in case G^s is already the unit square, i.e., $x_{11}^s = 1$ and $y_{11}^s = 1$, we have $\alpha_1 = x^s$ and $\alpha_2 = y^s$. This means that the area-weighting method is included as a special case.

III. Localization

The Problem

We consider a boundary-fitted grid consisting of arbitrary convex four-point meshes. In order to interpolate the fields from the grid point onto the particle position inside a cell and in order to assign the particle quantities to the mesh points, it is necessary to know the cell the particle is located in as well as the particle's location within the cell.

Notation. In order to identify cells each grid point is assigned the addresses in x- and y-directions as a pair of numbers (I, J) (see Fig. 5). For a given point $P(x, y)$ we define the grid-weights $(\alpha_1, \alpha_2) \in \mathbb{R}^2$ in such a way that

$$(I, J) = (\text{INT}(\alpha_1), \text{INT}(\alpha_2)),$$

where $\text{INT}(\alpha)$ stands for the integers part of α, represents the address of the left

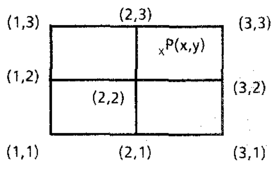

FIGURE 5

lower corner of the cell the particle is in. The weights g_{00}, g_{10}, g_{11}, g_{01} needed for the interpolation are determined by

$$g_{00} = (1-(\alpha_1 - I))(1-(\alpha_2 - J)) \qquad g_{11} = (\alpha_1 - I)(\alpha_2 - J)$$
$$g_{10} = (\alpha_1 - I)(1-(\alpha_2 - J)) \qquad g_{01} = (1-(\alpha_1 - I))(\alpha_2 - J).$$

The integer fraction of (α_1, α_2) hence indicates the address and the decimal fraction indicates the location within the cell. We term $(\alpha_1 - I, \alpha_2 - J)$ the cell-weights.

EXISTING METHOD. Let us consider first a rectangular grid (i.e., a grid where the cells are identical rectangles). In this case the addresses and cell-weights can be computed very conveniently; node (I, J) has the coordinates

$$(x_0 + (I-1)\,\Delta x,\ y_0 + (J-1)\,\Delta y),$$

where (x_0, y_0) are the coordinates of the left lower corner point of the grid and Δx and Δy are the mesh sizes in x- and y-directions, respectively. The address of the cell a particle with the coordinates (x, y) is located in, can be computed by

$$I = \mathrm{INT}((x - x_0)/\Delta x) + 1, \qquad J = \mathrm{INT}((y - y_0)/\Delta y) + 1, \tag{4}$$

and the cell-weights are given by

$$y_1 = (x - x_0)/\Delta x + 1 - I, \qquad y_2 = (y - y_0)/\Delta y + 1 - J. \tag{5}$$

In boundary-fitted grids this method cannot be applied anymore. We consider the fact that a particle is in cell (I, J), iff it is above the lower, left from the right, below the upper, and right from the left cell boundary. Hence, there are at least four IF-clauses necessary for a direct search-algorithm, which is reflected in a very large CPU time. Moreover, such an algorithm is not efficiently vectorizable.

Alternative Localization

We choose a finely meshed rectangular grid which is laid over the boundary-fitted grid (see Fig. 7), and we localize the particles first in the rectangular grid. In doing this, we profit from the rapidity of localization in rectangular grids. In order to obtain a relationship between both grids, each grid point of the rectangular grid is localized in a preparatory phase with respect to the boundary-fitted grid and provided with grid-weights. This can be done, e.g., with the time-consuming search-algorithm mentioned above and by the interpolation already described.

Then the particles are localized and weighted in the rectangular grid using the formulae (4) and (5). In order to obtain the grid-weights with respect to the boundary-fitted grid, the addresses and cell-weights of the four corners of the surrounding rectangular cell are interpolated onto the particle position. This procedure is sketched in Fig. 6.

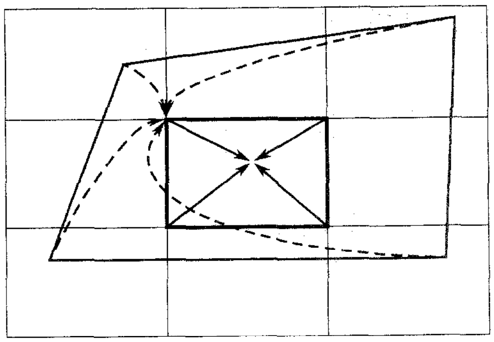

FIGURE 6

For the interpolation of the grid-weights from the corners of the rectangular cell onto the particle position the area-weighting method can be used. Thus the grid-weights at the particle position are given by

$$\alpha_1 = (1-y_1)(1-y_2)\,G^{(1)}_{i,j} + y_1(1-y_2)\,G^{(1)}_{i+1,j} + y_1 y_2 G^{(1)}_{i+1,j+1} + (1-y_1)\,y_2 G^{(1)}_{i,j+1}$$

$$\alpha_2 = (1-y_1)(1-y_2)\,G^{(2)}_{i,j} + y_1(1-y_2)\,G^{(2)}_{i+1,j} + y_1 y_2 G^{(2)}_{i+1,j+1} + (1-y_1)\,y_2 G^{(2)}_{i,j+1}, \tag{6}$$

where $y_1 = (x - x_0)/\Delta x + 1 - I$, $y_2 = (y - y_0)/\Delta y + 1 - J$, with (I, J) assumed to be the address of the equidistant cell the particle is located in, and $G^{(1)}_{i,j}$, $G^{(2)}_{i,j}$ the grid-weights in the x- and y-directions of node (I, J) of the rectangular grid with respect to the boundary-fitted grid.

Discussion. Since the grid-weights of the corner points of the rectangular cells are interpolated onto the particle position, an error is produced in the grid-weights at that position. Even if the method of interpolation presented in Section 2 is used, this causes loss of the property that linear functions are approximated exactly. If, however, the cells in the boundary-fitted grid are identical rectangles, localization including computation of the cell-weights is precise. Thus, in this special case the algorithm coincides with exact localization and application of the area-weighting method.

Let us consider the case that the cell-weights of the nodes of the rectangular grid have been determined using the method of interpolation in Section 2. We analyze the error in the cell-weights generated by the interpolation of the grid-weights onto the particle position.

(a) If the entire rectangular cell lies in one boundary-fitted cell, quadratic convergence with respect to the mesh size of the equidistant grid can be shown (by developing the functions for calculating the grid-weights at the corner points into a

Taylor series around the particle position): Let F_1, F_2 be the exact grid-weights at the particle position, and α_1, α_2 the grid-weights computed by formula (6). Then

$$|\alpha_i - F_i| \leqslant C_1\,\Delta x^2 + C_2 \Delta y^2 + C_3\,\Delta x\,\Delta y \qquad (i=1,2),$$

where Δx and Δy are the mesh-sizes of the equidistant grid in x- and y-directions, respectively, and $C_1, C_2, C_3 \geqslant 0$ only depend upon the boundary-fitted grid.

(b) If the corners of the rectangular cell lie in more than one boundary-fitted cell, linear convergence can still be shown:

$$\exists C_1, C_2 \geqslant 0\colon\ |\alpha_i - F_i| \leqslant C_1\,\Delta x + C_2\,\Delta y \qquad (i=1,2).$$

This raises the question how an error in the grid-weights gets visible in interpolation if values f_{ij} $(i, j=0, 1)$ are to be interpolated from the corners of a boundary-fitted cell onto the particle position. Let us assume that instead of the exact cell-weights $\alpha_1 - I$, $\alpha_2 - J$ wrong cell-weights $\alpha_1 - I + \varepsilon_1$, $\alpha_2 - J + \varepsilon_2$ $(0 \leqslant \alpha_1 - I + \varepsilon_1 \leqslant 1,\ 0 \leqslant \alpha_2 - J + \varepsilon_2 \leqslant 1)$ are used. Then an upper bound for the error in the functional value can be given,

$$|\Delta f/f| \leqslant (|\varepsilon_1| + |\varepsilon_2|)\,f^*,$$

where f is the value determined by the exact cell-weights, Δf is the difference between the exact and the wrong value, and f^* is the maximum difference of the functional values at two neighboring corners of the boundary-fitted cell,

$$f^* = \max\{|f(x_{ij}, y_{ij}) - f(x_{kl}, y_{kl})| : i, j, k, l \in \{0, 1\},\ |i-j| + |k-l| = 1\}.$$

Remarks and Modifications. The drawback of the method described above lies in the fact that by an error in the grid-weights the address of an adjacent cell may be found. By reasons of continuity it can be shown that in such a case the numerical error has no effect upon the simulation, unless this happens in the vicinity of the boundary. In this area there are several possibilities of taking action:

(a) Fitted rectangular grids. The rectangular grid can be constructed in such a manner that lines of this grid coincide with critical lines of the boundary-fitted grid (see Fig. 7). Due to the fact that an error in the addresses can only occur in the

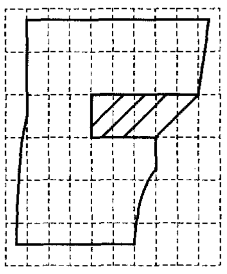
FIGURE 7

rectangular cell containing the particle lies in more than one boundary-fitted cell, the shaded area becomes uncritical.

(b) If a particle is close to the boundary (within a specified tolerance) one can check (e.g., with the search-algorithm described above), whether it is inside or outside the grid. Although such an additional test cannot be vectorized, the increase on CPU time is not too important, because the test has to be performed for few particles only.

As the error in the grid-weights may be explained by the fact that a nonlinear function (the grid-weights in the corners) is interpolated linearly (onto the particle position), one can try to replace the function used for linear interpolation by a function corresponding to the type of function to be interpolated (cf. formula (3)). In our numerical experiments we chose functions with free parameters, which were determined by the condition that the grid-weights in the center of each rectangular cell were to be interpolated exactly. In this way we achieved a considerable decrease in the mean error as well as in the maximum error.

IV. Results

(1) In order to measure the numerical accuracy of the interpolation scheme of Section 2, a simulation was performed in a quarter-circle-like domain (see Fig. 8) with 20×10 grid points. We assumed a monopolar flow of electrons according to the Child–Langmuir law [1] and followed the motion of an ion passing through the anode-cathode gap. We compared the numerically computed energy after about 3000 time-steps with the analytic one and obtained a deviation of 0.02%. Even when applying the localization of Section 3, the deviation was less than 0.03%, which lies in the area of rounding errors.

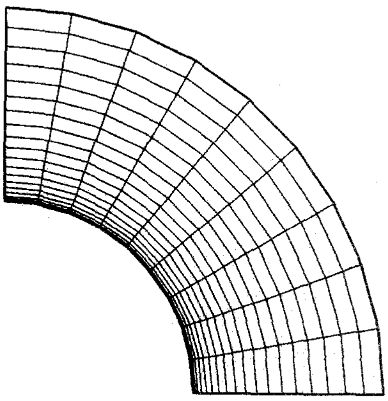

Figure 8

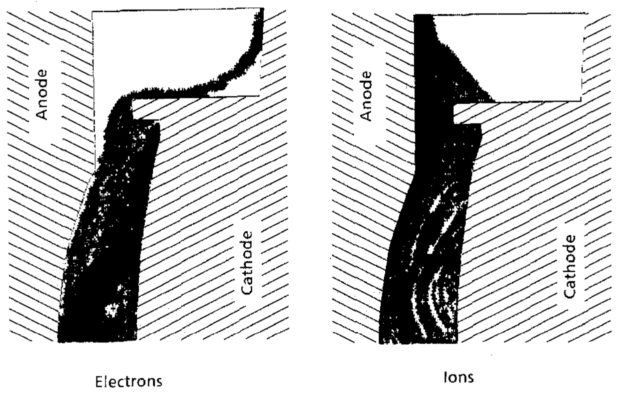

FIGURE 9

(2) A typical run of a PIC code in a high-voltage diode with 11,307 electrons and 10,153 ions was taken as the example (see Fig. 9) [7]. The number of mesh points of the boundary-fitted grid was 21×45, the rectangular grid was 4 and 8 times finer than the boundary-fitted grid. The particles close to the boundary where checked additionally ("inside–outside"). The errors in the grid-weights were computed by means of the l_1-norm. "Linear" stands for the area-weighting method used for the interpolation of the grid-weights from the corners to the particle position and "SQRT" for the nonlinear function applied in that position. The numerical computations were made on a Siemens 7890M and the CPU times are in milliseconds.

For the comparison of the different methods we use the notation

$\Delta\alpha = \Sigma(|\alpha_1 - F_1| + |\alpha_2 - F_2|)/n$ is the mean error in the grid-weights,

$\Delta\alpha_{\max} = \max(|\alpha_1 - F_1| + |\alpha_2 - F_2|)$ is the maximum error in the grid-weights,

n is the number of particles inside the grid, and when computing Σ and max only those particles were considered.

Fineness of the rectangular grid	Linear			With SQRT			Old Method
	$\Delta\alpha$	$\Delta\alpha_{\max}$	CPU	$\Delta\alpha$	$\Delta\alpha_{\max}$	CPU	CPU
4 Small	0.11E-2	0.15	174	0.59E-3	0.63E-1	293	479
Fitted	0.10E-2	0.83E-1	181	0.53E-3	0.72E-1	303	
8 Small	0.37E-3	0.11	171	0.22E-3	0.80E-1	297	
Fitted	0.33E-3	0.51E-1	180	0.19E-3	0.18E-1	306	

V. Conclusions

The methods presented are basic elements for the transition between discrete and mesh-free models. Under the given constraints interpolation allows an accurate determination of the functional value. Although localization in an individual case may assign a neighboring cell to a particle, this error does not impair the simulation as a whole. The methods are vectorizable and even in the scalar case they are associated with substantial saving in computer time.

References

1. E. W. V. Acton, *J. Electron. Control* **3**, 203 (1957).
2. E. Halter, *Die Berechnung elektrostatischer Felder in Pulsleistungsanlagen*, Kernforschungszentrum Karlsruhe GmbH, KfK 4072, Karlsruhe, 1986.
3. F. H. Harlow and W. M. Evans, Los Alamos Report LA-2139, 1957 (unpublished).
4. R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (McGraw–Hill, New York, 1981).
5. R. L. Morse and C. W. Nielson, *Phys. Fluids* **14**, 830 (1971).
6. J. F. Thompson and Z. U. A. Warsi, *J. Comput. Phys.* **47**, 1 (1982).
7. T. Westermann, *Nucl. Instrum. Methods Phys. Res. A* **263**, 271 (1988).