



KfK 4282
Juni 1987

Numerische Algorithmen für zweidimensionale Teilchen- Simulationsmodelle in technisch relevanten Geometrien

D. Seldner, T. Westermann
Institut für Datenverarbeitung in der Technik

Kernforschungszentrum Karlsruhe



KERNFORSCHUNGSZENTRUM KARLSRUHE

Institut für Datenverarbeitung in der Technik

KfK 4282

Numerische Algorithmen für zweidimensionale Teilchen-
Simulationsmodelle in technisch relevanten Geometrien

David Seldner*
Thomas Westermann

* Das diesem Bericht zugrundeliegende Vorhaben wurde aus Mitteln des Bundesministers für Forschung und Technologie unter dem Förderkennzeichen ITR8502K/4 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript vervielfältigt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

Zusammenfassung

In numerischen Simulationen von Power Systemen werden oft gleichzeitig diskrete und gitterfreie Modelle verwendet, z.B. in vielen Bereichen der Strömungsmechanik und vor allem in Particle-In-Cell Codes. Für den Übergang von einem Modell zum anderen ist eine Interpolation bzw. Lokalisierung erforderlich. Sofern die verwendeten Gitter aus äquidistanten Zellen bestehen, beruhen beide Methoden auf einfachen Modellen.

Um auch kompliziertere Geometrien behandeln zu können, werden seit einiger Zeit sog. randangepaßte Koordinaten verwendet. Hier können die existierenden Verfahren nicht mehr eingesetzt werden.

Wir stellen neue Algorithmen vor, die auch in diesen Fällen eingesetzt werden können und in Hinblick auf geringe Rechenzeiten und mögliche Vektorisierung entwickelt wurden.

Numerical Algorithms for 2D-Simulation-Models in Technical Geometries

Summary

In numerical simulations of power systems discrete models and mesh-free models are very often needed simultaneously in order to describe the phenomena, for example in large fields of Fluid Mechanics and especially in Particle-In-Cell codes.

When changing from the discrete to the mesh-free model, the mesh-free data is determined by an interpolation of the discrete data. Vice versa, a localization of the mesh-free data in the used grid is necessary to achieve the transition to the discrete model. Both methods, interpolation and localization, can easily be realized when working with a regular (i.e. an equidistant perpendicular) grid.

Within the past few years, however, irregular grids such as the boundary-fitted coordinates have been developed in order to work with technical geometries. In this case the existing methods can no longer be applied.

The aim of this paper is to introduce new techniques for interpolation as well as for localization in irregular four-point meshes. The algorithms were developed to obtain short CPU-times and possible vectorization.

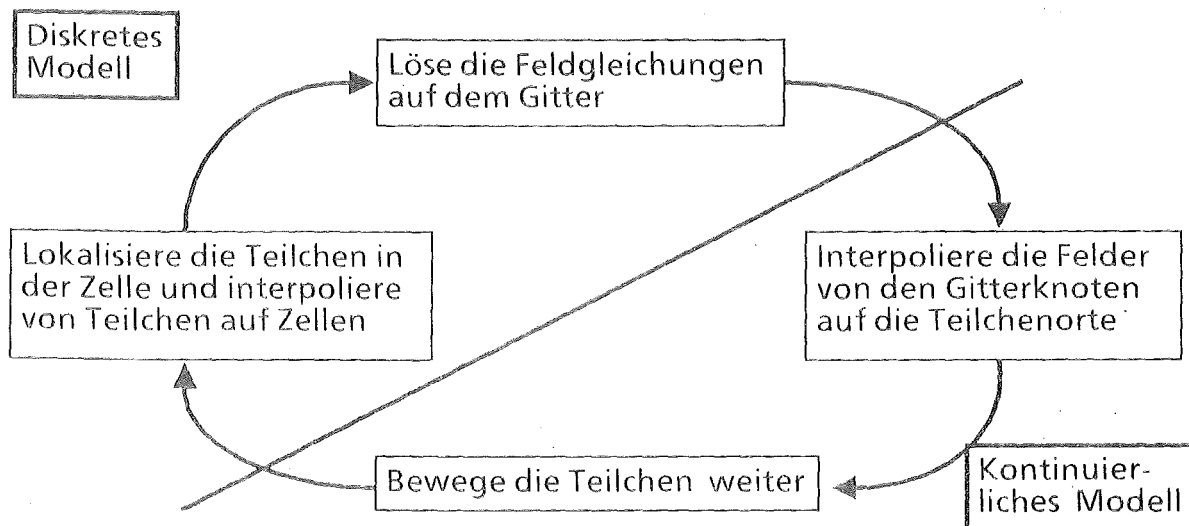
Inhaltsverzeichnis

I. Einleitung	S. 1
II. Interpolation	S. 2
Die Area-Weighting-Methode	S. 2
Problemstellung für eine Verallgemeinerung	S. 4
Notwendige Bedingungen für eine Verallgemeinerung	S. 5
Explizite Formeln	S. 7
Diskussion	S. 7
III. Lokalisierung	S. 8
Problemstellung	S. 8
Notation	S. 8
Bisherige Vorgehensweise	S. 9
Alternative Lokalisierung	S. 10
Diskussion	S. 12
Bemerkungen und Modifikationen	S. 13
Ergebnisse	S. 14
IV. Schlußbemerkung	S. 17
V. Literatur	S. 17

I. Einleitung

Ein geeignetes Verfahren, Simulationen von geladenen Teilchen in elektromagnetischen Feldern durchzuführen, ist die sog. Particle-In-Cell (PIC) Methode. Die Grundidee dieser Methode besteht darin, daß zur Berechnung der Kräfte, die auf die Simulationsteilchen wirken, ein diskretes Gitter eingeführt wird. Auf den Knotenpunkten dieses Gitters werden die Felder bestimmt, während die Fortbewegung der Teilchen in einem gitterfreien, kontinuierlichen Modell erfolgt.

Der prinzipielle Ablauf eines PIC-Codes ist dann i.a. dadurch bestimmt, daß durch eine Interpolation vom diskreten Modell auf die kontinuierlichen Teilchenorte die Felder am Teilchenort berechnet werden. Die Teilchen werden aufgrund der Kräfte, die auf sie wirken, fortbewegt. Damit die Felder auf den Gitterknotenpunkten von neuem selbstkonsistent berechnet werden können, muß die Teilcheninformation wieder auf das Gitter verteilt werden. Schematisch ist ein solcher Zeitschritt im folgenden Diagramm dargestellt:



Sofern die verwendeten Gitter aus äquidistanten Zellen bestehen, beruhen die für die Interpolation und Lokalisierung erforderlichen Algorithmen auf einfachen Modellen. Um auch kompliziertere Geometrien, insbesondere mit krummlinigen Rändern, behandeln zu können, werden seit einiger Zeit sog. randangepaßte Koordinaten verwendet. Die aus diesem Ansatz entstehenden

Zellen sind teilweise hochgradig nichtäquidistant und i.a. beliebig schief liegende, konvexe Vierecke. Insbesondere sind die bisher verwendeten Algorithmen nicht mehr direkt anwendbar. Sowohl für die Interpolation als auch für die Lokalisierung sollen im folgenden neue Algorithmen vorgestellt werden, welche die bisher verwendeten Algorithmen auf nichtäquidistante, konvexe Viereckgitter übertragen.

II. Interpolation

Die Area-Weighting-Methode

Befindet sich ein Teilchen in einer gewissen Zelle, so wird zur Berechnung der Kraft am Teilchenort üblicherweise nur auf die Information zurückgegriffen, die auf den vier Zelleckpunkten gegeben ist. Die hierbei meist benutzte Methode zur Interpolation der Funktionswerte f_{00} , f_{10} , f_{11} , f_{01} (siehe Abb. 1) auf den Teilchenort P ist die Area-Weighting-Methode.

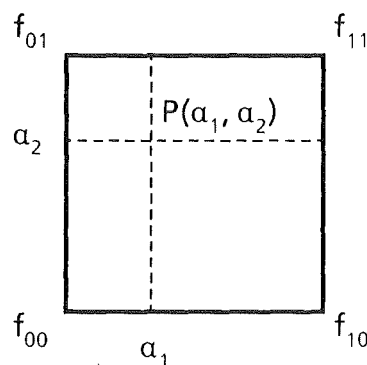


Abbildung 1

Nehmen wir einfachheitshalber an, daß die Rechteckzelle ein Einheitsquadrat ist, so ermittelt sich der gesuchte Funktionswert f am Teilchenort (α_1, α_2) als Summe der gewichteten Funktionswerte f_{ij} ($i, j = 0, 1$) durch

$$f = g_{00}f_{00} + g_{10}f_{10} + g_{11}f_{11} + g_{01}f_{01},$$

mit

$$g_{00} = (1 - \alpha_1)(1 - \alpha_2) \quad (1)$$

$$g_{10} = \alpha_1(1 - \alpha_2)$$

$$g_{11} = \alpha_1\alpha_2$$

$$g_{01} = (1 - \alpha_1)\alpha_2$$

Die Area-Weighting-Methode kann man in diesen quadratischen Zellen geometrisch einfach veranschaulichen (siehe Abb. 2). Die durch g_{00} , g_{10} , g_{11} , g_{01}

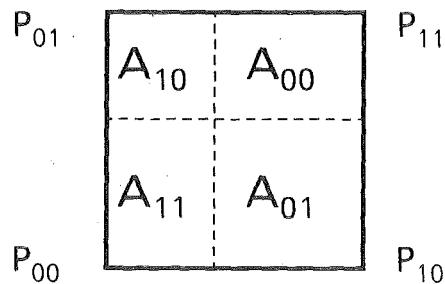


Abbildung 2

berechneten Gewichte entsprechen den Flächen A_{00} , A_{10} , A_{11} , A_{01} .

Diese Interpolations-Methode besitzt die folgenden Eigenschaften:

- E1) Liegt der Punkt P auf einer Randstrecke $P_{ij}P_{kl}$ ($i, j, k, l \in \{0, 1\}$, $|i - j| + |k - l| = 1$), so tragen nur g_{ij} und g_{kl} zur Funktionsbestimmung am Orte P bei.
- E2) $0 \leq g_{ij} \leq 1$ ($i, j = 0, 1$).
- E3) $g_{00} + g_{10} + g_{11} + g_{01} = 1$.
Diese Bedingung bedeutet physikalisch, daß bei der Verteilung der Teilchengrößen auf das Gitter die Teilchengrößen (wie z.B. die Ladung) eine Erhaltungsgröße bilden.
- E4) Bilineare Funktionen $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ mit

$$f(x, y) = ax + by + cxy + d, \quad a, b, c, d \in \mathbb{R},$$
werden in $[0, 1] \times [0, 1]$ exakt approximiert.

Die Area-Weighting-Methode ist in beliebig schief liegenden konvexen Vierecken zunächst nicht anwendbar, und es wird daher eine Verallgemeinerung benötigt. Eine Möglichkeit der Übertragung dieser Methode soll der folgende Ansatz bieten.

Problemstellung für eine Verallgemeinerung

Wir fassen die Problemstellung folgendermaßen zusammen:

Gegeben sei ein konvexes Viereck G (vgl. Abb. 3a) mit den Ecken (x_{00}, y_{00}) , (x_{10}, y_{10}) , (x_{11}, y_{11}) und (x_{01}, y_{01}) und das Einheitsquadrat $I^2 = [0, 1] \times [0, 1]$.

Gesucht ist eine Funktion

$$F: G \rightarrow I^2, \quad (x, y) \rightarrow F(x, y) = (\alpha_1, \alpha_2)$$

mit der Eigenschaft, daß für alle linearen Funktionen $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ gilt:

$$f(x, y) = (1 - \alpha_1)(1 - \alpha_2)f(x_{00}, y_{00}) + \alpha_1(1 - \alpha_2)f(x_{10}, y_{10}) + \alpha_1\alpha_2f(x_{11}, y_{11}) + (1 - \alpha_1)\alpha_2f(x_{01}, y_{01}) \quad (2)$$

für alle $(x, y) \in G$.

Die Beschränkung auf lineare Funktionen ist notwendig, da im Gegensatz zur Interpolation in Rechteck-Zellen beliebige bilineare Funktionen nicht mehr exakt approximiert werden können, wie das folgende Beispiel zeigt:

Wir wählen uns ein konvexes Viereck G und einen Punkt P in diesem Gebiet und zeigen, daß das Paar (α_1, α_2) im Punkte P von der jeweils gewählten bilinearen Funktionenklasse abhängt.

Sei G definiert durch die Ecken $(x_{00}, y_{00}) = (0, 0)$, $(x_{10}, y_{10}) = (1, 0)$, $(x_{11}, y_{11}) = (1, 2)$ und $(x_{01}, y_{01}) = (0, 1)$, P besitze die Koordinaten $(x, y) = (\frac{1}{2}, 1)$. Wir betrachten zunächst die bilinearen Funktionen

$$f_1: \mathbb{R}^2 \rightarrow \mathbb{R}, \text{ mit } f_1(x, y) = cxy + ax \quad (a, c \in \mathbb{R}).$$

Damit ist $(\alpha_1, \alpha_2) \in I^2$ gesucht, so daß

$$f_1(\frac{1}{2}, 1) = \alpha_1(1 - \alpha_2)a + \alpha_1\alpha_2(a + 2c), \text{ bzw.}$$

$$\frac{1}{2}a + \frac{1}{2}c = \alpha_1a + \alpha_1\alpha_22c \text{ für beliebiges } a, c.$$

Hieraus folgt $\alpha_1 = \frac{1}{2}$ und $\alpha_2 = \frac{1}{2}$.

Betrachten wir nun die bilinearen Funktionen

$$f_2: \mathbb{R}^2 \rightarrow \mathbb{R}, f_2(x, y) = cxy + by \quad (b, c \in \mathbb{R}).$$

Wir erhalten analog zu oben

$$f_2\left(\frac{1}{2}, 1\right) = \alpha_1 \alpha_2 (b + 2c) + \alpha_2 (1 - \alpha_1) b, \text{ bzw.} \\ b + \frac{1}{2}c = \alpha_2 b + \alpha_1 \alpha_2 2c \text{ für beliebiges } b, c$$

und damit $\alpha_1 = \frac{1}{4}$ und $\alpha_2 = 1$.

Somit ist das Paar (α_1, α_2) abhängig von den speziell gewählten bilinearen Funktionen. Eigenschaft (E4) der Area-Weighting-Methode kann also nicht in dieser Allgemeinheit auf randangepaßte Koordinaten übertragen werden. Wie wir sehen werden, ist sie jedoch für lineare Funktionen

$$f: \mathbb{R}^2 \rightarrow \mathbb{R} \text{ mit } f(x, y) = ax + by + c \quad (a, b, c \in \mathbb{R})$$

noch aufrecht zu erhalten.

Notwendige Bedingungen für eine Verallgemeinerung

Um notwendige Bedingungen für die gesuchte Funktion F zu finden, setzen wir in Formel (2) eine beliebige Funktion

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

ein:

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

Wir ordnen die rechte Seite dieser Gleichung nach den Koeffizienten a , b und c und erhalten:

$$ax + by + c = a\{\alpha_1 \alpha_2 (x_{11} - x_{10} - x_{01} + x_{00}) + \alpha_1 (x_{10} - x_{00}) + \alpha_2 (x_{01} - x_{00}) + x_{00}\} + \\ b\{\alpha_1 \alpha_2 (y_{11} - y_{10} - y_{01} + y_{00}) + \alpha_1 (y_{10} - y_{00}) + \alpha_2 (y_{01} - y_{00}) + y_{00}\} + \\ c$$

Da diese Gleichung für beliebige lineare Funktionen erfüllt sein soll, muß gelten:

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

Wie aus diesem Gleichungssystem ersichtlich wird, sind die Gleichungen invariant bezüglich einer Translation T des Vierecks um den Vektor (x_{00}, y_{00}) . Insbesondere können wir damit o.E.d.A. annehmen, daß $(x_{00}, y_{00}) = (0, 0)$. Das nun zu lösende Gleichungssystem

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

ist invariant bezüglich der Multiplikation mit einer invertierbaren Matrix. Definieren wir

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

so erhalten wir durch die Koordinatentransformation mit

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

das normierte Gleichungssystem

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

Geometrisch entspricht diese Transformation einer linearen Abbildung, die das auf den Ursprung verschobene Viereck auf ein Standard-Viereck G^s der Form (3b)

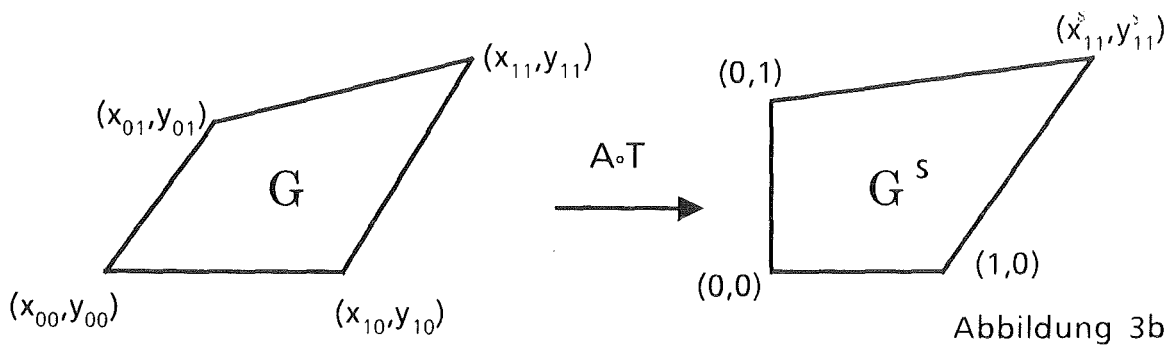


Abbildung 3a

Abbildung 3b

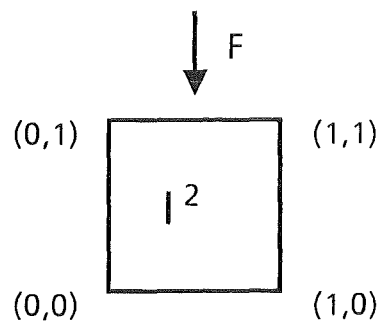


Abbildung 3c

abbildet.

Für α_1 und α_2 erhalten wir für festes (x, y) das nichtlineare Gleichungssystem

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

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

Explizite Formeln

Die Lösungen (α_1, α_2) des Gleichungssystems können explizit durch die Formeln

$$\alpha_2 = \frac{-p + \sqrt{p^2 + q}}{(x_{11}^s - 1)} \quad \text{für } x_{11}^s \neq 1$$

$$\alpha_2 = \frac{y^s}{1 + x^s(y_{11}^s - 1)} \quad \text{für } x_{11}^s = 1 \quad (3)$$

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

$$\text{mit } p = \frac{1}{2}(1 + x^s(y_{11}^s - 1) - y^s(x_{11}^s - 1)) \text{ und } q = y^s(x_{11}^s - 1)$$

berechnet werden.

Diskussion

- 1) Man kann leicht zeigen, daß für alle $(x, y) \in G$ gilt:
 $(\alpha_1(x, y), \alpha_2(x, y)) \in I^2$
 und die gesuchte Funktion $F: G \rightarrow I^2$, definiert durch
 $F(x, y) := (\alpha_1(x, y), \alpha_2(x, y))$,
 Formel (2) für beliebige lineare Funktionen für alle $(x, y) \in G$ erfüllt.

- 2) Für $x_{11}^s = 1$ ist

$$\alpha_2 = \lim_{x_{11}^s \rightarrow 1} \frac{-p + \sqrt{p^2 + q}}{(x_{11}^s - 1)} = \frac{y^s}{1 + x^s(y_{11}^s - 1)}, \quad \alpha_1 = x^s$$

womit insbesondere für $x_{11}^s = 1$ die Lösung stetig und beschränkt ist.

Für $y_{11}^s = 1$ gilt

$$\alpha_2 = y^s, \quad \alpha_1 = \frac{x^s}{1 + y^s(x_{11}^s - 1)}$$

- 3) Durch den Ansatz (2) ist sichergestellt, daß für die durch (1) definierten g_{ij} ($i, j = 0, 1$) gilt
 $0 \leq g_{ij} \leq 1$ ($i, j = 0, 1$)
 und
 $g_{00} + g_{10} + g_{11} + g_{01} = 1$.

- 4) Liegt der Punkt P auf einer Randstrecke $P_{ij}P_{kl}$ ($i, j, k, l \in \{0, 1\}$, $|i - j| + |k - l| = 1$), so tragen nur g_{ij} und g_{kl} zur Funktionsbestimmung am Orte P bei, denn es gilt

$$F \circ A \circ T(\{P_{00}P_{01}\}) = \{0\} \times [0, 1],$$

$$F \circ A \circ T(\{P_{00}P_{10}\}) = [0, 1] \times \{0\},$$

$$F \circ A \circ T(\{P_{10}P_{11}\}) = \{1\} \times [0, 1],$$

$$F \circ A \circ T(\{P_{01}P_{11}\}) = [0, 1] \times \{1\}.$$

- 5) Im Falle, daß das Gebiet G schon ein Einheitsquadrat ist, d.h. $x_{11} = 1$ und $y_{11} = 1$, gilt $\alpha_1 = x$ und $\alpha_2 = y$, womit die übliche Area-Weighting-Methode als Grenzfall enthalten ist.

III. Lokalisierung

Problemstellung

Wir betrachten ein randangepaßtes Gitter, das aus beliebigen konvexen Viereckzellen besteht. Zum Interpolieren der Feldstärken von den Knotenpunkten des Gitters auf den Teilchenort in der Zelle und zum Verteilen der Teilchengrößen auf die Knotenpunkte ist es notwendig, sowohl diejenige Zelle zu kennen, in der sich das Teilchen befindet, als auch die Position des Teilchens innerhalb der Zelle.

Notation

Zur Kennzeichnung von Zellen werden jedem Knoten des Gitters als Zahlenpaar (I, J) die Adressen in x - und y -Richtung zugewiesen (siehe Abb. 4). Wir definieren

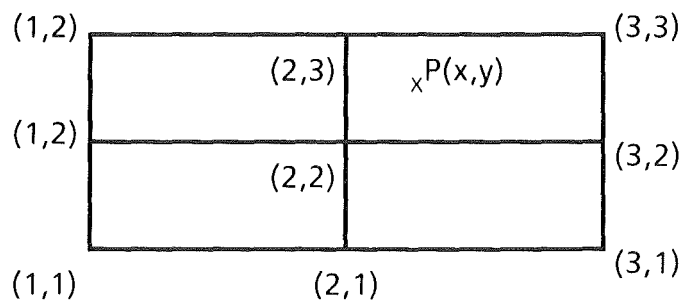


Abbildung 4

für einen Punkt die Gittergewichte $(\alpha_1, \alpha_2) \in \mathbb{R}^2$ so, daß

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

die Adresse des linken unteren Knotenpunktes der Zelle, in der sich der Punkt befindet, enthält und die zur Interpolation benötigten Gewichte $g_{00}, g_{10}, g_{11}, g_{01}$ durch

$$g_{00} = (1 - (\alpha_1 - I))(1 - (\alpha_2 - J))$$

$$g_{10} = (\alpha_1 - I)(1 - (\alpha_2 - J))$$

$$g_{11} = (\alpha_1 - I)(\alpha_2 - J)$$

$$g_{01} = (1 - (\alpha_1 - I))(\alpha_2 - J)$$

bestimmt sind. Der ganzzahlige Anteil von (α_1, α_2) gibt also die Zelladresse an und der Dezimalanteil die Lage innerhalb der Zelle. Im folgenden nennen wir $(\alpha_1 - I, \alpha_2 - J)$ Zellgewichte.

Bisherige Vorgehensweise

Wir betrachten zunächst ein Rechteck-Gitter (d.h. ein Gitter, in dem die Zellen identische Rechtecke sind). Hier ist die Berechnung der Zelladressen und der Zellgewichte sehr einfach, denn der Knoten (I, J) besitzt die Koordinaten

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

wobei (x_0, y_0) die Koordinaten des linken unteren Eckpunktes und Δx und Δy die Maschenweiten in x- bzw. y-Richtung sind. Die Adresse der Zelle, in der sich ein Teilchen mit den Koordinaten (x, y) befindet, berechnet sich aus

$$\begin{aligned} I &= \text{INT}((x - x_0)/\Delta x) + 1 \\ J &= \text{INT}((y - y_0)/\Delta y) + 1, \end{aligned} \quad (4)$$

und für die Zellgewichte gilt:

$$\begin{aligned} \gamma_1 &= (x - x_0)/\Delta x + 1 - I \\ \gamma_2 &= (y - y_0)/\Delta y + 1 - J. \end{aligned} \quad (5)$$

In randangepaßten Gittern ist diese Vorgehensweise nicht mehr möglich. Ein Teilchen befindet sich in der randangepaßten Zelle (I, J) , wenn es sich

- über der unteren,
- links von der rechten,
- unter der oberen und
- rechts von der linken Zellenbegrenzung

befindet. Dies legt folgenden Suchalgorithmus nahe:

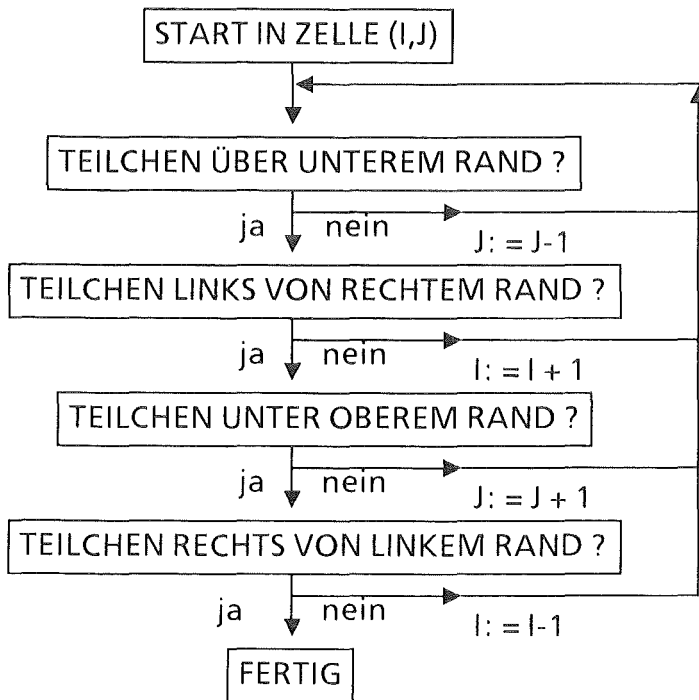
Schritt 1: Starte in Zelle (I,J) (z.B. der Zelle, in dem das Teilchen vorher war).

Schritt 2: Falls das Teilchen unter dem unteren Zellrand: Setze $J = J - 1$, gehe zu 2.

Schritt 3: Falls das Teilchen rechts vom rechten Zellrand: Setze $I = I + 1$, gehe zu 2.

Schritt 4: Falls das Teilchen über dem oberen Zellrand: Setze $J = J + 1$, gehe zu 2.

Schritt 5: Falls das Teilchen links vom linken Zellrand: Setze $I = I - 1$, gehe zu 2.



Der Nachteil dieser Methode besteht darin, daß für jedes Teilchen mindestens vier IF-Abfragen notwendig sind, um es zu lokalisieren. Dies äußert sich in einer sehr hohen CPU-Zeit. Zudem ist diese Vorgehensweise nicht effizient vektorisierbar.

Alternative Lokalisierung

Wir wählen ein feines Rechteck-Gitter, das über das randangepaßte Gitter gelegt wird, und lokalisieren die Teilchen zunächst im Rechteck-Gitter. Hierbei machen wir uns die Schnelligkeit der Lokalisierung in Rechteck-Gittern zunutze. Um anschließend eine Beziehung zwischen den beiden Gittern zu erhalten, wird in einer Vorbereitungsphase jeder Gitterknoten des Rechteck-Gitters bezüglich des randangepaßten Gitters lokalisiert und mit Gittergewichten versehen. Dies kann

beispielsweise mit dem o.g. zeitaufwendigen Suchalgorithmus und der oben beschriebenen Interpolation geschehen.

Sodann lokalisiert und gewichtet man ein Teilchen im Rechteck-Gitter mittels der Formeln (4) und (5) und interpoliert die Zelladressen und Zellgewichte der vier Eckpunkte des umgebenden Rechtecks auf den Teilchenort, um die Gittergewichte bezüglich des randangepaßten Gitters zu erhalten. Abbildung 5 veranschaulicht dies schematisch.

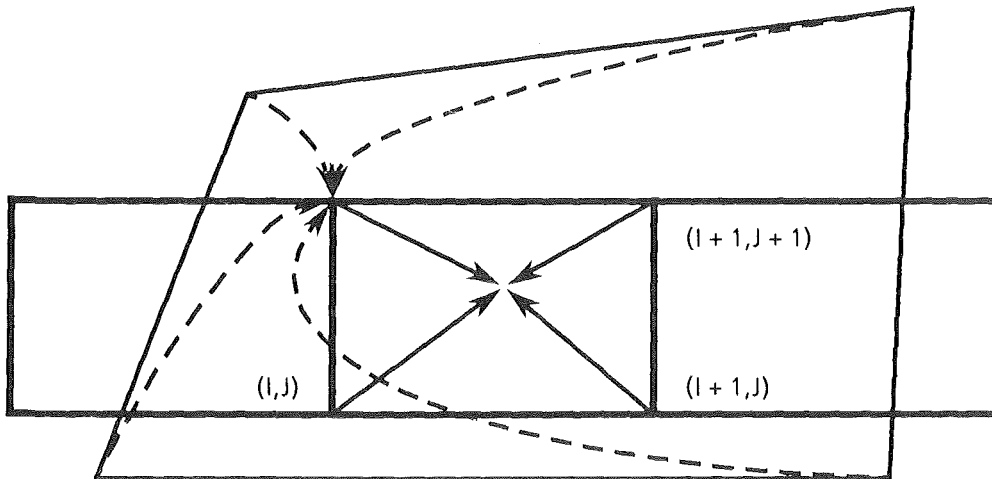


Abbildung 5

Für die Interpolation der Gittergewichte von den Ecken der rechteckigen Zelle auf den Teilchenort verwenden wir die Area-Weighting-Methode. Damit erhalten wir also die Gittergewichte am Teilchenort durch:

$$\begin{aligned} \alpha_1 &= (1 - \gamma_1)(1 - \gamma_2)G_{ij}^{(1)} + \gamma_1(1 - \gamma_2)G_{i+1j}^{(1)} + \gamma_1\gamma_2 G_{i+1j+1}^{(1)} + (1 - \gamma_1)\gamma_2 G_{ij+1}^{(1)} \\ \alpha_2 &= (1 - \gamma_1)(1 - \gamma_2)G_{ij}^{(2)} + \gamma_1(1 - \gamma_2)G_{i+1j}^{(2)} + \gamma_1\gamma_2 G_{i+1j+1}^{(2)} + (1 - \gamma_1)\gamma_2 G_{ij+1}^{(2)} \end{aligned} \quad (6)$$

mit $\gamma_1 = (x - x_0)/\Delta x + 1 - i$, $\gamma_2 = (y - y_0)/\Delta y + 1 - j$, dabei sei (i,j) die Adresse der äquidistanten Zelle, in der sich das Teilchen befindet, und

$$G_{ij}^{(1)}, G_{ij}^{(2)}$$

die Gittergewichte des Knotens (i,j) des äquidistanten Gitters bezüglich des randangepaßten Gitters in x- bzw. y-Richtung.

Diskussion

Dadurch daß die Gittergewichte der Eckpunkte der Rechteckzellen auf den Teilchenort interpoliert werden, entsteht ein Fehler in den Gittergewichten am Teilchenort. Selbst wenn wir die in Kapitel 2 vorgestellte Interpolationsmethode verwenden, geht dadurch i.a. die Eigenschaft verloren, daß lineare Funktionen exakt approximiert werden. Sind jedoch die durch die Methode der randangepaßten Koordinaten entstehenden Zellen identische Rechtecke, so ist die Lokalisierung einschließlich der Berechnung der Zellgewichte exakt. In diesem Spezialfall fällt also der vorgestellte Algorithmus mit einer exakten Lokalisierung und der Anwendung der Area-Weighting-Methode zusammen.

Wir betrachten im nachfolgenden den Fall, daß die Zellgewichte der Knotenpunkte des Rechteck-Gitters mittels der Interpolationsmethode von Kapitel 2 ermittelt wurden und analysieren den Fehler in den Zellgewichten, der bei der Interpolation der Gittergewichte auf den Teilchenort entsteht.

- a) Befindet sich die Rechteck-Zelle ganz in einer randangepaßten Zelle, so gilt quadratische Konvergenz bzgl. der Feinheit des äquidistanten Gitters, d.h.: Sind F_1, F_2 die exakten Gittergewichte am Teilchenort und α_1 und α_2 die durch die Formeln (6) berechneten Gittergewichte, so gilt:

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

dabei sind Δx und Δy die Maschenweiten des feinen Gitters in x - bzw. y -Richtung und C_1, C_2, C_3 nur vom randangepaßten Gitter abhängende positive Zahlen.

- b) Liegen die Ecken der rechteckigen Zelle in mehr als einer randangepaßten Zelle, so kann noch lineare Konvergenz gezeigt werden:

$$\exists C_1, C_2 > 0 : |\alpha_i - F_i| \leq C_1 \Delta x + C_2 \Delta y \quad (i = 1, 2).$$

Damit stellt sich die Frage, wie sich ein Fehler in den Zellgewichten bei der Interpolation bemerkbar macht, wenn Funktionswerte f_{ij} ($i, j = 0, 1$) von den Ecken einer randangepaßten Zelle auf den Teilchenort interpoliert werden sollen. Werden anstelle der Zellgewichte $\alpha_{1-l}, \alpha_{2-j}$ die Zellgewichte $\alpha_{1-l+\varepsilon_1}, \alpha_{2-j+\varepsilon_2}$ ($0 \leq \alpha_{1-l+\varepsilon_1} \leq 1, 0 \leq \alpha_{2-j+\varepsilon_2} \leq 1$) verwendet, so gilt folgende Abschätzung:

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

dabei sei f der durch die exakten Zellgewichte bestimmte Funktionswert, Δf die Differenz vom exakten Wert zum interpolierten Wert und f^* die maximale Differenz der Funktionswerte an zwei benachbarten Ecken der randangepaßten Zelle,

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

Bemerkungen und Modifikationen

Der Nachteil der oben beschriebenen Methode liegt vor allem darin, daß durch das Auftreten eines Fehlers in den Gittergewichten u.U. auch bei der Lokalisierung eines Teilchens in einer Zelle die Adresse einer benachbarten Zelle gefunden werden kann. Die wichtigste hieraus zu ziehende Konsequenz ist, daß Lokalisierung und Interpolation nicht getrennt betrachtet werden können. Aus Stetigkeitsüberlegungen kann gezeigt werden, daß in einem solchen Fall der numerische Fehler keine Auswirkungen auf die Simulation hat, falls man sich nicht am Rand des Gitters befindet.

Am Rand gibt es mehrere mögliche Vorgehensweisen:

a) Angepaßte Rechteck-Gitter

Man kann das Gitter so legen, daß Gitterlinien durch besonders kritische Stellen des zu behandelnden Gebietes verlaufen (siehe Abb. 6).

Da Fehler in den Adressen nur dann auftreten können, wenn die das Teilchen umgebende Rechteck-Zelle in mehreren randangepaßten Zellen liegt, wird das Problem bzgl. des schraffierten Gebiets beseitigt.

b) Wird für ein Teilchen festgestellt, daß es in Randnähe liegt (innerhalb einer vorgegebenen Toleranz), wird untersucht, ob es sich inner- oder außerhalb des Gitters befindet. Eine solche zusätzliche Untersuchung ist zwar nicht vektorisierbar, da sie aber nur für wenige Teilchen durchgeführt werden muß, fällt die hinzukommende Rechenzeit nicht allzu sehr ins Gewicht.

Da der Fehler in den Gewichten damit begründet werden kann, daß eine nichtlineare Funktion (nämlich die der Gittergewichte an den Ecken) linear interpoliert wird (auf den Teilchenort), kann man nun versuchen, statt der linearen Interpolation eine Interpolationsfunktion zu wählen, die dem Typ der zu interpolierenden Funktion (vgl. Formel (3)) entspricht.

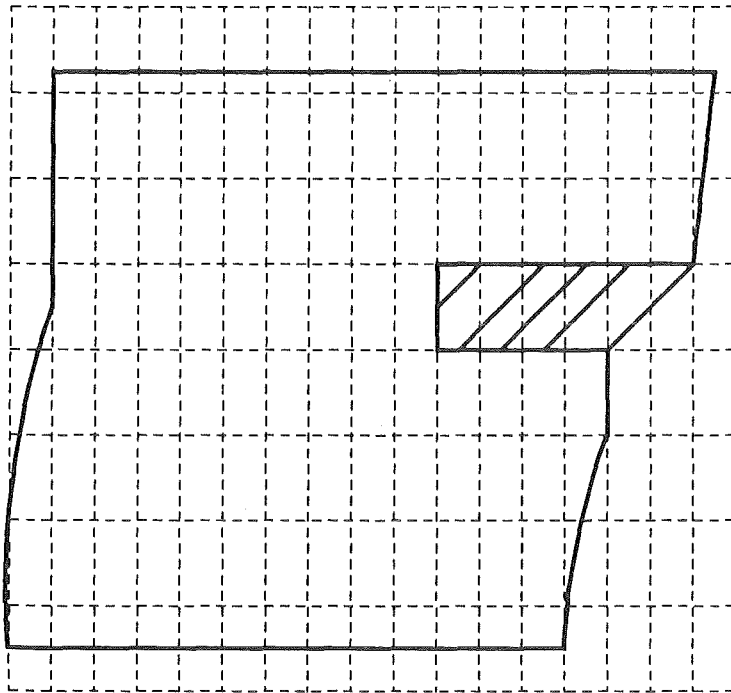


Abbildung 6

In unseren Versuchen wählten wir Interpolationsfunktionen mit jeweils einem freien Parameter, der so bestimmt wurde, daß die Gittergewichte im Mittelpunkt jeder rechteckigen Zelle exakt interpoliert wurden. Auf diese Art und Weise erhielten wir eine erhebliche Verringerung sowohl des durchschnittlichen als auch des maximalen Fehlers. In diesem Fall gilt die Aussage der quadratischen Konvergenz bzgl. der Maschenweite des feinen Gitters jedoch nicht mehr, da dort lineare Interpolation vorausgesetzt wurde.

Ergebnisse

Zugrundegelegt wurde ein typischer Lauf eines PIC-Codes in einer Hochstromdiode mit 11307 Elektronen und 10153 Ionen (siehe Abb. 7). Die Zahl der Gitterpunkte des randangepaßten Gitters betrug 11×21 , 15×41 und 21×45 , das feine Gitter war jeweils 4- bzw. 8-mal feiner als das randangepaßte Gitter. und die Teilchen, die sich in Randnähe des randangepaßten Gitters befanden, wurden nochmals untersucht. Die berechneten Fehler in den Gittergewichten wurden mit der l_1 -Norm ermittelt. Die Auswirkungen des Fehlers wurden durch

Interpolation der Funktion $f(x,y) = x-2y + 1$ getestet. Die numerischen Ergebnisse beziehen sich auf eine Siemens 7890 M und die angegebenen CPU-Zeiten sind in Millisekunden angegeben.

Dabei sind

$\Delta\alpha = \Sigma(|\alpha_1 - F_1| + |\alpha_2 - F_2|)/n$ der durchschnittliche Gittergewichtsfehler,

$\Delta\alpha_{\max} = \max(|\alpha_1 - F_1| + |\alpha_2 - F_2|)$ der maximale Gittergewichtsfehler,

$\Delta f = \Sigma|f' - f|/n$ der durchschnittliche Fehler in den Funktionswerten und

$\Delta f_{\max} = \max(|f' - f|)$ der maximale Fehler in den Funktionswerten.

f ist der durch die exakten Gittergewichte berechnete Funktionswert und f' der durch die fehlerhaften Gittergewichte interpolierte. n ist die Zahl der Teilchen, die sich innerhalb des Gitters befinden und Σ und \max laufen über diese Teilchen.

Randangepaßtes Gitter: 11×21

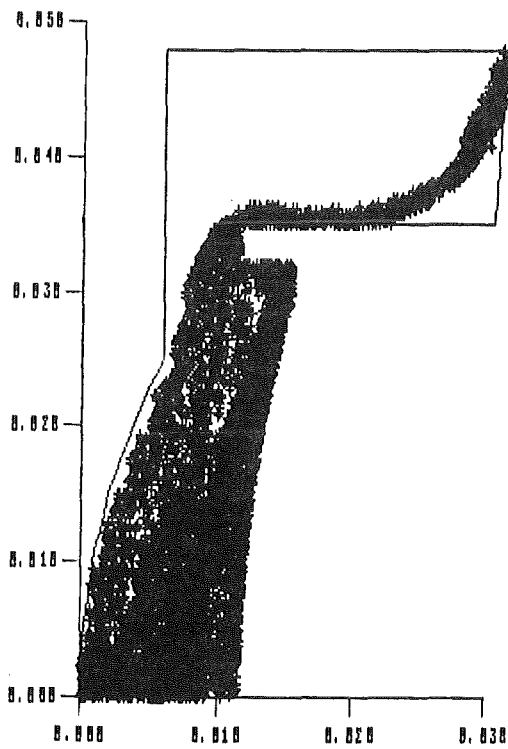
Feinheit des äquidist. Gitters		$\Delta\alpha$	$\Delta\alpha_{\max}$	Δf	Δf_{\max}	CPU-Zeit	CPU-Zeit (alte Methode)
4	mögl. klein	0.13E-2	0.15	0.19E-5	0.11E-3	187	317
	angepaßt	0.12E-2	0.68E-1	0.21E-5	0.11E-3	174	
8	mögl. klein	0.43E-3	0.80E-1	0.65E-6	0.46E-4	147	
	angepaßt	0.43E-3	0.55E-1	0.67E-6	0.44E-4	149	

Randangepaßtes Gitter: 15×41

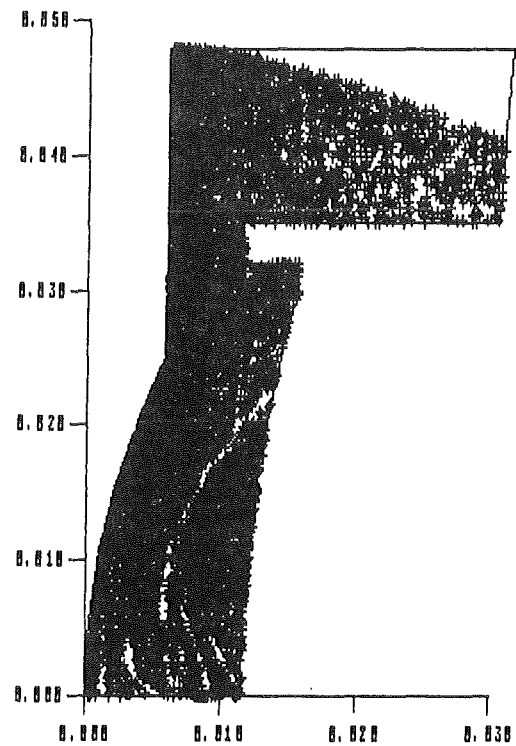
Feinheit des äquidist. Gitters		$\Delta\alpha$	$\Delta\alpha_{\max}$	Δf	Δf_{\max}	CPU-Zeit	CPU-Zeit (alte Methode)
4	mögl. klein	0.14E-2	0.14	0.13E-5	0.12E-3	176	340
	angepaßt	0.13E-2	0.80E-1	0.12E-5	0.10E-3	162	
8	mögl. klein	0.35E-3	0.31E-1	0.37E-6	0.47E-4	155	
	angepaßt	0.39E-3	0.34E-1	0.39E-6	0.54E-4	156	

Randangepaßtes Gitter: 21×45

Feinheit des äquidist. Gitters		$\Delta\alpha$	$\Delta\alpha_{\max}$	Δf	Δf_{\max}	CPU-Zeit	CPU-Zeit (alte Methode)
4	mögl. klein	0.11E-2	0.15	0.90E-6	0.18E-3	174	479
	angepaßt	0.10E-2	0.83E-1	0.87E-6	0.76E-4	181	
8	mögl. klein	0.37E-3	0.11	0.30E-6	0.49E-4	171	
	angepaßt	0.33E-3	0.51E-1	0.28E-6	0.65E-4	180	



Elektronen



Ionen

Abbildung 7 : Elektronen und Ionen in einer Hochstromdiode

IV. Schlußbemerkung

Die in diesem Artikel vorgestellten Methoden sind wichtige Bausteine des PIC-Codes. Die Interpolation ermöglicht unter den gegebenen Rahmenbedingungen (nur die vier Eckpunkte einer Zelle tragen zu den Zellgewichten bei) eine genaue Bestimmung des Funktionswertes. Die Lokalisierung kann im Einzelfall zwar eine benachbarte Zelle einem Teilchen zuweisen, jedoch beeinträchtigt dieser Fehler die Gesamtsimulation nicht. Sie ist vektorisierbar und bringt auch im skalaren Fall einen erheblichen Gewinn an Rechenzeit.

V. Literatur

- [1.] C. K. Birdsall, A. B. Langdon, Plasma Physics via Computer Simulation, McGraw-Hill, 1985
- [2.] J.M. Dawson, Particle simulation of plasmas, Reviews of Modern Physics, Vol. 55, No. 2, p. 403, April 1983
- [3.] E. Halter, Die Berechnung elektrostatischer Felder in Pulsleistungsanlagen, Kernforschungszentrum Karlsruhe GmbH, KfK 4072, Karlsruhe 1986
- [4.] R. W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, McGraw-Hill, 1981