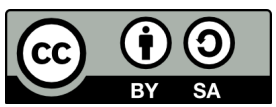
Fast Multipole Method für Potentiale in 2D

Robert Hemstedt

Rheinische Friedrich Wilhelms-Universität Bonn – Bonn,

betreut durch die Herren Prof. Dr. Schweitzer und Prof. Dr. Klein

27. April 2014



Gliederung

Motivation

Beispiel

In der Ebene seien m paarweise verschiedene Punkte $\{x_1, \ldots, x_m\}$ gegeben, die jeweils die Ladungen $\{q_1, \ldots, q_m\}$ tragen. Zusätzlich seien n paarweise verschiedene Punkte $\{y_1, \ldots, y_n\}$ in der Ebene gegeben, an denen die Potentiale $\Phi(y_i)$ $(i = 1, \ldots, n)$ berechnet werden sollen. Der Einfachheit halber gelte $x_i \neq y_j$ für alle $i = 1 \ldots m, j = 1, \ldots, n$.

Frage

In welcher Zeit können alle Potentiale an den ausgewählten Punkten berechnet werden?

Klassischer Ansatz – Direktes Ausrechnen

Fakt

Potential: $\Phi_{x_i}(y_j) = -q_i \log(||y_j - x_i||)$*[1]*

Also ergeben sich für alle Potentiale an den Punkten $y_j, j = 1, \ldots, n$:

$$\sum_{i=1}^{m} \Phi_{x_i}(y_j) \qquad \text{für alle } j = 1, \ldots, n$$

Also insgesamt $O(mn)$ Summationen für alle Potentiale. Insbesondere $O(m^2)$, wenn $m = n$.

G-Kernel

Die Funktion $\Phi_{x_i}(y_j)$ bezeichnet man in manchen Fällen auch als *Kernel*[2]. Bezeichne:

$G(\mathbf{y}, \mathbf{x}) := -\log(||\mathbf{x}-\mathbf{y}||)$ und $q(\mathbf{x}_i) := q_i$, also $\Phi_{\mathbf{x}_i}(\mathbf{y}_j) = G(\mathbf{y}_j, \mathbf{x}_i)q(\mathbf{x}_i)$

Ein Trick der FMM ist es, das Problem in die Komplexe Ebene zu legen. Identifiziere:

$$\begin{aligned} \mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2) \quad &\Rightarrow \quad z = \mathbf{x}_1 + i\mathbf{x}_2, \\ \mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2) \quad &\Rightarrow \quad z_0 = \mathbf{y}_1 + i\mathbf{y}_2, \\ G(\mathbf{y}, \mathbf{x}) = \Re(G(z_0, z)), \text{ mit } G(z_0, z) &= -\log(z_0 - z), \end{aligned}$$

wobei wir hier den komplexen (Hauptzweig des) Logarithmus verwenden.

Expansion des G-Kernels

In dieser Notation berechnet man:

$$\sum_{i=1}^{m} G(y_j, x_i) q(x_i) \qquad \text{für alle } i = 1, \ldots, n,$$

und nimmt den *Realanteil*. Dabei sind die Punkte $(x_i)_{i=1,\ldots,m}$, $(y_j)_{j=1,\ldots,n}$ entsprechend in die Komplexe Ebene eingebettet.
Die Idee der Multipolexpansion ist es nun für einen Expansionspunkt y_c den Kernel zu schreiben als

$$G(y, x) = \sum_i G_i^y(y_c, y) G_i^x(y_c, x)$$

$G_i^x(y_c, x)$ ist *unabhängig* von y, kann also wiederverwendet werden!

Expansion des G-Kernels

Konvention

Ab sofort arbeiten wir nur noch in $\mathbb{C}$.

Lemma

Es seien der Quellpunkt z_0, der Feldpunkt z und ein Expansionspunkt z_c nahe z gegeben, d.h. $|z - z_c| \ll |z_0 - z_c|$. Dann gilt:

$$G(z_0, z) = \sum_{k=0}^{\infty} O_k(z_0 - z_c) I_k(z - z_c). \tag{1}$$

Wobei

$$I_k(z) = \frac{z^k}{k!}, k \geq 0 \text{ sowie}$$
$$O_k(z) = \frac{(k-1)!}{z^k}, k \geq 1; \text{ und } O_0(z) = -\log(z).$$

Expansion des G-Kernels

Beweis.

$$G(z_0, z) = -\log(z_0 - z) = -\left[\log(z_0 - z_c) + \log\left(1 - \frac{z - z_c}{z_0 - z_c}\right)\right]$$

Taylor-Expansion von

$$\log(1 - \xi) = -\sum_{k=1}^{\infty} \frac{\xi^k}{k}, |\xi| < 1$$

liefert das gewünschte Ergebnis mit $\xi = \frac{z - z_c}{z_0 - z_c}$. □

Es ist einer der Schlüsselpunkte der FMM, dass durch Gleichung (1) im G-Kernel z_0 und z durch z_c separiert werden.

Multipolexpansion

Satz (Multipolexpansion)

Es seien die Punkte $z_1, \ldots, z_m$ mit Ladungen $q(z_1), \ldots, q(z_m)$ gegeben. Weiterhin seien z_0 und z_c derart, dass $|z_i - z_c| \ll |z_0 - z_c|$ für alle $i = 1, \ldots, m$. Dann gilt:

$$\sum_{i=1}^{m} G(z_0, z_i) q(z_i) = \sum_{i=1}^{m} \left[\sum_{k=0}^{\infty} O_k(z_0 - z_c) I_k(z_i - z_c) \right] q(z_i);$$

also die Multipolexpansion:

$$\sum_{i=1}^{m} G(z_0, z_i) q(z_i) = \sum_{k=0}^{\infty} O_k(z_0 - z_c) M_k(z_c), \tag{2}$$

mit den Momenten:

$$M_k(z_c) = \sum_{i=1}^{m} I_k(z_i - z_c) q(z_i). \tag{3}$$

Momente

Der Überlegung wert

Die Momente

$$M_k(z_c) = \sum_{i=1}^{m} I_k(z_i - z_c) q(z_i)$$

sind nur von den gegebenen Ladungspunkten $z_1, \ldots, z_m$ und dem Expansionspunkt z_c abhängig und brauchen nicht nochmal berechnet werden, wenn z_0 seine Lage ändert! Später werden die z_c Zentren von Zellen eines Gitters sein, sodass die Momente $M_k(z_c)$ häufig wiederverwendet werden können. Dies ist ein *Schlüsselfaktor*, warum die FMM so schnell ist!

Fehler der Multipolexpansion

In (2) tritt eine *unendliche* Summe auf. Fehler in der Fast Multipole Method können durch die Anzahl der ausgewerteten Terme in (2) kontrolliert werden.

Satz (Fehler der Multipolexpansion)

Für den Fehler E_M^p der Multipolexpansion gilt:

$$E_M^p := \left| \sum_{i=1}^{m} G(z_0, z_i) q(z_i) - \sum_{k=0}^{p} O_k(z_0 - z_c) M_k(z_c) \right|$$
$$\leq \frac{A}{1 - \frac{R}{|z_0 - z_c|}} \frac{R^{p+1}}{|z_0 - z_c|^{p+1}},$$

wobei R der Radius eines Kreises um z_c mit $|z_i - z_c| < R < |z_0 - z_c|$ für alle $i = 1, \ldots, m$ und $A = \sum_{i=1}^{m} |q_i|$.

Fehler der Multipolexpansion

Beweis.

$$
\begin{aligned}
E_M^p &:= \left| \sum_{k=p+1}^{\infty} O_k(z_0 - z_c) M_k(z_c) \right| \leq \sum_{k=p+1}^{\infty} |O_k(z_0 - z_c)||M_k(z_c)| \\
&\leq \sum_{k=p+1}^{\infty} |O_k(z_0 - z_c)| \left| \sum_{i=1}^{m} I_k(z_i - z_c) q(z_i) \right| \\
&\leq \sum_{k=p+1}^{\infty} |O_k(z_0 - z_c)| \sum_{i=1}^{m} |I_k(z_i - z_c)||q(z_i)| \\
&\leq A \sum_{k=p+1}^{\infty} |O_k(z_0 - z_c)| \frac{R^k}{k!} \leq A \sum_{k=p+1}^{\infty} \frac{(k-1)!}{|z_0 - z_c|^k} \frac{R^k}{k!} \\
&\leq A \sum_{k=p+1}^{\infty} \frac{R^k}{|z_0 - z_c|^k} = \frac{A}{1 - \frac{R}{|z_0 - z_c|}} \frac{R^{p+1}}{|z_0 - z_c|^{p+1}}
\end{aligned}
$$

□

Fehler der Multipolexpansion

Sei $\rho = \frac{|z_0 - z_c|}{R}$. Dann lässt sich E_M^p abschätzen als

$$E_M^p \leq \frac{A}{\rho - 1} \left(\frac{1}{\rho}\right)^p$$

Bemerkung

Je größer ρ, desto kleiner ist die Fehlerschranke. Beobachte, dass für den Fall $\rho \geq 2 \Leftrightarrow |z_0 - z_c| \geq 2R$:

$$E_M^p \leq A \left(\frac{1}{2}\right)^p .$$

Damit kann man für eine vorher festgelegte Genauigkeit ε die Anzahl p der zu evaluierenden Momente bestimmen.

Translationsoperatoren

Während des Algorithmus werden mehrere Expansionspunkte benötigt. Anstatt alle Momente für einen neuen Expansionspunkt $z_{c'}$ wieder neu zu berechnen, kann man die alten Momente von z_c „verschieben“.

Satz (Moment-to-Moment-Translation, M2M)

Es seien die Punkte $z_1, \ldots, z_m$ mit Ladungen $q(z_1), \ldots, q(z_m)$ gegeben. Weiterhin seien z_0, z_c und $z_{c'}$ derart, dass $|z_i - z_c| \ll |z_0 - z_c|$ und $|z_i - z_{c'}| \ll |z_0 - z_{c'}|$ für alle $i = 1, \ldots, m$. Dann gilt

$$M_k(z_{c'}) = \sum_{l=0}^{k} I_{k-l}(z_c - z_{c'}) M_l(z_c). \tag{4}$$

Moment-to-Moment-Translation

Beweis.

$$
\begin{aligned}
M_k(z_{c'}) &= \sum_{i=0}^{m} I_k(z - z_{c'}) q(z_i) \\
&= \sum_{i=0}^{m} I_k \left[(z - z_c) + (z_c - z_{c'}) \right] q(z_i) \\
&= \sum_{i=0}^{m} \sum_{l=0}^{k} I_l(z - z_c) I_{k-l}(z_c - z_{c'}) q(z_i) \\
&= \sum_{l=0}^{k} I_{k-l}(z_c - z_{c'}) M_l(z_c)
\end{aligned}
$$

Dabei haben wir die Binomische Formel in der dritten Gleichung verwendet. □

Bemerkung

Durch die endliche Zahl der Summanden wird keine weitere Fehlerquelle eingeführt.

Lokale Expansion

Statt des Expansionspunktes z_c verschieben wir jetzt den Quellpunkt z_0:

Satz (Lokale Expansion und Moment-to-Local-Translation)

Mit den Bedingungen der Multipolexpansion sei ein weiterer Punkt z_L nahe z_0, d.h. $|z_0 - z_L| \ll |z_L - z_c|$ gegeben. Dann gilt die folgende lokale Expansion

$$\sum_{i=1}^{m} G(z_0, z_i) q(z_i) = \sum_{l=0}^{\infty} L_l(z_L) I_l(z_0 - z_L), \tag{5}$$

wobei die lokalen Expansionskoeffizienten $L_l(z_L)$ durch die folgende Moment-to-Local-Expansion(M2L) gegeben sind:

$$L_l(z_L) = (-1)^l \sum_{k=0}^{\infty} O_{l+k}(z_L - z_c) M_k(z_c). \tag{6}$$

Lokale Expansion

Vorschau

Sind die lokalen Expansionskoeffizienten $L_l(z_L)$ einmal berechnet worden, ist die Gleichung (5)

$$\sum_{i=1}^{m} G(z_0, z_i) q(z_i) = \sum_{l=0}^{\infty} L_l(z_L) I_l(z_0 - z_L)$$

unabhängig von z_c. Wie bei der Moment-to-Moment-Translation lässt sich auch eine Local-to-Local-Translation ausdrücken. Ein weiterer Fall, wo bereits berechnete Terme wiederverwendet werden können. Die z_L werden auch wie die z_c später im Algorithmus Gitterpunkte darstellen.

Fehler der Moment-to-Local-Translation

Auch in (6) tritt eine endliche Summe auf. Es gilt:

Satz (Fehler der M2L Translation)

Für den Fehler E_L^p der Moment-to-Local-Expansion gilt:

$$E_L^p := \left| \sum_{i=1}^{m} G(z_0, z_i) q(z_i) - \sum_{l=0}^{p} L_l(z_L) I_l(z_0 - z_L) \right|$$
$$= \left| \sum_{l=p+1}^{\infty} L_l(z_L) I_l(z_0 - z_L) \right| \leq \frac{A\left[4e(p+\rho)(\rho+1)+\rho^2\right]}{\rho(\rho-1)} \left(\frac{1}{\rho}\right)^{p+1},$$

für alle $p \geq \max\{2, \frac{2\rho}{\rho-1}\}$, wobei e die Eulersche Zahl ist und A und ρ wie bisher definiert sind.

Local-to-Local-Translation

Ähnlich der Moment-to-Moment-Translation führen wir die Local-to-Local-Translation ein:

Satz (Local-to-Local-Translation, L2L)

Es sei $z_{L'}$ ein Punkt nahe z_L, d.h. $|z_0 - z_{L'}| \ll |z_0 - z_c|$. Dann gilt für eine lokale Expansion mit p Termen:

$$\sum_{i=1}^{m} G(z_0, z_i) q(z_i) \approx \sum_{l=0}^{p} L_l(z_L) I_l(z_0 - z_L) = \sum_{l=0}^{p} L_l(z_{L'}) I_l(z_0 - z_{L'}),$$

$$\text{wobei } L_l(z_{L'}) = \sum_{s=0}^{p-l} I_s(z_{L'} - z_L) L_{l+s}(z_L) \tag{7}$$

Bemerkung

Wie bereits bei der M2M-Translation (4) werden bei der L2L-Translation (7) nur endliche viele Terme addiert, also keine weiteren Fehler eingeführt.

Local-to-Local-Translation

Beweis.

$$\begin{aligned}
\sum_{l=0}^{p} L_l(z_L) I_l(z_0 - z_L) &= \sum_{l=0}^{p} L_l(z_L) I_l\left[(z_0 - z_{L'}) + (z_{L'} - z_L)\right] \\
&= \sum_{0 \le s \le l \le p} I_s(z_0 - z_{L'}) L_l(z_L) I_{l-s}(z_{L'} - z_L) \\
&= \sum_{l=0}^{p} \sum_{s=l}^{p} I_l(z_0 - z_{L'}) L_s(z_L) I_{s-l}(z_{L'} - z_L) \\
&= \sum_{l=0}^{p} I_l(z_0 - z_{L'}) \underbrace{\sum_{s=0}^{p-l} L_{l+s}(z_L) I_s(z_{L'} - z_L)}_{= L_l(z_{L'})}
\end{aligned}$$

Zwischenzusammenfassung

Uns stehen nun folgende Werkzeuge zur Verfügung:

$$\text{Momente: } M_k(z_c) = \sum_{i=1}^{m} I_k(z_i - z_c) q(z_i)$$

$$\text{Multipolexpansion: } \sum_{i=1}^{m} G(z_0, z_i) q(z_i) \approx \sum_{k=0}^{p} O_k(z_0 - z_c) M_k(z_c)$$

$$\text{M2M-Translation: } M_k(z_c') = \sum_{l=0}^{k} I_{k-l}(z_c - z_c') M_l(z_c)$$

$$\text{M2L-Translation: } L_l(z_L) \approx (-1)^l \sum_{k=0}^{p} O_{l+k}(z_L - z_c) M_k(z_c)$$

$$\text{Lokale Expansion: } \sum_{i=1}^{m} G(z_0, z_i) q(z_i) \approx \sum_{l=0}^{p} L_l(z_L) I_l(z_0 - z_L)$$

$$\text{L2L-Translation: } L_l(z_{L'}) = \sum_{s=0}^{p-l} I_s(z_{L'} - z_L) L_{l+s}(z_L)$$

FMM-Algorithmus

Der Fast Multipole-Algorithmus gliedert sich in mehrere Schritte:

1. Erstellen einer Quad-Tree-Struktur.
2. *Upward pass*.
3. *Downward pass*.
4. Berechnung der Potentiale.

Erstellen der Quad-Tree-Struktur

1. Lege alle Punkte in ein Quadrat. Dies bildet eine *Level* 0 *Zelle*.
2. Zerlege das Quadrat in Level 0 in vier Zellen (Quadrate) von Level 1.
3. Lege eine Zahl *maxl*, wie viele Punkte sich maximal innerhalb einer Zelle des Quad-Trees befinden dürfen, fest.
4. Für eine Zelle von Level l mache folgendes:
   Wenn sie mehr als *maxl* Punkte enthält, zerlege sie in vier *Tochterzellen* von Level $l + 1$. Sie ist jetzt eine *Vaterzelle*.
   Sonst bildet sie ein *Blatt* des Quad-Trees und wird nicht weiter zerlegt.
5. Fahre mit obigem Schritt fort, bis keine Zelle mehr in Quadrate zerlegt wird.

Erstellen der Quad-Tree-Struktur

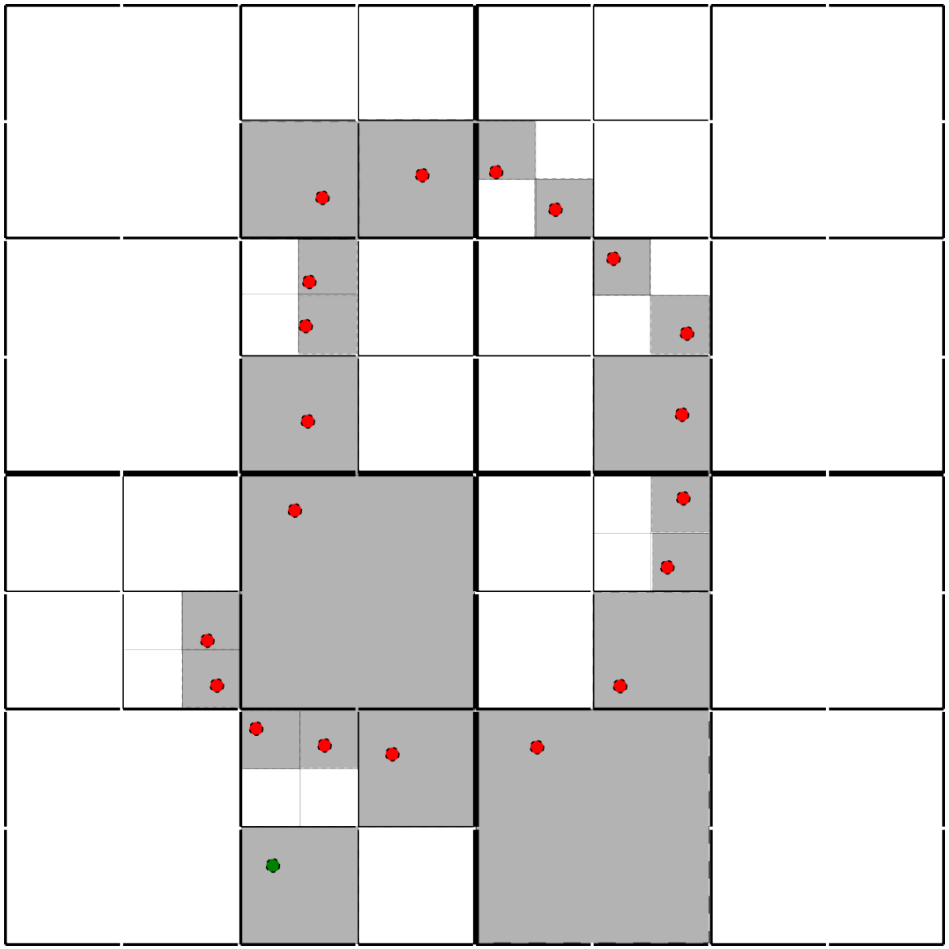


Abbildung 1 : Quad-Tree der geladenen Punkte (rot), eines Quellpunktes (grün) und Blätter (grau)

Upward pass

Berechnung aller Momente aller Zellenzentren von Level ≥ 2 mit p Termen, beginnend bei den Blättern:

- Handelt es sich bei der Zelle um ein Blatt, berechne das Moment der Zelle mittels der Gleichung (3) für alle geladenen Punkte innerhalb des Blattes, dabei ist z_c das Zentrum der Zelle.
- Handelt es sich um eine Vaterzelle, berechne ihr Moment mittels der Summation der M2M-Translationen (4) der Zentren ihrer Tochterzellen, dabei ist $z_{c'}$ das Zentrum der Vaterzelle und z_c das Zentrum einer Tochterzelle.

Upward pass

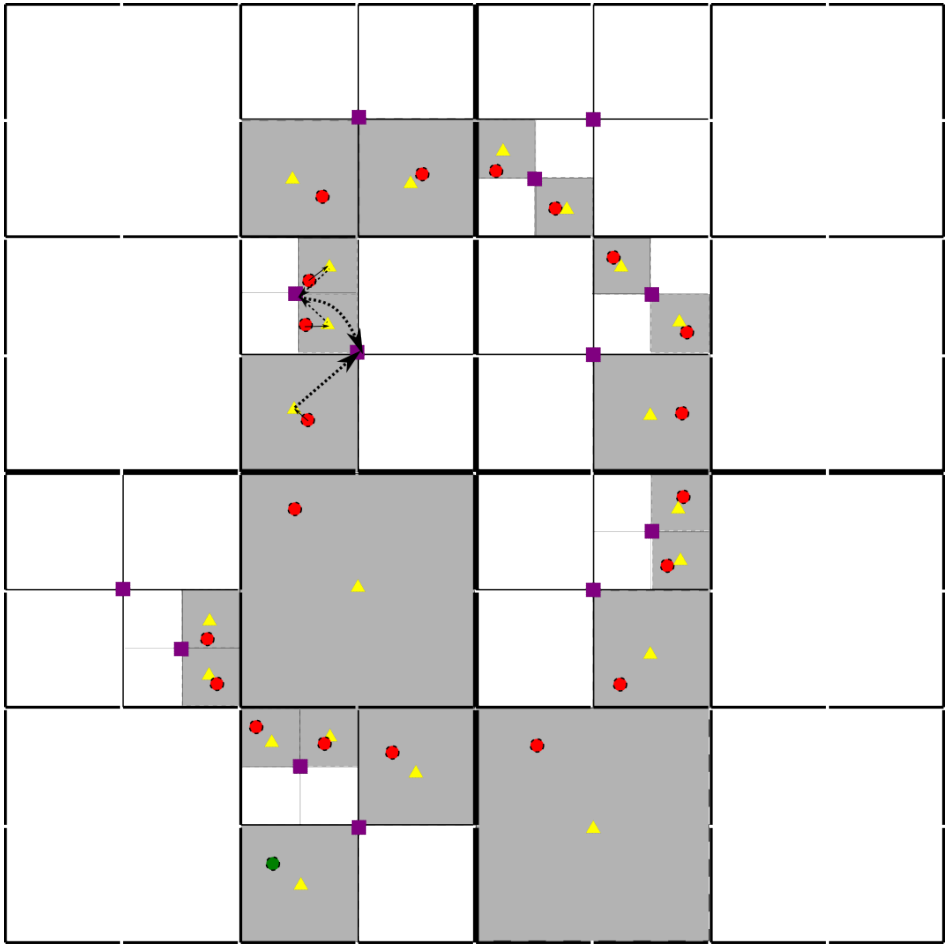


Abbildung 2 : Beispielhafter Upward pass von einer Blattzelle.
violett: Zentren von Vaterzellen, gelb: Zentren von Blättern,
$\rightarrow$: Moment direkt berechnet, $\dashrightarrow$: M2M-Translation

Downward pass

Hierfür benötigen wir noch etwas mehr Vokabular:

Definition

- Zwei Zellen heißen *benachbart in Level l*, wenn sie mindestens eine gemeinsame Ecke haben.
- Zwei Zellen heißen *gut separiert in Level l*, wenn sie auf Level l *nicht benachbart sind*, aber ihre *Vaterzellen in Level $l-1$* *benachbart* sind.
- Bezeichne die Menge aller gut separierten Zellen einer Zelle C in Level l als *Interaktionsliste* von C.
- Eine Zelle ist eine *Fernzelle* von C, wenn ihre jeweiligen Vaterzellen nicht benachbart sind. Dabei sind die größtmöglichen Vaterzellen die von Level 2.

Downward pass

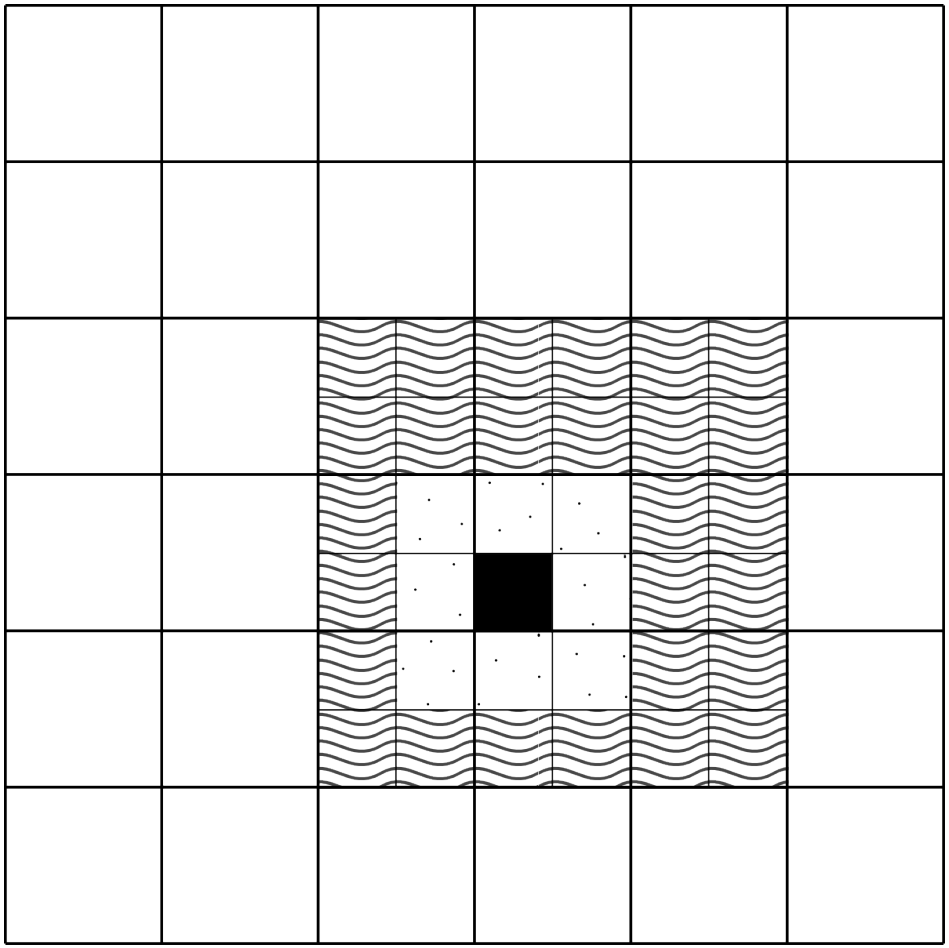


Abbildung 3 : schwarz: Zelle C, gepunktet: ihre benachbarten Zellen, gewellt: ihre Interaktionsliste, weiß: Fernzellen von C

Downward pass

Berechnung aller lokalen Expansionskoeffizienten aller Zellen startend bei Level 2 und die Baumstruktur nach unten zu allen Blättern folgend.

1. Der lokale Expansionskoeffizient einer Zelle C ist die Summe
   1. der Beiträge der Zellen von der Interaktionsliste von C und
   2. von allen Fernzellen.
2. 1.1 benutzt M2L-Translation (6), wobei die Momente die der Zellen der Interaktionsliste sind.
3. 1.2 benutzt L2L-Translation (7) um den lokalen Expansionspunkt vom Zentrum der Vaterzelle von C zum Zentrum von C zu verschieben.
4. Auf Level 2 wird für den lokalen Expansionskoeffizienten M2L-Translation verwendet. Dabei werden die Beiträge der in Level 2 nicht benachbarten Zellen aufsummiert.

Downward pass

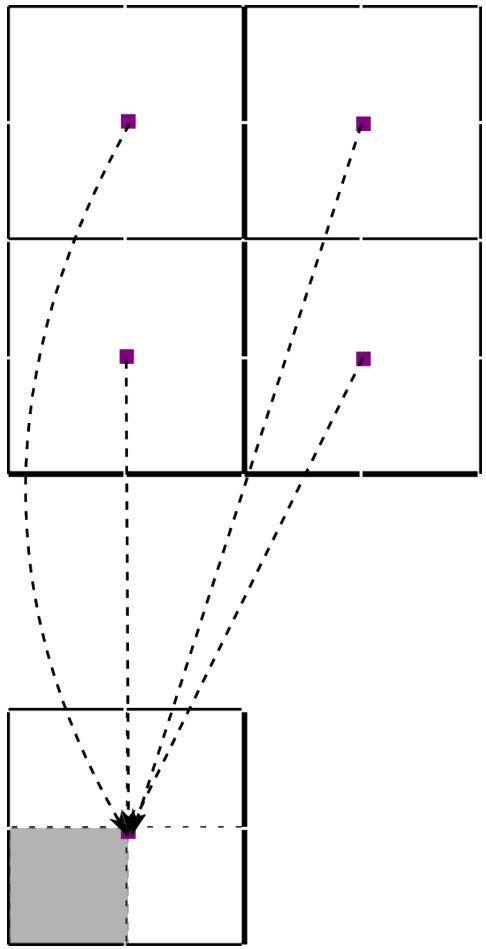


Abbildung 4 : Downward Pass bei Level 2. $\dashrightarrow$: M2L-Translation

Downward pass

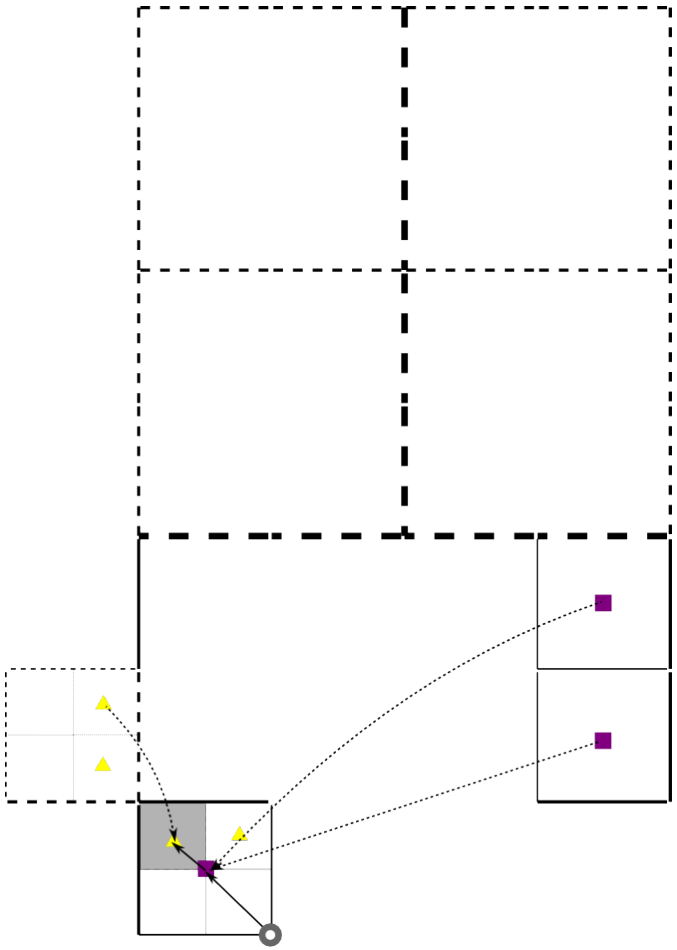


Abbildung 5 : Downward Pass bei Leveln 3 und 4. $\dashrightarrow$: M2L-Translation, $\rightarrow$: L2L-Translation, Kreis: Zellzentrum Level 2, Quadrat: Zellzentrum Level 3, Dreieck: Zellzentrum Level 4

Berechnung der Potentiale

Für die Punkte an denen das Potential berechnet werden soll, erfolgt nun die Berechnung dessen.
Sei z_0 in Blatt C. Dann berechnen wir

$$\Phi(z_0) = \Phi_{\text{nah}}(z_0) + \Phi_{\text{fern}}(z_0)$$

$$\sum_{i=1}^{m} G(z_0, z_i) q(z_i) = \underbrace{\sum_{z_i \text{ nahe } z_0} G(z_0, z_i) q(z_i)}_{\text{Nahwirkung}} + \underbrace{\sum_{z_i \text{ fern } z_0} G(z_0, z_i) q(z_i)}_{\text{Fernwirkung}},$$

wobei wir für die z_i, die sich in C oder einen seiner acht Nachbarzellen befinden, die *Nahwirkung* mit *direkter* Summation auswerten.
Hingegen verwenden wir für die Fernwirkung, also Beiträge aus der Interaktionsliste von C und fernen Zellen, die Lokalexpansion der Zelle C, also (5), wobei z_L das Zentrum von C ist.

Laufzeitanalyse

Zurück zur am Anfang gestellten Frage. Sind wir mit dem FMM-Algorithmus wirklich schneller?

Satz (Laufzeit des FMM-Algorithmus)

Sei N die Anzahl der geladenen Teilchen und N die Anzahl der Punkte deren Potentiale, verursacht durch die geladenen Teilchen, berechnet werden sollen. Sei p die Anzahl der Terme die für die Multipol- und Lokalexpansionen verwendet werden und maxl die maximale Anzahl von geladenen Teilchen in einem Blatt des Quad-Trees. Dann ist die Berechnung aller Potentiale mit dem hier vorgestellten Fast-Multipole-Method-Algorithmus in Laufzeit $\mathcal{O}(N)$ möglich.

Laufzeitanalyse

Beweis.

- N_B = Anzahl der Blätter im Quad-Tree $\approx N/maxl = \mathcal{O}(N)$
- N_Z = Anzahl der Zellen im Quad-Tree $\approx N_B \cdot (1 + \frac{1}{4} + \frac{1}{4^2} + \ldots) \leq N_B \cdot \frac{4}{3} = \mathcal{O}(N)$
- Anzahl benachbarter Zellen = 9; Anzahl der Zellen in der Interaktionsliste = 27 (2D)
- Anzahl Operationen, um Multipolmomente zu berechnen = $p \cdot maxl \cdot N_B = \mathcal{O}(N)$
- Anzahl Operationen im Upward pass = $N_Z \cdot 4 \cdot p^2 = \mathcal{O}(N)$
- Anzahl Operationen im Downward pass = $N_Z \cdot \left[p^2(L2L) + 27 \cdot p^2(M2L)\right] = \mathcal{O}(N)$
- Anzahl Operationen der Lokalexpansionen = $N \cdot p = \mathcal{O}(N)$
- Anzahl Operationen der direkten Berechnungen der Potentiale der Nahwirkung = $N \cdot 9 \cdot maxl = \mathcal{O}(N)$

Alle Abschätzungen haben höchstens Laufzeit $\mathcal{O}(N)$, damit ist die Gesamtlaufzeit $\mathcal{O}(N)$. □

Literaturverzeichnis

[1] V. Rohklin L. Greengard.
A fast algorithm for particle simulations.
Journal of Computational Physics, 73:325–348, 1987.

[2] Yijun Liu.
Fast Multipole Boundary Element Method.
Cambridge University Press, 2014.