

Verformung einer Platte

Hubert Weißmann

9. Oktober 2013

Inhaltsverzeichnis

1	Einführung	1
2	Diskretisierung und Aufstellung des Gleichungssystems	1
3	Vorbetrachtungen zur Lösung	3
4	zyklische Block-Reduktion	3
5	Stabilisierung: der Buneman-Algorithmus	4
5.1	Zur Implementierung	6
6	Berechnung von u mit $A^{(r)}u = v$	6

1 Einführung

Reale Festkörper sind stets verformbar; wenn auch meist nur so geringfügig, dass es kaum wahrnehmbar ist. Jede Kraft, die von außen auf einen Festkörper einwirkt, verformt diesen ein wenig. Wie diese Verformungen im Fall einer rechteckigen Platte aussehen, ist Gegenstand der vorliegenden Arbeit.

Herleitung der Differentialgleichung!!

Mathematisch wird die Platte durch eine Funktion $u(x, y)$ auf einem Rechteck $D = [0, a] \times [0, b]$ beschrieben, welche ohne darauf einwirkende Kräfte in der xy -Ebene liege ($u \equiv 0$). Die Verformung durch eine Kraft f wird durch das Randwertproblem

$$\begin{aligned} \frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) &= -k f(x, y) & x, y \in D \\ u(x, y) &= 0 & \forall x, y \in \partial D, \end{aligned} \tag{1}$$

die so genannte „Poisson-Gleichung“ beschrieben. Die Größe k ist eine Relationsgröße, welche im Folgenden auf $k \equiv 1$ festgelegt werden soll.

2 Diskretisierung und Aufstellung des Gleichungssystems

Das gegebene Gleichungssystem ist grundsätzlich analytisch lösbar, sofern f hölderstetig ist [Dzi10, S. 36]. Der analytische Weg ist jedoch sehr aufwändig und muss für jede gegebene Gewichtsfunktion, auch wenn sich diese nur geringfügig von bereits berechneten Lösungen unterscheidet, neu berechnet werden. Um dies zu umgehen und das Problem (1) mit Hilfe eines Computers berechenbar zu machen, ist es sinnvoll, sich eine direkte numerische Lösungsmethode zu überlegen. Hierfür wird der Bereich

D in ein Gitter mit konstanter Schrittweiten k in x -Richtung und h in y -Richtung unterteilt

$$D_h = \{(x, y) \in D | x = nh, y = mh; n = 1, 2, \dots, N-1, m = 1, 2, \dots, M-1\}$$

$$\partial D_h = \{(x, y) \in \partial D | x \in \{Nh, 0\}, y = mh, m \in \mathbb{Z}; y \in \{0, Mh\}, x = nh, n \in \mathbb{Z}\}.$$

Die Differentiale werden nun durch die Differenzenquotienten der Gitterpunkte apoximiert. Dabei ist es sinnvoll die zentralen Differenzen

$$\frac{\partial u(x, y)}{\partial x} \approx \frac{u(x+h, y) - u(x-h, y)}{2h}$$

zu betrachten. Die zweite Ableitung nach x bzw. y wird entsprechend zu

$$\frac{\partial^2 u(x, y)}{\partial x^2} = \frac{u(x+2h, y) + u(x-2h, y) - 2u(x, y)}{4h^2}$$

$$\frac{\partial^2 u(x, y)}{\partial y^2} = \frac{u(x, y+2h) + u(x, y-2h) - 2u(x, y)}{4h^2}$$

Es ist offensichtlich sinnvoll, in den oberen Gleichungen die Schrittweite zu halbieren. Da die Differentialgleichung nur in dem offenen Gebite D galt, muss auch hier die Randbedingung entsprechend eingepasst werden.

Randwert!!! wie kommen die Rand-Bedingungen mit hinein???

und wir können (1) durch das Gleichungssystem

$$\frac{u(x_{i+1}, y_j) + u(x_{i-1}, y_j) - 2u(x_i, y_j)}{k^2} + \frac{u(x_i, y_{j+1}) + u(x_i, y_{j-1}) - 2u(x_i, y_j)}{h^2} = -kf(x_i, y_j)$$

für $i, j \in D_h = \{(ih, jh) | i = 1, \dots, N-1; j = 1, \dots, M-1\}$

beschreiben. Der mit dieser Abschätzung gemachte Fehler U lässt sich bei gleichen Abständen k, h durch die Maximumsnorm der vierten Ableitung von u als $U(u) \leq \frac{h^2}{12} \|u^{(4)}\|_\infty$ abschätzen [L A00, S. 20].

Dieses System von $N \times M$ Gleichungen kann zu einer Matrix-Gleichung umgeschrieben werden, indem die Funktionswerte $u_{i,j} = u(x_i, y_j)$, $f_{ij} = f(x_i, y_j)$ in einen Vektor geschrieben werden. Zweckmäßig ist hier eine alphabetische Anordnung. Dadurch erhalten wir ein lineares Gleichungssystem

$$\mathcal{L}U = F \tag{2}$$

wobei \mathcal{L} eine Blockmatrix mit $M \times M$ Blöcken der Form

$$\mathcal{L} = \begin{bmatrix} A & I & & & \\ I & A & I & & \\ & I & A & I & \\ & & \ddots & \ddots & \ddots \\ & & & I & A \end{bmatrix}$$

mit der $N \times N$ -Einheitsmatrix I und

$$A = \begin{bmatrix} -2(1+\lambda) & \lambda & & & \\ \lambda & -2(1+\lambda) & \lambda & & \\ & \lambda & -2(1+\lambda) & \lambda & \\ & & \vdots & \vdots & \vdots \\ & & & \lambda & -2(1+\lambda) \end{bmatrix}$$

mit $\lambda = \left(\frac{k^2}{h^2}\right)$ [Dor70, S. 4] ist. U und F sind die Blockvektoren

$$U = \begin{bmatrix} u_1 \\ \vdots \\ u_j \\ \vdots \\ u_M \end{bmatrix} \quad F = -k^2 \begin{bmatrix} f_1 \\ \vdots \\ f_j \\ \vdots \\ f_M \end{bmatrix} \quad \text{mit} \quad u_j = \begin{bmatrix} u_{1j} \\ \vdots \\ u_{ij} \\ \vdots \\ u_{Nj} \end{bmatrix} \quad f_j = \begin{bmatrix} f_{1j} \\ \vdots \\ f_{ij} \\ \vdots \\ f_{Nj} \end{bmatrix}.$$

3 Vorbetrachtungen zur Lösung

Die Lösung der Gleichung

$$LU = F$$

erscheint auf den ersten Blick eine einfache Aufgabe, erweist sich jedoch als komplexeres Problem, wenn man die Dimension des Problems bedenkt. Der Gaußalgorithmus ist wegen seines Rechenaufwandes von $\frac{2}{3}n^3 + O(n^2)$ bereits für verhältnismäßig große Gitterabstände nicht mehr praktikabel. Grundsätzlich bieten sich bei schwachbesetzten Matrizen wie der Vorliegenden iterative Verfahren wie das Gauß-Seidel- oder Jacobi-Verfahren an. Da jedoch die Spektralnorm sehr dicht an 1 ist, haben sowohl das Gauß-Seidel- als auch das Jacobi-Verfahren eine Konvergenz von [Hac04, S. 101]

$$\rho = \cos^2(\pi h) \approx 1 - \pi^2 h^2.$$

Diese lässt sich zwar durch eine andere Anordnung bzw. eine Optimierung des Verfahrens (SOR-Verfahren) verdoppeln, ist jedoch nach wie vor sehr nahe an 1 und damit sehr langsam. Wegen dieser schlechten Eigenschaften der vorliegenden Matrix bezüglich klassischer Lösungsmethoden wurde eine ganze Klasse von Lösungsmöglichkeiten entwickelt, die genau auf diese Matrix abgestimmt ist: Die so genannten Fast-Poisson-Solver. Von diesen Verfahren soll im Folgenden ein zyklisches Verfahren vorgestellt werden.

4 zyklische Block-Reduktion

Im folgenden soll die Gleichung durch den Diskretisierungsfaktor $\frac{1}{k^2}$ geteilt werden, sodass die diskrete Gewichtsfunktion nun

$$f \rightarrow k^2 f$$

ist. Bei Gleichungssystemen, die sich in einer regelmäßigen Blockstruktur schreiben lassen, lässt sich dieses in Untersysteme, die die Größe dieser Blöcke haben, unterteilen. In dem Fall von (2) erhalten wir zunächst [BL 70, S. 4]:

$$\begin{aligned} u_1 + Au_2 &= f_1 \\ u_{j-1} + Au_j + u_{j+1} &= f_j \quad 2 \leq j \leq M-1 \\ Au_{M-1} + u_M &= f_M \end{aligned} \tag{3}$$

Da im Folgenden die Anzahl der Dimensionen der Blockmatrix (nicht der Blöcke) reduziert werden soll, nehmen wir zunächst die Dimension N der Blockmatrix L als ungerade an, was sich, da diese Dimension frei wählbar ist, erfüllen lässt. Nun Schreiben wir (3) um als

$$\begin{aligned} u_{j-2} + Au_{j-1} + u_j &= f_{j-1} \\ u_{j-1} + Au_j + u_j &= f_j \\ u_j + Au_{j+1} + u_{j+1} &= f_{j+1} \end{aligned}$$

, multiplizieren die mittlere Gleichung mit $-A$ und addieren daraufhin die drei Gleichungen. Nun bleiben nur noch Elementte mit u_j und $u_{j\pm 2}$ übrig. Ist j ein gerader index, bleiben also nur gerade Indizes in u übrig und wir erhalten das neue Gleichungssystem

$$\begin{bmatrix} 2 \cdot I - A^2 & I & & & \\ I & 2 \cdot I - A^2 & I & & \\ & I & 2 \cdot I - A^2 & I & \\ & & \ddots & \ddots & \ddots \\ & & & I & 2 \cdot I - A^2 \end{bmatrix} \begin{bmatrix} u_2 \\ u_4 \\ u_6 \\ \vdots \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} f_1 + f_3 - Af_2 \\ f_3 + f_5 - Af_4 \\ f_5 + f_7 - Af_6 \\ \vdots \\ f_{N-2} + f_N - Af_{N-1} \end{bmatrix}$$

dessen Dimension nun halb so groß ist. Die dazwischenliegenden u_j , deren Index ungerade ist, erhält man durch umstellen aus (3) durch Lösen des Gleichungssystems

$$\begin{bmatrix} A & 0 & 0 & \dots \\ 0 & A & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots \\ & & 0 & A \end{bmatrix} \begin{bmatrix} u_1 \\ u_3 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} f_1 - u_2 \\ f_3 - (u_2 + u_4) \\ \vdots \\ f_M - u_{M-1} \end{bmatrix}$$

Hier ergibt sich aus der Lösung der u_j mit ungeradem Indize also aus der Lösung der u_j mit geraden Indizes, welche wiederum mit einem Gleichungssystem gelöst werden, welches eine die gleiche Form wie das vorige hat. Daher definieren wir nun

$$\begin{aligned} A^{(0)} &= A & f^{(0)} &= f \\ A^{(r+1)} &= 2 \cdot I - \left(A^{(r)}\right)^2 & f_j^{(r+1)} &= f_{j-1}^{(r)} + f_{j+1}^{(r)} - A f_j^{(r)} \end{aligned}$$

und können das Gleichungssystem nun allgemein schreiben als

$$\begin{aligned} L^{(r)} u^{(r)} &= f^{(r)} \quad \text{mit} \\ L &= \begin{bmatrix} A^{(r)} & I & & \\ I & A^{(r)} & I & \\ & \vdots & \vdots & \vdots \\ & & I & A^{(r)} \end{bmatrix} & u^{(r)} = \begin{bmatrix} u_{1 \cdot 2^r} \\ \vdots \\ u_{j \cdot 2^r} \\ \vdots \\ u_{(2^{k+1}-r-1)2^r} \end{bmatrix} \\ f_k^{(r+1)} &= f_{k-1}^{(r)} + f_{k+1}^{(r)} - A f_k^{(r)} & A^{(r+1)} &= 2 \cdot I - \left(A^{(r)}\right)^2 \end{aligned} \tag{4}$$

und

$$\begin{bmatrix} A^{(r-1)} & 0 & \dots \\ 0 & A^{(r-1)} & 0 \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_{j \cdot 2^{r-2^{r-1}}} \\ \vdots \end{bmatrix} = \begin{bmatrix} f_1^{(r-1)} \\ \vdots \\ f_{2^r-2^{r-1}}^{(r-1)} \text{ hier fehlt der Rest!} \end{bmatrix}$$

Dabei muss stets die Blockdimension ungerade sein, wodurch wir ein Kriterium für die Dimension und damit für die Einteilung des Gebietes (zumindest in x-Richtung) erhalten: Diese sollte $N = 2^k - 1$ sein [BL 70, S. 5].

Damit haben wir ein Verfahren entwickelt, welches das ursprüngliche Problem der Größe $M \times N$ in M kleinere Systeme unterteilt und mit einer Komplexität von lediglich $O(n) = \frac{9}{2} N^2 \log N$ [Dor70, S. 9] löst.

5 Stabilisierung: der Buneman-Algorithmus

Auch wenn das oben vorgestellte Verfahren einige große Vorteile mit sich bringt, muss es doch zur praktischen Anwendung nochmals umformuliert werden, da die direkte Anwendung des Algorithmus' numerische Instabilitäten aufweist. [W G97, S. 10] Um diese Instabilitäten zu umgehen, reicht es bei dem vorliegenden System interessanter Weise, die rechte Seite umzuformulieren. Mathematisch bleibt es also bei der Gleichung, jedoch wird die Rekursionsformel

$$f_j^{(r+1)} = f_{j-2^r}^{(r)} + f_{j+2^r}^{(r)} - A^{(r)} f_j^{(r)}$$

mit Hilfe der Beziehung

$$A^{(r+1)} \left(A^{(r)}\right)^{-1} = 2 \left(A^{(r)}\right)^{-1} - A^{(r)}$$

umgeschrieben [Coh94, S. 13] zu

$$f_j^{(r+1)} = A^{(r+1)} \underbrace{\left(A^{(r)} \right)^{-1} f_j^{(r)}}_{=: p_j^{(r+1)}} + \underbrace{f_{j-2^r}^{(r)} + f_{j+2^r}^{(r)} - 2 \left(A^{(r)} \right)^{-1} f_j^{(r)}}_{=: q_j^{(r+1)}}.$$

Aus den Zusammenhängen

$$\begin{aligned} f_j^{(r+1)} &= f_{j-2^r}^{(r)} + f_{j+2^r}^{(r)} - A^{(r)} f_j^{(r)} \\ &= \left(2I - \left(A^{(r)} \right)^2 \right) \left(p_j^{(r)} - \left(A^{(r)} \right)^{-1} \left(p_{j-2^r}^{(r)} + p_{j+2^r}^{(r)} - q_j^{(r)} \right) \right) + q_{j-2^r}^{(r)} + q_{j+2^r}^{(r)} \\ &\quad - 2 \left(p_j - \left(A^{(r)} \right)^{-1} \left(p_{j-2^r}^{(r)} + p_{j+2^r}^{(r)} - q_j^{(r)} \right) \right) \\ &= A^{(r+1)} p_j^{(r+1)} + q_j^{(r+1)} \end{aligned} \quad (5)$$

lässt sich nun eine Rekursionsformel für p_j und q_j ableiten [Coh94, S. 14]:

$$p_j^{(r+1)} = p_j^{(r)} - \left(A^{(r)} \right)^{-1} \left(p_{j-2^r}^{(r)} + p_{j+2^r}^{(r)} - q_j^{(r)} \right) \quad (6)$$

$$q_j^{(r+1)} = q_{j-2^r}^{(r)} + q_{j+2^r}^{(r)} - 2p_j^{(r)} - 2 \left(A^{(r)} \right)^{-1} \left(p_{j-2^r}^{(r)} + p_{j+2^r}^{(r)} - q_j^{(r)} \right) \quad (7)$$

$$= q_{j-2^r}^{(r)} + q_{j+2^r}^{(r)} - 2p_j^{(r+1)} \quad (8)$$

wobei

$$p_j^{(0)} = 0 \quad q_j^{(0)} = f_j$$

ist [Coh94, S. 14]. Dabei lässt sich die explizite Berechnung der $p_j^{(r)}$ einsparen, da sich diese aus (8) durch Umstellen ergibt:

$$p_j^{(r+1)} = \frac{1}{2} \left(q_{j-2^r}^{(r)} + q_{j+2^r}^{(r)} - q_j^{(r+1)} \right) \quad (9)$$

Dadurch ist zwar ein etwa doppelt so hoher Rechenaufwand nötig, gleichzeitig halbiert sich jedoch der Speicheraufwand [Coh94, ??]. Die Rekursionsvorschrift (8) lässt sich mit Hilfe von (6), (9) nun in eine Dreitermrekursion umformen:

$$\begin{aligned} q_j^{(r+1)} &= q_{j-2^r}^{(r)} + q_{j+2^r}^{(r)} - q_{j-2^{r-1}}^{(r-1)} - q_{j+2^{r-1}}^{(r-1)} + q_j^{(r)} \\ &\quad + \left(A^{(r)} \right)^{-1} \left(q_{j-3 \cdot 2^{r-1}}^{(r-1)} + q_{j+3 \cdot 2^{r-1}}^{(r-1)} + q_{j+2^{r-1}}^{(r-1)} + q_{j-2^{r-1}}^{(r-1)} - q_{j-2^r}^{(r)} - q_{j+2^r}^{(r)} - 2q_j^{(r)} \right) \end{aligned} \quad (10)$$

Damit lässt sich der Buneman-Algorithmus nun formulieren:

1. initialisiere $q_j^{(0)} = f_n$, $q_j^{(1)} = f_{j-1} + f_{j+1} - 2A^{-1}f_j$ sowie $q_0^{(r)} = q_N^{(r)} = 0 \quad \forall r$
berechne

$$\begin{aligned} q_j^{r+1} &= q_{j-2^r}^{(r)} + q_{j+2^r}^{(r)} - q_{j-2^{r-1}}^{(r-1)} - q_{j+2^{r-1}}^{(r-1)} + q_j^{(r)} + \\ &\quad \left(A^{(r)} \right)^{-1} \left(q_{j-3 \cdot 2^{r-1}}^{(r-1)} + q_{j+3 \cdot 2^{r-1}}^{(r-1)} + q_{j+2^{r-1}}^{(r-1)} + q_{j-2^{r-1}}^{(r-1)} - q_{j-2^r}^{(r)} - q_{j+2^r}^{(r)} - 2q_j^{(r)} \right) \end{aligned} \quad (11)$$

wobei $r \in \{1, 2, \dots, k-1\}$, $j \in \{k \cdot 2^r \mid k \in \mathbb{N}, j < 2^{(k+1)} - 2^{(r+1)}\}$ ist.

2. Berechne nun die Lösung u_j für $j = k \cdot 2^r \mid k \leq N \in \mathbb{N}$ durch Umstellen von 3 und Einsetzen von f_j :

$$u_j = \frac{1}{2} \left(q_{j-2^{r-1}}^{(r-1)} + q_{j+2^{r-1}}^{(r-1)} - q_j^{(r)} \right) - \left(A^{(r)} \right)^{-1} \left(u_{j-2^r} + u_{j+2^r} - q_j^{(r)} \right) \quad \text{wobei } u_0 = u_{2^{k+1}} = 0. \quad (12)$$

wobei $r \in k, k-1, \dots, 0$, $j \in 2^{r-1}, 2^{r-3}, \dots, 2^{k+1} - 2^{r-1}$

5.1 Zur Implementierung

Im Algorithmus ist es nicht sinnvoll, die Gitterpunkte von U und F in vektorielle Form zu bringen, sondern vielmehr mit den Spaltenvektoren (oder Zeilenvektoren) der Matrizen zu arbeiten. Neben der numerischen Stabilität bietet der Buneman-Algorithmus damit noch weitere Vorteile bei der Implementierung. So ist sein Speicheraufwand sehr gering, da, wie bei näherer Betrachtung der Rekursionsformeln (11) und (12) deutlich wird, die Matrix F zunächst durch die $q_j^{(r)}$ und diese im zweiten Teil durch U überschrieben werden können. Deshalb wird nur eine Matrix $M \times N$ benötigt und auch die Matrix $A^{(r)}$ kann in expliziter Form vermieden werden. Wie im kommenden Abschnitt gezeigt wird, lässt sich das Blocksystem mit lediglich zwei zusätzlichen Vektoren als tridiagonalsystem lösen. Dem steht jedoch ein Rechenaufwand von

RECHENAUFWAND

gegenüber. Es sei jedoch darauf hingewiesen, dass dieser im Vergleich zu anderen Lösungssystemen recht gering ist QUELLE QUELLE QUELLE QUELLE QUELLE

6 Berechnung von u mit $A^{(r)}u = v$

Auch die einzelblöcke sind noch große Matrizen, weshalb hier die einfache Lösung mit Hilfe des Gauß-Algorithmus' nicht sinnvoll ist. Da jedoch $A^{(r)}$ keine Tridiagonalstruktur mehr hat und große Elemente hat [W G97, S. 9], müssen andere Eigenschaften der Matrix gesucht werden, die es möglich machen, die Gleichung möglichst einfach zu lösen.

Hierzu betrachten wir nochmals die Rekursionsformel (4):

$$A^{(r+1)} = 2I - \left(A^{(r)}\right)^2$$

Offensichtlich ist $A^{(r)}$ ein Polynom $2r$ -ten Graden in A , für das wir schreiben:

$$P(x) = x \quad P_{2r+1} = 2 - P_{2r}^2(x)$$

In dieser Form lässt sich eine Ähnlichkeit zu dem Tschebyscheff-Polynom feststellen, dessen Dreitermrekursion allgemein

$$T_0(t) = 1 \quad T_1(t) = t \quad T_{k+l}(t) = 2T_l(t) \cdot T_k(t) - T_{k-l}(t)$$

lautet [W G97, S. 10]. Für $k = l$ ergibt sich also:

$$T_{2k}(t) = 2T_k^2(t) - 1$$

Durch Multiplikation mit -2 und der Substitution $k = 2^r$ erhalten wir also eine Beziehung zwischen den Tschebyscheff-Polynomen und dem Polynom $A^{(r)}$:

$$P_{2r}(x) = 2T_{2r}\left(-\frac{x}{2}\right)$$

Damit sind nun auch die Nullstellen von $P(x)$

$$\lambda_i = -2 \cos\left(\frac{2i-1}{2^{r+1}}\pi\right)$$

bekannt und wir können, da die Nullstellen ein Polynom bis auf einen Faktor genau bestimmen¹, $A^{(r)}$ explizit ausdrücken:

$$\begin{aligned} P_{2r} &= -\prod_{i=1}^{2^r} (x - \lambda_i) \\ A^{(r)} &= -\prod_{i=1}^{2^r} (A - \lambda_i I) \end{aligned} \tag{13}$$

¹Satz von XY

!!! Achtung mit u, v. Das wird hier anscheinend noch vertauscht!!!

Nun können wir direkt die Inverse $\left(A^{(r)}\right)^{(-1)}$ berechnen. Dazu betrachten wir die Inverse des Polynoms $P(x)$ und führen mit ihm eine Partialbruchzerlegung durch.

$$\frac{1}{P_{2^r}(x)} = \frac{1}{\prod_{i=1}^{2^r} (x - \lambda_i)} = \sum_{i=1}^{2^r} \frac{c_i^{(r)}}{x - \lambda_i} \text{ mit } c_i = \frac{1}{P'_{2^r}(\lambda_i)} = - \prod_{j=1, j \neq i}^{2^r} (\lambda_j - \lambda_i)^{-1}$$

Nun gilt:[W G97, S. 10]

$$v = - \sum_{i=1}^{2^r} \left(\prod_{\substack{j=1 \\ j \neq i}}^{2^r} (\lambda_j - \lambda_i)^{-1} \right) (A - \lambda_i I)^{-1} u \quad (14)$$

Im Gegensatz zur obigen Gleichung ist die hier auftretende Matrix $A - \lambda_i I$ tridiagonal. Daher ist es hier sinnvoll, die Gleichung (14) umzustellen und in der Form

$$(A - \lambda_i I) z_i = \prod_{\substack{j=1 \\ j \neq i}}^{2^r} \frac{u}{\lambda_j - \lambda_i} \quad i = 1, 2, \dots, 2^r \quad (15)$$

$$v = \sum_{i=1}^{2^r} z_i$$

zu lösen.

Dieses Problem ist nun die Lösung eines linearen Gleichungssystems mit tridiagonaler Struktur, welches mit Hilfe der LR-Faktorisierung ² mit linearem Aufwand lösbar ist.

Interessanter Weise zeigt die Implementierung, dass diese Methode nicht, wie anzunehmen, zu einer schnelleren Lösung führt. Lediglich der Speicherverbrauch ist ein wenig geringer als die einfache Lösung:

$$U = \left(A^{(r)}\right)^{-1} F$$

mit expliziter Berechnung von $A^{(r)}$.

WIE GENAU IST DAS ZU ERKLÄREN????

Literatur

- [BL 70] C.W. Nielson B.L. Buzbee G.H. Golub. *On Direct Methods for Solving Poisson's Equations*. 1970. URL: <http://www.jstor.org/stable/2949380> [Stand:04.09.2013].
- [Coh94] S. Cohan. *Cyclic Reduction*. Techn. Ber. unknown, 1994. URL: <http://robotics.stanford.edu/~scohen/cr.ps> [Stand:24.09.2013].
- [Dor70] F. W. Dorr. *The Solution of the Discrete Poisson Equation on a Rectangle*. 1970. URL: <http://www.jstor.org/stable/2029223> [Stand:07.09.2013].
- [Dzi10] G. Dziuk. *Theorie und Numerik partieller Differentialgleichungen*. Berlin (u.a.): De Gruyter Studium, 2010. ISBN: 978-3-11-014843-5.
- [Hac04] W. Hackbusch. *Iterative Loesung großer Gleichungssysteme*. 2004. URL: <http://www.mis.mpg.de/scicomp/Fulltext/ggl.ps> [Stand:09.08.2013].
- [L A00] P. Knabner L. Angermann. *Numerik partieller Differentialgleichungen. Eine anwendungsorientierte Einführung*. Berlin (u.a.): Springer-Verlag, 2000. ISBN: 3-540-66231-6.
- [W G97] G. H. Golub W. Gander. *Cyclic Reduction- History and Applications*. Techn. Ber. ETH Zürich, Stanford University, 1997. URL: <http://www.inf.ethz.ch/personal/gander/papers/cyclic.pdf> [Stand:24.09.2013].

²SATZ 2.27 ????????? Vorlesung Numerische Mathematik I von Prof. Neymeyer, WS 2012/13