Verformung einer Platte

Hubert Weißmann

3. November 2013

Inhaltsverzeichnis

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 bei unterschiedlichen Gewichten aussehen, ist Gegenstand der vorliegenden Arbeit.
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 ($u \equiv 0$). Dabei soll sie an den Rändern fixiert sein, sodass hier keine Auslenkung vorkommt. Die Verformung wird durch eine Kraft f beschrieben. Aus diesen Bedingungen lässt sich nun das Randwertproblem

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

die „Poisson-Gleichung“ mit dem „Dirichlet-Randwert-Problem“, formulieren. Die Größe k ist eine Relationsgröße, welche im Folgenden auf $k = 1$ 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 durch ein Gitter mit $(N+1)\times(M+1)$ gleich großen Schritten der Länge k in x-Richtung und h in y-Richtung ersetzt:

$$D_h = \{(x,y) \in D | x = nh, y = mh;\ n = 1,2,\ldots,N-1\, m = 1,2,\ldots,M-1\}$$
$$\partial D_h = \{(x,y) \in \partial D | x \in \{(N+1)h, 0\}\ y = mh,\ m \in \mathbb{Z};\ y \in \{0,(M+1)h\}\ x = nh, n \in \mathbb{Z}\}.$$

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, mit welchen die Differentialgleichung mit der Festlegung von oben, dass $k = 1$, durch

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

approximiert wird. Analog erhalten wir auf ∂D_h die Gleichungen

$$u(x_N,y_j) = u(x_0,y_j) = u(x_i,y_M) = u(x_i,y_0) = 0$$

und wir können (1) durch das Gleichungssystem

$$\frac{u(x_{i+1},y_j) + u(x_{i-1},y_j) - 2u(x_i,y_j)}{k^2} + \frac{u(x_i,y_{j+1}) + u(x_i,y_{j-1}) - 2u(x_i,y_j)}{h^2} = -f(x_i,y_j)$$
$$\text{für } i,j \in D_h = \{(ih,jh) | i = 1,...,N-1; j = 1,...,M-1\}$$

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

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

3 Vorbetrachtungen zur Lösung

Die Lösung der Gleichung

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

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 $\frac{2}{3}n^3 + O(n^2)$ [ua00, !!] bereits für verhältnismäßig große Gitterabstände nicht mehr praktikabel (siehe hierzu die Rechenzeiten in 6). Grundsätzlich bieten sich bei schwachbesetzten Matrizen wie der Vorliegenden iterative Verfahren wie das Gauß-Seidel- oder Jacobi-Verfahren an. Allerdings zeigt sich, dass die Spektralnorm der Iterationsmatrix der Jacobi-Iteration $(I - D^{-1}A)$ mit der Einheitsmatrix I und der Diagonalmatrix D von A eine Spektralnorm von [Hac04, S. 101]

$$\rho\left(I - D^{-1}A\right) = \cos^2(\pi h) \approx 1 - \pi^2 h^2.$$

hat. Da jedoch die Konvergenzgeschwindigkeit dieser Verfahren nach der Formel [ua00, S.!?] ist, ist hier eine sehr hohe Anzahl an Iterationen nötig, um ein akzeptabel genähertes Ergebnis zu erhalten. Auch das Gauss-Seidel-Verfahren konvergiert ähnlich langsam (etwa doppelt so schnell) [Hac04, S. 101].
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 ganannten Fast-Poissen-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

$$F \to k^2 F$$

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]:

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

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 Elemtente mit u_j und $u_{j\pm2}$ 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 großist. Die dazwischenliegenden u_j, deren Index ungerade ist, erhält man durch umstellen aus (3) durch Lösen des Gleichungssystems

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

Hier ergibt sich aus der Lösung der u_j mit ungeradem Indizex 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

$$A^{(0)} = A \quad f^{(0)} = f$$
$$A^{(r+1)} = 2\cdot I - \left(A^{(r)}\right)^2 \qquad f_j^{(r+1)} = f_{j-2^{r-1}}^{(r)} + f_{j+2^{r-1}}^{(r)} - Af_j^{(r)}$$

und können das Gleichungssystem nun allgemein schreiben als

$$\mathcal{L}^{(r)} U^{(r)} = F^{(r)} \quad \text{mit}$$

$$\mathcal{L} = \begin{bmatrix} A^{(r)} & I & & \\ I & A^{(r)} & I & \\ & \ddots & \ddots & \ddots \\ & & I & A^{(r)} \end{bmatrix} \quad U^{(r)} = \begin{bmatrix} u_{1\cdot 2^r} \\ \vdots \\ u_{j\cdot 2^r} \\ \vdots \\ u_{2^{k+1}-2^r} \end{bmatrix} \tag{4}$$

und

$$\begin{bmatrix} A^{(r-1)} & 0 & \dots & \\ 0 & A^{(r-1)} & & \\ \vdots & \ddots & \ddots & \ddots \\ 0 & \dots & 0 & A^{(r-1)} \end{bmatrix} \begin{bmatrix} u_{2^r - 2^{r-1}} \\ \vdots \\ u_{j\cdot 2^r - 2^{r-1}} \\ \vdots \\ u_{M\cdot 2^2 - 2^{r-1}} \end{bmatrix} = \begin{bmatrix} f_{2^r-2^{r-1}}^{(r-1)} - u_{2^r} \\ \vdots \\ f_{j\cdot 2^r - 2^{r-1}}^{(r-1)} - (u_{j\cdot 2^r} + u_{(j-1)2^r}) \\ \vdots \\ f_{M\cdot 2^2 - 2^{r-1}}^{(r-1)} - u_{(M-1)2^r} \end{bmatrix}$$

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].
Dieses Verfahren teilt unser ursprüngliches Problem der Große $M \times N$ in M entsprechend kleinere Systeme, welche aufgrund der deutlich kleineren Dimensionsanzahl leichter zu lösen sind.

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

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

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

$$p_j^{(r+1)} = p_j^{(r)} - \left(A^{(r)}\right)^{-1}\left(p_{j-2^r}^{(r)} + p_{j+2^r}^{(r)} - q_j^{(r)}\right) \tag{5}$$

$$\begin{aligned} q_j^{(r+1)} &= q_{j-2^r}^{(r)} + q_{j+2^r}^{(r)} - 2p_j^{(r)} - 2\left(A^{(r)}\right)^{-1}\left(p_{j-2^r}^{(r)} + p_{j+2^r}^{(r)} - q_j^{(r)}\right) \\ &= q_{j-2^r}^{(r)} + q_{j+2^r}^{(r)} - 2p_j^{(r+1)} \end{aligned} \tag{6}$$

wobei

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

ist[Coh94, S. 14]. Dabei lässt sich die explizite Berechnung der $p_j^{(r)}$ einsparen, da sich diese aus (6) durch Umstellen ergibt:

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

Dadurch ist zwar ein etwa doppelt so hoher Rechenaufwand nötig, gleichzeitig halbiert sich jedoch der Speicheraufwand [Coh94, ??] Die Rekursionsvorschrift (6) lässt sich mit Hilfe von (5),(7) nun in eine Dreitermrekursion umformen:

$$\begin{aligned} q_j^{(r+1)} &= q_{j-2^r}^{(r)} + q_{j+2^r}^{(r)} - q_{j-2^{r-1}}^{(r-1)} - q_{j+2^{r-1}}^{(r-1)} + q_j^{(r)} \\ &\quad + \left(A^{(r)}\right) - 1\left(q_{j-3\cdot 2^{r-1}}^{(r-1)} + q_{j+2^{r-1}}^{(r-1)} + q_{j+3\cdot 2^{r-1}}^{(r-1)} + q_{j-2^{r-1}}^{(r-1)} - q_{j-2^r}^{(r)} - q_{j+2^r}^{(r)} - 2q_j^{(r)}\right) \end{aligned} \tag{8}$$

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 $q_0^{(r)} = q_N^{(r)} = 0 \quad \forall r$.
   Nun ist q_j^{r+1} nach (8) zu berechnen, wobei $r \in \{1, 2, \ldots, k-1\}$, $j \in \{k \cdot 2^r \, k \in \mathbb{N}, j < 2^{(k+1)} - 2^{(r+1)}\}$ ist.

2. Berechne nun die Lösung u_j für $j = k2^r \, k \leq N \in \mathbb{N}$ durch Umstellen von (ref41foo) und Einsetzen von f_j:

   $$u_j = \frac{1}{2}\left(q_{j-2^{r-1}}^{(r-1)} + q_{j+2^{r-1}}^{(r-1)} - q_j^{(r)}\right) - \left(A^{(r)}\right)^{-1}\left(u_{j-2^r} + u_{j+2^r} - q_j^{(r)}\right) \tag{9}$$

   wobei $u_0 = u_{2^{k+1}} = 0$ und $r \in k, k-1, \ldots, 0$, $j \in 2^{r-1}, 2^{r-3}, \cdots, 2^{k+1} - 2^{r-1}$.

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 auch sein Speicheraufwand sehr gering, da, wie bei näherer Betrachtung der Rekursionsformeln (8) und (9) deutlich wird, die Matrix F zunächst durch die $q_j^{(r)}$ und diese im zweiten Teil durch U überschrieben werden können. Es wird also lediglich 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 einem zusätzlichen Vektor als tridiagonalsystem lösen. Bei gleicher Unterteilung des Gitters in x- und y-Richtung ($M = N$) ergibt sich damit für den Algorithmus ein Aufwand von lediglich [Dor70, S. 14]

$$\frac{9}{2} N^2 \log_2 N.$$

Damit ist der Buneman-Algorithmus neben auf Fourier-Transformation beruhenden Algorithmen eine der effizientesten Lösungsmethoden der Poisson-Gleichung.
In der nebenstehenden Graphik ist nun die mit dem oben beschriebenen Algorithmus berechnete Verformung einer quadratischen Platte bei einer punktuellen Belastung (kleines Bild) zu sehen. Diese Lösung entspricht offensichtlich nicht der korrekten Verformung. Die Begründung hierzu liefert MATLAB in der Ausgabe

```
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 1.877730e-016.
```

Obwohl der Algorithmus numerisch stabil ist, wurde das Programm offensichtlich nahe an die Rechengenauigkeit geführt; ein Effekt, der verständlich wird, wenn man die Matrix $A^{(r)}$ versucht, zu berechnen: Die Elemente werden sehr groß und sie verliert ihre tridiagonale Gestalt. Dies führt entsprechend nicht nur zu numerischen Problemen, sondern auch zu einem sehr hohen Rechenaufwand.

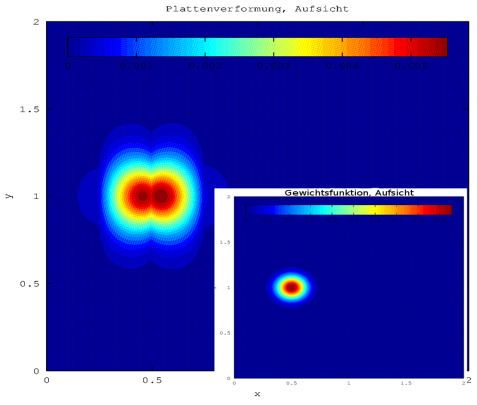


Abb. 1: Lösung der Poisson-Gleichung (großes Bild) bei einer Beispiel-Gewichtsverteilung (kleines Bild) für $N = M = 2^8 - 1$.

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

Wie die oben gezeigte Matrix zeigt, ist der bis hierher entwickelte Algorithmus noch nicht wirklich praxistauglich. Dabei ist die Rechengenauigkeit keineswegs das einzige Problem, denn auch die Blockmatrizen sind noch verhältnismäßig groß; in dem oben gezeigten Beispiel hat das Gleichungssysem immerhin 255 Dimensionen. Hinzu außerdem, dass die Tridiagonalstruktur verloren gegangen ist. Schon $A^{(1)}$ hat 5 Elemente pro Zeile, $A^{(2)}$ 7 und so weiter. Darum ist es sinnvoll, sich die Struktur nochmals genauer anzuschauen. 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:

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned} P_{2^r} &= -\prod_{i=1}^{2^r}(x-\mu_i) \\ A^{(r)} &= -\prod_{i=1}^{2^r}(A-\mu_i I) \end{aligned} \tag{10}$$

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.

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

Nun gilt nach [W G97, S. 10]:

$$v = -\sum_{i=1}^{2^r}\left(\prod_{\substack{j=1\\ j\neq i}}^{2^r}(\mu_j-\mu_i)^{-1}\right)(A-\mu_i I)^{-1}u \tag{11}$$

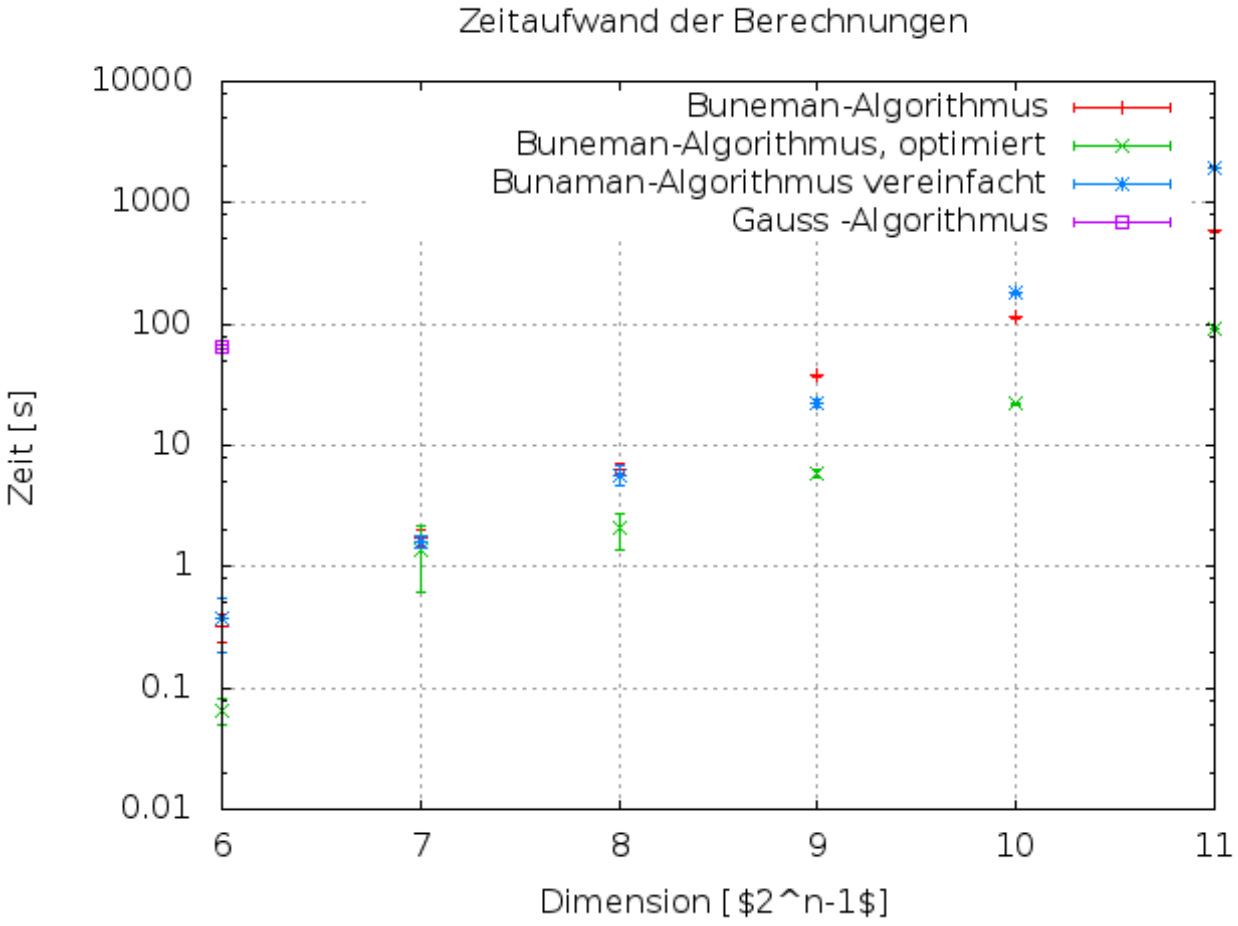
Im Gegensatz zur obigen Gleichung ist die hier auftretende Matrix $A - \mu_i I$ tridiagonal. Daher ist es hier sinnvoll, die Gleichung (11) umzustellen und in der Form

$$\begin{aligned} (A-\mu_i I)\,z_i &= \prod_{\substack{j=1\\ j\neq i}}^{2^r}\frac{v}{(\mu_i-\mu_j)} \quad i = 1,2,...,2^r \\ u &= \sum_{i=1}^{2^r} z_i \end{aligned} \tag{12}$$

zu lösen.
Dieses Problem ist nun die Lösung eines linearen Gleichungssystems mit tridiagonaler Struktur, welches mit Hilfe der LR-Faktorisierung (bzw. im Speziellen dem Thomas-Algorithmus QUELLE) nach dem SATZ 2.27 ????????? Vorlesung Numerische Mathematik I von Prof. Neymeyer, WS 2012/13[ua00, S. XY] mit linearem Aufwand lösbar ist.

6 Ergebnisse

Neben der Lösung des Problems sollen hier auch Analysen der unterschiedlichen im Text vorgestellten Algorithmen vorgestellt werden. Neben numerischen Problemen, die im Wesentlichen betrachtet wurden, spielt dabei auch der Rechenaufwand stets eine entscheidende Rolle, zumal die Lösung der Poisson-Gleichung oft nicht von primärem Interesse ist, sondern nur eine Teillösung eines Problems darstellt.



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.

[ua00] I. Bronstein u.a. *Taschenbuch der Mathematik*. 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 Z"urich, Stanford University, 1997. URL: http://www.inf.ethz.ch/personal/gander/papers/cyclic.pdf[Stand:24.09.2013].