

Universität Marburg

Sommersemester 2019

Fachbereich Mathematik u. Informatik

Prof. S. Dahlke, Prof. T. Surowiec, L. Sawatzki

PRAKTIKUM ZUR NUMERIK UND OPTIMIERUNG

- 1 Eine Innere-Punkt-Methode
- 2 Douglas-Rachford Forward-Backward Splitting
- 3 Parameter-Schätzung in der Optimierung bei Differentialgleichungen
- 4 Glättungs-Spline
- 5 Block-Iterationen
- 6 Bruttowertschöpfung mit QR-Zerlegung
- 7 Werner vs. Newton

Aufgabe 1 (Eine Innere-Punkt-Methode): Das Ziel dieses Praktikums ist, die sogenannte *primal affine scaling method* (PASM) zum Lösen linearer Optimierungsaufgaben zu implementieren. Im Folgenden seien $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$ mit $1 \leq m < n$ gegeben. Dazu sei stets angenommen, dass A vollen Rang m hat. Wir betrachten folgendes lineares Programm:

$$\min \{ \langle c, x \rangle \text{ über } x \in \mathbb{R}^n \mid Ax = b, \quad x \geq 0 \}. \quad (\text{LP})$$

Zur Motivation nehmen wir an, dass ein strikt primal-zulässigen Punkt x_0 vorhanden ist. Ist der Rand der zulässigen Menge vergleichsweise weiter weg vom x_0 , dann wäre es sinnvoll einen Schritt Δx möglichst weit in Richtung des steilsten Abstiegs der Zielfunktion zu machen. In diesem Fall ist diese Richtung durch $-c \in \mathbb{R}^n$ gegeben. Damit wir die Zulässigkeit beibehalten, müssen wir diesen Schritt auf den Nullraum der Matrix A projizieren. Das ergibt die Formel

$$\Delta x = -Pc$$

mit

$$P = I - A^T(AA^T)^{-1}A.$$

Die Richtung $-Pc$ heißt projizierte Richtung des steilsten Abstiegs (projected steepest descent direction).

Ist x_0 relativ nah am Rand der zulässigen Menge, dann würde dieser Schritt vermutlich keinen vernünftigen Abstieg in der Zielfunktion liefern. Motiviert durch diese Beobachtung transformieren wir (LP) in ein neues LP derart, dass der transformierte Punkt \bar{x}_0 nun einen strikten gleichmäßigen Abstand zum Rand der zulässigen Menge hat. Es ist

$$\bar{x}_0 := X_0^{-1}x_0 \text{ mit } X_0 := \text{diag}(x_0)$$

mit $\bar{x}_0 = e$ (der Vektor $(1, \dots, 1)^T \in \mathbb{R}^n$) und X_0 die Skalierungsmatrix. Damit betrachten wir in jedem Schritt das LP

$$\min\{\langle \bar{c}, \bar{x} \rangle \text{ über } \bar{x} \in \mathbb{R}^n \mid \bar{A}\bar{x} = b, \quad \bar{x} \geq 0\} \quad (\overline{\text{LP}})$$

mit $\bar{c} := X_0 c$ und $\bar{A} := AX_0$, und wir bestimmen die neue projizierte Richtung des steilsten Abstiegs im transformierten Problem:

$$\Delta \bar{x} = -\bar{P}\bar{c} = -(I - XA^T(AX^2A^T)^{-1}AX)Xc.$$

Der neue Iterierte \bar{x}_1 ist

$$\bar{x}_1 = e + \alpha \Delta \bar{x}$$

mit $\alpha > 0$ eine passende Schrittweite. Für das ursprünglich (LP) haben wir nun

$$x_1 = x_0 + \alpha \Delta x,$$

wobei

$$\Delta x = -X\bar{P}Xc = -(X^2 - X^2A^T(AX^2A^T)^{-1}AX^2)c$$

die primal affine-scaling direction ist. Die Schrittweite wird wie in der Vorlesung Lineare Optimierung mittels eines Quotientenkriterium bestimmt. Das Abbruchkriterium basiert idealerweise auf die Dualitätslücke. Allerdings ist eine Approximation dieser Lücke nicht immer berechtigt. Siehe die Aufgabenstellung für mehr Information.

Aufgabenstellung: In einer Programmiersprache Ihrer Wahl sollen Sie ein Programm schreiben, das ein Array $(A, b, c, x_0, \varepsilon, \text{maxit})$ mit $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$, $\text{maxit} \in \mathbb{N}$ nimmt, und mittels der PASM das entsprechend LP löst. Dabei muss gelten:

1. $1 \leq m < n$,
2. $\text{Rang } A = m$,
3. $x_0 > 0$ (strikt zulässig in allen Komponenten),
4. $\varepsilon > 0$ (kleine Stopp-Toleranz),
5. $\text{maxit} \in \mathbb{N}$ (maximale erlaubte Iterationszahl als Absicherung).

Ihr Code sollte die in der Vorlesung Lineare Optimierung hergeleitete Schrittweitenstrategie verwenden, um α (für das Ausgangsproblem (LP)) zu bestimmen. Zudem muss der Algorithmus sinnvolle Abbruchbedingungen enthalten:

1. Ist $\Delta x \geq 0$, STOPP: LP unbeschränkt.
2. Ist $\frac{c^T x_k - c^T x_{k+1}}{\max\{1, |c^T x_k|\}} \leq \varepsilon$, OUTPUT: aktuelle Dualitätslücke (wenn möglich), STOPP.

Ist das LP nicht degeneriert, dann darf man die Dualitätslücke durch

$$s_k = c - A^T y_k$$

mit

$$y_k := (AX_k^2 A^T)^{-1} AX_k^2 c$$

approximieren. Ihr Code sollte in keinem Fall die Matrix-Matrix-Produkte und explizite Inverse-Matrizen berechnen! Testbeispiele können Sie zuerst selber ausdenken. Weitere Testbeispiele bekommen Sie von Prof. Surowiec, sobald Ihr Code läuft.

Aufgabe 2 (Douglas-Rachford Forward-Backward Splitting): Das Ziel dieses Praktikums ist, das *Douglas-Rachford Forward-Backward Splitting* (DR) Verfahren für ein einfaches Optimierungsproblem in unendlich-dimensionalen Hilberträumen zu implementieren. Das DR-Verfahren ist ein sehr häufig verwendetes Verfahren mit vielen Anwendungen in den Naturwissenschaften inklusive maschinelles Lernen. Das Verfahren eignet sich für konvexe Optimierungsprobleme der Gestalt:

$$\min f(\Lambda x) + g(x) \text{ über } x \in \mathcal{H}.$$

Hier seien \mathcal{H}, \mathcal{V} reelle Hilberträume. Wir identifizieren \mathcal{H} mittels des Riesz'schen Darstellungssatzes mit dessen topologischen dualen Raum \mathcal{H}^* . Das innere Produkt auf \mathcal{H} bezeichnen wir mit $\langle \cdot, \cdot \rangle$. Dazu sei $f, g : \mathcal{H} \rightarrow \overline{\mathbb{R}}$ eigentliche konvexe unterhalbstetige Funktionale, und sei $\Lambda : \mathcal{H} \rightarrow \mathcal{H}$ ein beschränkter linearer Operator. Wir nehmen stets an, dass mindestens ein $x \in \mathcal{H}$ mit

$$0 \in \partial(f \circ \Lambda)(x) + \partial g(x)$$

existiert. Hier ist mit “ ∂ ” das übliche Subdifferential aus der konvexen Analysis gemeint, d.h.

$$\partial g(x) = \{x^* \in \mathcal{H} \mid g(y) \geq g(x) + \langle x^*, y - x \rangle \quad \forall y \in \mathcal{H}\}.$$

Für $\tau > 0$ zwei (maximal monotone) Operatoren $A, B : \mathcal{H} \rightrightarrows \mathcal{H}$ und $x_n, w_n \in \mathcal{H}$ ist das DR-Verfahren zum Lösen der verallgemeinerten Operatorgleichung

$$0 \stackrel{!}{\in} (A + B)(x)$$

durch die Vorschrift

$$\begin{aligned} w^{n+1} &= (I + \tau A)^{-1}(2x^n - w^n) + w^n - x^n \\ x^{n+1} &= (I + \tau B)^{-1}(w^{n+1}) \end{aligned}$$

gegeben. Für das obige Optimierungsproblem lässt sich diese Vorschrift mittels drei Iterierten $x^n, y^n, z^n \in \mathcal{H}$ schreiben:

$$\begin{aligned} y^{n+1} &= \arg \min_y f^*(y) - \langle \Lambda^* y, x^n \rangle + \frac{\tau}{2} \|\Lambda^* y + z^n\|_{\mathcal{H}}^2, \\ x^{n+1} &= (I + \tau \partial g)^{-1}(x^n - \tau \Lambda^* y^{n+1}), \\ z^{n+1} &= \frac{x^n - x^{n+1}}{\tau} - \Lambda^* y^{n+1}. \end{aligned}$$

Hier ist $f^*(y)$ mit $y \in \mathcal{H} \equiv \mathcal{H}^*$ das Fenchel-konjugierte Funktional von f aus der konvexen Analysis:

$$f^*(y) := \sup_{x \in \mathcal{H}} \{ \langle y, x \rangle - f(x) \}.$$

Das Konvergenzverhalten dieses Verfahrens finden Sie beispielsweise im 2010 erschienenen Buch von H. Bauschke und P.L. Combettes (Kapitel 27). Insbesondere ist die Wahl von $\tau > 0$, welches adaptiv gewählt werden darf, entscheidend für die rapide Konvergenz der Methode.

Aufgabenstellung: Sie sollen mithilfe des DR-Algorithmus ein einfaches Problem aus der Optimierung bei partiellen Differentialgleichungen (pDG) lösen. Sei $\Omega \subset \mathbb{R}^2$ mit hinreichend glatten Rand $\partial\Omega$ und betrachte

$$\inf \left\{ \frac{1}{2} \int_{\Omega} |(\Lambda u)(x) - y_d(x)|^2 dx + \frac{\alpha}{2} \int_{\Omega} |u(x)|^2 dx \text{ über } u \in L^2(\Omega) : \right. \\ \left. a(x) \leq u(x) \leq b(x) \quad \text{f.f.a. } x \in \Omega \right\}.$$

Hier ist $\alpha > 0$, und Λu ist die Einbettung in $L^2(\Omega)$ von der schwache Lösung y der elliptischen pDG

$$\begin{aligned} -\Delta y &= u \text{ in } \Omega, \\ y|_{\partial\Omega} &= 0 \text{ auf } \partial\Omega. \end{aligned}$$

Zudem seien $y_d, a, b \in L^2(\Omega)$ mit $a < b$ punktweise fast überall. Sie sollen folgende Aufgaben erledigen:

1. Bestimmen Sie \mathcal{H}, f, g .
2. Leiten Sie im Kontext dieses Problems eine vereinfachte Vorschrift für den DR-Algorithmus.
3. Mittels Ihrer Lieblings-Diskretisierungsmethode leiten Sie eine implementierbare Variante Ihres DR-Algorithmus her.
4. In einer Programmiersprache Ihrer Wahl sollen Sie ein Programm schreiben, das eine Array $(y_d, \alpha, a, b, \varepsilon, \text{maxit}, h)$ nimmt, und das Optimierungsproblem mit Ihrem DR-Algorithmus löst. Hier ist $h > 0$ für finite Elemente die Maschenweite.

Alle weitere Details bekommen Sie in Konsultationen mit Prof. Surowiec.

Aufgabe 3 (Parameter-Schätzung in der Optimierung bei Differentialgleichungen):

Es sei folgendes Modell für die Verschiebung y eines ein-dimensionalen elastischen Körpers von einem Restzustand gegeben:

$$-\frac{\partial}{\partial x}\left(a\frac{\partial}{\partial x}y(x)\right) + \beta(b(x))y(x) = f(x), \quad x \in (0, 1); \quad y(0) = y(1) = 0. \quad (1)$$

Hier ist $a > 0$ ein bekannter Materialparameter und $f \in C[0, 1]$ eine bekannte verteilte Kraft auf den Körper. Der Nemytski-Operator β ist durch die Funktion $\widehat{\beta} : \mathbb{R} \rightarrow \mathbb{R}$ mit

$$\widehat{\beta}(z) := \varepsilon + \begin{cases} z - \frac{\varepsilon}{2}, & z \geq \varepsilon, \\ \frac{z^3}{\varepsilon^3} - \frac{z^4}{2\varepsilon^4}, & z \in (0, \varepsilon), \\ 0, & \text{sonst} \end{cases}$$

mit $\varepsilon > 0$ gegeben. Die Funktion $b : (0, 1) \rightarrow \mathbb{R}$ ist unbekannt und sollte durch ein Inverse-Problem bestimmt werden. Wir gehen davon aus, dass Messungen y_1, \dots, y_m an den Stellen $x_1, \dots, x_m \in (0, 1)$ gegeben sind. Mit diesem Datensatz versuchen wir b anzunähern, indem wir folgendes Funktional betrachten:

$$J(y, b) := \sum_{i=1}^m |y(b)(x_i) - y_i|^2 + \frac{\nu}{2} \int_0^1 |b(x)|^2 dx, \quad \nu > 0,$$

wobei $y(b)$ eine Lösung von (1) für ein gegebenes $b : (0, 1) \rightarrow \mathbb{R}$ ist. Das ergibt folgendes Optimierungsproblem

$$\min \frac{1}{2} \sum_{i=1}^m |y(x_i) - y_i|^2 + \frac{\nu}{2} \int_0^1 |b(x)|^2 dx \text{ über } (b, y) \in B \times Y \quad (2a)$$

$$\text{U.d.N.} \quad (2b)$$

$$-\frac{\partial}{\partial x}\left(a\frac{\partial}{\partial x}y(x)\right) + \beta(b(x))y(x) = f(x), \quad x \in (0, 1); \quad y(0) = y(1) = 0, \quad (2c)$$

$$0 \leq b(x) \leq \bar{b}, \text{ f.f.a } x \in (0, 1), \quad (2d)$$

wobei $\bar{b} > 0$ eine fixe obere Schranke ist.

Die Aufgaben dieses Projekts sind wie folgt (einige Details finden Sie unten, alle weitere Details und Erklärungen bekommen Sie im persönlichen Gesprächen mit Prof. Surowiec):

1. Bestimmen Sie passende Räume B und Y damit (2) einen zulässigen Punkt hat.
2. Verwenden Sie den Satz über implizite Funktionen, um eine Sensitivität des Zustands y bezüglich des unbekanntes b herzuleiten. Dabei sollen Sie eine stetig (Fréchet) differenzierbare Funktion $y(b)$ bekommen, deren Ableitung eine verwandte lineare Differentialgleichung löst
3. Leiten Sie eine Formel für die Ableitung von $J(y(b), b)$ her.

4. Leiten Sie notwendige Bedingungen erster Ordnung in der Form einer Variationsungleichung her.
5. Lösen Sie (2) mit Hilfe des Verfahrens der Projiziertengradienten.

Details

1. Offenbar sollte b mindestens messbar und integrierbar sein. Die Verschiebung y löst eine Differentialgleichung und sollte deshalb mehr Regularität haben. Hier sollen Sie für die Differentialgleichung eine verwandte, b -abhängige Bilinearform finden, damit eine schwache Lösung für (1) definiert werden kann. Schauen Sie zum Beispiel in das Buch von Evans “Partial Differential Equations” nach.
2. Lesen Sie im Buch von E. Zeidler “Applied Functional Analysis” Sec. 4.8 für mehr zu diesem Thema nach. Wenn man (1) als Operator-Gleichung der Gestalt

$$e(y, b) := Ay + \beta(b)y - f = 0$$

betrachtet, dann kann man die Bedingungen des Satzes direkt überprüfen und anwenden.

3. Hier können Sie die Kettenregel anwenden. Denn nach 2. ist $y(b)$ stetig Fréchet-differenzierbar. Es hilft auch an dieser Stelle Auswertungsoperatoren $E_{x_i} : Y \rightarrow \mathbb{R}$ zu definieren, sodass

$$J(y(b), b) = \frac{1}{2} \sum_{i=1}^m |E_{x_i}(y(b)) - y_i|^2 + \frac{v}{2} \int_0^1 |b(x)|^2 dx$$

ist.

4. Nach 1.-3. können Sie klassische Ansätze aus der Optimierung verwenden, um diese Bedingungen herzuleiten. Für mehr über Variationsungleichungen können Sie im Buch “An Introduction to Variational Inequalities and Their Applications” von Kinderlehrer und Stampacchia lesen.
5. Das Verfahren der Projiziertengradienten ist eine der einfachsten numerischen Optimierungsalgorithmen. Für den Fall in diesem Projekt können Sie ohne weitere Schwierigkeiten die Projektion auf die bilaterale Schranken an b berechnen. Allerdings ist es notwendig, die Gradienten der Funktion $\mathcal{J}(b) := J(y(b), b)$ zu bestimmen. Letzteres erfolgt über die Lösung einer sogenannten adjungierten Gleichung.

Aufgabe 4 (Glättungs-Spline): Bei der Approximation von Meßwerten (x_i, y_i) , $i = 0, \dots, n$, mit $a = x_0 < x_1 < \dots < x_n = b$ durch Funktionen will man meist neben einer guten Approximation der Daten auch einen glatten Verlauf dieser Funktion haben. Beide Forderungen kann

man dadurch berücksichtigen, dass man mit einem festen Parameter $\omega \in (0, 1)$ das gewichtete Funktional

$$J(f) := \omega \sum_{j=0}^n \left(f(x_j) - y_j \right)^2 + (1 - \omega) \int_a^b \left[f^{(m)}(t) \right]^2 dt$$

in $C^m[a, b]$ minimiert. Im Fall $m = 2$ ist jede Minimalstelle von J tatsächlich ein natürlicher kubischer Spline mit den Knoten x_i , $i = 0, \dots, n$. Mit einer Basisdarstellung

$$s(x) = \sum_{j=0}^n a_j B_j(x),$$

des Minimalsplines durch B-Splines, erhält das Funktional J die Form

$$J(g) = \omega \|y - Wa\|^2 + (1 - \omega) a^T G a$$

mit den Vektoren $y = (y_0, \dots, y_n)^T$, $a = (a_0, \dots, a_n)^T$ und das Minimum ist Lösung des Gleichungssystems

$$\left(\omega W^T W + (1 - \omega) G \right) a = \omega W^T y.$$

Die Matrix hier ist symmetrisch positiv definit und besitzt Bandstruktur mit 7 Diagonalen, die Lösung kann daher mit einer Band-Cholesky-Zerlegung berechnet werden.

Testen Sie die Approximation mit den Tages-Maximaltemperaturen der HLUK-Wettermeßstation Marburg für die Monate Juni bis August letzten Jahres. Variieren Sie dabei das Gewicht $\omega \in (0, 1]$ und stellen Sie den Interpolationsfehler $\sum_{j=0}^n \left(f(x_j) - y_j \right)^2$ als Funktion von ω dar.

Literatur: Carl deBoor, A practical guide to splines, Kap. XIV

Aufgabe 5 (Block-Iterationen): Es soll die Lösung $X \in \mathbb{R}^{m \times n}$ eines lineare Gleichungssystems gefunden werden. Genauer sind $A \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{n \times n}$ und $C \in \mathbb{R}^{m \times n}$ gegeben, dann ist die sogenannte Sylvestergleichung

$$AX + XB = C \tag{3}$$

ein lineares Problem. Speichert man die Spalten der Matrizen X und C in großen Vektoren $x, c \in \mathbb{R}^{mn}$, ist die Sylvestergleichung tatsächlich äquivalent mit einem linearen System $(\tilde{A} + \tilde{B})x = c$ mit der Eigenschaft, dass die beiden Matrizen kommutieren, $\tilde{A}\tilde{B} = \tilde{B}\tilde{A}$. Diese Strukturen kann man sich bei Iterationsverfahren zu Nutze machen, etwa wenn A und B definit sind.

E. Block-Einzelschrittverfahren: Hier löst man der Reihe nach die j -te Spalte des Systems auf nach dem entsprechenden Spaltenvektor $x^{(j)} = X e^{(j)}$ und bekommt den Iterationsschritt

$$(A + b_{jj}I)x^{(j)} := c^{(j)} - \sum_{i \neq j} x^{(i)} b_{ij}, \quad j = 1, \dots, n,$$

bei dem nur ein kleineres $m \times m$ -System zu lösen ist.

- A.** ADI-Verfahren: Mit einem Parameter $r > 0$ formt man die Sylvestergleichung um in $(A + rI)X + X(B - rI) = C$ bzw. $X(B + rI) + (A - rI)X = C$ und nutzt diese als Iterationsvorschrift,

$$\begin{aligned}(A + rI)Y^{(k)} &= C - X^{(k-1)}(B - rI), \\ X^{(k)}(B + rI) &= C - (A - rI)Y^{(k)},\end{aligned} \quad k \in \mathbb{N},$$

in der jeweils auch nur Systeme der Größe m bzw. n auftreten.

Bei Diskretisierung der Poissongleichung $-\Delta u(t_1, t_2) = g(t_1, t_2)$ auf dem Einheitsquadrat durch Differenzen auf einem gleichförmigen Gitter mit $n + 1$ Knoten in jeder Richtung sind die Gitterwerte $x_{ij} \cong u(\frac{i}{n+1}, \frac{j}{n+1})$, $i, j = 1, \dots, n = m$ die Unbekannten in einem System (3), vgl. Literatur §8. Die Matrizen A, B sind gleich und sogar symmetrisch und tridiagonal. Daher ist der Gesamtaufwand in jedem Schritt der beschriebenen Verfahren nur $O(mn)$ und daher in der Größenordnung des einfachen Einzelschrittverfahrens, die Konvergenzgeschwindigkeit aber höher. Zur Implementierung benötigt man Unterprogramme, welche mit einer Choleskyzerlegung $LL^T = A + rI$ Systeme der Formen $LL^TY = R$ und $XMLL^T = S$ lösen. Vergleichen Sie beide Verfahren für $n = 32$.

Literatur: Stoer/Bulirsch, Numerische Mathematik 2.

Aufgabe 6 (Bruttowertschöpfung/QR): Bei Meßgrößen y , deren Wert von vielen Variablen abhängt, läßt sich der Einfluß der einzelnen Variablen $w_2, \dots, w_n \in \mathbb{R}$ durch ein affin lineares Modell

$$y(w_2, \dots, w_n) = x_1 + x_2 w_2 + \dots + x_n w_n$$

schätzen (d.h. $w_1 \equiv 1$), in dem die x_i die unbekanntenen Koeffizienten sind. Dazu zieht man $m \geq n$ Messungen $(y_k, w_{k2}, \dots, w_{kn})$, $k = 1, \dots, m$, aller Größen heran und minimiert den Defekt

$$\sum_{k=1}^m (y_k - x_1 - w_{k2}x_2 - \dots - w_{kn}x_n)^2.$$

Man löst also das Kleinste-Quadrate-Problem

$$\min\{\|Ax - b\|_2^2 : x \in \mathbb{R}^n\} \quad \text{mit} \quad A = \left(w_{ij}\right)_{i,j=1}^{m,n}, \quad w_{i1} \equiv 1.$$

Programmieren Sie dazu die Methode der QR-Zerlegung mit Householder-Spiegelungen und bestimmen Sie den Einfluß der verschiedenen Wirtschaftsbereiche der Bundesrepublik auf die Bruttowertschöpfung der Jahre 1991–2017. Die Daten dazu finden Sie unter <https://www.deutschlandin zahlen.de/tab/deutschland/volkswirtschaft/entstehung/bruttowertschoepfung-nach-wirtschaftsbereichen> oder fragen Sie Ihren Betreuer.

Möglicherweise haben einzelne Variable keinen merklichen Einfluß auf die Ergebnisvariable y = Bruttowertschöpfung, erkennbar an einem kleinen Betrag des zugehörigen Koeffizienten x_i . Identifizieren Sie solche Variablen anhand der ersten Lösung und wiederholen Sie die Rechnung mit entsprechend reduzierten Modellen (geht das ohne vollständige Neuberechnung der QR-Zerlegung?). Dadurch wird der Beitrag dieser Variablen im konstanten Term x_1 versteckt. Die Güte der verschiedenen Approximationen ist an den Residuen $\|Ax - b\|_2$ erkennbar.

- Literatur:**
1. Skript Num.1
 2. Golub/vanLoan: Matrix Computations, §12.6

Aufgabe 7 (Werner-Iteration): Das folgende Iterationsverfahren zur Lösung eines nicht-linearen Gleichungssystems $F(x) = 0$, mit der Vorschrift $x_0 = y_0 \in \mathbb{R}^n$, $k = 0, 1, \dots$,

$$\begin{aligned} F' \left(\frac{x_k + y_k}{2} \right) (x_{k+1} - x_k) + F(x_k) &= 0, \\ F' \left(\frac{x_k + y_k}{2} \right) (y_{k+1} - x_{k+1}) + F(x_{k+1}) &= 0, \end{aligned}$$

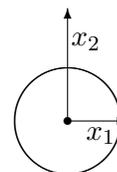
hat eine höhere Konvergenzordnung als das Newtonverfahren (bezogen auf den Aufwand). Man vergleiche die Konvergenzgeschwindigkeit dieses Verfahrens mit der des Newtonverfahrens

$$F'(x_k)(x_{k+1} - x_k) + F(x_k) = 0$$

bei dem unten beschriebenen Problem dadurch, dass als Startwert x_0 alle Punkte eines quadratischen Gitters verwendet werden. Am Startwert wird markiert, wieviele Schritte das jeweilige Verfahren benötigt, um die Nullstelle mit vorgegebener Genauigkeit (z.B. 10^{-7}) zu bestimmen. Dabei empfiehlt sich eine Graphik in Falschfarbendarstellung.

Als Anwendung soll der "optimale" Eimer berechnet werden, d.h. derjenige, der bei fester Mantelfläche π das größte Volumen hat. Die Radien x_1, x_2 dieses Eimers sind Nullstellen des Systems

$$\begin{aligned} f_1(x_1, x_2) &= (1 - 4x_1^2)(x_1 + 2x_2 - 2x_2^3) + 4x_1x_2^2(x_1^2 - 1) + 3x_1x_2^4 \\ f_2(x_1, x_2) &= 2x_1(1 - x_1^2)^2 + x_2(1 - 2x_1^2 + 4x_1^4) + 6x_1x_2^2(x_1^2 - x_2^2) - 3x_2^5. \end{aligned}$$



Man untersuche den Bereich $[0.1, 1] \times [0.1, 1]$.

Literatur: W.Werner, Numerische Mathematik 32 (1979), 333-342.