

Praktikum Numerische Mathematik: Das Jacobi-Verfahren

Lehrgebiet Mathematische Physik, FernUniversität Hagen

Eric Hambuch

`Eric.Hambuch@fernuni-hagen.de`

4. April 2003

1 Abstract

In diesem Praktikumsvortrag wird das Jacobi-Verfahren zur näherungsweisen Berechnung von Eigenwerten vorgestellt. Die Berechnung von Eigenwerten zu großen Matrizen wird z.B. in der Physik und Wahrscheinlichkeitsrechnung benötigt.

2 Einführung

Definition 2.1 (Eigenwerte) *Ein Eigenwert ist ein Skalar $\lambda \in \mathbb{C}$ mit*

$$Ax = \lambda x$$

mit einer Matrix $A \in \mathbb{C}^{n,n}$ und für $0 \neq x \in \mathbb{C}^n$. Die x werden als Eigenvektoren bezeichnet.

Satz 2.1 (Berechnung) *Die Eigenwerte $(x_1, \dots, x_n)$ einer Matrix A können als Nullstellen des charakteristischen Polynoms*

$$\chi_A(x) = \det(Ex - A)$$

berechnet werden.

Beweis: Wir setzen mit der Definition an und lösen $Ax = \lambda x \Leftrightarrow \lambda x - Ax = 0 \Leftrightarrow (\lambda E - A)x = 0$. Diese Gleichung hat nun genau dann für $x \neq 0$ Lösungen, wenn $\det(\lambda E - A) = 0$. qed.

Definition 2.2 (Spur) Die Spur $tr(A)$ einer Matrix A ist als Summe der Diagonalelemente gegeben durch

$$tr(A) = \sum_{i=0}^n a_{ii}.$$

Ein nützlicher Satz zur Abschätzung der Lage von Eigenwerten ist

Satz 2.2 (Gerschgorin-Kriterium) Zu einer Matrix $A = (a_{ij})$ bilde man in der komplexen Ebene die Kreise

$$K_i = \{z \in \mathbb{C} : |z - a_{ii}| \leq r_i; r_i = \sum_{j=1, j \neq i}^n |a_{ij}|\}$$

deren Zentren durch die Diagonalelemente und Radien durch die Betragssummen der restlichen Zeilenelemente gegeben sind. Jeder Eigenwert von A liegt nun in mindestens einem dieser Kreise.

Beweis: siehe [1, 6.1].

3 Das Jacobi-Verfahren

Sei A im folgenden eine reelle symmetrische Matrix. Damit sind alle Eigenwerte automatisch reell und A diagonalisierbar (vgl. [2]).

3.1 Grundidee

Grundlage des Verfahres bildet der folgende Satz

Satz 3.1 Jede symmetrische Matrix A läßt sich mittels einer orthogonalen Matrix Q ähnlich auf eine Diagonalmatrix D transformieren.

$$Q^{-1}AQ = D$$

.

Weiterhin besitzen A und D die gleichen Eigenwerte, denn ihre charakteristischen Polynome sind gleich: $\chi_D = \det(\lambda E - D) = \det(\lambda E - Q^{-1}AQ) = \det(Q^{-1}(\lambda E - A)Q) = \det(Q^{-1}) \det(\lambda E - A) \det C = \det(\lambda E - A) = \chi_A$.

Die Grundidee des Jacobi-Verfahrens besteht also darin, die Matrix durch (unendlich viele) orthogonale Ähnlichkeitstransformationen auf Diagonalgestalt zu bringen. Es handelt sich also um ein iteratives Näherungsverfahren.

Dabei soll jeder Iterationsschritt so gestaltet werden, daß die Elemente außerhalb der Diagonalen (die es ja schließlich zu eliminieren gilt) “im Mittel“ kleiner werden. Ziel ist also eine gegen die gewünschte Diagonalmatrix konvergierende Folge. Nach endlich vielen Iterationsschritten kann man die Eigenwerte (mit einer gewissen Genauigkeit) direkt ablesen oder z.B. mit dem *Gerschgorin*-Kriterium abschätzen.

Als Maß wird dazu die *Quadratsumme der nichtdiagonalen Elementen* herangezogen.

Definition 3.1 (Maß)

$$N(A) = \sum_{i,j=1, i \neq j}^n a_{ij}^2 = \operatorname{tr}(A^2) - \sum_{i=1}^n a_{ii}^2$$

Beweis: Da A symmetrisch, ist $\operatorname{tr}(A^2) = \sum_{i,j} a_{ij}a_{ji} = \sum_{i,j} a_{ij}^2$.

3.2 Rotation

Mit $N(A)$ erhält man zugleich ein Abbruchkriterium für das Jacobi-Verfahren:

Satz 3.2 Seien $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ die Eigenwerte der symmetrischen Matrix A , $d_1 \leq d_2 \leq \dots \leq d_n$ die Diagonalelemente von A . Dann gilt

$$|\lambda_i - d_i| \leq \sqrt{N(A)}.$$

Beweis: siehe [1, 6.23].

Wenn wir nun die Eigenwerte mit einer Genauigkeit von mindestens ϵ bestimmen wollen, so muß $N(A) < \epsilon^2$ werden.

Nun wird nur noch eine möglichst einfache orthogonale Matrix gesucht, die die Diagonaldominanz einer Matrix verstärkt. In $\mathbb{R}^2$ ist dies eine Rotationsmatrix

$$Q_2 = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}, \text{ mit } c = \cos \phi, s = \sin \phi.$$

Mit $A_2 = \begin{pmatrix} a_{pp} & a_{pq} \\ a_{pq} & a_{qq} \end{pmatrix}$ folgt daraus

$$\tilde{A}_2 = Q_2^T A_2 Q_2 = \begin{pmatrix} c^2 a_{pp} + 2cs a_{pq} + s^2 a_{qq} & cs(a_{qq} - a_{pp}) + (c^2 - s^2)a_{pq} \\ cs(a_{qq} - a_{pp} + (c^2 - s^2)a_{pq} & s^2 a_{pp} - 2cs a_{pq} + c^2 a_{qq} \end{pmatrix}.$$

Überträgt man dies auf $\mathbb{R}^n$, so rotiert man in der durch die Komponenten p und q gegebenen Koordinatenebene. Die entsprechende Matrix hat die folgende Form (dabei sei $p < q$):

$$Q = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & \cos \phi & & -\sin \phi & & \\ & & & 1 & & & \\ & & & & \ddots & & \\ & & \sin \phi & & & 1 & \\ & & & & \cos \phi & & \\ & & & & & 1 & \\ & & & & & & \ddots \\ & & & & & & & 1 \end{pmatrix} \begin{matrix} \leftarrow p \\ \\ \\ \leftarrow q \\ \\ \end{matrix}$$

Man beachte, daß bei der Transformation $\tilde{A} = Q^T A Q$ nur die p -te und q -te Zeile und Spalte verändert wird. Speziell ergeben sich

$$\begin{aligned} \tilde{a}_{ip} &= \tilde{a}_{pi} = \cos(\phi) a_{ip} + \sin(\phi) a_{iq}, i \neq p, q, \\ \tilde{a}_{iq} &= \tilde{a}_{qi} = -\sin(\phi) a_{ip} + \cos(\phi) a_{iq}, i \neq p, q, \\ \tilde{a}_{pp} &= \cos(\phi)^2 a_{pp} + 2 \cos(\phi) \sin(\phi) a_{pq} + \sin(\phi)^2 a_{qq}, \\ \tilde{a}_{qq} &= \sin(\phi)^2 a_{pp} - 2 \cos(\phi) \sin(\phi) a_{pq} + \cos(\phi)^2 a_{qq} \\ \tilde{a}_{ij} &= a_{ij} \text{ sonst.} \end{aligned}$$

Die Parameter p und q , sowie der Rotationswinkel ϕ soll nun so gewählt werden, daß $N(\tilde{A})$ um einen möglichst großen Betrag gegenüber $N(A)$ verkleinert wird.

Übrigens lassen sich parallel mit wenig Mehraufwand auch gleich die Eigenvektoren bestimmen. Dazu läßt man die Spaltvektoren aus der Matrix $X^{(v)} = Q^{(0)} Q^{(1)} \dots Q^{(v)}$ ab.

3.3 Wahl von ϕ

Um die Diagonaldominanz der Matrix $\tilde{A}$ zu stärken, sind die Elemente $\tilde{a}_{pq}$ und $\tilde{a}_{qp}$ außerhalb der Diagonalen möglichst zu eliminieren. Man setzt an

$$\tilde{a}_{pq} = \tilde{a}_{qp} = \cos \phi \sin \phi (a_{qq} - a_{pp}) + (\cos \phi^2 - \sin \phi^2) a_{pq} = 0.$$

Unter Zuhilfenahme der Additionstheoreme folgt daraus

$$\cot 2\phi = \frac{a_{pp} - a_{qq}}{2a_{pq}}.$$

Möchte man zudem die Auswertung von trigonometrischen Funktionen vermeiden, so kann man umformen und erhält für einen Drehwinkel $|\phi| \leq \pi/2$ (vgl. [1]):

$$C = \frac{a_{pp} - a_{qq}}{\sqrt{(a_{pp} - a_{qq})^2 + 4a_{pq}^2}}, c = \sqrt{(1 + C)/2}, s = \text{sign}(a_{pq})\sqrt{(1 - C)/2}.$$

3.4 Wahl der Rotationskoordinaten

3.4.1 Maximums-Variante

Wie ist der Rotationsindex p, q nun zu wählen, um $N(A)$ um einen möglichst großen Betrag zu verkleinern? Nach [1] gilt

Satz 3.3

$$N(\tilde{A}) = N(A) + 2\tilde{a}_{pq}^2 - 2a_{pq}^2.$$

Also sollte man das betragsgrößte Element a_{pq} wählen und durch die Rotation $\tilde{a}_{pq} = 0$ erreichen. Damit gilt nach der Rotation $N(\tilde{A}) = N(A) - 2a_{pq}^2$.

In Abschnitt 3.3 wurde schon die Wahl des Rotationswinkels ϕ beschrieben. Um nun das betragsgrößte Element a_{pq} zu ermitteln, sind rund $\frac{1}{2}(n^2 - n) \in O(n^2)$ Vergleiche durchzuführen (da A symmetrisch ist, reicht es aus, sich auf eine Hälfte zu beschränken). Im Verhältnis von nur $4n$ Multiplikationen zur Durchführung der Rotation ist dies jedoch zuviel, für jeden Iterationsschritt wären rund $O(n^2)$ Multiplikations- oder Vergleichsoperationen notwendig.

Eine Berechnung des Abbruchkriteriums $N(A)$ benötigt pro Iterationsschritt nochmals $O(n^2)$ Operationen.

Wir schätzen hier einmal grob die Konvergenz ab. Wenn a_{pq} ein betragsmaximales Element unter den $n(n - 1)$ Elementen außerhalb der Diagonalen ist, so gilt offensichtlich

$$N(A) \leq n(n - 1)a_{pq}^2.$$

Mit unseren bisherigen Überlegungen gilt für die transformierte Matrix

$$N(\tilde{A}) \leq kN(A) \text{ mit } k = 1 - \frac{2}{n(n - 1)}.$$

Nach v Schritten ergibt sich also

$$N(A^{(v)}) \leq k^v N(A),$$

und damit mindestens lineare Konvergenz. Es sind bei einer vorgegebenen Genauigkeit $N(A) < \epsilon^2$ mindestens $O(n^2)$ Rotationen durchzuführen [1]. Da jede Rotation rund $4n$ Multiplikationen benötigt, jede Suche nach dem maximalen Element rund $O(n^2)$ Vergleiche, so ergibt sich für das gesamte Verfahren ein Aufwand von $O(n^4)$ Operationen!

3.4.2 Zyklische Strategie

Bei der *zyklischen Strategie* wählt man die Rotationsindizes nun reihenweise

$$(1, 2), (1, 3), \dots, (1, n), (2, 3), \dots, (2, n), (3, 4), \dots, (n-1, n)$$

. Weiterhin verzichten wir natürlich auf eine Rotation, falls schon $a_{pq} = 0$ gilt. Einen Durchlauf nennt man *Zyklus* oder *Sweep*. Man kann zeigen, daß nach einigen Zyklen sogar eine quadratisch Konvergenz einsetzt (vgl. [3]).

Das Abbruchkriterium $N(A)$ berechnet man nur nach einem Zyklus, dies erspart nochmals unnötige Multiplikationen pro Rotation. Ein Zyklus ist also mit $O(n^3)$ Operationen durchführbar, auch wenn sicherlich $N(A)$ nicht so schnell konvergiert, wie bei der Maximumsvariante. Dafür spart man die aufwändige Suche nach dem betragsmaximalen Element. In der Praxis reichen schon wenige Zyklen, da in vielen Fällen das Verfahren (in Bezug die Zyklen) quadratisch konvergiert [1, 3].

3.4.3 Schwellenwertstrategie

Eine weitere Verbesserung bietet die *Schwellenwertstrategie* nach Rutishauser ([4]). Das Verfahren basiert auf der zyklischen Strategie aus dem vorherigen Abschnitt. Zusätzlich werden zwei neue Kriterien eingeführt:

1. *Schwellenwert*: In den ersten drei Zyklen werden nur Rotationen durchgeführt, wenn $|a_{pq}| > t$, wobei die Schranke auf $t = 0.2 \sum_{i,j,i \neq j} |a_{ij}| / (n^2)$ gesetzt wird. In späteren Zyklen wird $t = 0$ gesetzt. Der Schwellenwert wird nur jeweils zum Anfang eines Zykluses berechnet. Damit werden in den ersten Zyklen Rotationen eingespart, die zur Verkleinerung von $N(A)$ nur minimal beitragen.
2. Falls vor der Rotation a_{pq} klein im Verhältnis zu a_{pp} und a_{qq} ist, dann ist auch ϕ sehr klein und es wird einfach $a_{pq} = 0$ gesetzt und die Rotation eingespart. Man kann zeigen, daß dies keine größeren Fehler produziert, als wenn die Rotation durchgeführt worden wäre [4].

In [1] wird  brigens als Schwellkriterium $|a_{pq}|^2 > cN(A)/(n(n-1))$ f r eine Konstante $0 < c < 1$ genannt. Damit ist garantiert, da  in jedem Zyklus mindestens eine Rotation durchgef hrt wird. Rutishauser verzichtet in seiner Variante auf die zur Berechnung der Quadrate a_{ij}^2 notwendigen Multiplikationen und vernachl ssigt den Fehler im Nenner durch die N herung $n^2 \approx n(n-1)$.

3.4.4 Voevodin-Strategie

Hier wird nun eine weitere Strategie zur Bestimmung des Rotationselements vorgestellt. Dazu berechnet man f r $1 \leq i \leq n$:

$$d_i = a_{1i}^2 + \dots + a_{i-1,i}^2 + a_{i,i+1}^2 + \dots + a_{in}^2.$$

Dies sind die Quadrate der euklidischen L ngen der Zeilen ohne das Diagonalelement - die Matrix A ist ja reell symmetrisch. Die Rotationsstelle i, j wird nun gew hlt, indem erst das gr  te d_i gefunden wird, dann in der kompletten i -ten Zeile das betragsgr  te a_{ij} . Bei der Implementierung ist zu beachten, da  wir ggf. aus Geschwindigkeitsgr nden nur die obere H lfte der Matrix wirklich rotieren, wir aber eine komplette Zeile nach a_{ij} durchsuchen m ssen. Hier k nnen wir aber wieder die Symmetrie der Matrix ausnutzen und setzen

$$\bar{a}_{ij} = \begin{cases} a_{ij} & \text{f r } i < j \\ a_{ji} & \text{f r } i > j \end{cases}$$

Bei jeder Rotation  ndern sich nur zwei Elemente von $d_1, \dots, d_n$, also ben tigen wir zur Bestimmung der Rotationskoordinaten nur rund $2n \in O(n)$ Multiplikationen. Unser Abbruchkriterium $N(A)$ l  t sich ebenfalls schnell vor jeder Rotation bestimmen:

$$N(A) = \sum_{i=1}^n d_i.$$

Satz 3.4 *Seien $d_i = a_{1i}^2 + \dots + a_{i-1,i}^2 + a_{i,i+1}^2 + \dots + a_{in}^2$. Bei jeder Rotation in den Koordinaten p, q  ndern sich nur die Werte d_p und d_q , alle anderen d_i f r $i \neq p, q$ bleiben unver ndert.*

Beweis: Durch die Rotation werden nur die p -te und q -te Zeile und Spalte ver ndert. F r $i \neq p, q$ sind damit zu jedem d_i nur zwei Werte a_{ip} und a_{iq} ver ndert. Durch die Rotation gilt $\tilde{a}_{ip}^2 = (\cos \phi a_{ip} + \sin \phi a_{iq})^2$ und $\tilde{a}_{iq}^2 = (\cos \phi a_{iq} - \sin \phi a_{ip})^2$. Es folgt $\tilde{a}_{ip}^2 + \tilde{a}_{iq}^2 = \cos^2 \phi a_{ip}^2 + 2 \cos \phi \sin \phi a_{ip} a_{iq} + \sin^2 \phi a_{iq}^2 + \cos^2 \phi a_{iq}^2 - 2 \cos \phi \sin \phi a_{iq} a_{ip} + \sin^2 \phi a_{ip}^2 = a_{ip}^2 (\cos^2 \phi + \sin^2 \phi) + a_{iq}^2 (\cos^2 \phi + \sin^2 \phi) = a_{ip}^2 + a_{iq}^2$. Also  ndert sich durch die Rotation die Summe d_i nicht.

Die Idee dieser Strategie liegt also darin begründet, möglichst mit wenig Aufwand ein (betragsmäßig) sehr großes, wenn auch nicht unbedingt das größte Element zu bestimmen. Vergleicht man die Strategie mit Satz 2.2, so stellt man fest, daß hier versucht wird, den größten Gerschgorin-Kreis zu ermitteln und durch eine Rotation zu verkleinern.

3.4.5 Zusammenfassung

Bei Jacobi-Verfahren zeichnen sich also drei aufwandskritische Schritte ab:

1. die eigentliche Rotation, die immer $O(n)$ Operationen benötigt,
2. die Wahl der Rotationskoordinaten. Diese liegt in der Komplexität je nach Variante zwischen $O(1)$ bis $O(n^2)$,
3. die Berechnung des Abbruchkriteriums. Diese benötigt i.d.R. $O(n^2)$ Operationen pro Schritt.

Gerade bei der Berechnung des Abbruchkriteriums gibt es jedoch Optimierungspotential. So kann man z.B. $N(A)$ nicht nach jeder Iteration, sondern erst einem (Pseudo-)Zyklus berechnen und einige vielleicht unnötige Rotationen in Kauf nehmen. Zudem ist es nicht notwendig $N(A)$ jedesmal über die Summenformel (Def. 3.1) zu berechnen, sondern man kann $N(A)$ gemäß Satz 3.3 nachverfolgen. Um Rundungsfehler zu vermeiden, kann $N(A)$ dann z.B. nach jedem Zyklus einmal komplett berechnet werden.

4 Implementierung

4.1 Beschleunigung

Die Anzahl der Multiplikationen bei der Rotation läßt sich halbieren, wenn man nur jeweils die obere (oder untere) Hälfte der Matrix rotiert. Das Ergebnis der Rotation ist wieder symmetrisch, so daß dies keinen Einfluß auf die Korrektheit des Algorithmus hat.

4.2 Numerische Stabilität

Der Rotationswinkel läßt sich numerisch stabiler wie folgt berechnen [1, 3]. Man setzt

$$\kappa = \cot 2\phi = \frac{a_{pp} - a_{qq}}{2a_{pq}}$$

und

$$t = \tan \phi = \frac{1}{\kappa + \text{sign}(\kappa)\sqrt{1 + \kappa^2}}.$$

Dabei stellt t die betragskleinere Lösung der quadratischen Gleichung $t^2 + 2\kappa t - 1 = 0$ dar, womit der Drehwinkel wieder auf $|\phi| \leq \pi/2$ beschränkt wird. Für $\kappa = 0$ setzen wir $t = 1$. Es folgt

$$\cos \phi = \frac{1}{\sqrt{1 + t^2}}, \sin \phi = t \cos \phi.$$

5 Ergebnisse

Im Folgenden Abschnitt werden die Ergebnisse der vorgestellten Varianten verglichen.

5.1 Gesamtschritte

Als erster Test wurden die verschiedenen Varianten mit einigen Testmatrizen verglichen. Dabei wurde jeweils die *durchgeführten Anzahl der Rotationen*, ggf. aufgrund eines Schwellenwerts die *Anzahl der übersprungen Rotationen* und die *Gesamtdauer in ms*¹ für eine Genauigkeit $\epsilon = 0.0001$ gemessen.

$$A_1 = (a_{ij})_{1 \leq i, j \leq n} = \max(i, j).$$

Verfahren	Rotationen	übersprungen	Dauer in ms
Maximum	14709	-	12608
Zyklisch	60356	13895	761
Schwellenwert	17573	17078	310
Voevodin	14682	-	681

Tabelle 1: Matrix A_1 mit $n = 100$

$$A_2 = (a_{ij})_{1 \leq i, j \leq n} = i + j.$$

Bei der Matrix A_2 sind alle Eigenwerte bis auf zwei gleich 0.

$$A_3 = (a_{ij})_{1 \leq i, j \leq n} = \begin{cases} \text{random}([0, 100]) & \text{für } i \geq j \\ a_{ji} & \text{sonst} \end{cases}$$

¹Implementierung in Java auf einem Pentium III, 600 MHz Rechner

Verfahren	Rotationen	übersprungen	Dauer in ms
Maximum	441	-	451
Zyklisch	14851	-	281
Schwellenwert	569	19231	150
Voevodin	464	-	71

Tabelle 2: Matrix A_2 mit $n = 100$

Verfahren	Rotationen	übersprungen	Dauer in ms
Maximum	16669	-	14260
Zyklisch	34249	402	461
Schwellenwert	27143	7508	420
Voevodin	16622	-	701

Tabelle 3: Matrix A_3 mit $n = 100$

6 Fazit

Das *Jacobi-Verfahren* zur Berechnung von Eigenwerten läßt sich verhältnismäßig einfach programmieren. Das ursprüngliche Verfahren ist wegen einer Laufzeit von $O(n^4)$ jedoch nur für kleine Matrizen einsetzbar. Die verbesserten Varianten (insbesondere Schwellenwertverfahren) sind aber auch für große Matrizen geeignet. Es gibt jedoch noch weitere Näherungsverfahren, die nach entsprechenden Vorausberechnungen eine bessere Laufzeit bieten (z.B. QR-Verfahren).

Literatur

- [1] W. Oevel, *Einführung in die numerische Mathematik*, Spektrum Akademischer Verlag, 1996.
- [2] H. Anton, *Lineare Algebra*, Spektrum Akademischer Verlag, 1995.
- [3] H.R.Schwarz, *Numerische Mathematik*, B.G.Teubner Stuttgart, 1986.
- [4] H. Rutishauser, *The Jacobi Method for Real Symmetric Matrices*, Handb. of Autom. Comp, Vol II, 2002-211.