

Numerische Integration. Eigenwertaufgaben.

7. Vorlesung

170 004 Numerische Methoden I

Clemens Brand und Erika Hausenblas

Montanuniversität Leoben

2. April 2020

Numerische Integration. Eigenwertaufgaben.

① Numerische Integration

Klassisch: Newton-Cotes

Weitere Quadraturformeln

② Eigenwertprobleme

Definition: Eigenwerte und Eigenvektoren

Anwendungen: Netzwerke, Schwingungen. . .

Eigenwertaufgaben lösen: Methoden

Gliederung 7. Vorlesung

① Numerische Integration

Klassisch: Newton-Cotes

Weitere Quadraturformeln

② Eigenwertprobleme

Definition: Eigenwerte und Eigenvektoren

Anwendungen: Netzwerke, Schwingungen. . .

Eigenwertaufgaben lösen: Methoden

Numerische Integration

Gegeben:

eine Funktion $f(x)$ in einem Intervall $a \leq x \leq b$.

Gesucht:

deren Integral $\int_a^b f(x) dx$

Oft lässt sich das Integral nicht durch elementare Funktionen ausdrücken, oder die Funktion selbst ist nur tabellarisch gegeben. \rightarrow *Numerische Verfahren*

Integrationsformeln nach Newton-Cotes

Gegeben:

Eine Funktion $f(x)$ in einem Intervall (a, b) durch ihre Werte f_i an $n + 1$ äquidistanten Stützstellen,

$$f_i = f(a + ih), \quad \text{mit } h = \frac{b - a}{n}, \quad i = 0, \dots, n.$$

Prinzip:

Interpoliere $f(x)$ durch ein Polynom $p(x)$. Nähere das Integral von f durch das Integral von p . Die Näherung ist gegeben als gewichtete Summe der f_i ,

$$\int_a^b f(x) dx \approx (b - a) \sum_{i=0}^n \alpha_i f_i$$

mit fixen Gewichten α_i

Newton-Cotes-Formeln

Klassische Beispiele

mit Fehlertermen

Trapezregel

$$\int_a^b f(x) dx = \frac{b-a}{2} (f(a) + f(b)) - \frac{(b-a)^3}{12} f''(\xi)$$

Simpson-Regel

$$\int_a^b f(x) dx = \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right) - \frac{(b-a)^5}{2880} f^{IV}(\xi)$$

3/8-Regel („pulcherrima“)

$$\int_a^b f(x) dx = \frac{b-a}{8} (f(a) + 3f(a+h) + 3f(a+2h) + f(b)) - \frac{(b-a)^5}{6480} f^{IV}(\xi)$$

Newton-Cotes-Formeln

Klassische Beispiele

mit Fehlertermen

Trapezregel

$$\int_a^b f(x) dx = \frac{b-a}{2} (f(a) + f(b)) - \frac{(b-a)^3}{12} f''(\xi)$$

Simpson-Regel

$$\int_a^b f(x) dx = \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right) - \frac{(b-a)^5}{2880} f^{IV}(\xi)$$

3/8-Regel („pulcherrima“)

$$\int_a^b f(x) dx = \frac{b-a}{8} (f(a) + 3f(a+h) + 3f(a+2h) + f(b)) - \frac{(b-a)^5}{6480} f^{IV}(\xi)$$

Zusammengesetzte N.-C.-Formeln

Wenn feinere Intervallteilung vorliegt

Achtung bei Intervallbreite h ! Es sind $n + 1$ Datenpunkte, und um **eins weniger** Intervalle. $n =$ Anzahl **Datenpunkte-1**, und $h = (b - a)/n$

Zusammengesetzte Trapezregel

$$\int_a^b f(x) dx = \frac{h}{2} (f_0 + 2f_1 + 2f_2 + \cdots + 2f_{n-1} + f_n) + E$$

Zusammengesetzte Simpson-Regel

$$\int_a^b f(x) dx = \frac{h}{3} (f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 2f_{n-2} + 4f_{n-1} + f_n) + E$$

(Nur für gerades n möglich!)

Zusammengesetzte N.-C.-Formeln

Wenn feinere Intervallteilung vorliegt

Achtung bei Intervallbreite h ! Es sind $n + 1$ Datenpunkte, und um **eins weniger** Intervalle. $n =$ Anzahl **Datenpunkte-1**, und $h = (b - a)/n$

Zusammengesetzte Trapezregel

$$\int_a^b f(x) dx = \frac{h}{2}(f_0 + 2f_1 + 2f_2 + \cdots + 2f_{n-1} + f_n) + E$$

Zusammengesetzte Simpson-Regel

$$\int_a^b f(x) dx = \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 2f_{n-2} + 4f_{n-1} + f_n) + E$$

(Nur für gerades n möglich!)

Weitere Quadraturformeln

Gauß-Quadratur, Gauß-Lobatto-Formeln

Nicht äquidistante Stützstellen, dafür höhere Genauigkeit. (Typische Anwendung: Finite Elemente)

Romberg-Verfahren

berechnet mit zusammengesetzter Trapezregel mehrere Werte zu verschiedenen h und extrapoliert auf $h = 0$.

MATLAB

bietet zwei Verfahren: `quad` (adaptive Simpson-Regel) und `quadl` (trickreichere Gauß-Lobatto Methode)

Gliederung 7. Vorlesung

① Numerische Integration

Klassisch: Newton-Cotes

Weitere Quadraturformeln

② Eigenwertprobleme

Definition: Eigenwerte und Eigenvektoren

Anwendungen: Netzwerke, Schwingungen. . .

Eigenwertaufgaben lösen: Methoden

Eigenwertproblem

Definition: Eigenwerte und Eigenvektoren

Gegeben:

eine $n \times n$ -Matrix A

Gesucht:

- ▶ ein vom Nullvektor verschiedener Vektor \mathbf{x} und
- ▶ ein Skalar λ (auch $\lambda = 0$ ist erlaubt),

welche die Gleichung $A\mathbf{x} = \lambda\mathbf{x}$ erfüllen.

Ein solches λ heißt *Eigenwert* von A , ein passendes \mathbf{x} heißt *Eigenvektor* von A zum Eigenwert λ .

Anwendung:

Schwingungen, Hauptträgheitsachsen starrer Körper, Spannungstensor, Stabilitätstheorie, Hauptkomponentenanalyse, Quantenmechanik. . .

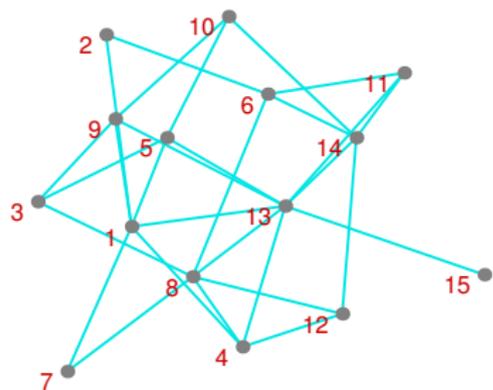
Eigenwerte und Eigenvektoren, anschaulich

Siehe Skriptum, Kapitel 8.1!

- ▶ „Der Hauptberuf einer Matrix ist, Vektoren zu multiplizieren!“
- ▶ $n \times n$ - Matrix beschreibt allgemein lineare Abbildung $\mathbb{R}^n \rightarrow \mathbb{R}^n$
- ▶ $\mathbf{y} = \mathbf{Ax}$: Matrix, angewandt auf Vektor \mathbf{x} , gibt neuen Vektor \mathbf{y} . Der zeigt normalerweise in andere Richtung und hat andere Länge.
- ▶ Jede $n \times n$ -Matrix hat aber ganz spezielle *Eigenvektoren*. Für sie ändert sich bei Multiplikation nur die *Länge*, aber die *Richtung* bleibt gleich (oder entgegengesetzt).
- ▶ sehen Sie sich die MATLAB-Demo EIGSHOW an (Übungsunterlagen)!

Beispiel: Erreichbarkeit in einem Netzwerk

Ein Netzwerk (Verkehrsverbindungen, verlinkte Seiten im Internet, soziales Netz... , mathematisch: ein Graph) lässt sich durch seine *Adjazenzmatrix* beschreiben



$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & \dots \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & \dots \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \dots \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & \dots \\ \vdots & \ddots \end{bmatrix}$$

Welcher Knoten ist am besten erreichbar?

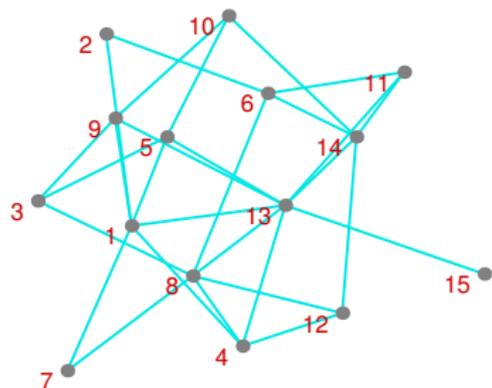
Welcher Knoten ist am besten vernetzt, am wichtigsten... ?

(Diese Matrix ist in Datei Netzwerk.m abgespeichert)

Beispiel: Erreichbarkeit in einem Netzwerk

Lässt sich als Eigenwertproblem $Ax = \lambda x$ formulieren

Ein Eigenvektor x zum betragsgrößtem Eigenwert λ liefert Bewertung!



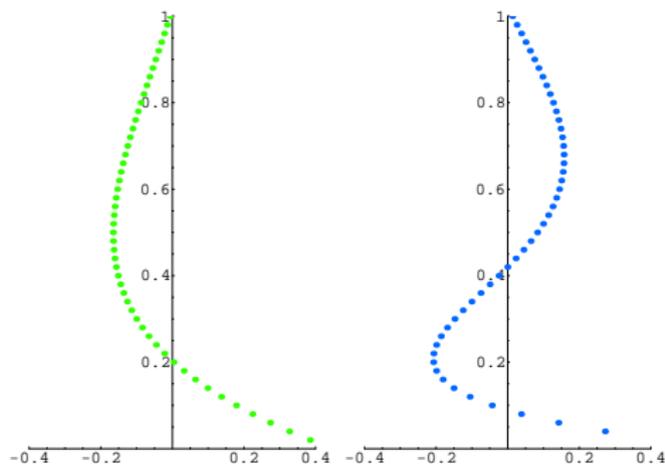
$$x = \begin{bmatrix} 74 \\ 27 \\ 41 \\ 64 \\ 56 \\ 45 \\ 32 \\ 72 \\ 56 \\ 38 \\ 46 \\ 44 \\ 100 \\ 60 \\ 22 \end{bmatrix}$$

(Die Datei pagerank.mlx zeigt ein ausführliches Beispiel)

Beispiel: Schwingungen der frei hängenden Kette

Das Eigenwertproblem $A \cdot \mathbf{x} = \frac{\ell \omega^2}{ng} \mathbf{x}$ beschreibt die Schwingungsformen einer (idealisierten) n -gliedrigen freihängenden Kette (Bild: erste und zweite Oberschwingung)

$$A = \begin{bmatrix} 1 & -1 & & \dots & & 0 \\ -1 & 3 & -2 & & & \vdots \\ & -2 & 5 & -3 & & \vdots \\ & & -3 & 7 & \ddots & \\ \vdots & & & \ddots & \ddots & \\ 0 & \dots & & & 1-n & 1-n \\ & & & & 1-n & 2n-1 \end{bmatrix}$$



Schwingungsberechnungen (Saiten, Membrane, Balken, Platten, Schall, ...) führen typischerweise zu Eigenwertaufgaben.

Beispiel: Gesichtserkennung mit *eigenfaces*

Hier sind einige zufällig ausgewählte
Gesichter aus einem Datensatz (Yale
Face Database)

The Yale Face Database contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink.



Beispiel: Gesichtserkennung mit *eigenfaces*

Das ist das gemittelte Gesicht und die ersten 15 „Eigengesichter“.
Sie enthalten bereits etwa 80% der gesamten Information der Daten.
Jedes einzelne Gesicht lässt sich aus dem „Mittelgesicht“ und den „Eigengesichtern“ F_1, F_2, F_3, \dots rekonstruieren.



Die Datei `eigenfaces.m` berechnet die eigenfaces

Beispiel: Gesichtserkennung mit *eigenfaces*

Hier ist ein Gesicht aus dem Original-Datenatz.

Die Zutaten-Liste lautet: „Mittelgesicht“ + ...

$$-1694.2F_1 - 2810.8F_2 + 382.4F_3 + 599.7F_4 \dots$$

$$-765.83F_5 + 358.55F_6 - 286.64F_7 \dots$$



Darunter die Gesichter rekonstruiert aus den ersten 15 und den ersten 30 Eigengesichtern.

Es ist viel billiger, statt aller Bild-Pixel nur 15 oder 30 Zahlenwerte pro Bild zu speichern.



Auch für ein unbekanntes Gesicht ist es einfacher, die 15 oder 30 Eigen-Zahlen zu berechnen und diese mit den Werten in der Datenbank zu vergleichen.



Methoden

- ▶ Nullstellen im charakteristische Polynom bestimmen (klassisch, ineffizient bei voller Matrix, gut für Tridiagonalmatrix!)
- ▶ Vektoriteration (langsam!)
- ▶ QR-Verfahren (Standard)
- ▶ Lanczos-Verfahren (Spezial-Verfahren bei schwach besetzter Matrix)

MATLAB-Befehle eig und eigs

Anwendungsbeispiele

$d = \text{eig}(A)$ liefert Vektor von Eigenwerten

$[V,D] = \text{eig}(A)$ Spalten von V sind Eigenvektoren, Diagonalelemente von D sind Eigenwerte.

$d = \text{eigs}(A)$, $d = \text{eigs}(A)$ analoge Befehle für **schwach besetzte** Matrizen. Liefern die sechs betragsgrößten Eigenwerte und zugeh. Eigenvektoren.

Wichtige Feststellungen zu $A\mathbf{x} = \lambda\mathbf{x}$

Siehe Skriptum, Kapitel 8.1 und 8.2!

- ▶ Eigenwerte sind Nullstellen des **charakteristischen Polynoms** $\det(A - \lambda I)$
- ▶ Eine $n \times n$ -Matrix hat genau n **reelle oder komplexe** Eigenwerte (bei entsprechender Zählung)
- ▶ Eigenwerte **symmetrischer** Matrizen sind immer **reell**.
- ▶ Ähnlichkeitstransformation: $X^{-1}AX$ und A haben die gleichen Eigenwerte .
- ▶ Hauptachsentransformation: Symmetrische Matrizen sind durch orthogonale Transformation diagonalisierbar, $Q^T \cdot A \cdot Q = D$.

Vektoriteration bestimmt den betragsgrößten Eigenwert

Iterationsverfahren nach Richard von Mises und Hilda Pollaczek-Geiringer

Gegeben eine $n \times n$ -Matrix A , ein Startvektor $\mathbf{x}^{(0)} \neq 0$ und ein fix gewähltes $i \in \{1, \dots, n\}$

Iteriere für $k = 1, 2, \dots$

berechne $\mathbf{y}^{(k)} = A\mathbf{x}^{(k-1)}$

setze $\lambda^{(k)} = y_i^{(k)}$ (i -te Komponente von $\mathbf{y}^{(k)}$)

setze $\mathbf{x}^{(k)} = \mathbf{y}^{(k)} / \lambda^{(k)}$ (Skalierung)

Falls ein reeller Eigenwert betragsmäßig größer ist als alle anderen Eigenwerte und ein zugehöriger Eigenvektor in Komponente i ungleich 0 ist, konvergieren die $\lambda^{(k)}$ und die $\mathbf{x}^{(k)}$.

Varianten dieses Verfahrens verwenden unterschiedliche Skalierungsvorschriften.

Das QR-Verfahren

bestimmt alle Eigenwerte einer Matrix A durch eine Folge orthogonaler Transformationen

Iteriere bis zur Konvergenz

berechne QR-Zerlegung $A = Q \cdot R$

setze $A = R \cdot Q$ (Ähnlichkeitstransform. $A \rightarrow Q^T \cdot A \cdot Q$)

Das Verfahren konvergiert (unter gew. Voraussetzungen und u.U. sehr langsam) zu einer Matrix in oberer Dreiecksform. Deren Hauptdiagonale enthält die Eigenwerte.

Für den praktischen Einsatz wird das Verfahren durch gezielte Verschiebungen der Art $\tilde{A} = A + sI$ beschleunigt.