Ernst-Abbe-Fachhochschule Jena
Hochschule für angewandte Wissenschaften

Beleg

Numerische Mathematik - Problem2: schlecht konditioniertes LGS

Marc Ludwig
Marvin Hohlfeld
Matthias Bukovsky
Matthias Springstein

Jena, 16. Mai 2013

Betreuer: Prof. P. Wilde
Gutachter: Dipl. Ing. Mustermann

Inhaltsverzeichnis

1. Entwickle die Funktion *f* aus obigem Problem in eine Taylorreihe 1

1.1. Die Taylorreihe 1

1.2. Definition 1

1.3. Eigenschaften 1

1.4. Konstruktion 2

1.5. Entwicklung von *f(x)* in eine Taylor Reihe 2

2. Nachweis der hermiteschen Form von *H* 4

2.1. Definition - hermitesch 4

2.1.1. direkte Folgen aus der Definition 4

2.2. Mathematischer Beweis für die gegebene Matrize 'H' 5

Glossar und Abkürzungsverzeichnis 6

Abbildungsverzeichnis 7

Quelltextverzeichnis 8

Literaturverzeichnis 9

A. Anhang 10

A.1. Quelltexte 10

Abstract

Hier kommt das Geschwafel hin, welches als Einführung in unsere Problemstellung dient.

1. Entwickle die Funktion *f* aus obigem Problem in eine Taylorreihe

1.1. Die Taylorreihe

In der Analysis verwendet man Taylorreihen (auch Taylor-Entwicklungen, nach dem Mathematiker Brook Taylor[1]), um Funktionen in der Umgebung bestimmter Punkte durch Potenzreihen darzustellen (Reihenentwicklung). So kann ein komplizierter analytischer Ausdruck durch eine nach wenigen Gliedern abgebrochene Taylorreihe (oftmals gut) angenähert werden, z.B. in der Physik oder bei der Ausgleichung geodätischer Netze: So ist die oft verwendete Kleinwinkelnäherung des Sinus eine nach dem ersten Glied abgebrochene Taylorreihe dieser Funktion.

1.2. Definition

Sei $I \subset \mathrm{R}$ ein offenes Intervall, $f\colon I \to \mathrm{R}$ eine unendlich oft differenzierbare Funktion und a ein Element von I. Dann heißt die unendliche Reihe

$$P_f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \cdots \tag{1.1}$$

$$= \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n \tag{1.2}$$

die Taylor-Reihe von f mit Entwicklungsstelle a.

Hierbei bezeichnet

- $n!$ die Fakultät $n! = 1 \cdot 2 \cdot \cdots \cdot n$ und
- $f^{(n)}(a)$ die n-te Ableitung von f an der Stelle a, wobei man $f^{(0)} := f$ setzt.

1.3. Eigenschaften

Die Taylorreihe $P_f(x)$ zur Funktion $f(x)$ ist eine Potenzreihe P in x. Im Fall einer analytischen Funktion f hat die Taylor-Reihe in jedem Punkt a einen positiven Konvergenzradius r und stimmt in ihrem Konvergenzbereich mit f überein, d.h. es gibt ein

[1] Brook Taylor (* 18. August 1685 in Edmonton, Middlesex; † 29. Dezember 1731 in Somerset House, London) war ein britischer Mathematiker. Nach ihm wurde u.a. die Taylorreihe benannt.

$r > 0$, sodass für $|x - a| < r$ gilt

$$P_f(x) = \sum_{n=0}^{\infty} a_n (x-a)^n = f(x) \tag{1.3}$$

. Außerdem stimmen die Ableitungen der Reihe im Entwicklungspunkt a mit den tatsächlichen Ableitungswerten überein:

$$P_f^{(k)}(a) = f^{(k)}(a) \tag{1.4}$$

.

1.4. Konstruktion

Damit die Ableitungen der Reihe im Entwicklungspunkt a mit den tatsächlichen Ableitungswerten übereinstimmen, sind die Koeffizienten a_n der Taylorreihe wie folgt konstruiert:

$$a_n = \frac{f^{(n)}(a)}{n!} \Rightarrow P_f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n \tag{1.5}$$

.

1.5. Entwicklung von *f(x)* in eine Taylor Reihe

Bei gegebener Funktion

$$f\left(x\right) = \frac{1}{1+x} \tag{1.6}$$

und den Gleichungen 1.1 und 1.2 folgt für deren Ableitungen

$$f'\left(x\right) = -\frac{1}{\left(1+x\right)^2} \tag{1.7}$$

$$f''\left(x\right) = \frac{2\left(1+x\right)}{\left(1+x\right)^4} = \frac{2}{\left(1+x\right)^3} \tag{1.8}$$

$$f'''\left(x\right) = \frac{2\left(1+x\right)^4 - 8\left(1+x\right)\left(1+x\right)^3}{\left(1+x\right)^8} = -\frac{6}{\left(1+x\right)^4} \tag{1.9}$$

Hieraus läßt sich folgende Bildungsvorschrifft für eine beliebige Ableitung bestimmen

$$f^n\left(x\right) = \left(-1\right)^n \frac{n!}{\left(1+x\right)^{n+1}} \tag{1.10}$$

hieraus folgt nach Gleichung 1.5

$$f\left(x\right) = P_f\left(x\right) = \sum_{n=0}^{\infty} \left(-1\right)^n \frac{n!}{n!\left(1+a\right)^{n+1}} \left(x-a\right)^n \tag{1.11}$$

$$= \sum_{n=0}^{\infty} \left(-1\right)^n \frac{\left(x-a\right)^n}{\left(1+a\right)^{n+1}} \tag{1.12}$$

2. Nachweis der hermiteschen Form von *H*

Eine hermitesche Matrix oder selbstadjungierte Matrix wird im mathematischen Teilgebiet der Linearen Algebra untersucht. Es handelt sich um eine spezielle Art von quadratischen Matrizen. Benannt sind diese nach dem Mathematiker Charles Hermite[1].

2.1. Definition - hermitesch

Eine $n \times n$-Matrix A mit Einträgen in C heißt hermitesch, wenn sie mit ihrer (hermitesch) Adjungierten A^*, also der transponierten und komplex konjugierten Matrix übereinstimmt. Das heißt, wenn

$$A = A^* = \overline{A}^{\mathrm{T}} = \overline{A^{\mathrm{T}}} \tag{2.1}$$

gilt.

Für die adjungierte Matrix finden sich auch die Bezeichnungen $A^\dagger$ und A^{H}. Die Schreibweise A^* wird auch für die komplex konjugierte Matrix verwendet.

Für die Einträge einer hermiteschen Matrix gilt also:

$$a_{jk} = \overline{a_{kj}} \tag{2.2}$$

Anders formuliert ist eine Matrix A genau dann hermitesch, wenn ihre Transponierte gleich ihrer komplex Konjugierten ist, d.h. $A^{\mathrm{T}} = \overline{A}$.

2.1.1. direkte Folgen aus der Definition

- Der Realteil einer hermiteschen Matrix ist symmetrisch, $\mathrm{Re}(a_{jk}) = \mathrm{Re}(a_{kj})$, der Imaginärteil ist schiefsymmetrisch, $\mathrm{Im}(a_{jk}) = -\mathrm{Im}(a_{kj})$.
- Ist A hermitesch, dann ist iA schiefhermitesch.
- Die Hauptdiagonalelemente einer hermiteschen Matrix sind reell.
- Für reelle Matrizen fallen die Begriffe hermitesch und symmetrisch zusammen.
- Hermitesche Matrizen sind normal, d. h $A^* \cdot A = A \cdot A^*$

[1] Charles Hermite (* 24. Dezember 1822 in Dieuze (Lothringen); † 14. Januar 1901 in Paris) war ein französischer Mathematiker.

2.2. Mathematischer Beweis für die gegebene Matrize 'H'

Mit der gegebenen Bildungsvorschrifft

$$H := \left(\frac{1}{j+k+1}\right)_{j,k=0}^{n} \tag{2.3}$$

Gleichung 2.2 und den Aussagen aus 2.1.1 folgt, dass für H der folgende Zusammenhang gillt:

$$H = H^{\mathrm{T}} \tag{2.4}$$

$$H_{j,k} = H_{k,j} \tag{2.5}$$

$$\left(\frac{1}{j+k+1}\right)_{j,k=0}^{n} = \left(\frac{1}{j+k+1}\right)_{k,j=0}^{n} \tag{2.6}$$

Glossar und Abkürzungsverzeichnis

Begriff Erklärung des Begriffs

Abbildungsverzeichnis

Quelltextverzeichnis

A.1. Foo 10
A.2. Berechnungen für eine nxn Matrize 10

Literaturverzeichnis

Karp, Richard M. (1972). *Reducibility Among Combinatorial Problems*. In: Miller, R. E. und J. W. Thatcher, Hrsg.: *Complexity of Computer Computations*, S. 85–103. Plenum Press.

Prechelt, Lutz (2000). *An Empirical Comparison of Seven Programming Languages*. Foobar book of lazyness, S. 23–29.

Seifart, Manfred (1990). *Analoge Schaltungen*. Huethig Buch Verlag Heidelberg, Heidelberg, Baden-Wuerttemberg, Germany.

A. Anhang

A.1. Quelltexte

```
#
# Öffnen der Datei und merken der File-Id, um später in Sie zu schreiben.
# ACHTUNG! Wird 'fopen' mit dem Parameter 'w' aufgerufen, so werden existente
# Dateien überschrieben.
#
fileIdData = fopen ('data', 'w');
fileIdTask3 = fopen ('aufgabe3', 'w');
#
# Iterationen um 3x3 bis 10x10 Matrizzen zu berechnen
#
for i = 3:10
  if (!numa (i, fileIdData, fileIdTask3))
    break;
  endif
endfor
# Schlie ß en der Dateien
fclose (fileIdData);
fclose (fileIdTask3);
```

Quelltext A.1: Foo

```
function retVal = numa (N, fileIdData, fileIdTask3)
  retVal = false;      # default Zuweisung

  if (isnumeric (N) && is_valid_file_id (fileIdData) && is_valid_file_id (
      fileIdTask3))
    clc;
    N--;
    #
    # Zähler und Nenner der gegebenen Funktion
    #
    numerator = [1];
    denominator = [1, 1];
    #
    # Erzeugen der Matrix
    #
    for i = 0:N
      for j = 0:N
        H(i+1, j+1) = 1/(1 + i + j);
      endfor
    endfor
    #
    # Anlegen des Lösungsvektors
    #
    R(1) = log(2);     # r0 ist gleich ln2
    for i = 1:N
      sum = 0;
      for j = 1:i
        sum = sum + ((-1)^j)/j;
      endfor
       R(i+1) = (-1)^i * (log(2) + sum);
    endfor
```

```
    #
    # Lösen des LGS
    #
    solved = H\R'
    #
    # Ausgabe
    #
    printf ("Matrix 'H' %ix%i:\n" ,N+1 ,N+1)
    disp (H)
    commentLine = strcat ("#\n# 'H' Matrix-", num2str (N+1), "x", num2str (N+1), "\n#");
    fdisp (fileIdData, commentLine);
    fdisp (fileIdData, H);
    printf ("zugehöriger Lösungsvektor 'r':\n")
    disp (R)
    commentLine = "#\n# 'R' Lösungsvektor\n#";
    fdisp (fileIdData, commentLine);
    fdisp (fileIdData, R);
    commentLine = "#\n# Lösung des LGS\n#";
    fdisp (fileIdData, commentLine);
    fdisp (fileIdData, solved');
    printf ("Eigenwerte:\n")
    lambda = eig (H);
    disp (lambda)
    commentLine = strcat ("#\n# Eigenwerte der Matrizze 'H' -", num2str (N+1), "x", num2str (N+1), "\n#");
    fdisp (fileIdTask3, commentLine);
    fdisp (fileIdTask3, lambda');   # lambda ist transponiert, um besser lesbar in eine Datei zu schreiben
    printf ("Kondition:\n")
    cond = cond (H);
    disp (cond)
    commentLine = "#\n# Kondition\n#";
    fdisp (fileIdTask3, commentLine);
    fdisp (fileIdTask3, cond);
    printf ("\n")
    #
    # Return bool 'true'
    #
    retVal = true;
  else
    printf ("Fehlerhafter Funktionsaufruf!\nSyntax: function (numericValue, fileId, fileId)\n")
  endif
endfunction
```

Quelltext A.2: Berechnungen für eine nxn Matrize