Stabilität, Fehleranalyse, Qualität

Feifan Chen
Betreuer: Julian Kunkel
Seminar: Modellierung und Simulation
Wintersemester 2012/2013
Universität Hamburg – Informatik – Wissenschaftliches Rechnen

Index

Einführung

- Einsatz der wissenschaftlichen Software in immer mehreren Bereichen für wichtigere Anwendungen
- Exaktheit und Zuverlässigkeit sind wesentlich

Einführung

- Einer der Schwierigkeiten bei der Entwicklung - Numerische Berechnung:
  - Funktionen: stetig -> diskretisiert
  - Prozesse: unbegrenzt -> begrenzt
  - Zahlen: reelle -> endliche Genauigkeit
    - ➢ Approximation
    - ➢ Vermeidung der Fehler unmöglich

Einführung

- Verwaltung der Fehler, um anzupassen und zu optimieren
- Berücksichtigung:
  - Eigenschaften der Fehler
  - Wie die Fehler fortpflanzen

Einführung

- Inhalt der Präsentation:
  - Was für Fehler kommen (häufig) vor?
  - Wie wird „Qualität“ einer Software/Methode definiert? Welche Bestandteile bilden den Begriff „Qualität“?
  - Wie wird die Qualität bemesst bzw. bewertet?

2. Probleme und Fehler bei der numerischen Berechnung

- Hauptproblem
- Grundfehler
  - Abrundung
  - Kürzbarkeit
  - Rekursion
  - Ganzzahlüberlauf
- Anomale Verhalten des Programms
- Softwarefehler
- Reales Beispiel

Hauptprobleme

- Der genaue Eingabewert ist nicht bekannt
- Manche Berechnungen können nicht genau durchgeführt werden

Grundfehler – Abrundung (Rounding)

- Nach jeder Operation wird das Ergebnis gerundet.
- Die Fehler pflanzen fort.

Grundfehler – Abrundung (Rounding)

- Beispiel: Ein MATLAB Code in der Norm IEEE 754 double:

```
function step(a,b,n)
% step from a to b with n steps
h=(b-a)/n;
x=a;
disp(x)
while x < b,
  x = x + h;
  disp(x)
end
```

- a = 1, b = 2, n = 3
  - h = 1/3 = 0,333……
  - Nach drei Schritten: x = 1 + 0,333*3 = 1,999…… < 2
  - Schleife geht weiter.
  - Vierter Schritt gebraucht → Fehler!

Grundfehler – Abrundung (Rounding)

- Beispiel: Ein MATLAB Code in der Norm IEEE 754 double:

```
function step(a,b,n)
% step from a to b with n steps
h=(b-a)/n;
x=a;
disp(x)
while x < b,
  x = x + h;
  disp(x)
end
```

- a = 1, b = 1,1, n = 3
  - h = 0,1/3, x und b beide sind Dezimalzahlen.
  - Vergleich von b und x wie gewünscht
  - Nach drei Schritten: x = b
  - → Richtig!

Grundfehler – Abrundung (Rounding)

- Empfehlung: bei Schleifen reelle Variable statt ganzzahlige Variable zu benutzen
- Lösung bei der Verwendung von reeller Variable:

```
function step(a,b,n)
% step from a to b with n steps
h=(b-a)/n;
x=a;
disp(x)
while x < b, while x < b-h/2,
  x = x + h;
  disp(x)
end
```

Grundfehler – Kürzbarkeit (Cancellation)

- Kürzbarkeit ist eine Eigenschaft von Elementen einer algebraischen Struktur.
- Definitioin:
  - Ein Element $c \in M$ heißt linkskürzbar, wenn für alle $a, b \in M$ gilt:

    $c*a = c*b \rightarrow a = b$,
  - Ein Element $c \in M$ heißt rechtskürzbar, wenn für alle $a, b \in M$ gilt:

    $a*c = b*c \rightarrow a = b$,
  - Ein Element $c \in M$ heißt kürzbar, wenn c links- und rechtskürzbar ist.
  - Eine Halbgruppe (S,*) heißt kürzbar, wenn jedes $c \in S$ kürzbar ist.
- Die Menge der reellen Zahlen mit üblichen Addition/Multiplikation/Subtraktion/Division ist kürzbar.

Grundfehler – Kürzbarkeit (Cancellation)

- Kommt während Subtraktion zwei fast gleichen Größen vor
- Sei x_1=1,243±0,0005 und x_2=1,234±0,0005.

  x_1-x_2=0,009±0,001.

  → Großer Unterschied

Grundfehler – Kürzbarkeit (Cancellation)

- Beispiel: Die Lösung für die allgemeine quadratische Gleichung $ax^2 + bx + c = 0$ $(a \neq 0)$ lauten:

$$x^{\alpha} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$x^{\beta} = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

- Mathematisch gleich, numerisch ungleich

Grundfehler – Kürzbarkeit (Cancellation)

- Sei $a = 1{,}0 \cdot 10^{-5}$, $b = 1{,}0 \cdot 10^{3}$, $c = 1{,}0 \cdot 10^{3}$,

$$x^{\alpha} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$x^{\beta} = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

- $x_1^{\alpha} = -3{,}0518$, $x_2^{\alpha} = -1{,}0000 \cdot 10^{8}$,
- $x_1^{\beta} = -1{,}0000$, $x_2^{\beta} = -3{,}2768 \cdot 10^{7}$.
- → Ganze unterschiedliche Ergebnisse!

Grundfehler – Kürzbarkeit (Cancellation)

- Sei $a = 1{,}0 \cdot 10^{-5}$, $b = 1{,}0 \cdot 10^{3}$, $c = 1{,}0 \cdot 10^{3}$,

$$x^{\alpha} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$x^{\beta} = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

- $x_1^{\alpha} = -3{,}0518$, $x_2^{\alpha} = -1{,}0000 \cdot 10^{8}$,
- $x_1^{\beta} = -1{,}0000$, $x_2^{\beta} = -3{,}2768 \cdot 10^{7}$.

- x_2^{α} und x_1^{β} sind korrekte Lösungen!

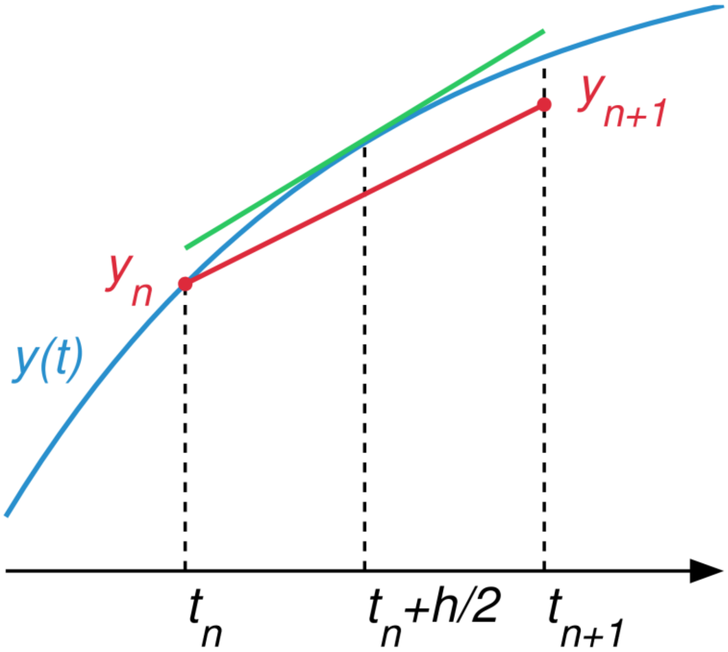
Grundfehler – Kürzbarkeit (Cancellation)

- Lösung:
  - Verwendung von double Genauigkeit verbessert nicht
  - Algorithmus wechseln
  - Das Zeichen $\pm$ von b entscheidet, welche Formel benutzt werden soll.

Grundfehler – Rekursion (Recursion)

- Beispiel: Midpoint Method

$$y_{n+1} = y_n + hf\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} f(t_n, y_n)\right)$$



- Quelle: http://en.wikipedia.org/wiki/Midpoint_method

Grundfehler – Ganzzahlüberlauf (Integer overflow)

- Beispiel: Fakultätsfunktion auf einem Computer mit der 32 bit integer Arithmetik:
- Bis 12! kann richtig ausgerechnet werden
- Bei 13! ist das Ergebnis falsch
- Bei 17! kommt eine negative Zahl als Ergebnis raus

- Lösung: Keine Berechnung von der Fakultätsfunktion

Anomale Verhalten des Programms

- Unangemessene Behandlung mit nicht abbrechenden Dezimalzahlen
- Inkonsistente Genauigkeit
- Falsche Abbruchkriterien
- Nicht offensichtliche falsche Code
- Instabile Algorithmen
- Überprüfung der Gleichheit

Softwarefehler

- Requirement bugs
- Implementation bugs
- Process bugs
- Build bugs
- Deployment bugs
- Future planning bugs
- Documentation bugs

Reales Beispiel

- Kommunalwahl in Schleswig-Holstein
  - Eine Partei 4,97% → 5%
  - Durfte deswegen in den Landtag

3. Qualität einer Methode

- Genauigkeit
- Exaktheit
- Zuverlässigkeit
- Stabilität
- Fehleranalyse
- Rückwärtsfehler
- Machine Precision
- Qualität einer Software

Genauigkeit (Precision)

- Die Genauigkeit bezieht sich auf die Anzahl der Dezimalstellen, die für das Rechnen, die Eingabe und die Ausgabe verwendet wird.

Exaktheit (Accuracy)

- Die Exaktheit bezieht sich auf die absoluten oder relativen Fehler von einer approximativen Größe.

- Absoluter Fehler:

$$|x - x^*|$$

- Relativer Fehler:

$$\frac{|x - x^*|}{|x^*|}$$

x^*: tatsächlicher Wert

x: ausgerechneter Wert

Exaktheit vs. Genauigkeit

- Die Exaktheit und die Genauigkeit müssen nicht übereinstimmend sein. Hohe Genauigkeit bedeutet nicht hohe Exaktheit.

Zuverlässigkeit (Reliability)

- Die Zuverlässigkeit misst, wie oft (in Prozent) die Software versagt, in dem Sinne, dass der herausgekommene echte Fehler größer als der erlaubte/geplante Fehler ist.

Stabilität (Stability)

- Die Stabilität befasst sich mit der Empfindlichkeit einer Methode bei (Abrundungs-) Fehlern im Lösungsprozess.
- Eine Methode, die die Exaktheit der Lösung genauso garantiert wie die Daten es gewährleistet, heißt stabil.

Stabilität (Stability)

- Beispiel einer instabilen Methode:

$$1{,}6x^2 + 100{,}1x + 1{,}251 = 0$$

Durch die Formel

$$x^\alpha = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

ergeben

$$x_1 = 62{,}53, x_2 = 0{,}03125.$$

Durch die Beziehung

$$x_1 x_2 = c/a$$

ergibt

$$x_2 = 0{,}01250.$$

Die richtigen Lösungen:

$$x_1 = 62{,}55, x_2 = 0{,}0125.$$

Fehleranalyse (error analysis)

- Die Fehleranalyse behandelt die Analyse der kumulativen Auswirkungen von Fehlern.
- Meistens Abbruch- und Abrundungsfehler.

- Beispiel:

$$p(x) = p_0 + p_1 x + p_2 x^2 + \cdots + p_n x^n$$

- Um zu wissen, ob das Polynom mit $x = \alpha$ richtigen Wert ausgibt, kann eine Fehleranalyse durchgeführt werden.

Fehleranalyse (error analysis)

- Vorwärtsfehleranalyse:
  - Wie nah ist die berechnete Lösung an die exakte Lösung?
- Rückwärtsfehleranalyse
  - Wie gut erfüllt die berechnete Lösung das zu lösende Problem?

Fehleranalyse (error analysis)

- Beispiel:

$$\text{Sei } A = \begin{pmatrix} 99 & 98 \\ 100 & 99 \end{pmatrix} \; und \; b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Die Lösung für die Gleichung $Ax = b$ ist $x = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

$$\text{Sei } \hat{x} = \begin{pmatrix} 2{,}97 \\ -2{,}99 \end{pmatrix}, \hat{x} - x = \begin{pmatrix} 1{,}97 \\ -1{,}99 \end{pmatrix}$$

Ergebnis der Vorwärtsfehler -analyse

$\hat{x}$ schlechte Lösung? Jedoch das Residuum

$$r = b - A\hat{x} = \begin{pmatrix} -0{,}01 \\ -0{,}01 \end{pmatrix}$$

Ergebnis der Rückwärtsfehler -analyse

Fehleranalyse (error analysis)

- Beispiel:

$$\text{Sei } A = \begin{pmatrix} 99 & 98 \\ 100 & 99 \end{pmatrix} \; und \; b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Die Lösung für die Gleichung $Ax = b$ ist $x = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

$$\text{Sei } \hat{x} = \begin{pmatrix} 1{,}01 \\ -0{,}99 \end{pmatrix}, \hat{x} - x = \begin{pmatrix} 0{,}01 \\ -0{,}01 \end{pmatrix}$$

Ergebnis der Vorwärtsfehler -analyse

$\hat{x}$ schlechte Lösung? Jedoch das Residuum

$$r = b - A\hat{x} = \begin{pmatrix} -1{,}97 \\ -1{,}97 \end{pmatrix}$$

Ergebnis der Rückwärtsfehler -analyse

Rückwärtsfehler (backward error)

Der Rückwärtsfehler an der Stelle $\tilde{x}$:

$$B(\tilde{x}) = \inf(\|\Delta z\|_z; \Delta z \in (\tau)\ und\ g(z + \Delta z) = \tilde{x})$$

Wobei ,

(τ) ist die Menge der akzeptierbaren Störung/Abweichung,

und $x = G(y) = g(z)$.

Machine Precision

- Notation von „machine precision“:

$$\Psi = 1^{+} - 1$$

Wobei 1^{+} ist Nachfolger von 1 in der Computer Arithmetik

Qualität einer Software

- Qualitätsquote eines zuverlässigen Algorithmus an der Stelle $\tilde{x}$ ist definiert durch:

$$J(\tilde{x}) = \frac{B(\tilde{x})}{\Psi}$$

Nach der Definition, $J(\tilde{x}) \geq 1$

$J = 1$ → beste Qualität

J signifikant größer als 1 → schlechte Qualität

Diagnose

- PRECISE: ein Set von Tools.
- Hilft Nutzer dabei, Experimente einzurichten, um die Exaktheit zu untersuchen.

Diagnose

- Modul 1: ein Modul für statistische Rückwärtsfehleranalyse, das die folgenden Bewertungen bereitstellt:
  - Rückwärtsfehler
  - Quote der Zuverlässigkeit und der Qualität
  - Messung des Sicherheitsintervalls
  - ……
- Modul 2: ein Modul für Analyse der Empfindlichkeit mit graphischer Darstellung.

Diagnose

- Seit 1988 eingesetzt für Entdeckung und Untersuchung der Probleme in industriellen Anwendungen.
- Prüfung der Algorithmen und Simulationssoftware in akademischen Bereichen.

Zusammenfassung

- Häufig vorkommende Fehler und Probleme bei der wissenschaftlichen Software, insbesondere bei der numerischen Berechnung
- Definition und Elemente von der Qualität einer Methode bzw. einer Software
- PRECISE als Diagnose-Tools

Quelle

- Einarsson, B. (2005): *Accuracy and Reliability in Scientific Computing*. SIAM, Philadelphia.
- http://en.wikipedia.org/wiki/Midpoint_method
- http://de.wikipedia.org/wiki/IEEE_754
- http://de.wikipedia.org/wiki/K%C3%BCrzbarkeit