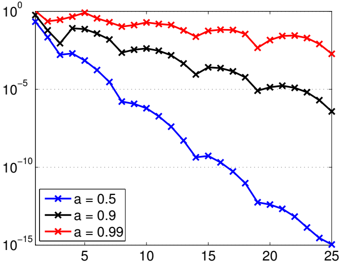
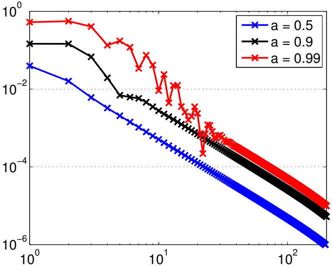
Numerische Methoden für D-MATH / D-PHYS, FS 2010





Vorlesung

Daten

Prof. Daniel Kressner
Seminar für angewandte Mathematik
D-MATH
Spezialisierung: Numerische Lineare Algebra
HG G 58.1, `kressner@math.ethz.ch`

- Vorlesungstermine:
  - Di 08.45–10.30 (HPH G 1)
  - Mi 13.15–15.00 (HG F 1)
- Erste Vorlesung: 24.02.2010.
  Letzte Vorlesung: 02.06.2010.

Kommunikation

- Stellen Sie Fragen in der Vorlesung! Wenn Sie etwas nicht verstehen, dann verstehen es Ihre Kollegen vermutlich auch nicht.
- Dozenten machen auch Fehler[1], weisen Sie auf Fehler hin!
- Nutzen Sie das Forum "Numerische Methoden" unter `http://metaphor.math.ethz.ch/forum/`.
- In dringlichen Fragen Dozenten oder Assistenten per E-mail kontaktieren.
- Homepage der Vorlesung: `http://www.math.ethz.ch/education/bachelor/lectures/fs2010/math/nm`
  Am besten zur D-MATH-Seite (`http://www.math.ethz.ch`) gehen und "Lecture Homepages" auswählen.

[1] Auch wenn ich dies nie zugeben würde.

Empfohlene Literatur

Tafelanschrieb, Skript und Übungen reichen zum Erlernen des Stoffs.

Ergänzende Literatur:

- Dahmen, Wolfgang; Reusken, Arnold. Numerik für Ingenieure und Naturwissenschaftler. Springer, 2005.
- Deuflhard, Peter; Hohmann, Andreas. Numerische Mathematik. I. Eine algorithmisch orientierte Einführung. Walter de Gruyter & Co., Berlin, 1993.
- Quarteroni, Alfio; Sacoo, Riccardo; Saleri, Fausto. Numerische Mathematik 1 und 2. Springer, 2002.

Übungen

Allgemeine Infos

- Übungstermin ist Freitag 08–10 bzw. 10–12.
- Gestern war auch Übung! (MATLAB-Übung – siehe nächste Folie.)
- Jede Woche wird eine neue Aufgabenserie verteilt und als PDF online gestellt. Ausnahme: Diese Woche *zwei* Serien (Serie 0 = MATLAB, Serie 1) Die Lösungen müssen in der darauffolgenden Woche in der jeweiligen Übung abgegeben werden.
- **Testatbedingung:** Jede Aufgabe wird einzeln bewertet. Wer am Semesterende 70% der theoretischen Aufgaben und 50% der Implementierungsaufgaben in MATLAB sinnvoll bearbeitet hat, erhält das Testat.
- Semesterpräsenz: nach Absprache mit Ihrem Assistenten. Nutzen Sie dieses Angebot!
- Nehmen Sie die Übungen und Serien ernst![2] Nur eine sorgfältige eigenständige Bearbeitung der Serien führt zum Prüfungserfolg.

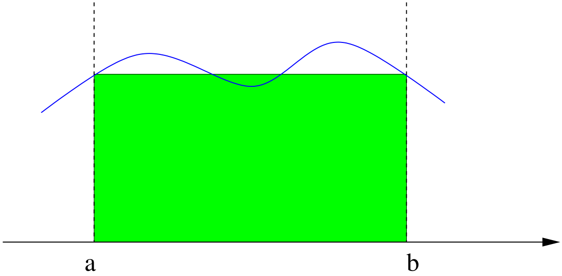
[2] Noch ernster als die Vorlesung.

MATLAB

- Der Umgang mit der Software MATLAB ist wichtiger Bestandteil der Vorlesung und Prüfung.
- Installationshinweise auf der Homepage der Vorlesung.
- Viele numerische Methoden werden klarer, wenn man sie einmal in MATLAB ausprobiert.
- Eine Open-Source Alternative zu MATLAB ist Octave.

Numerische Methoden: Genauigkeit

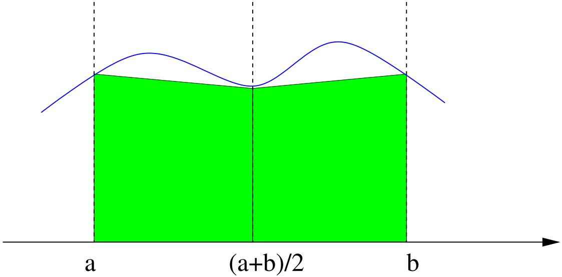
Trapezregel



Approximation:

$$\int_a^b f(x)\,\mathrm{d}x \approx \text{Fläche des Trapezes } = (b-a)\frac{f(a)+f(b)}{2}$$

Summierte Trapezregel



Approximation:

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &\approx \sum \text{Fläche der Trapeze}\\ &= h\frac{f(a)+f(a+h)}{2} + h\frac{f(a+h)+f(b)}{2}\\ &= h\Big(\frac{f(a)}{2} + f(a+h) + \frac{f(b)}{2}\Big)\end{aligned}$$

wobei $h = (b-a)/2$.

Summierte Trapezregel

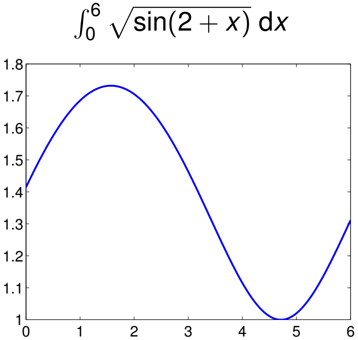
Allgemein: $N+1$ Funktionswerte an Stützstellen

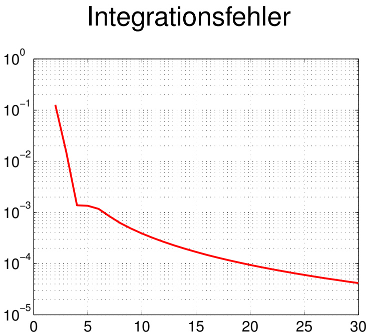
$$x_0 = a,\ x_1 = a+h,\ \ldots,\ x_{N-1} = a+(N-1)h,\ x_N = a+Nh = b,$$

mit $h = \frac{b-a}{N}$.

$$\int_a^b f(x)\,\mathrm{d}x \approx h\Big(\frac{f(x_0)}{2} + f(x_1) + \cdots + f(x_{N-1}) + \frac{f(x_N)}{2}\Big)$$

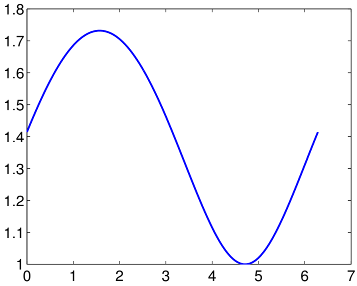
Beispiel 1


$\int_0^6 \sqrt{\sin(2+x)}\,\mathrm{d}x$
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0 1 2 3 4 5 6

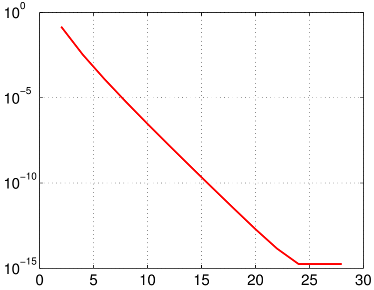

Integrationsfehler
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
0 5 10 15 20 25 30

Beispiel 2

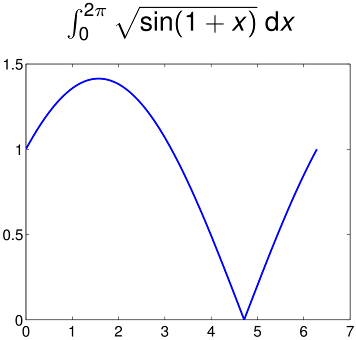
$$\int_0^{2\pi} \sqrt{\sin(2+x)}\,\mathrm{d}x$$

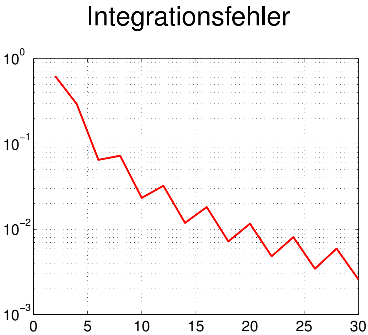


Integrationsfehler



Beispiel 3


$\int_0^{2\pi} \sqrt{\sin(1+x)}\, dx$
1.5
1
0.5
0
0
1
2
3
4
5
6
7


Integrationsfehler
10^{0}
10^{-1}
10^{-2}
10^{-3}
0
5
10
15
20
25
30

Berechnung von $1 + 2 + \cdots + 5\,000$

C code parallelisiert mit OpenMP für Mehrkernprozessoren:

```
#include <stdio.h>
#include <omp.h>
int main(void){
  float inp[5000];
  float sum = 0.0;
  int i,k = 5000;
  for (i = 0; i < k; i++) { inp[i] = (float)i; }
  omp_set_num_threads(4);
  #pragma omp parallel for
  for (i = 0; i < k; i++) {
      sum = sum + inp[i];
  }
  printf("Result: %f\n", sum);
  return 0;
}
```

Kompilieren mit `gcc -fopenmp ...`

Sind Rechner deterministisch?

Beispielausgabe bei mehrfacher Ausführung:

```
Result: 9928727.000000
Result: 12497500.000000
Result: 9501816.000000
Result: 9045588.000000
Result: 10535881.000000
Result: 12497500.000000
```

Ursache:

- Summe kann in einfacher Genauigkeit (`float`) nicht exakt dargestellt werden.
- Rundungsfehler hängen von Summationsreihenfolge ab.

Lösung:

- Doppelte Genauigkeit (`double`) behebt Rundungsfehler in *diesem* Beispiel.
- Indeterminismus der Rundungsfehler bei Parallelrechnern fundamentales Problem.

Sind Rechner deterministisch?

Beispielausgabe bei mehrfacher Ausführung:

```
Result: 9928727.000000
Result: 12497500.000000
Result: 9501816.000000
Result: 9045588.000000
Result: 10535881.000000
Result: 12497500.000000
```

Ursache:

- Summe kann in einfacher Genauigkeit (`float`) nicht exakt dargestellt werden.
- Rundungsfehler hängen von Summationsreihenfolge ab.

Lösung:

- Doppelte Genauigkeit (`double`) behebt Rundungsfehler in *diesem* Beispiel.
- Indeterminismus der Rundungsfehler bei Parallelrechnern fundamentales Problem.

Numerische Methoden: Effizienz

Fouriertransformation

$$x_{ij} \in \mathbb{R}, \qquad i, j = 0, \ldots, N-1,$$

Beispiel: Grauwerte eines $N \times N$-Bild.

Diskrete Fourier-Transformation von x_{ij}:

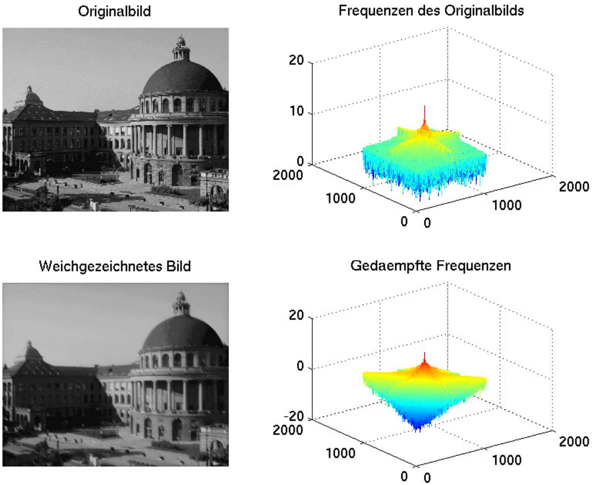
$$X_{k\ell} = \sum_{i=0}^{N-1} \omega_N^{ik} \sum_{j=0}^{N-1} \omega_N^{j\ell} x_{ij}, \qquad k, \ell = 0, \ldots, N-1,$$

mit $\omega_N = \exp(-2\pi\mathrm{i}/N)$.

Inverse diskrete Fourier-Transformation von $X_{k\ell}$:

$$x_{ij} = \frac{1}{N^2} \sum_{k=0}^{N-1} \overline{\omega_N}^{ik} \sum_{\ell=0}^{N-1} \overline{\omega_N}^{j\ell} X_{k\ell}, \qquad k, \ell = 0, \ldots, N-1,$$

Beispiel



- Berechnung mit schneller Fourier-Transformation benötigt 0.2 Sekunden.
- Berechnung nach Definition benötigt ca. 2 Sekunden.

Matrixmultiplikation

```
n = 700; A = rand(n); B = rand(n);
tic
C = zeros(700);
for j = 1:n,
   for i = 1:n,
      for k = 1:n,
         C(i,j) = C(i,j) + A(i,k)*B(k,j);
      end
   end
end
toc

tic
C = A*B;
toc
```

Beispielausgabe:

```
Elapsed time is 8.094434 seconds.
Elapsed time is 0.048986 seconds.
```

Matrixmultiplikation

```
n = 700; A = rand(n); B = rand(n);
tic
C = zeros(700);
for j = 1:n,
   for i = 1:n,
      for k = 1:n,
         C(i,j) = C(i,j) + A(i,k)*B(k,j);
      end
   end
end
toc

tic
C = A*B;
toc
```

Beispielausgabe:

```
Elapsed time is 8.094434 seconds.
Elapsed time is 0.048986 seconds.
```