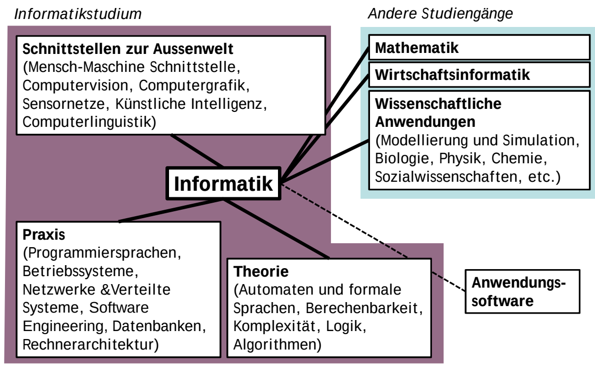
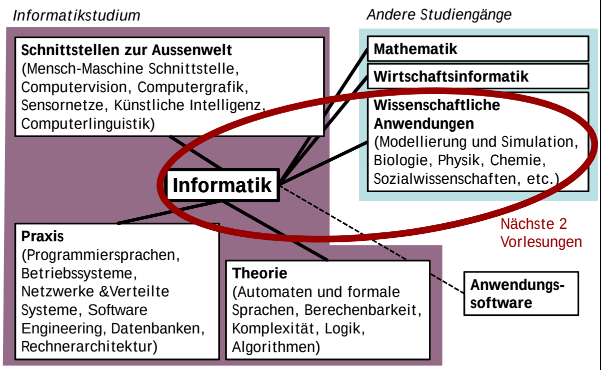
Einführung Informatik Modellierung und Simulation

Matthias Zwicker
Universität Bern
Herbst 2009

Übersicht



Übersicht



Übersicht

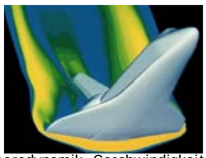
Modellierung und Simulation

- Einleitung
- Diskretisierung und Interpolation
- Kontinuierliche Modelle: Differential-gleichungen
- Diskrete Modelle: Warteschlangen

Modellierung & Simulation

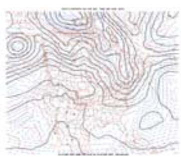
Modellierung & Simulation

- Wissenschaftliche & technische Anwendungen



Aerodynamik, Geschwindigkeits-verteilung der Luft
http://en.wikipedia.org/wiki/Computational_fluid_dynamics

Gehirn von Säugetieren (Blue Brain Project)
http://bluebrain.epfl.ch/



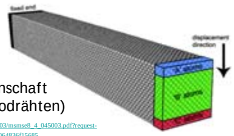
Numerische Wettervorhersage
http://de.wikipedia.org/wiki/Numerische_Wettervorhersage

Modellierung & Simulation

- Wissenschaftliche & technische Anwendungen

Baustatik

Metallbearbeitung
http://www.autoform.com/

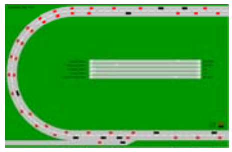


Materialwissenschaft
(Biegen von Nanodrähten)
http://www.iop.org/EJ/article/0965-0393/16/4/045003/msmse8_4_045003.pdf?request-id=2ebc01a2-2005-4bd7-a015-064836f15685

Modellierung & Simulation

- Soziale und politische Systeme
- Modellierung und Vohersage von Konflikten
  http://www.computerworld.com/s/article/print/338800/Bruce_Bueno_de_Mesquita?taxonomyId=13&taxonomyName=Government

Simulationsspiel „Die Sims“
http://www.diesims.de/



Verkehrssimulation
http://www.traffic-simulation.de/ger

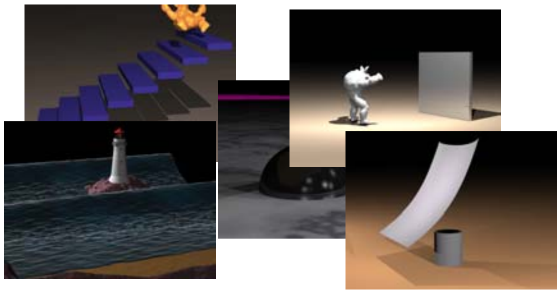
Modellierung & Simulation

- Filme, Computerspiele, virtuelle Welten
- Lichttransport, photorealistische Bildberechnung

http://en.wikipedia.org/wiki/Rendering_%28computer_graphics%29

Modellierung & Simulation

- Filme, Computerspiele, virtuelle Welten
- Brechen, biegen, brennen, etc.



http://physbam.stanford.edu/~fedkiw/

Modellierung & Simulation

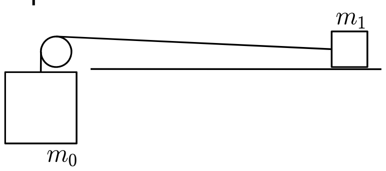
- Physikalische Welt
  - Beobachten mit Experimente und Messungen
- Modell
  - Abstrakte Beschreibung, die eine gewisse Klasse von Experimenten und Messungen wiedergeben kann
  - Darstellung mittels mathematischer Ausdrücke
- Simulation
  - Auswertung eines Modells für ein bestimmtes Experiment
  - Direkte Auswertung mathematischer Ausdrücke oder algorithmische Auswertung mittels Computer

Modellierung & Simulation

- Häufige Forderung: Verallgemeinerung
  - Modell soll anwendbar sein auf neue Experimente, die nicht zur Herleitung des Modells verwendet wurden
  - Im Idealfall: Vorhersagbarkeit, d.h., Simulation von neuen Experimenten stimmt mit neuen Messungen überein

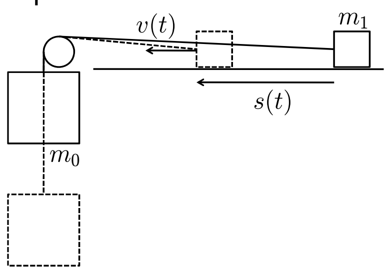
Beispiel

- Experiment



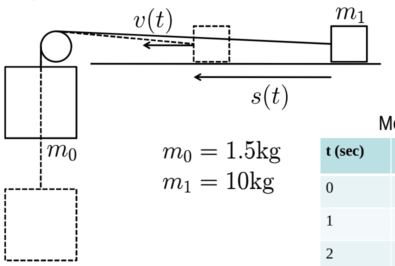
Beispiel

- Experiment



Beispiel

- Experiment

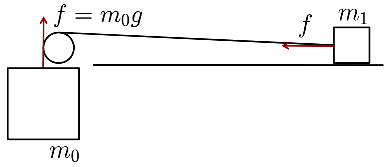


$m_0 = 1.5\text{kg}$
$m_1 = 10\text{kg}$

Messungen

t (sec)	v (m/sec)	s (m)
0	0	0
1	1.5	1.5
2	3	6
3	4.5	13.5
4	6	24

Modell



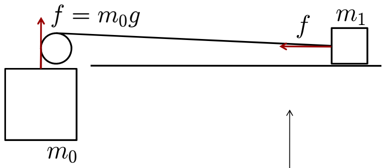
$a_1 = f/m_1 = \frac{m_0}{m_1} g$

$v_1(t) = a_1 t$

$s_1(t) = v_1 t = a_1 t^2$

Stimmt mit den Messungen überein!

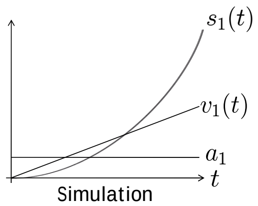
Simulation



$a_1 = f/m_1 = \frac{m_0}{m_1} g$

$v_1(t) = a_1 t$

$s_1(t) = v_1 t = a_1 t^2$



Simulation
(Verallgemeinerung: für beliebige Parameter m_0, m_1 gültig)

Modellierung & Simulation

- Nutzen
  - Verständnis und Erklärung von Beobachtungen
  - Vorhersage von Beobachtungen
  - Unterstützung bei Konstruktion und Manipulation von Systemen
- Probleme
  - Qualität des Modells oft schwierig zu beurteilen (Validierung)
  - Vor allem wenn Messungen von Experimenten aufwändig/unmöglich (Sozial-, Wirtschaftswissenschaften)

Modellierung & Simulation

- Rolle der Informatik und Computer-technologie
  - Auswertung und Simulation von vielen Modellen unmöglich ohne Computer
- Modellierung & Simulation ist interdisziplinär
  - Computational Science
    http://en.wikipedia.org/wiki/Computational_science

Klassifikation von Modellen

- Deterministisch
  - Mehrfache Ausführung des serselben Simulation ergibt immer das selbe Resultat
- Stochastisch
  - Mehrfache Ausführung derselben Simulation ergibt unterschiedliche Resultate
- Statisch
  - Zeit spielt keine Rolle
- Dynamisch
  - Zeitabhängig

Klassifikation von Modellen

- Diskret
  - Modell enthält Objekte aus einer abzählbaren Menge
  - Häufig Menge von Ereignissen
- Stetig
  - Modell enthält kontinuierliche Grössen
  - Häufig Differentialgleichungen, stetige Funktionen

Übersicht

Modellierung und Simulation

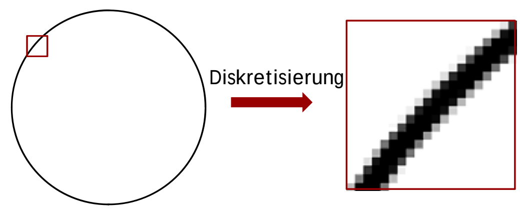
- Einleitung
- Diskretisierung und Interpolation
- Kontinuierliche Modelle: Differential-gleichungen
- Diskrete Modelle: Warteschlangen

Diskretisierung

- Digitale Computer können nur eine endliche Menge von Zahlen darstellen
  - Nur Teilmenge der reellen Zahlen
  - Nicht alle können exakt dargestellt werden
- Digitale Computer können nur eine endliche Menge von Zahlen speichern
  - Kontinuierliche Objekte müssen durch endliche Anzahl von Werten dargestellt werden
- Kontinuierliche Modelle müssen diskretisiert werden

Kontinuierliche Objekte

- Geometrische Objekte, z.B., Kreis



$\{(x,y)|x^2+y^2=100\}$

Unendliche Menge von Punkten

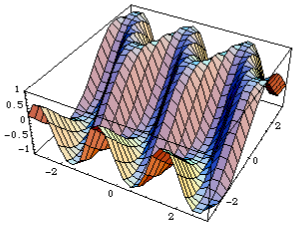
$\{(0,10), (1,9.8),...\}$

Endliche Menge von Pixels (die wenigsten exakt auf Kreis)

Kontinuierliche Objekte

- Mathematische Funktionen

In[5]:= Plot3D[Sin[y + Sin[3 x]], {x, -3, 3}, {y, -3, 3}]

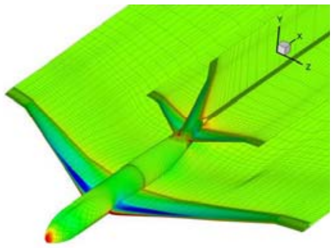


Out[5]= - SurfaceGraphics -

3D Visualisierung einer Funktionstabelle

Kontinuierliche Objekte

- Physikalische Grössen



Druckverteilung auf Gitter (Tabelle)
http://www.structures.ethz.ch/research/detail?id=15366

Kontinuierliche Objekte

- Multimedia Daten
- Bilder, Töne, Sprache
- Diskretisierung schon im Aufnahmegerät

Digitale Bilder bestehen aus 2D Tabellen von Pixeln

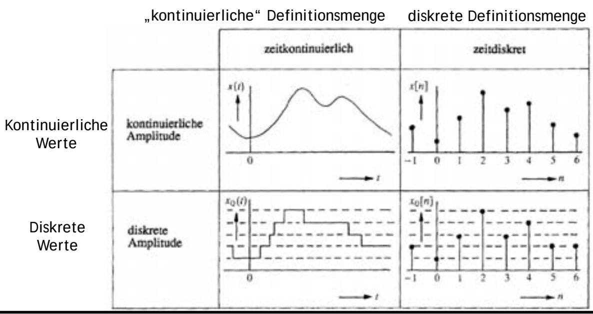
Diskretisierung

Zwei Komponenten

- Quantisierung einzelner Werte
  - Digitale Approximation reeller Zahlen mit Gleitkomma Darstellung
- Abtastung kontinuierlicher Objekte
  - Tabellierung einer endlichen Anzahl von Werten

Diskretisierung

- Vier Signalklassen (Signal synonym zu Funktion)



Beispiele

- Quantisierung von reellen Zahlen mit IEEE Gleitkommazahlen
- Rasterkonvertierung von Strecken

Beispiel: Reelle Zahlen

- In jedem Intervall zwischen zwei unterschiedlichen reellen Zahlen gibt es unendlich viele (sogar überabzählbar viele) weitere reellen Zahlen
- Wertebereiche digitaler Repräsentationen sind jedoch endlich
  - Z.B. Java Datentypen float, double

Digitale Repräsentation

- Basiert auf Gleitpunktdarstellung (Floating Point Representation)
- Normalisierung: Jede dezimale Zahl kann so dargestellt werden, dass
  - Nur eine Ziffer vor Dezimalpunkt
  - Ziffer vor Dezimalpunkt ist nicht Null

$80000 = 8.0*10^{4}$

$0.0023 = 2.3*10^{-3}$

$5.044 = 5.044*10^{0}$

Normalisierung

- Binärzahlen
  - Hier: Exponent in Dezimaldarstellung
  - Schiebe Ziffern eins nach links =>
    Multiplikation mit 2 =>
    verkleinere Exponent um 1 (zur „Kompensation“)
  - Schiebe Ziffern eins nach rechts =>
    Division mit 2 =>
    erhöhe Exponent um 1

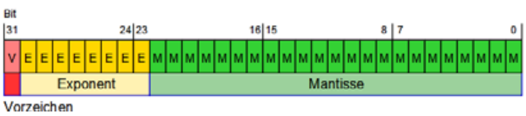
$0.0011001 = 1.1001*2^{-3}$

$100101*2^{9} = 1.00101*2^{14}$

IEEE Gleitpunktdarstellung

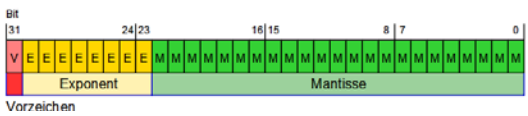
- Standardisierte Formate zur Darstellung von binären Gleitpunktzahlen
  http://de.wikipedia.org/wiki/IEEE_754
  - IEEE: Institute of Electrical and Electronics Engineers
  - Standard IEEE 754 verabschiedet 1985
- Verschiedene Formate, hauptsächlich
  - 32 Bits single precision Format, Java float
  - 64 Bits double precision Format, Java double
- Implementiert in allen modernen Prozessoren

IEEE single precision float



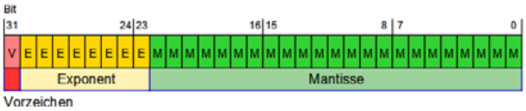
- 1 Bit Vorzeichen V
- 8 Bits Exponent E
- 23 Bits Mantisse M

IEEE single precision float



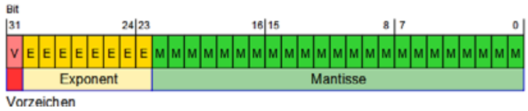
- Gegeben: Normalisierte binäre Gleitkommazahl
  $x = s \cdot m \cdot 2^{e}$
  - Vorzeichen s, -1 oder 1
  - Mantisse (normalisiert) $m=1.X_0X_1...$
  - Exponent e

IEEE single precision float



- Kodierung in IEEE
  $x = s \cdot m \cdot 2^e$
  - Vorzeichenbit: wenn s=-1, V=1, sonst V=0
  - Exponent mit Biaswert: $E = e+127$
  - Bias erlaubt negative e ohne Vorzeichenbit
  - Mantisse: sei m=1. X_0X_1... X_{23}, dann M=X_0X_1... X_{23}
  - Ein Bit gespart, möglich wegen Normalisierung!

IEEE single precision float



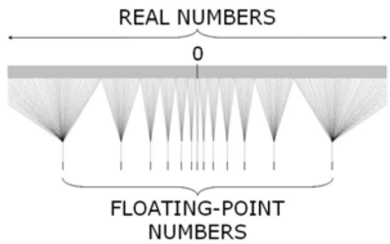
- Gegeben 32 Bit IEEE Zahl mit V, E, M
- Vorzeichen $s = (-1)^V$
- Exponent $e = E$ -127
  - Bias wieder abziehen!
- Mantisse: $m = 1.\,M = 1 + M / 2^{23}$
  - Ziffer 1 vor dem Gleitpunkt ist implizit!
- Ergibt Binärzahl $x = s \cdot m \cdot 2^e$

Spezialfälle

- Darstellung von 0: E=M=0
  - Verschiedene Darstellung von +0, -0
- Denormalisierte Zahlen: E=0
  - Mantisse $m = M / 2^{23}$ (statt $m = 1 + M / 2^{23}$)
  - Exponent $e = 2^{1-127}$
  - Erlaubt Zahlen näher bei 0 als mit normalisierter Darstellung
- Unendlich: E=255, M=0
- Keine Zahl (Not a Number, NaN): E=255, M>0

Verteilung von IEEE Gleitpunktzahlen

- Dichte der Gleitpunktzahlen nicht uniform, sondern proportional zum Betrag
- Relativer Rundungsfehler beinahe konstant
  - Relativer Fehler: Fehler / Betrag



http://jasss.soc.surrey.ac.uk/9/4/4.html

Numerische Auslöschung

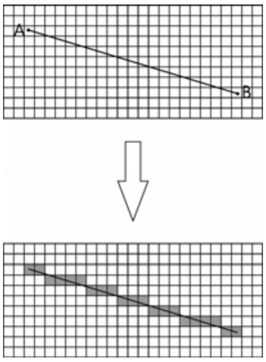
- Beispiel: a=2.345678, b=2.346789
- Exakt: b-a=0.001111
- Mit 2 Nachkommastellen Genauigkeit
  - a~2.35, b~2.35, b-a=0
  - Ergebnis hat keine gültigen Stellen mehr, obwohl 2 Nachkommastellen Genauigkeit zur Verfügung stehen!
- „Rechenoperation vergrössert relativen Fehler viel mehr als absoluten Fehler"
- Beispiel http://de.wikipedia.org/wiki/Ausl%C3%B6schung_%28numerische_Mathematik%29

Fazit

- Reelle Zahlen müssen im Computer approximiert werden
- Standardisierte Darstellung mit IEEE Gleitpunktzahlen
- Rundungsfehler sind unvermeidlich
- Bei ungeschickter Konstruktion von numerischen Algorithmen können Rundungsfehler zu katastrophal falschen Resultaten führen
  - Numerische Auslöschung

Beispiel: Rasterkonvertierung

- Strecke gegeben durch Endpunkte
- Soll auf einem Rasterschirm abgebildet werden
- Anforderungen
  - Strecke soll zusammenhängend sein
  - Konvertierung soll zeiteffizient sein
- Direkte Auswertung der Geradengleichung ist nicht optimal



Algorithmus von Bresenham

- Rundung des Anfangs- und Endpunkts auf integer x1, y1, x2, y2
- Fallunterscheidung
  - Annahme: Steigung zwischen 0 und 45 Grad
    dx = x2-x1 > 0
    dy = y2-y1 > 0
    (y2-y1)/(x2-x1) <=1
  - Total 8 Fälle, lassen sich leicht auf diesen einen zurückführen
  - Anderes Vorzeichen, Vertauschung von x und y

Algorithmus von Bresenham

- In jedem Schritt x um 1 vergrössern (inkrementieren)
  - Schritte nummeriert mit i
  - 2 Möglichkeiten
    $y_{i+1} = y_i$ oder
    $y_{i+1} = y_i + 1$
  - Diejenige wählen, die kleineren Fehler ergibt
- Fehlermass e
  - Initialiasieren auf e = 2*dx-dy

Algorithmus von Bresenham

```
// runde x, y; berechne dx, dy
for i=0 to dx do
  setpixel(x,y)
  if e>0
     y = y+1
     e = e-2*dx
  x = x+1
  e = e+2*dy
```

- Nur Integer Additionen und Subtraktionen

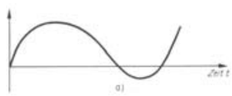
Diskretisierung

- Zur Erinnerung: Diskretisierung beinhaltet Abtastung und Quantisierung
- Beispiel: Tonsignal $s = f(t)$
  - s bedeutet Schalldruck zu jeder Zeit t

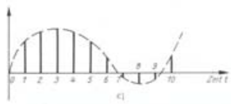
1. Zeitquantisierung
   $t_i = t_0 + i*dt$
2. Abtastung
   $s_0 = f(t_0),\ s_1 = f(t_1),\ \ldots,\ s_n = f(t_n)$
3. Wertquantisierung
   $s_0, s_1, \ldots, s_n$ als float oder integer dargestellt

Diskretisierung

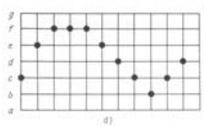
Eingabe, kontinuierliches Signal



Zeitquantisierung & Abtastung

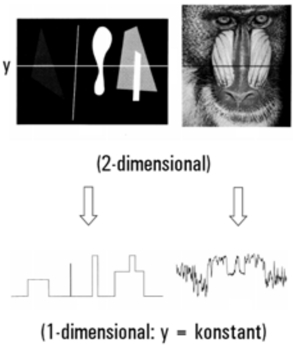


Wertquantisierung, diskretes Signal



Diskretisierung

- Beispiel: 2D Bilder



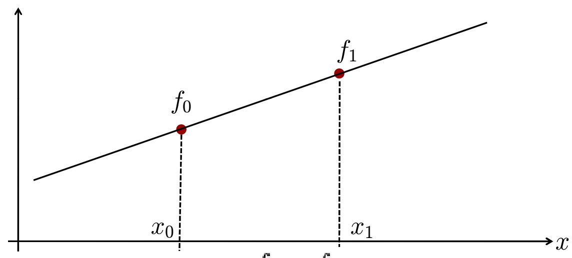
Interpolation

- Wollen Signal an beliebiger Stelle auswerten, nicht nur an Abtastpositionen
- „Umkehroperation“ der Abtastung
  - Gegeben diskrete Daten, finde kontinuierliche Funktion, welche Daten abbildet
  - Kontinuierliche Funktion interpoliert Daten
- Interpolation ist i.A. nicht eindeutig definiert, viele Möglichkeiten

  http://de.wikipedia.org/wiki/Interpolation_%28Mathematik%29
  - Verschiedene Qualitätsmerkmale, z.B. Stetigkeit
  - In gewissen Fällen „perfekte“ Interpolation möglich

Lineare Interpolation

- Interpoliert genau 2 Abtastpunkte



$$f(x) = f_0 + \frac{f_1 - f_0}{x_1 - x_0}(x - x_0)$$

Interpolationsproblem

- Gegeben: Stützpunkte (Abtaststellen, Funktionswerte) $(x_0, f_0), (x_1, f_1), \dots (x_{n-1}, f_{n-1})$
- Gesucht: Funktion $f(x)$, so dass $f(x_i) = f_i$
- Mögliche Lösung

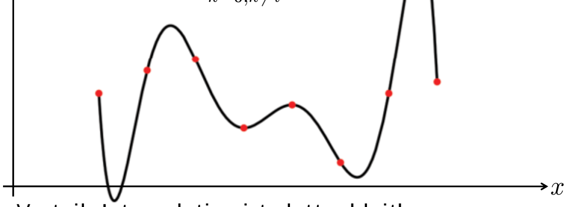
$$f(x) = \sum_{i=0}^{n-1} f_i L_i(x)$$

  - Kontinuierliche Basisfunktionen $L_i(x)$
  - wobei $L_i(x_j) = \delta_{ij}$
  - Kronecker-Delta $\delta_{ij} = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j \end{cases}$

Polynominterpolation

- Lagrange Polynome

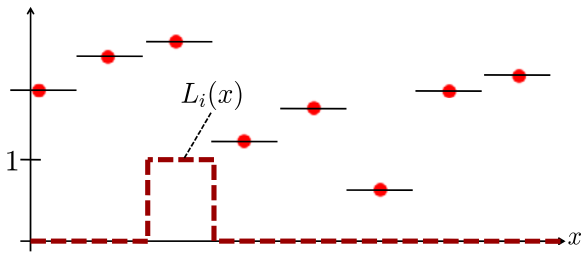
$$L_i(x) = \prod_{k=0, k \neq i}^{n-1} \frac{x - x_k}{x_i - x_k}$$



- Vorteil: Interpolation ist glatt, ableitbar
- Nachteil: übermässige Schwingungen

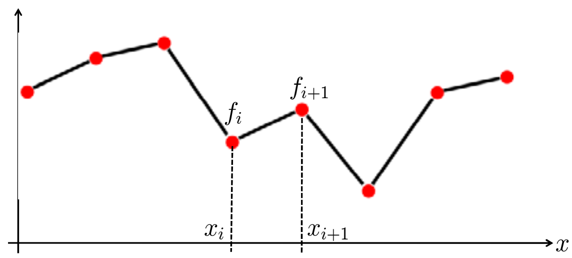
Stückweise konstante Interpolation

- Basisfunktionen stückweise konstant
- Interpolation nicht stetig



Stückweise lineare Interpolation

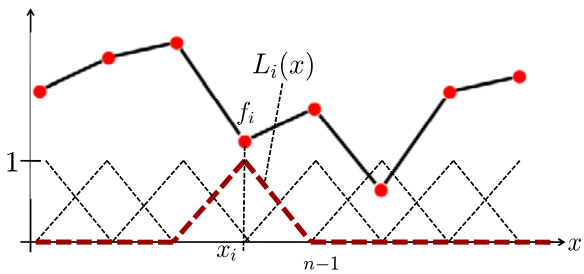
- Stetig, aber nicht ableitbar



$$f(x) = f_i + \frac{f_{i+1} - f_i}{x_{i+1} - x_i}(x - x_i), \quad x_i \le x < x_{i+1}$$

Stückweise lineare Interpolation

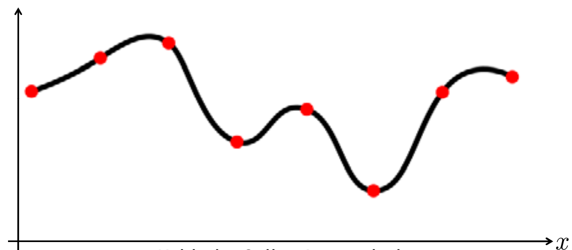
- Interpretation mit Basisfunktionen (lineare Splines)
- Basisfunktionen stückweise linear, lokaler Support



$$f(x) = \sum_{i=0}^{n-1} f_i L_i(x)$$

Spline Interpolation

- Stückweise Polynome höherer Ordnung
- Stetige Basisfunktionen, stetige Interpolation

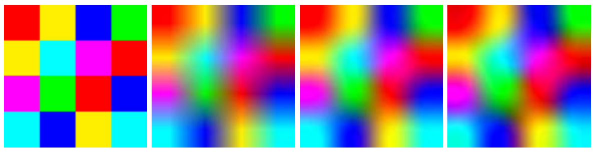


Kubische Spline Interpolation

Mehr auf http://de.wikipedia.org/wiki/Spline-Interpolation

Skalieren von Rasterbildern

- Skalieren: Interpolation & Neuabtastung
- Eingabe: 4x4 Pixels 



Nächste Nachbar Interpolation	Bilineare Interpolation	Bikubische Interpolation	Lanczos-3-Interpolation

http://de.wikipedia.org/wiki/Interpolation_%28Mathematik%29#Anwendungen

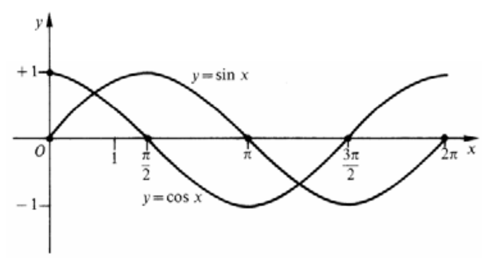
Zusammenfassung

- Interpolation mittels gewichteter Summe von kontinuierlichen Basisfunktionen
- Interpolation immer gewährleistet, wenn jede Basisfunktion an genau einer Abtaststelle Wert 1 hat, an allen anderen Wert 0
- Polynome, stückweise definierte Polynome
- Konstruktion der Basisfunktionen $L_i(x)$ bis jetzt unabhängig von Funktionswerten f_i
  - Muss nicht so sein

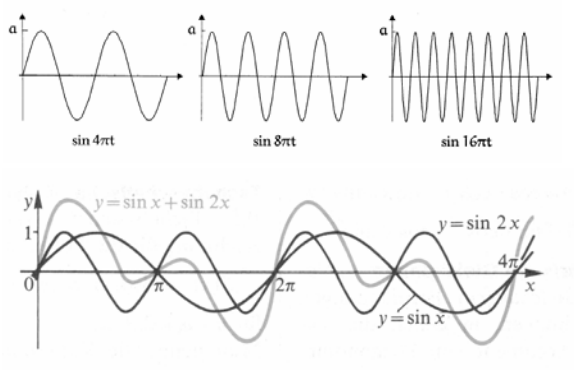
Trigonometrische Basisfunktionen

- Interpolation der Form

$$f(x) = a_0 + a_1\cos(x) + b_1\sin(x) + a_2\cos(2x) + b_2\sin(2x) + \ldots$$



Beispiele



Frequenzen

- Kreisfrequenz ω in $\sin(\omega x)$, $\cos(\omega x)$
  - Periode $2\pi/\omega$
  - Auf Intervall der Länge 2π hat es ω Wiederholungen
- Frequenz ν in $\sin(2\pi\nu x)$, $\cos(2\pi\nu x)$
  - Periode $2\pi/2\pi\nu = 1/\nu$
  - Auf Intervall der Länge 1 hat es ν Wiederholungen

Fourier Analyse

- Benannt nach J. Fourier (1768-1830)
- Gegeben: beliebige periodische Fuktion f mit Periode $T=2\pi$
- Funktion f lässt sich durch Superposition (Summation) von Basisfunktionen $a_k \cos(kx), b_k \sin(kx)$ beliebig genau rekonstruieren
  - Fourier-Reihe von f
  - Fourier Koeffizienten a_k, b_k müssen geeignet berechnet werden

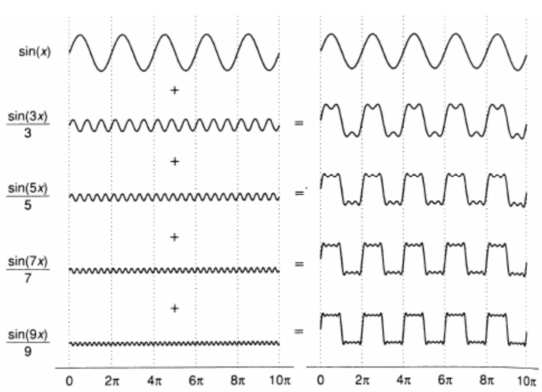
Fourier Koeffizienten

- Gegeben: f periodisch mit $T=2\pi$

1. Ein Intervall $[x_0, x_0+T]$ in $2n$ gleich lange Stücke teilen
   $x_0, x_0 + \frac{T}{2n}, \ldots, x_0 + (2n-1)\frac{T}{2n}$
2. f abtasten
   $f_k = f(x_k)$
3. Koeffizienten berechnen
   $a_k = 1/n \sum_{i=0}^{2n-1} f_i \cos(kx_i) \quad b_k = 1/n \sum_{i=0}^{2n-1} f_i \sin(kx_i)$

- Approximation
  $f(x) \approx 1/n \sum_{k=0} 2n - 1(a_k \cos(kx) + b_k \sin(kx))$

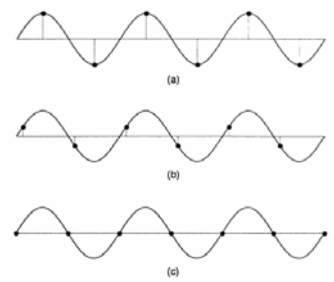
Beispiel



Abtasttheorem

- Eine beliebig genaue Rekonstruktion von f ist möglich, falls
  1. f ist bandbegrenzt, d.h., für alle in f auftretenden Frequenzen ω gilt $|\omega| < \omega_G$ (Grenzfrequenz, Bandbreite)
  2. f wird höchstens in Abständen von $dt < 1/(2\omega_G)$ abgetastet
- Abtastfrequenz heisst Nyquist Frequenz (H. Nyquist, 1924)
- Details
  http://de.wikipedia.org/wiki/Nyquist-Shannon-Abtasttheorem

Nyquist Frequenz



Abtastfrequenz $= 2\omega_G$

Nyquist Frequenz

- Unterabtastung führt zu Aliasing
- Abgetastete Funktion erscheint als tiefere Frequenz als Eingabe

Abtastfrequenz $< 2\omega_G$

Übersicht

Modellierung und Simulation

- Einleitung
- Diskretisierung und Interpolation
- Kontinuierliche Modelle: Differential-gleichungen
- Diskrete Modelle: Warteschlangen

Differentialgleichungen

- Weitverbreitete Form, um Modelle zu formulieren
  - Grosse Anzahl physikalischer Prozesse durch Differentialgleichungen modelliert
- Untersuchte Grössen sind kontinuierlich, häufig kontinuierliche Funktionen
  - Kräfte, Geschwindigkeiten
  - Kräfte-, Geschwindigkeits-, Druck-, Ladungs-, Wärmeverteilungen

Differentialgleichungen

- Setzen Änderungen der untersuchten Grössen miteinander in Verbindung
  - Änderungen kontinuierlicher Grössen sind Ableitungen
- „Mathematische Gleichungen, in der eine gesuchte Funktion, die von einer oder mehreren Variablen abhängt, und Ableitungen dieser Funktion auftreten“

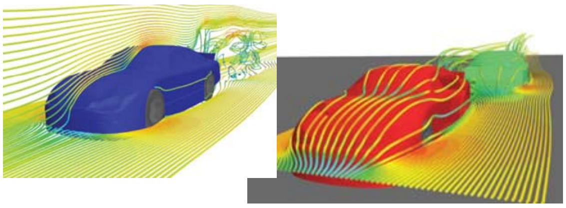
Differentialgleichungen

Bekannte Gleichungen

- Wellengleichung
  - Ausbreitung von Schall, elektromagnetischen Wellen (Licht), etc.
- Bewegungsgleichung
  - $f=ma$ und Verallgemeinerungen
- Wärmeleitungsgleichung
- Navier-Stokes Gleichung
  - Verhalten von Flüssigkeiten und Gasen
- Etc.

Beispiel

- Simulation von Luftströmungen mittels Navier-Stokes Gleichung



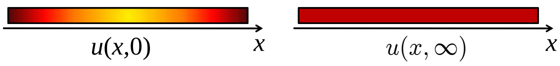
http://www.eng.fea.ru/FEA_news_433.html

Differentialgleichungen

- Können nur in Spezialfällen direkt gelöst werden
- Lösung häufig durch Computersimulation
- Riesiges Gebiet in der (angewandten) Mathematik

Wärmeleitungsgleichung

- Ziel: Modelliere Wärmeverteilung auf einer eindimensionalen Strecke, z.B. in dünnem Stab
  - Gegeben: Temperaturverteilung $u(x,0)$ zu einer bestimmten Zeit $t=0$
  - Gesucht: $u(x,t)$, Temperatur an Position x zur Zeit t
- Intuition: erwarten, dass sich Temperatur über Zeit regelmässig verteilt



Wärmeleitungsgleichung

- Modell $$\frac{\partial}{\partial t}u(x,t) = a \cdot \frac{\partial^2}{\partial x^2}u(x,t)$$
  - Materialkonstante a
- „Veränderung der Temperatur über die Zeit an einem Punkt x ist proportional, mit Konstante a, zur zweiten Ableitung der Temperaturverteilung über den Raum"
- Rechtfertigung für dieses Modell
  - Stimmt mit Experimenten überein
  - Kann aus fundamentaleren Überlegungen abgeleitet werden (z.B. Energieerhaltung), siehe z.B. http://en.wikipedia.org/wiki/Heat_equation

Wärmeleitungsgleichung

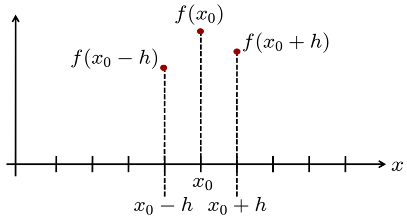
- Modell $$\frac{\partial}{\partial t}u(x,t) = a \cdot \frac{\partial^2}{\partial x^2}u(x,t)$$
  - Materialkonstante a
- Lineare partielle Differentialgleichung zweiter Ordnung
  - Linear: unbekannte Funktion u kommt linear vor
  - Partiell: enthält partielle Ableitungen
  - Zweite Ordnung: höchste Ableitung hat Grad 2

Lösung

- Analytisch
  - Lösung direkt als Formel angeben
  - Wärmeleitungsgleichung: z.B. über Fourierreihen http://en.wikipedia.org/wiki/Heat_equation#Solving_the_heat_equation_using_Fourier_series
  - Häufig unmöglich
- Nummerisch
  - Durch Diskretisierung
  - Müssen nicht nur Funktion selbst, sondern auch Ableitungen diskretisieren
  - Einfachstes Verfahren: finite Differenzen
  - Viele raffiniertere Verfahren existieren

Finite Differenzen

- Funktion und Ableitungen approximieren mittels diskretem Gitter
- Beispiel: Funktion einer Variablen
  - Gittergrösse h



Finite Differenzen

- Funktion und Ableitungen approximieren mittels diskretem Gitter
  - Gittergrösse h
- Approximation der ersten Ableitung
  - Vorwärtsdifferenzquotienten
    $$D^+(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h},$$
  - Rückwärtsdifferenzquotienten
    $$D^-(x_0) \approx \frac{f(x_0)-f(x_0-h)}{h}.$$
  - Zentrale Differenzquotienten (bevorzugt, besserer Fehler)
    $$D(x_0) \approx \frac{f(x_0+h)-f(x_0-h)}{2h}.$$

Finite Differenzen

- Approximation der zweiten Ableitung durch wiederholte Anwendung der ersten Ableitung

$$D^+(D^-(x_0)) \approx \frac{f(x_0+h)-2f(x_0)+f(x_0-h)}{h^2}$$

Finite Differenzen

- Kontinuierliche Version
$$\frac{\partial}{\partial t}u(x,t) = a \cdot \frac{\partial^2}{\partial x^2}u(x,t)$$
- Approximation mit finiten Differenzen
$$\frac{u_j^{n+1}-u_j^n}{k} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2}.$$
  - Diskrete Zeitschritte indiziert mit n, Schrittweite k
  - Diskrete Raumschritte indiziert mit j, Schrittweite h

Finite Differenzen

- Kontinuierliche Version
$$\frac{\partial}{\partial t}u(x,t) = a \cdot \frac{\partial^2}{\partial x^2}u(x,t)$$
- Approximation mit finiten Differenzen
$$\frac{u_j^{n+1}-u_j^n}{k} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2}.$$
  - Vorwärtsdifferenz für Zeit
  - Zentrale Differenz für Raum
  - Explizites Verfahren (explizite Euler Integration)
  - Konvergiert nur falls $k/h^2 < 1/2$
  - Details http://en.wikipedia.org/wiki/Finite_difference_method

Finite Differenzen

- Berechnung der neuen Werte zur Zeit $n+1$ durch Auflösung nach u_j^{n+1}
$$\frac{u_j^{n+1}-u_j^n}{k} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2}.$$

ergibt
$$u_j^{n+1} = (1-2r)u_j^n + ru_{j-1}^n + ru_{j+1}^n$$

wobei $r = k/h^2$

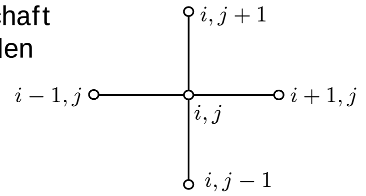
Erweiterung auf 2D

- Finite Differenzen ($r = k/h^2$)

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{k} = \frac{u_{i+1,j}^{n} + u_{i,j+1}^{n} + u_{i-1,j}^{n} + u_{i,j-1}^{n} - 4u_{i,j}^{n}}{h^2}$$

$$u_{i,j}^{n+1} = (1-4r)u_{i,j}^{n} + r(u_{i+1,j}^{n} + u_{i,j+1}^{n} + u_{i-1,j}^{n} + u_{i,j-1}^{n})$$

- 2D Nachbarschaft der Gitterzellen



Algorithmus

- Initialisiere 2D Feld u_{ij} mit gewünschten Temperaturwerten zur Anfangszeit
- In jedem Zeitschritt
  - Für alle Elemente ij, berechne neuen Temperaturwert und speichere in temporärem Feld t_{ij}
  - Für alle Elemente ij, kopiere temporäre Werte t_{ij} nach u_{ij}

Zusammenfassung

- Differentialgleichungen: Gleichung, in der unbekannte Funktion und Ableitungen davon vorkommen
- Grundlage für grosse Anzahl physikalischer Modelle
- Nummerische Lösung: nicht nur Funktion, sondern auch Ableitungen diskretisieren
- Finite Differenzen: Funktion und Ableitungen auf Gitter diskretisieren
- Beispiel: Wärmeleitungsgleichung
  - Diskretisierung mit explizitem Verfahren
  - Neue Werte schrittweise über Zeit berechnen

Übersicht

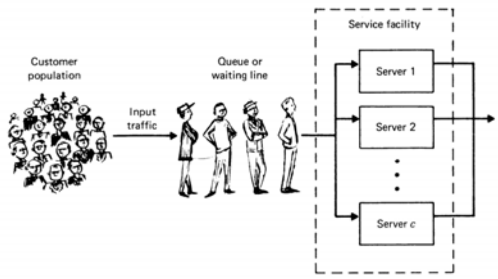
Modellierung und Simulation

- Einleitung
- Diskretisierung und Interpolation
- Kontinuierliche Modelle: Differential-gleichungen
- Diskrete Modelle: Warteschlangen

Diskrete Simulation

- Modell besteht typischerweise aus Computerprogramm, das in „zufälligen“ Zeitabständen Ereignisse (Events) generiert und das modellierte System auf den neuen Stand bringt
- Am Schluss der Simulation wird eine statistische Auswertung (Evaluation**)** vorgenommen
- Beispiele
  - Modellierung und Simulation von Warteschlangen, z.B. Verkehrskreuzung
  - Geplantes Computersystem simulieren

Warteschlangen



- Server: z. B. Team von Arbeitern, Prozessor, Übertragungsleitung, Strassen-abschnitt, etc.

Beispiel: Coiffeurgeschäft

- Nur ein Coiffeur
- Immer nur ein Kunde bedient, ohne Unterbrechung
- Keine Reservationen
  - First-come First-served (FCFS)
- Genügend Platz zum Warten
- Kein Kunde geht unbedient weg

Nutzen der Simulation

Messungen

- Mittlere Bedienzeit? Wartezeit?
- Mittlere Wartezeit der Kunden, die warten müssen?
- Wie stark ist der Coiffeur ausgelastet?
  - (= „busy"-Zeit/Gesamtzeit, oft in % ausgedrückt)
- Wie viele Kunden warten im Mittel? Maximal?
- Wie viele Kunden warten höchstens 10 Minuten?
- Etc.

Entscheidungsgrundlagen

- Genügt ein Coiffeur?
- Sollten die Preise geändert werden?
- Etc.

Statistiken

- Zur Auswertung der Simulation
- In periodischen Abständen messen
- Mögliche interessante Daten
  - Anzahl angekommener Kunden
  - Summe aller bisherigen Wartezeiten
  - Durchschnittliche Wartezeit
  - Durchschnittliche Länge der Schlange
  - Summe der freien Zeiten des Coiffeurs

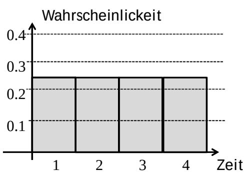
Parameter

- Konstanten
  - Anzahl Warteplätze in der Schlange
- Zufallsvariablen
  - Zwischenankunftszeit (Zeit zwischen Ankunft zweier Kunden)
  - Bedienzeit

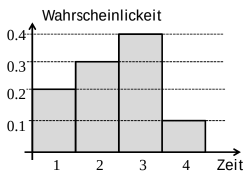
Parameter

- Zufallsvariablen modelliert mit Hilfe von Wahrscheinlichkeitsverteilungen
  - Diskret: Wahrscheinlichkeit, dass Zufallsvariable einen diskreten Wert annimmt
  - Kontinuierlich: Dichtefunktion, Integral 1
- Diskrete Beispiele

Gleichverteilung



Beliebige Verteilung



Erzeugung von Zufallszahlen

- Algorithmisch erzeugte Pseudozufalls-zahlen
- Linearer Kongruenzgenerator

$$y_i = (ay_{i-1} + b) \bmod m$$

  - Sequenz von Pseudozufallszahlen y_i
  - Parameter a, b, m
- Günstige Wahl der Parameter

  $a = 7^5 = 16807, b = 0, m = 2^{31} - 1 = 2147483647$

- Details
  http://de.wikipedia.org/wiki/Linearer_Kongruenzgenerator

Erzeugung von Zufallszahlen

- Brauchen Test für „Güte“ der Sequenz von Zufallszahlen
  - Möglichst lange Periode
    - Linearer Kongruenzgenerator: Periode höchstens so lange wie Wertebereich
  - Aufeinanderfolgende Zahlen voneinander unabhängig
- Viele andere Tests möglich
- Besser als Kongruenzgeneratoren z.B. Mersenne-Twister http://de.wikipedia.org/wiki/Mersenne-Twister

Erzeugung von Zufallszahlen

- Gleichverteilte Zahlen im Bereich $[0,1)$ mit Hilfe von $x_i = y_i / m$
- Diskrete Gleichverteilung, z.B. über $1..6$ aus $\lfloor 6x_i + 1 \rfloor$
- Knobelaufgabe: Erzeugung von Zufallszahlen mit beliebiger gewünschter Zufallsverteilung?
  - Diskret
  - Kontinuierlich

Zusammenfassung

- Möglichkeiten und Anwendungen der Computersimulation kennen
- Grundproblem der Diskretisierung und Interpolation
  - IEEE Fliesspunktzahlen
  - Lineare Interpolation, Polynominterpolation
  - Stückweise Interpolation
  - Fourier Reihen
- Stetige Simulation
  - Differentialgleichungen am Beispiel Wärmeleitung
  - Diskretisierung mit finiten Differenzen
- Diskrete Simulation
  - Warteschlangen
  - Generierung von Zufallszahlen