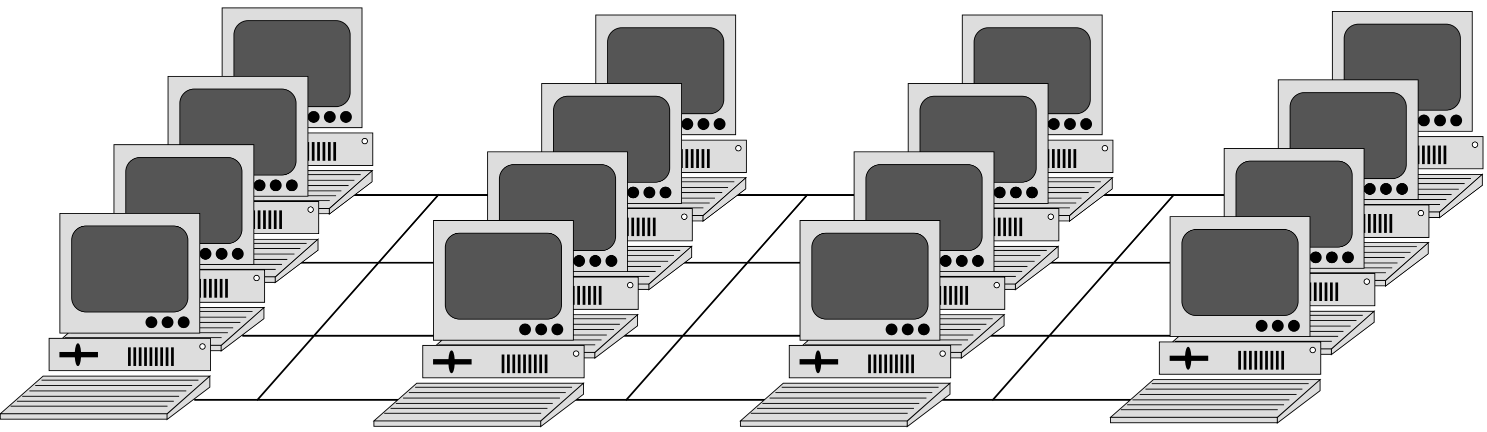
Die numerische Behandlung großer Gleichungssysteme auf Parallelrechnern

Kees Oosterlee

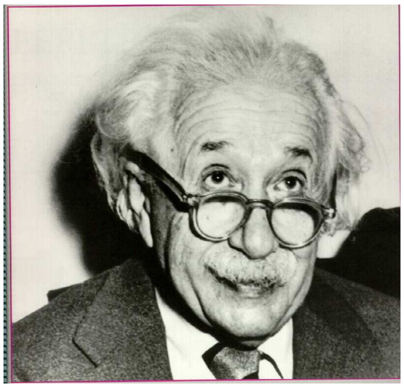
Antrittsvorlesung

Universität zu Köln

30. April 1999



Einführung



"Insofern sich die Sätze der Mathematik auf die Wirklichkeit beziehen, sind sie nicht sicher, und insofern sie sicher sind, beziehen sie sich nicht auf die Wirklichkeit."

- reine Mathematik – angewandte Mathematik – numerische Mathematik

- unterschiedliche Zielgebiete, unterschiedliche Methoden

- Numerische Mathematik:
Herleitung und Untersuchung konstruktiver Methoden zur (praktischen) Berechnung gewisser Größen

- Ziel: Algorithmen, Verfahren (und Lösungen)

- Methoden: z.T. die der klassischen Analysis und linearen Algebra
(Existenzbeweise, Konvergenzbeweise, Eindeutigkeitsaussagen, exakte Abschätzungen)
aber auch andere wie in experimentellen Wissenschaften: Numerische Tests, Erfahrung, geschicktes Vorgehen, (intuitives) Verständnis

Numerische Aufgabenstellung

Scheinbar: nicht exaktes Vorgehen

- Ein wichtiger Aspekt der Numerik ist aber die Beschäftigung mit Fehlern

$\Rightarrow$ Man möchte die exakte Lösung x berechnen und erhält eine Näherung $\hat{x}$

$\Rightarrow$ Berechne eine Näherung $\hat{x}$ für die Lösung x mit einer bestimmten Genauigkeit

(hierbei sollen mögliche Fehler und deren Einfluß auf die Genauigkeit berücksichtigt werden)

- Die Ermittlung einer Näherungslösung ohne jegliche Kenntnis bzgl. des Fehlers ist sinnlos !

Fehlermaße

Für $x \in \mathbb{R}$:

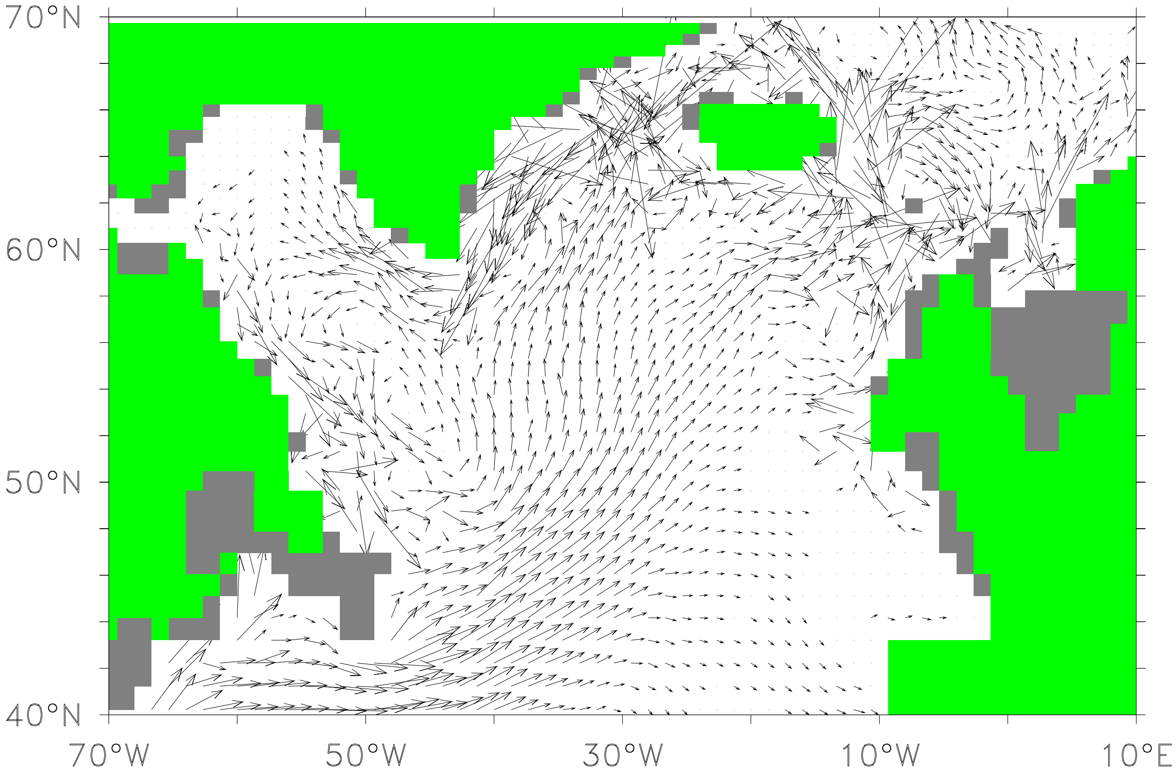
$$|x - \hat{x}| \leq \epsilon \quad \text{Abschätzung des absoluten Fehlers}$$

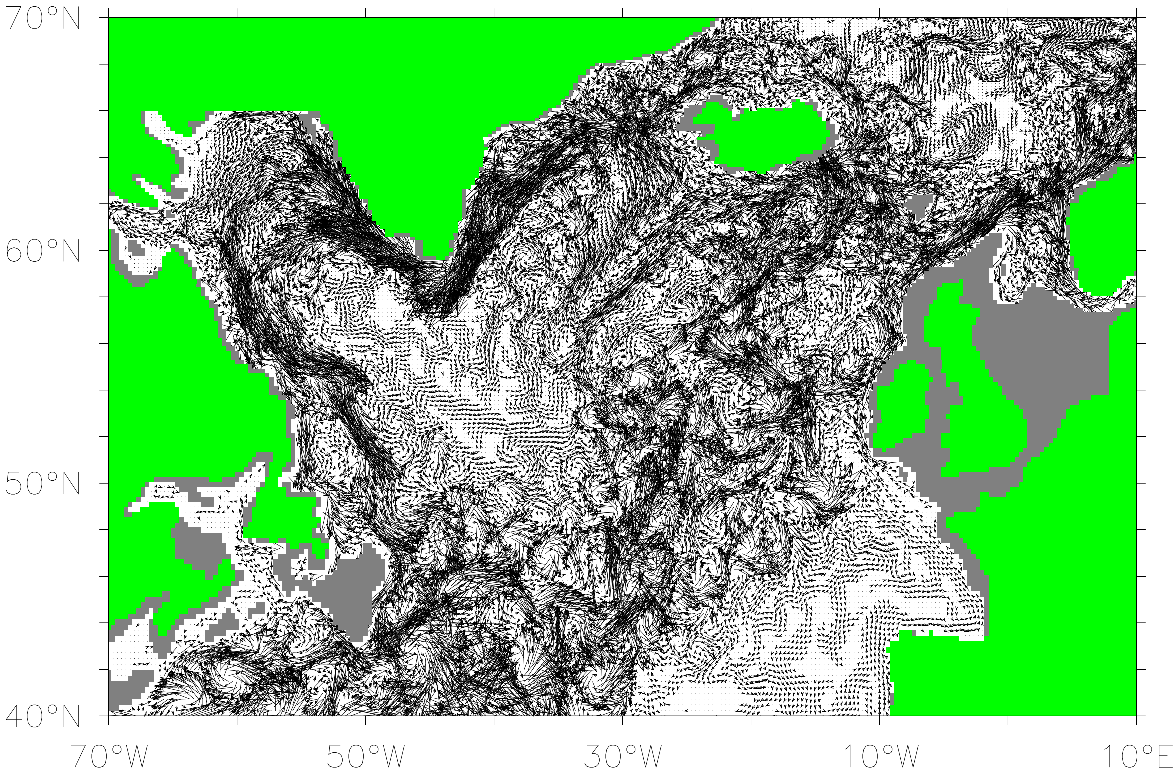
$$\frac{|x - \hat{x}|}{|x|} \leq \epsilon \quad \text{Abschätzung des relativen Fehlers}$$

- Spannungsfeld: Abschätzungen sind nur interessant, wenn sie halbwegs realistisch sind

Schema

Fehler


70°N
60°N
50°N
40°N
70°W
50°W
30°W
10°W
10°E


70°N
60°N
50°N
40°N
70°W
50°W
30°W
10°W
10°E

Diskretisierungsfehler

Nur n Werte berücksichtigt: ergibt i.A. nur eine Näherungskurve.

Modellfehler

• Praktische Probleme sind z.B. physikalische, chemische, biologische oder ökonomische Probleme.

Dann erforderlich: Modellbildung

z.B: lineare Vereinfachung nichtlinearer Zusammenhänge

Newton´sche Mechanik – Quantenmechanik

In der Strömungsmechanik:

Euler-Gleichungen $\longleftrightarrow$ Navier-Stokes-Gleichungen
(reibungsfreies – reibungsbehaftetes Modell)

$\Rightarrow$ Auswirkung auf Ergebnisse

Differentialgleichungen

Wasser fließt mit einer Geschwindigkeit U in einem dünnen Rohr

Gesucht: Die Temperaturverteilung T (des Wassers)

(Annahme: Geschwindigkeit U ist konstant)

- Konvektion (Wärme wird vom Medium Wasser mitgeführt)
- Wärme-Diffusion (Wärme breitet sich auch im ruhendem Medium aus) mit Wärmeleitfähigkeit D

$\Rightarrow$ stationäre Energiebilanz liefert die (1D)
Konvektions-Diffusionsgleichung für T

$$-D\frac{d^2T}{dx^2} + U\frac{dT}{dx} = f(x) \quad \text{für} \quad x \in [0,1]$$

mit den Randbedingungen: $\quad T(0) = g_0, \quad T'(1) = g_1$

Wir suchen die Lösung $T(x)$ numerisch durch "Diskretisierung".

- Das Intervall $[0,1]$ wird in n äquidistante Teilintervalle $[x_i, x_{i+1}]$ eingeteilt

Die Punkte x_i und die Maschenweite

$$h = \frac{x_n - x_0}{n} = \frac{1}{n}$$

bestimmen ein eindimensionales Gitter, auf dem die Lösungsfunktion $T(x)$ näherungsweise bestimmt werden soll.

1.2.0 Finite Differenzen

Dazu brauchen wir Näherungsformeln für alle Ableitungen
Wir entwickeln T am Knoten $i-1$ in eine Taylorreihe

$$T_{i-1} \;=\; T_i - h\frac{dT_i}{dx} + \frac{1}{2}h^2\frac{d^2T_i}{dx^2} - \frac{1}{6}h^3\frac{d^3T_i}{dx^3} + \ldots$$

(Für T_{i-2} findet man eine ähnliche Entwicklung)

Hieraus erhält man als Approximation der Ableitung

$$\frac{dT_i}{dx} \simeq \frac{T_i - T_{i-1}}{h} \quad (V1), \qquad \text{oder z.B.} \qquad \frac{dT_i}{dx} \simeq \frac{3T_i - 2T_{i-1} + T_{i-2}}{2h} \quad (V2)$$

und

$$\frac{d^2T_i}{dx^2} \simeq \frac{T_{i+1} - 2T_i + T_{i-1}}{h^2}$$

Diskretisierung der stationären Konvektions-Diffusionsgleichung

Diese finiten Differenzen Diskretisierungen lassen sich durch die Einführung eines Parameters α verallgemeinern:

$$-D\frac{T_{i-1} - 2T_i + T_{i+1}}{h^2} + U\left(\alpha\frac{T_i - T_{i-1}}{h} + (1-\alpha)\frac{3T_i - 2T_{i-1} + T_{i-2}}{2h}\right) = f(x_i)$$

Die Analyse des lokalen Verfahrensfehlers liefert

• Verfahren V2 liefert $\mathcal{O}(h^2)$ Genauigkeit, V1 nur $\mathcal{O}(h)$ Genauigkeit.

(Eine Halbierung der Maschenweite h verringert den Fehler des V2-Schemas um einen Faktor 4, bei V1 nur um einen Faktor 2.)

$\Rightarrow$ Diskretisierung V2 wäre somit der Diskretisierung V1 vorzuziehen.

• Diskretisierungen liefern (lin.) Gleichungssysteme mit speziellem Struktur

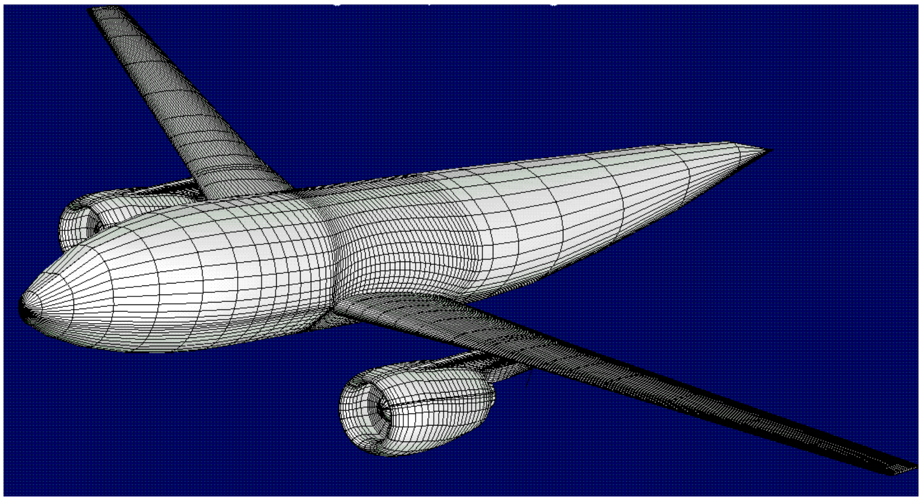
Forderung: Effizienz der numerischen Verfahren zur Lösung diskreter Gleichungssysteme

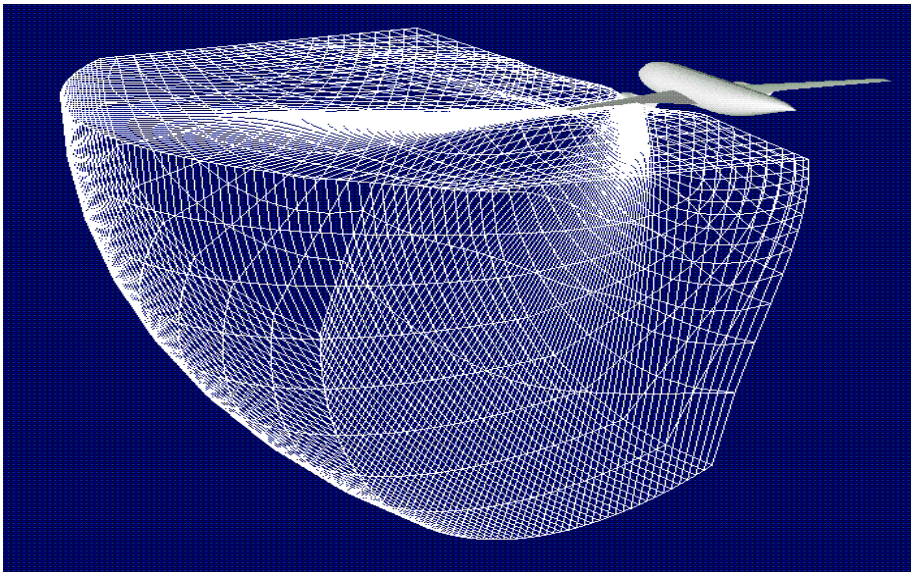
- Rechenzeit / Kosten
- Einfachheit der Anwendung (z.B. Benutzung vorhandener Software)

Verallgemeinerung

Flugzeugentwurf

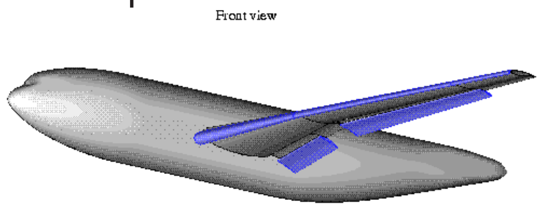
- Berechnung von Auftriebs- und Widerstandskoeffizienten neuer Flugzeuge
- System 3D nichtlinearer partieller Differentialgleichungen für Geschwindigkeit, Druck, Dichte und Temperatur
- Heute: Flügel-Rumpf-Triebwerksgeometrie
- Für eine genaue Auflösung braucht man $\sim$ 2-4 Millionen Gitterpunkte

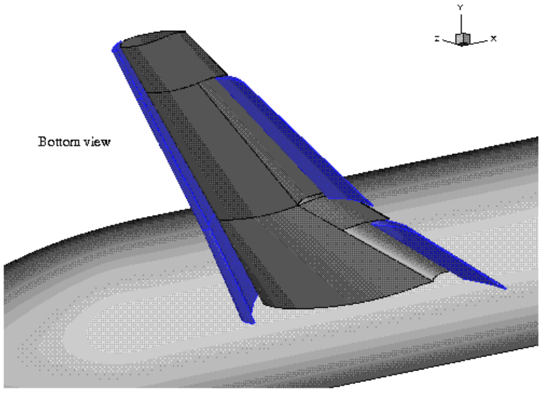


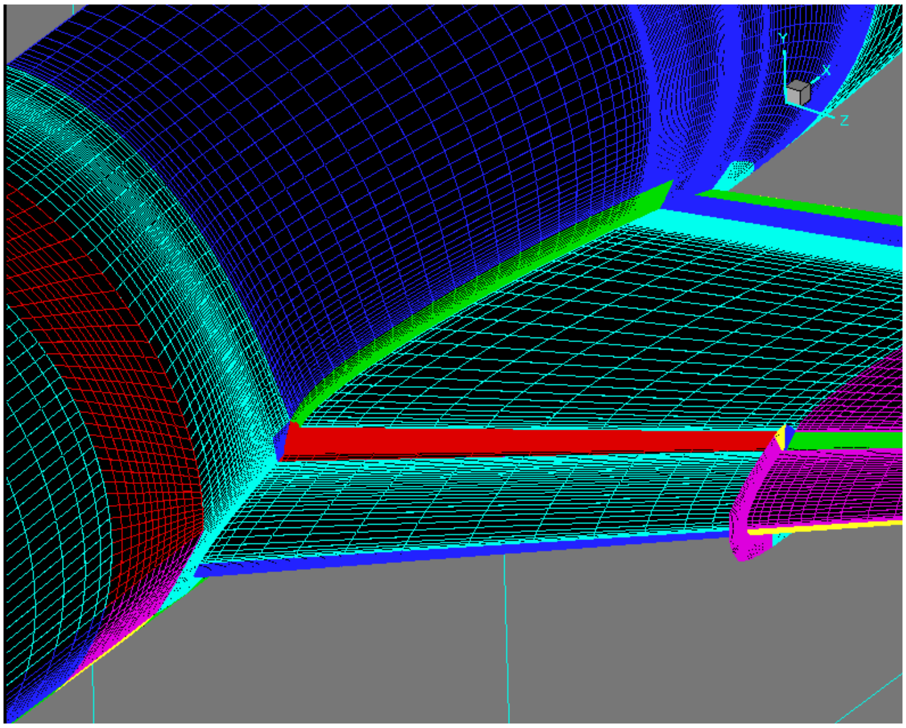


Genauere Modelle

⇒ Für eine genauere Flugzeug-Modellierung und mit besserer mathematischer Beschreibung (z.B der Turbulenz) braucht man noch mehr Gitterpunkte



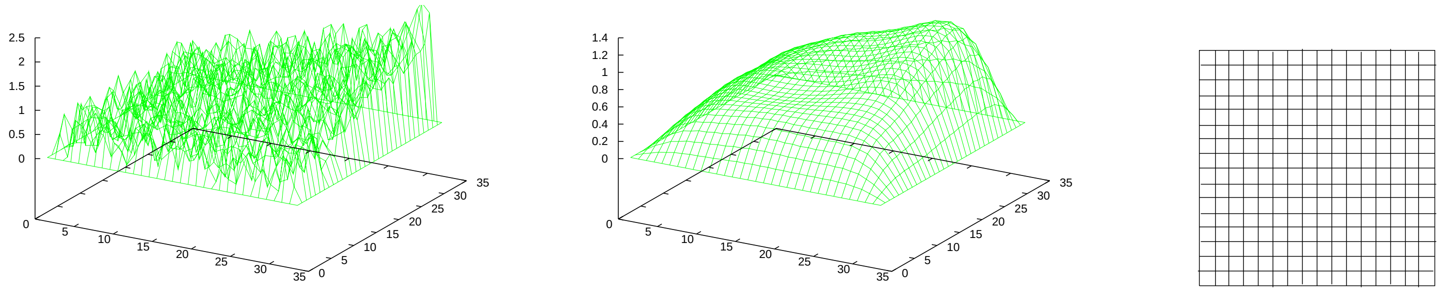




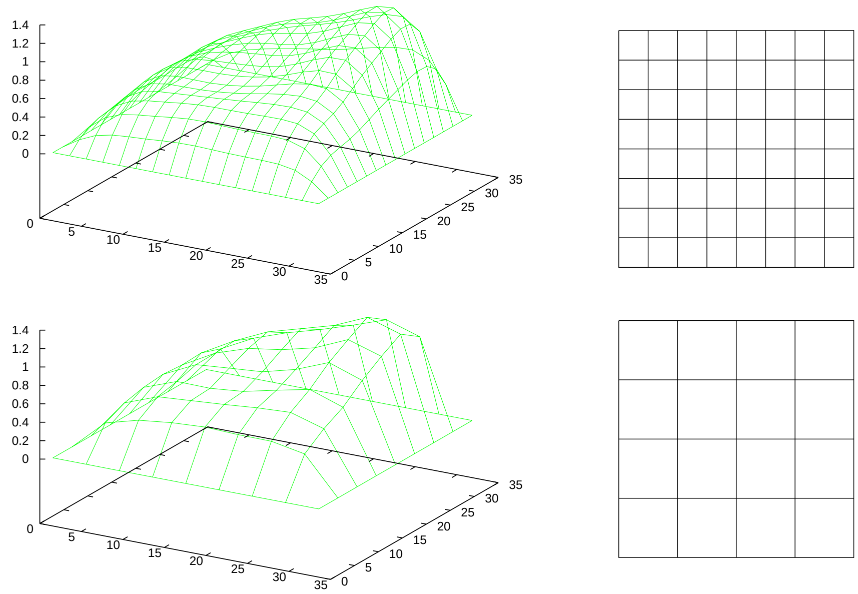
Effiziente Lösungsverfahren für diskrete Gleichungssysteme

Mehrgitterverfahren (seit 1973)

- Iterativen Verfahren wie Jacobi und Gauß-Seidel konvergieren auf feinen Gittern sehr langsam, aber sie glätten den Fehler



- Glatte Fehler können auf groben Gittern mit (viel) weniger Gitterpunkte approximiert werden (und mit weniger Rechenaufwand reduziert werden)



- Mehrgitter ist ein $\mathcal{O}(n)$- Verfahren !

- Für die (3D) Gleichungen aus der Strömungsmechanik ist die Entwicklung effizienter paralleler Mehrgitterverfahren noch immer ein aktuelles Forschungsthema !

Rechenkapazität

$\Rightarrow$ besonderer Aufschwung der Numerik durch maschinelle Hilfsmittel: Rechenmaschinen, Computer

(aber die Quellen der Numerik sind ganz alt: Vermessung, Astronomie. Älter als die Axiomatisierung)

- Flops = Floating point operations per second
  - Mega $= 10^6$ (1 Millionen)
  - Giga $= 10^9$
  - Tera $= 10^{12}$
  - Peta $= 10^{15}$
- Reynoldsgleichungen

	Re	# Gitterpunkte	Speicher (Byte)	Notw. Rechengeschwindigkeit (Lauf in einer Stunde)
Flügel-Rumpf	10^7	2 M	60 M	600 M Flops
Flügel-Rumpf	10^8	4 M	100 M	1200 M Flops

- LES (Bessere Turbulenz Modellierung)

	Re	# Gitterpunkte	Speicher (Byte)	Not. Rechengeschwindigkeit (Lauf in einer Stunde)
Flügel-Rumpf	10^7	0.4 G	5 G	100 G Flops
Flügel-Rumpf	10^8	1 G	15 G	300 G Flops

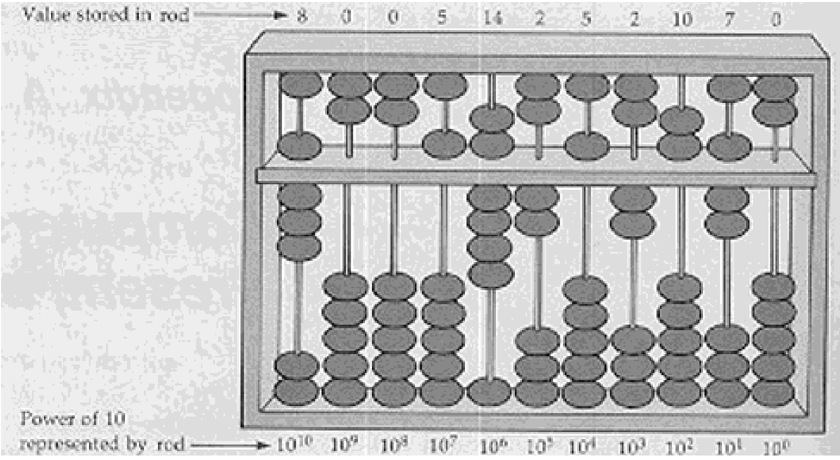
- zukünftiger Superrechner: Earth-Simulator (NEC)
  5 T Flops (sustained) > 30 T Flops (peak)

A Chronology of Computer History:

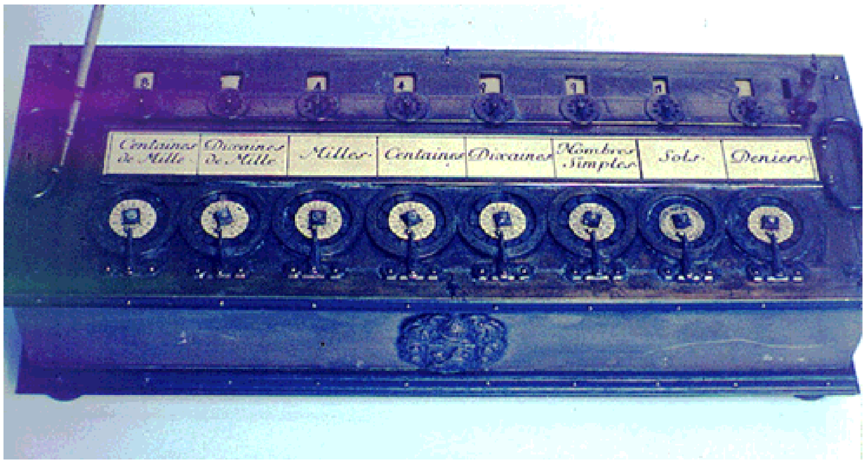
- 3000 BC : Dust abacus is invented, probably in Babylonia.
- 1800 BC : Babylonian mathematician develops algorithms to resolve numerical problems.
- 200 AD : Soroban computing tray used in Japan.
- 1624 : Wilhelm Schickard builds first four-function calculator-clock at the University of Heidelberg.
- 1642 : Blaise Pascal builds the first numerical calculating machine in Paris.
- 1673 : Gottfried Leibniz builds a mechanical calculating machine that multiplies, divides, adds and subtracts.
- 1780 : American Benjamin Franklin discovers electricity.
- 1833 : Charles Babbage designs the Analytical Machine that follows instructions from punched-cards.
  It is the first general purpose computer.
- 1854 : Irishman George Boole publishes The Mathematical Analysis of Logic using the binary system.
- 1876 : Telephone is invented by Alexander Graham Bell.
- 1924 : Computing-Tabulating-Recording Company changes its name to International Business Machines (IBM).

Rechner

Abacus



Pascals Rechner (1642)



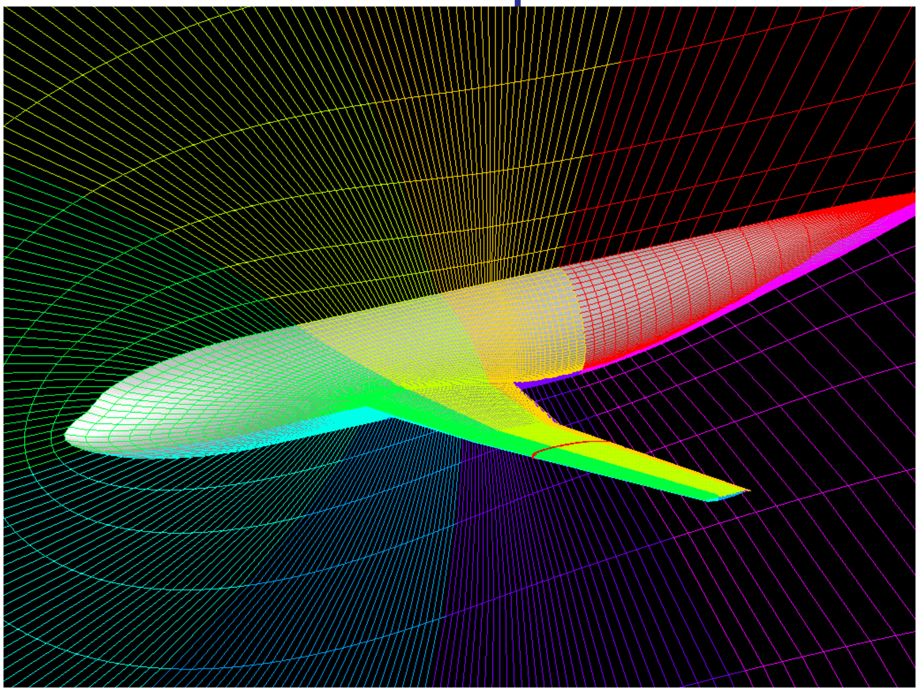
Babbage Maschine (1833)

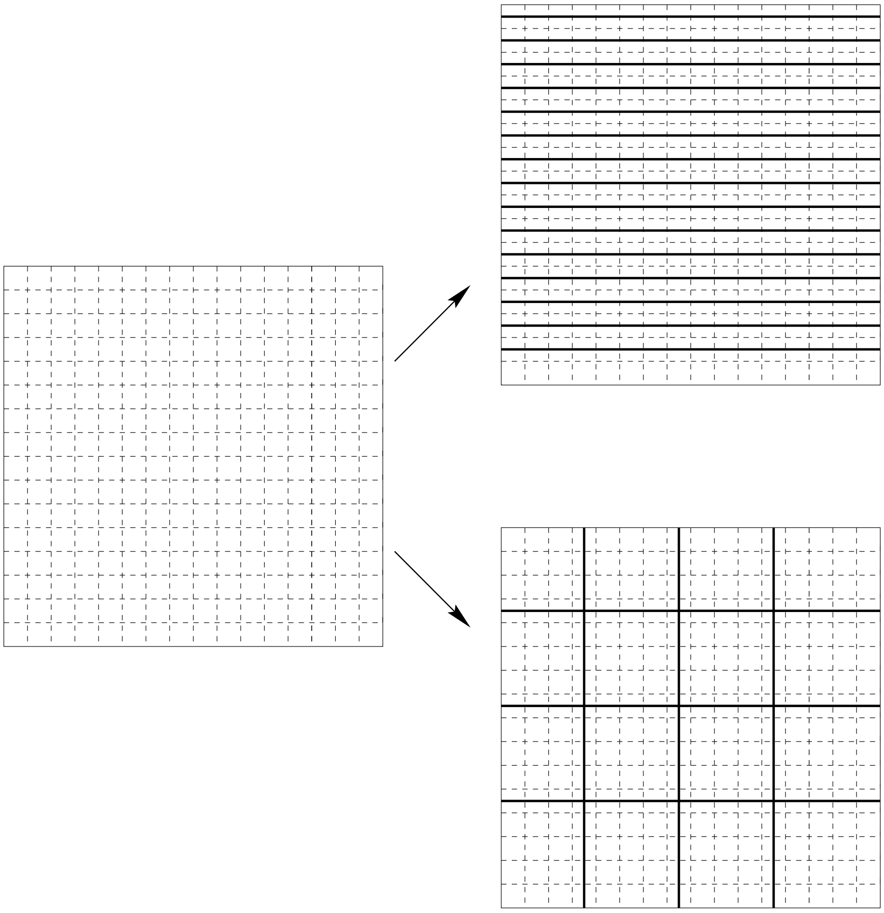
- 1927: First public demonstration of television. Radio-telephone becomes operational between London and New York.
- 1940: First color TV broadcast.
- 1941: Konrad Zuse builds the Z3 computer, the first calculating machine with automatic control of its operations.
- 1946: ENIAC (Electronic Numerical Integrator and Computer), with 18,000 vacuum tubes, is introduced. It was 2.5 by 30.5 meters and weighted 72500 kilograms. It could do 5,000 additions and 360 multiplications per second.
- 1946: Term bit for binary digit is used for first time.
- 1948: Transistor is invented by William Bradford Shockley
- 1953: IBM ships its first stored-program computer, the 701. It is a vacuum tube, or first generation, computer.
- 1954: FORTRAN is created by John Backus at IBM. Harlan Herrick runs the first successful FORTRAN program.
- 1954: Gene Amdahl develops the first operating system, used on IBM 704.
- 1958: First electronic computers are built in Japan by NEC: the NEC-1101 and -1102.
- 1958: Seymour Cray builds the first fully transistorized supercomputer, the CDC 1604.

- 1958: Jack Kilby of Texas Instruments makes the first integrated circuit (IC).
- 1959: IBM ships its first transistorized, or second generation, computers, the 1620 and 1790.
- 1961: IBM delivers the Stretch computer to Los Alamos. This transistorized computer with 64-bit data paths is the first to use eight-bit bytes; it remains operational until l971.
- 1964: Control Data Corporation introduces the CDC 6600, the most powerful computer for several years.
  It was designed by Seymour Cray.
- 1965: IBM ships the first System 360, its first integrated circuit-based, or third generation, computer.
- 1968: Integrated Electronics (Intel) Corp. is founded.
- 1975: Microsoft is founded after Bill Gates and Paul Allen adapt and sell BASIC to MITS for the Altair PC.
- 1976: Seymour Cray engineers and delivers the Cray 1 vectorcomputer with 200,000 freon-cooled ICs and 100 million floating point operations per second (MFLOP) performance.

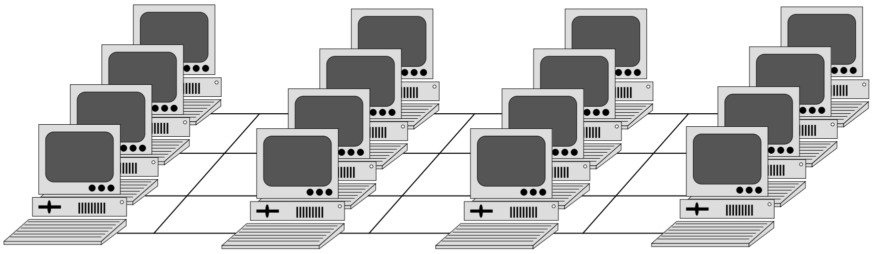
- 1983: Cray 2 vectorcomputer introduced with one billion Flops performance rating.
- 1983: NEC announces the SX-1 and SX-2 supercomputers.
- 1988: Cray Research introduces the Cray Y-MP, a $20M supercomputer.
- 1989: Intel announces the I860 RISC/coprocessor chip with over one million transistors. and launches a parallel supercomputer using over 500 860 RISC microprocessors.
- 1991: Japan announces plans for sixth-generation computers based on neural networks
- 1992: Intel says its next microprocessor will be called Pentium
- 1993: IBM presents its SP series, RISC based parallel computers
- 1993 –>: Huge parallel machines exist in the USA, the ASCI machines (Intel, IBM), the numerical wind channel in Japan (Fujitsu VPP-based)
- 1994: Leonard Adleman introduces ideas in which DNA is used as computing medium

Parallelisierungskonzept: Gitterpartitionierung



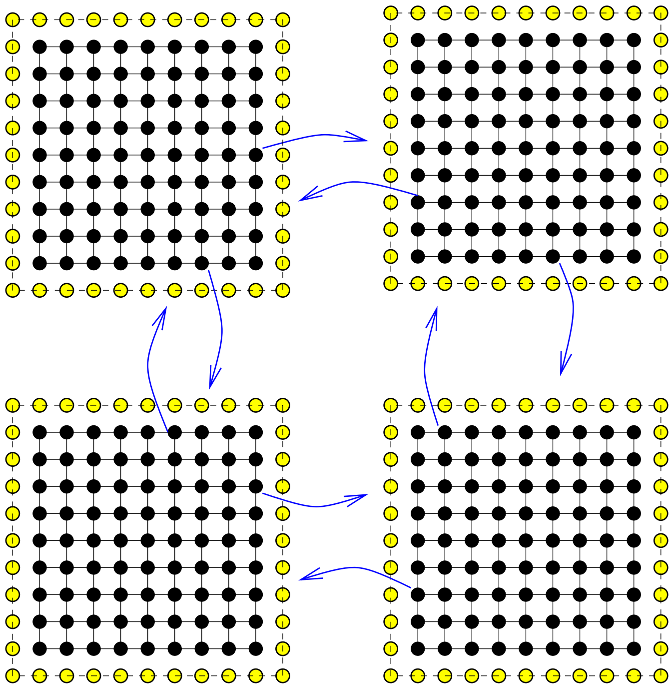


Gitterpartitionierung in der Praxis



Gitterpartitionierung ist eine effiziente Parallelisierungstechnik, falls

- Das gegebene Problem durch lokale Abhängigkeiten gekennzeichnet wird (z.B. in finite Differenzen Diskretisierungen).
- Die Lösungsverfahren parallelisiert werden können und ausreichend "lokal" sind.
- Die Anzahl der Gitterpunkte und Rechenoperationen pro Gitterpunkt etwa gleich ist für alle Blöcke.



Paralleler Speed-Up

- Annahme: P parallele Prozessoren
- Definition von parallelem Speed-Up $S(P)$:

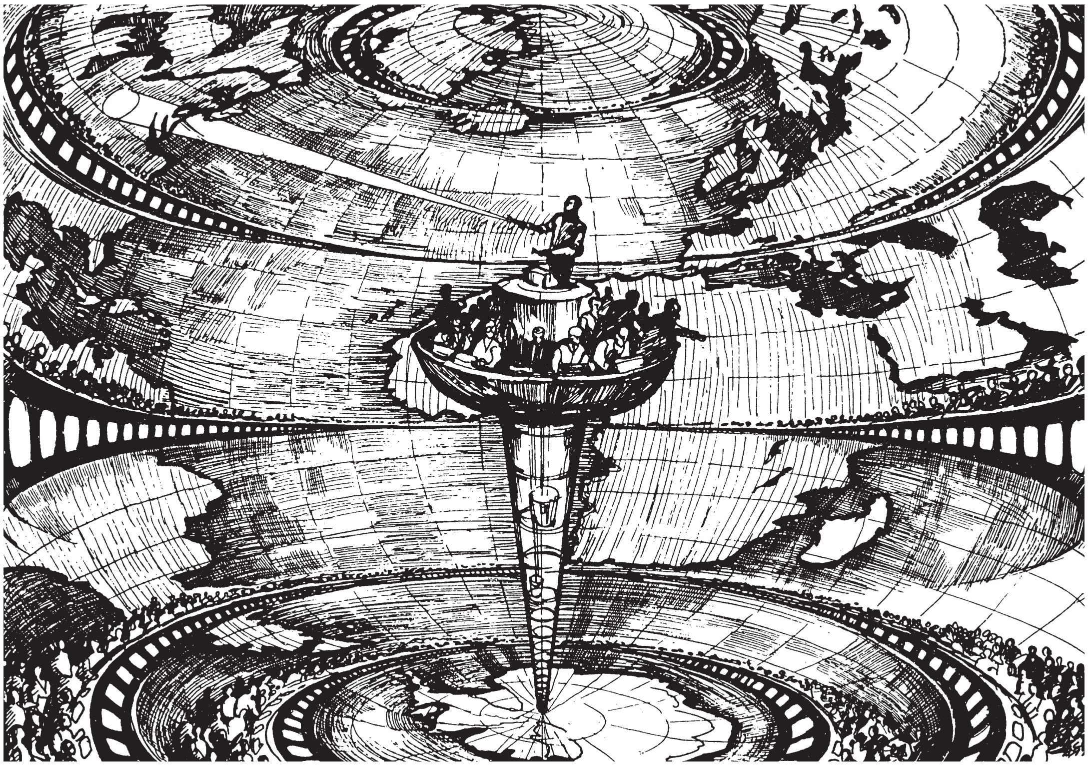
$$S(P) = \frac{T(1)}{T(P)}$$

- Ideal: $S(P) \approx P$
- In der Praxis ist dies aber i.A. für effiziente Lösungsverfahren kaum erreichbar wegen folgender Faktoren:
  - Probleme bei der (möglichst gleichen) Verteilung der Gitterpunkte auf die Prozessoren (Last-Balanzierung)
  - Kosten der Inter-Prozessor-Kommunikation
- Der letzte Punkt ist etwas problematisch im Mehrgitter Verfahren auf den groben Gittern
- Totale Rechenzeit ist aber wichtiger als Speed-Up
- Aber mit P fest, n (# Gitterpunkte) $\to \infty$
  folgt für den Speed-Up $S(P,N) \to P$
  "Boundary-volume-Effect"

$\Rightarrow$ Es ist möglich, effiziente parallele Mehrgitterverfahren für komplexe Anwendungen zu konstruieren

Parallele Rechner

Richardsons
Traum
(1914)



(1999)

die Simulation der Atlantischen Zirkulation

	4/3 Grad Atlantik Modell	1/3 Grad Modell
Auflösung am Äquator	148 km	ca. 27 km
Auflösung in nördlichen Breiten	70 km	13 km
# Gitterpunkte	$78 \times 150 \times 45$	$308 \times 600 \times 45$
# Proz. Cray T3E	15	120
elapsed Zeit (für eine 1 Jahr Simulation)	1.5 Std.	8 Std.

Simulationszeiträume:

für Simulation biologisch-physikalischer Prozesse: 3-5 Jahre

für die Ausbreitung geochemischer Spurenstoffe: 50 Jahre

für die Ausbreitung und Aufnahme von CO2: > 100 Jahre

Bemerkung: Für viele der Anwendungen ist es zweckmäßiger, globale Modelle zu betreiben, der Aufwand an Rechenzeit würde sich etwa vervierfachen.

Weiterhin ist mittelfristig die Anwendung gekoppelter Modelle (Ozean-Atmosphäre-Biospäre-Geospäre) wünschenswert.