

Computerphysik

1. Vorlesung

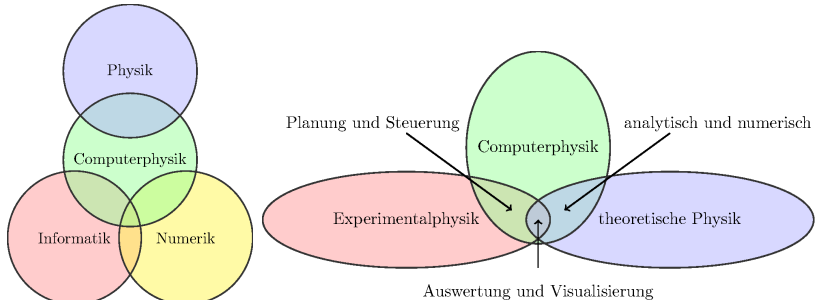
S. Gerlach

WiSe 2025/26

Universität
Konstanz

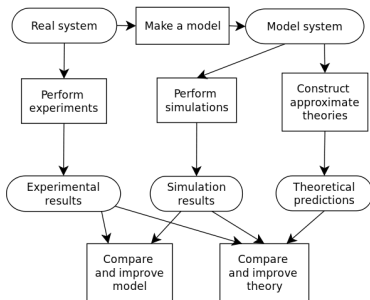


Was ist Computerphysik?



- ▶ Lösen von physikalischen Problemen mit dem Computer
- ▶ Prüfen von Theorien, Experimente planen und steuern
- ▶ Modellbildung + Computersimulation
- ▶ Scientific Computing, Numerik

Was ist Computerphysik?

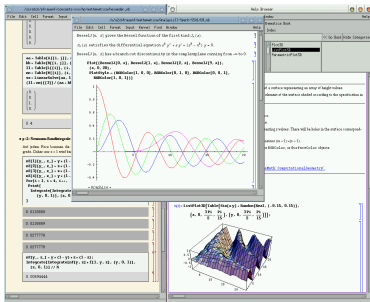


```

#include <stdio.h>

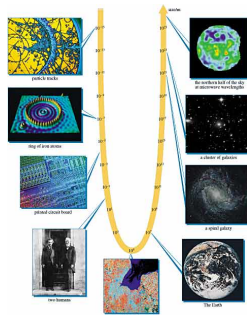
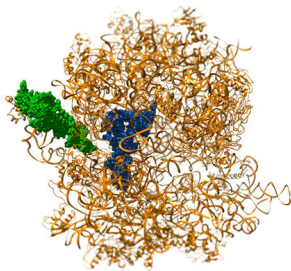
int main()
{
    int i;
    int a[10];
    printf("Enter student's scores: \n");
    for(i = 0; i < 10; i++) {
        scanf("%d", &a[i]);
    }
    printf("Your student's scores are: \n\n");
    for(i = 0; i < 10; i++) {
        printf("%d\n", a[i]);
    }
    return 0;
}

```



Warum Computerphysik?

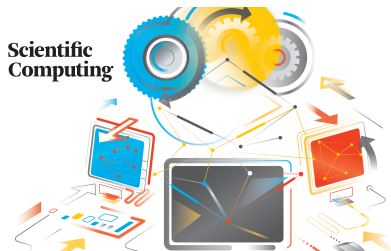
- ▶ Viele physikalische Probleme nicht (oder nur schwer) analytisch lösbar
 - ▶ Vielteilchensysteme, Nichtlineare Dynamik, Realistische Modelle (Anwendungen), ...
- ▶ Computer sind sehr schnell: Laptop ca. 100 GFLOPs (100.000.000.000 Multiplikationen/Sekunde !)
- ▶ Computer können viele Daten speichern (Festplatte ca. 10.000.000.000.000 Byte)



Was ist Scientific Computing (wiss. Rechnen)?

Sammlung von Grundlagen, Techniken und Tools, um wiss. Probleme auf dem Computer zu lösen.

- ▶ Computergrundlagen, Hard- und Software
- ▶ Wiss. Programmieren
- ▶ Gute wiss. Praxis und Open Science
- ▶ Datenkompetenz, Forschungsdatenmanagement (FDM)
- ▶ Umgang mit Daten, Aufbereiten, Analysieren, Präsentieren
- ▶ Numerische Methoden und Anwendung → Computerphysik



Was gehört zu einem Softwareprojekt?

▶ Planung

- ▶ Ziele definieren
- ▶ Aufwand/Resourcen abschätzen

▶ Implementierung

- ▶ Algorithmen+Datenstrukturen überlegen
- ▶ Verwendung von Bibliotheken

▶ Testen (automatisch)

- ▶ Einfache Beispiele
- ▶ Parameterbereiche prüfen
- ▶ Performance

▶ Dokumentation

- ▶ Quellcode, Implementierung
- ▶ Schnittstellen, Formate
- ▶ Benutzung, Daten
- ▶ Beispiele angeben

→ *Software engineering*

Softwareprojekte



How the customer explained it



How the project leader understood it



How the analyst designed it



How the programmer wrote it



How the consultant described it



How the project was documented



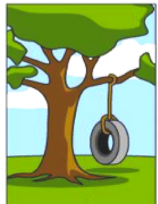
What was installed by operations



How the customer was billed



How the project was supported

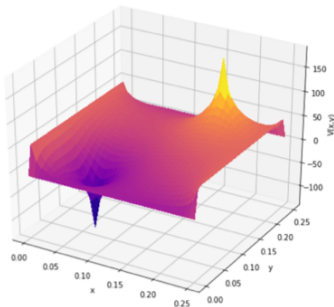


What the customer really needed

Computerphysik-Projekt

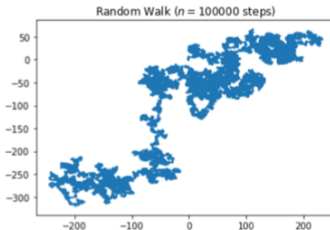
1. Physikalische und Mathematische Grundlagen (Model und Gleichungen)
2. Numerik und Datenstrukturen
3. Implementierung und Tests
4. Auswertung und Dokumentation

Poisson solver



Random walk

```
walk = randomWalk(nsteps)
plt.plot(walk[0],walk[1],label='Random walk')
plt.title("Random Walk ($n = " + str(nsteps) + "$ steps)")
plt.show()
```



Versionsverwaltung

Was ist das?

- ▶ Verwalte alle Versionen von Quelltext, Dokumentation, Daten, etc.
- ▶ Änderungen werden schrittweise übernommen (Commit) bzw. zurückgerollt
- ▶ Koordinierter Zugriff aller Entwickler auf alle Versionen

Wofür?

- ▶ Änderungen nachvollziehen/wiederherstellen (History)
- ▶ Versionsstände speichern (stabil/Entwicklung)
- ▶ Zweige (Branches) anlegen
- ▶ Gemeinsame Entwicklung an einem Projekt
- ▶ CI/CD-Anbindung

