

Physik im Weltall: Computersimulation von Planetenbewegungen

angeboten von der Arbeitsgruppe Rethfeld

Numerische Physik

Warum benötigt man numerische Physik?

- > Viele physikalische Probleme können nicht durch einfach lösbare Formeln beschrieben werden.
- > Oft existiert die Formel, ist aber nicht geeignet, um die Lösung schnell zu bestimmen.

Was ist numerische Physik?

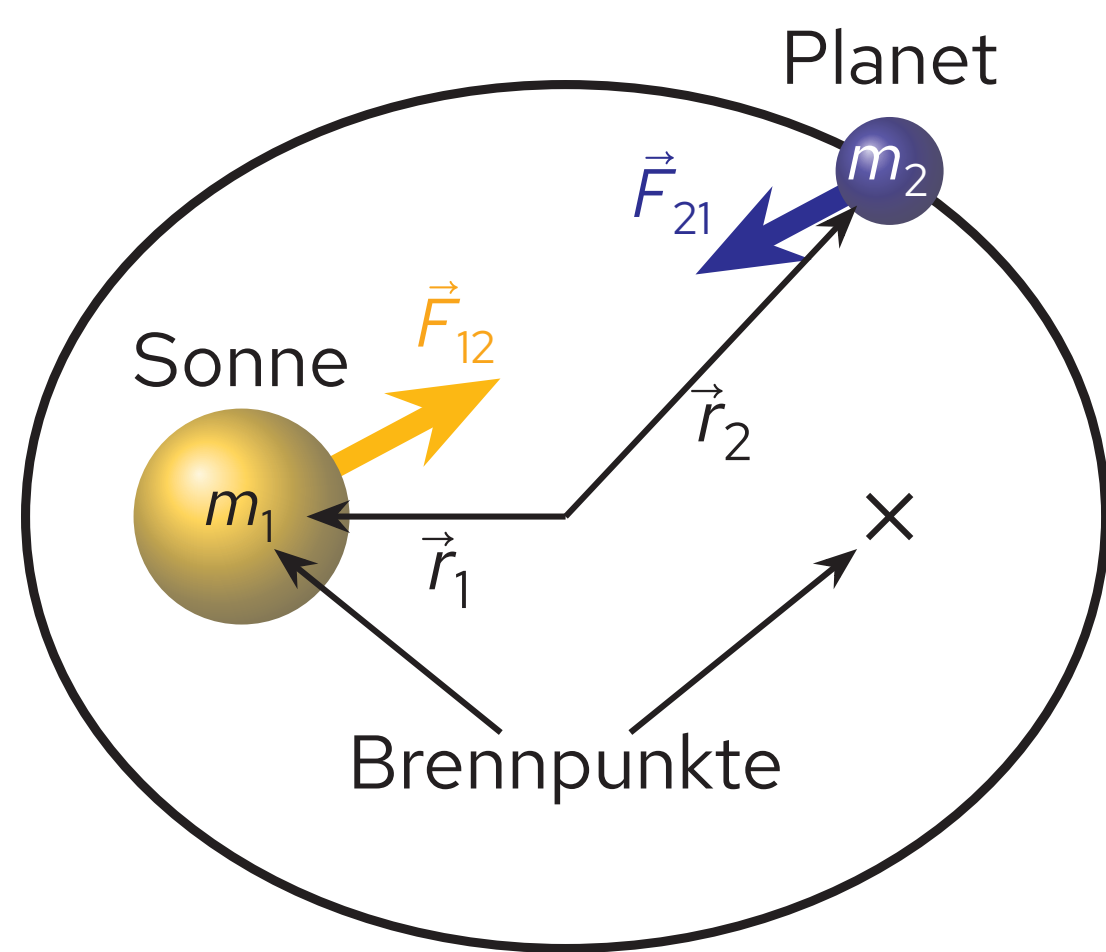
- > Beschreibung von Problemen durch mathematische Gleichungen
- > Näherungsweise Lösung der Gleichungen mit dem Computer
- > Modifikation der Gleichungen unter Berücksichtigung von experimentellen Daten
- > Erkenntnisgewinn aus der Auswertung der computergenerierten Daten

Wo wird numerische Physik verwendet?

- > Berechnung der Bahnen von Himmelskörpern und Raketen
- > Vorhersagen von Wetter und Klimawandel
- > Prognose von Infektionswellen während Epidemien
- > und für noch viel mehr!

Newtonsches Gravitationsgesetz

$$\vec{F}_{12} = G \frac{m_1 m_2}{|\vec{r}_2 - \vec{r}_1|^2} (\vec{r}_2 - \vec{r}_1) \\ = -\vec{F}_{21}$$



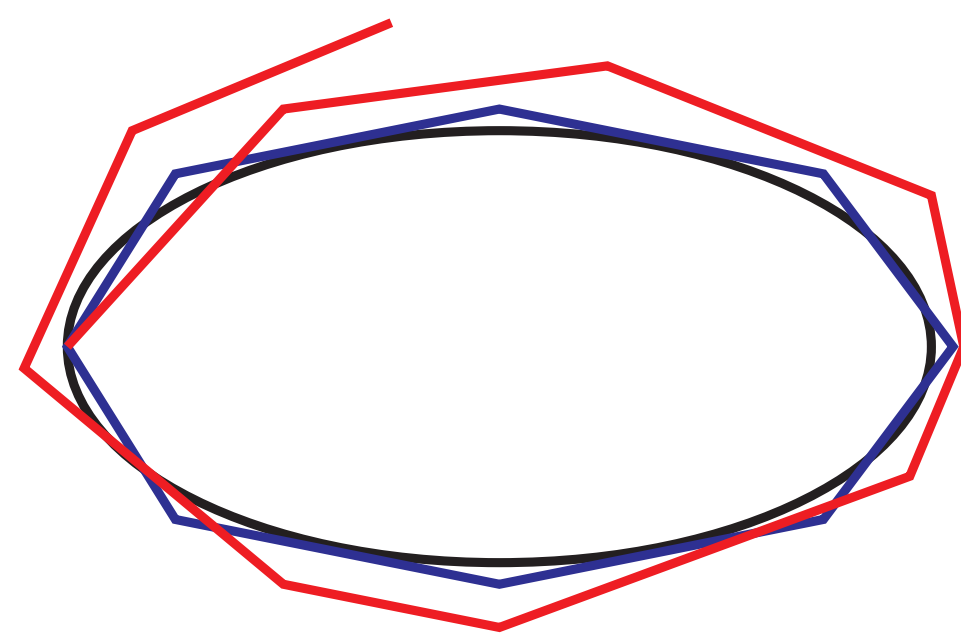
- > Bereits 1687 von Isaac Newton aufgestellt
- > Wirkt auf alle Objekte mit Masse
- > 1. Keplersches Gesetz: Die Planeten bewegen sich auf Ellipsen, in deren einem Brennpunkt die Sonne steht
- > Gravitationskraft eines Objekts von allen anderen Objekten abhängig \Rightarrow andere Planeten "stören" die Bahn eines Planeten um die Sonne
- > Entdeckung des Planeten Neptuns aufgrund von Berechnungen aus Bahnstörungen des Uranus in 1846

Euler Verfahren

- $$t_{\text{neu}} = t_{\text{alt}} + \Delta t, \quad \vec{a}_1(t_{\text{alt}}) = \frac{\vec{F}_{12}(t_{\text{alt}}) + \vec{F}_{13}(t_{\text{alt}}) + \dots}{m_1}$$
- $$\Rightarrow \vec{v}_1(t_{\text{neu}}) = \vec{a}_1(t_{\text{alt}}) \cdot \Delta t + \vec{v}_1(t), \quad \vec{r}_1(t_{\text{neu}}) = \vec{v}_1(t_{\text{alt}}) \cdot \Delta t + \vec{r}_1(t)$$
- > Erhöhe Zeit schrittweise um Δt .
 - > Berechne Ort und Geschwindigkeit aller Objekte zur Zeit $t + \Delta t$ aus den bereits berechneten Werten.

Der Zeitschritt

- > ist eine der wichtigsten Größen in der numerischen Beschreibung dynamischer Prozesse.
- > legt fest, wie oft Ort und Geschwindigkeit neu berechnet werden.
- > bestimmt Genauigkeit der Berechnung:
 - > kleiner Zeitschritt: gute Auflösung, lange Rechendauer
 - > grosser Zeitschritt: schlechte Auflösung, kurze Rechendauer
 - > Komplexere Lösungsverfahren erlauben größere Zeitschritte bei akzeptabler Auflösung.



Schwarz: exakte Bahn
Blau: passabler Zeitschritt
Rot: zu großer Zeitschritt

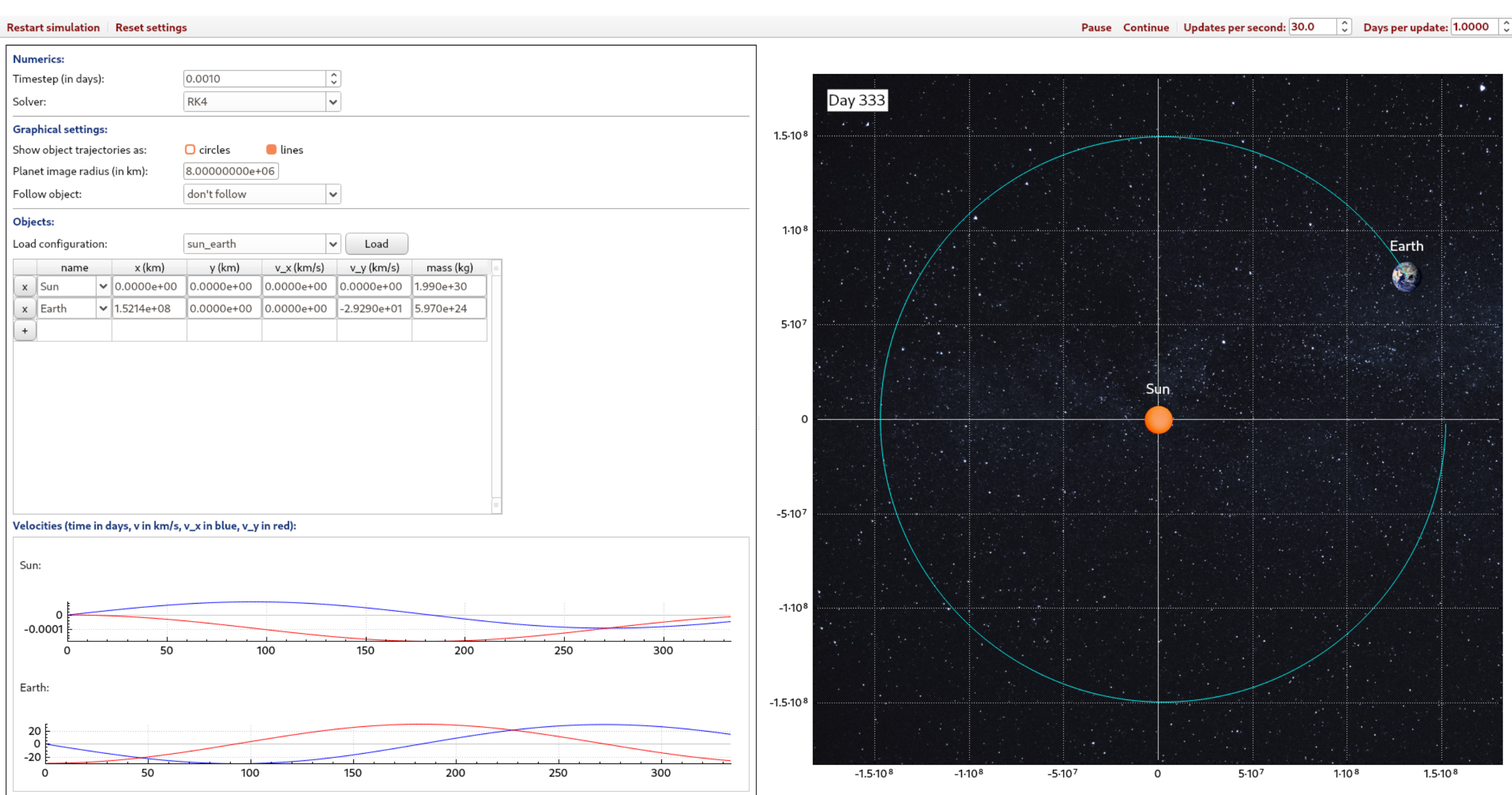
Unser Programm

- > Programmcode kann von Menschen gelesen werden.
- > Computer können ihn verstehen und ausführen.
- > Unser Programm besteht aus zwei Teilen: numerischer Berechnung und grafischer Oberfläche.

Programmcode (Auszug):

```
array<double, 2> getForce(const Object& o1, const Object& o2)
{
    double x_diff = o2.getX() - o1.getX();
    double y_diff = o2.getY() - o1.getY();
    double r_squared = x_diff * x_diff + y_diff * y_diff;
    if(r_squared == 0.)
    {
        return {0, 0};
    }
    double r = std::sqrt(r_squared);
    double force = G * o1.getMass() * o2.getMass() / r_squared;
    return {force * x_diff / r, force * y_diff / r};
}
```

Grafische Oberfläche:



Kontakt:
Arbeitsgruppe für Ultrakurzzeitdynamik
laserangeregter Festkörper unter Leitung von
Prof. Bärbel Rethfeld (rethfeld@rptu.de)

