

Übung 05: Spektrum einer Klaviersaite

Tobias Blesgen und Leonardo Thome

27.06.2021

Schlägt man eine Klaviersaite an, so verhält sie sich nach:

$$\frac{\partial^2 \psi(x, t)}{\partial t^2} = c^2 \frac{\partial^2 \psi(x, t)}{\partial x^2} - \frac{\gamma}{l} \left| \frac{\partial \psi(x, t)}{\partial t} \right| \left(\frac{\partial \psi(x, t)}{\partial t} \right). \quad (1)$$

Wir wollen dieses Verhalten im Folgenden numerisch untersuchen. Hierzu schreiben wir Gleichung 1 um zu:

$$\frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{\Delta t^2} = c^2 \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{\Delta x^2} - \frac{\gamma}{l} \left| \frac{\psi_{i,j} - \psi_{i-1,j}}{\Delta t} \right| \frac{\psi_{i,j} - \psi_{i-1,j}}{\Delta t} \quad (2)$$

Dies lässt sich umschreiben zu:

$$\psi_{i+1,j} = \frac{\Delta t^2 c^2}{\Delta x^2} (\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}) + 2\psi_{i,j} - \psi_{i-1,j} - \frac{\gamma}{l} |\psi_{i,j} - \psi_{i-1,j}| (\psi_{i,j} - \psi_{i-1,j}) \quad (3)$$

Runge-Kutta 2 Verfahren

merische Stabilität (soweit das System selbst stabil ist) und weist mit einem Verfahrensfehler von $\mathcal{O}(h^2)$ einen kleineren Fehler als das Eulerverfahren auf für kleine Schrittweiten h .

Implementation des DGS nach dem Runge-Kutta 2 Verfahren

```
#include <Rcpp.h>
#include <vector>
#include <algorithm>
#include <math.h>

using namespace Rcpp;

// [[Rcpp::export]]
std::vector<double> klaviersaite(const double l, const double roh,
                                const double force, const double
                                gamma, const int xSchritte,
                                const double dt, const int
                                zeitSchritte){

    // Array der Werte zur späteren Ausgabe
    std::vector<std::vector<double>> zeitvektor(zeitSchritte);
```

```

// Quelltext
// Startwerte
std::vector<double> startwerte(xSchritte);
for (int i = 0; i<xSchritte; i++){
    startwerte[i] = 0.0;
}
startwerte[(int)(0.26*(xSchritte-1))] = 0.01;

zeitvektor[0] = startwerte;
zeitvektor[1] = startwerte;

// Funktionsdurchläufe
double dx = 1/(xSchritte - 1);
double c = 299792458;
double C = dt*dt*c*c/(dx*dx);

for (int i = 2; i<zeitSchritte; i++){
    std::vector<double> tVektor(xSchritte);
    // Randbedingungen
    tVektor[0] = 0;
    tVektor[xSchritte-1] = 0;
    for (int j = 1; j<xSchritte-1; j++){
        tVektor[j] = C*(zeitvektor[i-1][j+1] - 2*zeitvektor[i-1][j] +
            zeitvektor[i-1][j-1]) + 2*zeitvektor[i-1][j] - zeitvektor[i-2][j] -
            gamma/1 * fabs(zeitvektor[i-1][j]-zeitvektor[i-2][j]) *
            (zeitvektor[i-1][j]-zeitvektor[i-2][j]);
    }
    zeitvektor[i] = tVektor;
}

// Rückgabe für eine grafische Wiedergabe
return zeitvektor[9];
}

```

Fazit

Wir erhalten bei den meisten Werten ein nicht chaotisches Verhalten zwischen θ und $\dot{\theta}$, können jedoch auch chaotische Situationen konstruieren. Dies ist im physikalischen Sinne auch nachvollziehbar, da die Eigenrotation des Mondes durch geschickt gewählte Bahnen stark von der Radiusvariation beeinflusst werden kann.

Wir können aber sehen, dass dieses Verhalten nicht immer eintritt und wir im Allgemeinen ein strukturiertes Verhalten vorfinden.

Literatur

- [1] Keplersche Gesetze https://de.wikipedia.org/wiki/Keplersche_Gesetze, Stand 22.06.2021

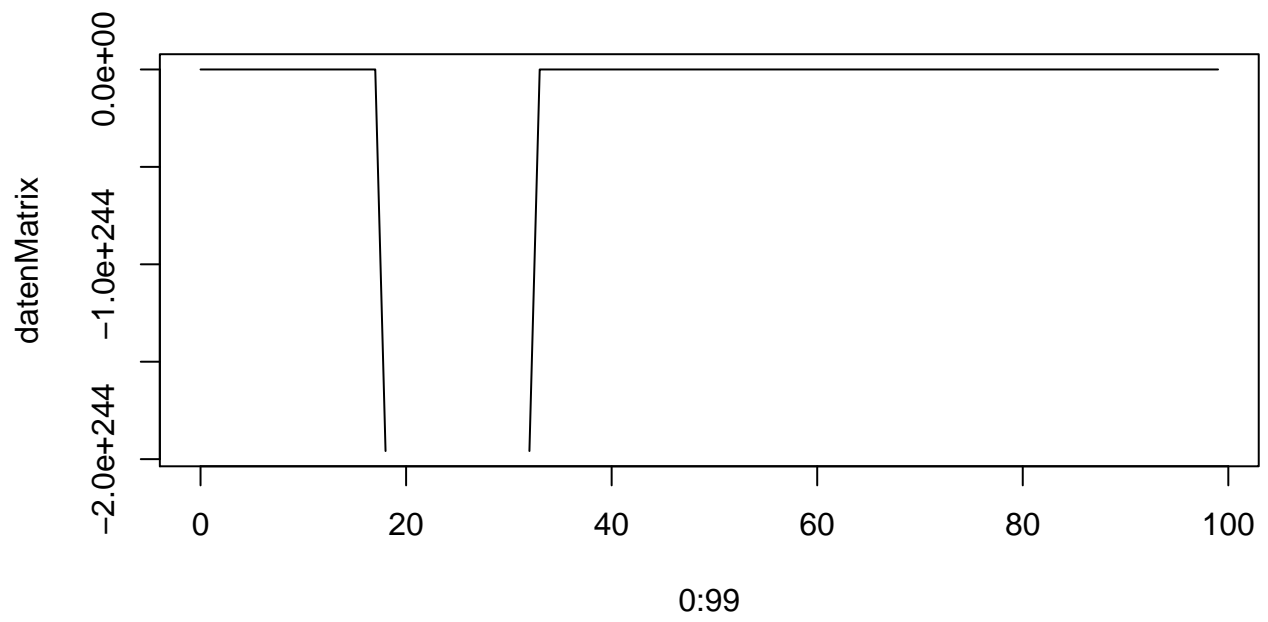


Abbildung 1: k)