

11. naloga – Gibanje neraztegljive vrvice

Jože Zobec

1 Razmislek

V tej nalogi bomo simulirali gibanje neraztegljive vrvice. Imamo dve enačbi in dve neznanki:

$$F \left(\frac{\partial \varphi}{\partial s} \right)^2 - \frac{\partial^2 F}{\partial s^2} = \left(\frac{\partial \varphi}{\partial t} \right)^2 \quad (1)$$

$$F \frac{\partial^2 \varphi}{\partial s^2} + 2 \frac{\partial F}{\partial s} \frac{\partial \varphi}{\partial s} = \frac{\partial^2 \varphi}{\partial t^2} \quad (2)$$

Imamo začetni pogoj $\varphi(s, t = 0) = \varphi_0(s) = \text{konst.}$ V prijemališču vrvice velja robni pogoj

$$F \frac{\partial \varphi}{\partial s} + \cos \varphi = 0, \quad \frac{\partial F}{\partial s} + \sin \varphi = 0,$$

na odprtem koncu pa je

$$F(s = 1) = 0, \quad \frac{\partial^2 \varphi}{\partial s^2} = 0.$$

Da lahko rešimo prvo enačbo potrebujemo še $\dot{\varphi}(s, t = 0)$. Tu uporabimo namig, da vrvica na začetku miruje, tj. $\dot{\varphi}(s, t = 0) = 0$. Sedaj poznamo vse.

Nalogo bomo reševali z diskretnimi približki – vrvico aproksimiramo z odsekoma ravno krivuljo, kjer so vozlišča razmaknjena za $h = 1/N$.

Diferenčne enačbe so reda $\mathcal{O}(h)$ in se glasijo:

$$F_{i-1}^t + F_{i+1}^t - F_i^t \left[2 + \left(\frac{\varphi_{i+1}^t - \varphi_{i-1}^t}{2} \right)^2 \right] = -\frac{h^2}{h_t^2} (\varphi_i^t - \varphi_i^{t-1})^2, \quad (3)$$

$$\varphi_i^{t+1} = \frac{h_t^2}{h^2} \left[\frac{1}{2} (\varphi_{i+1}^t - \varphi_{i-1}^t) (F_{i+1}^t - F_{i-1}^t) + F_i^t (\varphi_{i+1}^t + \varphi_{i-1}^t - 2\varphi_i^t) \right] + 2\varphi_i^t - \varphi_i^{t-1}, \quad (4)$$

na koncéh za silo velja

$$F_1^t = h \sin \varphi_1^t - F_2^t, \quad (5)$$

$$F_N^t = 0, \quad (6)$$

kar moramo v enačbi (3) upoštevati kot posebno enačbo za F_2 :

$$F_3^t - F_2^t \left[1 + \frac{1}{4} (\varphi_3^t - \varphi_1^t) \right] = -\left(\frac{h}{h_t} \right)^2 (\varphi_2^t - \varphi_2^{t-1}) - h \sin \varphi_1^t. \quad (7)$$

Robni pogoji za kot φ so

$$\varphi_1^t = \frac{2h}{F_2^{t-1}} \cos \varphi_2^t + \varphi_3^t \quad (8)$$

$$\varphi_N^t = 2\varphi_{N-1}^t - \varphi_{N-2}^t \quad (9)$$

Poznamo $\varphi_i^0 = \varphi_i^1 = \varphi_0$, od koder lahko izračunamo vse ostalo.

1.1 Algoritem

Najprej iz $\varphi(s, t = 1)$ izračunamo $F^1(s)$ prek enačb (3) (z robnim pogojem (7)) za $i = 2, \dots, N - 1$. F_1^1 in F_N^1 določimo iz robnih pogojev (5) in (6). Sedaj, ko imamo začetna F_i^t in φ_i^t izvajamo algoritem:

1. Prek enačbe (4) izračunamo φ_i^{t+1} za $i = 2, \dots, N - 1$.
2. Rešimo enačbo (8) in izračunamo φ_1^{t+1} , iz enačbe (9) pa izračunamo φ_N^{t+1} .
3. Rešimo tridiagonalni sistem enačb (3) (in (7) za F_2^{t+1}) in izračunamo F_i^{t+1} za $i = 2, \dots, N - 1$.
4. Izračunamo F_1^{t+1} in F_N^{t+1} po enačbah (5) in (6).
5. Premaknemo časovni indeks $t \rightarrow t + 1$ in se vrnemo na točko 1.

Rezultati so močno odvisni od tega, če smo pravilno upoštevali predznake, kar mi je povzročilo nemalo preglavic.

2 Rezultati

Grobo šablono programa sem najprej napravil v `Octave`, ki pa se je izvajala prepočasi, zato sem napravil za hitrost opzimirano varianto v programskem jeziku `C`, pri čemer sem si dodatno pomagal s programom za profilno vodeno optimizacijo, `gprof`. Kot ponavadi, sem tudi tokrat žrtvoval nekaj natančnosti na račun hitrosti in vklopil `-ffast-math`. Prevajalnik `gcc-4.7` pri tem ne daje več konsistentnih rezultatov, `clang-3.3` pa je še vedno stabilen pravilno zaokrožuje, zato sem uporabil slednjega.

Ugotovil sem, da v moji implementaciji največ klicev dobimo za izračun diagonale našega tridiagonalnega sistema, časovno najpotratnejši proces pa je izračun φ^{t+1} .

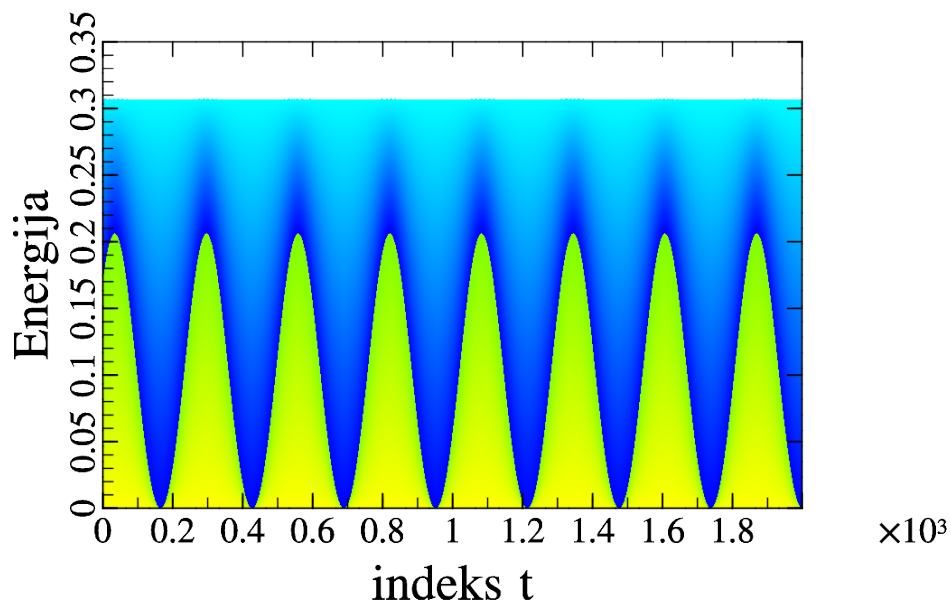
Rezultate sem za različne začetne pogoje prikazal na priloženih animacijah, barva vzdolž verige prikazuje silo $F(s, t)$ med členki, vektorji pa prikazujejo hitrost in so dolžine $2\sqrt{v_x^2 + v_y^2}$. Izris sem napravil s knjižnico `MathGL`, animacijo pa s programom `mencoder`.

- `ravna.avi` upošteva začetne pogoje, kot jih navaja navodililo, veriga je videti nekam toga. Začetni kot sem nastavil na $10\pi/3$.
- `opletanje.avi`, tu sem za začetne pogoje vzel $\varphi_i^1 = \sin(i\pi/2N)$, za $\varphi_i^0 = \varphi_i^1 + h_t i/N$. Opazoval sem opletanje verige. Ugotovil sem, da model vozlov na verigi ne opiše pravilno – v veliko primerih se takrat veriga strga (tj. $F \rightarrow \infty$), kar pa, kot vemo iz lastnih izkušenj, se ne zgodi prav pogosto. Zelo lepo je na začetku videti povratni udarec verige („backlash“), vendar pa zaradi vozlov model na koncu dinamike ne opiše realistično.
- `valovanje.avi` prikazuje kako lepo se valovi širijo vzdolž verige, vendar pa eden izmed vozlov verigo spet na koncu strga. φ_i^1 je isti kot za datoteko `opletanje.avi`, vendar pa je $\varphi_i^0 = \varphi_i^1 + h_t \sin(i\pi/N)$.
- `cela.avi` prikazuje $\varphi_i^0 = \varphi_i^1 = \frac{\pi}{10}i/N$. Ta začetni pogoj je verigo obdržal celo dlje, zato sem to datoteko kar poimenoval `cela.avi`.

Animacije so narejene pri $h_t = 10^{-5}$, $N = 500$, zlepil sem po 34 sličic na sekundo. Med izrisom dveh zaporednih sličic, sem jih 10^5 izpustil, razen za datoteke `opletanje.avi`, `valovenje.avi` in `cela.avi`, kjer sem jih izpustil po 1000.

Ker imamo vse hitrosti in pozicije, lahko iz tega izračunamo tudi energijo. Ohranjanje energije nam bo pokazatelj natančnosti metode.

Energija je vsota kintetične in potencialne: $E_{\text{tot.}} = T + V = T_{\text{trans.}} + T_{\text{rot.}} + V$. Grag 1 prikazuje ohranjanje energije.



Slika 1: Ohranjanje energije za animacijo `ravna.avi`. Zeleni del prikazuje kinetično energijo, modri del pa potencialno, pri kateri sem dodal $h_0 = 0.6$, da bi zagotovil $V \geq 0$. Indeks ' t ' je časovni indeks, ki je bil inkrementiran po vsakih h_t^{-1} (tj. 10^5) korakih. Energija se ohranja na tri decimalna mesta.