

Fakultet elektrotehnike i računarstva

Računalna grafika

Ak. god. 2020./2021.

Luka Mesarić, 0036505985

Treća laboratorijska vježba - projekt

ODE Explorer

Zagreb, 22. siječnja 2021.

Sadržaj

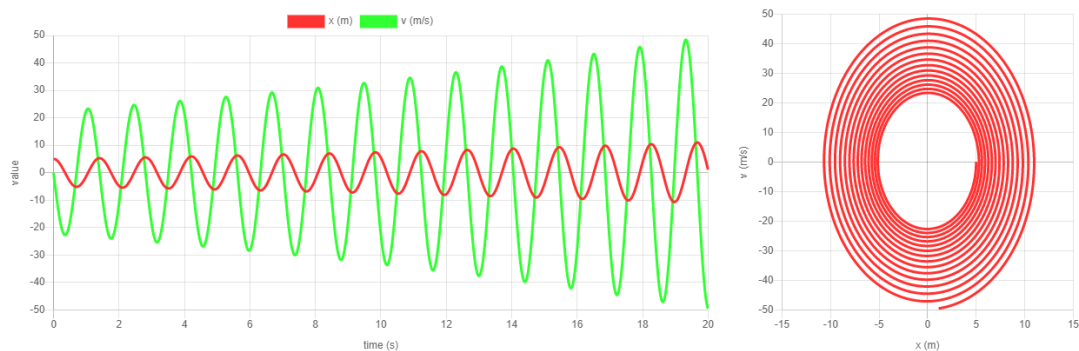
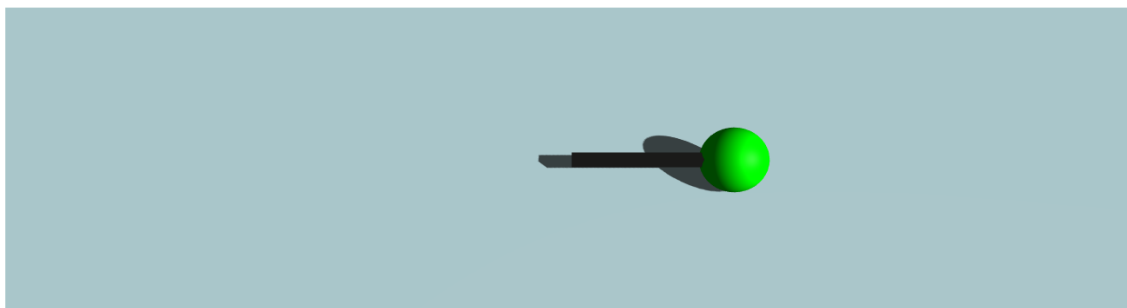
1	Uvod	2
2	Korištene tehnologije	3
3	Postupci numeričke integracije	4
4	Mogućnosti aplikacije	5
5	Upute za pokretanje	6

1. Uvod

U ovom projektu napravljena je vizualizacija različitih postupaka numeričke integracije u realnom vremenu. Vizualizacija se radi na primjeru jednostavnog sustava mase na jednoj opruzi bez trenja. Projekt je izveden kao *JavaScript* web aplikacija koja se u potpunosti izvodi na klijentu. Aplikacija bi trebala zamijeniti *Java applet* koji se za demonstrativne potrebe trenutno koristi na diplomskom kolegiju Računalna grafika.

Forward Euler Total time (s): dt (s): k (N/m): m (kg): x0 (m): Time warp:

Total energy (J): 12379.412201872685



ODE Explorer - Copyright (c) 2021 Luka Mesarić - MIT License

Slika 1.1: Izgled aplikacije nakon izvršavanja uz zadane parametre

2. Korištene tehnologije

Aplikacija je napisana koristeći HTML5 i *JavaScript*. Može se pokrenuti u svim modernim web-preglednicima, uključivo i na mobilnim uređajima. Od biblioteka korištene su Math.js¹, Chart.js² i Three.js³. Sve navedene automatski se preuzimaju s CDN-ova (engl. *Content delivery network*) prilikom pokretanja aplikacije.

Math.js korišten je za sve operacije s matricama i vektorima koje su bile potrebne za implementiranje numeričkih izračuna.

Chart.js korišten je za dinamičko prikazivanje stanja varijabli sustava, primarno trenutne pozicije i brzine.

Three.js korišten je za 3D prikaz kuglice i opruge te njihovo animiranje, kao i slobodno pomicanje očista korištenjem miša ili ekrana osjetljivog na dodir. Ako je moguće, Three.js automatski će izračune izvršavati na grafičkoj kartici čime se oni značajno ubrzavaju. Za to se u pozadini koristi WebGL API na što programer ne mora obraćati posebnu pažnju.

¹<https://mathjs.org>

²<https://chartjs.org>

³<https://threejs.org>

3. Postupci numeričke integracije

Sustav opruge modeliran je pomoću Hookeov zakona, $F = -k\Delta x$. Nakon jednostavnog izračuna dobiva se linearni sustav jednažbi koji povezuje varijable i njihove derivacije. U matricnom zapisu to je $\vec{x}' = A\vec{x}$, pri čemu vrijedi $\omega^2 = k/m$ i

$$A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix}$$

Prvi element vektora \vec{x} predstavlja poziciju, a drugi brzinu mase na kraju opruge.

U sklopu projekta implementirano je šest postupaka numeričke integracije. Svi postupci su jednokoračni. Oznaka T predstavlja korak integracije.

Eksplisitni (izravni) Eulerov postupak (engl. *forward Euler method*) za rješavanje sustava koristi jednažbu $x_{k+1} = x_k + T \cdot \vec{x}'_k$. Kroz iteracije nakuplja nezanemarlivu grešku povećanjem ukupne energije sustava. Povećavaju se i amplituda i brzina.

Implicitni (obrnuti) Eulerov postupak (engl. *backward Euler method*) za rješavanje koristi izraz $x_{k+1} = x_k + T \cdot \vec{x}'_{k+1}$. Vektor \vec{x}_{k+1} pojavljuje se s obje strane jednažbe pa ju je potrebno preurediti prije korištenja. Kroz iteracije nakuplja nezanemarlivu grešku smanjenjem ukupne energije sustava. Smanjuju se i amplituda i brzina.

Polu-implicitni Eulerov postupak (engl. *semi-implicit Euler method*, *symplectic Euler*) za određivanje v_{k+1} koristi x_k , a x_{k+1} računa preko v_{k+1} čime se poboljšava stabilnost uz zadržavanje jednostavnosti računanja. Matricni sustav izgleda ovako:

$$\begin{bmatrix} x_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 - \omega^2 T^2 & T \\ -\omega^2 T & 1 \end{bmatrix} \cdot \begin{bmatrix} x_k \\ v_k \end{bmatrix}$$

Trapezni postupak (engl. *trapezoidal rule*) implicitni je postupak čija je formula $x_{k+1} = x_k + \frac{T}{2}(\vec{x}'_k + \vec{x}'_{k+1})$. Kao i kod implicitnog Eulerovog postupka, potrebno je preurediti jednažbu. Očuvanje energije u sustavu gotovo je savršeno za svaki korak integracije.

Heunova metoda (engl. *Heun's method*) prediktorsko-korektorski je par eksplisitne Eulerove metode i trapeznog postupka. Stabilnija je od izravnog Eulerovog postupka, ali pri većim koracima integracije ima jednake nedostatke.

Runge-Kutta metoda 4. reda eksplisitni je postupak s najsloženijim izračunom od ovdje navedenih. U pravilu je vrlo stabilna, ali za veće korake integracije ima iste nedostatke kao implicitni Eulerov postupak.

4. Mogućnosti aplikacije

U ovom poglavlju ukratko će biti pobrojane razne manje mogućnosti aplikacije.

Konfigurabilni parametri simulacije su numerički integrator, trajanje simulacije, korak integracije, konstanta opruge, masa kuglice, početna pozicija i faktor ubrzanja prikaza simulacije.

Tijekom izvođenja simulacije uživo se prikazuje ukupna energija sustava izračunata formulom $E = \frac{1}{2}kx^2 + \frac{1}{2}mv^2$.

Na jednom grafu tijekom simulacije prikazuju se pozicija i brzina u ovisnosti o vremenu. Na drugom grafu prikazuje se brzina u ovisnosti o poziciji kako bi numeričke greške bile uočljivije.

Osim klikom na gumb *Run*, simulacija se može pokrenuti tipkama Enter ili R. Omogućeno je pauziranje i nastavljavanje izvođenja klikom na gumb *Pause/Continue* ili tipke P i C. Izvođenje se može pokrenuti s drugim parametrima bez čekanja da se prethodna simulacija dovrši.

Mijenjanje parametra x_0 uživo mijenja poziciju kuglice, osim kada je izvođenje pauzirano.

Promjenom veličine prozora automatski se mijenja veličina 3D prikaza i eventualno se mijenja jesu li grafovi poredani vertikalno ili horizontalno, što je korisno za prikaz na mobilnim uređajima.

Kamera (očiste) u 3D prikazu može se pomicati korištenjem miša – lijevi klik za rotaciju, desni za translaciju i *scroll* za zumiranje. Pomicanje se može ostvariti i koristeći ekran osjetljiv na dodir.

Sjene kuglice i opruge vidljive su na podlozi.

5. Upute za pokretanje

Programski kôd u cijelosti je javno dostupan na *GitHub* repozitoriju:
<https://github.com/LMesaric/ODE-Explorer>.

Aplikaciji se može pristupiti preko sljedeće poveznice:
<https://lmesaric.github.io/ODE-Explorer>.

Pokretanje bez servera nije moguće u svim preglednicima. Jedna od mogućnosti korištenje je lokalnog servera poput *Live Server* dodatka za *Visual Studio Code*¹. Aplikacija se pokreće otvaranjem `index.html` datoteke u web-pregledniku, postavljanjem parametara simulacije i pritiskom na gumb *Run*.

¹<https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>