

Symulacja flagi metodami Rungego-Kutty

Jan Wojtecki

Model flagi został wykonany jako układ masa-sprężyna-tłumik, sieć równomiernie rozłożonych punktów (mas) połączonych sprężynami, mającymi reprezentować strukturę i naprężenia materiału, oraz jego skłonność do zginania się.

Równania ruchu punktów są rozwiązywane metodą Rungego-Kutty, odpowiednio 2. rzędu (metoda znana również jako midpoint - punktu pośredniego, lub zmodyfikowana Eulera) i 4. rzędu, według następującego algorytmu:

Oblicz F_{wyp} dla v_0 i x_0 : $k_{1v} = \Delta t \frac{F_{wyp}}{m}$, $k_{1x} = \Delta t v_0$

Oblicz F_{wyp} dla $v_0 + \frac{1}{2}k_{1v}$ i $x_0 + \frac{1}{2}k_{1x}$: $k_{2v} = \Delta t \frac{F_{wyp}}{m}$, $k_{2x} = \Delta t \frac{1}{2}k_{1v}$

Wynik dla RK2: $v_1 = v_0 + k_{2v}$, x analogicznie

Oblicz F_{wyp} dla $v_0 + \frac{1}{2}k_{2v}$ i $x_0 + \frac{1}{2}k_{2x}$: $k_{3v} = \Delta t \frac{F_{wyp}}{m}$, $k_{3x} = \Delta t \frac{1}{2}k_{2v}$

Oblicz F_{wyp} dla $v_0 + k_{3v}$ i $x_0 + k_{3x}$: $k_{4v} = \Delta t \frac{F_{wyp}}{m}$, $k_{4x} = \Delta t k_{3v}$

Wynik dla RK4: $v_1 = v_0 + \frac{1}{6}(k_{1v} + 2k_{2v} + 2k_{3v} + k_{4v})$, x analogicznie

Na F_{wyp} składają się siły:

- Grawitacji $\vec{F}_G = m\vec{g}$
- Sprężystości $\vec{F}_S = -k\Delta\vec{x}$ od każdego sąsiada w strukturze, z k odpowiednim dla rodzaju więzów
 - Więzy strukturalne, do 4 sąsiadów, prostopadłe do kratki
 - Więzy ścinania, do 4 sąsiadów, po skosie
 - Więzy zginania, do 8 sąsiadów, prostopadłe i po skosie - o dwa dalej
- Oporu powietrza $\vec{F}_D = -c\vec{v}^2$, gdzie c to pewna empirycznie wyznaczona stała - układ dobrze spisuje się także dla $c = 0$
- Wiatru $\vec{F}_W = \frac{1}{2N} \sum_N \vec{F}_{WS}$
 - $\vec{F}_{WS} = \vec{n}(\hat{n} \bullet \vec{F}_{WE})$ to składowa siły wiatru, napór na trójkąty materiału, jakie punkt tworzy z sąsiadami strukturalnymi
 - \vec{F}_{WE} to wektor siły wiatru z otaczającego środowiska (uproszczenie obliczeń prędkości/ciśnienia)
 - $\vec{n} = \vec{a} \times \vec{b}$ to wektor normalny do trójkąta, gdzie \vec{a} i \vec{b} to wektory odległości do sąsiadów
 - \hat{n} to znormalizowany/jednostkowy wektor \vec{n}
 - Wynikowy wektor to średnia wektorów składowych podzielona przez 2, ponieważ gdy liczymy dla wszystkich punktów, każda powierzchnia jest liczona podwójnie.

Symulacja odbywa się w czasie rzeczywistym. Program został wykonany w C#, w silniku Unity. Kod projektu można znaleźć w repozytorium:
https://gitlab.com/_j4/flag-simulation.

Literatura

- [1] Matthew Fisher. Cloth. URL: <https://graphics.stanford.edu/~mdfisher/cloth.html>.
- [2] Jacek Matulewski. Metody runge-kutty w praktyce. URL: <https://www.fizyka.umk.pl/~jacek/dydaktyka/modsym/notatki/programista/ODE2.pdf>.
- [3] Jacek Matulewski. Podstawy mechaniki klasycznej dla programistów gier, czyli rzecz o tym, jak całkować równanie ruchu. URL: <https://www.fizyka.umk.pl/~jacek/dydaktyka/modsym/notatki/programista/ODE1.pdf>.
- [4] Jesper Mosegaard. Mosegaards cloth simulation coding tutorial. URL: <https://viscomp.alexandra.dk/?p=147>.
- [5] Adam Sosnowski. Implementacja i porównanie metod numerycznych stosowanych do realistycznych symulacji dynamiki zbioru punktów materialnych. wizualizacja z użyciem grafiki 3d. URL: https://www.fizyka.umk.pl/~jacek/dydaktyka/gfn/ASosnowski_inz.pdf.