Heinrich–Heine–Universität Düsseldorf
Institut für Theoretische Physik II
Computational Physics
Wintersemester 2018/2019

Prof. Dr. J. Horbach
Dr. S. Ganguly
(saswati@thphy.uni-duesseldorf.de.de)
M. Golkia (mehrdad.golkia@hhu.de)
Blatt 2 vom 23.10.2018
Abgabe bis 12:30 Uhr am 30.10.2018

Problem 2.1: **Euler Algorithm**

Consider a particle of mass m that is attached to a spring with spring constant k and that performs small oscillations in one dimension. The displacement of the particle from its position at rest is denoted by x and its velocity by v. Thus, the equations of motion can be written as

$$\frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{k}{m}x \qquad \frac{\mathrm{d}x}{\mathrm{d}t} = v. \tag{1}$$

In the following, we set $k = 1$ and $m = 1$ and choose the initial conditions $v(0) = v_0$ and $x(0) = x_0$.

a) Solve Eq. (1) numerically for $v_0 = 0$ and $x_0 = 1$ using the Euler algorithm with time step $\delta t = 0.1$ up to a time $t_{\max} = 50$,

$$\begin{aligned} x_{n+1} &= x_n + v_n \delta t \\ v_{n+1} &= v_n - x_n \delta t. \end{aligned} \tag{2}$$

Plot $x(t)$ and $v(t)$ as a function of t and compare these results to the analytical solutions. What do you observe?

b) Compute and plot the total energy E as function of time t. Interpret the result.

Programming Hint: Write a function `euler()` that implements the above relation (2). As functions have only one return value, you may want to use the following scheme to make the "input" arguments to this function also "output" arguments, by defining the arguments adding an ampersand (&) to the type, as follows:

```
void euler(double& x, double& v, double dt) {
  x = 1;
  v = 2;
  dt = 1;
}

int main() {
  double x = 3;
  double v = 4;
  double dt = 0.1
  euler(x, v, dt);

  // will output "1 2 0.1"
  cout << x << " " << v << " " << dt << endl;
}
```

Problem 2.2: **Euler-Cromer algorithm**

a) Solve Eqs. (1) numerically for $v_0 = 0$ and $x_0 = 1$ using the Euler-Cromer algorithm with a time step $\delta t = 0.2$:

$$\begin{aligned} x_{n+1} &= x_n + v_n \delta t \\ v_{n+1} &= v_n - x_{n+1} \delta t. \end{aligned} \tag{3}$$

and compare with the results of problem 2.1 (Euler Algorithm). Make a phase space portrait, i.e. plot the numerical results of Eqs. (3) in the x-v plane. What is the difference to the exact phase space portrait of Eqs. (1)?

b) Compute and plot the total energy E as a function of time t and compare them to the Euler method, as discussed in Problem 2.1.

c) Compute the eigenvalues of the stability matrix for Eq. (3). For which values of δt is the algorithm stable?

Problem 2.3: **Gauß-Legendre Quadrature**

Consider the Gauß-Legendre quadrature for a function $f(x)$. So we aim at computing the integral over $f(x)$ in the interval $[-1, 1]$ via

$$\int_{-1}^{1} f(x)\,\mathrm{d}x \approx \sum_{i=1}^{N} w_i f(x_i), \tag{4}$$

where the weight factors w_i and the nodes x_i $(i = 1, \dots, N)$ are determined such that Eq. (4) is exact for polynomials of degree $\leq 2N - 1$. Then, the nodes x_i are given by the roots of the Legendre polynomial $P_N(x)$ of degree N.

a) Show that

$$\int_{-1}^{1} f(x)\,\mathrm{d}x \approx A_1 f(x_1) + A_2 f(x_2) + A_3 f(x_3) + A_4 f(x_4) \tag{5}$$

with $A_1 = A_4 = (18 - \sqrt{30})/36 = 0.347854845137454$ and $A_2 = A_3 = (18 + \sqrt{30})/36 = 0.652145154862546$ being the roots of the Legendre polynomial $P_4(x) = (35x^4 - 30x^2 + 3)/8$.

b) Compute numerically the roots of $P_4(x)$ using the Newton-Raphson method and plot the difference between the estimated and the exact roots as a function of the number of Newton-Raphson iterations.

c) The error function is defined by

$$f(x) = \mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} \exp(-t^2)\,\mathrm{d}t. \tag{6}$$

Write subroutines that compute this function via Eq.(5) and via Boole's rule. Hint: Use the substitution $u = at + b$ to change the integration domain in Eq. (6) to the interval $[-1, 1]$.

d) Calculate the error function at $x = 2$ using Eq. (5) and Boole's rule. Determine the deviation from the exact value in both cases.