SUBMITTED BY: BREITENBERGER, NICODEMUS

1 Exercise: Numerical integration

```cpp
#include <iostream>
#include <fstream>
#include <vector>
#include <cmath>

using namespace std;

int main(int argc, char* argv[]) {

   double k = 1.0;
   double m = 1.0;
   double h = 0.01;

   // double h = 0.01; does not affect the simulation in the range of t = 40
   // greater timessteps lead to a divergence of the Ekin and Pges especially for the euler method

   ofstream euler;
   ofstream verlet;
   euler.open("euler.txt");
   verlet.open("verlet.txt");

   euler << "t Ekin Pges" << endl;
   verlet << "t Ekin Pges" << endl;

   double t0 = 0.0;

   vector<double> eulerx;
   vector<double> eulerv;
   vector<double> eulerF;

   vector<double> verletx;
   vector<double> verletv;
   vector<double> verletF;

   eulerx.push_back(1.0);
   eulerv.push_back(1.0);
   eulerF.push_back(0);

   verletx.push_back(1.0);
   verletv.push_back(1.0);
   verletF.push_back(0);
```

```

    for(int n=1;n<10000;n++){
        eulerx.push_back(eulerx[n-1] + h*eulerv[n-1]);
        eulerF.push_back(-k*eulerx[n-1]);
        eulerv.push_back(eulerv[n-1] + h/m*eulerF[n]);

        verletx.push_back(verletx[n-1]+verletv[n-1]*h+verletF[n]/(2*
    m)*pow(h,2));
        verletF.push_back(-k*verletx[n]);
        verletv.push_back(verletv[n-1] + h/(2*m)*(verletF[n-1] +
     verletF[n]));
    }

    for(int n=0;n<10000;n++){
        euler << t0+n*h << " " << 3/2*m*pow(eulerv[n],2) << " " << 3
     *m*eulerv[n] << endl;
        verlet << t0+n*h << " " << 3/2*m*pow(verletv[n],2) << " " <<
      3*m*verletv[n] << endl;
    }

    euler.close();
    verlet.close();

    system("gnuplot plot.gnu");

    return 0;
}
```

File plot.gnu:

```
# gnuplot script to plot NumInt output
set   autoscale                          # scale axes automatically
unset log                                # remove any log-scaling
unset label                              # remove any previous
    labels
set xtic auto                            # set xtics automatically
set ytic auto                            # set ytics automatically
set xlabel "t"
set ylabel "Ekin(t), Pges(t)"


set size 2,2
set origin 0,0
set multiplot layout 2,1 columnsfirst scale 1,1

set xr [0.0:20.0]
set yr [-20:20]
```

```
set title "Numerical integration - harmonic oscillator - Euler algorithm"
plot "euler.txt" using 1:2 title 'Ekin(t)' with points, "euler.txt" using 1:3 title 'Pges(t)' with points

set xr [0.0:20.0]
set yr [-20:20]
set title "Numerical integration - harmonic oscillator - Velocity-Verlet algorithm"
plot "verlet.txt" using 1:2 title 'Ekin(t)' with points, "verlet.txt" using 1:3 title 'Pges(t)' with points

unset multiplot
```

2 Exercise: Local Minimizer - C++

Given function:

$$f(x,y) = e^{-x^2-y^2}$$

For local minima/maxima the condition $\partial_x f(x,y) = 0$ and $\partial_y f(x,y) = 0$ has to be fullfilled.

$$\partial_x f(x,y) = -2xe^{-x^2-y^2}$$

and

$$\partial_y f(x,y) = -2ye^{-x^2-y^2}$$

Because e^z is never zero for any z, so $-2x = 0$ and $-2y = 0$, equal to $x = 0$ and $y = 0$. That means a local extremal point is at (0,0).

To determine if it is a local minimum or maximum the second derivatives have to be calculated. H is the Hessian matrix :

$$H(x,y) = \begin{pmatrix} f_{xx}(x,y) & f_{xy}(x,y) \\ f_{yx}(x,y) & f_{yy}(x,y) \end{pmatrix}$$

$$D(x_c,y_c) = \det(H(x,y)) = f_{xx}(x_c,y_c)f_{yy}(x_c,y_c) - [f_{xy}(x_c,y_c)]^2$$

With the cases:

- If $D(a,b) > 0$ and $f_{xx}(a,b) > 0$ (a,b) is a local minimum of f.

- If $D(a,b) > 0$ and $f_{xx}(a,b) < 0$ (a,b) is a local maximum of f.
- If $D(a,b) < 0$ then (a,b) is a saddle point of f.
- If $D(a,b) = 0$ then the derivative test is inconclusive.

$$f_{xx}(x,y) = -2e^{-x^2-y^2} + 4xe^{-x^2-y^2}$$
$$f_{yy}(x,y) = -2e^{-x^2-y^2} + 4ye^{-x^2-y^2}$$

$$f_{xy}(x,y) = f_{yx}(x,y) = 4xye^{-x^2-y^2}$$

$$D(0,0) = (-2+0)\cdot(-2+0) - 0 = 4 > 0$$
$$f_{xx}(0,0) = -2+0 = -2 < 0$$

This means that there is a local maximum at (0,0).