MAT801 Numerics I
FS16

Yannis Bähni
yannis.baehni@uzh.ch

SOLUTIONS SHEET 9

Exercise 1. First we have to plot the function on a suitable interval, say $[-4, 4]$, to get a feeling for the situation. From figure 1 we clearly see that the global minimizer is located in the interval $[0, 0.5]$ and the global maximizer in $[-1.5, -1]$. There are no other possibilities since $\lim_{x \to \pm\infty} f(x) = 0$ since the exponential grows faster than any power (the trigonometric function sin is also bounded). With the golden section method (listing 1) we get

$$x_{\text{min}} = 0.171765643671360 \qquad x_{\text{max}} = -1.104004845629775 \tag{1}$$

with a little trick: one has to consider $-f(x)$ to find the global maxima since the golden section method only detects global minima (unimodal) according to the lecture slides. If we want to apply the Newton method it is mandatory to choose a good starting value. From figure 2 we can tell that around -1 the maxima is located and around 0 the minima. With the Newton method (listing 2) we get

$$x_{\text{min}} = 0.171765558609737 \qquad x_{\text{max}} = -1.104004849603309 \tag{2}$$

Hence the results coincide quite well.

```
function [ xstar ] = golden_section( f,a,b,epsilon,maxit )
phi = (1 + sqrt(5))/2;
c = a + 1/phi^2 * (b - a);
d = a + 1/phi * (b - a);
if f(c) >= f(d);
    a = c;
else
    b = d;
end
xstar = (a + b)/2;
it = 1;
while (b - a)/(abs(c) + abs(d)) >= epsilon && it < maxit
    c = a + 1/phi^2 * (b - a);
    d = a + 1/phi * (b - a);
    if f(c) >= f(d);
        a = c;
    else
        b = d;
    end
    xstar = (a + b)/2;
    it = it + 1;
end
end
```

Listing 1. `src/golden_section.m`

```
function [ x ] = newton( F,invDF,x0,epsilon,maxit )
s = invDF(x0) * F(x0);
x = x0 - s;
it = 1;
while norm(x - x0) > epsilon && it < maxit
    x0 = x;
    s = invDF(x0) * F(x0);
    x = x0 - s;
    it = it + 1;
end
end
```

Listing 2. `src/newton.m`

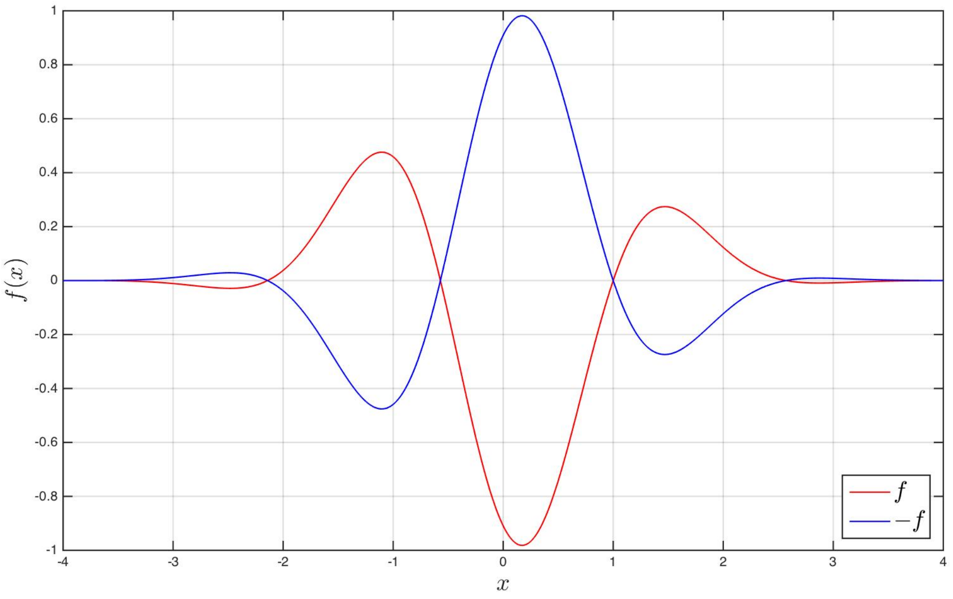

Figure 1. Plot of the function $f(x) := e^{-\frac{x^2}{2}}\sin(2(x-1))$ and $-f(x)$ on the interval $[-4,4]$.

Exercise 2. Remark: The Crank-Nicolson method goes also under the name *implicit trapezoidal rule* which I will use here (the name Crank-Nicolson method originates from PDEs whereas we are studying here ODEs).

a. The source code for the *explicit Euler method* can be found in listing 3 and the one for the *implicit trapezoidal rule* in listing 4.

Remark: Since the new discretization is given implicitely by $y_1 = y_0 + \frac{1}{2}h\left(f(x_0,y_0) + f(x_1,y_1)\right)$ we have to solve a (in general) nonlinear system of n equations. Let us formalize this discussion. Assume we have a first order IVP $y' = f(x,y)$ with $f \in C\left([a,b]\times\mathbb{R}^n;\mathbb{R}^n\right)$. Then we get by the implicit trapezoidal rule (superscrips are enumerations of the components)

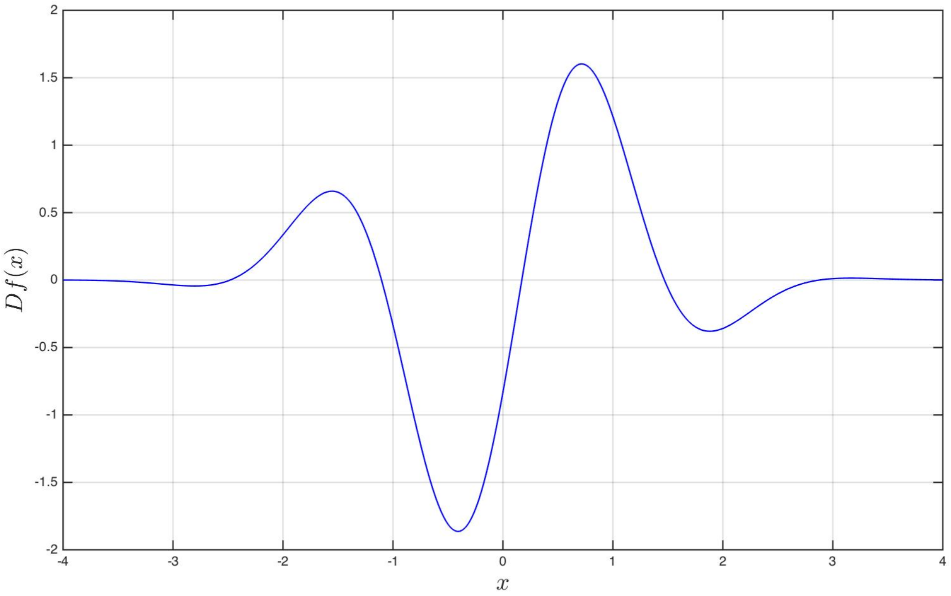

FIGURE 2. Plot of the derivative of the function $f(x)$ on the interval.

```
function [ t,y ] = EE( f,t0,tN,y0,N )
t = linspace(t0,tN,N+1);
h = (tN - t0)/N;
y = zeros(length(y0),N+1);
y(:,1) = y0;
for k = 1:N
    y(:,k+1) = y(:,k) + h * f(t(k),y(:,k));
end
end
```

LISTING 3. `src/EE.m`

$$\begin{bmatrix} y_1^1 \\ \vdots \\ y_1^n \end{bmatrix} = \begin{bmatrix} y_0^1 \\ \vdots \\ y_0^n \end{bmatrix} + \frac{1}{2}h \begin{bmatrix} f^1(x_0, y_0) + f^1(x_1, y_1) \\ \vdots \\ f^n(x_0, y_0) + f^n(x_1, y_1) \end{bmatrix} \tag{3}$$

Thus y_1 is a fixed point of the function $\Phi \in C(\mathbb{R}^n; \mathbb{R}^n)$ defined by

$$\Phi(\eta) := \begin{bmatrix} y_0^1 \\ \vdots \\ y_0^n \end{bmatrix} + \frac{1}{2}h \begin{bmatrix} f^1(x_0, y_0) + f^1(x_1, \eta) \\ \vdots \\ f^n(x_0, y_0) + f^n(x_1, \eta) \end{bmatrix} \tag{4}$$

or equivalently a root of the function $F \in C(\mathbb{R}^n; \mathbb{R}^n)$ defined by $F(\eta) := \Phi(\eta) - \eta$. That this is an equivalent condition can be shown as follows: assume $\eta \in \mathbb{R}^n$ is a fixed

```
function [ t,y ] = CN( f,t0,tN,y0,N )
t = linspace(t0,tN,N+1);
h = (tN - t0)/N;
y = zeros(length(y0),N+1);
y(:,1) = y0;
for k = 1:N
    F = @(eta) y(:,k) + h/2 * (f(t(k),y(:,k)) + f(t(k+1),eta)) - eta;
    y(:,k+1) = fsolve(F,y(:,k));
end
end
```

Listing 4. `src/CN.m`

point of Φ, then $\Phi(\eta) = \eta$ and thus $\phi(\eta) - \eta = 0 = F(\eta)$. Conversly assume $F(\eta) = 0$. But then it is immediate that $\Phi(\eta) - \eta = 0$ and so η is by definition a fixed point of Φ.

b.

- The *explicit Euler method* is given by $y_1 = y_0 + hf(x_0, y_0)$. This is a 1-stage Runge-Kutta method since comparison with $y_1 = y_0 + hb_1k_1$ yields $b_1 = 0$. Further by stipulating $k_1 = f(x_0, y_0)$ we immediately get $c_1 = 0$ and $a_{11} = 0$. Thus the Butcher tableaux of the explicit Euler method is given by

0	0
	1

Table 1. Butcher tableaux of the explicit Euler method.

The explicit Euler method is thus of order 1.

- The *implicit trapezoidal rule* is given by the Butcher tableaux

0	0	0
1	1/2	1/2
	1/2	1/2

Table 2. Butcher tableaux of the implicit midpoint rule.

since by $y_1 = y_0 + \frac{1}{2}h\left(f(x_0, y_0) + f(x_1, y_1)\right)$ we immediately see that the implicit trapezoidal rule corresponds to a 2-stage Runge-Kutta method with $k_1 = f\left(x_0, y_0\right)$ and $k_2 = f\left(x_1, y_1\right)$, $b_1 = b_2 = 1/2$. From $k_1 = f\left(x_0, y_0\right)$ it is immediate that $c_1 = a_{11} = a_{12} = 0$ and form $k_2 = f\left(x_1, y_1\right)$ it follows $c_2 = 1$. Plugging the initial definition of y_1 in k_2 we get

$$k_2 = f\left(x_1, y_1\right) = f\left(x_0 + h, y_0 + \frac{1}{2}h\left(f(x_0, y_0) + f(x_1, y_1)\right)\right) \tag{5}$$

and we see that $a_{21} = a_{22} = 1/2$. By the order-conditions for Runge-Kutta methods we thus have that the implicit midpoint rule is of order 2.

Since for the provided IVP we have $f(x, y) = \sin(x) + y$ and hence f is Lipschitz-continuous in the second variable (the identity is Lipschitz-continuous on every metric space) and thus the partial derivative $\frac{\partial f}{\partial y}$ is bounded we can use the global error estimation of a Runge-Kutta method. We have for the explicit Euler method that

$e_i = O(h^2)$ and thus for the global error we expect $E = O(h)$. This agrees quite well with the experimentally determined rate below in table 3. As one can see the experimentally determined rate agrees quite well with the theoretical one. In table 4 one can see the convergence rate for the implicit trapezoidal rule. Again the experminetally determined rate agrees quite well with the theoretical one.

0.940793142386221
0.965166057415819
0.975148778101867
0.980649145273068
0.984145218368224
0.986567192230107
0.988345385296780
0.989706881719533
0.990783029587586

Table 3. Experimentally determined convergence rate of the explicit Euler method.

2.001461346411569
2.000452946242377
2.000233579486706
2.000158778084204
2.000083456453824
2.000029677642035
2.000038440501600
2.000136978838223
1.999911382110358

Table 4. Experimentally determined convergence rate of the implicit trapezoidal rule.

Exercise 3. **a.** The source code can be found in listing 5.

```
function [ t,y ] = heun( f,t0,tN,y0,N )
h = (tN - t0)/N;
t = t0:h:tN;
y = zeros(length(y0),N+1);
y(:,1) = y0;
for k = 1:N
    y(:,k+1) = y(:,k) + .5 * h * (f(t(k),y(:,k)) + ...
        f(t(k) + h,y(:,k) + h * f(t(k),y(:,k))));
end
end
```

Listing 5. `src/heun.m`

b. The *Heun method* is given by

$$y_1 = y_0 + \frac{1}{2}h\left[f(x_0, y_0) + f\left(x_0 + h\right), y_0 + hf\left(x_0, y_0\right)\right] \tag{6}$$

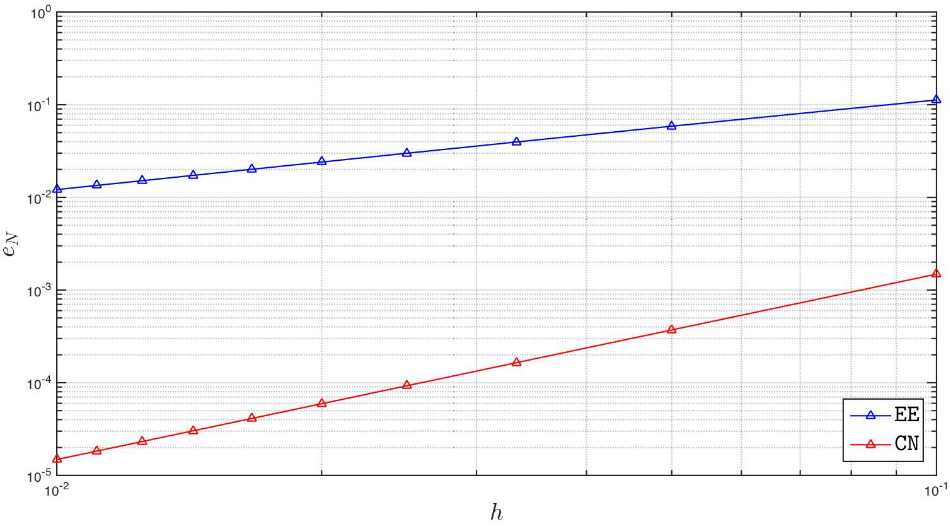

Figure 3. Plot of the discretization error of the explicit Euler method and the implicit trapezoidal rule applied to a test IVP with various stepsize h.

The Heun method can be seen as a 2-stage Runge-Kutta method as follows: comparison with $y_1 = y_0 + hb_1k_1 + hb_2k_2$ immediately yields $b_1 = b_2 = \frac{1}{2}$. Further by stipulating $k_1 = f(x_0, y_0)$ and $k_2 = f\left(x_0 + h, y_0 + hf\left(x_0, y_0\right)\right)$ we get by comparison with $k_1 = f\left(x_0 + c_1h, y_0 + ha_{11}k_1 + ha_{12}k_2\right)$ immediately $c_1 = a_{11} = a_{12} = 0$ and further by $k_2 = f\left(x_0 + c_2h, y_0 + ha_{21}k_+ + ha_{22}k_2\right)$ we get $c_2 = a_{21} = 1$ and $a_{22} = 0$. Thus the Butcher tableaux for the Heun method is given by

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

Table 5. Butcher tableaux of the Heun method.

Hence by the order-conditions for Runge-Kutta methods we have that the Heun method is of order 2. Thus we expect the Heun method to converge (since again for the given IVP the right-hand side is Lipschitz continuous and thus the partial derivative $\frac{\partial f}{\partial y}$ is bounded) with order $O(h^2)$ (since for the local error we have $O(h^3)$). As one can see in table 6 the experimentally determinde rate agrees quite well with the theoretical one.

Appendix A. Runge-Kutta Methods

I give some useful theorems and definitions to formalize the discussion in **Exercise 2** and **3**. First of all the definition found in [HLW06, p. 25].

Definition A.1. *Let b_i, a_{ij} $(i, j = 1, \ldots, s)$ be real numbers and let $c_i := \sum\limits_{j=1}^{s} a_{ij}$. An s-*stage *Runge-Kutta method* is given by*

2.082415387082784
2.048940505067162
2.035313005267801
2.027576969019389
2.021436800840220
2.018599763254124
2.016417457804357
2.014354882847216
2.012506169734002

TABLE 6. Experimentally determined convergence rate of the heun method.

$$
\begin{aligned}
k_i &= f\left(x_0 + c_i h, y_0 + h\sum_{j=1}^{s}\right) a_{ij}k_j, \qquad i = 1, \ldots, s \\
y_1 &= y_0 + h\sum_{i=1}^{s} b_i k_i
\end{aligned}
\tag{7}
$$

The numbers b_i, a_{ij} and c_i will usually be displayed after the outstanding work of J.C. Butcher in a so called Butcher tableaux

$$
\begin{array}{c|ccc}
c_1 & a_{11} & \ldots & a_{1s} \\
\vdots & \vdots & & \vdots \\
c_s & a_{s1} & \ldots & a_{ss} \\
\hline
 & b_1 & \ldots & b_s
\end{array}
$$

Further the

Definition A.2. *A Runge-Kutta method (or a general one-step method) has* order p, *if for all sufficiently regular problems $y' = f(x, y)$, $y(x_0) = y_0$ the* local error $y_1 - y(x_0 + h)$ *satisfies*

$$
y_1 - y(x_0 + h) = O(h^{p+1}) \quad \textit{as } h \to 0 \tag{8}
$$

To check the order of a Runge-Kutta method, one has to compute the Taylor series expansions of $y(x_0 + h)$ and y_1 around $h = 0$. This leads to the following algebraic conditions for the coefficients for orders 1, 2, and 3:

$$
\begin{aligned}
& \sum_{i=1}^{s} b_i = 1 && \text{for order 1;} \\
\text{in addition} \quad & \sum_{i=1}^{s} b_i c_i = 1/2 && \text{for order 2;} \\
\text{in addition} \quad & \sum_{i=1}^{s} b_i c_i^2 = 1/3 && \\
\text{and} \quad & \sum_{i=1}^{s}\sum_{j=1}^{s} b_i a_{ij} c_j = 1/6 && \text{for order 3;}
\end{aligned}
\tag{9}
$$

Since the definition of the order of a Runge-Kutta method only contains the local error, one wishes to have something about the *global error*. Such a theorem can be found in [HNW93, p. 160].

Theorem A.1. *Let U be a neighbourhood of $\{(x, y(x)) | x_0 \leqslant x \leqslant X\}$ where $y(x)$ is the exact solution of $y' = f(x, y)$, $y(x_0) = y_0$. Suppose that in U we have $\left\| \frac{\partial f}{\partial y} \right\| \leqslant L$, and that the local error estimate $\|y_i - y(x_i + h_i)\| = \|e_i\| \leqslant Ch_{i-1}^{p+1}$ holds for $i = 1, \ldots, N$ in U. Then the global error $E = y(X) - y_N$ can be estimated by*

$$\|E\| \leqslant h^p \frac{C}{L} \left(\exp\left(L(X - x_0)\right) - 1\right) \tag{10}$$

where $h := \max_i h_i$.

References

[HLW06] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration : structure-preserving algorithms for ordinary differential equations.* Springer series in computational mathematics. Berlin, Heidelberg, New York: Springer, 2006. ISBN: 978-3-540-30663-4.

[HNW93] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I (2nd Revised. Ed.): Nonstiff Problems.* New York, NY, USA: Springer-Verlag New York, Inc., 1993. ISBN: 0-387-56670-8.