TTK4130 assignment 4

Martin Brandt

February 7, 2019

1 Task 1

1.1

Relative tolerance for one orbit round: 10^{-3}.
Relative tolerance for two orbit rounds: 10^{-5}.
Relative tolerance for three orbit rounds: 10^{-9}.
We observe that the tolerance requirement goes up very fast as we increase the simulation time.

1.2

With ERK5 we need approximately $h = 0.0005$ to simulate three orbit rounds accurately. But now the simulation is very slow, while the variable step method was almost instantaneous.

1.3

Variable time step methods are based on estimating the error by simulating the system twice with two different order methods. If the error estimate is off because both methods are failing to accurately simulate the system, then the time step is set to something wrong and the simulation will quickly fall off the correct trajectory.

$\boxed{2}$ $k_1 = f(y_n, t_n)$,

$k_2 = f(y_n + h a_{21} k_1, t_n + h c_2)$,

$y_{n+1} = y_n + h(b_1 k_1 + b_2 k_2)$

ⓐ We have that

$\underline{\underline{0 \le c_2 \le 1}}$,

$\sum_j a_{ij} = c_i \Rightarrow \underline{\underline{a_{21} = c_2}}$,

$\sum b_i = 1 \Rightarrow \underline{\underline{b_1 + b_2 = 1}}$

$$k_2 = f(y_n + h a_{21} k_1, t_n + h c_2)$$

$$= f(y_n, t_n) + \frac{\partial f(y_n, t_n)}{\partial y} h a_{21} k_1 + \frac{\partial f(y_n, t_n)}{\partial t} h c_2 + O(h^2)$$

$$= f(y_n, t_n) + h a_{21} \left(\frac{\partial f(y_n, t_n)}{\partial y} f(y_n, t_n) + \frac{\partial f(y_n, t_n)}{\partial t} \right) + O(h^2)$$

$$= f(y_n, t_n) + h a_{21} \frac{df(y_n, t_n)}{dt} + O(h^2)$$

$$y_{n+1} = y_n + h b_1 f(y_n, t_n) + h b_2 \left(f(y_n, t_n) + h a_{21} \frac{df(y,t)}{dt} \right)$$

$$= y_n + h(b_1 + b_2) f(y_n, t_n) + h^2 b_2 a_{21} \frac{df(y_n, t_n)}{dt}$$

$$= y_n + h f(y_n, t_n) + \frac{h^2}{2} \frac{df(y_n, t_n)}{dt}$$

$\Rightarrow b_1 + b_2 = 1,$

$b_2 a_{21} = \frac{1}{2}$

$$\Rightarrow \underline{c_2 = a_{21}}, \quad \underline{b_2 = \frac{1}{2a_{21}}}, \quad \underline{b_1 = 1 - \frac{1}{2a_{21}}}$$

$(a_{21} = a_{21})$

ⓑ $\dot{y} = \lambda y$

$$y_{n+1} = y_n + hb_1 \lambda y_n + hb_2 \lambda (y_n + h a_{21} \lambda y_n)$$

$$= y_n (1 + (b_1 + b_2) h\lambda + b_2 a_{21} (h\lambda)^2)$$

$$= y_n (1 + h\lambda + \frac{1}{2} (h\lambda)^2)$$

$$\Rightarrow \underline{R(h\lambda) = 1 + h\lambda + \frac{1}{2}(h\lambda)^2}$$

The parameters dissappeared because of the constraints.

2 Task 3

2.1

```
g = 9.81;
x_d = 1.32;
k = 2.4;
h = 0.01;
t = 10;
N = t/h;
y_0 = [2; 0];
f = @(y) [y(2); - g * (1 - (x_d/y(1))^k)];
y = [y_0];

for i = 1:N
    k_1 = f(y(1:2, i));
    k_2 = f(y(1:2, i) + h * k_1);
    y = [y y(1:2, i) + h * (k_1 + k_2) / 2];
end

plot(y(1,:));
grid on;
```

Figure 1: Modified euler code

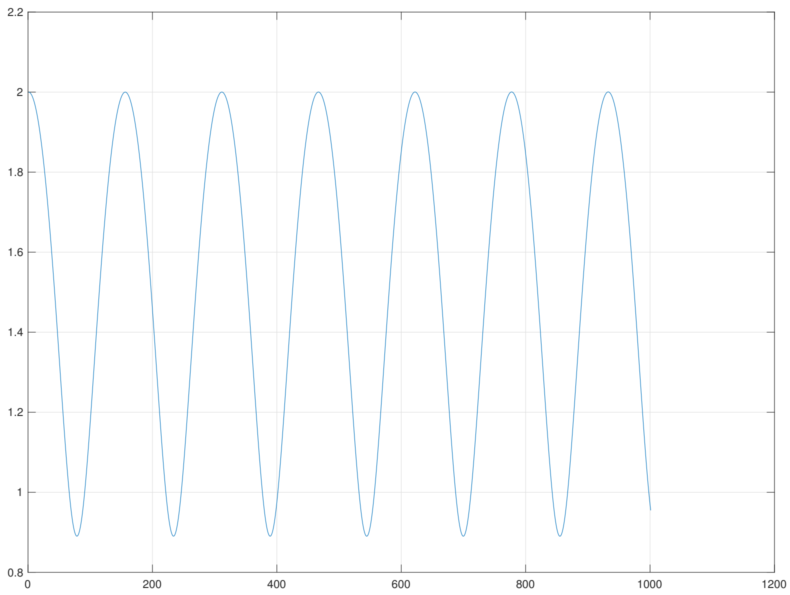

Figure 2: Modified euler plot

The modified euler method seems to simulate the system quite well - the oscillations are neither decaying or growing.

2.2

```
g = 9.81;
x_d = 1.32;
k = 2.4;
h = 0.01;
t = 10;
N = t/h;
y_0 = [2; 0];
f = @(y) [y(2); - g * (1 - (x_d/y(1))^k)];
y = [y_0];

opt = optimset('Display','off','TolFun',1e-8); % Options for fsolve

for i = 1:N
    r = @(ynext) (y(:,i) + h*f(ynext) - ynext);
    y(:,i+1) = fsolve(r, y(:,i), opt);
end

plot(y(1,:));
grid on;
```

Figure 3: Implicit euler code

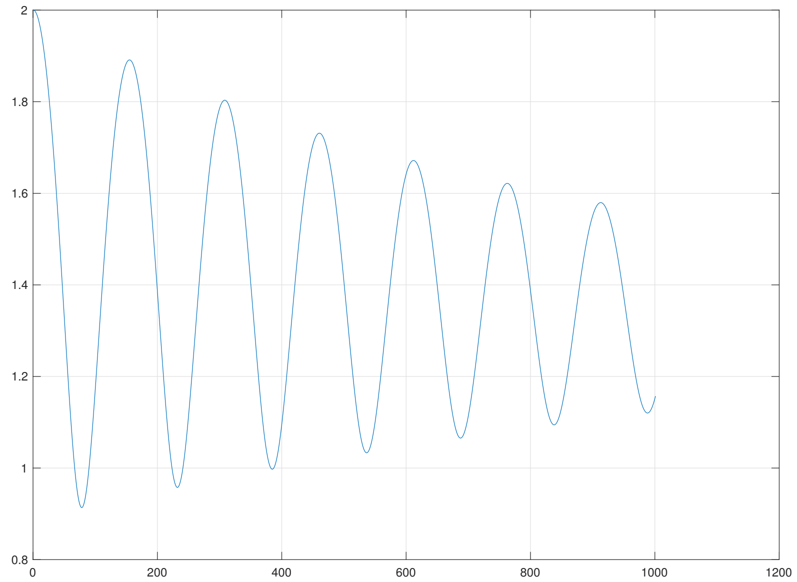

Figure 4: Implicit euler plot

The implicit euler method decays quite fast, as it typical for the implicit methods, as they have a much larger stability region.

2.3

```
g = 9.81;
x_d = 1.32;
k = 2.4;
h = 0.01;
t = 10;
N = t/h;
y_0 = [2; 0];
f = @(y) [y(2); - g * (1 - (x_d/y(1))^k)];
y = [y_0];

opt = optimset('Display','off','TolFun',1e-8); % Options for fsolve

for i = 1:N
    r = @(ynext) (y(:,i) + h*f((y(:,i) + ynext)/2) - ynext);
    y(:,i+1) = fsolve(r, y(:,i), opt);
end

plot(y(1,:));
grid on;
```

Figure 5: Implicit midpoint code

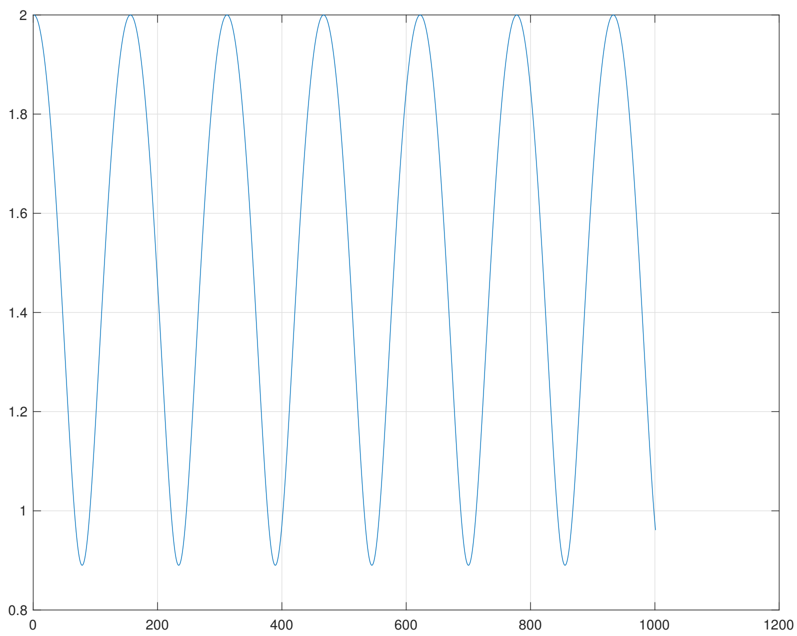

Figure 6: Implicit midpoint plot

The implicit midpoint method is stable!

(d) $$E = \frac{mg}{\kappa - 1} \frac{x_d^{\kappa}}{x^{\kappa-1}} + mgx + \frac{1}{2} m \dot{x}^2$$

$$\dot{E} = \frac{mg}{\kappa - 1} x_d^{\kappa} (-\kappa + 1) x^{-\kappa} + mg\dot{x} + m\dot{x}\ddot{x}$$

$$= -mg x_d^{\kappa} x^{-\kappa} + mg\dot{x} + m\dot{x}(-g(1 - x_d^{\kappa} x^{-\kappa}))$$

$$= -mg x_d^{\kappa} x^{-\kappa} \dot{x} + mg\dot{x} - mg\dot{x} + mg x_d^{\kappa} x^{-\kappa} \dot{x}$$

$$\underline{\underline{\dot{E} = 0}} \quad \square$$

2.4

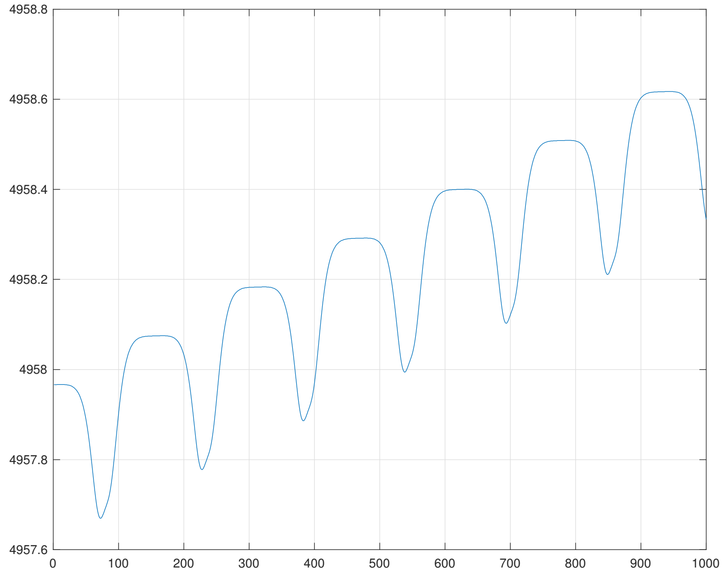

Figure 7: Modified Euler energy

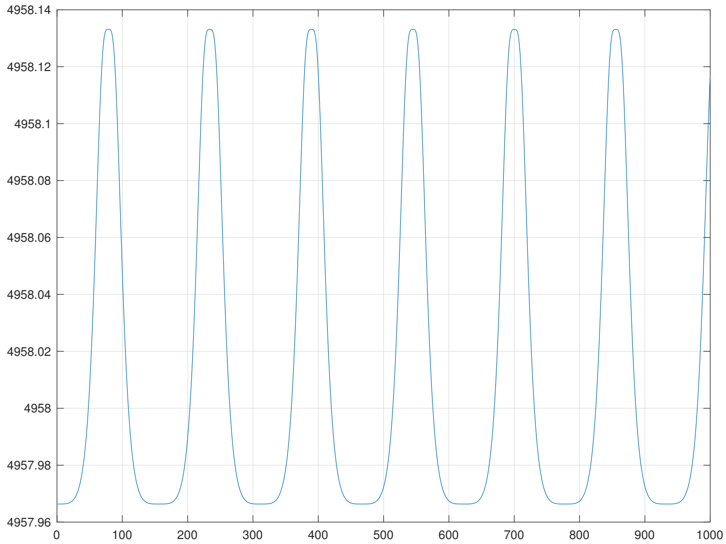

Figure 8: Implicit midpoint energy

We observe that modified Euler, which is explicit, has an increasing energy level. Since explicit methods have polynomial stability functions, it makes some sense that the energy is increasing. Implicit midpoint, which has constant amplitude oscillations, have a oscillating energy level which doesn't increase or decrease over time, which also makes sense. The implicit Euler method, which has added damping to the oscillations, have a decreasing energy level, which also is to be expected. Why the modified Euler method has oscillations with constant amplitude, but an increasing energy level, I do not understand.

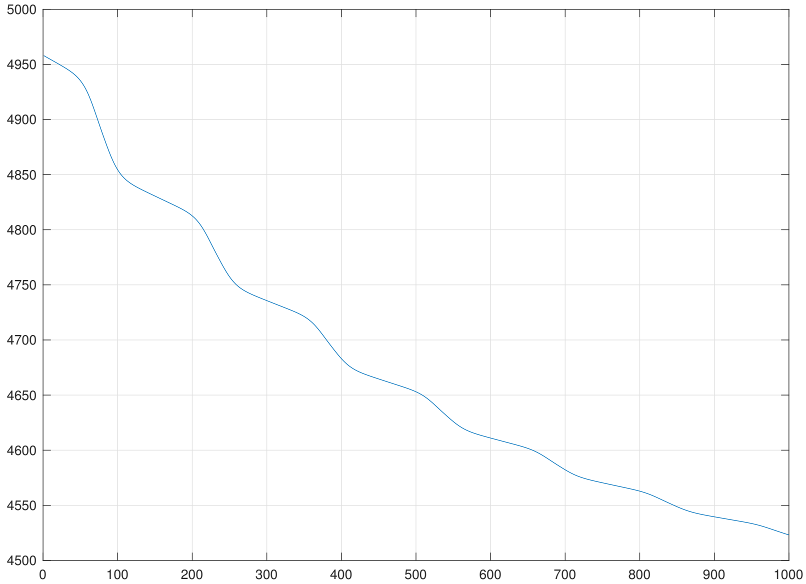

Figure 9: Implicit Euler energy

1.

$\boxed{4}$ ⓐ $\dot{z}_2 = -z_3 + 4z_4^3,$

$\dot{z}_3 = -z_1 + 2z_2,$

$z_1 = z_4^3 - z_2 + q_1,$

$z_4 = -z_1 + z_3 - q_2$

$$x = [z_2 \ z_3]^T, \quad y = [z_1 \ z_4]^T, \quad u = [q_1 \ q_2]^T$$

$$\dot{x} = \begin{bmatrix} -z_3 + 4z_4^3 \\ -z_1 + 2z_2 \end{bmatrix}, \quad 0 = \begin{bmatrix} z_4^3 - z_1 - z_2 + q_1 \\ z_3 - z_1 - z_4 - q_2 \end{bmatrix}$$

ⓑ $$\dot{g} = \begin{bmatrix} 3z_4^2\dot{z}_4 - \dot{z}_1 - \dot{z}_2 + \dot{q}_1 \\ \dot{z}_3 - \dot{z}_1 - \dot{z}_4 - \dot{q}_2 \end{bmatrix}$$

With a little algebra this can be solved for $\dot{z}_1$ and $\dot{z}_4$:

$$\dot{z}_4 = \frac{4z_4^3 + 2z_2 - z_1 - z_3 - \dot{q}_1 - \dot{q}_2}{(1+3z_4^2)}, \quad \dot{z}_1 = \dot{z}_3 - \dot{z}_4 - \dot{q}_2$$

so we only need to differentiate the system once to get an ODE, i.e. the index is 1.

Scanned by CamScanner

2.

ⓐ

$$\dot{z}_2 = q_1 - z_1,$$
$$\dot{z}_3 = q_2 - (1+a)z_2 - at(q_1 - z_1),$$
$$q_3 = atz_2 + z_3$$

$$x = [z_2 \; z_3]^T, \; y = z_1, \; u = [q_1 \; q_2 \; q_3 \; a]$$

$$\dot{x} = f(t,x,y,u) = \begin{bmatrix} q_1 - z_1 \\ q_2 - (1+a)z_2 - at(q_1 - z_1) \end{bmatrix},$$

$$0 = g(t,x,y,u) = atz_2 + z_3 - q_3$$

ⓑ $$\dot{g} = az_2 + at\dot{z}_2 + \dot{z}_3 - \dot{q}_3$$

$$\cancel{\dot{g} = a\dot{z}_2 + a\dot{z}_2}$$

$$= az_2 + at(q_1 - z_1) + q_2 - (1+a)z_2 - at(q_1 - z_1) - \dot{q}_3$$
$$= az_2 - (1+a)z_2 + q_2 - \dot{q}_3 = -z_2 + q_2 - \dot{q}_3$$

$$\ddot{g} = -\dot{z}_2 + \dot{q}_2 - \ddot{q}_3 = z_1 - q_1 + \dot{q}_2 - \ddot{q}_3$$

$$\dddot{g} = \dot{z}_1 - \dot{q}_1 + \ddot{q}_2 - \dddot{q}_3 = 0$$

Since we have to differentiate g thrice, the index is 3.

3. ⓐ

$$\dot{q}(t) = r - G^T \eta,$$
$$M\dot{v} = Fq - G^T\lambda,$$
$$0 = Gv,$$
$$r = Gq$$

$$x = [q \; v]^T, \quad y = [\eta, \lambda]^T, \quad u = [G \; M \; F r]$$

$$\dot{x} = \begin{bmatrix} v - G^T\eta \\ M^{-1}Fq - M^{-1}G^T\lambda \end{bmatrix}, \quad g = \begin{bmatrix} Gv \\ Gq - r \end{bmatrix} = 0$$

ⓑ $$\dot{g} = \begin{bmatrix} G\dot{v} \\ G\dot{q} - \dot{r} \end{bmatrix} = \begin{bmatrix} GM^{-1}(Fq - G^T\lambda) \\ G(v - G^T\eta) - \dot{r} \end{bmatrix}$$

If we differentiate again we are able to solve for $\dot{x}$ and $\dot{\eta}$.
Notice that we then get a $\ddot{r}$-term, which is not able to solve by regarding r as a state, so it must be a parameter (?).
Anyway index is 2.

4. (a) $m_1 \ddot{x}_1 = k(x_2 - x_1 - x_0) + F$

$$m_2 \ddot{x}_2 = -k(x_2 - x_1 - x_0)$$

$$x_2 = r$$

$\mathbf{x} = [x_1 \; x_2 \; \dot{x}_1 \; \dot{x}_2]$, $y = [F]$,

$u = [m_1 \; m_2 \; k \; x_0 \; r]$

$$\dot{\mathbf{x}} = f(x, y, u) = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \frac{k}{m_1}(x_2 - x_1 - x_0) + F \\ -\frac{k}{m_2}(x_2 - x_1 - x_0) \end{bmatrix}$$

$$0 = g(x, y, u) = r - x_2$$

(b) $\dot{g} = \dot{r} - \dot{x}_2$

$$\ddot{g} = \ddot{r} - \ddot{x}_2 = \ddot{r} + \frac{k}{m_2}(x_2 - x_1 - x_0)$$

$$\dddot{g} = \dddot{r} + \frac{k}{m_2}(\dot{x}_2 - \dot{x}_1)$$

$$\ddddot{g} = \ddddot{r} + \frac{k}{m_2}\left(-\frac{k}{m_2}(x_2 - x_1 - x_0) - \frac{k}{m_1}(x_2 - x_1 - x_0) + F\right)$$

$\Rightarrow$ <u>index 5</u> (?)

$\boxed{5}$ 1.

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{3} & \frac{1}{3} & & \\ \frac{2}{3} & 0 & \frac{2}{3} & \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array}$$

$$R(s) = 1 + s b^{T} (\mathbb{I} - sA)^{-1} \mathbb{1}$$

$$\mathbb{I} - sA = \begin{bmatrix} 1 & 0 & 0 \\ -s/3 & 1 & 0 \\ 0 & -\frac{2}{3}s & 1 \end{bmatrix}$$

Solve system to find inverse:

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ -s/3 & 1 & 0 & 0 & 1 & 0 \\ 0 & -\frac{2s}{3} & 1 & 0 & 0 & 1 \end{array}\right] \sim \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & s/3 & 1 & 0 \\ 0 & \cancel{\frac{2s}{3}}0 & 1 & \frac{2}{9}s^2 & \frac{2s}{3} & 1 \end{array}\right] = \left[\mathbb{I} \,\middle|\, (\mathbb{I} - sA)^{-1}\right]$$

$$\Rightarrow R_1(s) = 1 + s \begin{bmatrix} \frac{1}{4} & 0 & \frac{3}{4} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ s/3 & 1 & 0 \\ \frac{2}{9}s^2 & \frac{2s}{3} & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= s\left(\frac{1}{4} + \frac{1}{6}s^2 + \frac{1}{2}s + \frac{3}{4}\right) + 1$$

$$R_1(s) = 1 + s + \frac{1}{2}s^2 + \frac{1}{6}s^3$$

2.

$$\begin{array}{c|cc} 0 & 1/4 & -1/4 \\ 2/3 & 1/4 & 5/12 \\ \hline & 1/4 & 3/4 \end{array}$$

$$\mathbb{I} - sA = \begin{bmatrix} 1 - s/4 & s/4 \\ -s/4 & 1 - 5/12\, s \end{bmatrix}$$

$$(\mathbb{I} - sA)^{-1} = \frac{1}{(1 - s/4)(1 - 5/12\, s) + s^2/16} \begin{bmatrix} 1 - 5/12\, s & -s/4 \\ s/4 & 1 - s/4 \end{bmatrix}$$

$$= \frac{1}{1 - \frac{2}{3}s + \frac{1}{6}s^2} \begin{bmatrix} 1 - 5/12\, s & -s/4 \\ s/4 & 1 - s/4 \end{bmatrix}$$

$$R_2(s) = 1 + s \cdot \frac{1}{1 - \frac{2}{3}s + \frac{1}{6}s^2} \begin{bmatrix} 1/4 & 3/4 \end{bmatrix} \cdot \begin{bmatrix} 1 - 5/12\, s & -s/4 \\ s/4 & 1 - s/4 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$R_2(s) = 1 + \frac{s(1 - \frac{1}{6}s)}{1 - \frac{2}{3}s + \frac{1}{6}s^2} \approx \frac{1 - \frac{2}{3}s + \frac{1}{6}s^2 + s - \frac{1}{6}s^2}{1 - \frac{2}{3}s + \frac{1}{6}s^2} = \frac{1 + \frac{1}{3}s}{1 - \frac{2}{3}s + \frac{1}{6}s^2}$$