Σ 4+7+7 = 18p

Solutions to Problem Set 3

Sorry, for uploading that I don't forget it...

Fundamentals of Simulation Methods 8 ECTS
Heidelberg University WiSe 20/21

Elias Olofsson
ub253@stud.uni-heidelberg.de

November 25, 2020

Reminder to myself: Also provide link to pdf with comments on code

1 Order of an ODE integration scheme (4 pts)

Show that the explicit midpoint method is second order accurate by calculating the local and global truncation errors.

For an ordinary differential equation on the general form

$$\frac{dy}{dt} = f(y), \tag{1}$$

for the function $y(t)$, the explicit midpoint method takes the form

$$y_{n+1} = y_n + \Delta t\ f\left(y_n + \frac{\Delta t}{2} f(y_n, t_n), t_n + \frac{\Delta t}{2}\right), \tag{2}$$

where y_n is the n:th iterate of the numerical integration, corresponding to the time $t_n = t_0 + n \cdot \Delta t$. If we let $y(t)$ be the exact solution to eq.(1), then the local truncation error is defined as

$$\epsilon = y(t_{n+1}) - y_{n+1}. \tag{3}$$

To find the LTE, we can reformulate the expression for y_{n+1} in a smart way, noting that we can substitute the function f for the exact derivative y' and use a backwards Taylor expansion around $t_n + \frac{\Delta t}{2}$ as per

$$\begin{aligned}
y_{n+1} &= y_n + \Delta t\ f\left(y_n + \frac{\Delta t}{2} f(y_n, t_n), t_n + \frac{\Delta t}{2}\right) && (4)\\
&= y_n + \Delta t\ f\left(y(t_n) + \frac{\Delta t}{2} y'(t_n), t_n + \frac{\Delta t}{2}\right) && (5)\\
&= y_n + \Delta t\ f\left(y(t_n + \frac{\Delta t}{2}) + \mathcal{O}(\Delta t^2), t_n + \frac{\Delta t}{2}\right), && (6)
\end{aligned}$$

which we then can approximate by taking the small second order term out of the function expression and substituting for the exact derivative at $t_n + \frac{\Delta t}{2}$,

$$\simeq y_n + \Delta t \left[f\left(y(t_n + \frac{\Delta t}{2}), t_n + \frac{\Delta t}{2} \right) + \mathcal{O}(\Delta t^2) \right] \tag{7}$$

$$= y_n + \Delta t \; y'(t_n + \frac{\Delta t}{2}) + \mathcal{O}(\Delta t^3) \tag{8}$$

$$\simeq y_n + \Delta t \; \frac{y(t_{n+1}) - y(t_n)}{\Delta t} + \mathcal{O}(\Delta t^3) \tag{9}$$

$$= y_n + y(t_{n+1}) - y(t_n) + \mathcal{O}(\Delta t^3). \tag{10}$$

as discussed in tutorial, this is not so obvious

From eq.(8) to eq.(9) we used the central finite difference centered at $t_n + \frac{\Delta t}{2}$ with steps $\pm\frac{\Delta t}{2}$ to approximate the derivative y'. Thus, we finally arrive at the expression for the local truncation error,

is this $\mathcal{O}(\Delta t^2)$

$$\therefore \quad \epsilon = y(t_{n+1}) - y_{n+1} = y(t_n) - y_n + \mathcal{O}(\Delta t^3) = \mathcal{O}(\Delta t^3). \tag{11}$$

In other words, the error in each individual step will for this numerical integration scheme be of third order in stepsize Δt. However, since we need $N = T/\Delta t$ steps in total to integrate over time duration T, the global truncation error will scale as per

$$N\epsilon = \frac{T}{\Delta t}\mathcal{O}(\Delta t^3) = \mathcal{O}(\Delta t^2), \tag{12}$$

4 / 4p

one order less in accuracy.

2 Integration of a stiff equation (8 pts)

Out of time constraints, I was not able to type this section out here in this document. Instead, please refer to my Jupyter Notebook `fsm_ex3.ipynb` for the implementation and short comments. Alternatively, a PDF- or HTML-version of said notebook can also be viewed if preferred over the executable code.

(no need for HTML)

Ok, see comments in pdf. You could also just "paste" the code / jupyter notebook here

3 Double pendulum (8 pts)

Derive the Lagrangian equations of motions

$$\frac{\mathrm{d}}{\mathrm{dt}}\frac{\partial L}{\partial \dot{\phi}} - \frac{\partial L}{\partial \phi} = 0, \tag{13}$$

for a friction-less double pendulum with masses m_1,m_2 and length of rods l_1, l_2 where the Lagrangian given by the expression

$$\begin{aligned} L =& \frac{m_1}{2}\left(l_1\dot{\phi}_1\right)^2 + \frac{m_2}{2}\left[\left(l_1\dot{\phi}_1\right)^2 + \left(l_2\dot{\phi}_2\right)^2 + 2l_1l_2\dot{\phi}_1\dot{\phi}_2\cos\left(\phi_1 - \phi_2\right)\right] \\ &- m_1gl_1\left(1-\cos\phi_1\right) - m_2g\left[l_1\left(1-\cos\phi_1\right) + l_2\left(1-\cos\phi_2\right)\right], \end{aligned} \tag{14}$$

where ϕ_1 and ϕ_2 are the angles for respective pendulum.

Starting by defining the conjugate momenta

$$q \equiv \frac{\partial L}{\partial \dot{\phi}}, \tag{15}$$

which we then can for both angles ϕ_1 and ϕ_2 get the first order derivative, using eq.(13) with eq.(14) as per

$$\begin{aligned}\dot{q}_1 &= \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\phi}_1} = \frac{\partial L}{\partial \phi_1} \\ &= -m_2 l_1 l_2 \dot{\phi}_1 \dot{\phi}_2 \sin(\phi_1 - \phi_2) - g(m_1 + m_2) l_1 \sin\phi_1,\end{aligned} \tag{16}$$

$$\begin{aligned}\dot{q}_2 &= \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\phi}_2} = \frac{\partial L}{\partial \phi_2} \\ &= m_2 l_1 l_2 \dot{\phi}_1 \dot{\phi}_2 \sin(\phi_1 - \phi_2) - g m_2 l_2 \sin\phi_2.\end{aligned} \tag{17}$$

By explicitly calculating the conjugate momenta as defined in eq.(15) with the expression for L from eq.(14), we get

$$q_1 = \frac{\partial L}{\partial \dot{\phi}_1} = (m_1 + m_2) l_2^2 \dot{\phi}_1 + m_2 l_1 l_2 \dot{\phi}_2 \cos(\phi_1 - \phi_2) \tag{18}$$

$$q_2 = \frac{\partial L}{\partial \dot{\phi}_2} = m_2 l_2^2 \dot{\phi}_2 + m_2 l_1 l_2 \dot{\phi}_1 \cos(\phi_1 - \phi_2), \tag{19}$$

which can be written in matrix form $\boldsymbol{q} = \mathbf{A}\dot{\boldsymbol{\phi}}$ as

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} (m_1 + m_2) l_2^2 & m_2 l_1 l_2 \cos(\phi_1 - \phi_2) \\ m_2 l_1 l_2 \cos(\phi_1 - \phi_2) & m_2 l_2^2 \end{bmatrix} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 . \end{bmatrix} \tag{20}$$

The matrix $\mathbf{A}$ is can be inverted to yield explicit expressions for $\dot{\boldsymbol{\phi}}$, and performing the calculations we arrive at

$$\dot{\phi}_1 = \frac{m_2 l_2^2 q_1 - m_2 l_1 l_2 \cos(\phi_1 - \phi_2) q_2}{C_1} \tag{21}$$

$$\dot{\phi}_2 = \frac{(m_1 + m_2) l_2^2 q_2 - m_2 l_1 l_2 \cos(\phi_1 - \phi_2) q_2}{C_1} \tag{22}$$

where

$$C_1 = l_1^2 l_2^2 [(m_1 + m_2) m_2 - m_2^2 \cos^2(\phi_1 - \phi_2)]. \tag{23}$$

Thus, the full system of first order differential equations of motion for the given system is then given by eq.(16-17) and eq.(21-23). These equations can be arranged into a 4-vector ODE on the form $\dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y})$ where $\boldsymbol{y} = (\phi_1, \phi_2, q_1, q_2)$.

The above text only concerned the first part (a) of exercise 3 of this problem set. However, due to time constraints, I was not able to type the remainder of this section here in this document. Instead, please refer to my Jupyter Notebook `fsm_ex3.ipynb` for the implementation and short comments. Alternatively, a PDF- or HTML-version of said notebook can also be viewed if preferred over the executable code.