Goethe-Center for Scientific Computing (G-CSC)
Goethe-Universität Frankfurt am Main

Modeling and Simulation I

(Practical SIM1, WS 2018/19)
and
NeuroBioInformatik
(Übung NBI, WS 2018/19)

M. Huymayer, J. Wang, Dr. A. Nägel, Dr. M. Hoffer

Exercise sheet 5 (Due: Mo., 10.12.2018, 16h)

Aufgabe 1 (8P + 2P + 2P)
In `Sheet02, Exercise 3` we solved the two-body system, consisting of the earth and an artificial satellite. Now, with far more capable numerical solvers, we can finally solve a much more realistic N-Body system, such as our own solar system. Analogous to `Sheet02, Exercise 3`, the force between two bodies $body_i$ and $body_j$ is defined as:

$$F_{ij} = G m_i m_j \frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_j - \mathbf{r}_i\|^3} \tag{1}$$

where G is the gravitational constant, $\mathbf{r}_i$ is defined as the location of $body_i$ and $\|\mathbf{r}_j - \mathbf{r}_i\|$ is the magnitude of the distance between $\mathbf{r}_i$ and $\mathbf{r}_j$. The system of ODEs that describes the behavior of n bodies in space is defined as follows:

$$\begin{aligned}
\frac{d^2\mathbf{r}_1}{dt^2} &= \sum_{\substack{j=1\\ j\neq 1}}^{n} G m_j \frac{\mathbf{r}_j - \mathbf{r}_1}{\|\mathbf{r}_j - \mathbf{r}_1\|^3} \\
&\vdots \\
\frac{d^2\mathbf{r}_i}{dt^2} &= \sum_{\substack{j=1\\ j\neq i}}^{n} G m_j \frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_j - \mathbf{r}_i\|^3} \\
&\vdots \\
\frac{d^2\mathbf{r}_n}{dt^2} &= \sum_{\substack{j=1\\ j\neq n}}^{n} G m_j \frac{\mathbf{r}_j - \mathbf{r}_n}{\|\mathbf{r}_j - \mathbf{r}_n\|^3}
\end{aligned}$$

Our simulation will start at 01.12.2018 and ends exactly 31558149 seconds later which is the duration of one earth orbit. The following bodies shall be simulated:

1. Sun
2. Mercury
3. Venus
4. Earth
5. Moon (Luna)
6. Mars
7. Jupiter
8. Saturn
9. Uranus
10. Neptune
11. 1P/Halley
12. Eros
13. 67P/Churyumov-Gerasimenko

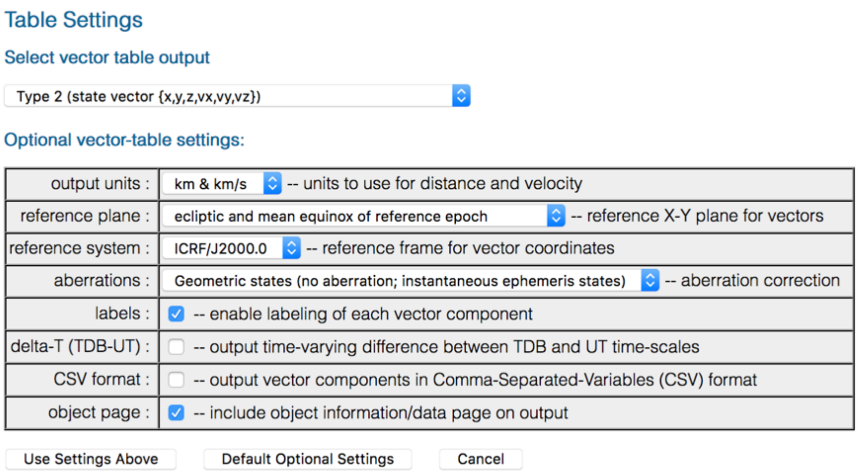
To acquire the initial values (mass, **r** and **v**), visit the NASA HORIZONS webpage (http://ssd.jpl.nasa.gov/horizons.cgi). To get data in the required format, the default settings have to be changed:

Current Settings

Ephemeris Type [change] : **VECTORS**
Target Body [change] : **Mars** [499]
Coordinate Origin [change] : **Solar System Barycenter (SSB)** [500@0]
Time Span [change] : Start=**2018-12-01**, Stop=**2019-11-30**, Step=**1 d**
Table Settings [change] : quantities code=**2**; output units=**KM-S**; labels=**NO**
Display/Output [change] : ***default*** (formatted HTML)

Generate Ephemeris

Necessary changes in table settings:



The settings are listed below:

Ephemeris:

- Ephemeris Type: Vector

Coordinate Origin:

- Coordinate Origin: Solar System Barycenter (SSB) [500@0]

Time Span:

- Time Span: 2018-12-01 to 2019-11-30, 00:00
Format: (yyyy-mm-dd, HH-MM)

Table Settings:

- Output Units: km & km/s
- Quantities Code: 2 (state vector {x,y,z,vx,vy,vz})

After adjusting the settings, click "Generate Ephemeris". The result is shown as html output, which will look like this:

Results

```
*******************************************************************************
Ephemeris / WWW_USER Wed Nov 28 10:36:58 2018 Pasadena, USA      / Horizons
*******************************************************************************
Target body name: Sun (10)                        {source: DE431mx}
Center body name: Solar System Barycenter (0)     {source: DE431mx}
Center-site name: BODY CENTER
*******************************************************************************
Start time      : A.D. 2018-Dec-01 00:00:00.0000 TDB
Stop  time      : A.D. 2019-Nov-30 00:00:00.0000 TDB
Step-size       : 1440 minutes
*******************************************************************************
Center geodetic : 0.00000000,0.00000000,0.0000000 {E-lon(deg),Lat(deg),Alt(km)}
Center cylindric: 0.00000000,0.00000000,0.0000000 {E-lon(deg),Dxy(km),Dz(km)}
Center radii    : (undefined)
Output units    : KM-S
Output type     : GEOMETRIC cartesian states
Output format   : 2 (position and velocity)
Reference frame : ICRF/J2000.0
Coordinate systm: Ecliptic and Mean Equinox of Reference Epoch
*******************************************************************************
JDTDB
   X     Y     Z
   VX    VY    VZ
*******************************************************************************
$$SOE
2458453.500000000 = A.D. 2018-Dec-01 00:00:00.0000 TDB
 X =-7.697595619083715E+04 Y = 1.101604469399394E+06 Z =-9.498528508639545E+03
 VX=-1.336041555364204E-02 VY= 3.829635187066665E-03 VZ= 3.360187927427952E-04
2458454.500000000 = A.D. 2018-Dec-02 00:00:00.0000 TDB
 X =-7.813057529188611E+04 Y = 1.101934836389468E+06 Z =-9.469481425182021E+03
 VX=-1.336690700834025E-02 VY= 3.817747143317579E-03 VZ= 3.363693395897519E-04
2458455.500000000 = A.D. 2018-Dec-03 00:00:00.0000 TDB
```

The object properties, such as mass, can be found in the "Object Data Page":

Object Data Page

```
 Revised: July 31, 2013                   Sun                                 10

 PHYSICAL PROPERTIES (updated 2018-Aug-15):
  GM, km^3/s^2          = 132712440041.93938  Mass, 10^24 kg        = ~1988500
  Vol. mean radius, km  = 695700              Volume, 10^12 km^3    = 1412000
  Solar radius (IAU)    = 696000 km           Mean density, g/cm^3  = 1.408
  Radius (photosphere)  = 696500 km           Angular diam at 1 AU  = 1919.3"
  Photosphere temp., K  = 6600 (bottom)       Photosphere temp., K  = 4400(top)
  Photospheric depth    = ~500 km             Chromospheric depth   = ~2500 km
  Flatness, f           = 0.00005             Adopted sid. rot. per.= 25.38 d
  Surface gravity       =  274.0 m/s^2        Escape speed, km/s    =  617.7
  Pole (RA,DEC), deg.   = (286.13, 63.87)     Obliquity to ecliptic = 7.25 deg.
  Solar constant (1 AU) = 1367.6 W/m^2        Luminosity, 10^24 J/s = 382.8
  Mass-energy conv rate = 4.260 x 10^9 kg/s   Effective temp, K     = 5772
  Sunspot cycle         = 11.4 yr             Cycle 24 sunspot min. = 2008 A.D.

  Motion relative to nearby stars = apex : R.A.= 271 deg.; DEC.= +30 deg.
                                    speed: 19.4 km/s (0.0112 au/day)
  Motion relative to 2.73K BB/CBR = apex : l= 264.7 +- 0.8; b= 48.2 +- 0.5 deg.
                                    speed: 369 +-11 km/s
```

Hints:

- To allow an intuitive interpretation of the results, a Java program called `VRL-Solar-System-Viewer` has been published on the webpage. It allows the visulization of your simulated solar system. The program requires a recent Java installation (Version > 1.8.70). To start the program, either double-click on the .jar-file or run the program via `java -jar VRL-Solar-System-Viewer.jar`. The source code is also available on GitHub.

- Stick to the order of the planetary objects as specified above. Otherwise, results cannot be interpreted by the `VRL-Solar-System-Viewer`. Use the time slider at the bottom of the program window to visualize planetary movements.

Tasks/Questions:

(a) Setup the right-hand side for the N-Body system with the 13 bodies listed above and simulate the time interval $[t_0 = \texttt{0}, t_n = \texttt{3.1558149E7 sec}]$ with TOL = `1e-4`, h_{min} = `0.01` and h_{max} = `1e6`. Use the "VectorTrajectoryToFile" component to save your results as text file. To visualize your results, use the `VRL-Solar-System-Viewer.jar`. Compare the location of the earth at t_0 and t_n. Since the specified time-span is exactly one earth orbit, this is a good estimation for the global error of your simulation.

(b) Change the simulation interval to $[t_0 = \texttt{0}, t_n = \texttt{1 year}]$ and compare your results with the numbers published by NASA. Specify the global error.

(c) Simulate the time intervals $[t_0 = \texttt{0}, t_n = \texttt{10 years}]$ and $[t_0 = \texttt{0}, t_n = \texttt{100 years}]$. Please note that on older machines the computation can take a while. It is adviced to filter the solution trajectory, e.g., only add every n-th value to the solution. For the last interval, it is allowed to increase the TOL value to `1e-2` if your computer is too slow. If you do so, specify the TOL value that has been used for the computation.

Aufgabe 2 (3 Points)
Design and simulate your own solar system with at least 3 bodies.

Remark: Send your implemented source code as VRL-Studio project (.vrlp file) and the answers to the questions as plain text in an email. Append the pdfs produced with the TrajectoryPlotter to the email.
Send your solution to `practical.sim1@gcsc.uni-frankfurt.de` until Monday, 10.12.2018, 16h.