ssnbody

A Nbody Package

Document Version 1.28

Prepared Using LaTeX

by

Stefan Spännare
E-mail: stefan@spaennare.se
July 28, 2008

Contents

1 Introduction

This is a users guide to the nbody package *ssnbody.* It is not necessary to understand how the program works just to run it, but some knowledge in astronomy, mathematics and numerical methods makes it easier. The program makes a non relativistic (Newtonian) or a relativistic (PPN) integration of n bodies (point masses). The computation time is proportional to n^2, which makes the program useful only up to a few hundred bodies on a reasonably fast workstation or PC. On the other hand the integration of the bodies is made with high accuracy using 4-, 5- or 8-order Runge-Kutta or Bulirsch-Stoer integration. The initial data for the bodies are usually stored in a ascii file, but the program also has a simple function for generating random spherical distributions of bodies. The program package is available for Unix, Linux and DOS under Windows 98/2000/XP. Some auxiliary programs and plot programs to make nbody plots are also available. Several easy to use examples are also included in the package.

There are many reasons for using nbody programs for scientific purposes, interest and fun. In astronomy it can be of interest to integrate systems with few bodies i.e. double or multiple stars, the Solar system and up to whole galaxy simulations with many thousand bodies. The program package *ssnbody* is useful for the first kind of simulations when the number of bodies is relatively small.

Many other software packages for nbody simulations of different kinds exist, not at least on Internet (see the references).

2 Disclaimer

For all the programs in the *ssnbody* package the following statement is valid:

I make no warranties that this program is (1) free of error, (2) consistent with any standard merchantability, or (3) meeting the requirements of a particular application. This software shall not, partly or as a whole, participate in a process, whose outcome can result in injury to a person or loss of property. It is solely designed for analytical work. Permission to use, copy, and distribute is hereby granted without fee, providing that the header above including this notice appears in all copies.

If you have problems or suggestions to improvements or changes, please send comments to:

E-mail: stefan@spaennare.se

3 About the package

3.1 General information

The *ssnbody* package was made by the author in the spare time as a hobby project. A simple version of the nbody program was made already in February 1998 but most of the programming work was made during January and February 2000. But the code has been somewhat improved since then. Most of the documentation has also been written later. The latest updates and additions to the package and this documentation are from September and October 2006.

The nbody package is available with executables (and source) for Linux and Windows 98/2000/XP. The package can also be used on Unix computers but then the C-programs must be recompiled. See section 6 how to compile the programs.

If you are not familiar with nbody simulations it is recommended that you run the easy to use examples in section 5. It it also a good idea to read section 3.2 below.

3.2 Directories and files

The *ssnbody* package consists of the files ”ssnbody.tar.gz” (executables, source and examples for Linux), ”ssnbody.zip” (executables, source and examples for Windows 98/2000/XP) and this documentation ”ssnbody.pdf” and ”ssnbody.ps” as separate files.

The home page of the author, the *ssnbody* nbody package and this document are found on the web-pages:

http://www.spaennare.se/index.html

http://www.spaennare.se/ssnbody.html

Unpack the ”ssnbody.tar.gz” or ”ssnbody.zip” file whichever is appropriate. The `SSNBODY` directory then has the following contents (under directories):

`EXE`: executables and source for all the C-programs. Scripts to compile the programs are also present here.

`SRC`: source for all the C-programs (the same as in the `EXE` directory).

`DOC`: contains a very short ”readme.txt” file with package version and a link to the web-page of the nbody package and this documentation. A short package history is also given here.

`EXAMPLES`: contains files for the easy to use examples. The under directory `ALL` contains all the example files in one directory.

`BENCH`: contains files for the benchmark test.

`JPL`: extra JPL data for Solar system simulations. Note, these files come with no documentation.

4 Description of the C-programs

Here follows a description of the input and output parameters for all the C-programs included in this package. How to use these programs see also the examples in section 5.

The program names have at maximum 8 characters (plus 3 extension characters) to be easy to use with DOS under Windows.

Input parameters ending with "i" are of integer type and parameters ending with "r" are of real type (i.e. double precision floating point type).

Note, the programs make no check that the input parameters have appropriate values.

The C-code of the nbody program itself has some comments, but not the other programs (except for the program headers).

4.1 The nbody program

This nbody program makes a non relativistic (Newtonian) or a relativistic (PPN) numerical integration of n bodies with various positions, velocities and masses. The program can either use spherical distributed random data or data from a file. After each calculation data for the simulation (the bodies) is saved in a binary file that can be used later to continue the calculation (the -append function). The calculation time is proportional to n^2. The time step can automatically be adapted to a smaller value if some bodies are very close to each other. The program use 4-, 5- or 8-order Runge-Kutta integration or Bulirsch-Stoer integration. The program (nbody) use double precision (64 bit) floating point numbers but is also available (nbodyl) with long double (80 bit) floating point numbers on Intel compatible (x86) computers. Long double is activated by setting the LDOUBLE variable to 1 in the C-code and recompile the program.

If you type the name of the program the usage appears:

```
Usage 1: nbody algmodei relmodei barymodei printmodei
               seedi ni tmaxr tpr dtr dtminr gr mfrr cr
               rmaxr vmaxr mmaxr mminr outfile outfile.bin

Usage 2: nbody -file infile outfile outfile.bin

               In file parameters:
               algmodei relmodei barymodei printmodei
               ni tmaxr tpr dtr dtminr gr mfrr cr
               x(i) y(i) z(i) vx(i) vy(i) vz(i) m(i)

Usage 3: nbody -append outfile infile.bin tmaxr

algmode: (0/4/5/8); BS 0; RK4 4; RK5 5; RK8 8.

relmode: (0/1); Newtonian 0; Relativistic (PPN) 1.

barymode: (0/1); Not subtract barycentrum 0; Subtract barycentrum 1.

printmode: (0/1/2); Plot mode 0; Verify mode 1; Extended mode 2.
```

Parameter description (usage 1):

`algmode`: 0 Bulirsch-Stoer and 4, 5, and 8 Runge-Kutta of 4-, 5- and 8-order respectively.

`relmode`: 0 Newtonian calculation and 1 relativistic (PPN) calculation.

`barymode`: 0 not subtract barycentrum and 1 subtract barycentrum.

`printmode`: 0 plot mode, 1 verify mode and 2 extended mode. Plot mode prints positions and masses for each output step `tp` with about float precision. Verify mode prints positions, velocities and masses only for the last time step `tmax` with full machine precision (i.e. double or long double). Extended mode prints positions, velocities and masses for each output step `tp` with full machine precision (i.e. double or long double).

`seed`: integer random seed.

`n`: total number of bodies.

`tmax`: total integration time.

`tp`: time between print to output.

`dt`: time step. If `dt` is negative the integration is made backwards in time. Note, `tmax` and `dtmin` must not be negative.

`dtmin`: minimum possible time step. If `dtmin` = `dt` all time steps have equal size (except for the last one in some cases).

`g`: gravitational constant.

`mfr`: minimum force radius (usually `mfr` = 0.0).

`cr`: velocity of light (only valid for relativistic calculations but must be present).

`rmax`: maximum distance from the origin (0,0) for the bodies in a spherical distribution.

`vmax`: maximum velocity for the bodies.

`mmax`: maximum body mass.

`mmin`: minimum body mass.

`outfile`: output text file containing the x-, y- and z-positions and mass of the bodies for each print step. If `printmode` is 1 the x-, y- and z-positions and x-, y- and z-velocities and masses are printed with high accuracy only for the last time step.

`outfile.bin`: binary output file that is used if you want to continue the calculation later (using the -append function).

Parameter description (usage 2):

`infile`: input file containing `algmode, relmode, barymode, printmode, n, tmax, tp, dt, dtmin, g, mfr, c` and the initial position, velocity and mass for each body.

`outfile`: output text file containing the x-, y- and z-positions and mass of the bodies for each print step. If `printmode` is 1 the x-, y- and z-positions and x-, y- and z-velocities and masses are printed with high accuracy only for the last time step.

`outfile.bin`: binary output file that is used if you want to continue the calculation later (using the -append function).

Parameter description (usage 3):

`outfile`: output text file containing the x-, y- and z-positions and mass of the bodies for each print step. If `printmode` is 1 the x-, y- and z-positions and x-, y- and z-velocities and masses are printed with high accuracy only for the last time step.

`infile.bin`: binary file from an earlier calculation.

`tmax`: new maximum integration time (from time zero).

4.2 Auxiliary programs

How to use these programs see also the examples in section 5.

4.2.1 jd

This program calculates the Julian day. This is useful to for example calculate the number of days between two dates for Solar system integration. Just subtract the two Julian days from each other to get the number of days.

If you type the name of the program the usage appears:

```
Usage: jd yeari monthi dayi
```

Parameter description:

`year`: year of date.

`month`: month of date.

`day`: day of date.

4.2.2 getone

This program is useful to pick plot data (coordinates and mass) for a single body from a plot output with many bodies from the nbody program.

If you type the name of the program the usage appears:

```
Usage: getone plot-infile plot-outfile ii
```

Parameter description:

`plot-infile`: input plot file with `n` bodies.

`plot-outfile`: output plot file with only one body.

`i`: index (`i` $\leq$ `n`) for the selected body in the input file (with `n` bodies).

4.2.3 suncomp

This program is used to compare Solar system integration data from the nbody program with data integrated by JPL (for example DE200 and DE403 reference systems). The output from the program are the errors in position and velocity given in au, m, au/day and m/day for all the integrated bodies (in the JPL file). The output from the nbody program must be printed in verify mode output. Initial data for the nbody integration is taken from JPL data as well. See the Internet references for more information about JPL Solar system data.

If you type the name of the program the usage appears:

```
Usage: suncomp solsyst-reference-infile solsyst-verify-infile
```

Parameter description:

`solsyst-reference-infile`: JPL Solar system data from a certain date and reference system.

`solsyst-verify-infile`: Solar system verify data integrated with the nbody program to the same date.

4.2.4 oscel

This program calculates the osculating orbital elements of the Solar system from a certain point of time after a run with the nbody program. The output from the nbody program must be printed in verify mode output.

```
Usage: oscel solsyst-verify-infile
```

Parameter description:

`solsyst-verify-infile`: Solar system verify data integrated with the nbody program to a certain point of time.

4.2.5 eq2eclp and eq2eclv

These programs transform the coordinates from a nbody plot or verify mode file from equatorial to ecliptic coordinates at a certain point of time (Julian day or year, month and day). Useful to plot nbody calculations with the Solar system viewed in the ecliptic plane. JPL data are usually given in equatorial coordinates.

```
Usage 1: eq2eclp eq-plot-infile ecl-plot-outfile julian-dayr

Usage 2: eq2eclp eq-plot-infile ecl-plot-outfile -date yeari monthi dayi
```

```
Usage 1: eq2eclv eq-verify-infile ecl-verify-outfile julian-dayr

Usage 2: eq2eclv eq-verify-infile ecl-verify-outfile -date yeari monthi dayi
```

Parameter description:

`eq-plot-infile`: Solar system plot data integrated with the nbody program in equatorial coordinates.

`ecl-plot-outfile`: Solar system plot data transformed to ecliptic coordinates at a given time.

`eq-verify-infile`: Solar system verify data integrated with the nbody program in equatorial coordinates.

`ecl-verify-outfile`: Solar system verify data transformed to ecliptic coordinates at a given time.

`julian-day`: Julian day for the transformation.

`-date year month day`: date for the transformation.

4.2.6 ecl2eqp and ecl2eqvv

These programs transform the coordinates from a nbody plot or verify mode file from ecliptic to equatorial coordinates at a certain point of time (Julian day or year, month and day).

```
Usage 1: ecl2eqp ecl-plot-infile eq-plot-outfile julian-dayr

Usage 2: ecl2eqp ecl-plot-infile eq-plot-outfile -date yeari monthi dayi
```

```
Usage 1: ecl2eqv ecl-verify-infile eq-verify-outfile julian-dayr

Usage 2: ecl2eqv ecl-verify-infile eq-verify-outfile -date yeari monthi dayi
```

Parameter description:

`ecl-plot-infile`: Solar system plot data integrated with the nbody program in ecliptic coordinates.

`eq-plot-outfile`: Solar system plot data transformed to equatorial coordinates at a given time.

`ecl-verify-infile`: Solar system verify data integrated with the nbody program in ecliptic coordinates.

`eq-verify-outfile`: Solar system verify data transformed to equatorial coordinates at a given time.

`julian-day`: Julian day for the transformation.

`-date year month day`: date for the transformation.

4.2.7 pparam

This program prints data (body masses, semi major axes of orbit and orbital period) for a double pulsar data file (input file to the nbody program). Note, the first parameter line must be present in the file.

If you type the name of the program the usage appears:

```
Usage: pparam pulsar-infile
```

Parameter description:

`pulsar-infile`: pulsar data file (input file to the nbody program).

4.2.8 galaxy

This program generates input data to the nbody program to make simple ”galaxy” simulations. The ”galaxies” can then interact and collide. Note, these ”galaxies” consists of a few hundred simple point masses and have very little to do with real galaxies. But still it is interesting to see the result. The ”galaxies” are two-dimensional and present in the x- and y-plane. To make the ”galaxies” stable most of the mass is concentrated in the central body. To run the program a small input file must be compiled. Note, the first parameter line (to the nbody program) must be added separately to the output file.

If you type the name of the program the usage appears:

```
Usage: galaxy infile outfile

Infile parameters:
       gr
       r1r nr1i m1r mf1r x1r y1r vx1r vy1r
       r2r nr2i m2r mf2r x2r y2r vx2r vy2r
                     ...
       rnr nrni mnr mfnr xnr ynr vxnr vynr
```

Parameter description:

`g`: gravitational constant.

`r`: max radius of the galaxy.

`nr`: number of rings (at different radii) in the galaxy.

`m`: total mass of the galaxy.

`mf`: mass-factor. This variable tells how much mass is concentrated in the central body and the bodies at different radii. Appropriate values are 4.0, 5.0 or larger.

`x`: x-coordinate for the center of the galaxy.

`y`: y-coordinate for the center of the galaxy.

`vx`: x-velocity for the center of the galaxy.

`vy`: y-velocity for the center of the galaxy.

Input data file example (ex1.gal):

```
100.0
100.0 6 1000.0 5.0  200.0 0.0 0.0   5.0
 60.0 4  500.0 4.0 -200.0 0.0 0.0 -15.0
```

4.2.9 ring

This program generates input data to the nbody program to generate one or many rotating rings of bodies. The rings are not real astronomical objects but still interesting gravitational constellations. The rings are quite unstable and usually fall apart after a while. The rings are two-dimensional and present in the x- and y-plane. To run the program a small input file must be compiled. Note, the first parameter line (to the nbody program) must be added separately to the output file.

If you type the name of the program the usage appears:

```
Usage: ring infile outfile

Infile parameters:
       gr
       r1r nr1i m1r x1r y1r vx1r vy1r
       r2r nr2i m2r x2r y2r vx2r vy2r
                  ...
       rnr nrni mnr xnr ynr vxnr vynr
```

Parameter description:

`g`: gravitational constant.

`r`: radius of the ring.

`nr`: number of bodies in the ring.

`m`: total mass of the ring.

`x`: x-coordinate for the center of the ring.

`y`: y-coordinate for the center of the ring.

`vx`: x-velocity for the center of the ring.

`vy`: y-velocity for the center of the ring.

Input data file example (ex3.rng):

```
100.0
100.0 8 800.0  200.0 0.0 0.0   5.0
 60.0 8  400.0 -200.0 0.0 0.0 -15.0
```

4.3 Plot programs

How to use these programs see also the examples in section 5.

4.3.1 nbplotl and nbplotd

Here follows a description of the plot programs nbplotl and nbplotd. The names means nbody-plot-line and nbody-plot-dot. The graphics routines use the ALLEGRO programming library (see the Internet references). The programs can be compiled for both Unix/Linux and Windows 98/2000/XP (see section 6 for more details).

The resolution of the plot window, DOS (console) or windows mode, save image flag and background and border colours are given in the files "nbplotl.cfg" and "nbplotd.cfg" (see these files), that must be present in the same directory as the plot executables. Note, the plot window resolution must be somewhat smaller than the screen resolution to give space for the window border (in window mode). There is no check for this. Default plot resolution is 640×480 but can easily be changed by editing the "nbplotl.cfg" and "nbplotd.cfg" files.

Whenever a key is pressed the plot program is terminated and the plot is written to the image file "nbplotl.bmp" or "nbplotd.bmp" (Windows bitmaps) if the save image flag is set. To save the image copy the file to another name before doing the next plot.

If you type the name of the program the usage appears:

```
Usage: nbplotl plot-infile(x y z m) x1r y1r x2r y2r delayi line-widthi xyz-flagi cmodi

xyz-flag = 1 => x y
xyz-flag = 2 => x z
xyz-flag = 3 => y z
```

```
Usage: nbplotd plot-infile(x y z m) x1r y1r x2r y2r delayi line-widthi xyz-flagi cmodi

xyz-flag = 1 => x y
xyz-flag = 2 => x z
xyz-flag = 3 => y z
```

Parameter description:

`plot-file(x y z m)`: plot file from the nbody program (plot mode output).

`x1 y1 x2 y2`: coordinates of the corner of the plot (in the nbody data reference system).

`delay`: integer plot speed delay (depends on the speed of the computer).

`line-width`: the width (size) of the dots in pixels for nbplotd. The line width is always 1 for nbplotl.

`xyz-flag`: tells the plot program which coordinates to be plotted (1: x y, 2: x z and 3: y z).

`cmod`: color-mode. Tells the plot program (nbplotl) how many colors that are used for the lines. Maximum number of colors is 14. If `cmod` > 14 the number of line colors will still be 14 cyclic repeated. The dots are always black for nbplotd (independent of `cmod`).

5 Some easy to use examples

Here follows a description of some easy to use examples how to use the nbody program and the plot programs.

The examples are run as executable "bat" (*.bat) files that works both for Unix and Linux and DOS (command line interpreter) under Windows 98/2000/XP. Under Windows they can also be run by double clicking on the ”bat” file icons. Some of the examples also use auxiliary data (*.dat) and other files. First copy all the files under `/EXAMPLES/ALL` to the `EXE` directory. Under Windows copy and paste can be used. Under Unix and Linux it is then wise to make sure that the ”bat” files are executable.

```
>chmod +x *.bat
```

The ”bat” files beginning with ”do” (do*.bat) runs the nbody program itself generating a file (*.plo) that can be used by the plot files. ”Bat” files beginning with ”pl” (pl*.bat) means ”plot line” and ”bat” files beginning with ”pd” (pd*.bat) means ”plot dot”. The plot windows resolution is by default 640 $\times$ 480 pixels. This can be changed by editing the ”nbplotl.cfg” and ”nbplotd.cfg” files in the `EXE` directory. Note, the plot window resolution must be somewhat smaller than the screen resolution to give space for the plot window border.

Whenever a key is pressed the plot program is terminated and the plot is written to the image file ”nbplotl.bmp” or ”nbplotd.bmp” (Windows bitmaps). To save the image copy the file to another name before doing the next plot.

Note, in these examples the nbody program generates files (*.plo) up to 35 Mbyte in size. The nbody calculation time is less than 5 minutes per example on an Intel Pentium III computer at 1000 MHz. The only exception is ”do100a.bat” that takes about 30 minutes to calculate on such a computer.

The ”bat” and ”dat” files can easily be edited and modified. It is for example interesting to change the random seed for the examples in section 5.1 below. The delay in the plot ”bat” files depends on the speed of the computer and must perhaps be adjusted. The default delay values works well with a computer comparable to an Intel Pentium III computer at 1000 MHz. In all the examples Runge-Kutta 8-order integration is used, but this can be changed to Runge-Kutta 4-, 5-order or Bulirsch-Stoer. For Bulirsch-Stoer the same time steps can be used as for Runge-Kutta 8-order, but Runge-Kutta 4- and 5-order must have smaller time steps for the same calculation accuracy. It is also interesting to change the subtract or not subtract barycentrum or use the long double (nbodyl) version of the program.

5.1 Random spherical distribution of bodies

1. A unusually stable 3-body system. Newtonian integration.

```
>do3a.bat
>pl3a.bat
>pd3a.bat
```

2. A 10-body system. Newtonian integration.

```
>do10a.bat
>pl10a.bat
>pd10a.bat
```

3. A 20-body system. Newtonian integration.

```
>do20a.bat
>pl20a.bat
>pd20a.bat
```

4. A 100-body system. Newtonian integration.

```
>do100a.bat
>pl100a.bat
>pd10i0a.bat
```

5.2 The Solar system

5. This is a relativistic (PPN) integration of the Solar system (the Sun, Moon, planets and the 5 large asteroids Ceres, Pallas, Vesta, Iris and Bamberga) from year 1980 January 1 to 2000 January 1. The coordinates and masses for the Sun, Moon and planets come from JPL DE403. The coordinates (here given in equatorial coordinates) and some of the masses of the asteroids comes from the JPL HORIZON program. See section 8 and the references for more details. The plots show the Solar system out to 4 au so that Jupiter's orbit is partly seen.

```
>dosma80.bat
>plsma80.bat
>pdsma80.bat
```

6. The same calculation as in 5 but in ecliptic coordinates and then plot it. The ecliptic plane is the plane of the Earth's orbit (and approximately the other planets) around the Sun.

```
>dosma80ecl.bat
>plsma80ecl.bat
>pdsma80ecl.bat
```

7. This example shows the difference (errors) between this Solar system integration and the one made by JPL DE403 during the same period (1980 January 1 to 2000 January 1). Note, the errors are not between this calculation and the true position of the bodies in space although the difference is small. The position errors after the integration are very small, only about 107 m for Mercury and a few kilometers for the outer planets. The Earth and especially the Moon have large errors due to the close interaction between them. The errors for the asteroids are a few hundred kilometers. This calculation is made in relativistic (PPN) mode and verify mode output. It is instructive to also make the calculation in Newtonian integration mode. Then the errors will be much larger (especially for the inner planets). It is also interesting to see if smaller time steps or the long double version of the program (nbodyl) gives better accuracy. The asteroids can also be removed from the calculation to see how the errors are affected.

In the calculations made by JPL for DE403 the asteroids were not included in the integration as separate bodies, but calculated by Chebyshev expansions that approximate fixed Keplerian ellipses. This is the main reasons why the results for the solar system calculated with this package differ somewhat from the JPL results. The coordinates for the asteroids (and some of the masses) used here comes from the JPL HORIZON program.

Note, this example must be run in a DOS (command line interpreter) window under Windows 98/2000/XP.

```
>doma80eclv.bat
>suncomp sma00ecl.403 dosma80eclv.plo
```

Output results. Absolute errors after 20 years integration (from 1980 January 1 to 2000 January 1) for the Sun (0), planets (1 - 9), Moon (10) and the asteroids Ceres (11), Pallas (12), Vesta (13) Iris (14) and Bamberga (15) compared to JPL DE403 data (rotated to ecliptic coordinates) and JPL HORIZON data (ecliptic coordinates) for the asteroids.

```
 nr             (au)               (m)          (au/day)          (m/day)
  0    2.434513086749e-10        36.42    4.789719730155e-13        0.07
  1    7.116664253401e-10       106.46    3.437501520478e-11        5.14
  2    1.203567683733e-09       180.05    2.772804224705e-11        4.15
  3    6.876603615327e-08     10287.25    7.307597718856e-09     1093.20
  4    9.539889073696e-09      1427.15    1.035786635776e-10       15.50
  5    1.850910443841e-08      2768.92    2.091691361180e-11        3.13
  6    2.031023426651e-08      3038.37    1.113383055705e-11        1.67
  7    9.346292477662e-09      1398.19    2.546589393031e-12        0.38
  8    6.481580387422e-09       969.63    1.209751054162e-12        0.18
  9    5.141398540351e-09       769.14    7.362389155991e-13        0.11
 10    2.560869930893e-06    383100.69    5.611442213555e-07    83945.98
 11    5.118516583525e-07     76571.92    1.870301034861e-09      279.79
 12    1.262798953993e-06    188912.03    6.778735173771e-09     1014.08
 13    2.108745624800e-06    315463.86    9.068147111504e-09     1356.58
 14    7.804285339982e-07    116750.45    3.224116625316e-09      482.32
 15    6.916705822861e-06   1034724.46    2.971398823673e-08     4445.15
```

5.3 Double pulsars

8. A double pulsar. Relativistic (PPN) integration. The mass of each pulsar body is about 101 M_{Sun}. This is much higher than for real pulsars to enhance the relativistic periastron rotation.

```
>dopuls1.bat
>plpuls1.bat
>pdpuls1.bat
```

9. A pulsar with a smaller orbiting body. Relativistic (PPN) integration. The masses of the bodies are about 101 M_{Sun} and 0.101 M_{Sun}. The first value is much higher than for real pulsars to enhance the relativistic periastron rotation.

```
>dopuls2.bat
>plpuls2.bat
>pdpuls2.bat
```

5.4 "Galaxies" and rings

10. Two interacting "galaxies". Newtonian integration. Note, these "galaxies" have very little to do with real galaxies, but it is still interesting to see the result.

```
>dogal1.bat
>pdgal1.bat
```

11. A rotating 32-body ring. Very unstable. Newtonian integration.

```
>doring1.bat
>pdring1.bat
```

12. A rotating 8-body ring. Very unstable. Newtonian integration.

```
>doring2.bat
>pdring2.bat
```

13. Two rotating and interacting 8-body rings. Very unstable. Newtonian integration.

```
>doring3.bat
>pdring3.bat
```

6 How to compile the C-programs

6.1 Compiling under Unix and Linux

Note, executables for Linux are included in the file "ssnbody.tar.gz".

To compile all the C-programs (except the plot programs) under Unix or recompile the programs under Linux the `compall` script can be used. It is assumed that the GNU gcc compiler is used. If another compiler is used the script must perhaps be edited somewhat. The script and source files are located in the `EXE` directory. The expression `chmod +x` makes the target file executable. If problems occur to compile the long double version of the nbody program (nbodyl), simply remove it from the `compall` script. Note, fast long double functions are only supported on Intel compatible (x86) computers.

```
>chmod +x compall
>compall
```

To compile the plot programs (nbplotl and nbplotd) the ALLEGRO game programming library must first be compiled and installed (see the Internet references). Follow the instructions in the ALLEGRO package documentation for Unix and Linux (/docs/build/unix.txt). Important note, when running the "configure" script for ALLEGRO the following option must be used.

```
>./configure --enable-static
```

Then the plot programs can then be compiled by running the following script (located in the `EXE` directory).

```
>chmod +x compplot
>compplot
```

6.2 Compiling under Windows 98/2000/XP

Note, executables for Windows 98/2000/XP are included in the file ”ssnbody.zip”.

To recompile all the C-programs (except the plot programs) under Windows 98/2000/XP the `compall.bat` script can be used. It is assumed that the GNU DJGPP gcc compiler for DOS under Windows is used (see the Internet references). The programs can also be compiled using the Microsoft Visual C++ 6.0 compiler but then the code will be slower. If another compiler than DJGPP is used the script must perhaps be edited somewhat. The script and source files are located in the `EXE` directory. If problems occur to compile the long double version of the nbody program (nbodyl), simply remove it from the `compall.bat` script. Note, fast long double functions are only supported on Intel compatible (x86) computers.

Note, under Windows just double click on the `compall.bat` icon to run the script or run the script at the DOS prompt (command line interpreter under Windows 2000/XP).

```
>compall.bat
```

To compile the plot programs (nbplotl and nbplotd) the ALLEGRO game programming library must first be compiled and installed (see the Internet references). Follow the instructions in the ALLEGRO package documentation for Windows 98/2000/XP (/docs/build/msvc.txt). The Microsoft Visual C++ 6.0 compiler must then be used to allow graphics. Important note, the files ”alleg.lib” and ”alleg40.dll” from the ALLEGRO package must be present in the same directory as the plot program source code when the programs are compiled and run.

Then the plot programs can then be compiled by running the following script (located in the `EXE` directory). Note, under Windows just double click on the `compplot.bat` icon to run the script or run the script at the DOS prompt (command line interpreter under Windows 2000/XP).

```
>compplot.bat
```

7 Algorithms

This nbody program makes non a relativistic (Newtonian) or a relativistic Parametrized Post Newtonian (PPN) numerical integration of n bodies with various positions, velocities and masses. The integration method is Runge-Kutta 4-, 5- or 8-order or Bulirsch-Stoer.

7.1 Calculation of Newtonian acceleration

The non relativistic (Newtonian) acceleration on body i is given by

$$\ddot{\boldsymbol{r}}_{i_{\text{pointmass}}} = \sum_{j \neq i} \frac{\mu_j(\boldsymbol{r}_j - \boldsymbol{r}_i)}{r_{ij}^3}$$

where $\mu_j = Gm_j$.

The position and velocity for barycentrum is calculated from

$$\sum_i \mu_i \boldsymbol{r}_i = \mathbf{0}$$

$$\sum_i \mu_i \boldsymbol{v}_i = \mathbf{0}$$

7.2 Calculation of relativistic (PPN) acceleration

The relativistic (PPN) acceleration on body i is given by

$$\ddot{\boldsymbol{r}}_{i_{\text{pointmass}}} = \sum_{j \neq i} \frac{\mu_j(\boldsymbol{r}_j - \boldsymbol{r}_i)}{r_{ij}^3} \Bigg\{ 1 - \frac{2(\beta+\gamma)}{c^2} \sum_{k \neq i} \frac{\mu_k}{r_{ik}} - \frac{2\beta - 1}{c^2} \sum_{k \neq j} \frac{\mu_k}{r_{jk}} + \gamma \left(\frac{v_i}{c}\right)^2 + (1+\gamma)\left(\frac{v_j}{c}\right)^2$$

$$- \frac{2(1+\gamma)}{c^2} \dot{\boldsymbol{r}}_i \cdot \dot{\boldsymbol{r}}_j - \frac{3}{2c^2}\left[\frac{(\boldsymbol{r}_i - \boldsymbol{r}_j)\cdot \dot{\boldsymbol{r}}}{r_{ij}}\right]^2 + \frac{1}{2c^2}(\boldsymbol{r}_j - \boldsymbol{r}_i)\cdot \ddot{\boldsymbol{r}}_j \Bigg\}$$

$$+ \frac{1}{c^2} \sum_{j \neq i} \frac{\mu_j}{r_{ij}^3} \{[\boldsymbol{r}_i - \boldsymbol{r}_j] \cdot [(2+2\gamma)\dot{\boldsymbol{r}}_i - (1+2\gamma)\dot{\boldsymbol{r}}_j]\}(\dot{\boldsymbol{r}}_i - \dot{\boldsymbol{r}}_j) + \frac{3+4\gamma}{2c^2} \sum_{j \neq i} \frac{\mu_j \ddot{\boldsymbol{r}}_j}{r_{ij}}$$

where $\boldsymbol{r}_i$, $\dot{\boldsymbol{r}}_i$, $\ddot{\boldsymbol{r}}_i$ are the barycentric position, velocity and acceleration of body i. The quantity $\mu_j = Gm_j$ where G is the gravitational constant and m_j is the mass of body j. $r_{ij} =\mid \boldsymbol{r}_j - \boldsymbol{r}_i \mid$. β is the PPN parameter measuring the non linearity in superposition of gravity and γ is the PPN parameter measuring space curvature produced per unit rest mass. In this integration (as in general relativity) $\beta = \gamma = 1$, $v_i =\mid \dot{\boldsymbol{r}}_i \mid$ and c is the velocity of light. The quantity $\ddot{\boldsymbol{r}}_j$ appearing in two terms on the right hand side denotes the barycentric acceleration of each body j due to Newtonian effects of the remaining bodies.

The position and velocity for the relativistic (PPN) barycentrum is calculated from

$$\sum_i \mu_i^* \boldsymbol{r}_i = \mathbf{0}$$

$$\sum_i \mu_i^* \boldsymbol{v}_i = \mathbf{0}$$

where

$$\mu_i^* = \mu_i \left\{ 1 + \frac{1}{2c^2} v_i^2 - \frac{1}{2c^2} \sum_{j \neq i} \frac{\mu_j}{r_{ij}} \right\}$$

7.3 Integration methods

The nbody program use 4-, 5- or 8-order Runge-Kutta or Bulirsch-Stoer integration. This means that the error is of order $O(h^5)$, $O(h^6)$ and $O(h^9)$ respectively, where h is the time step. The error for Bulirsch-Stoer integration is of high order. Here x_i, $\dot{x}_i$ and $\ddot{x}_i$ denotes the position, velocity and acceleration in x. The calculations are performed in similar way for y and z. Here is only a description of the 4-order Runge-Kutta with first order derivatives. In the nbody program the velocities are integrated from the accelerations and the positions from the velocities for each time step. See the references for descriptions of higher order Runge-Kutta and Bulirsch-Stoer algorithms.

Runge-Kutta 4-order (integration of position):

$$h = dt$$

$$A = h \cdot \dot{x}(x_i, t_i)$$

$$B = h \cdot \dot{x}(x_i + \frac{1}{2}A, t_i + \frac{1}{2}h)$$

$$C = h \cdot \dot{x}(x_i + \frac{1}{2}B, t_i + \frac{1}{2}h)$$

$$D = h \cdot \dot{x}(x_i + C, t_i + h)$$

$$x_{i+1} = x_i + \frac{1}{6}(A + 2B + 2C + D)$$

Runge-Kutta 4-order (integration of velocity):

$$h = dt$$

$$A = h \cdot \ddot{x}(\dot{x}_i, t_i)$$

$$B = h \cdot \ddot{x}(\dot{x}_i + \frac{1}{2}A, t_i + \frac{1}{2}h)$$

$$C = h \cdot \ddot{x}(\dot{x}_i + \frac{1}{2}B, t_i + \frac{1}{2}h)$$

$$D = h \cdot \ddot{x}(\dot{x}_i + C, t_i + h)$$

$$\dot{x}_{i+1} = \dot{x}_i + \frac{1}{6}(A + 2B + 2C + D)$$

8 JPL Solar system data

8.1 Solar system coordinates

The coordinates and masses for the Sun, Moon and the planets in the examples included in this package are from the JPL DE200, DE403 and DE405 data (see the references). The asteroid coordinates are from the JPL HORIZON data which probably differ somewhat from the coordinate reference system for DE200, DE403 and DE405. Many different values of the asteroid masses can be found (see below).

In the calculations made by JPL (DE102, see the references) and also DE200, DE403 and DE405 the asteroids were not included in the integration as separate bodies, but calculated by Chebyshev expansions that approximate fixed Keplerian ellipses. These are the main reasons why the results for the solar system calculated with this package differ somewhat from the JPL results.

8.2 Asteroid masses

The five large asteroids Ceres, Pallas, Vesta, Iris and Bamberga should be included in the Solar system integration to get accurate results especially for the inner planets and the Solar system barycentrum. However many different values of the masses are available from Internet and other sources. Here are some examples:

Asteroid	Exp	JPL HORIZON	Ref. [5]	DE102 Article	DE403	DE405	Ref. [7]	Ref. [6]
Ceres	10^{-13}	1.561	1.746	1.746	1.373	1.391	1.287	1.294
Pallas	10^{-14}	4.460	3.200	3.847	3.107	2.959	4.735	4.729
Vesta	10^{-14}	4.460	4.080	3.551	3.965	3.847	4.498	4.461
Iris	10^{-15}	—	1.600	1.580	—	—	—	—
Bamberga	10^{-15}	—	2.600	2.576	—	—	—	—

Ref. [5]: de118i.zip. Solar system integration program. See the Internet references.

Ref. [6]: "Asteroid Fact Sheet". Are the most resent and probably also most accurate asteroid data results on Internet. See the Internet references.

Ref. [7]: A relatively recent article on Internet by J. L. Hilton. See the Internet references.

The asteroid masses m are given in $au^3 d^{-2}$. If $\mathrm{M_{Sun}} = 1.99 \cdot 10^{30}$ kg and $k^2 = 0.01720209895^2$ the masses m_{kg} can be calculated in kg from:

$$m_{\mathrm{kg}} = \frac{m \cdot \mathrm{M_{Sun}}}{k^2}$$

9 Some benchmark results

The benchmark examples below were run on an Intel Celeron computer at 1400 MHz. The computer has 256 kbyte L2 cache and 512 Mbyte primary memory (100 MHz SDRAM). Operating system was Red Hat 8.0 Linux with the GNU gcc 3.20 C-compiler. The calculations were made using verify mode output to avoid time consuming outputs. Results for both double and long double precision are shown. Below the mode (N) means Newtonian calculation and (R) Relativistic (PPN) calculation. Runge-Kutta 8-order integration was used for all the examples. Note, for the four first examples the calculation time can vary not only to precision itself but also due to different time steps depending on the effect of close encounters of bodies. The example numbers below refer to the numbers in section 5. The programs were compiled using the following options:

```
>gcc -O3 nbody nbody.c -lm

>gcc -O3 nbodyl nbodyl.c -lm
```

Benchmark results:

Nr	Example	Mode	Bodies	CPU-time (s) (double)	CPU-time (s) (long double)
1	do3a	N	3	29.41	48.65
2	do10a	N	10	74.21	65.57
3	do20a	N	20	230.95	219.09
4	do100a	N	100	1719.05	2276.76
5	dosma80	R	16	85.30	136.25
8	dopuls1	R	2	126.76	216.19
10	dogal1	N	170	98.63	130.50
11	doring1	N	32	49.44	68.65

10 References

10.1 Book and article reference

[**1**] Newhall X. X., Standish E. M. Jr., Williams J. G.: 1983, *Astron. Astrophys.* **125**, 150-167, *"DE 102: a numerically integrated ephemeris of the Moon and planets spanning forty-four centuries"*.

[**2**] "Practical Astronomy With Your Calculator", Third Edition, by Peter Duffett-Smith, 1988, Cambridge University Press.

10.2 Internet references

The home page of the author, the *ssnbody* nbody package and this document are found on the web-pages:

http://www.spaennare.se/index.html

http://www.spaennare.se/ssnbody.html

[**3**] Nbody / Particle Simulation Methods.

http://www.amara.com/papers/nbody.html

[**4**] JPL Solar System Dynamics.

http://ssd.jpl.nasa.gov/

[**5**] Astronomical and numerical software source codes. For example a Solar system integration program (de118i.zip).

http://www.moshier.net

[**6**] Asteroid Fact Sheet.

http://nssdc.gsfc.nasa.gov/planetary/factsheet/asteroidfact.html

[**7**] Hilton J. L., U.S. Naval Observatory, *"New Masses and Densities for 1 Ceres, 2 Pallas and 4 Vesta"*.

http://www.aas.org/publications/baas/v29n5/aas191/abs/S071006.html

[**8**] Numerical Recipes Home Page.

http://www.nr.com

[**9**] Ordinary Differential Equations.

http://csep1.phy.ornl.gov/ode/ode.html

[**10**] ode/rksuite; Runge-Kutta codes in Fortran.

http://www.netlib.org/ode/rksuite/

[11] ode; Ordinary differential equation solver codes in Fortran.

http://www.netlib.org/ode/index.html

[12] DJGPP. A C/C++ compiler for DOS under Windows 98/2000/XP.

http://www.delorie.com/djgpp/

[13] ALLEGRO. A game programming library.

http://sunsite.dk/allegro/wip.html