Sagan Summer School 2015:
GlobalPFE
A practical guide on how to run the model

Christoph Mordasini

PlanetsInTime Research Group, University of Bern, Switzerland

1. Introduction

This document describes how to install, compile, and run the GlobalPFE (Global Planet Formation and Evolution) model. At the 2015 Sagan Summer Workshop, the participants will find the code already installed and compiled on the Amazon Cloud machines. However, since the code will be publicly distributed after the workshop, in §2 and §3 below we provide the installations and compilation instructions. While advanced users are encouraged to modify and re-compile the code as they wish, *most participants can skip ahead to §4* - the instructions for running the model for a single planet.

2. Installation

GlobalPFE comes as a software package called GlobalPFE.tar.gz. From the server used for the Sagan Summer Workshop, obtain a copy of GlobalPFE.tar.gz and unpack it with

```
cp /ssw/GlobalPFE.tar.gz .
tar xvzf GlobalPFE.tar.gz
```

This generates a directory called "GlobalPSF" with two sub-directories, "exercises" and "sourceCode". The "sourceCode" directory contains all the fortran code, while "exercises" contains the input files and examples. This is also where we will run the code.

3. Compilation

To compile the code, the *gfortran* compiler and gnu *make* need to be installed.

Before compiling, first view the source code with

```
cd GlobalPFE/sourceCode
ls
```

This shows the fortran files and their Makefile:

```
Makefile         gasdev.f90       initdisk.f90     mdotgas.f90      stopsim.f90
calcmiso.f90     getmcdists.f90   initialize.f90   migration.f90    timestep.f90
driver.f90       getparams.f90    interpol.f90     popsynth.f90     units.h90
evodiskd.f90     globalPFE.f90    makeinitcond.f90 ran1.f90         writeout.f90
evodiskg.f90     initcond.f90     mdotcore.f90     rhills.f90
```

Each of these fortran files contains a subroutine with the same name.

To build the software from the provided Makefile, simply type

```
make
```

If everything works fine, you should see something like

```
gfortran -fimplicit-none -ffree-line-length-none -O3 -funroll-loops -ftree-vectorize
-ftree-loop-optimize -msse -msse2 -m3dnow -Wall  -c -w -fdefault-real-8
-fdefault-double-8      driver.f90
...
gfortran  -O3 -Wall -lstdc++ -finit-real=nan -o globalPFE  driver.o popsynth.o
getmcdists.o getparams.o makeinitcond.o ran1.o gasdev.o initcond.o initdisk.o
stopsim.o timestep.o initialize.o writeout.o evodiskg.o evodiskd.o mdotcore.o
mdotgas.o rhills.o calcmiso.o migration.o interpol.o globalPFE.o
```

This produces a binary file with the name *globalPFE.* To clean the directory and to compile everything again, type

```
make clean
```

Finally, copy the binary (*globalPFE*) into the directories where you will run through various example exercises

```
cp globalPFE ../exercises/singleplanet/
cp globalPFE ../exercises/systematic/
cp globalPFE ../exercises/popsynth/*
```

4. Simulation of a single planet

In the following, we will illustrate how to run globalPFE in three different cases: a simulation of an individual Jupiter-like planet (this section), a simulation of one Solar-System-like system (§5), and finally a simulation of 8 different population syntheses (§6).

From the "mondayTuesday_handsOn" directory, change into the sub-directory for the single planet calculation with

```
cd exercises/singleplanet/
```

Here you will find the following files

```
Jupiter_aM.eps Jupiter_aM.gnu Jupiter_tM.eps
Jupiter_tM.gnu diskSigma.eps diskSigma.gnu
globalPFE.in paramsPFE.in globalPFE
```

These are 1) the gnuplot scripts (.gnu) to generate images (.eps), 2) the two input files (.in), and 3) the binary executable.

globalPFE.in is the input file with the physical initial conditions (disk and stellar properties). The contents of this file are shown here, with the nominal values values for this exercise in parentheses.

```
1.0              !Stellar mass in Msol  (1.0)
0.03             !Inner disk radius in AU (0.03)
30.0             !Outer disk radius in AU (30)
0.0              !Disk [Fe/H] (dex) (0)
5.0              !Scaling factor for gas surface density (5.0)
1.2              !disk dispersion timescale in Myr (1.2)
7.5              !planet initial distance in AU (7.5)
```

These are initial conditions for an embryo starting to form at an initial distance of 7.5 AU from a 1 Solar mass star in a disk with a surface density of gas equal to 5 times the MMSN (minimum mass solar nebula) surface density with a solar metallicity ([Fe/H]=0) that extends form 0.03 to 30 AU (exponential decrease outside). The disk has a characteristic dispersal timescale of 1.2 Myr (this is the e-folding timescale; the total disk lifetime is longer). These initial conditions were chosen to lead to an approximately Jupiter-like planet.

The second input file is the *paramsPFE.in* file. It contains the parameters and settings for the model physics:

```
-0.5D0     Power law exponent for temperature in disk (-0.5D0)
1D-3       Alpha viscosity in disk (1D-3)
0.0149D0   Solar dust to gas ratio (0.0149D0)
1D-2       Opacity in protoplanetary atmosphere [cm**2/g] (1D-2)
10.4D0     Power law TKH of planet 10**pKH*M**qKH pKH (10.4)
-1.5D0     Power law TKH of planet 10**pKH*M**qKH qKH (-1.5)
5          Mode of limiting gas accretion (5)
3          Mode of type I  migration (3)
4D-3       Scaling factor for type I rate  (4D-3)
2          Mode of type II  migration(2)
2D-3       Scaling factor for type II rate (2D-3)
1D0        Scaling factor for core accretion rate (1D0)
```

The values in parentheses are again the nominal values. More information is given below describing each of these parameters. In this simulation, the type I migration rate is in particular reduced by a factor 0.004 and the type II migration is reduced by 0.002.

Next we run the simulation.

```
./globalPFE
```

Once started, the program will ask

```
Mode of operation: single planet (1), systematic study (2), population synthesis (3)
1
```

We choose 1 (single planet), leading after a few seconds of calculation time to

```
 **************************************************************
 **************************************************************
       globalPFE: Global Planet Formation and Evolution Model
      Toymodel for the formation and evolution of (exo)planets
         (C) Christoph Mordasini, University of Bern, 2015
 **************************************************************
 **************************************************************
 Initial conditions
 Planet No                      1
 Mstar/Msol             1.0000000000000000
 Rindisk/AU             2.9999999999999999E-002
 Routdisk/AU            30.000000000000000
 Disk fg                5.0000000000000000
 Disk gas mass/Msol    0.21372576490919437
 Disk [Fe/H]            0.0000000000000000
 Disk solid mass/Me     991.81694129843731
 Disk taudisk/yr        1200000.0000000000
  Planet a/AU           7.5000000000000000
************************************
Simulation finished
Timestep                   8791
Time/yr                10008256868.111757
Planet mass/Mearth     318.86500894007287
Core mass/Mearth       65.918118927781592
Enve mass/Me           252.94689001229131
Planet a/AU            5.6247613162395114
************************************
```

Thus the planet has final core and envelope masses (at 10 Gyr which is the integration time we run) of about 66 and 252 Earth masses, respectively, similar to Jupiter. The final semi-major axis is 5.62 AU. The calculation required 8791 timesteps. You may get a warning like "The following floating-point exceptions are signaling: ..." but this can be ignored.

The simulation generated a file called *track_00001.dat* which contains the planet properties as a function of time (see §8.1 below for a description of each column). The program also generated a *diskevo.dat* file. It contains the disk properties (gas and solid surface density as function of distance and time).

We next look at the results using gnuplot. Open gnuplot and load in its input file with

```
gnuplot
gnuplot> load 'Jupiter_tM.gnu'
```

This will plot to the screen using X11. If we instead want to generate an eps file, we type (see in the *Jupiter_tM.gnu* file)

```
gnuplot> set term postscript eps enhanced color dashed lw 2 "Times-Roman" 20
gnuplot> set out 'Jupiter_tM.eps'
gnuplot> load 'Jupiter_tM.gnu'
```

In the resulting *Jupiter_tM.eps* figure (which should be ∼identical to the file which is included in the software package), we have the classical plot of the core, envelope, and total mass as a function of time (Figure 1). At about 3.5 Myr, gas runaway accretion occurs.

With the second gnuplot script *Jupiter_aM.gnu* we can generate in the same way the plot that shows how the planet grows and migrates in the distance-mass plane. This is called a planetary formation track. We see that the planet first migrates rapidly inward in type I migration (despite the strong reduction factor). When it transitions into type II

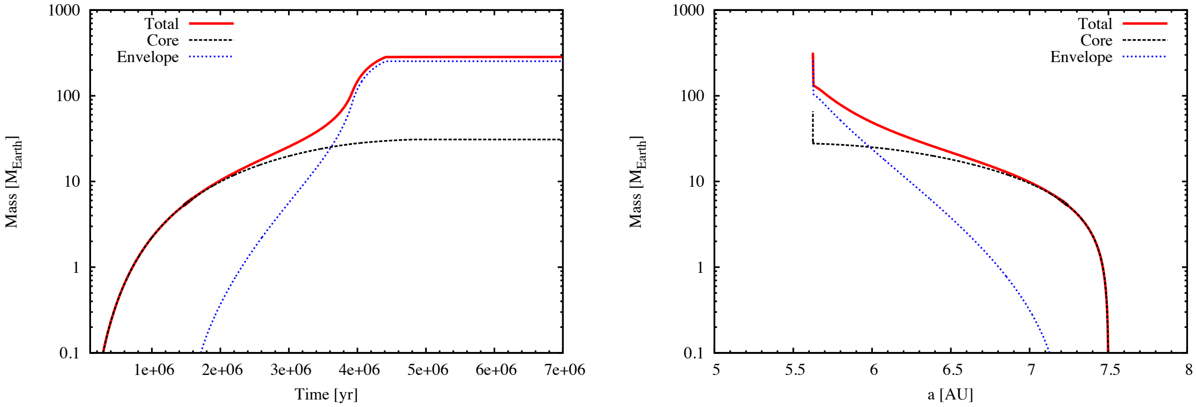

Fig. 1.— Core, envelope, and total mass as a function of time (left panel) and semi-major axis (right panel).

migration, its inward movement becomes much slower, and the planet track goes almost vertically upwards (but not exactly, it still migrates a little bit).

Finally we want to see how the gas surface density evolves as a function of time and distance. Using the *diskSigma.gnu* script we generate a plot showing the disk evolution (Figure 2). The color code gives the time in years. We see how the disk surface density decreases as it is accreted onto the star and photoevaporated. The disk also shrinks in size, due to photoevaporation. The radial slope varies as r^{-1}, because the temperature goes like $r^{-0.5}$ (as specified in the *paramsPFE.in* file).

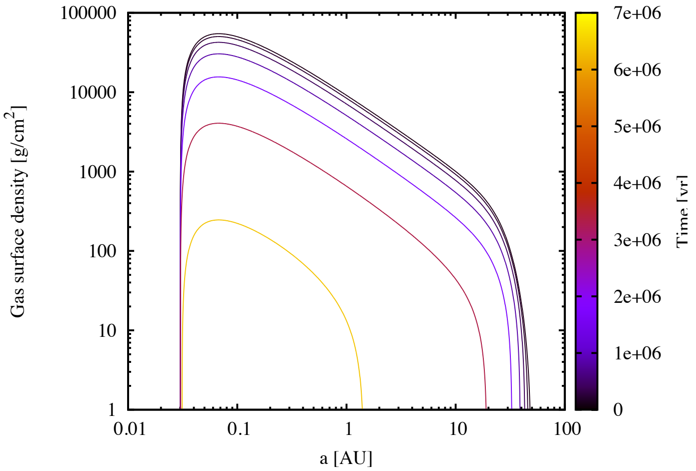

Fig. 2.— Gas surface density as a function of semi-major axis and time.

5. Simulation of a planetary system

Next we want to make a systematic study of planet formation in one system (one protoplanetary disk). All initial conditions are kept constant, except for the semi-major axis which is systematically varied between 0.1 and 100 AU, distributed uniformly in log(a). We change into the systematic directory.

```
cd ../systematic/
```

The *globalPFE.in* file is the same as in the Jupiter simulation. The same the initial conditions from this file are used, except for the planet semi-major axis. If we assume that the Solar System had everywhere a surface density of gas and solids corresponding to 3 times the MMSN, and a decay time of 1.2 Myr, this corresponds to a simulation of Solar System formation. For this we run the code in mode 2:

```
./globalPFE
Mode of operation: single planet (1), systematic study (2), population synthesis (3)
2
```

The code produces a lot of outputs in the terminal as it runs 301 simulations like in the single planet case, but each with different starting positions from 0.1 to 100 AU. We want to investigate how the planet mass changes as a function of semi-major axis and time. To this end, we use the *ref_redXeX.dat* output file, which contains a snapshot of the mass, distance, etc. of all 301 synthetic planets at 0.1 Myr, 1 Myr, ... 10 Gyr. We type in gnuplot

```
gnuplot> set term postscript eps enhanced color solid lw 2 "Times-Roman" 20
gnuplot> set out 'Solarsystem.eps'
gnuplot> load 'Solarsytem.gnu'
```

In Figure 3 (left panel), the lines show the mass of the planet as a function of distance and time. The solid black line gives the final state at 10 Gyr. We notice that first the core forms (red line). The effect of the iceline - the distance at which the temperature falls below 170 K - is visible as an increase of the mass outside of about 2.7 AU. Next, further solid accretion occurs, and the planet starts to migrate inwards (green line). This is followed by gas accretion for the more massive cores, and migration slows down (type II migration). Finally, after the dispersion of the disk, the planets at large distances grow by accretion of solids. The plot shows that this model does not really produce an analogue of the Solar system...

It is interesting to plot the masses also as a function of the embryo starting position, instead of the planet final semi-major axis. This is done with the *SolarsytemAini.gnu* script.

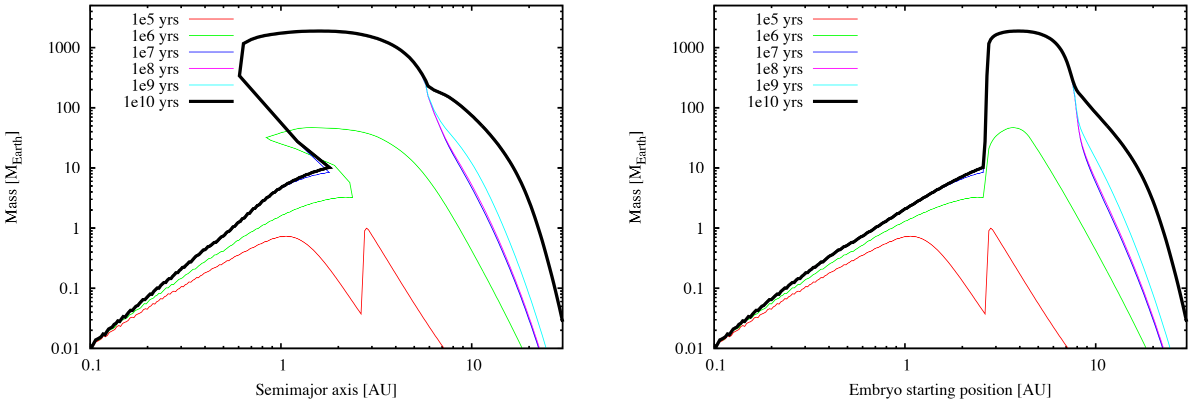

Fig. 3.— Total mass as a function of the final semi-major axis of the planet (left) and the starting position of the embryo (right panel).

In Figure 3 (right panel), we see that we obtain the classical system architecture without migration: inside the iceline we find low-mass rocky planets, followed by gas giants, and finally lower mass ice giants, as the core growth at large distances takes a long time.

6. Population synthesis

Finally, we run 8 population syntheses. We change into the population synthesis directory. There, the following populations are included:

1. nominal
2. fixRminRmax
3. insitu
4. fullmig
5. ILtypeIC1_1e-2
6. CMdotc10
7. kappa1
8. kappa1e-3

where, in each directory, we have the *paramsPFE.in* file with the model setting (same file as before) and the new file *popsynth.in*, which defines the probability distributions. This file contains first the number of synthetic planets that will be generated, and then, for each of the random variables, the probability distribution and its parameters. For each of the seven random variables, there are two lines: on the first line, the type of distribution (1: one fixed value, 2: uniform, 3: uniform in log, 4: normal, 7: lognormal), and on the second line the parameters of the distribution, such as the minimum and maximum for a uniform distribution (see below). The size of synthetic population has initially been set to a moderate number of planets (1000) for a first look at the resulting planet distributions; if your resulting plots are only sparsely populated, you might consider generating larger numbers of planets, particularly if you're running simulations overnight.

6.1. Nominal population

For the nominal population, the *paramsPFE.in* file is the same as in the simulation of Jupiter and in the systematic study above, while the probability distributions are:

```
1000                 !Number of synthetic planets
3                    !Distribution for Mstar
0.7 1.4              !3=uniform in log between 0.7 and 1.4 Msun as IL04
2                    !Distribution for Rindisk
0.01 0.03            !2=uniform 0.01 to 0.03 AU
7                    !Distribution for Routdisk
1.48  0.5 0  10 300  !7=lognormal with mu=log10(30) AU=1.48,sigma 0.5 dex, between 10-300 AU
4                    !Distribution for [Fe/H]
-0.02 0.22 -1 1      !4=normal with mu=-0.02,sigma=0.22,between -1 and 1
7                    !Distribution for fg (disk gas mass scaling factor)
0.25 0.5 0 0.01 10   !7=lognormal with mu=log10(1.78)=0.25 dex,Sigma=0.5 dex,between 0.01-10
3                    !Distribution for taudisk (in Myr)
1.0 10.0             !3=uniform in log from 1 to 10 Myr as in IL04
3                    !Distribution for astart:
0.03 100             !3=uniform in log between 0.03 and 100 AU
```

We start the code in mode 3:

```
./globalPFE
Mode of operation: single planet (1), systematic study (2), population synthesis (3)
3
We generate  1000 synthetic planets
```

Generating 1000 synthetic planets takes about 6 minutes on a modern laptop. Times will be somewhat longer during the hands-on session, depending on how many people are running at the same time.

The state of the population at 1e5, 1e6, ... yrs is written into the *ref_redXeX.dat* files. Finally, we generate the classical aM plot which is one of the most important predictions of a population synthesis model:

```
gnuplot> set term postscript eps enhanced color dashed lw 2 "Times-Roman" 20
gnuplot> set out 'aMnominal.eps'
gnuplot> load 'aM.gnu'
```

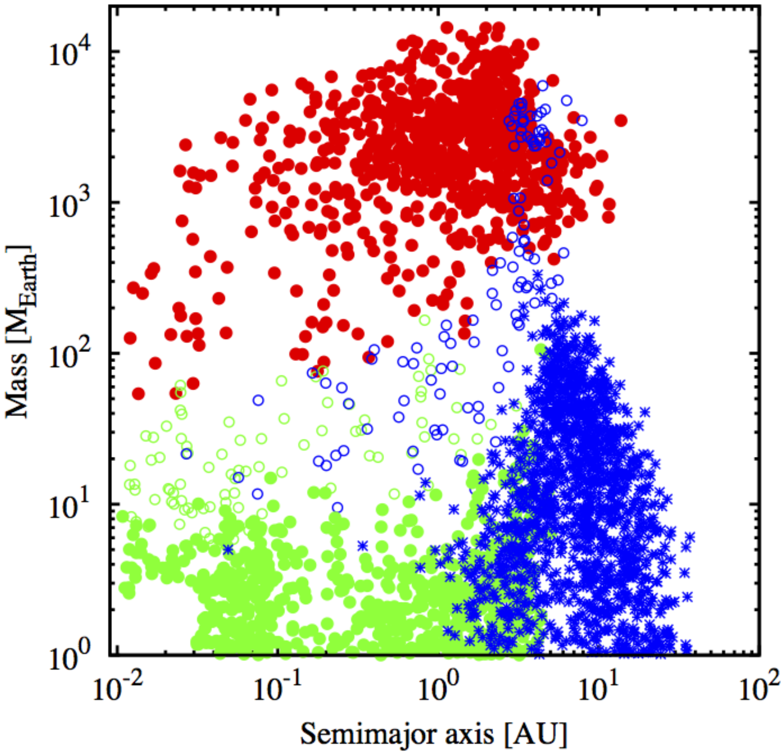

Fig. 4.— Mass-distance diagram at 10 Gyrs in the nominal population.

The results is shown in Figure 4. The colors and symbols are the same as in Ida and Lin (2004): “The green filled circles and the blue crosses represent rocky and icy planets with gaseous envelopes less massive than their cores. The green and blue open circles represent gas-rich rocky and icy planets with gaseous envelopes that are 110 times more massive than their cores. The red filled circles represent gas giants with envelopes more massive than 10 times their cores.” At this stage, many different kinds of analyses are possible, i.e. the frequency of giant planets as a function of disk mass or [Fe/H], the mass function, the semi-major axis distribution, etc. These results can be compared statistically with the observed distributions.

The remaining seven populations differ from the nominal population in some input parameters and probability distributions. In the following, for each population, we specify which quantity was modified with respect to the nominal population case.

6.2. fixRminRmax

This population uses the same parameters as the nominal one, but the inner radius of the disk is fixed to 0.03 AU while the outer is 30 AU. This is achieved by setting in the *popsynth.in* file

```
1                                 !Distribution for Rindisk
0.03                              !1=fixed 0.03 AU
1                                 !Distribution for Routdisk
30                                !1=fixed to 30 AU
```

The resulting aM diagram (Figure 5) shows that now low-mass planets migrating in type I migration are stopped at about 0.1 AU where the surface density starts to decrease due to the magnetospheric cavity (see Figure 2). The planets migrating in type II migration stop at 0.03 AU. We also note that there are less planets at large distances, because Rout is never larger than 30 AU, in contrast to the nominal case.

6.3. Insitu

This population assumes that no orbital migration is occurring at all. This is achieved by setting $C1 = C2 = 0$ in the *paramsPFE.in* file. The distributions and all other parameters are the same as in the nominal case.

```
3                  Mode of type I  migration (3)
0D0                Scaling factor for type I rate  (0D0) (no migration)
2                  Mode of type II  migration(2)
0D0                Scaling factor for type II rate (0D0) (no migration)
```

Figure 6 shows that no close-in planets are created, as expected. In order to have more planets at small distances, we would have to increase the solid surface density, as recently proposed by various authors.

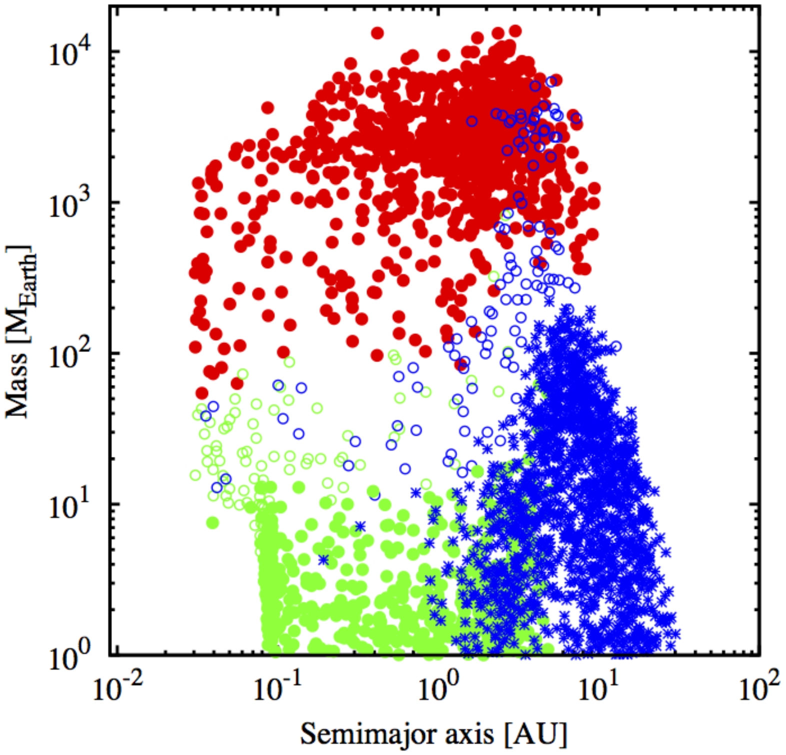

Fig. 5.— Mass-distance diagram at 10 Gyrs in the fixRminRmax population.

6.4. FullMig

This population assumes that orbital migration is occurring at the full rate ($C1 = C2 = 1$ in the *paramsPFE.in* file). This has very strong consequences for the population (Figure 7). No giant and terrestrial planet exists, as most embryos migrate towards the star. This reflects the well-known problem that orbital migration, as calculated from classical theory, is very fast.

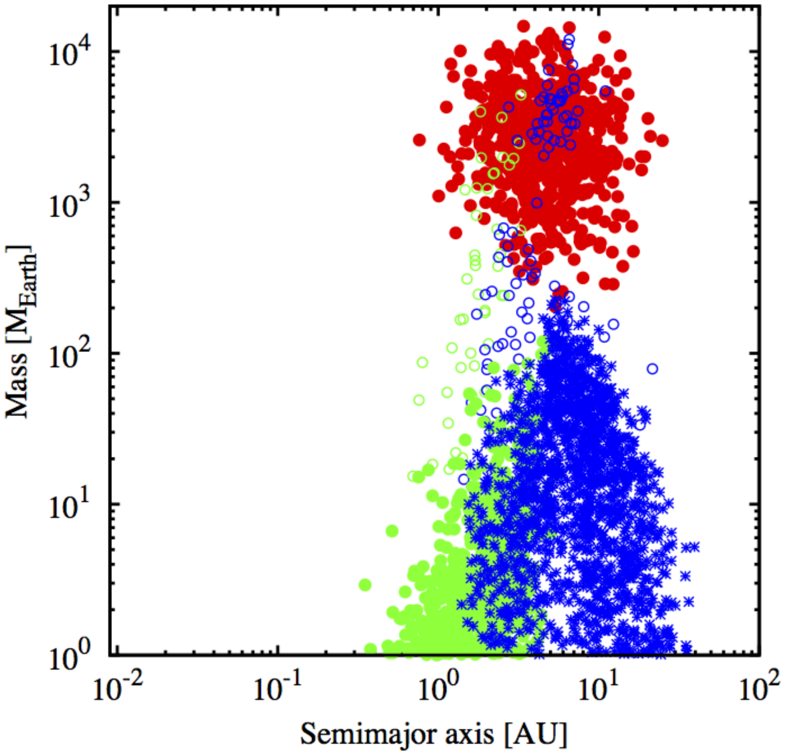

Fig. 6.— Mass-distance diagram at 10 Gyrs in the insitu population.

6.5. ILtypeIC1_1e-2

In this population, the type I and II migration is calculated as in Ida and Lin (2004), and the reduction factor for type I migration is 0.01, and 0.1 for type II migration. Thus we set in the parameter file:

```
1                                 Mode of type I  migration (1)
1D-2                              Scaling factor for type I rate  (1D-2)
1                                 Mode of type II  migration(1)
1D-1                              Scaling factor for type II rate (1D-1)
```

Figure 8 shows that the population is similar to the nominal one, but without planets at about 0.1 AU due to the different dependency of the migration rate in type I migration on the radial slope of the temperature and surface density. In addition, the positive correlation of the (maximal) mass of giant planets with the semi-major axis seen in the nominal case is absent here, due to the different type II migration prescription.

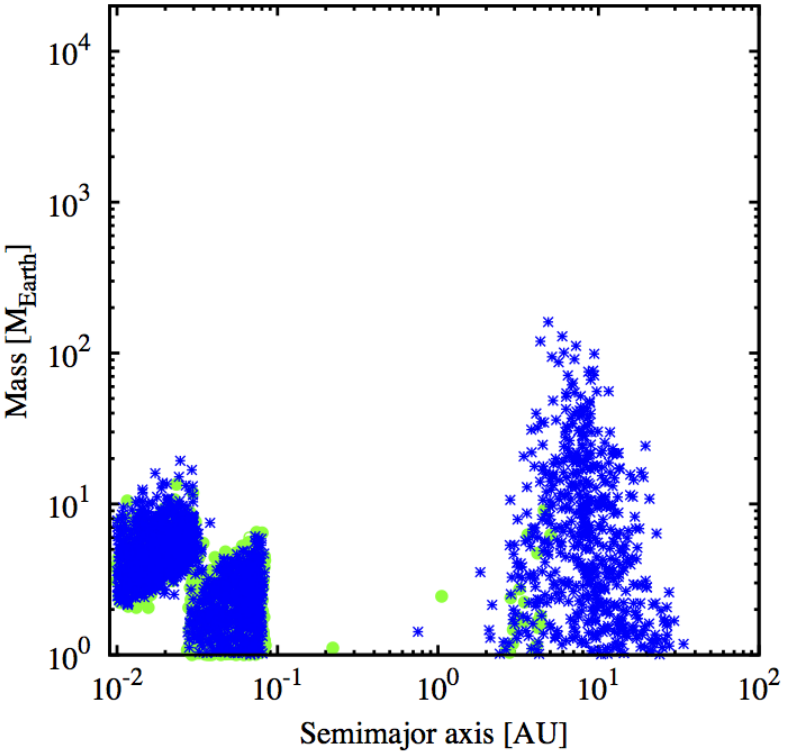

Fig. 7.— Mass-distance diagram at 10 Gyrs in the fullmig population.

6.6. cmdotc10

In this population the accretion rate of planetesimals is arbitrarily assumed to be 10 times faster than in the nominal case. A faster solid accretion rate could for example occur if planets form by accreting pebbles rather than planetesimals. This is achieved by setting in the parameter file:

```
1D1               Scaling factor for core accretion rate (1D1)
```

From Figure 9, we note more giant planets at larger distances, out to about 30-40 AU. This is interesting in the context of directly imaged giant planets like those orbiting HR8799.

6.7. kappa1

In this population, the opacity κ in the protoplanetary atmosphere is 1 cm^2/g instead of 0.01 cm^2/g, making gas accretion less efficient. This is obtained by setting in the parameter

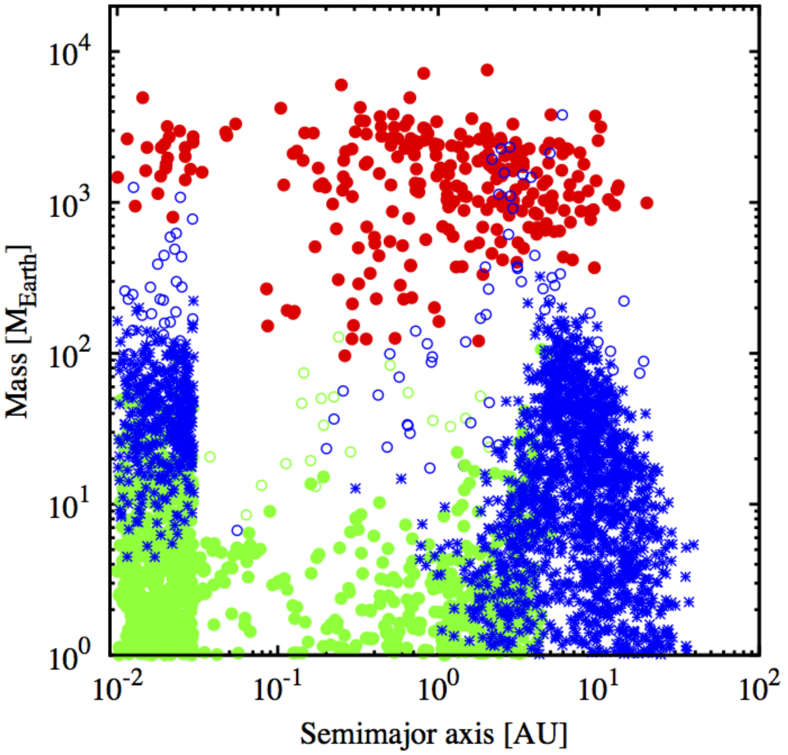

Fig. 8.— Mass-distance diagram at 10 Gyrs in the ILtypeIC1_1e-2 population.

file:

```
1D0           Opacity in protoplanetary atmosphere [cm**2/g] (1D0)
```

The resulting aM is shown in Figure 10. This is another example of a population that is clearly not compatible with observations, suggesting that the typical opacity in protoplanetary atmospheres is much smaller than in the ISM, likely due to grain growth (Podolak 2003; Ormel 2014).

6.8. kappa1e-3

In this population, the opacity in the protoplanetary atmosphere is reduced to 1e-3 cm^2/g instead of 0.01 cm^2/g, making gas accretion more efficient. From Figure 11 we note the more massive and numerous giant planets, and their larger orbital distances compared to the nominal case. Also the low-mass planets are more gas-rich.

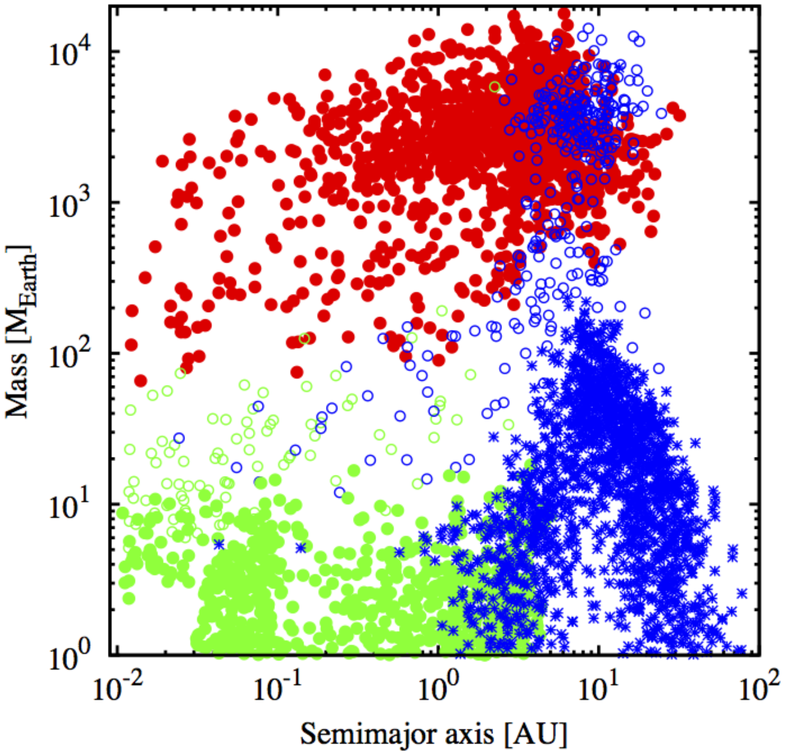

Fig. 9.— Mass-distance diagram at 10 Gyrs in the cmdotc10 population.

7. Input files

7.1. globalPFE.in

This is the input file with the physical initial conditions (disk and stellar properties). It is used in the individual planet simulation and planetary system simulation mode. In the population synthesis mode of the program, it is not used.

7.2. ParamsPFE.in

This file contains the parameters for the model. It also allows one to choose between different theoretical prescriptions of the same mechanism (e.g. orbital migration). Here, we discuss its content using as a reference the nominal case (nominal values are in parentheses). The file is needed in all three working modes of the program. Below are the "knobs" that can be turned to modify the outcome of the population synthesis (as well as, but to a lesser extent, the probability distributions):

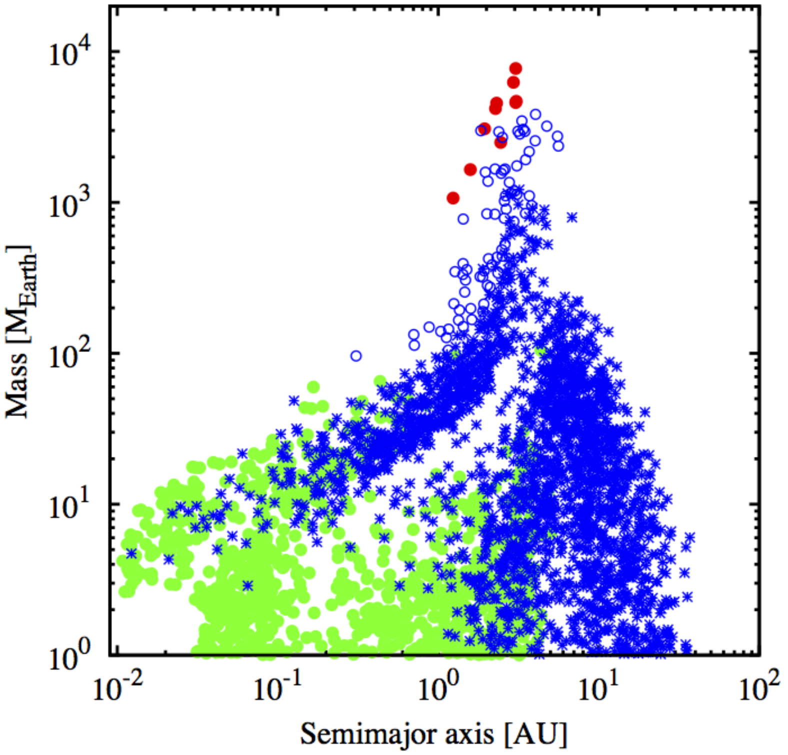

Fig. 10.— Mass-distance diagram at 10 Gyrs in the kappa1 population.

- p_T: power law exponent for temperature in the disk. Nominal: -0.5. Range: -0.5 to -1 (?, in reality not uniform in the disk due to opacity transitions). See Equation 1 in GlobalPFE Theoretical Background doc;

- α: alpha viscosity parameter in the disk. Nominal: 10^{-3}. Range: 10^{-5} (dead zone) - 10^{-2} (?). Influences the type II migration rate and accretion rate in the disk limited regime, but does not influence the disk lifetime (not self-consistent). See Equation 18 in GlobalPFE Theoretical Background doc;

- $f_{D/G,\odot}$: dust to gas mass ratio corresponding to [Fe/H]=0.0 (solar). Nominal: 0.0149 (Lodders 2003). Range: 0.0149-0.04 (higher values due to advection/drift of dust from the outer parts of the disk into the planet forming part. See Equation 6 in GlobalPFE Theoretical Background doc;

- κ: typical opacity in protoplanetary atmosphere [cm^2/g]. Nominal: 0.01. Range: 0-1. See Equation 13 and Equation 14 in GlobalPFE Theoretical Background doc;

- p_{KH}: absolute scaling for power law of envelope Kelvin Helmholtz timescale. Nominal:

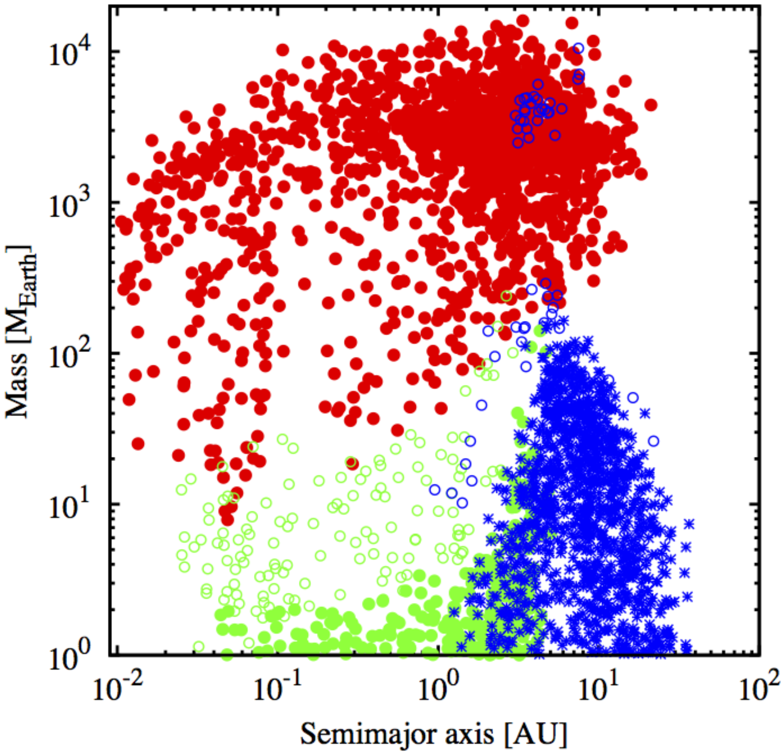

Fig. 11.— Mass-distance diagram at 10 Gyrs in the kappa1e-3 population.

10.4. Range: 8-11. See Equation 14 in GlobalPFE Theoretical Background doc;

- $q_{\rm KH}$: power-law exponent for mass dependency of envelope Kelvin Helmholtz timescale. Nominal: -1.5. Range: -1 to -4. Lower values give an emptier planetary desert. See Equation 14 in GlobalPFE Theoretical Background doc;

- $ilimMe$: mode of limiting gas accretion. Nominal: 5. Range: 1-5. See Section 2.3.1 in GlobalPFE Theoretical Background doc;

- $itype1$: mode of type I migration. Nominal: 3. Range: 1-3. See Section 2.4.1 in GlobalPFE Theoretical Background doc;

- $C1$: scaling factor for type I migration rate. Nominal: 0.004. Range: 0 (no type I migration) - 1 (full rate). See Section 2.4.1 in GlobalPFE Theoretical Background doc;

- $itype2$: mode of type II migration. Nominal: 2. Range 1-2. See Section 2.4.2 in GlobalPFE Theoretical Background doc;

- *C*2: scaling factor for type II migration rate. Nominal: 0.002. Range: 0 (no type II migration) - 1 (full rate). See Section 2.4.2 in GlobalPFE Theoretical Background doc;
- *Cmdotc*: scaling factor for core accretion rate. Nominal: 1.0. Range: 0.1-100 (?). High values could mimic pebble accretion (?). See Section 2.2 in GlobalPFE Theoretical Background doc.

8. Output files

8.1. tracks_XXXXX.dat

Temporal evolution of an individual planet. The values are printed at each timestep. The columns are:

1. itime: Timestep number
2. time/an: Time in years
3. Mc/Mearth: Core mass in Earth masses
4. Me/Mearth: Envelope mass in Earth masses
5. (Mc+Me)/Mearth: Total mass in Earth masses
6. Mgdisk/Msol: Disk gas mass in solar masses
7. Mddisk/Mearth: Disk solid mass in Earth masses
8. Mdotc/Mearth*an: Core accretion rate (Mearth/yr)
9. Mdote/Mearth*an: Gas accretion rate (Mearth/yr)
10. ap/AU: Planet semi-major axis/AU
11. dt/an: Timestep duration/yrs
12. Sigmag(ip): Disk gas surface density at planet position [g/cm^2]
13. Sigmad(ip): Disk planetesimal surface density at planet position [g/cm^2]
14. Hdisk(ip)/AU: Disk vertical scale height
15. Rhills/AU: Hill sphere
16. dlsigmagdlr: Local slope of gas surface density
17. adot/AU*an: Migration rate (AU/yr)

8.2. diskevo.dat

The structure of the disk as a function of distance and time:

1. time/an: time/yr
2. rdisk(i)/AU: radius/AU
3. Sigmag(i): gas surface density
4. Sigmad(i): solid surface density
5. yderi: power law exponenent of the surface density
6. Tdisk(i): temperature
7. Hdisk(i)/AU: vertical scale height

8.3. ref_redXeX.dat

Snapshot of the state of all planets (1 planet/line) in a population at a time XeX. The first 17 columns are the same as in the tracks file, but at the end there is also the planet number and the initial conditions.

17. As in tracks_XXXXX.dat
18. iplanet: Planet number (in systematic study or synthesis)
19. Mstar/Msol: Stellar mass / Msun (initial condition)
20. fg: Gas disk scaling factor relative to solar (initial condition)
21. [Fe/H]: Metallicity (initial condition)
22. taudisk/an: Disk decay time in yr (initial condition)
23. aice/AU: Iceline position / AU
24. apini/AU: Planet initial position / AU (initial condition)
25. Rindisk/AU: Disk inner radius / AU (initial condition)
26. Routdisk/AU: Disk outer radius / AU (initial condition)

9. Essential code elements

The formation code (written in Fortran 90) consists of a first part in which the initial conditions are read, and then one big loop over time in which the core mass, envelope mass, and semi-major axis are numerically integrated. Other subroutines calculate the evolution of the gas disk and of the disk of planetesimals. There are also subroutines determining the duration of the numerical time step, and one that writes the output files. The formation code is embedded in a wrapper which calls it for the three different modes: 1. single planet; 2. planetary system; 3 population synthesis. In the population synthesis mode, there are subroutines generating the probability distributions with which the formation code is called. Each source file contains a subroutine of the same name and essential documentation.

10. Warnings, Troubleshooting, & ”Help, Please!”

We have tested the model for a number of parameters and probability distributions, but there is a much higher number of combinations that we have not tested. So it is possible that in some situations, the formation model crashes or produces bogus results.

If you do not understand something, or do not know how to do something, feel free to ask. You have very limited time for this exercise, so no question is too stupid!

REFERENCES

S. Ida, D.N.C. Lin, Toward a deterministic model of planetary formation. i. a desert in the mass and semimajor axis distributions of extrasolar planets. ApJ **604**, 388 (2004)

K. Lodders, Solar system abundances and condensation temperatures of the elements. ApJ **591**, 1220 (2003)

C.W. Ormel, An Atmospheric Structure Equation for Grain Growth. ApJ **789**, 18 (2014)

M. Podolak, The contribution of small grains to the opacity of protoplanetary atmospheres. Icarus **165**, 428 (2003)

This preprint was prepared with the AAS LaTeX macros v5.2.