Changes to TaperChain2008 code

Michelle Przedborski, June 2015

The functionality of the 2008 code has been improved, and it now has the ability to specify central mass impurities in the parameter file. Thus the source code (cpp) file has been renamed to "TaperChain2008_Mass_Impurity.cpp".

For full instructions on how to run the code, one should thoroughly read the TaperChain 2009 manual, and only after doing so consult this current document for additions to the functionality.

Note that, as with the original parameter commands, the case of the letters is irrelevant as everything is converted to lowercase in the code. Note also that the 2008 code currently does not have the "chain pattern" command.

The following commands have been added:

(1) The ability to easily incorporate central mass impurities.

(a) numimpure: specify the number of impurity grains. The syntax is e.g. "numimpure: 6" which specifies that we want 6 mass impurities. The impurity grains are placed at the center of the chain automatically. NOTE that the program will only run if the parity of N and numimpure ARE THE SAME, and this is to ensure symmetry is maintained when placing impurity grains at the center of the chain. The default value of numimpure is set to 0, so if you do not specify numimpure in the parameter file it will remain 0, and none of the following commands will actually be used in the program.

(b) rhoImp: specify the density of the mass impurity grains. Like the original command, density is expressed in mg/mm^3. Syntax is given as e.g. rhoimp: 6

(c) rhofac: specify the radius factor of mass impurity grains. The radius (in mm) of impurity grains = rhofac*Rlarge, where Rlarge is the radius of the large (left) grain, defined in the TaperChain 2009 manual. Syntax is given as e.g. rhofac: 0.5

(2) How to specify the interaction parameters for the mass impurities.

(a) D: specifies the grain-grain and grain-wall interaction in the host chain. This parameter was already defined in the previous version of the TaperChain2008 code. In the current code, it is read in and set as parameter D2. One can also slightly modify the code so that the grain-wall interaction of the host chain is set to a different value. This is done by resetting the D value in smalla[0] and smalla[nptles] in the function ChainRelPositions(). Syntax is given as e.g. D: 4

(b) Dhi: specifies the interaction between host chain and impurity grains. This parameter is read into the code as D1. Syntax is given as e.g. Dhi: 10

(c) Dii: specifies the interaction between impurity-impurity grains. This parameter is read into the code as D3. It is only used if numimpure > 1.

(3) Additional driving capabilities:

(a) New functions have been added to the code so that the chain can be driven by forces either symmetrically from both the left and right, or asymmetrically from either the left or right side only, or asymmetrically on both sides. Note that the driving starts at t=0.

The nature of the driving of the chain is now controlled by switches, whose default values are all zero.
(i) asymLeft: if ==0, no force is applied to left edge of chain. If asymLeft ==1, a left asymmetric force, as specified by "Addforce" command is applied.
(ii) asymRight: same as above but for the right edge of the chain. If one uses "Addforcelast" command in the parameter file, then this switch is set to 1.
(iii) symLeftRight: if ==1, then a symmetric driving force is applied to both ends of the chain, as specified by the "Addforcefirstlast" command.

The syntax for "Addforcefirstlast" and "Addforcelast" is identical to "Addforce". The "Addforce" command adds a force to left edge grain only, as specified in the TaperChain 2009 manual. Consult the TaperChain 2009 manual for the syntax**.

Addforcelast: adds a force to the last grain (e.g. right edge grain only) as specified.

Addforcefirstlast: adds a force to both edge grains (symmetrically) as specified.

Note that to control the program flow with these added functions, the "velocityVerletStep()" function has now been modified to include all cases and switches.

How to specify the added forces is provided in the TaperChain 2009 manual. The only change with the updated code is that **you must now additionally specify a window for all types of forces, e.g. "Addforce: k 2 1000" specifies a constant force of magnitude 2kN for a window of 1000 micro-s to the left edge grain only. Since one may wish to drive the chain with e.g. a sinusoidal force for the duration of the simulation, for any non-constant force, one can specify the window as "-1" to mean the duration of the simulation.

(b) Specifying output files: recall that the force applied file is specified by "A", and this file contains the force applied to the left grain if using the "Addforce" command. If using "Addforcelast" command, one can also print out the forces applied to the last (right) grain by putting "L" into the list of output files. If applying a symmetric driving force to both ends of the chain, one can add "S" to the list of output files to output the symmetric force that is applied to both left and right edge grains.

(4) Delta velocity driving capabilities:

The delta velocity function has now been changed completely to make it easier if one wishes to apply multiple delta velocity perturbations to a particular grain, or to apply finite-time velocity perturbations. The function now requires 5 arguments: (1) grain #, (2) time, (3) velocity, (4) number of hits, (5) interval between hits. This new function can be used now in two ways:

(a) To apply one or more delta velocity perturbations to a specified grain beginning at a specific time, and occurring at specified intervals. For example, “deltaV: 3 10 0.001 6 20” will apply 6 delta velocity perturbations to grain 3 of magnitude 0.001 mm/micro-s. The first will occur at t=10 micro-s, and the remainder will occur at 20 micro-s intervals, e.g. at 30, 50, 70, etc. micro-s.

If one wishes to apply a single delta velocity perturbation, the interval is irrelevant, but it is suggested to set this value to 0, e.g. “deltaV: 3 10 0.001 1 0” will apply a single delta velocity perturbation of magnitude 0.001 micro-s to grain 3 at t=10 micro-s.

(b) One can now use this function to apply a constant velocity perturbation to a specified grain for a specified window of time. This idea is similar to driving the system with a constant force (acceleration) for a specified window. Energy will be put into the system to keep the velocity of a particular grain to the value specified by the delta velocity function. This can be used instead of the “largeinitvelocity” command to initially drive the system, allowing for a finite-time velocity perturbation. The syntax in this case is slightly different from above. One must set the number of hits (4) = -1, so the program knows this is what you want to do. Then the interval (5) will be taken as the window of the finite-time perturbation. For example: “deltaV: 1 0 0.001 -1 20” will apply a finite-time velocity perturbation of magnitude 0.001 mm/micro-s to grain 1 beginning at t=0 and lasting for 20 micro-s. Note that this function will maintain the velocity of the specified grain at the specified value, hence this requires an input of energy into the system and one will see the total system energy increase for the duration of the window.

(4) Resuming a simulation:

If a simulation was not taken for a long enough duration, rather than start it over from the beginning, there is now the option to resume it from where it left off. There is a new function in the code that writes to a file called “resume.txt” in the last loop of a simulation. This has become a standard part of the code now, and the file will be created for any simulation. The resume.txt data file contains a list of particle relative positions followed by a blank line, followed by particle velocities. If the user wishes to restart the simulation from where it left off, all they have to do is specify “restart: 1” in the parameter file along with “timemin: #”, where “#” is the value of the time (micro-s) where the simulation left off. Currently this time value has to be placed in by hand in the parameter file, and must come **after** restart: 1 in the parameter file. It is important that one puts the correct value of timemin, and this value must correspond directly to the data in the “resume.txt” file. The Taperchain code will then go into resume mode and read in the values from “resume.txt” and restart the simulation. The default value is “restart: 0”, and if this parameter is not specified in the parameter file, the code will run normally, i.e. will not enter restart mode.

Note that the option to resume a simulation should only be used if one wishes to collect longer term data on a previous simulation. The parameter file must still contain the vital information (e.g. the mass impurities, and interaction parameters, radii, etc.), but should not contain any “addforce” commands, as forces are applied at t=0 in the current version of the code. If one includes the “largeinitv” or “Smallinitv” commands in the parameter file, although these values will be printed to the Readme file,

they will not actually be used in the simulation. If one wishes to add an additional velocity perturbation to a simulation that is being resumed, use the deltaV command.

Note also that all files are appended to in resume mode. This includes the “Readme” file, thus details of the resumed simulation can be found there, following the original output.

****

Sample parameter file using these new commands:

****

N: 36
dt: 0.00001
nsteps: 5000000000
files: KTVFXAL
rHO: 7.82
numimpure: 6
rhoimp: 2.23
rhofac: 0.5
d: 0.007049
dhi: 0.0143
dii: 0.0215
addforce: k 161.5 100
addforcelast: r 10 2 1000
Wall: 11
timeSPItS: 10.0
fileName: CMI_Jun.29.2015.Pyrex_Center.Equal_Const_Force.10Percent.N36_6_Soft_HW
exponential: 2.5
largeinitv: 0
smallinitv: 0
preload: 0
Rlarge: 6
q: 0

The sample parameter file sets up a chain with a total number of 36 grains, 6 central impurities with density 2.23 mg/mm^3 which are 3mm in size, embedded in a host chain with grain radius 6mm and density 7.82 mg/mm^3. Interactions for the host chain are specified by D value 0.007049, and host-impurity are 0.0143, and impurity-impurity are 0.0215. The simulations are using an integration timestep of 0.00001 micro-s, with a total number of 5000000000 steps (meaning the total length of the simulation is 50,000 micro-s), and is spitting to output files KTVFXAL every 10.0 micro-s. There are left and right walls on the chain, and the interaction potential is specified by the exponent 2.5. Further, there is no tapering or initial preload, and the initial velocity of the “large” (left) and “small” (right) edge

grains are both zero. All file names will have the string "CMI_Jun.29.2015.Pyrex_Center.Equal_Const_Force.10Percent.N36_6_Soft_HW" at the end of them. We perturb this chain by adding a constant force of magnitude 161.5 kN to the left edge grain for a window of 100 micro-s, as well as a random force with maximum value 10 kN and a minimum value of 2kN to the right edge grain, which oscillates with a frequency of 1000.

For further information on the other types of forces that can be implemented, see TaperChain 2009 manual.

Note that all of these parameters are printed to the ReadMe file, including the force perturbations. I have also changed the format of the ReadMe file slightly so that all force pre-factor (a) values are printed out. One can double-check the ReadMe file to make sure they are setting up the chain as desired.

****

Another sample parameter file using these new commands:

****

N: 36
dt: 0.00001
nsteps: 5000000000
files: KTVFX
restart: 1
timemin: 10000
rHO: 7.82
numimpure: 6
rhoimp: 2.23
rhofac: 0.5
d: 0.007049
dhi: 0.0143
dii: 0.0215
Wall: 11
timeSPItS: 10.0
fileName: CMI_Jun.29.2015.Pyrex_Center.Equal_Const_Force.10Percent.N36_6_Soft_HW
exponential: 2.5
preload: 0
Rlarge: 6

This last parameter file puts the program into restart/resume mode, continuing at 10,000 micro-s, and running an additional 50,000 micro-s.