A. RUNNING PARALLELIZED PRIME20

I. Generating initial configuration:

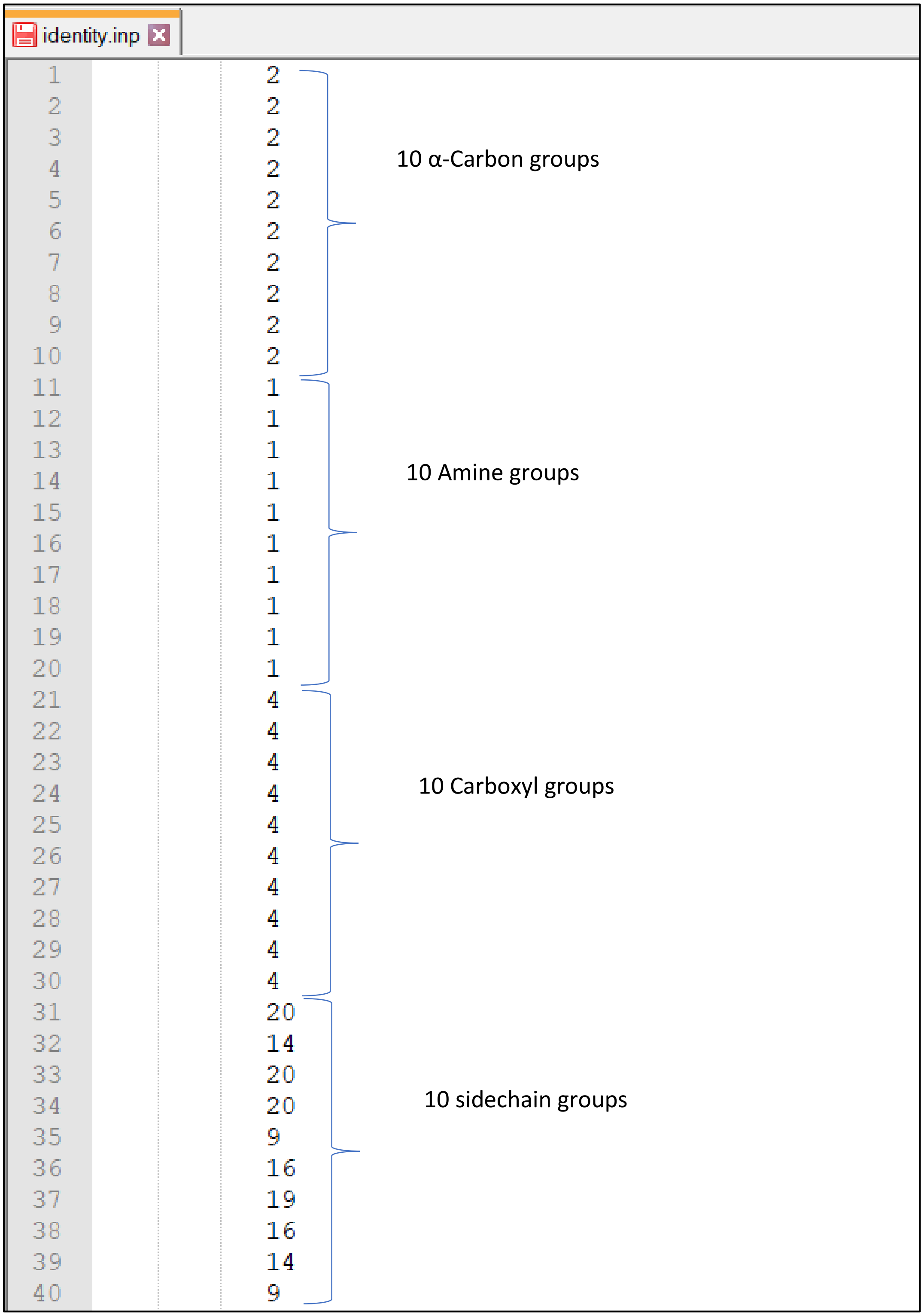
DMD simulation using PRIME20 starts with building an initial configuration. The current version is effective for systems of peptides with less than 31 residues. It is recommended that concentration and number of peptide chains are reduced for longer peptides to avoid overlap due to overcrowding. User should check output files for overlap error and reduce the system size if an error is reported. The code for configuration generation is under the **genconfig** directory. Steps to prepare initial configuration is as follow.

1. Under the **inputs** directory, locate files **identity.inp** and **identity2.inp**.
   Notes: the program can run simulations of systems with two different peptides. Therefore, each peptide is defined in either **identity.inp** or **identity2.inp**. In case of only one peptide is considered, **identity.inp** and **identity2.inp** are identical.
2. Modify identity.inp and identity2.inp to define your peptides.
   The current two files are examples how PRIME20 defines peptides. Each type of beads associates with a number as listed in **Table 1**. The sidechain-sidechain energetic and geometric parameters are derived in Cheon et al., Protein, 2010 (https://doi.org/10.1002/prot.22817).

Table 1: Beads' identifications in PRIME20

Beads	**PRIME20 identification**	**Beads**	**PRIME20 identification**
NH (Amine group)	1	S (Serine)	18
CA (α-Carbon group)	2	T (Threonine)	19
CO (Carboxyl group)	4	A (Alanine)	20
G (Glycine)	9	C (Cysteine)	21
R (Arginine)	10	I (Isoleucine)	22
N (Asparagine)	11	L (Leucine)	23
D (Aspartic Acid)	12	M (Methionine)	24
Q (Glutamine)	13	F (Phenylalanine)	25
E (Glutamic Acid)	14	W (Tryptophan)	26
H (Histidine)	15	Y (Tyrosine)	27
K (Lysine)	16	V (Valine)	28
P (Proline)	17		

All α-Carbon group beads are listed first then following with amine group beads, carboxyl group beads and amino acid sidechain beads at last. A correct **identity.inp** file of a 10-residue peptide is as following figure.



```
2
2
2
2
2
2
2
2
2
2
1
1
1
1
1
1
1
1
1
1
4
4
4
4
4
4
4
4
4
4
20
14
20
20
9
16
19
16
14
9
```

10 α-Carbon groups

10 Amine groups

10 Carboxyl groups

10 sidechain groups

3. Modify **gen_config_random-SQZ.f90**
   9 parameters need to be changed to present the system of your interest. These parameters are listed on **Table 2**.

Table 2: Important parameters to define an initial configuration.

number of beads in peptide 1	nb
number of beads without glycines in peptide 1	nb2
chain length in peptide 1	chnln
chain number in peptide 1	nc
number of beads in peptide 2	nb3
number of beads without glycines in peptide 2	nb4
chain length in peptide 2	chnln2
chain number in peptide 2	nc2
box length (Angstrom)	boxl

$$boxl = \left(\frac{Total\ number\ of\ peptide\ chains * 1000}{Avogadro's\ number * Concentration}\right)^{1/3} * 10^9$$

$where\ concentration\ is\ in\ mM$

4. Use any fortran compiler to compile the code **gen_config_random-SQZ.f90** and then run the code to generate initial configuration system. Although the DMD simulation is paralleled, the **gen_config_random-SQZ.f90** is a serial code.
5. Results of the initial configuration generation step are recorded in different sub_directories
   - Within **genconfig** directory:
     - Compiled file
     - Output file
     - chninfo-n1.data
     - chninfo-n2.data

     Notes: these files need to be deleted before the new system is generated if the entire package is copied over.

   - Within the sub_directory **results**:
     - run0000.lastvel

    - run0000.energy
    - run0000.config
    - run0000.bptnr

    Notes: These files contain initial configurations and velocities

- Within the sub_directory **parameters**:
    - identity.inp
    - hp1.inp
    - hp2.inp
    - firstside1.data
    - firstside2.data

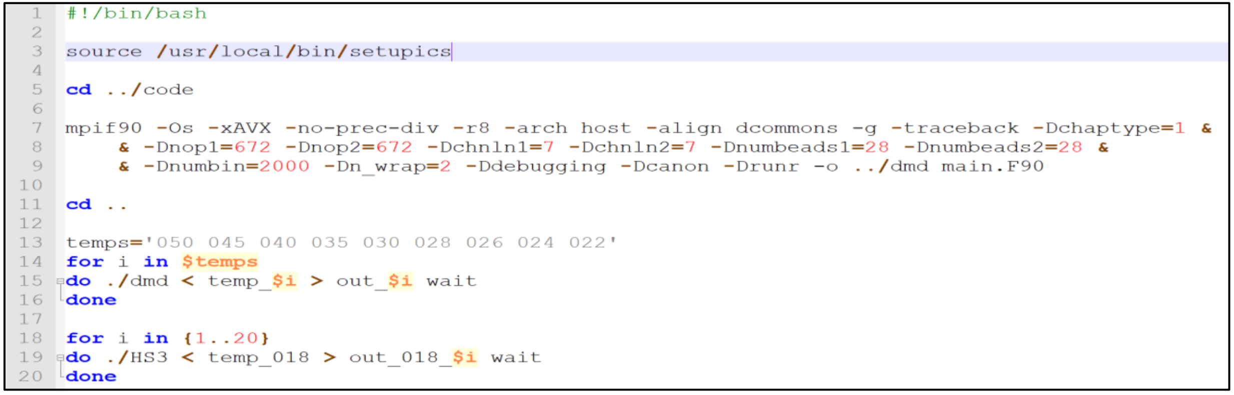
    Notes: These files contain identity of peptides in the system

- the sub_directory **check**:
    - All files in here are for the users to check configuration, velocities, mass and energy of the initial system.

6. The generated files in **results** and **parameters** are essential to the DMD simulation and need to be copied over to the corresponding **results** and parameters directories that locate one directory before **genconfig** directory.

II. DMD Simulations:

1. Modify the box length in **inputinfo**.
    - **inputinfo** is a code under directory **code**. The code reads in all information that is required for DMD simulation.
    - Search for **boxl** and update the value to similar **boxl** used for generating initial configuration.
2. The script to submit simulation job is under **qfile** directory. The script is written for GNU Fortran Compiler. PRIME20 is paralleled using Message Passing Interface (MPI). Users are recommended to check and modify the code to adapt to their compilers.
3. Change the source (highlighted line in the code) to where Fortran compiler and library is setup.



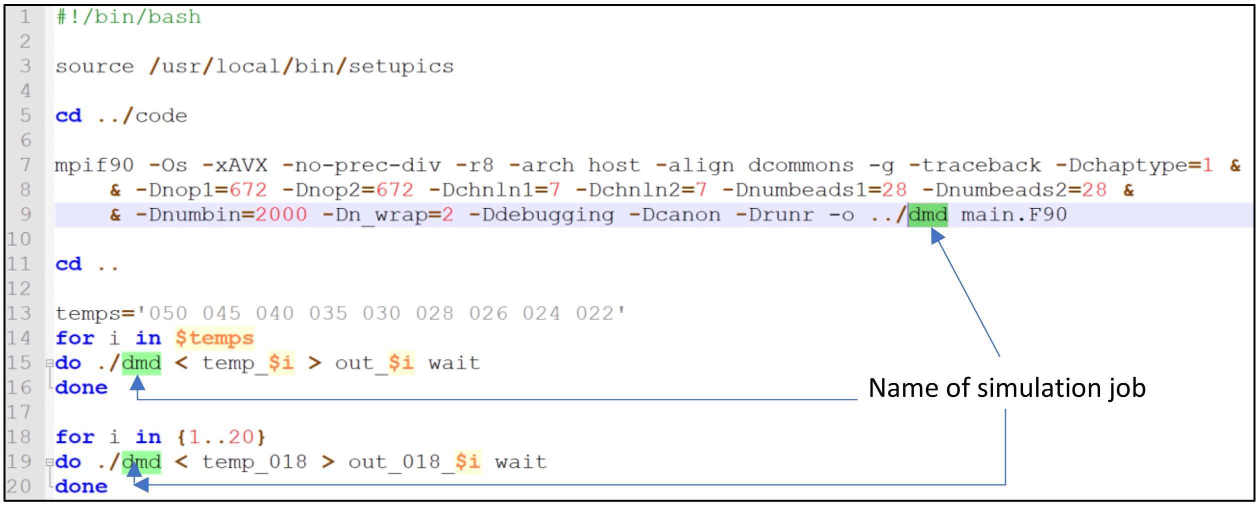
```
#!/bin/bash

source /usr/local/bin/setupics

cd ../code

mpif90 -Os -xAVX -no-prec-div -r8 -arch host -align dcommons -g -traceback -Dchaptype=1 &
     & -Dnop1=672 -Dnop2=672 -Dchnln1=7 -Dchnln2=7 -Dnumbeads1=28 -Dnumbeads2=28 &
     & -Dnumbin=2000 -Dn_wrap=2 -Ddebugging -Dcanon -Drunr -o ../dmd main.F90

cd ..

temps='050 045 040 035 030 028 026 024 022'
for i in $temps
do ./dmd < temp_$i > out_$i wait
done

for i in {1..20}
do ./HS3 < temp_018 > out_018_$i wait
done
```

4. Change 6 parameters listed on Table 3 before submitting the script.

Table 3: Essential parameters in script.sh

Total numbers of beads of peptide 1 chains	Dnop1 = nb2*nc
Total number of beads of peptide 2 chains	Dnop2 = nb4*nc2
Chain length of peptide 1	Dchnln1 = chnln
Chain length of peptide 2	Dchnln2 = chnln2
Numbers of beads in 1 chain of peptide 1	Dnumbeads1 = nb2
Numbers of beads in 1 chain of peptide 2	Dnumbeads2 nb4

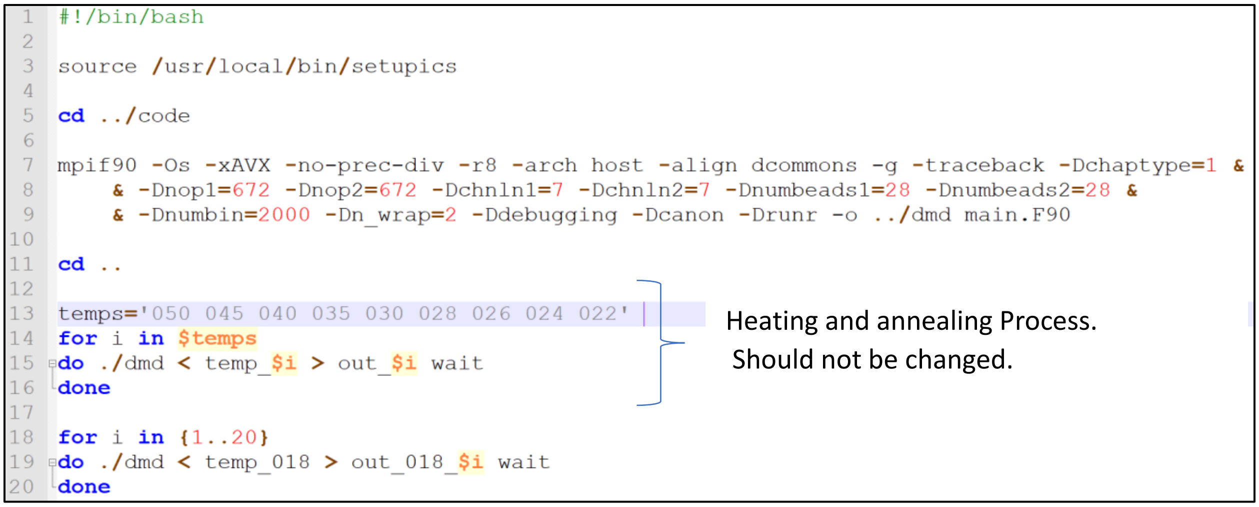
5. The simulation job can be renamed for the convenience of users.



```
#!/bin/bash

source /usr/local/bin/setupics

cd ../code

mpif90 -Os -xAVX -no-prec-div -r8 -arch host -align dcommons -g -traceback -Dchaptype=1 &
     & -Dnop1=672 -Dnop2=672 -Dchnln1=7 -Dchnln2=7 -Dnumbeads1=28 -Dnumbeads2=28 &
     & -Dnumbin=2000 -Dn_wrap=2 -Ddebugging -Dcanon -Drunr -o ../dmd main.F90

cd ..

temps='050 045 040 035 030 028 026 024 022'
for i in $temps
do ./dmd < temp_$i > out_$i wait
done

for i in {1..20}
do ./dmd < temp_018 > out_018_$i wait
done
```

Name of simulation job

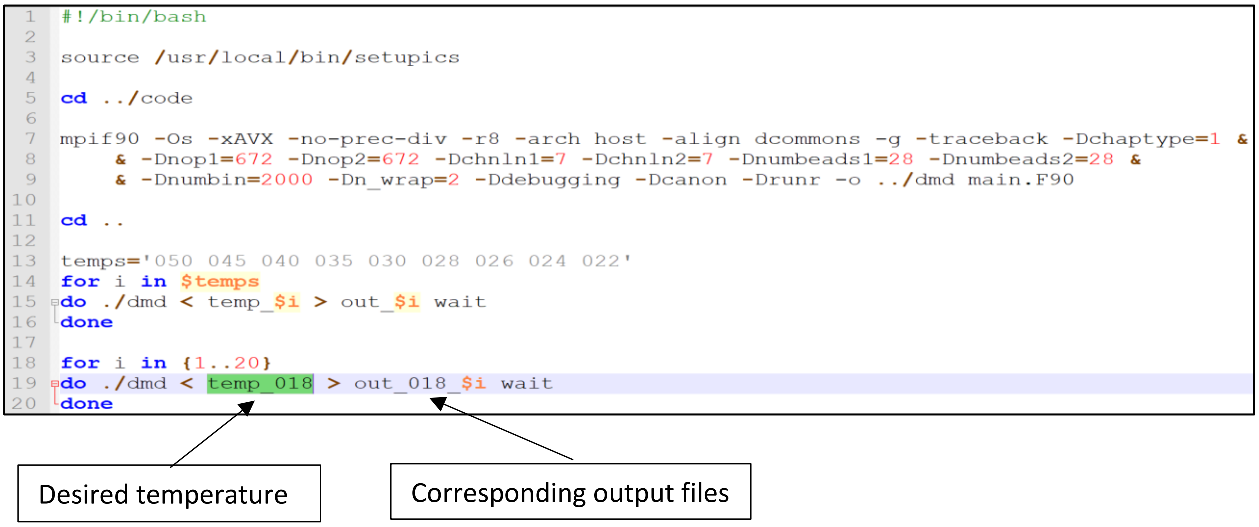
6. At the beginning of DMD simulation, the system will be heated to high temperature and then be slowly cooled to the desired temperature. This step is to make sure that all peptide chains are denatured and that the DMD simulation starts with all random coils. This highlighted section should not be changed for any simulation.



```bash
#!/bin/bash

source /usr/local/bin/setupics

cd ../code

mpif90 -Os -xAVX -no-prec-div -r8 -arch host -align dcommons -g -traceback -Dchaptype=1 &
     & -Dnop1=672 -Dnop2=672 -Dchnln1=7 -Dchnln2=7 -Dnumbeads1=28 -Dnumbeads2=28 &
     & -Dnumbin=2000 -Dn_wrap=2 -Ddebugging -Dcanon -Drunr -o ../dmd main.F90

cd ..

temps='050 045 040 035 030 028 026 024 022'
for i in $temps
do ./dmd < temp_$i > out_$i wait
done

for i in {1..20}
do ./dmd < temp_018 > out_018_$i wait
done
```

Heating and annealing Process.
Should not be changed.

7. Enter desired temperature. Desired temperature is read in from a file that is located one directory before qfile. The file can be named as users preferred. In Hall group, temperature files are named as *temp_"temperature value"* as highlighted in the following figure. Similarly, output files are also renamed corresponding to desired temperature.



```bash
#!/bin/bash

source /usr/local/bin/setupics

cd ../code

mpif90 -Os -xAVX -no-prec-div -r8 -arch host -align dcommons -g -traceback -Dchaptype=1 &
     & -Dnop1=672 -Dnop2=672 -Dchnln1=7 -Dchnln2=7 -Dnumbeads1=28 -Dnumbeads2=28 &
     & -Dnumbin=2000 -Dn_wrap=2 -Ddebugging -Dcanon -Drunr -o ../dmd main.F90

cd ..

temps='050 045 040 035 030 028 026 024 022'
for i in $temps
do ./dmd < temp_$i > out_$i wait
done

for i in {1..20}
do ./dmd < temp_018 > out_018_$i wait
done
```

Desired temperature

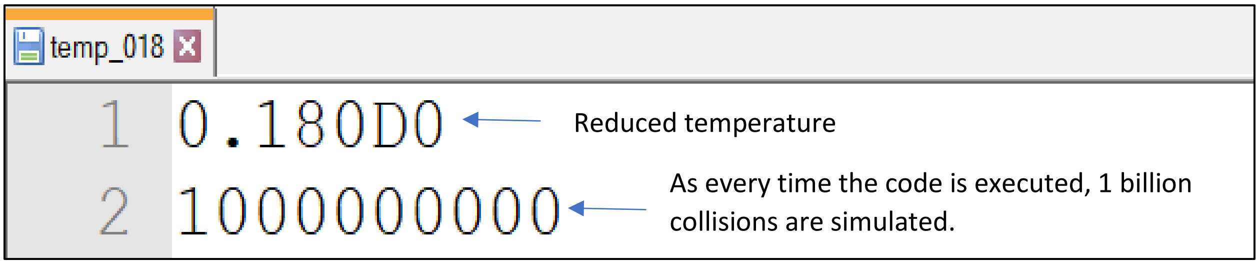
Corresponding output files

Notes:

- All temperatures that are used in PRIME20 are reduced temperatures. The conversion is as follow. Details of temperature conversion can be found in the publication by Wang et al., JBC, 2016 (https://doi.org/10.1074/jbc.M116.744573).

$$real\ temperature\ (in\ Kelvin) = 2288.46 * (reduced\ temperature) - 115.79$$

- A correct temperature file is demonstrated in the following figure.

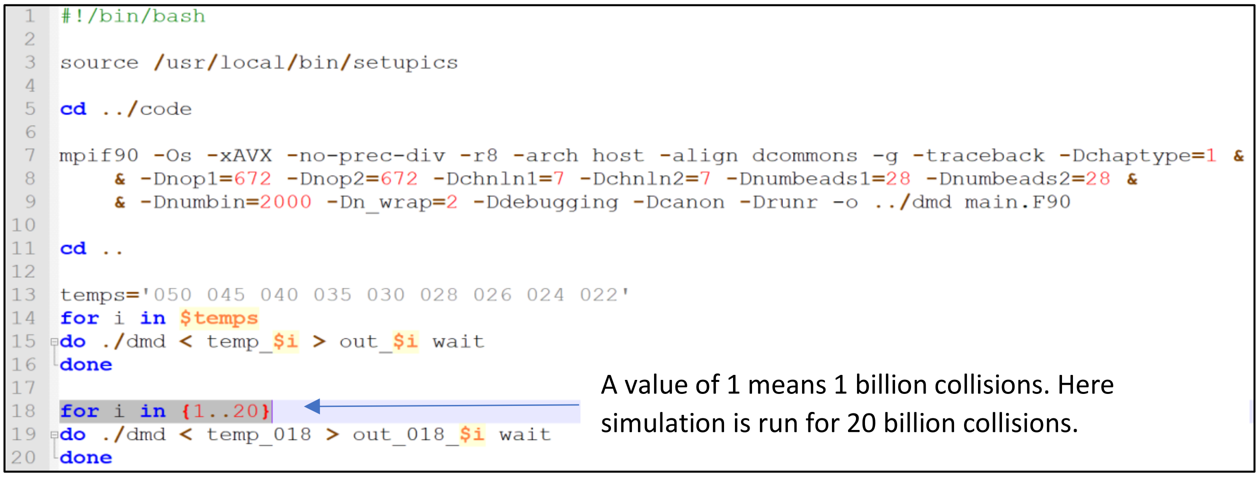


```
0.180D0
1000000000
```

Reduced temperature

As every time the code is executed, 1 billion collisions are simulated.

8. The numbers of collisions are defined by users. Larger system will need longer simulation times.



```
#!/bin/bash

source /usr/local/bin/setupics

cd ../code

mpif90 -Os -xAVX -no-prec-div -r8 -arch host -align dcommons -g -traceback -Dchaptype=1 &
     & -Dnop1=672 -Dnop2=672 -Dchnln1=7 -Dchnln2=7 -Dnumbeads1=28 -Dnumbeads2=28 &
     & -Dnumbin=2000 -Dn_wrap=2 -Ddebugging -Dcanon -Drunr -o ../dmd main.F90

cd ..

temps='050 045 040 035 030 028 026 024 022'
for i in $temps
do ./dmd < temp_$i > out_$i wait
done

for i in {1..20}
do ./dmd < temp_018 > out_018_$i wait
done
```

A value of 1 means 1 billion collisions. Here simulation is run for 20 billion collisions.

***The current code is under development. This user manual only covers the steps to run Parallelized PRIME20 codes. The manual will be updated with more background detailes and result analysis. We will inform users when new version is realeased. Please contact us if you have any questions about the code. We very appreciate feedbacks from users.