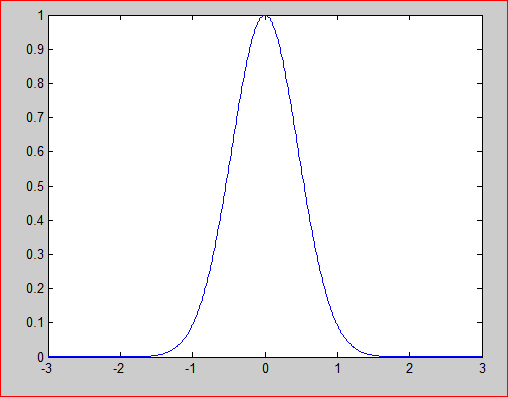
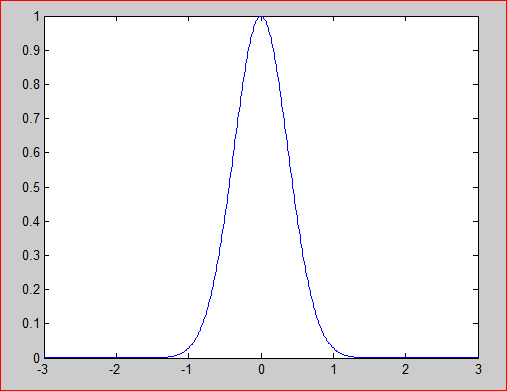
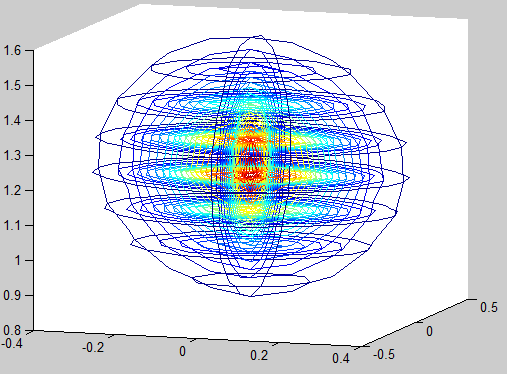
Steps. Week 10 Setting up Simulations of Spatially Homogeneous Relaxation

*Steps to perform various tasks related to dgvlib0d3v topics.*

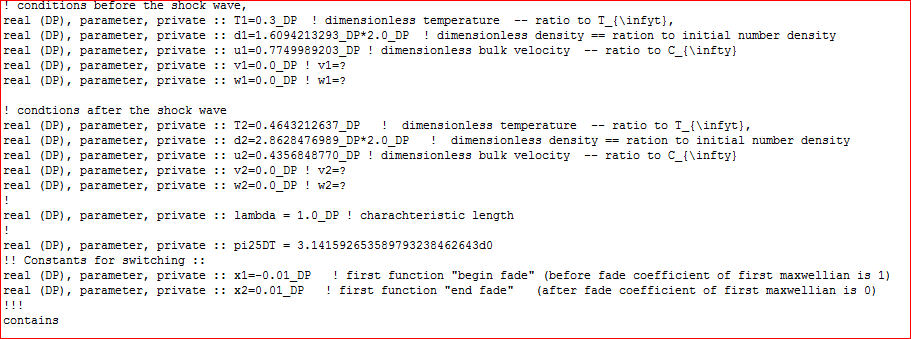
**1. Steps to calculate dimensionless parameters to be used with the DGVlib0d3v code in the spatially homogeneous relaxation problem.**

* DGVlib library uses dimensionless variables. This means that stream conditions need to be converted using the provided formulas to obtain values suitable for simulation. In the following, some guidance is provided to how to calculate dimensionless parameters. We refer to file spat\_hom\_rel\_setup.pdf for the formulas of dimensionless reduction.
* The starting point is to select reference length scale, L, and reference temperature, T\_{\infty}.
  + For simplicity, choose L = 1 m. In the case of spatially homogeneous relaxation, this value is as good as any. For the one and two dimensional problem, L should be chosen to represent characteristic dimensions of the device/process under consideration.
  + To choose T\_{\infty} we follow the following rule: suppose, the gas has temperature \hat{T} Kelvins. According to the formulas of dimensionless If T\_{\infty} is provided, the dimensionless value of the temperature is T=\hat{T}/T\_{\infty}. The dimensionless Maxwellian distribution is   
      
    f(u)=n(\pi T)^{-3/2} exp(-\|u-v\|^2/T)  
      
    A possible way to select the value of the reference temperature T\_{\infty} is by making sure that the dimensionless Maxwellian distribution function f(u) is well contained in the computational domain.   
      
    For example, the dimensional stream has \hat{T}=422.9 K. If we choose T\_{\infty}=1000 K then T=\hat{T}/T\_{infty} = .4229
  + Checking the results of scaling:
    - 1D. A very simple check can be done in 1D. We can select a dimension, say the variable u, and look at the reduced 1D distribution densities. For a 3D Maxwellian with bulk velocity v=(vu,vv,vw) and temperature T, the reduced (marginal) distribution density has the formula  
        
      f(u)=n(\pi T)^{-1/2} exp(-(u-vu)2/T)  
      we can use provided Matlab function **try\_maxwell\_sum.m** to plot the this Maxwellian. All simulations in this course will use domain [-3,3] in velocity variable u. This is a consequence of the fact that values of pre-computed operator A that is provided to you in sol080909/OpA/1xN use these limits. Adjusting parameters of the first Maxwellian in **try\_maxwell\_sum.m** we can visualize a 1D Maxwellian stream with zero bulk velocity and temperature .4229 as the following:   
        
      As we can see, the Maxwellian with zero bulk velocity is well contained in the domain. Of course this still does not guarantee that the entire solution is contained in the domain. But at least we can hope that the solution will be contained in the domain when solution is near continuum.  
        
      Please be aware that if the bulk velocity v is larger than 1.5, then the solution most likely will not be contained in computational domain. In this case, the solution will lose physical meaning. In the case of spatially homogeneous relaxation, things to watch are changes in mass, momentum and energy/temperature. These are often caused by truncating the solution too much. Which happens when solution does not fit into the computational domain.  
        
      If the solution does not fit into the computational domain, one can use a larger value of T\_{\infty}. Here is the result for T\_{\infty}=1500  
        
      
    - 3D. A slightly better check and also a slightly more difficult one is to plot the 3D initial data. The difficulty is, of course, to select a meaningful view of 3D plots. You can try to use the provided Matlab file test\_sf02.m to visualize the initial data using 3D contour plot.
    - In particular, a 3D Maxwellian with temperature, density and bulk velocity given by  
      T1=0.0428571429   
      d1=1.6094213293   
      u1=1.2283828258   
      v1=0.0   
      w1=0.0  
      visualizes as   
        
        
        
      The important thing on this plot is to see that the bulk of the function is within bounds of [-3,3]x[-3,3]x[-3,3]
    - By changing parameters of the plots you can adjust the view of the initial data.
    - By default, the Matlab function plots first Maxwellian, second Maxwellian and then the sum of the two.
* Once T\_{\infty} is selected, we compute the reference velocity value C\_{\infty}=\sqrt(2RT\_{\infty}), where R is the gas constant.
* Once L is selected and C\_{\infty} is computed, we compute reference time Tref time = L/C\_{\infty}
* Once L is selected, we can compute the reference number density from the mass density of the stream. For example, let the density of a stream of argon gas is \rho=2.97E-4 kg/m3.  The reference volume is L3=1 m3. The total mass of the gas in the reference volume is m=2.97E-4 kg. The number of molecules in the reference volume is obtained by dividing the mass of gas by the mass of a single molecule. For argon, n=m/marg = 4.4722e+021. This suggests to choose the reference number density to be N=1.0e+21 so that the normalized number density was less on the order of 1.
* The rest of the dimensionless parameters are computed from N, L, T\_{\infty}, C\_{\infty} using formulas provided in document spat\_hom\_rel\_setup.pdf. In particular for argon with T\_{\infty}=1000, the calculated value of C\_{\infty}=645.18 m/s. If the bulk velocity of the stream is \hat{v}=3.598797619265936e+002, the dimensionless bulk velocity v=\hat{v}/C\_{\infty}=0.5578 and the Maxwellian with dimensionless temperature T=.423 and v=0.5578 is reasonably well contained in the domain [-3,3].

**2. Steps to set up simulations of spatially homogeneous relaxation code.**

You will be performing two types of simulations. In the first type, you will evaluate values of the collision operator on the initial data save them on the drive. In the second type, you will be computing full solution to the problem of spatially homogeneous relaxation. The first type of the simulation is very short. You will only make one evaluation of the collision operator and will stop. The second is about 1000 times longer, since you will be evaluating collision operator on each time step for 500-1000 time steps. The steps below are for the simulation of the Second type. **However they can also be used for the simulation of the first type. So, please use these steps for both problems.**

**Changes to the code.**

* A copy of the DGVlib0d3v code has been placed on the course Canvas page that contain modifications to enable simulation of spatially homogeneous relaxation. Just in case, this is the same code that you used in Project 1 and should still have in directory FFB0D. In this section we will discuss how to set up the code to use different initial data.
* A copy of the code to check the effect of aliasing in the evaluation of the collision operator has been placed on Canvas page. Look for the file Mach3CheckAliasing.zip. The same directions apply to this code as for FFB0D code because up to very minor adjustments, this is the same code.
* Initial data is implemented in modules DGV\_sf02.f90.
* Change entries in file DGV\_sf02.f90 to enable the initial data for the problem of spatially homogeneous relaxation. Refer to file spat\_hom\_rel.pdf for detail on meaning of constants. The following variables need to be set up:   
  **T1,rho1,u1,T2,rho2,u2, x1 and x2,** see example for Mach 155 shock wave in argon gas below. We note that this code uses dimensionless parameters. A brief recall of all the constants is given next  
  
  + **T1, rho1 and u1** are the dimensionless temperature, density and the bulk velocity of the upstream Maxwellian stream. Downstream and upstream conditions for a few shock waves were provided in the file: **dimensional\_shock\_wave\_conditions.docx**
  + **T2, rho2 and u2** are the temperature, density and the bulk velocity of the downstream Maxwellian Stream condition.
  + **x1 and x2** are the endpoints in the piece wise function determining the initial data. For x<x1 the gas property are given by the upstream conditions. For x>x2 by the downstream conditions and for x1<x<x2 the two streams are superimposed linearly.
* To set up a different initial data which is a sum of two Maxwellian streams follow the steps described in the previous section on completing dimensionless reduction.
  + Please note that you select reference values only once and use it for both streams.
  + Once the dimensionless values are obtained, use the above template to set **T1,rho1,u1,T2,rho2,u2, x1 and x2,** to the new values.
  + Re-compile the code.

**3. Steps to set up simulations of spatially homogeneous relaxation code.  
  
Parameters Files**  
NOTE: Again, the directions below apply for both types of simulations in Project 2.

* The OpenMP version of the code uses DGVparameters.dat and parameters.dat files. The OpenMP 0D driver is very picky about different directories where different files are written. Following the provided example will help.
* The values that need to be specified in DGVparameters.dat and parameters.dat depend on the type of the simulation and also on the specifics of the simulation. The followings is an example of entries in parameters.dat suitable for Mach 1.55 simulation in argon. Please note that the values in parameters.dat should be in agreement with those in sf02\_mod.f90. This means, in particular, that it is possible to have very different values for variables and still being computing Mach 1.55 wave in argon. You only would like to change the values that are in blue  
    
  + OpenMP: You only want to change the parameters that are in blue

**DGVparameters.dat**:

!!!!!!!!!!!!!!!!!!!!!!!!  
left endpoint in u = -3.0 !! u\_L  
right endpoint in u = 3.0 !! u\_R  
left endpoint in v = -3.0 !! v\_L  
right endpoint in v = 3.0 !! v\_R  
left endpoint in w = -3.0 !! w\_L  
right endpoint in w = 3.0 !! w\_R  
uniform mesh in u = yes !! mesh\_u\_uniform  
uniform mesh in v = yes !! mesh\_v\_uniform  
uniform mesh in w = yes !! mesh\_w\_uniform

use secondary velocity grid = no !! if this parameter is selected provide   
at least two numbers for each list below: the first number will determine the primary grid and the second number. !! determines the secondary grid. The third (also fourth and fifth) numbers are optional

!! and is not used at the moment

!set the parameters of the primary and secondary grid

number of cells in u = 15,15,4 !! Mu\_list  
degree of local Lagrange basis in u = 1,1,1 !! su\_list  
number of cells in v = 15,15,4 !! Mv\_list  
degree of local Lagrange basis in v = 1,1,1 !! sv\_list  
number of cells in w = 15,15,4 !! Mw\_list  
degree of local Lagrange basis in w = 1,1,1 !! sw\_list

! directory and the base name for the solution  
current solution save directory = sol080909/ !! needs a comment, otherwise includes CR in the name  
current solution base name = M155\_ !!  
current A operator base name = OpA/1xN/exp0912\_ !! chunk reader will add ch000\_ to the name (max 20 symbols)  
secondary operator A base name = OpA/1xN/exp0912\_ !!  
num of chunks for Aarray = 1 !! numchnks --- the total number of files in which the Aarray is stored,  
num of chunks for sec Aarray = 1 !! numchnks II -- the total number of files that contain the Aarray for the secondary mesh

! Parameters related to the integration of moments

gauss order for moments in u = 9 ! moments\_u\_gauss\_order  
mesh refinement in u for moments = 1 ! moments\_refine\_u

! Parameters of Calculating the A-array:

cutoff radius of A-array = 3.00 ! Trad -- velocities xi1 and xi that farther apart than Arad are neglected   
error in integral in Chi = 1.0E-4 ! ErrChi -- error in the outside double integral  
error in integral in Epsilon = 1.0E-5 ! ErrEps -- error in the inside double integral  
cutoff values for A-array = 1.0E-11 ! min\_sens -- the values of A-array below thi slevel are not accepted   
list of basis functions for A-array = 365,14896 ! I1\_list -- list of basis functions numbers for which A-array will be calculated!  
range of node numbers I2 = 1,729 ! I2\_from and I2\_to - the range of the indices in I2. Values ignored if I1\_list has more then 2 records

! OpenMP parameters: (can ignore if not using OpenMP)

number of OMP threads = 28 ! Num\_OMP\_threads

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
! Model threshholds: The order in which the models are envoked are   
! 1) the Boltzmann equation, non-split formulation   
! 2) the Boltzmann equation, split f=f\_M + df, formulation  
! 3) the linearized Boltzmann equation (unless solving spatially homogeneous problem, does not give an advantage)  
! 4) the BGK-type Model with velocity dependent collision frequency  
! 5) the ES-BGK/BGK or Shakhov model

! the parameters below define the thresholds for switching between models. They represent the maximum allowed values of the L1-norm of the   
! deviation of the solution from the local maxwellian. When the L1-norm of the deviation drops below the treshhold, the appropriate model is envoked.   
! NOTICE: keep linear\_lev at least an order of magnitude less than decomp\_lev!  
! VERY IMPORTANT: the tresholds for the models above should be given in decreasing order.   
! Otherwise the algorithms will make a mistake.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

error maxwell ESBGK = 0.00001 ! ES\_lev  
error maxwell Vel ESBGK = 0.0001 ! vel\_lev -- velocity dependent regime  
error maxwell linearization = 0.001 ! linear\_lev,   
error maxwell decomposition = 0.03 ! decomp\_lev

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Parameters for dimensionless reduction  
!! C\_inf = thermal velocity in m/s   
! L\_inf = characteristic length in m  
! N\_inf = the total number of molecules in the volume  
! T\_inf = the normalization for time is selected from the condition T\_inf\*C\_inf = L\_inf calculated automatically  
! Temp\_inf = the characteristic temperature  
! mol\_diam = molecular diameter in hard shperes model  
!!!!!!!!!!!!!!!

!ref termal velocity = 2.040255509841958E+03 ! C\_inf m/s Mach 6.50  
ref termal velocity = 6.451854419808672E+02 ! Mach 1.55  
ref characteristic length = 1.0 ! L\_inf m  
ref number of molecules = 1.0E+21 ! N\_inf   
ref molecular diameter = 3.76E-10 ! mod\_diam m-

!!!!!!!!!!!!!!!

! Parameters of the gas and ES-BGK  
the ordinary gas constant = 208.1322 ! gasR (=208.1322\_DP for Ar; =296.95 for N\_2)  
gas alpha = .5 !! gas\_alpha (when compare to Boltzmann hard spheres use .5)  
gas viscosity = 5.337379160633744E-005 !! gas\_viscosity  
Temperature reference = 2000 !! gas\_T\_reference  
ESBGK modifier = -0.5 !! alpha for prandtl number, -0.5 corresponds to 2/3 , 0 to BGK if ES model is used.

!!!!!!!!!!!!!!!!!!!!!!!

!! Parameters VD-BGK (BGK model with velocity dependent collision frequency  
N of enforced moments = 6 !! Order -- the variable determining how many or the built-in moments are enforced in relaxation  
! !! order=1 invokes the first enforced moment. Current allowed Maximum is 15 moments total  
N of coefficients in VDCF = 6 !! Order\_nu -- the number of coefficients/basis functions in the representation of the velocity-dependent collision  
! !! frequency ATTENTION: Currently only the same number if allowed for both Order and Order\_nu  
mean free time rates update = 0.5 !! mft\_coeff -- time period in mean free times between eadh

! end of file

* Remarks: number of cells in u,v,w and degree of local Lagrange basis in u,v,w
  + Each one of these parameters is a list. The first entry from each list is used to set up the primary mesh in velocity variable. The second entry from each list is used to set up secondary mesh. Other entries are not used.
  + Before changing the parameters, please make sure to use values for which pre-computed operators A have been provided.
  + You may want to estimate the time it will take to run the code when changing the numbers of cells.
* Remarks: error maxwell ESBGK, Vel ESBGK, linearization, decomposition:
  + These values determine which model is used in simulation. DGVlib OpenMP implements several models in it. At each time step the L1 norm of the difference between the solution and the local Maxwellian is computed. The value of this difference determines how far from the continuum is the solution. Depending on the value, different collision models are used.
  + To make sure the code works correctly, please make sure that   
    error maxwell ESBGK < Vel ESBGK < linearization < decomposition

Parameters.dat

!!!!!!!!!!!!!!!!!!!!!!!!  
degree of local Legendre basis in x = 4,5 !! k\_c\_list   
number of G-L boundary nodes in x = 5,6 !! k\_b\_list   
left endpoint in x = 0.0 !! x\_left   
right endpoint in x = 0.1 !! x\_right  
uniform mesh in x = yes !! mesh\_x\_uniform  
number of cells in x = 4,6,8,11,16 !! N\_list

degree of local Legendre basis in t = 4,5 !! d\_c\_list  
number of G-L boundary nodes in t = 5,6 !! d\_b\_list

! choose from different kinds of boundary conditions:   
! 1 - periodic BCs  
! 2 - exact BCs (will need to know the selected exact solution  
! 3 - diffusive reflection BCs.   
!

! select the type of boundary conditions   
conditions on the left boundary = 3 !! selected\_leftbcond   
conditions on the right boundary = 3 !! selected\_rightbcond

! choose from different kinds of exact solutions:   
! 1 - sin(x+u) -- type of source  
! 2 - ext(-1/x^2) -- type of source   
! 3 - diffusion BCs

! Select the exact solution  
! 1 = "f(ut-nx+cu)", f(x)=a\*\exp(-b/((x-r\_1)(x-r\_2))) zero right side exact solution (or 1a = 1- cos(x) periodic wave)   
! 1(a) = "f(ut-nx+cu)", f(x)=1-cos(2\alpha\pi x) zero right side exact solution  
! 2 = maxwelveldist (T1,u1,u)\*N1 + maxwelveldist (T2,u2,u)\*(1-N1) --- mimic shock wave bimodal distribution (time independent)  
! function sf\_02 only makes sense for initial data. Error is replaced by L\_2 norm

type of exact solution = 2 !! selected\_exact\_sol

! directory and the base name for the solution  
current solution save directory = sol080909/ !! needs a comment, otherwise includes CR in the name  
current solution base name = M155\_ !!

! Parameters of time evolution

initial time = 0.0 !! beginning value of time (t\_L)   
final time = 0.004 !! final value of time (t\_R) (USC Mach 10: 0.00025  
time step = 0.000005 !! value of the time step dt

instances to evaluate error = 500 !! num\_eval\_error = how many (evenly spaced in time) evaluations of error  
instances to save solution = 10 !! num\_save\_sol = how many times the solution (and error and other quantities) is saved during the evolution

! Parameters related to non-unifom mesh generation  
type of nonuniform mesh in x = 2 ! x\_nonuniform\_mesh\_type

! supported types of nonuminform mesh:  
! For variable u:  
! 1 --- the mesh is build based on the gauss nodes used for the integration of moments, moments\_x\_gauss\_nodes and   
! moments\_u\_gauss\_nodes. intervals [x\_left,x\_right], [u\_left,u\_right] is divided in subintervals   
! as to have gauss nodes at centerpoints. Some extra points need to be introduced to make this possible.  
! 2 --- this mesh is to be used with the diffusive boundary conditions. The velocity of the wall (currently only u\_{w}=0)  
! will be included in the mesh. Also, the cell near the walls will be 1/8 - 1/4 - 1/2   
! 3 -- for variable u -- this mesh will have small cells surrounding u=0 as prescribed by parameters  
! sml -- small cell levels amd smr -- small cell refinement factor   
! 3 --- This is a non-uniform mesh in "x" with cells near wall be 1/4-1/2. Currenlty is not supported for meshes in "u"

! Parameters related to the integration of moments

gauss order for moments in x = 5 ! moments\_x\_gauss\_order  
mesh refinement in x for moments = 1 ! moments\_refine\_x

! Parameters of diffusive BCs -- only used with diffusive BCs

temperature of the left wall = 1.0 ! T\_w\_left -- temperature of the wall on the left  
temperature of the right wall = 3.0 ! T\_w\_right -- temperature of the wall on the right

! Restart of the solution

solution restart = no ! need\_to\_restart yes to restart  
restart time = 0.0120000000 ! restart\_time\_txt restart/last saved solution time in text format --- use the exact numbers from the filename!

! OpenMP parameters:   
number of OMP threads = 1 ! Num\_OMP\_threads -- this variable is duplicated in the DGV-library --- be careful...

! number of OMP threads is defined in the DGVparameters.dat for 0D3V driver.

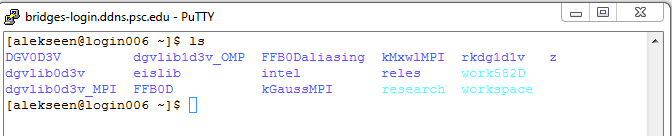
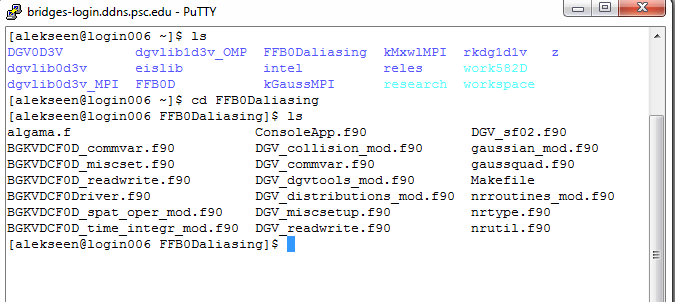
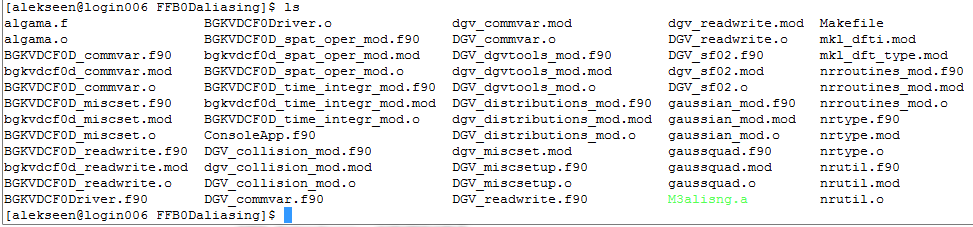
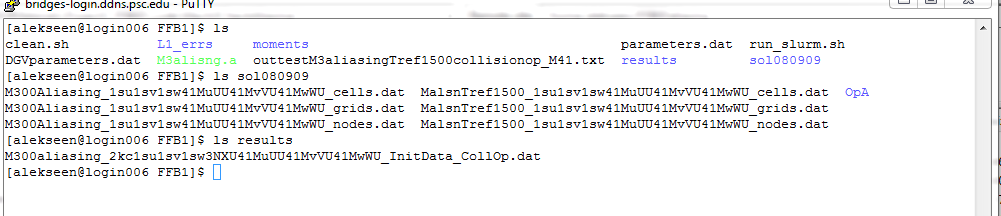
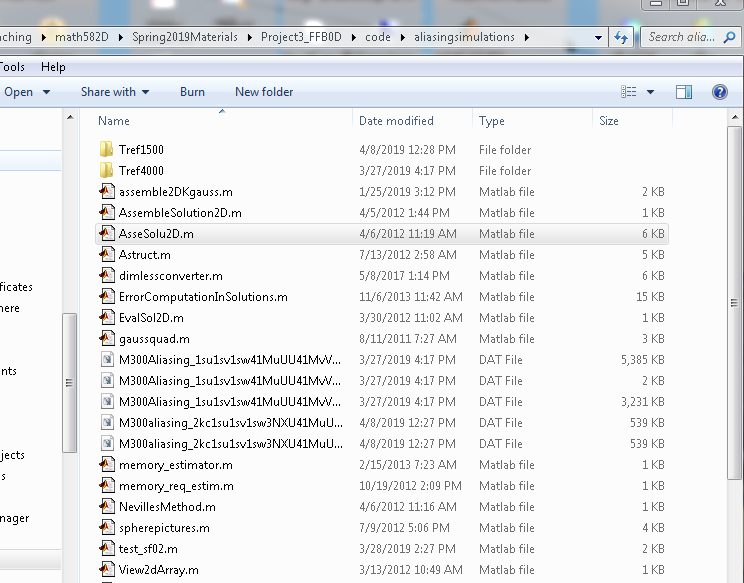
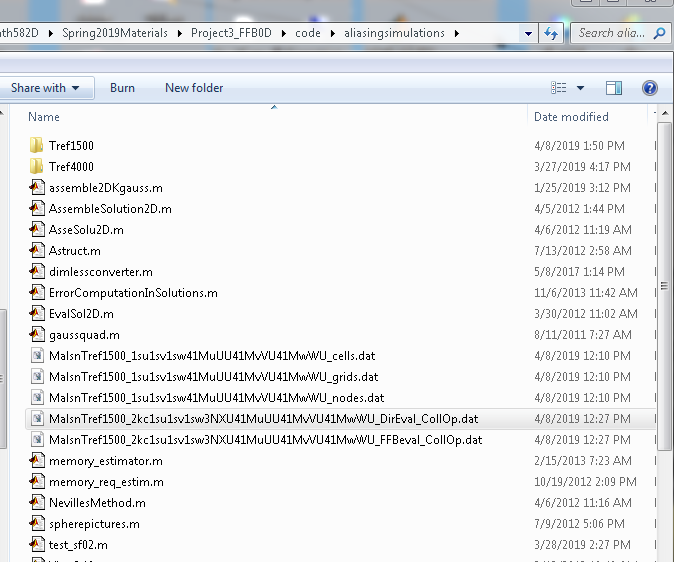
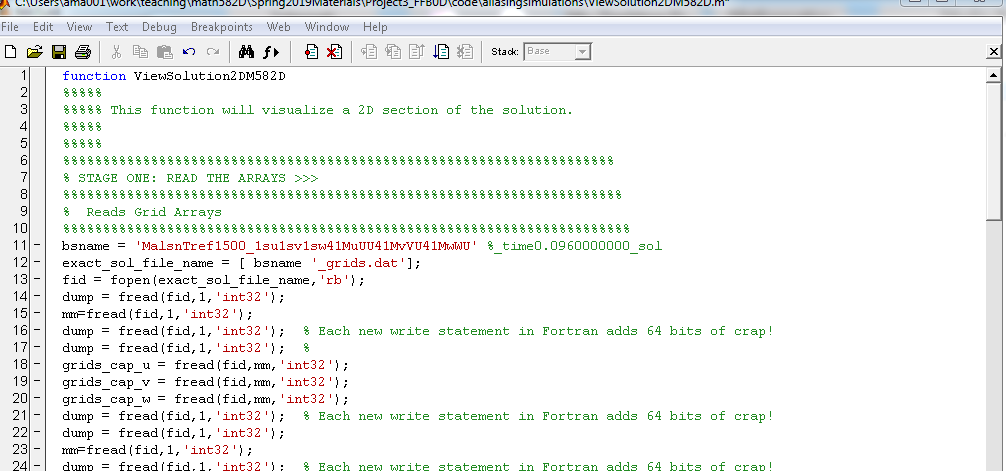
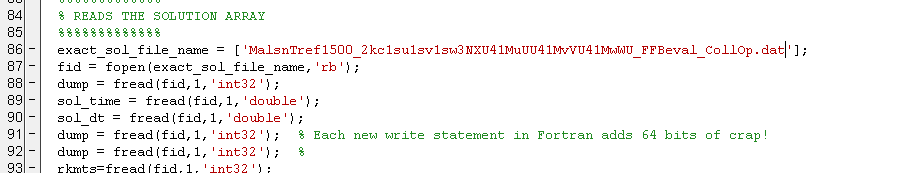
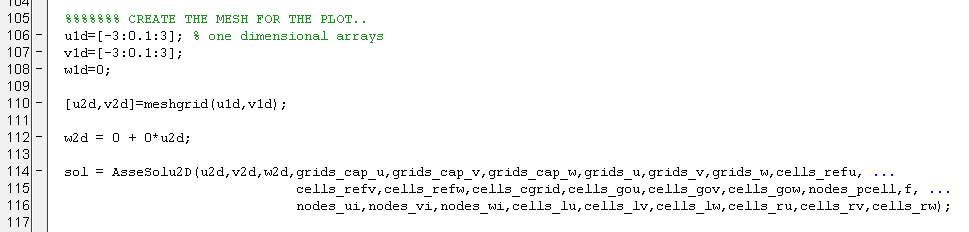
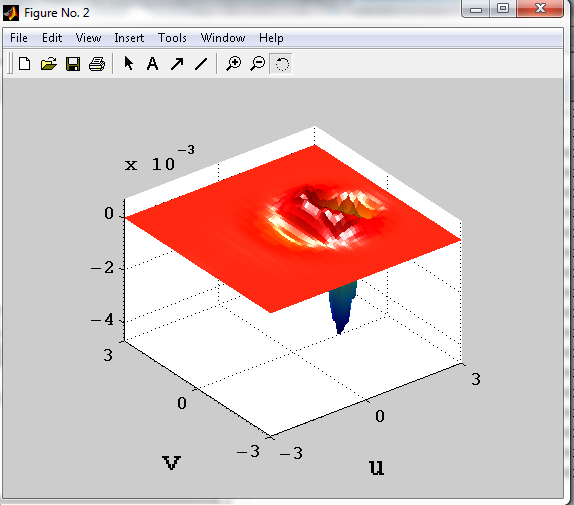
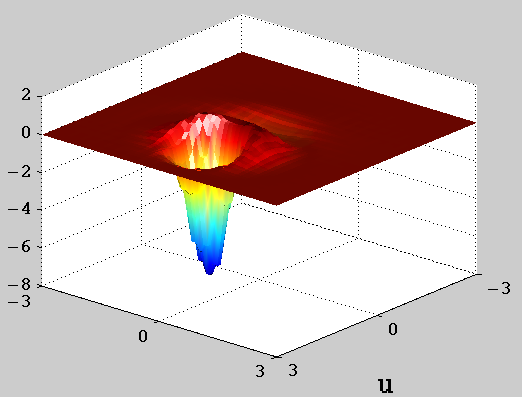
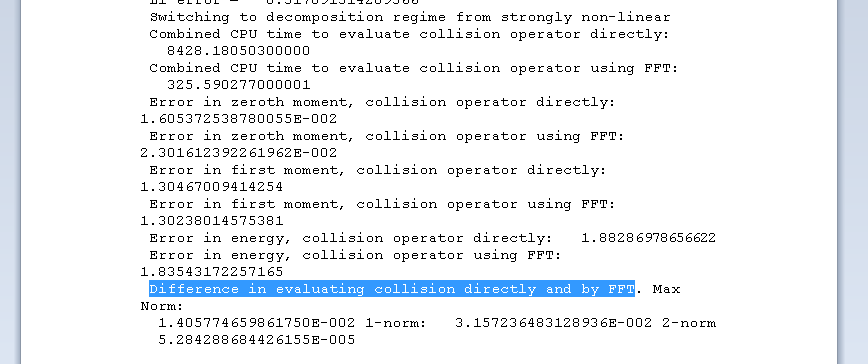
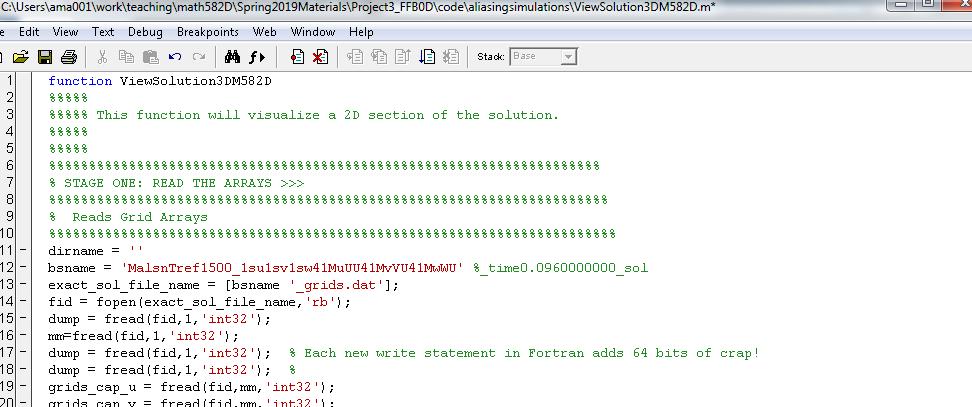
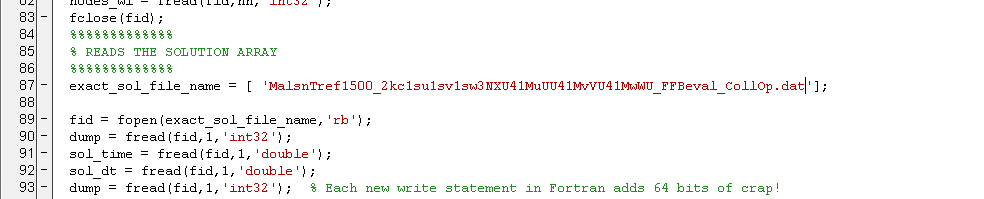
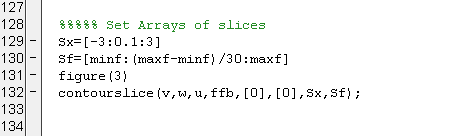
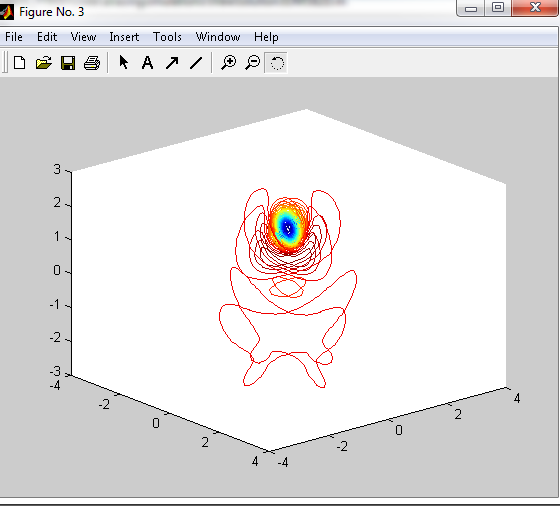
! end of file

* Remarks: final time   
  To determine a good value of the final time, first calculate the mean free time for the combined stream. Most of the relaxation happens in 10-15 mean free times. This will give you a good estimate for the finite time in seconds. Divide this by Treftime to compute the final time in dimensionless units.
* Remarks: solution restart and restart time  
  You may use this parameters lines to force the program to continue the solution from a saved point rather than start from the beginning. This option becomes very handy when simulations take a long time and you can not finish the simulation in the allocated time. To invoke the option, find the saved solutions. For OpenMP version of the code the saved solution are in the directory “results” and for the MPI version of the code the saved solutions are in “sol080909”.   
  The saved solutions can be recognized by their name that bear the time stamp, e.g.,   
    
  M650MX5dt40\_2kc1su1sv1sw3NXU21MuUU21MvVU21MwWU\_time0.0017600000\_sol.dat  
    
  Note that “0.0017600000” is the value of time when the solution is saved.   
    
  To restart the simulation, make sure all parameters are the same as they were to produce the solution file. Then use option   
    
  solution restart =yes  
    
  and set the time value from the file name to be the value of the restart time. Make sure to copy all zeros so that it appears exactly as in the file name:

restart time = 0.0017600000

**4. Steps to perform numerical experiment to test the effect of aliasing on the evaluation of the Collision operator.**

The goal of these numerical experiments is two-fold. First, we will observe the effect of aliasing on the evaluation of the collision operator. Second, by comparing results, we will identify scaling that is free from aliasing, or at least sufficiently free. This scaling is then will be used in simulation of the problem of spatially homogeneous relaxation.

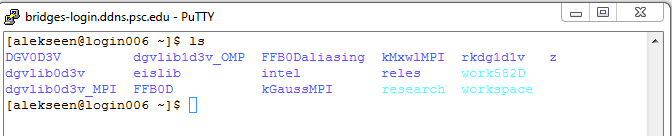
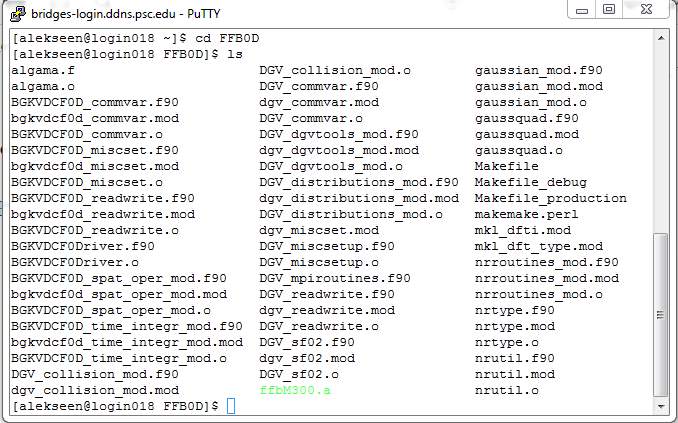
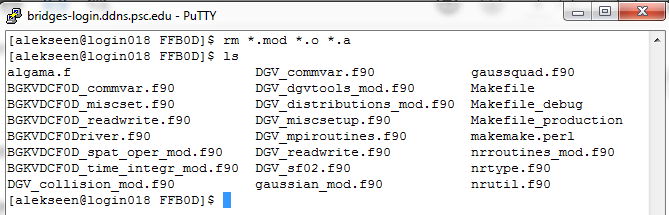
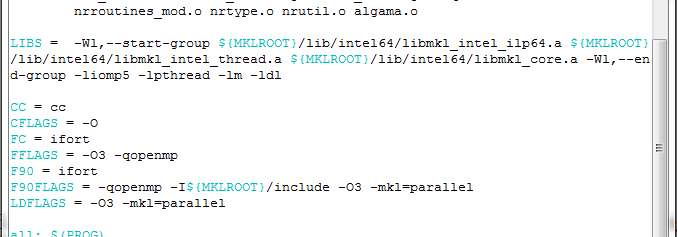
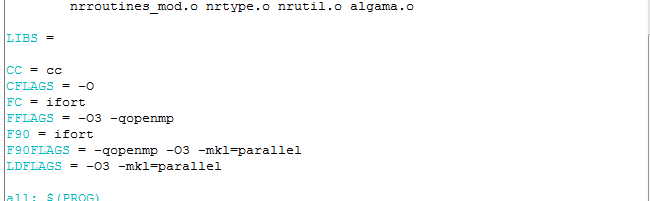
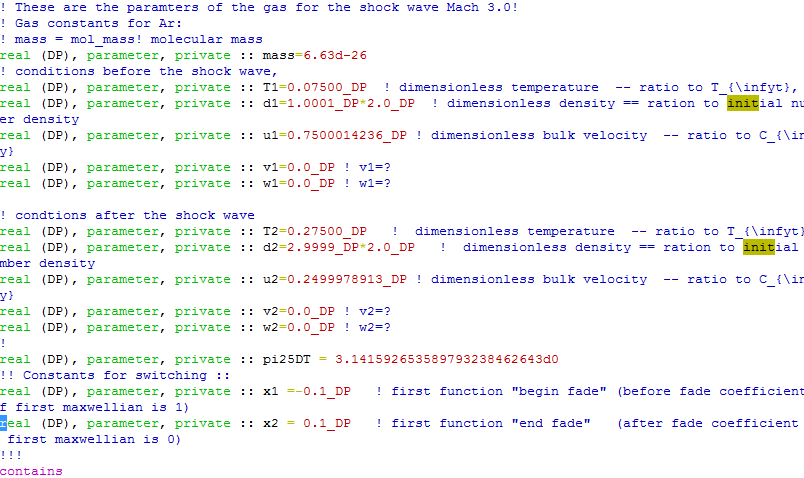
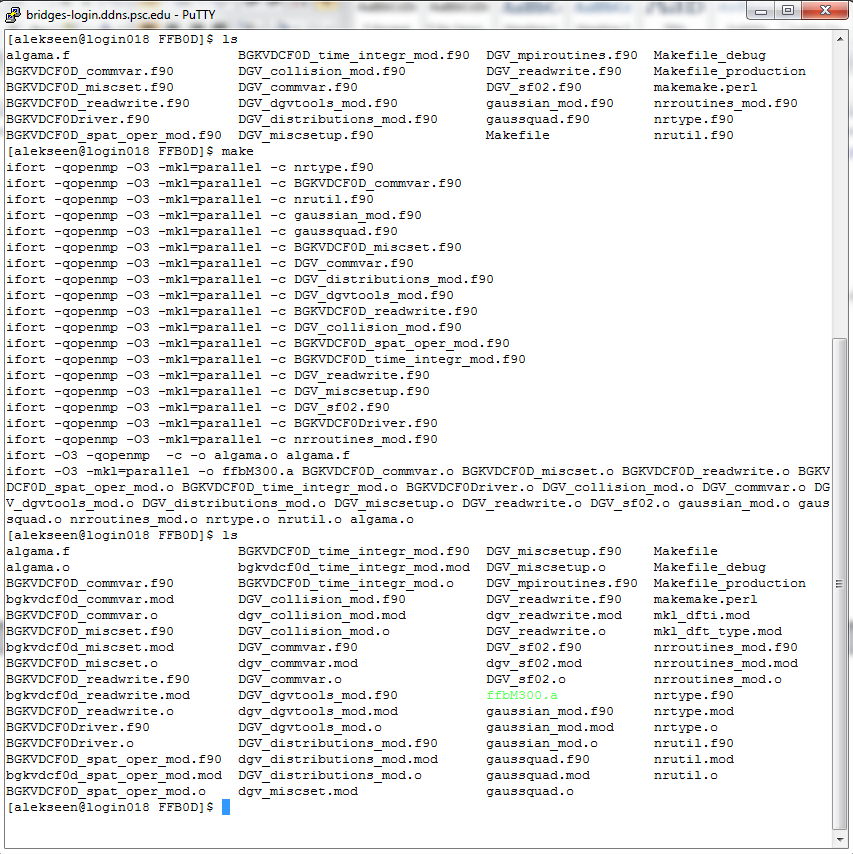
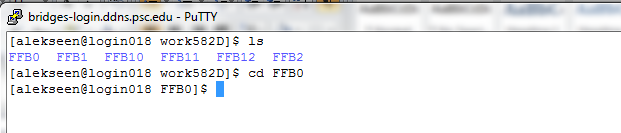
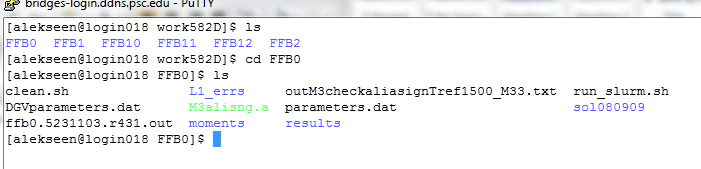
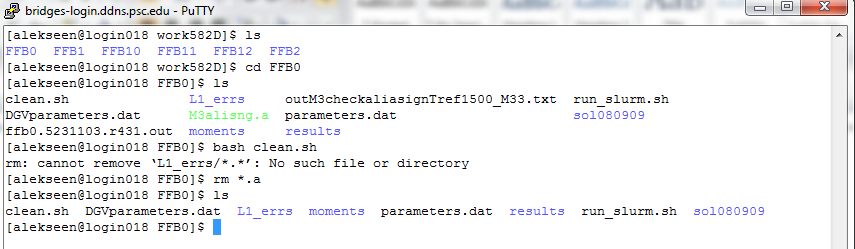
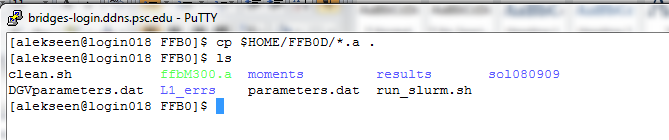
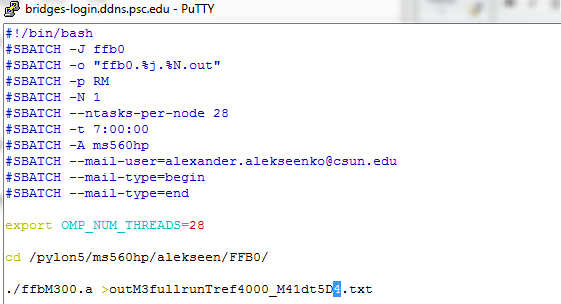
* Locate the word document file that contains dimensional data for mixtures of Maxwellian streams. The file’s name is **dimensional\_shock\_wave\_conditions.docx** and it is posted on Canvas.
* Identify the conditions for Mach 3 shock wave. These are the parameters we will be practicing with.
* Prepare a spreadsheet file that allows to calculate dimensionless values for the mixture of Maxwellian streams. Follow the steps in Part 1 and also refer to **spat\_hom\_rel\_setup.pdf** that is available on Canvas.
* Prepare several sets of parameters using different reference temperatures: 1000K, 1500 K, 2500 K and 4000 K. The rest of the reference values are kept the same and are Ninf=1.0d+20, L = 1m
* Download a copy of the code that is prepared to test the effect of aliasing **Mach3CheckAliasing.zip**. Create a folder in your $HOME file space with the name **FFB0Daliasing** and copy the code in this directory.   
  
* Switch to the directory FFB0aliasing and
  + 
  + Adjust the file **DGV\_sf02.f90** as is described earlier to specify the desired dimensionless parameters.
  + OPTIONAL: if you want to “switch off” direct evaluation of the collision operator, open in vim the file **DGV\_collision\_mod.f90** and go to line **3593** and comment the line:   
      
    ! call EvalCollisionPeriodicMixedTermsA\_DGV\_OMP(f-Df/2.0\_DP,Df,fcol\_scr)   
      
    This can save you time in class, since evaluation for the collision directly is very slow. However, you will need to re-run this simulations later. So, you will have to uncomment the line to do so.
  + Type **make** and compile the code. The default name for the executable is M3alisng.a  
      
    
* Prepare a work directory. This includes cleaning all output, copying the executable, adjusting the run\_slurm.sh, DGVparameters.dat
  + Adjust the **DGVparameters.dat** as is described above to specify the reference values.
  + IMPORTANT: IN **DGVparameters.dat** and **parameters.dat** include the reference temperature in the solution base file name.
  + Use Mu, Mv, Mw=41 in the DGVparameters.dat for these simulations – it will make for a better picture.
  + Only one parameter is important in **parameters.dat** for this simulation, the base solution name. Please include the information on the reference temperature in the solution base name, e.g.   
      
    MalsnTref1500\_
  + In run\_slurm.sh you would request the same amount of time as you needed to test complexity for M =41
  + IMPORTANT: In run\_slurm.sh you will need to specify the name of the output file. Make sure the name of the output file includes value of reference temperature, e.g.,  
      
     **outtestM3aliasingTref1500collisionop\_M41.txt**  
    to mark that the simulation uses reference temperature 1500 K.
* Run the code. After you run the code, you should have the following output:  
    
  
* Collect and save on your computer:   
    
  **run\_slurm.sh**  
  **DGVparameters.dat**,  
  The output file, e.g., **outtestM3aliasingTref1500collisionop\_M41.txt**The \_cells.dat, \_grids.dat and \_nodes.  
  The content of folder results
* It is recommended that you create a folder for these simulations, say, **aliasingsimulations**, and that you copy results of each simulation are stored in a separate subfolder, e.g.,   
  **Tref1500**, **Tref4000**, etc.  
    
  .
* Once a simulation is completed, we can visualize the result using Matlab subroutines provided in the Project 1 (Week 3). The files are located in the ZIP file **FFB0D\_VIZ.zip**. Copy files contained in   
  **FFB0D\_VIZ.zip** into the folder **aliasingsimulations.** Next, please copy the “\_grids.dat”, “\_cells.dat” and “\_nodes.dat” and the collision operator output “\_CollOp.dat” files from the simulation result subfolders and place them next to the Matlab files (in their parent directory **aliasingsimulations**)  
    
  
* Start Matlab and select **aliasingsimulations** to be its “**current directory**”. You can visualize the output of the collision operator using either **ViewSolution2DM582D.m** or **ViewSolution3DM582D.m**. Open both files in Matlab editor. We will go over the steps to plot the output of the collision operator next.
* Making 2D plots of the collision operator using **ViewSolution2DM582D.m**
  + Select **aliasingsimulations** to be the Matlab current directory and open **ViewSolution2DM582D.m** in the Matlab editor.
  + In line 11, specify the common part of the name for the “\_grids.dat” , “\_cells.dat” and “\_nodes.dat” files that came with your solution. For the files seen in the previous screen shot it will be  **MalsnTref1500\_1su1sv1sw41MuUU41MvVU41MwWU**  
      
    
  + Next go to line 86 and specify the name of the file that you want to visualize. It has to be one of the files containing the output of the collision operator. Make sure the file is consistent with the “\_grids.dat” , “\_cells.dat” and “\_nodes.dat” files specified in line 11. It has to use the same number of cells and nodes. In this example it is M=41 cells and s=1 node per cell. In the screen shot above we have two files that we can visualize:   
      
    MalsnTref1500\_2kc1su1sv1sw3NXU41MuUU41MvVU41MwWU\_DirEval\_CollOp.dat  
    and   
    MalsnTref1500\_2kc1su1sv1sw3NXU41MuUU41MvVU41MwWU\_FFBeval\_CollOp.dat  
    We copy their names in the line 86:   
      
    
  + Next we go to lines 106—107 and select the 2D mesh to make a plot. In particular, we need to specify a bunch of points in variable u, say u\_i, and bunch of points in variable v, say v\_j. Then the code will interpolate the solution (the output of collision operator) at all points (u\_i,v\_j). The arrays u1d and v1d contain points u\_i and v\_j, respectively. We keep w1d=0 in line 108.   
      
      
    In particular in the example below we selected uniformly spaced points between -3 and 3 with the step size .1.
  + Run the code to produce a plot of the collision operator  
      
    
  + Chang the name of the output file to plot in line 86, e.g., MalsnTref1500\_2kc1su1sv1sw3NXU41MuUU41MvVU41MwWU\_FFBeval\_CollOp.dat, and change the output window of the plot in line 118 say to figure (3). Run the code again to obtain a different plot.   
    
  + You can also take a look at the output file where the max norm of the difference between direct and FFT based evaluation is compared. Open the output file for this simulations in any text editor that does not object to Linux end of line symbol. In Windows, it will be Wordpad rather than Notepad. Look for the key phrase “Difference in evaluating collision directly and by FFT”  
      
      
      
    The first number is the max norm of the difference. In the case of files on plots above, the difference is about 0.014 and should be visible on the plots, albeit not easily.
* Making 3D plots of the collision operator using **ViewSolution3DM582D.m**
  + Select **aliasingsimulations** to be the Matlab current directory and open **ViewSolution3DM582D.m** in the Matlab editor.
  + In line 12, specify the common part of the name for the “\_grids.dat” , “\_cells.dat” and “\_nodes.dat” files that came with your solution. For the files seen in the previous screen shot it will be **MalsnTref1500\_1su1sv1sw41MuUU41MvVU41MwWU  
    **
  + Next go to line 87 and specify the name of the file that you want to visualize. It has to be one of the files containing the output of the collision operator. Make sure the file is consistent with the “\_grids.dat” , “\_cells.dat” and “\_nodes.dat” files specified in line 11. It has to use the same number of cells and nodes. In this example it is M=41 cells and s=1 node per cell. In the screen shot above we have two files that we can visualize:   
      
    MalsnTref1500\_2kc1su1sv1sw3NXU41MuUU41MvVU41MwWU\_DirEval\_CollOp.dat  
    and   
    MalsnTref1500\_2kc1su1sv1sw3NXU41MuUU41MvVU41MwWU\_FFBeval\_CollOp.dat  
    We copy their names in the line 87:  
      
    
  + Adjust lines 129 and 130 and possibly 132 to achieve desired look of the solution. You are strongly advised to learn about use of Matlab’s countourslice.  
      
    
  + Run the code to visualize the output of the collision operator. You will need to rotate the figure to pick a good view.  
      
    

**Steps to submit an DGV0D3V run into execution**

* All production runs are submitted to execution through the SLRUM scheduler. To submit the job, use command  
    
  **sbatch run-slurm.sh**
* To check the status of the submitted job, use the command   
    
  **squeue –u username**
* To delete a submitted job, including terminating execution of a submitted job use command:  
    
  **scancel #job**  
    
  The number of the job is determined by running **squeue –u username** command.

**5. Steps to perform numerical experiment of spatially homogeneous relaxation test the effect of aliasing on the evaluation of the Collision operator.**

The goal of these numerical experiments is to obtain a solution to the problem of spatially homogeneous relaxation and to compare it to the DSMC solution provided by Ian Boyd. The results of the previous simulation will used to select the dimensionless reduction to use.

* Use the results of the previous numerical study to select a value of the reference temperature in dimensionless reduction. A successful value should not be too high, because in this case the solution will not be resolved by 41x41x41 mesh. Also the value should not be too low to avoid aliasing. Pick a value among 1000 K, 1500 K, 2500 K, or other (but not 4000 K, since reserved by the instructor) you tried using these considerations and use it as Tref. The rest of the reference values are kept the same and are Ninf=1.0d+20, L = 1m
* Download a copy of the code that is prepared for running the problem of spatially homogeneous relaxation. **FFBOpenMP\_working\_Bridges02062019.zip** We used this code in Project 1. You may still have the folder **FFB0D** in your **$HOME** directorycontaining the code**.** If not sure that you have the correct copy of the code, delete the folder using command   
    
  rm –r FFB0D  
    
  Then create the folder **FFB0D** in your **$HOME** directory, again, get a clan copy of the code and move the files into folder **FFB0D** and copy the code in this directory.   
  
* Preparing the executable. Switch to the directory FFB0.
  + You probably still have all the \*.o and \*.mod files that are produced during compilation.  
      
      
      
    If you deleted the directory and made a fresh copy of the files move to the next step. Otherwise, delete the \*.o and \*.mod files using   
      
    rm \*.o \*.mod  
      
    you should also delete the executable, just in case.  
      
    rm \*.a
  + Once the folder is cleaned, you only have the code source files.   
    
  + OPTIONAL: Recently, Bridges configuration has changed. Either download the updated Makefile and copy it into this directory. Or manually to adjust the Makefile. To manually adjust the Makefile, open Makefile in vim editor. We need to make changes in the line   
      
    LIBS =   
      
    and in the line   
      
    F90FLAGS =   
      
    This is the lines before the change:  
      
      
      
    These are the same lines after the change.   
      
      
      
    make adjustments and save the file.
  + Next we will adjust the file **DGV\_sf02.f90** as is described earlier to specify the desired dimensionless parameters of the Maxwellian streams. Use the values that correspond to your desired temperature.   
      
    after the streams parameters were updated, save file and exit the editor.
  + Use the command make to produce an executable. The default name for the executable is **ffbM300.a  
      
    **
* Preparing the work directory.
  + Switch to the workspace. Go to the either of the work directories your created, or make a new one by cloning one of the existing ones. Suppose it will be FFB0:  
    
  + Inspect the directory:  
      
    If you used this directory before to run your simulations, your directory will contain files that you do not need. We will delete all of these files, including the executable:   
      
    bash clean.sh  
      
    rm \*.a  
    
  + We copy the executable that you just prepared in $HOME/FFB0D/ to this directory :   
      
    cp $HOME/FFB0D/\*.a .  
      
    
  + Next follow the directions to modify DGVparameters.dat and parameters.dat.
  + Finally, we need to modify run\_slurm.sh. Make sure that the correct executable is called. Also, make sure to include the reference temperature, number of cells in the name of the output file. You can also include information about the size of the time step.   
    
  + Submit the code for execution.

**6. Steps to organize results of simulations on a local computer.**

* Create a directory that will contain all simulations that you will perform using DGVlib0D3V code. Copy the matlab files that contain visualization subroutines into this directory. Create one or more subdirectories to organize simulation results.
  + Ideally, each group of simulations goes into separate folders.
  + Please include all output files produced by the DGVlib0D3V software and
  + Please include the runtime output file \*.txt
  + Prepare a copy of the sf02.f90, DGV\_sf02.f90 files, Makefile or similar that were substantially modified to perform the numerical simulation
  + You may consider adding a Readme file that gives any additional detail about the simulations in the folder