

LeMANS-MOPARMD Version 1.0 User Manual

Developed by the research group of Professor Iain D. Boyd
Manual updated by (listed in order):
Abhilasha Anna, Jonathan Wiebenga, Peter Cross, Erin Farbar

Send bug reports to: iainboyd@umich.edu

Last Updated: January 7, 2016

Contents

1	Overview	2
2	Current limitations	2
3	Getting started	3
4	Obtaining a Copy of LeMANS-MOPARMD	3
5	Compiling	4
6	Mesh Generation	4
7	Input files	4
7.1	LeMANS	4
7.1.1	prob_setup.dat	4
7.2	MOPARMD	7
7.2.1	mopar.nml	7
7.2.1.1	Notes on Input Files for Ablation	10
7.2.1.2	Notes on Input Files for Structural Calculations	10
7.2.2	para.bnd.inp	10
7.2.3	trajectory_points.inp	11
7.2.4	mesh	12
7.2.5	atmos_gas	12
7.2.6	virgin() or char()	12
7.2.7	decomp()	12
7.2.8	Bprime	13
7.2.9	pyro_gas	13
7.2.10	solid()	13
7.2.11	mechanical()	14
7.3	FRSC	14
7.3.1	lewis.bulk	14
7.3.2	frsc.inp	14
7.3.3	wall_data.processorid.plt	14
7.3.4	surf_coverage.processorid.plt	14
8	Boundary conditions and volume conditions	15

9 Mesh motion	15
10 Output files	16
10.1 LeMANS	16
10.2 MOPARMD	17
10.3 FRSC	17
11 Examples	17
11.1 IRV-2	17
11.1.1 Conduction only	17
11.1.2 Conduction and ablation	18
11.1.3 Conduction, ablation and CFD mesh motion	18
11.2 2d panel	18
11.3 Planar ablation	19
11.4 HIPPO nozzle	19
11.5 Ablation Workshop Test Cases	19
11.5.1 Test Case 1	19
11.5.2 Test Case 2.1	19
11.5.3 Test Case 2.2	19
11.5.4 Test Case 2.3	19
11.5.5 Test Case 3.0	19

1 Overview

This document will describe the steps necessary to run MOPARMD and LeMANS-MOPARMD.

The /docs directory includes the LeMANS Users Guide, Jon Wiebenga’s thesis describing the development of the MOPARMD code, and Leonardo Scalabrin’s thesis describing the development of the LeMANS CFD code. Users should consult these references for details about the LeMANS and MOPARMD codes.

Multiple test cases are included with the distribution. In examples/coupled, the files required for a coupled simulation of the IRV-2 case are provided. An additional test case called 2dpanel is located in the examples/coupled directory. This test case utilizes the structural mechanics capability of MOPARMD. In examples/mopar, the files required for standalone simulations of 2d planar ablation and axisymmetric ablation using MOPARMD are provided. In examples/lemans, the files required for a standalone simulation of a rocket nozzle including a condensed phase species are provided.

2 Current limitations

This section describes currently known issues and limitations with the distribution. It will be updated as these issues are addressed.

1. Coupling to the FRSC module was not extensively used, and no example case exists to demonstrate these capabilities.
2. Pyrolysis is currently supported only in stand-alone mode, with the main limitation that the input file format for the pyrolysis kinetics is limited to three “species” (but all the actual computations are unrestricted).
3. MOPARMD can only be executed on a single processor.
4. MOPARMD does not have a working restart capability. This means that multiple trajectory points cannot not analyzed in succession yet, as was done in [3] for the IRV-2 vehicle.
5. Both MOPARMD and LeMANS contain legacy code from the original version of MOPAR which had only a 1D capability. This 1D version is no longer supported.
6. Multi-material simulations are not fully supported at this time.

3 Getting started

LeMANS is a parallel CFD code developed for the simulation of high-enthalpy hypersonic atmospheric flows. It is written in the C programming language. MOPARMD is a material response code developed for the simulation of ablating materials. It is written in the Fortran 90 programming language. The two codes have been coupled together as the LeMANS-MOPARMD code.

Currently, even when running LeMANS-MOPARMD in parallel, when the code runs the MOPARMD module on a single processor.

Simulations should be run on the Flux parallel cluster at the University of Michigan Center for Advanced Computing (CAC). (Note that, with modifications to the Makefile, LeMANS-MOPARMD may be run on any MPI capable parallel machine or single processor Linux computer.)

As a first step before using LeMANS-MOPARMD, you should apply for an account on Flux through the CAC website:

<https://www.engin.umich.edu/form/cacaccountapplication>

Accounts are available for all University of Michigan students, and for others with submission of an external access form. Once you have an account, you will need an SSH client, such as PuTTY for Windows, to connect to Flux. You can access Flux by logging into

flux-login.engin.umich.edu

using your U-M Login ID and Kerberos Password. If you're logging in from off-campus, you'll either want to use the U-M VPN client (<http://www.itcom.itd.umich.edu/vpn/>) or SSH to a publicly available login node like login.itd.umich.edu or login.engin.umich.edu, and then from there login to flux-login.engin.umich.edu.

4 Obtaining a Copy of LeMANS-MOPARMD

After logging in to your account, download the latest trunk version of LeMANS-MOPARMD from the SVN repository to your local work directory by doing the following:

1. First, checkout a copy of the entire LeMANS-MOPARMD package (this command is one line):

```
svn checkout svn+ssh://uniquename@login.itd.umich.edu/afs/umich.edu/group/acadaff/ngpd/  
svn-repos/svnLEMANS_MOPARMD DirWhereFullWorkingCopyGoes
```

2. Next, make a copy of the trunk into the /branches directory, which will be your branch:

```
cd DirWhereFullWorkingCopyGoes  
svn cp trunk branches/YourBranchName (this should be your initials or uniquename)  
svn update  
svn commit -m "Created $uniquename's branch"
```

3. Remove your local FullWorkingCopy (but you just committed it to the server so you can retrieve your branch in the next step):

```
cd ../  
rm -r DirWhereFullWorkingCopyGoes
```

4. Now, checkout just your branch so no accidental updating of the trunk occurs (this command is one line):

```
svn checkout svn+ssh://uniquename@login.itd.umich.edu/afs/umich.edu/group/acadaff/ngpd/  
svn-repos/svnLEMANS_MOPARMD/branches/YourBranchName DirWhereYourBranchWorkingCopy-  
Goes
```

Now you have your branch checked out and can update it however you want. A quick reference sheet of subversion commands is found in /documents. There is exhaustive documentation on SVN at <http://svnbook.red-bean.com/>.

5 Compiling

metis-4.0.3/ Contains the source code for Metis, needed by LeMANS

sources/ Contains the source code for the combined LeMANS-MOPARMD code. The *.c files in this directory are the LeMANS source.

sources/MOPAR/ Contains the source code for both stand-alone MOPARMD, and the MOPARMD library needed for the coupled code.

The compilers that are supported are GCC (Version 4.4.7 or later) and Intel (Version 11.1 or later). Stand-alone MOPARMD can be compiled using either the GCC or the Intel compilers. The MOPARMD library must be compiled with using GCC compilers at this time. LeMANS and Metis can be compiled using either the GCC or the Intel compilers.

To build stand-alone MOPAR go to sources/MOPAR/ and run make. To build LeMANS-MOPARMD go to sources/ and run make. This will automatically compile the Metis and MOPARMD libraries, then compile the LeMANS source files, and link it all together. There is not a separate executable for “stand-alone” LeMANS. Just use LeMANS-MOPARMD and don’t use the coupled features.

6 Mesh Generation

Refer to the LeMANS Users Guide in /docs/LeMANS_Users_Guide.pdf for instructions on how to generate a mesh for LeMANS. These instructions are applicable to both the flow field and solid meshes.

7 Input files

7.1 LeMANS

Please consult /docs/LeMANS_Users_Guide.pdf for details about the required input files for LeMANS. Some parameters are different, or new, in the current implementation, and they are discussed in the sections below.

7.1.1 prob_setup.dat

The following input variables are specified in prob_setup.dat. These values are different than what is specified in the LeMANS User Guide or are not discussed in that guide because they are used only for coupled simulations.

The following general values have been added since the latest release of standalone LeMANS.

LINEAR_ITERATIONS For controlling the number of iterations when solving the linear system, instead of using hard-coded value. Default value is 4.

IS_THERMO_EQ 0 No forced thermodynamic equilibrium (default). 1 Change relaxation times to try to force thermodynamic equilibrium (needs further work).

IS_CHEM_EQ 0 No forced chemical equilibrium (default). Change reaction rates by factor of 1000 to try to force chemical equilibrium (has been used successfully).

LOCAL_TIMESTEP 0 Time step same for all cells. 1 Time step different for each cell to give more uniform CFL number. This was an experimental feature that did not pan out. Probably should be left at default value of 0.

DISS_FACT Default value is 0.3. Used to add numerical dissipation to the flux vector splitting scheme used in LeMANS. See Equation (3.11) of [6].

INFO Larger values will print out more information that is useful for debugging. Default value is 1.

EXTRAP Used to switch between first and second order boundary conditions at walls and pressure outlets. Recommended and default value is 0.

The following values are used to set boundary conditions at walls.

IS_SUPER_CAT = 0 $dY/dx = 0$

IS_SUPER_CAT = 1 $Y_{w,i} = Y_{freestream}$

IS_SUPER_CAT = 2 solve species conservation equations at wall using Y_{cl} and Y_g (g = blowing gas, cl = left cell)

IS_SUPER_CAT = 3 use $Y_{w,i} = Y_i$, Y_i is specified for each i species in prob_setup.dat

IS_SUPER_CAT = 4 allow ions and electrons to recombine at wall

IS_SUPER_CAT = 5 simple model for only $O_g + C_s \rightarrow CO_g$ reaction

IS_SUPER_CAT = 6 use FRSC module to solve for $Y_{w,i}$ including effects of finite rate surface chemistry

WALL_COND = 0 T_w specified in prob_setup.dat, $\dot{m} = 0$

WALL_COND = 1 T_w from radiative equilibrium, $\dot{m} = 0$

WALL_COND = 2 T_w from MOPARMD, \dot{m} from MOPARMD and FRSC. IS_ABL_SPECIES flag needs to be set for each species that is produced through finite-rate surface reactions. Requires wall_data.processor_number.plt files to exist at startup if simulating trajectory point two or later using the FRSC module. Requires wall_init.plt to exist at startup if simulating trajectory point two or later using MOPARMD.

WALL_COND = 3 T_w from MOPARMD, \dot{m} from MOPARMD and FRSC. IS_ABL_SPECIES flag needs to be set for each species that is produced through finite-rate surface reactions. Requires wall_data.processor_number.plt files to exist at startup if simulating trajectory point two or later using the FRSC module. Requires wall_init.plt to exist at startup if simulating trajectory two or later using MOPARMD. Use if simulating trajectory point three or later.

WALL_COND = 4 T_w specified in prob_setup.dat, \dot{m} from Blasius solution.

WALL_COND = 5 T_w specified in prob_setup.dat, \dot{m} from prob_setup.dat.

WALL_COND = 6 T_w specified in prob_setup.dat, \dot{m} given by an expression that allows the ablation rate to vary as a function of axial location along the surface.

WALL_COND = 7 T_w and \dot{m} profiles read from input files. This boundary condition was used by Prof. Alex Martin (UKy) to model the Stardust vehicle.

WALL_COND = 60 T_w from solution of steady state energy balance equation (see LeMANS manual for this equation) and $\dot{m} = 0$.

WALL_COND = 61 T_w from solution of steady state energy balance equation (see LeMANS manual for this equation) and \dot{m} from the FRSC module. IS_ABL_SPECIES flag needs to be set for each species that is produced through finite-rate surface reactions.

P_STAG Stagnation pressure for stagnation inflow BC.

TT_STAG Stagnation translational temperature for stagnation inflow BC.

TV_STAG Stagnation vibrational temperature for stagnation inflow BC.

The following values are used when coupled simulations are run.

IS_MOPAR_MD Set to 1 to run a coupled simulation.

NEW_TEMP_WEIGHT Multiply the value returned from MOPARMD by this value to “damp” the feedback coupling. Currently only used with the recession rate.

TRAJECTORY_DT The interval of real time that MOPARMD will simulate.

qHold_perc Difference between average heat flux over entire coupled surface at subsequent calls to MOPARMD. When the difference reaches this value, MOPARMD is not called for future iterations and moves to the next trajectory point (if applicable). Value is given in percent relative difference. Default value is 1.0.

qWall_perc Maximum L2 norm of the change in wall heat flux between successive CFD iterations before MOPARMD is called. Default value is 5e-2.

IS_MOVING_MESH Set to a non-zero value to include mesh movement. See Section 9 for more information.

BEZ_MESH_DTYPE Used when IS_MOVING_MESH = 9 or -1. Type of mesh distribution for Bezier mesh movement.

BEZ_MESH_XO Used when IS_MOVING_MESH = 9 or -1. Initial cell length for Bezier mesh movement.

SOLID_MESH_FILE The filename for the solid mesh.

SOLID_GRID_FACTOR The factor by which the coordinates in the solid mesh file are multiplied at run time.

The following values are used to model turbulence.

VISC_MODE 0 Inviscid, 1 Laminar, 2 Baldwin-Lomax, 3 Menter-BSL, 4 Menter-SST, 5 Spalart-Allmaras (reserved but not yet implemented)

TURB_PR Turbulent Prandtl number. Default value is 0.9.

TURB_SC Turbulent Schmidt number. Default value is 0.9.

MUT_FACTOR_INF Free-stream ratio of eddy viscosity to molecular viscosity. Default value is 0.009.

OMEGA_INF Free-stream specific dissipation. Default is zero because this is case-specific, no good general default value.

TURB_VORT_SOURCE 0 “Standard” turbulence source term formulation, 1 “Vorticity-based” turbulence source term formulation (default and recommended).

MUT_FACTOR_STAG Ratio of eddy viscosity to molecular viscosity for stagnation inflow BC.

OMEGA_STAG Specific dissipation for stagnation inflow BC.

MUT_FACTOR_STAG Ratio of eddy viscosity to molecular viscosity for stagnation inflow BC.

OMEGA_STAG Specific dissipation for stagnation inflow BC.

The following older inputs should no longer be explicitly set by the user (will return an error message):

IS_VISCOUS

IS_LOMAX

7.2 MOPARMD

There are two different groups of input files: files for specifying program options, and files containing numerical values (material properties, boundary values, etc.). The necessary files are listed below, and a detailed description of each file can be found in the subsequent sections.

Files marked with an * are not necessary for all types of simulations, and the name of the file can change. This is explained in the detailed sections. Note that the formatting of the input files must be consistent with the example files or there may be errors when the files are read.

1. mopar.nml
2. para_bnd.inp
3. trajectory_points.inp*
4. mesh
5. atmos_gas*
6. virgin() or char()*
7. decomp()*
8. Bprime*
9. pyro_gas*
10. solid()*
11. mechanical()*

7.2.1 mopar.nml

This file contains all of the simulation parameter setup options for the physical models, time steps, and input files. A description of each of the input values is listed below. The older version of this file, para_setup.inp is still supported by the code but it is recommended to use this newer version when setting up new cases. An example of the older version of this file is found in examples/coupled/2dpanel.

General namelist

num dim The number of dimensions for the problem. Can be either two or three.

axi Flag to indicate an axisymmetric simulation. T = run an axisymmetric simulation, F = run a planar simulation. In order to run an axisymmetric case, num dim should be set to 2. Note that axisymmetric simulations currently only work for the thermal portion of the code, and the capability has not yet been added to the structural code.

grid_factor The mesh coordinates are divided by this value when the mesh is read. This allows the mesh to be created using arbitrary units. After dividing by grid_factor, the mesh should be in units of meters.

coupled Flag to indicate whether a coupled case is being run. T = obtain boundary conditions at flow interface from LeMANS, F = obtain boundary conditions at flow interface from input files.

num_mat Value indicating the number of different materials used in the solid domain. This determines the number of different material property files that need to be specified. Multi-material simulations are not fully supported at this time.

time_start Initial time (seconds) of the simulation. This should generally be set to 0.0. For coupled cases where the simulation time changes at each trajectory point the correct start time will be passed from LeMANS, and this value does not need to be modified.

time_final Final time (seconds) of the simulation. Once again, for coupled cases this value will be passed from LeMANS.

time_step Time step in seconds for the energy equation and the structural (Cauchy's) equation.

output_step Time step for writing output files.

allout Flag for how often to output new solution files. T = output the solution at each time step, F = only output the final solution.

Energy namelist

active Flag for whether or not to solve the energy equation. T = solve the energy equation, F = don't solve the energy equation.

ablation Flag to include ablation effects in the aerodynamic heating boundary condition. T = include ablation, F = don't include ablation. Setting this to T will allow for a surface mass flux to be computed, and the code will assume a recessing surface and compute an updated mesh.

pyrolysis Solve pyrolysis (thermal decomposition). This option is not currently supported but is under development.

frsc Flag to use the finite rate surface chemistry (FRSC) module in LeMANS to solve for an ablating surface. T = use the FRSC module, F = use a B' table. A coupled simulation with LeMANS must be specified in order for the FRSC approach to work. Note that this is a very new option and has not been thoroughly tested or debugged.

blowing_corr T = include a blowing correction for the aerodynamic heating BC, F = don't include correction. This option is most useful for running uncoupled simulations. It could also be used with coupled simulations to try and speed convergence, but it is of questionable physical accuracy for coupled cases and generally should not be used. More details can be found in [1].

hotwall_corr T = include a hot wall correction for the aerodynamic heating BC, F = don't include correction. This option is most useful for running uncoupled simulations. It could also be used with coupled simulations to try and speed convergence, but it is of questionable physical accuracy for coupled cases and generally should not be used. More details can be found in [1].

const_cmch T = use a constant value of $C_m=Ch$, F = use a Lewis number based value of $C_m=Ch$. C_m is the mass transfer coefficient and Ch is the heat transfer coefficient. The ratio is used in the computation of the aerodynamic heating sensitivity to temperature, but is only important when the B' ablation approach is used.

corr_sdot Flag to correct the surface recession rates for curved surfaces. T = apply correction, F = use uncorrected recession rate. The need for this correction arises from the fact that as the surface recesses on a curved geometry, the recession rate must change in order to produce the same amount of total mass loss over a time step. In most cases the difference between the corrected and uncorrected values should be quite small. This option is currently not supported in the code.

log_interp Flag for changing interpolation within a B' table. T = interpolate in B' table using the log of the pressure values, F = interpolate on the original pressure values.

cmch Value of constant $C_m=Ch$ to be used if **const_cmch** is T.

lewis Value to use for the constant Lewis number.

mesh_poisson Value to use for Poisson's ratio for mesh deformation due to ablation. This value is used only to move the mesh when ablation is present, and is not a physical value.

num_slide Number of non-ablating boundary nodes permitted to slide with mesh motion.

preconditioner Value indicates the type of preconditioner to use for the GMRES method when solving the energy equation. Options are listed below. Generally option 2 or 4 works best. Implementation of more advanced preconditioners would be useful. 0. No preconditioner 1. Gauss-Seidel preconditioner 2. Symmetric Gauss-Seidel preconditioner 3. Diagonal scaling 4. ILU0 preconditioner

T_init Initial temperature of solid domain in Kelvin.

P_init Initial pressure in Pa.

mat_orient(ψ, β, α , material number) (energy) Rotation angles (in degrees) for use with anisotropic materials for the energy equation. This allows for the material properties to be specified in a material reference frame, and then rotated into the mesh coordinate system. There should be a line for each material used in the solid domain. For example, if the number of materials is 3, then there should be 3 angle lines. For isotropic materials all angles should be set to 0.0. Specified angles represent a counter-clockwise rotation about the given axis.

eps_NR_energy Newton-Raphson iteration convergence tolerance for material energy equation. Default value is 1e-10.

eps_NR_pore Newton-Raphson iteration convergence tolerance for pyrolysis gas momentum equation (not currently supported). Default value is 1e-10.

Structural namelist

active Flag for whether or not to solve the structural equation. T = solve the structural equation, F = don't solve the structural equation.

static Flag for static structural calculations. T = use a static structural formulation (neglect acceleration and damping), F = use a full dynamic structural formulation.

orthotropic Flag for orthotropic structural material properties. T = structural material properties are orthotropic, F = structural material properties are isotropic. This modifies how the constitutive matrix is assembled.

num_elastic Gives the number of structural ("elastic") materials.

elastic_mat Values indicating which of the materials should be treated as elastic. Elastic materials are ones in which the structural mechanics equation is solved. This allows some of the materials to be considered non-load bearing. See the insulated metallic plate test case in [7] for an example. Multiple values can be input using a comma separated list. For example, if three materials are used in a simulation and materials 1 and 2 are elastic, this would be input as 1,2. But be warned that in many places the code is hard-coded to only use a single material. Simulations with multiple materials should *not* be performed at this time.

preconditioner Preconditioner to use for the GMRES method when solving the structural equation. Options are the same as for energy equation. Option 4 generally seems to work best.

mat_orient(ψ, β, α , material number) (structural) Rotation angles (in degrees) for use with orthotropic materials for the structural equation. The same description applies as for the rotation angles for the energy equation.

Files namelist

mesh Name of the solid domain mesh file.

atmo_gas Name of the atmospheric gas thermodynamic input table.

decomp(i) Name of the file containing a description of the resin decomposition for material i.

Bprime Name of the file containing the thermochemical ablation (B') table.

pyro_gas Name of the file containing the pyrolysis gas thermodynamic table.

solid(i) Name of the file containing solid properties (porosity and permeability) for virgin and fully-charred material i.

virgin(i) Name of the file containing the material properties for the energy equation for virgin material i. A separate file should be listed for each material with quotes around the names.

char(i) Name of the file containing the material properties for the energy equation for char material i. A separate file should be listed for each material with quotes around the names.

mechanical(i) Name of the file containing the structural material properties for material i. A separate file should be listed for each material, with quotes around the names, whether the material is elastic or not.

7.2.1.1 Notes on Input Files for Ablation When simulating an ablating material, the material response code reads in values assuming that a virgin material will decompose into a char material as the ablation process proceeds. This is consistent with the behavior of charring (pyrolyzing) ablative materials. To specify a non-charring material in the input files, simply specify the same input files for the char materials as the virgin materials.

7.2.1.2 Notes on Input Files for Structural Calculations The structural capability of the code has not been used since it was originally implemented in [7]. If this capability is required at this point, it will be necessary to use the old-style input file (which should still work, but has been superceded by the namelist format such as that in mopar.nml). It would not be that hard to update the new style input to include the structural information, which is the path that should be taken when there is a need to work with the structural solver again.

7.2.2 para_bnd.inp

This file allows for the specification of boundary conditions, and for the boundary condition input files to be specified. Note that the actual specification of most boundary conditions is found in the mesh file, which is described in more detail in Section 8. The file is split into three sections: boundary conditions for the energy equation, boundary conditions for the structural equation, and verification test setup.

The verification tests listed in the file are typically no longer used since the Method of Manufactured Solutions is also implemented. More details on these tests can be found in the source code, but in general they can be ignored. Removing them from the file, however, may cause problems with reading the file unless the source code is also modified.

Within the energy equation section of the file, the following boundary conditions and options can be specified:

Specified temperature The name of the input file containing the specified temperature values to be applied is listed here. The input file contains a list of (x, y) coordinates with a temperature value for each coordinate. The temperature values are linearly interpolated between the coordinates depending on the actual boundary mesh points in the domain. Coordinate values should cover the range of the domain over which a specified temperature boundary condition is applied. This method of specifying boundary values works fairly well for simple two-dimensional domains, but is not very good for three-dimensional cases. A better method for specifying non-constant boundary values should be implemented (maybe using an external program such as Ansys). Note that a full loop should not be specified here, that is, the last point should not be the same as the first point in the list.

Specified heat flux The name of the input file containing the specified heat flux values to be applied is listed here. The setup of the file is the same as for the specified temperature case, and the same limitations apply.

Aerodynamic heating The aerodynamic heating inputs found here are used for running uncoupled analysis. Uncoupled analysis is run by specifying one or more files that list pressure, heat flux or heat transfer coefficient, boundary layer edge velocity, recovery enthalpy, and cold wall temperature at different locations along the aerodynamic boundary. The values in the files are used to compute the aerodynamic heat flux as well as the hot wall and blowing corrections if desired. Multiple files can be listed for different simulation times, and the values in the files will be interpolated linearly in time between the files. This is again only implemented for two-dimensional analysis. Note that the heat flux or coefficient must be negative to produce heat transfer to the solid material.

[LSUB] Option to change the meaning of the “Heat Transfer” column in the input files. T = the heat transfer column lists actual heat flux values in $[W/m^2]$, F = the heat transfer column lists a heat transfer coefficient ($\rho_e u_e C_H$) in $[kg/m^2/s]$

[nTimes] Number of aerodynamic heating input files

Radiative boundary condition Two different options, as well as an input file, can be specified for the radiative boundary condition

[LRAD] Flag for turning the radiative boundary condition on or off. T = the radiative boundary condition is applied, F = the radiative boundary condition is not applied. The radiative boundary condition is applied at boundaries that have an aerodynamic heating boundary condition, a specified flux boundary condition, or a specified temperature boundary condition specified. This can be changed in the setBflags subroutine within the boundary.f90 source file.

[radBV] Value to use for the reservoir temperature. The radiative heat flux is proportional to the difference between the computed boundary temperature and the reservoir temperature, both raised to the fourth power [1]. If radBV is less than zero, then the reservoir temperatures listed in the file on the next line are used instead. This allows for a non-constant reservoir temperature to be specified.

[input file] The input file consists of a series of (x, y) coordinates and an associated reservoir temperature. These temperatures are only used if radBV is less than zero.

Within the structural equation section of the file, the following boundary conditions and options can be specified:

nTimes struc Value indicating the number of input files to be used for specifying the structural boundary conditions. Each input file represents a different time, and values are linearly interpolated in time between the files. A value of at least two must be used, however, the input files can be identical if a constant boundary condition case is being simulated.

Specified displacement A list of input files for the specified displacement boundary condition. The files consist of coordinates and displacements (u, v, w) at each coordinate. While the files include a column for the z-direction, this input method really only works for two-dimensions, and has the same limitations as the input files for the specified temperature boundary condition.

Specified traction A list of input files for the specified traction boundary condition. The files consist of coordinates and stress tensor components at each coordinate. There is also a column for specifying a pressure, which is treated as a stress force normal to the boundary. While the files include a column for the z-direction, this input method really only works for two-dimensions, and has the same limitations as the input files for the specified temperature boundary condition.

7.2.3 trajectory_points.inp

This file contains a list of trajectory times starting with 0.0, and ending with the final simulation time. These times control how long the material response code is run during a coupled simulation. This file is read by LeMANS and is only necessary when a coupled simulation is run. The time values listed in the file replace the TIME and TFINAL values defined in the mopar.nml file. The code will move from one time to the next once the surface properties have converged to the level defined by the **qHold_perc** variable in the prob setup.dat file that is used by LeMANS. This file allows long simulation times to be split up into a series of “sub-trajectory” steps. This feature is not currently operational, since the restart capability in MOPARMD is not fully operational.

7.2.4 mesh

Mesh files can be generated in Pointwise using the Ansys Fluent format (same as LeMANS). Two-dimensional domains should be meshed using triangular elements, and three-dimensional domains should be meshed using tetrahedral elements. Details on specifying boundary conditions and material types can be found in Section 8.

7.2.5 atmos_gas

This file contains tables of temperature $[K]$, enthalpy $[J/kg]$, density $[kg/m^3]$, Prandtl number, and viscosity $[kg/m-s]$ at different pressure values. These tables are used to compute the hot wall correction for the aerodynamic heating boundary condition when running an uncoupled case.

7.2.6 virgin() or char()

These files contain the material properties needed for solving the energy equation. The file format used is from the old material properties database that used to be available from NASA. Temperature dependent material properties can be specified, with the properties assumed to vary linearly between the temperature values that are input. The format is fairly straightforward, but care should be taken to ensure that the formatting matches the example file **exactly**, or there will be issues with reading the files. To make changes to the number or type of material properties that are specified, see the source code (ideas.f90) for more details on how the properties are read and stored.

The virgin() file contains properties for a material in its 100% virgin (non-decomposed) state, and the char() files contains properties for a material in its 100% charred state. Within the code, material properties are determined by linearly interpolating between the virgin and char states based on the degree of char. In the case of a non-decomposing ablator, the virgin() and char() should be the same so that the virgin properties are used in the computations. A different file must be provided for each different material that is being modeled. The material properties that currently need to be specified in the files are as follows:

1. Density $[kg/m^3]$ (constant)
2. Thermal conductivity - x $[W/m-K]$ (temperature dependent)
3. Thermal conductivity - y $[W/m-K]$ (temperature dependent)
4. Thermal conductivity - z $[W/m-K]$ (temperature dependent)
5. Thermal conductivity - xy $[W/m-K]$ (temperature dependent)
6. Thermal conductivity - xz $[W/m-K]$ (temperature dependent)
7. Thermal conductivity - yz $[W/m-K]$ (temperature dependent)
8. Heat of formation $[J/kg]$ (constant)
9. Emissivity (constant)

7.2.7 decomp()

This file contains information on the decomposition of a resin material in a charring ablator.

The model for a charring material is taken from [1], and assumes that the material consists of a resin and a binder, with the total material density given by Equation 1. Γ is the resin volume fraction in the virgin composite, and ρ_A , ρ_B , and ρ_C are concentrations of different components making up the resin and binder materials. Note that this is only a modeling assumption, and is not necessarily an accurate physical description of a material.

$$\rho_s = \underbrace{\Gamma(\rho_A + \rho_B)}_{\text{resin}} + \underbrace{(1 - \Gamma)\rho_C}_{\text{binder}} \quad (1)$$

Each component (A, B, C) is assumed to decompose according to an Arrhenius relationship as shown in Equation 2.

$$\frac{\partial \rho_i}{\partial t} = -k_i \rho_{v_i} \left(\frac{\rho_i - \rho_{c_i}}{\rho_{v_i}} \right)^{\psi_i} e^{-E_i/RT} \quad \text{for } i = A, B, \text{ and } C \quad (2)$$

The values listed in the decomp file(s) define the number of pyrolyzing components, the resin volume fraction, the number of species that make up the resin, and the necessary constants for the Arrhenius relation for each decomposing species. The input values are listed below.

- **NBSP:** Number of pyrolyzing species. The term species refers to the *A*, *B*, or *C* components in Equation 1. This allows for more than three components to be defined for a decomposing solid.
- **GAMMA:** The volume fraction of the resin in the virgin composite.
- **NG:** The number of different species that make up the resin portion of the solid material.
- **Arrhenius constants:** The remainder of the file lists the Arrhenius constants for each pyrolyzing species. The constants for the resin species should be listed first.
 - RHOVi: Virgin density of species *i* [*kg/m*³]
 - RHOCi: Char density of species *i* [*kg/m*³]
 - ki: Pre-exponential factor for species *i* [*1/s*]
 - psi: Reaction order for species *i*
 - EORi: Scaled activation energy ($\frac{E}{RT}$) for species *i* [*K*]
 - MINTi: Minimum reaction temperature for species *i* [*K*]. Temperatures below this value should not lead to any decomposition of the material.

7.2.8 Bprime

This file contains the B' table necessary for implementing the ablation boundary condition, and for modeling charring ablators. The second line of the file allows for specification of different input units; internally, the code uses SI units. The B' table consists of pressure, a nondimensional gas blowing rate, a nondimensional char blowing rate, temperature, and wall enthalpy. This data can be generated in a program such as ACE [5]. More details on the arrangement of the B' table and how values are interpolated within it can be found in [1].

7.2.9 pyro_gas

This file contains the necessary input values for computing the gas flux term in the energy equation, and for computing the terms in the pyrolysis gas continuity equation. The pyrolysis gas quantities are listed as functions of temperature. The quantities that are listed in the file are temperature [*K*], enthalpy [*J/kg*], *c_v* [*J/kg - K*], *c_p* [*J/kg - K*], molar mass [*kg/mol*], and viscosity [*kg/m - s*].

7.2.10 solid()

This file lists the permeability and porosity of the material in the virgin and char states, which is given by β .

$$\beta = \frac{\rho_v - \rho_s}{\rho_v - \rho_c} \quad (3)$$

It is assumed that the listed β values start at 0.0 (fully virgin material) and go to 1.0 (fully char material), increasing by a constant increment. The permeability and porosity values are interpolated linearly between the input values based on the currently computed degree of char. An option exists to interpolate the permeability values based on the log of the input values, but this has not yet been implemented.

7.2.11 mechanical()

This file contains the material properties necessary for the structural mechanics calculations. The file format is the same as for the virgin and char files. A separate file should be included for each different material. The material properties that are currently listed in the file are given below.

1. Density [kg/m^3] (constant)
2. Youngs modulus - x [N/m^2] (temperature dependent)
3. Youngs modulus - y [N/m^2] (temperature dependent)
4. Youngs modulus - z [N/m^2] (temperature dependent)
5. Youngs modulus - xy [N/m^2] (temperature dependent)
6. Youngs modulus - xz [N/m^2] (temperature dependent)
7. Youngs modulus - yz [N/m^2] (temperature dependent)
8. Poisson's ratio - xy (temperature dependent)
9. Poisson's ratio - xz (temperature dependent)
10. Poisson's ratio - yz (temperature dependent)
11. Thermal expansion coefficient - x [$1/K$] (temperature dependent)
12. Thermal expansion coefficient - y [$1/K$] (temperature dependent)
13. Thermal expansion coefficient - z [$1/K$] (temperature dependent)

7.3 FRSC

7.3.1 lewis.bulk

This file contains thermodynamic property data for condensed species. The data has been compiled by Gordon and McBride (NASA/TP-2002-211556) and is presented in the form of nine coefficients used to calculate thermodynamic properties as functions of temperature.

7.3.2 frsc.inp

This file contains information regarding the chemistry model used by the FRSC module. The file is divided into three different sections: the phase and active site section, the species section, and the surface reaction(s) section.

7.3.3 wall_data.processorid.plt

One file for each processor, used to set the initial values of the wall properties for a simulation that uses the FRSC model. These files come from a previous simulation without surface chemistry. The FRSC model cannot be used without this, thus a FRSC simulation must always start from a previous simulation without surface chemistry.

7.3.4 surf_coverage.processorid.plt

Gives the surface concentration for each species. Will be read if exists in directory. There is a bug in the current implementation so this feature currently does not work as intended.

Please see docs/FRSC_User_Manual.pdf for more details about the FRSC module, including the format of the required input files.

8 Boundary conditions and volume conditions

Boundary conditions and volume conditions are specified in the associated mesh file. As a first step, read “Mesh Generation” in the LeMANS Users Guide. This describes how to generate a mesh for the CFD calculation, and the available types of CFD boundary conditions.

The mesh for the solid material must be generated by setting each volume type to a solid, rather than to a fluid as is done when the CFD mesh is generated. The solid material mesh must be generated using the following types of Fluent boundary conditions:

- wall

The name of the wall boundary surface must be one of the following:

Flowfield For coupling to LeMANS. Also used for standalone MOPARMD simulations when an aeroheating boundary condition is desired.

SpecifiedFlux For energy equation.

SpecifiedTemp For energy equation.

Adiabatic For energy equation.

Radiation For energy equation.

Cond_Ver For verification test cases.

Rad_Ver For verification test cases.

Aniso_Ver For verification test cases.

SpecifiedDisp For Cauchy’s equation.

SpecifiedTracFF For Cauchy’s equation.

SpecifiedTrac For Cauchy’s equation.

If both the energy and structural equations are being solved, then the boundary conditions are named in the following format:

NameOfEnergyBC : NameOfStructuralBC

The boundary conditions for the solid mesh are read by MOPARMD in sources/MOPAR/mesh_read.f90. NameOfEnergyBC is for the energy equation and NameOfStructuralBC is for the structural equation. If you are only solving one of the equations, then you should only specify the boundary condition that you need. In the Fluent.f90 file, the Fluent_face_read subroutine performs a check to determine if the boundary condition line in the .cas file has one or two entries on it. The fluent2mr_bc subroutine in the mesh_read.f90 file then determines whether an energy or structural boundary condition has been read based on the identifying number given to each boundary condition in the code. The energy boundary conditions all have positive numbers, and the structural boundary conditions all have negative numbers.

9 Mesh motion

There are several different options for mesh movement available. A summary is given here, and much more information can be found by looking at sources/mesh_movement_desc.c in the source code.

Values of IS.MOVING_MESH:

0 No mesh movement (defunct).

1 Random movement (defunct).

- 2 Constant movement (defunct).
- 3 Constant movement (defunct).
- 4 Constant movement using Landau coordinates (defunct, see [4] and associated references for more information about Landau coordinates)
- 5 Using material response (defunct).
- 6 Using material response (defunct).
- 7 Material response with shock tailoring. Unreliable, so additional associated options are not discussed in this manual (defunct).
- 9 Bezier curve mesh motion. Will only work for “structured-like” meshes. This option attempts to adapt the mesh in a time accurate manner. It is an implementation of the Arbitrary Lagrangian-Eulerian (ALE) Method. Currently, this option can only be used for simulations that are run on a single processor. It is recommended that option [-1] be used instead.
- 1 Bezier curve mesh motion. Will only work for “structured-like” meshes. Adapts mesh only when MOPARMD is called.

Values 0 through 7 were implemented for the MOPAR 1D code, and as discussed in Section 2, they are no longer supported. However, they have been retained in the source code in case they become useful for future work.

The method that Jon Wiebenga implemented is based on the use of Bézier curves to define new node-lines within the mesh. This method corresponds to `IS_MOVING_MESH = 9` and `IS_MOVING_MESH = -1`. This approach was suggested by Dr. Ryan Gosse at AFRL. Note that 3rd order Bézier curves are used. This approach requires the variables `BEZ_MESH_X0` and `BEZ_MESH_DTYPE` to be set in `prob_setup.dat`. When this approach is used the following variables can also be specified in `prob_setup.dat`:

SHOCK_TAIL_X0 Initial cell length when refining near shock. Default value is 1.0e-4.

SHOCK_TAIL_OFFSET Distance to offset the inlet from the shock surface. Default value is 1.0e-3.

SHOCK_TAIL_NODES Number of nodes to place on the inlet side of the shock. Default value is 10.

10 Output files

10.1 LeMANS

See the LeMANS Users Guide, Section “Running a LeMANS simulation” for a discussion of the types of output files that are created by LeMANS.

In addition to the files described in the Users Guide, the following additional files are output during a simulation:

outputCC.plt Same contents as `output.plt`, but values are reported at cell centers rather than at nodes.

residuals.dat Residuals of all equations as a function of iteration number. Can be used for plotting.

monitors.dat This file contains the values of certain wall properties at each iteration. It can be used to monitor a simulation for convergence.

10.2 MOPARMD

mopar_inputs_X.dat The result of reading the wall_init.plt file. These are the surface conditions present at the start of the time interval being simulated in a coupled simulation. This file is printed each time MOPARMD is called in a coupled simulation.

mrOut_X.dat Position and temperature of each wall face in a coupled simulation. This file is printed each time MOPARMD is called in a coupled simulation.

SProbe.dat and IProbe.dat Sprobe files are "surface probe" files, and IProbe files are "interior probe" files. These contain data for each time step related to specific points. This is in contrast to the Tecplot data, which gives the whole domain (and may not be desirable each time step). The user controls the number and location of these probes in the mopar.nml file, through the "surface_probes" and "interior_probes" namelists. Note that surface probes are fixed to the nearest node on the surface (and move as the mesh moves), while the interior probes are fixed points in space.

tecEnergy_time.dat Properties of the solid at each node and the specified time.

solidMesh_*.dat, solidMesh_*.cdb, and new_LeMANS_mesh.cas These files write out the mesh (which can change due to surface recession) to various formats. These files can be useful for debugging.

energyTerms.dat This file contains the different terms of the energy equation useful for debugging. It can safely be ignored.

dt_qwall.dat This file contains some additional items to help understand when/why MOPARMD is triggered. The source code that creates this file is found in sources/mopar.c.

eachIter_X.dat This file contains the solid properties at each iteration. This file is printed each time MOPARMD is called in a coupled simulation.

flowWall_time.dat These files contain the properties at walls at the specified time.

10.3 FRSC

This section will be updated once the implementation of FRSC has been completed.

11 Examples

11.1 IRV-2

11.1.1 Conduction only

The IRV-2 mission is described in [3] and also in [7]. The following directories are required to run this test case:

examples/coupled/IRV2/t_0s This is the time = 0s case (LeMANS only, not coupled).

examples/coupled/IRV2/t_4.25s This is the time = 4.25s case (coupled).

In both directories, the results are in results/ and a clean set of input files can be found in clean/.

To run this simulation, compile LeMANS-MOPARMD, and then call the executable from the case directory. This case is simulated by completing the following steps:

Step 1 First obtain a solution using uncoupled LeMANS at time $t = 0$. The input `problem_setup.dat` file contains the following values that specify a constant temperature, non-ablating boundary surface:

IS_SUPER_CAT = 0
WALL_COND = 0

Step 2 Run the coupled simulation from $t = 0$ to 4.25 s. The values of `IS_SUPER_CAT` and `WALL_COND` that should be used in `problem_setup.dat` file for coupled simulations are:

```
IS_SUPER_CAT = 2
WALL_COND = 2
```

To run this simulation, the following files from the simulation at $t=0$ s need to be included in the case direction for the simulation at $t=4.25$ s:

1. `restart.dat`
2. `wall_data.plt`

Copy the `wall_data.plt` file from the output of `t_0s` case and rename to `wall_init.plt` and for use as input. Use the `restart.dat` file from the `t_0s` case simulation to start this new case. Reset the iteration number at the top of the `restart.dat` file to 0.

11.1.2 Conduction and ablation

In this case, surface ablation is added using the `Bprime` table option. The steps to set up and run this simulation are the same as those for the conduction only simulation.

11.1.3 Conduction, ablation and CFD mesh motion

In this case, CFD mesh motion is added. The steps to set up and run this simulation are the same as those for the conduction only simulation.

11.2 2d panel

This is a coupled aerothermoelastic case of a 2D panel at Mach 8. The aerothermoelastic test case consists of a two-dimensional wedge in an hypersonic flow. A portion of the wedge surface is a compliant panel that can deform due to the aerothermal and aerodynamic loads from the external hypersonic flow. The pressure load specified in the `tracSpecBC.bnd` file is applied to the back side (non-flow side) of the plate. The pressures on the flow surface of the plate come from LeMANS. The ends of the plate are clamped. See Section 5.3 of [7] for a description of this case.

First, run the `t_0` simulation. Then run the coupled simulation using the same procedure described for the IRV-2 case. The following files from the simulation at $t=0$ s need to be included in the case direction for the coupled simulation:

1. `restart.dat`
2. `wall_init.plt`

Copy the `wall_data.plt` file from the output of `t_0s` case and rename to `wall_init.plt` and for use as input. Use the `restart.dat` file from the `t_0s` case simulation to start this new case, and remember to reset the iteration number.

11.3 Planar ablation

The test case is ablation of a carbon-carbon composite (no pyrolysis) in response to a somewhat arbitrary aero-heating condition. The heating condition is taken from the Ablation Workshop test case 2.3. Description of the heating condition is found in /docs/Ablation_workshop_testcase.pdf. Duration of simulation is 10 real seconds. The purpose of the test case is to demonstrate how to set up an ablation simulation.

To run this simulation, compile MOPARMD in standalone mode, and then call the executable from the case directory.

11.4 HIPPO nozzle

This test case is included to demonstrate the condensed phase capability in LeMANS. This is a standalone CFD simulation of the HIPPO nozzle #1 [2]. More details about this case are found in the associated readme.txt file.

11.5 Ablation Workshop Test Cases

Test cases #1 through #3.0 use the TACOT material. Test cases #1 through #2.3 use a 1D slab geometry 5 cm thick. A mesh with 300 nodes in the wall-normal direction with a 1% growth ratio is used. The has five columns of elements (six columns of nodes) for a total mesh size of 1800 nodes. Unless otherwise indicated, these simulations use a time step of 0.1 s. Documentation for these test cases is found in each folder.

11.5.1 Test Case 1

This test case features a specified-temperature boundary condition, which ramps up from 298 K to 1644 K over 0.1 seconds. This test case does not involve an ablating boundary condition or mesh motion.

11.5.2 Test Case 2.1

This test case builds upon test case #1, replacing the specified-temperature boundary condition with a moderate aeroheating boundary condition. Surface recession is prevented by setting $B_c = 0$ in the B tables. This test case is useful for verifying correct implementation of the ablating boundary condition. Only the first 60 seconds of heating are simulated; the 60 second cool-down period for this test case is neglected.

11.5.3 Test Case 2.2

This test case builds upon case #2.1, adding surface recession and mesh motion. With this test case it is possible to verify a complete quasi-1D ablation problem. Both the 60 heating and the 60 second cool-down period are simulated.

11.5.4 Test Case 2.3

This test case is the same as case #2.2, except that the recovery enthalpy is much higher (2.5E7 J/kg vs. 1.5E6 J/kg), yielding a more severe aero-heating boundary condition. The MOPAR-MD simulation stops after about 54 seconds, due to a mesh motion error. This simulation was attempted several times with different mesh settings (e.g. Youngs modulus) with the same result. It is believed that this is due to some limitation with how far the mesh can be compressed. Surface recession at 54.2 seconds is 1.3 cm (over half an inch), which is quite significant. To complete simulations with very large surface recessions it may be necessary to periodically stop and re-mesh the domain. It may also be possible to complete the simulation with a different mesh (possibly one with coarser elements near the back wall).

11.5.5 Test Case 3.0

This test case involves a slightly modified iso-Q geometry constructed from TACOT material. Temporally- and spatially-varying pressure and convection coefficient are applied to the test item, consisting of 40 seconds

of aero-heating followed by 80 seconds of cooling. For case #3.0 recession is prevented by setting $B_c = 0$ in the B tables.

It is not possible to successfully run test case #3.0 as described in the test case definition document. Specifically, it is observed that the temperature would drop in the region of the shoulder of the geometry, ultimately leading to a negative temperature error or some other error. The issue appears to be related to the high magnitude outwards blowing that occurs in the shoulder region. This blowing occurs because the pressure is being increased substantially on the face of the iso-Q geometry, while it is being reduced (slightly) on the side surfaces. This causes significant pressure-driven flow from the front face, through the material, and out the shoulder. It is unclear why this would result in a drop in temperature at this point. Time steps as small as 1E-6 seconds were explored, but this does not resolve the issue. Also, the mesh was refined in the vicinity of the shoulder, to no apparent effect. Based on some text included in the test case description document, it appears that other groups have encountered a similar error with an earlier variant of this test case.

For this reason, the test case is modified as follows. The temporally- and spatially-varying convection coefficient is retained, but it is assumed that the pressure at the surface is constant and uniform pressure with a value of 1 atm.

References

- [1] A. J. Amar. Modeling of one-dimensional ablation with porous flow using finite control volume procedure. Master's thesis, North Carolina State University, 2006.
- [2] Dodson J. Arnold, J. and B. Laub. Subscale solid motor nozzle tests - Phase IV and nozzle materials screening and thermal characterization - Phase V. Technical Report NASA-CR-161254, NASA, 1979.
- [3] D. W. Kuntz, B. Hassan, and D. L. Potter. Predictions of ablating hypersonic vehicles using an iterative coupled fluid/thermal approach. *Journal of Thermophysics and Heat Transfer*, 15(2):129–139, 2001.
- [4] A. Martin and I. D. Boyd. Strongly coupled computation of material response and nonequilibrium flow for hypersonic ablation. *41st AIAA Thermophysics Conference*, Paper 2009-3597, 2009.
- [5] C. A. Powars and R. M. Kendall. *Aerotherm Chemical Equilibrium (ACE) Computer Program - User's Manual*. Aerotherm Corporation, Mountain View, CA, May 1969.
- [6] L. C. Scalabrin. *Numerical Simulation of Weakly Ionized Hypersonic Flow Over Reentry Capsules*. PhD thesis, University of Michigan, 2007.
- [7] J. E. Wiebenga. *High-Fidelity Material Response Modeling as Part of an Aerothermoelastic Framework for Hypersonic Flows*. PhD thesis, University of Michigan, 2012.