

MSPM User Manual

Developed by Steven Middleton at the University of Alberta

Written by Cedric Boucher, Matteo Falsetti

Contact: David S. Nobes, dnobes@ualberta.ca

August 1, 2023

Copyright © 2023

1	Overview	1
1.1	Main Functionality.....	1
1.2	The Program	1
1.2.1	Installation.....	1
1.2.2	Uninstalling.....	1
1.2.3	Starting MSPM	1
1.2.4	MATLAB.....	1
1.2.5	Code Structure	2
1.2.6	User Configuration File	2
1.2.7	Folder Structure	2
1.2.8	Assumptions	2
2	Main Components.....	3
2.1	Model	3
2.2	Group.....	3
2.3	Body.....	3
2.3.1	Body Validation.....	3
2.3.2	Discretization Modes	3
2.3.3	Custom Heat Transfer Coefficient.....	5
2.4	Connections.....	5
2.5	Matrix.....	5
2.5.1	Regenerators.....	5
2.5.2	Heat Exchangers.....	6
2.5.3	Creating Matrices.....	8
2.6	Bridge	8
2.6.1	Bridge Types.....	9
2.7	Environment	10
2.8	Frame.....	10
2.9	Leak	10
2.10	Relation.....	10
2.10.1	Constant Offset.....	10
2.10.2	Zero X Based Scale.....	11
2.10.3	Smallest X Based Scale	11

2.10.4	Width Set.....	11
2.11	Snapshots	11
3	Other Components.....	12
3.1	Custom Minor Losses	12
3.2	Faces.....	12
3.2.1	Gas-Solid.....	12
3.2.2	Gas-Gas and Gas-Environment	12
3.2.3	Solid-Solid	12
3.2.4	Solid-Environment.....	13
3.3	Materials.....	13
3.4	Nodes.....	13
3.5	Non-Connections.....	13
3.6	Sensors	13
3.6.1	Point Sensor.....	13
3.6.2	Line Sensor.....	14
4	Mechanical System.....	16
4.1	Linear to Rotational Mechanism	16
4.2	Mechanical System.....	16
4.3	Motion Profiles	16
4.3.1	Ideal Sinusoid	16
4.3.2	Slider-Crank.....	16
4.3.3	Custom Motion Profile.....	16
5	User Interface	17
5.1	Overview	17
5.1.1	Main Window.....	17
5.1.2	Controls.....	17
5.1.3	List View	18
5.1.4	The Display Window	18
5.2	File Controls.....	19
5.2.1	Save Model	19
5.2.2	Save Model As.....	19
5.2.3	Load Model.....	20

5.2.4	Change Model Save Location.....	20
5.2.5	Change Results Save Location.....	20
5.3	Insert	20
5.3.1	Body.....	21
5.3.2	Group.....	21
5.3.3	Bridge	22
5.3.4	Leak Connection.....	22
5.3.5	Sensor.....	22
5.3.6	PV Loop Output.....	23
5.3.7	2-Body Non-Connection.....	23
5.3.8	Custom Minor Loss.....	23
5.3.9	Relation	23
5.4	Calculations	23
5.4.1	Calculate Volume.....	24
5.5	Dynamics	24
5.5.1	Animate.....	24
5.5.2	Create Motion.....	24
5.5.3	Delete Motion.....	26
5.6	Select.....	27
5.6.1	Select Objects.....	27
5.6.2	Multi-Select Objects	27
5.6.3	Delete Selected	27
5.7	Edit Model	27
5.7.1	Change All Gasses.....	28
5.7.2	Scale Model	28
5.8	View.....	28
5.8.1	Clear Axes.....	28
5.8.2	Recenter View.....	28
5.8.3	Box Zoom.....	28
5.8.4	Update Positions	28
5.9	Show Elements	29
5.10	Face Controls	29

5.11	Node Controls.....	30
5.12	Discretization Controls	30
5.12.1	Discretize	30
5.12.2	Reset Discretization	30
5.12.3	Refinement Factor.....	30
5.13	Output Controls.....	31
5.14	Study Controls	31
5.14.1	Switch Dropdown Mode	31
5.14.2	Toggle Relations.....	32
5.14.3	Switch Current Study	32
5.15	Simulation Controls	32
5.15.1	Run.....	32
5.15.2	Run Study.....	32
5.15.3	Run Test Set.....	32
5.15.4	Run Parallel Test Set.....	32
5.15.5	Stop Simulation After Current Cycle	32
5.15.6	Terminate Simulation	33
6	Simulation	34
6.1	Single Run.....	34
6.2	Test Sets	34
6.2.1	Single-Threaded Test Sets	35
6.2.2	Multi-Threaded (Parallel) Test Sets	35
6.3	Live Plots.....	36
6.3.1	PV Loop Plots	36
6.3.2	Power Output Plot.....	37
6.4	Outputs.....	37
6.4.1	Default Outputs.....	37
6.4.2	Optional Outputs.....	37
6.4.3	Record Only Last Cycle.....	38
7	Optimization Studies	39
7.1	What is a Study?	39
7.2	Gradient Ascent	39

7.3	Creating a Study	39
7.4	Switching Studies.....	39
7.5	Running a Study.....	39
8	References	40

1 Overview

The Modular Single-Phase Model (MSPM) is a software interface and numerical model for designing and modelling low-temperature Stirling engines. MSPM is a two-dimensional third-order model. Steven Middleton initially developed it at the University of Alberta to create a more suitable numerical model for low-temperature Stirling engine performance modelling. Middleton's work is recorded in his thesis, *"A Modular Numerical Model for Stirling Engines and Single-Phase Thermodynamic Machines,"* available at <https://doi.org/10.7939/r3-x8qd-p159>. MSPM will output simulation results like shaft power, temperature, pressure, and pressure-volume (PV) loops. Units in MSPM are meters for distances, Pascals for pressure, Joules for work, Watts for power, simulated seconds for time, Kelvin for temperatures, and rpm or Hz for the engine's rotational speed. MSPM runs on Windows 10/11; MacOS and Linux are not supported.

1.1 Main Functionality

A range of functionality is built into MSPM to facilitate modelling and simulation. These include:

- Modelling a Stirling engine, including the mechanism and heat exchangers
- Running a combined thermodynamic and mechanical simulation of a modelled Stirling engine
- Running a test set with different engines and environmental parameters automatically
 - Both single- and multi-threaded options
- Optimizing engine dimensions, engine speed and charge pressure using a gradient ascent algorithm

1.2 The Program

1.2.1 Installation

MSPM is written in MATLAB and runs in version 2023a, therefore, a major requirement for installing MSPM is to have MATLAB installed. Version 2023a has been tested to work properly, older and newer versions of MATLAB will likely work as well but have not been tested. Once MATLAB has been installed, download MSPM. This can be done by git cloning the repository or downloading the release ZIP file and unpacking it. Place the MSPM folder in a location of your choice. At this point your installation is complete and everything should work as expected. Open the MSPM folder in MATLAB and run the Run.m file to start up MSPM.

1.2.2 Uninstalling

Delete the entire MSPM folder to uninstall.

1.2.3 Starting MSPM

There are two ways to start MSPM. If you already have MATLAB open, navigate to the MSPM folder and add it to MATLAB's PATH, open Run.m, and run it. If you don't have MATLAB open, we have provided you with a windows batch file (MSPM.bat) in the MSPM files. You can create a shortcut of that file and place it anywhere you want, for example your desktop. Then, double click the batch file or its shortcut to run it, which will open MATLAB and immediately launch MSPM.

1.2.4 MATLAB

The MATLAB code plots the outputs and then saves them as GIFs and PNGs, along with the raw data as a .mat MATLAB data file. The models also save themselves as MATLAB data files. The saved model files

allow for direct editing of the models without using the MSPM GUI. There can only be one instance of the MSPM GUI open for each instance of MATLAB that is running.

1.2.4.1 Required MATLAB Toolboxes

The only toolbox required is the “Parallel Computing Toolbox” which is only used for running parallel test-sets (6.2.2).

1.2.5 Code Structure

The MSPM code is split into different classes and helper functions. Each class corresponds to an element of the model, a body, or a connection. Every class holds all the methods that act on the element and holds references to their parent body and any child bodies. For example, the model class contains all the groups in the model, and a group contains all the bodies in the group.

1.2.6 User Configuration File

The User Configuration File is saved in the Config Files folder. It contains any user parameters that must persist across software launches but can change if the user chooses to. The only two parameters in the file are the model, and results save locations. This file is not synchronized to GitHub and is created by the program using the default locations if the file is missing or if the file locations are not valid.

1.2.7 Folder Structure

In the MSPM folder, you will find many subfolders, and a few files. The files in this parent directory are Run.m, the MATLAB script that launches MSPM, startup.m, which adds the required MSPM directories to MATLAB’s PATH on startup, and MSPM.bat, which can be used to launch MSPM when MATLAB is closed. As for the folders, most of them contain MSPM’s source code. The folders which contain files which will be useful to you are the following: Runs, which stores the results of each simulation that has completed, and Saved Files, which stores the MATLAB data files containing your MSPM engine models. These two folders are set as the default folders in the parameters.mat config file but can be changed from within MSPM’s GUI.

1.2.8 Assumptions

MSPM assumes axial symmetry in all components. The GUI shows a cross-section of the model through the central axis of the group. It is possible to create two groups with different axes of symmetry and to bridge connections between them, to model an entirely axially symmetric engine. Bodies can only translate and stretch in the direction of this axis of symmetry. Additionally, solid or gas bodies must fill the entire engine interior, empty space surrounded by other bodies is not allowed. All gasses in MSPM are ideal, kinetic energy and gravitational potential energy are ignored and the speed of sound is effectively infinite. MSPM also assumes all connected gas bodies have a single uniform pressure across all bodies. The program assumes single dimensional gas flow within a body and solid head conduction in two dimensions. Object positions are defined by the mechanical system which has responded to forces generated by the gas system, on the next iteration, the mechanical system transfers forces to the gas system which is then used to define the forces on the mechanical system. Pressure drops are approximated from gas velocities and the modified pressures are used to calculate the forces on the mechanical system.

2 Main Components

2.1 Model

The model class holds all the elements in the engine model. It also has toggles for what outputs to generate and other simulation options. This class also contains the code for showing the model in the GUI, discretizing and saving the model file.

2.2 Group

A group is a collection of bodies around a common axis of symmetry. Groups are intended to model parts of an engine which are axially symmetric about a different axis to the rest of the engine and will connect to the main group using bridges. MSPM displays Groups as a thick set of vertical and horizontal axes; the default $x=0$ and $y=0$ connection location. A model can contain multiple groups with different axes of symmetry. The center axis of a group can be at any angle (default 90° from horizontal). By default, the code creates a group with the axial zero aligned with the plot zero; however, a group can move and rotate using the list view. The group class contains the functionality to form connections between the bodies contained within it and the environment.

2.3 Body

A body is a cylinder or annulus with the center aligned with the axis of symmetry of the group. Each body references the four connections defining the top, bottom, inner, and outer dimensions. A body can be any material, gaseous or solid, and the body takes on the properties of that material. A body can contain a matrix component that modifies how the nodes and faces generate to form either a regenerator or heat exchanger. The body class contains checks for body validation and will highlight the body in red if it finds an issue. Gas bodies cannot be in direct contact with the environment at any point, but this problem will only be shown if showing the environment is enabled (see section 5.9).

2.3.1 Body Validation

MSPM checks some conditions to ensure a body is valid:

- A solid body cannot change its dimensions but can translate in the same direction as the group axis.
- A gas body can both translate and stretch in the direction of the group axis.
- No bodies can be overlapping or inverted.

2.3.2 Discretization Modes

There are two different modes of discretization for bodies: equal and smart partitioning. The modes can discretize solid bodies in axial and radial directions; however, they can only discretize gas bodies in a single direction (preferably the direction of gas flow). For solid bodies, it is possible to use one discretization method in one direction and the other mode in the other direction.

2.3.2.1 Equal Partitioning

MSPM slices the body into several equal partitions in the selected direction, as seen in Figure 2.1. The parameters of each body set the number of partitions. Note that the number of radial partitions affects only the right-hand side of the group axis, so a cylindrical body will appear to have double that many partitions due to symmetry.

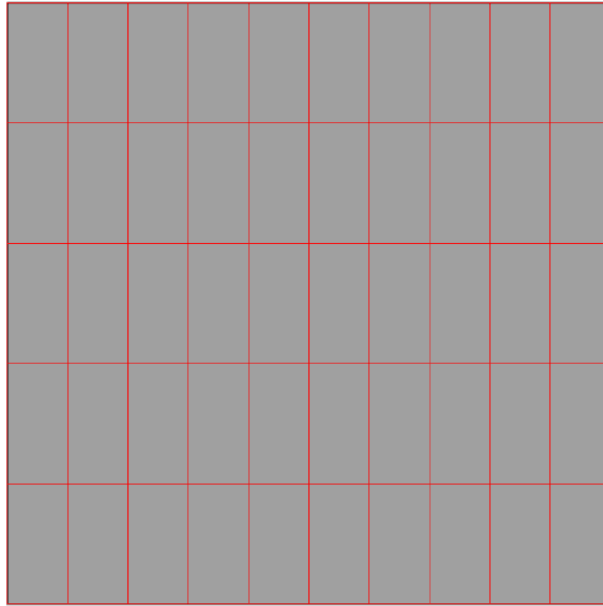


Figure 2.1: 5x5 equal partitioning on a solid body. Source: Primary

2.3.2.2 Smart Partitioning

The body is sliced into smaller nodes at the edges of the body, growing to a maximum size towards the center of the body. For gas bodies only, the divides (either axial or radial) in the direction of discretization must be increased above one in the direction to smart discretize. For example, to smart partition a gas body in the axial direction, as seen in Figure 2.2 (a), set the axial discretization function to smart discretize, and the axial divides to any number greater than one while keeping the radial discretization function blank and radial divides at one. Smart partitioning will only function for gas bodies with a Matrix. A reminder that MSPM can only discretize gas bodies in a single direction.

For solid bodies, any faces touching a gas node have smaller nodes, as seen in Figure 2.2 (b); the oscillation penetration zone (defined by the thermal diffusivity and expected engine frequency) defines the depth. Any faces not touching a gas node are partitioned equally, with no node larger than the maximum node size. For a gas body, the first 15% has smaller nodes with the size growing towards the body's center (Reminder MSPM can only discretize a gas body in a single direction). The parameters for smart partitioning are in the tree view for editing if necessary. They are in the mesher object inside of the model.

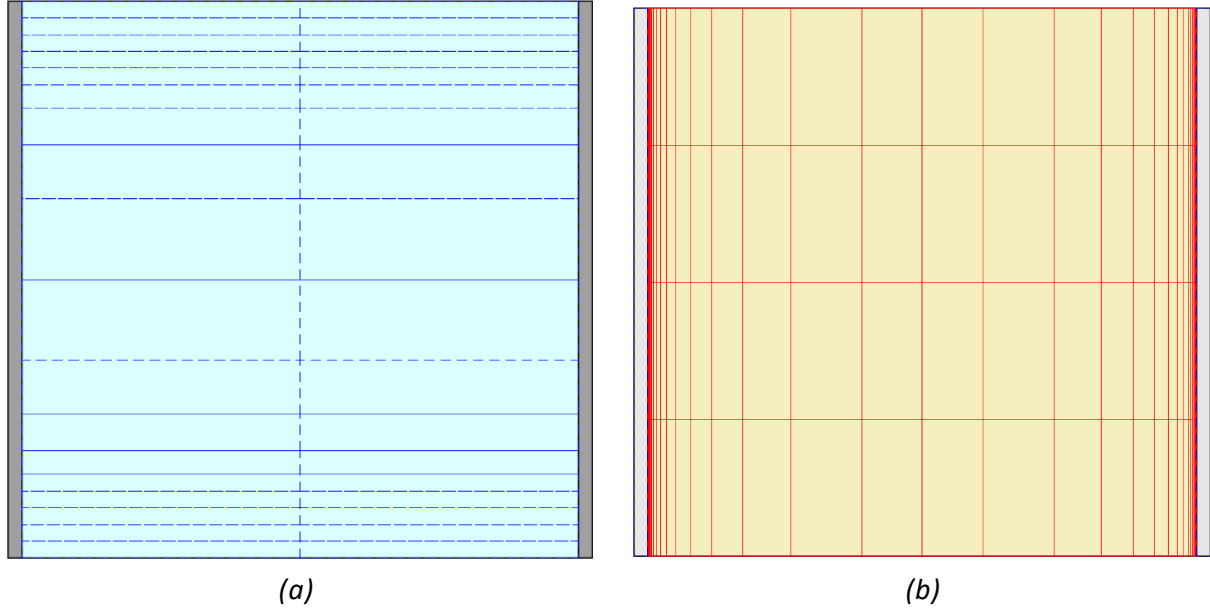


Figure 2.2: Smart partitioning of (a) a gas body containing a matrix and (b) a solid body with a gas body on the left and right.

2.3.3 Custom Heat Transfer Coefficient

In MSPM, a custom heat transfer coefficient can be set for any gas body with a matrix containing a source or a constant temperature body. The coefficient is in units of $\frac{W}{m^2K}$. The surface area used is the surface area of the solid formed by revolving the body around the central group axis.

2.4 Connections

Connections represent a surface as either a cylindrical shell in the radial direction or a circular/annular plane in the axial direction. Each connection has a position, orientation (either axial or radial) and an optional reference to a frame for motion. Connections define the edges of bodies; each body has four connections, one per edge. A connection can use a frame's motion profile to shift its position over an engine cycle. The motion profile will move all body edges associated with the connection. MSPM can only define motion for an axial connection. Connections can be moved manually after creation to change the dimensions and positions of bodies.

2.5 Matrix

Matrices represent a heat exchanger or a regenerator in a model. They can only be attached to gas bodies and add solid internal geometry without changing the external dimensions of the body. A matrix overrides some of the default parameters of a body; it adds a solid, which decreases the gas volume and adds surface nodes and faces to model the conduction of heat into solid components of the heat exchanger/regenerator. A matrix can only have flow in a single direction (the direction of discretization). There are many types of matrices to model common heat exchanger and regenerator geometries, as described below.

2.5.1 Regenerators

Real regenerators almost always have strictly laminar flow [1]; therefore, MSPM only models regenerators as having laminar flow for its calculations. The Darcy-Weisbach equation is used instead of

Darcy's law for these calculations. Schematic representations of the regenerator types are shown in Figure 2.3.

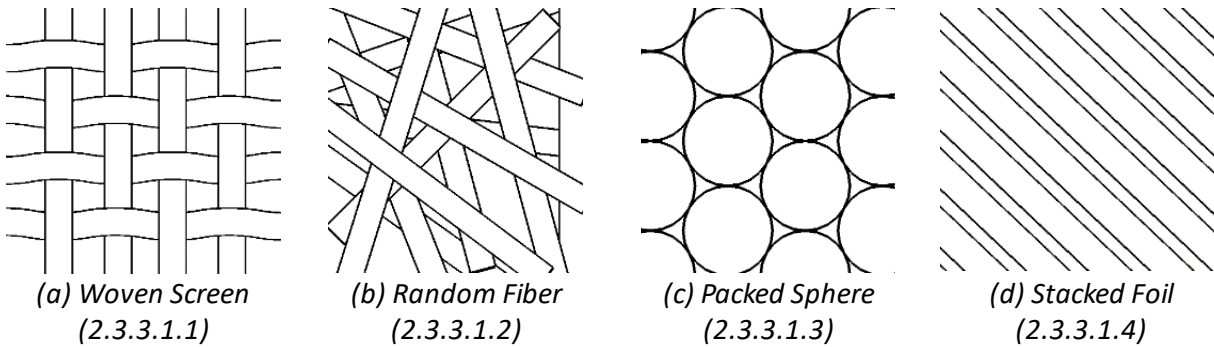


Figure 2.3: Images of different regenerator types. Source: Adapted from [1]

2.5.1.1 Woven Screen

A tight weave of filament arranged in layers perpendicular to the flow direction. It has inputs of:

- Porosity
- Wire Diameter

2.5.1.2 Random Fiber

A felt or batting of randomly oriented fibers. It has inputs of:

- Porosity
- Wire Diameter

2.5.1.3 Packed Sphere

Many small, packed spheres contained such that they don't move. It has inputs of:

- Porosity
- Sphere Diameter

2.5.1.4 Stacked Foil

Many thin parallel channels between thin foil elements. This regenerator type is most similar to an open channel, thus the turbulent flow and Nusselt number are calculated. It has inputs of:

- Gap width
- Thickness
- Surface roughness

2.5.2 Heat Exchangers

These devices improve the heat transfer from a source into the working gas or vice versa. Conduction and convection automatically handle the heat transfer between a gas body and a bare wall; therefore, a heat exchanger would not be necessary to model such a design. A few types of heat exchangers are designed to model real-world heat exchanger designs, as described below. Schematic representations of the heat exchanger types are shown in Figure 2.4.

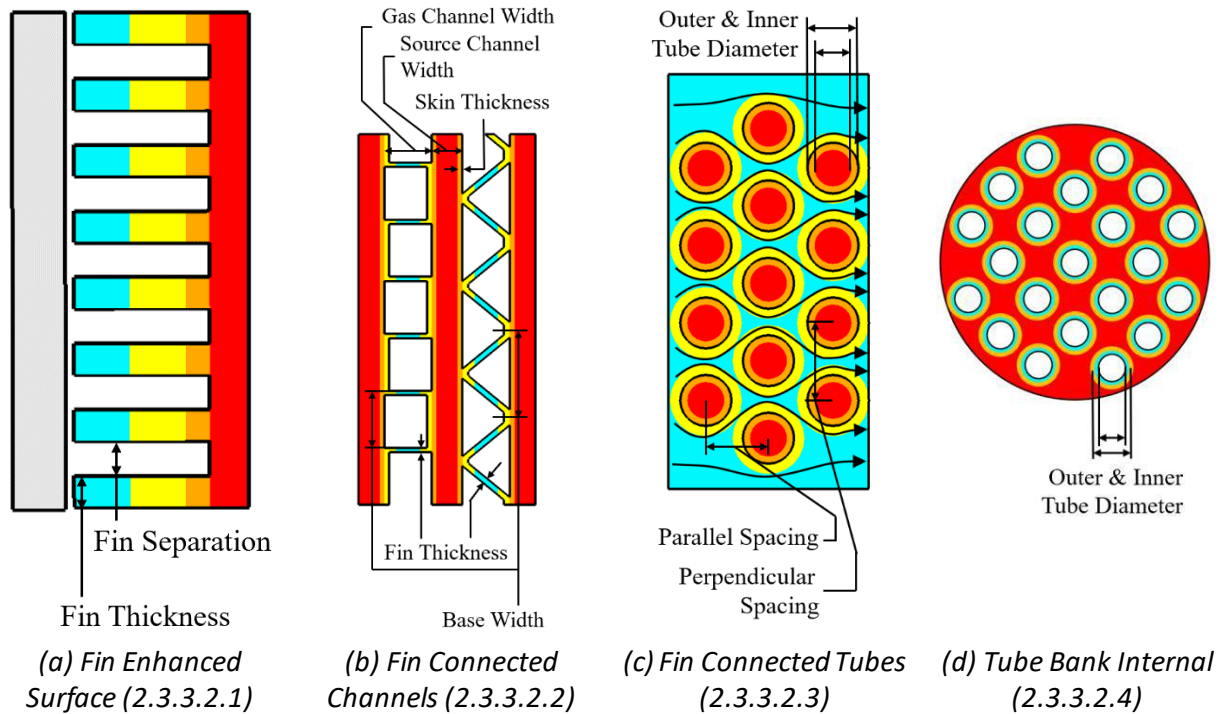


Figure 2.4: Images of different heat exchanger types. Source: Adapted from [1]

2.5.2.1 Fin Enhanced Surface

One of the walls of a solid body next to the gas-matrix body is the heat source/sink for the heat exchanger. This surface is covered in long parallel fins spanning the gas space. It has inputs of:

- Target connection (a connection that the fins are attached to)
- Distance between fins
- Fin thickness
- Surface roughness

2.5.2.2 Fin Connected Channels

This type of heat exchanger encompasses plate and frame heat exchangers. Fluid channels through which a fluid flows to transfer heat run from the heat source/sink to the opposite wall of the heat exchanger through which a fluid flows to transfer heat, connected with thin fins to create rectangular or triangular prism-shaped channels for the working gas to flow through (in a straight line). This heat exchanger type has inputs of:

- Fin type (rectangular/triangular)
- Gas space between source channels
- Source channel width
- Skin thickness
- Surface roughness
- Base width/fin separation
- Fin thickness

2.5.2.3 Staggered Fin Connected Tubes

A heat exchanger where the source/sink fluid traverses across a series of tubes arranged in a triangular tiling covered in surface area enhancing plates perpendicular to the tubes and parallel to the gas flow, like a tower-style PC air cooler. This category encompasses compact heat exchangers as well as finned tube heat exchangers. The fins can be connected to a single tube or form a continuous plate across the entire heat exchanger (default). It has inputs of:

- Spacing perpendicular to the flow
- Spacing parallel to the flow
- Fin thickness
- Fin separation
- Tube outer diameter
- Tube inner diameter

2.5.2.4 Tube Bank Internal

A heat exchanger where the gas is forced to flow through many parallel tubes submersed in a thermally charged environment. It has inputs of:

- Number of tubes
- Outer tube diameter
- Inner tube diameter

2.5.3 Creating Matrices

To create a matrix in MSPM:

1. Create a gas body representing the gas flowing through the matrix.
2. Select the gas body and click on it in the list/tree view to expand its properties.
3. Click change matrix.
4. Select the material of the fins, channel walls or fibers of the matrix.
5. Select the matrix type and enter all required parameters, or leave them as defaults.
6. If you have created a fin-enhanced surface heat exchanger, select the connection which acts as the boundary between the source/sink and the fins.
If you have created a different kind of heat exchanger, set the source/sink temperature. MSPM will assume that the fluid flowing through the tubes/channels will provide a fin surface temperature identical to the set source/sink temperature.
7. Increase the axial or radial divides (only one of those two), to 2 or more, in the direction of gas flow.
8. Change the discretization function to smart discretize in the direction of gas flow; the other should be blank. You have now successfully created a functional matrix.

You may occasionally encounter errors when running simulations after creating a new matrix. The errors usually suggest that some part of your matrix is improperly defined. Carefully follow the steps above to define your matrix if you have any problems.

2.6 Bridge

A bridge allows for the connection between any two bodies. The bodies do not need to be a part of the same group; bridges are the only way to connect different groups. The edges of the bodies being bridged do not have to have the same orientation or be in physical contact. A bridge allows the two bodies to

connect without regard for the physical geometry while still considering the areas of the connecting faces. It would be the same as the two bodies physically touching in the model. The two bodies can also partially overlap if their connecting sides do not have the same shape or the offset is set so they do not completely cover each other. The size of the overlap is calculated at the time of bridge discretization. You cannot have a vertical-to-vertical bridge with different radii. There are four types of bridges, shown in Figure 2.5 and described further below.

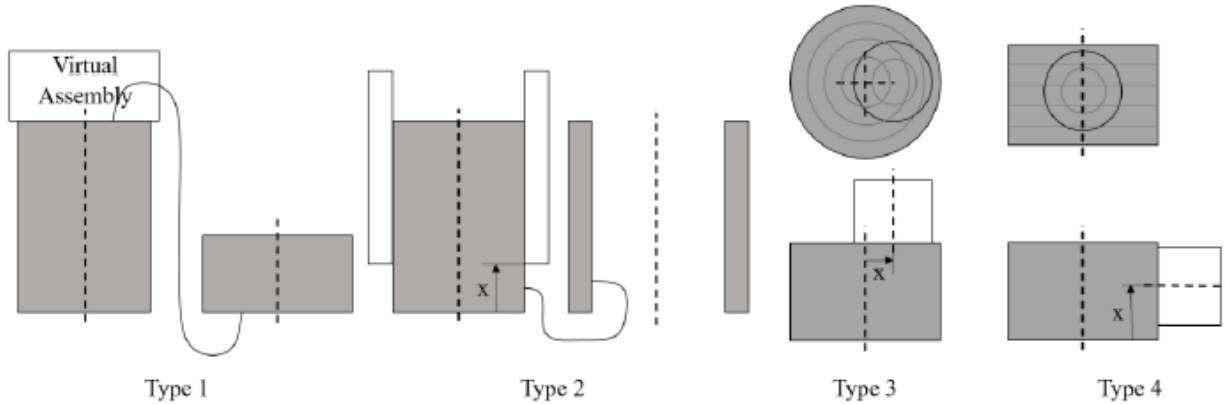


Figure 2.5: The different bridge types (2.3.5). Source: Adapted from [1]

2.6.1 Bridge Types

2.6.1.1 Type 1 – Zero Offset

Type 1 encompasses all bridges where both connections have the same orientation and zero offset. Zero offset means the inner and outer connections are the same for both bodies if the bridge is between horizontal connections, or the top and bottom connections are the same if the bridge is between vertical connections. This type of bridge is not usually necessary, as the model could have been built in a single group. Still, it can be created to clarify the model visual or to make some modifications easier. The secondary connection adds its node contacts to the node contacts of the primary connection and removes node contacts that overlap the added contacts but are not from the secondary body.

2.6.1.2 Type 2 – Misaligned Vertical-Vertical

Type 2 encompasses bridges where both connections are vertical, but the top and bottom connections are not the same between both bodies. To create this bridge, the program will create a series of faces (3.2) with their area equal to the area of overlap between the two bodies (two rectangles). The existing node contacts are then modified by changing their porosity. The software assumes the model is axially symmetric; any resulting faces will also be axially symmetric, so the area ends up stretched around the entire cylindrical outside of the body rather than only on one side.

2.6.1.3 Type 3 – Misaligned Horizontal-Horizontal

Type 3 encompasses bridges where both connections are horizontal, but the inner and outer connections are different between both bodies. To create this bridge, the program will create a series of faces (3.2) with their area equal to the area of overlap between the two bodies (two annular rings). The existing node contacts are then modified by changing their porosity. The software assumes the model is axially symmetric; any resulting faces will also be axially symmetric, so the area ends up stretched around the entire annular face of the connection rather than only on one side.

2.6.1.4 Type 4 – Mixed

Type 4 encompasses bridges where one connection is horizontal, and the other is vertical. To create this bridge, the program will create a series of faces (3.2) with their area equal to the area of overlap between the two bodies (an annular ring and a rectangle). The existing node contacts are then modified by changing their porosity. The software assumes the model is axially symmetric; any resulting faces will also be axially symmetric, so the area of the horizontal connection ends up stretched around the vertical connection rather than only on one side.

2.7 Environment

The environment is any region of the model that does not have a body. The default environment is standard room temperature (300K) air at atmospheric pressure (101.325 KPa). The environment is used in MSPM to simulate environmental losses, such as heat convection from the outer surface of the engine, as well as leaks if any are manually defined. Environmental conditions such as the gas material, temperature, and pressure can be modified. There can be no environment inside of the engine; any area completely enclosed in (solid) bodies must be fully filled with solid or gas bodies.

2.8 Frame

A (motion) frame holds information about linear motion defined by a motion profile (see section 5.3). This is what is applied to a body to make it move and connects the body to the linear-to-rotational motion mechanism to move work into and out of the virtual crank. A frame is applied to connections; when you apply a frame to a body, it is applied to all its axial connections, as only axial connections can move. Each connection that contains a frame will be subject to a pressure, if axial, or a shear force if radial, which will transfer into the mechanism of the model.

2.9 Leak

A leak connection simulates a minor connection between two bodies. It allows for a small amount of gas to flow between the bodies, which depends on the pressure difference between them. A leak connection is defined by two parameters C1 and C2. These are coefficients for the leak equation, which is in units of cubic meters per second. The equation is as follows: $C1 \cdot (dP)^{C2}$, where dP is the difference in pressure between the two bodies, in Pa.

2.10 Relation

A relation is a rule for a set of connections. This allows you to keep certain dimensions or proportions constant in relation with each other, so you can move connections around without having to manually ensure all your wall thicknesses and such are maintained. Relations are also useful for optimizations, where MSPM could move connections to try to optimize the engine. All relations require at least two connections. To add the first two connections to a relation, select two parallel connections. The program will then ask for a name for the relation. It is possible to put more than two connections in one relation; create the relation with two connections selected, and then create another relation of the same type that includes one of those two initial connections, MSPM will not prompt you for a new relation name. There are multiple types of relation:

2.10.1 Constant Offset

A constant offset relation will constrain the distance between connections to be constant. Any connections with this relation will always keep the same absolute relative distances from each other.

2.10.2 Zero X-Based Scale

A zero x-based scale relation will constrain the ratio between the connection distances from the X or Y axis of the group to be constant. For example, if you have two connections, one with an axial position of 1 and another with an axial position of 0.5; if you move the connection at 1 to 2, the program will also move the connection at 0.5 to 1.

2.10.3 Smallest X-Based Scale

A smallest X-based scale relation is very similar to a zero X-based scale relation. The only difference between the two types of relation is that with smallest X-based scale, connections scale about the smallest positioned connection in the relation as opposed to about position 0. This type of relation requires more than 2 connections to be part of it to be useful.

2.10.4 Width Set

A width set relation will scale the outermost connections in opposite directions by the same amount, and scale matching inner connections in opposite directions by the same amount. This is useful for scaling heat exchangers that sandwich a regenerator, to keep the dimensions of both heat exchangers identical. This will require either 4 or 6 connections to be added to the relation.

2.11 Snapshots

Snapshots are essentially a save of the temperatures of the bodies in your model across one cycle. When a simulation finishes, you are asked if you want to save a snapshot. If you do save one, then the next time you start a simulation you will have the option of using that snapshot. What this does is it will set the initial temperatures of the bodies in your model to the temperatures they were at in the snapshot, which will be used to start the first cycle of the simulation. This dramatically decreases simulation time to convergence, since the engine doesn't have to "warm up" to the right temperatures. Snapshots are smart in the sense that if you delete bodies or create new bodies, it will generally figure out what temperatures to assign bodies. Simulations will still run correctly as these are just initial temperatures - MSPM should find approximate final temperatures of bodies within one to three cycles regardless of initial temperatures.

Snapshots are stored within the model save file itself. If you save a snapshot then close MSPM without saving the model, you will lose the snapshot. If you want to delete snapshots, you can find them in the tree view, inside of the model. Clicking on the snapshots object will bring up a window asking you to select snapshots, shift and ctrl work here to select multiple at a time. Once you have selected the snapshots you want to delete, click ok to delete them. Again, this change won't be saved until you save the model again.

3 Other Components

3.1 Custom Minor Losses

If you define a custom minor loss between two bodies, the default calculated Darcy friction pressure loss will be replaced by the custom minor loss you define. The default minor loss, if a custom one isn't set, is based on the initial area, the orifice area, and the final area. The ratio of cross-sectional areas, B , is then calculated by dividing the smaller area by the larger area. Then the loss coefficient K is calculated as

follows: $K = \begin{cases} (1-B^2)^2 & B > 0.76 \\ 0.42(1-B^2) & \text{otherwise} \end{cases}$. When you define a custom minor loss, you must define the loss

coefficients in each direction, where K_{12} is the loss coefficient between the first body you selected, and the second body, in that direction, and vice versa for K_{21} . For example, you can use a custom minor loss to represent a check valve by defining a small loss in one direction and a large loss in the other. However, keep in mind that if the only flow path for a gas is through this custom minor loss, it will flow backwards regardless of the coefficient.

3.2 Faces

Faces are the automatically defined physical interfaces between the surfaces of nodes. There are four different types of faces, depending on the type of bodies that make the face. Faces are how nodes transfer energy and matter (gas) between each other. The properties of faces can change during an engine cycle, and some faces can even disappear for parts of a cycle. For example, if a node in a piston moves past nodes in the adjacent wall, faces will be created for each wall node that the piston node passes by, but these faces will only exist for the increments of the simulation cycle where the two nodes are touching. Below are the automatically calculated parameters for each type of face. These cannot be changed manually but may change throughout an engine cycle.

3.2.1 Gas-Solid

- Surface area: Wetted area of the solid node concerning the gas node.
- Resistance: The conduction resistance from the surface to the center of the node.

3.2.2 Gas-Gas and Gas-Environment

- Area: Overlapping area between the two nodes.
- Friction Distance: The length the face calculates its friction over. It is calculated differently depending on the orientation and size of the nodes.
- Minor loss coefficient: Replaces the Darcy Friction Factor pressure loss if defined.
- Hydraulic diameter: The hydraulic diameter between the nodes.
- Conduction Distance: The distance between the center of the two nodes.
- Stability distance: The minimum dimension of the interacting nodes normal to the orientation of the face. Use for solving for the local Courant number.
- Reynolds Number: Calculated at the start of each cycle and for friction updates.
- Friction function: The function used to calculate the Darcy Friction Factor.
- Shear Factors: The shearing rate of the gas node in the axial direction
- Velocity Factor.: The relative motion of the gas node relative to the nodes around it.

3.2.3 Solid-Solid

- Conductance: Inverse of the resistances from the center of one node to the other

3.2.4 Solid-Environment

- Conductance: Inverse of the resistances from the center of the solid node to the environment

3.3 Materials

A material is essentially a list of real or ideal thermodynamic properties that are given to a body. It is possible to create your own materials for MSPM to use, however there are many premade, commonly used materials provided for you. These materials can have one of two phases of matter: solid or gas. Each body in an MSPM model must have a material. Some idealized material types are also provided, such as a perfect insulator and a constant temperature body.

3.4 Nodes

In MSPM, the process of discretization subdivides bodies into what we call nodes. These nodes are simply subsections of a body which act somewhat independently in a simulation. Increasing the number of nodes in each body of your simulation will increase the accuracy of the results (up to a point), as well as the number of unique sections in the output animated plots (if enabled), as each node has its own properties during a simulation, such as temperature. Keep in mind that increasing the number of nodes in your simulation will also significantly increase computation time. Ideally, tune the number of nodes in each body of your model to balance accuracy and simulation time.

3.5 Non-Connections

Non-Connections are a flag set between two bodies in MSPM that lets the simulation know that those two bodies cannot interact with each other. It will prevent the two selected bodies from exchanging energy or mass. In the program, this is done by removing any faces created between the two bodies. This was implemented because it may occasionally be difficult to accurately model an engine in MSPM due to geometric constraints of the software, so a non-connection can be used to separate two adjacent bodies without physically creating a perfect insulator wall in the model.

3.6 Sensors

A sensor is used when you want to get data about a certain part of your model. A sensor can only be placed in a single body and can be either a point sensor or a line sensor. There are multiple types of sensors:

- Temperature
- Pressure
- Turbulence
- Reynolds Number

Multiple sensors can be placed in the same body, or even the same location within a body, as these sensors are purely virtual; they do not affect the simulation in any way. However, solid bodies can only contain temperature sensors. The results given by sensors can be either phase independent or time independent. A phase independent sensor will return only the values from the last full cycle of the simulation, whereas a time independent sensor will return all values from the start time (0 seconds) to the end of the simulation.

3.6.1 Point Sensor

A point sensor measures values at a single location within one body. This location can be one of:

- Center – at the center of the rectangle representing the body.
- X max – at the maximum X-coordinate of the body, at the center Y-coordinate.
- X min – at the minimum X-coordinate of the body, at the center Y-coordinate.
- Y max – at the maximum Y-coordinate of the body, at the center X-coordinate.
- Y min – at the minimum Y-coordinate of the body, at the center X-coordinate.

A point sensor will save a 2D line plot of the results, a sample of which is shown in Figure 3.1, over time/phase (whichever was selected) once the simulation is complete.

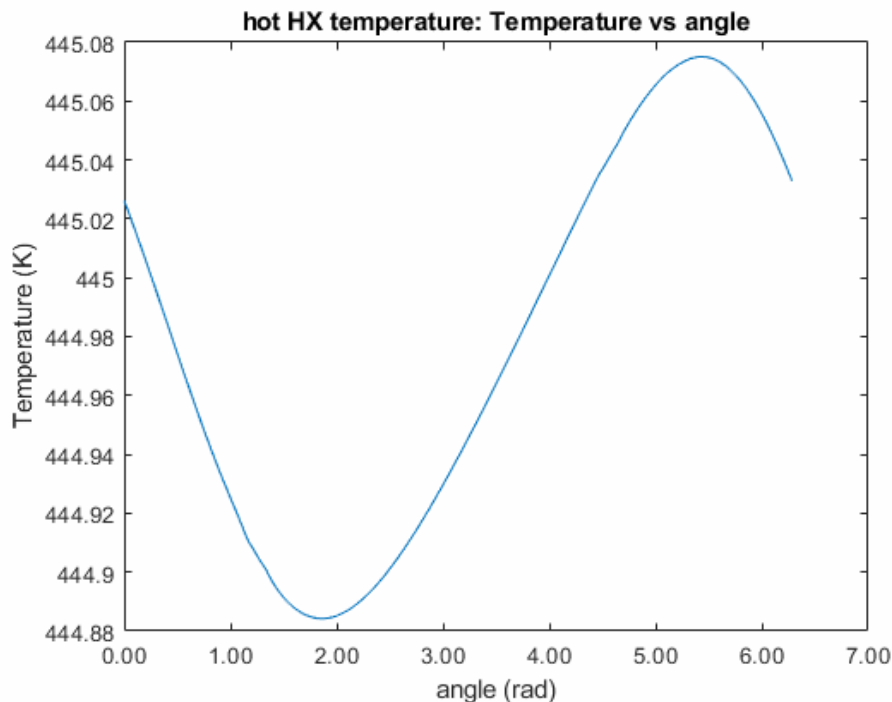


Figure 3.1: The output plot of a point temperature sensor in a gas body. Source: Primary

3.6.2 Line Sensor

A line sensor measures values at multiple discrete locations within one body. You can select the number of discrete points to sample. The line along which the discrete locations will be uniformly distributed can be one of:

- X-axis – parallel to the X-axis, along the center of the body
- Y-Axis – parallel to the Y-axis, along the center of the body

A line sensor will save a 2D colourmap plot of the results over time/phase (x-axis) and location (y-axis), once the simulation is complete. A sample result plot is shown in Figure 3.2.

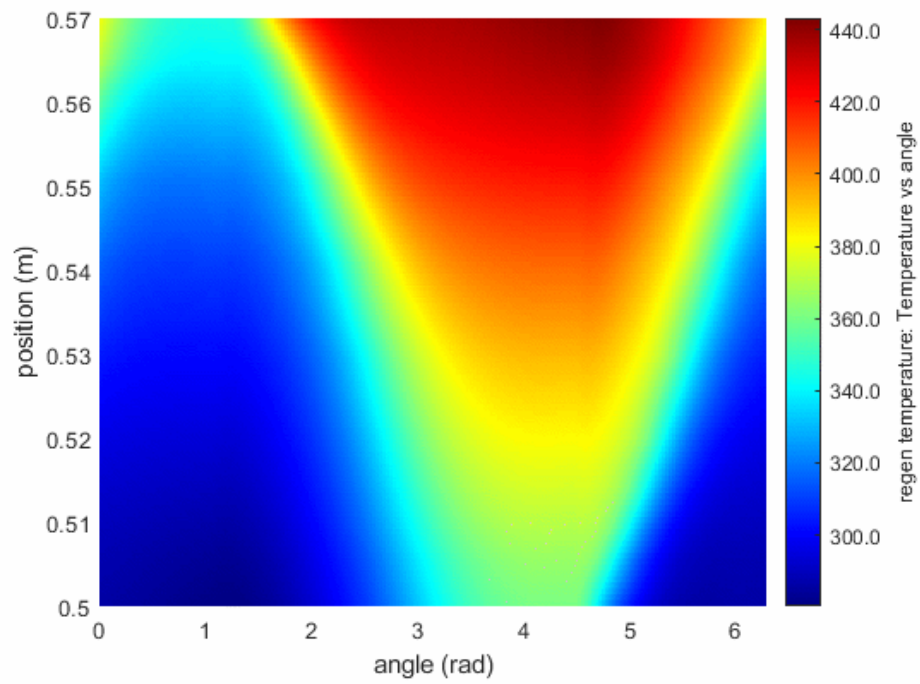


Figure 3.2: The output plot of a line temperature sensor in a gas body containing a regenerator matrix.
Source: Primary

4 Mechanical System

4.1 Linear to Rotational Mechanism

Simulates the crank of an engine by converting rotational motion into linear reciprocating piston motion. The virtual mechanism contains a mechanical efficiency parameter to simulate the friction in a real mechanism. This mechanism will use a motion profile (4.3) to determine the motion of any body that has this linear to rotational motion mechanism, through a motion frame (2.8). Note that piston strokes in MSPM – the total distance that a body will move – can only be positive. This means that you must create all moving bodies/pistons at the bottom of their strokes regardless of their phasing. The phase offset in the linear to rotational mechanism will offset the phase of a body from its initial position of minimum Y location. Setting the orientation to “d” (down) instead of “u” (up) simply adds π radians to the phase input.

4.2 Mechanical System

The mechanical system is connected to every linear to rotational motion mechanism (4.1) and combines all the input and output forces from these mechanisms to determine the shaft power output of the model during a simulation. There can only be one mechanical system per model, as a model can only run at a single speed at any point in time. The mechanical system can have its own mechanical efficiency, which is given by the drivetrain normal friction coefficient, which has a default value of 0 friction, 100% efficiency.

4.3 Motion Profiles

A motion profile is essentially an equation that defines the shape of a pseudo-sinusoid. It converts the constant speed rotational motion of the crank to linear reciprocating motion for bodies in the model, the pistons. There can be multiple motion profiles in a model, as these are associated with the individual cranks of the linear to rotational mechanisms as opposed to the output shaft of the mechanical system. There are 2 built-in motion types in MSPM, an ideal sinusoid and a slider-crank. However, custom motion profiles can be created in the same way that the 2 built-in motion profiles are defined in the source code files; simply create an additional appropriately named MATLAB code file, and program the parameters and math for your own custom motion.

4.3.1 Ideal Sinusoid

This motion profile, as the name implies, defines the motion of a perfect sinusoidal oscillation. Internally, it is defined as a frictionless scotch yoke mechanism.

4.3.2 Slider-Crank

The slider-crank mechanism is commonly used in internal combustion engines and many other engines. This motion profile defines its motion and has all the necessary parameters to accurately model a real slider-crank.

4.3.3 Custom Motion Profile

You can define as many of these as you would like to model real (or not) motion of bodies.

5 User Interface

5.1 Overview

The interface for MSPM is the graphical user interface (GUI), shown in Figure 5.1. Other than defining custom motion profiles, “Test Files” and “Study Files”, everything is done through the MSPM GUI. MSPM’s GUI has been designed for 1920x1080 (1080p) or greater resolution displays, however, it can be resized to smaller or larger sizes.

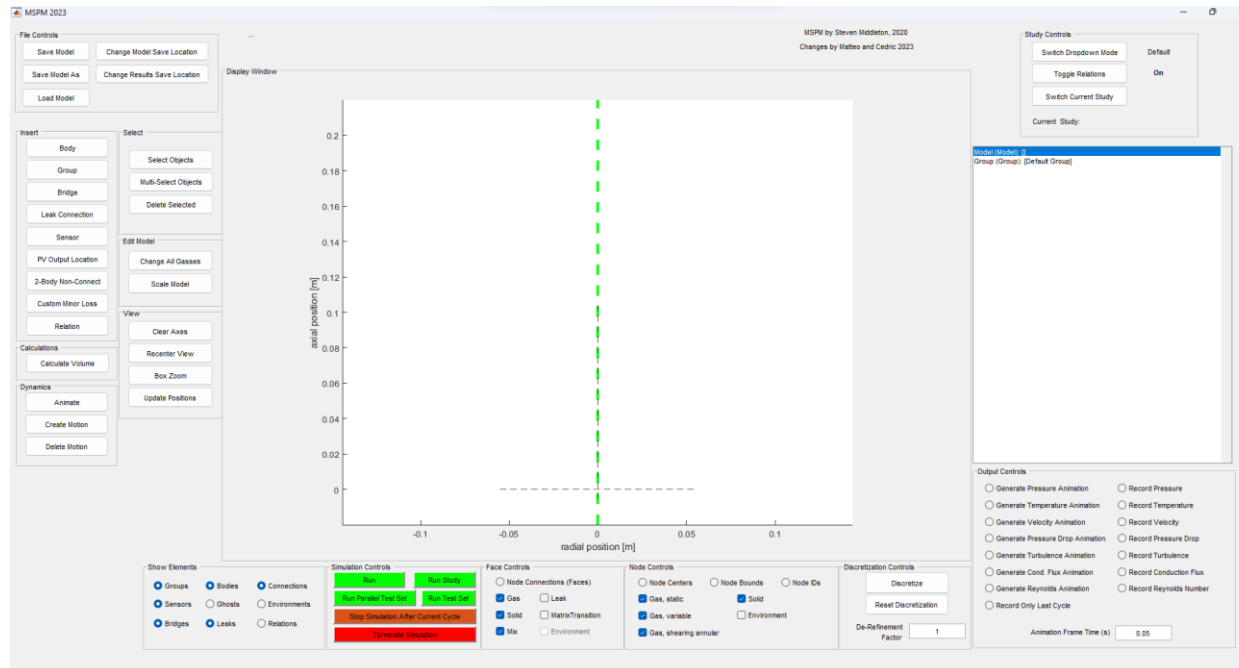


Figure 5.1: The MSPM GUI open after running “Run.m”. Source: Primary

In the GUI window, you can find the display/view window in the center, where the model’s cross-sectional view will be displayed. On the left side, there are buttons to save and load models, buttons to edit models, and buttons to move around the view in the view window. On the bottom of the GUI, there are toggles to select which elements of the model are being displayed, as well as the buttons to run and stop simulations. On the right side there is the list/tree view box, a few toggle buttons in the top right for relations and studies, and toggles in the bottom right for enabling or disabling animated simulation outputs.

5.1.1 Main Window

The main window contains all the buttons you need to create and work with a model. After clicking one of the buttons on the left-hand side of the GUI in either the insert or select sections, the selected button will appear green and a text box above the display window will populate with brief instructions. For example, if you click the select objects button, the text box will inform you that you should click on a object in the display window to select it.

5.1.2 Controls

The main way to control MSPM is using the buttons in the GUI. It has different toggles for outputs and the controls for creating the model. It is technically possible to run the individual functions using just code, but this is not recommended.

5.1.3 List View

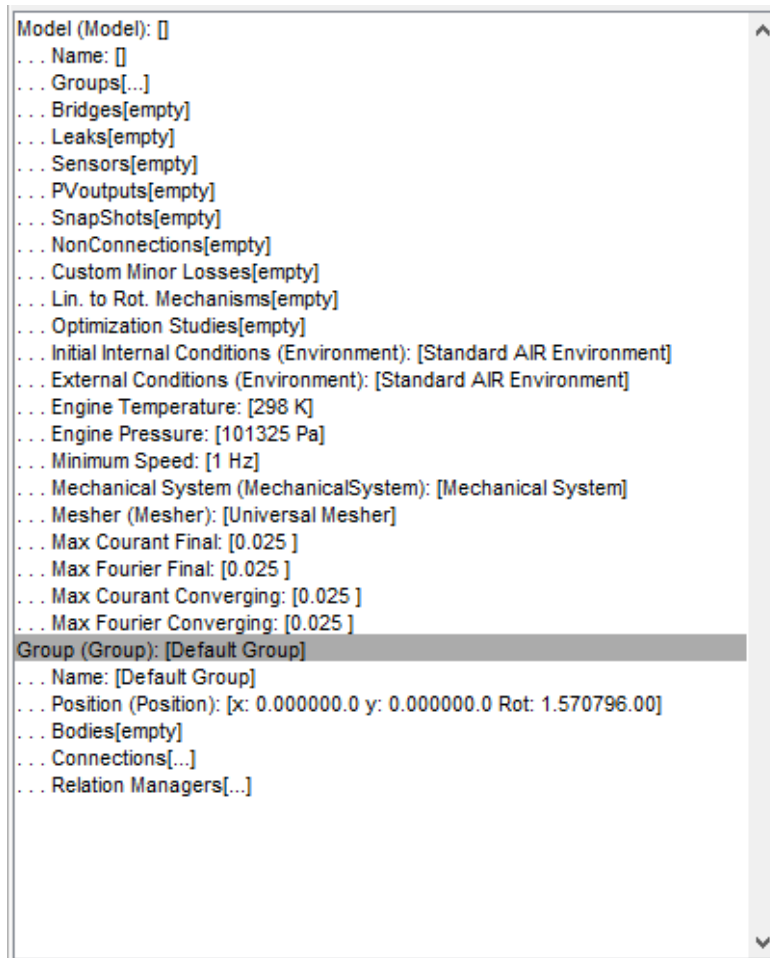


Figure 5.2: The “List/Tree View”. Source: Primary

The list/tree view is the white box on the right end of the GUI, shown in Figure 5.2. It displays the internal object structure of the current model, as well as any additional objects which are currently selected, such as bodies or connections. By default, you will just see the line for the model. To go inside of this object, simply click on it, and it will open the tree one level deeper from the object you clicked on and show all its properties and child objects. To hide the sub-elements of an object, simply click it again to collapse it. From within this white box, all parameters and objects can be viewed and modified, and in some cases deleted, however, you cannot create anything from this view. Some objects, like snapshots or PV outputs, can only be deleted from within this page, as they don’t have any physical representation within the graphical model.

5.1.4 The Display Window

The display window, shown in Figure 5.3, is where the created model is shown, and the user clicks to define positions and what they want to select. In the code, it is a MATLAB plot. It is in the center of the GUI.

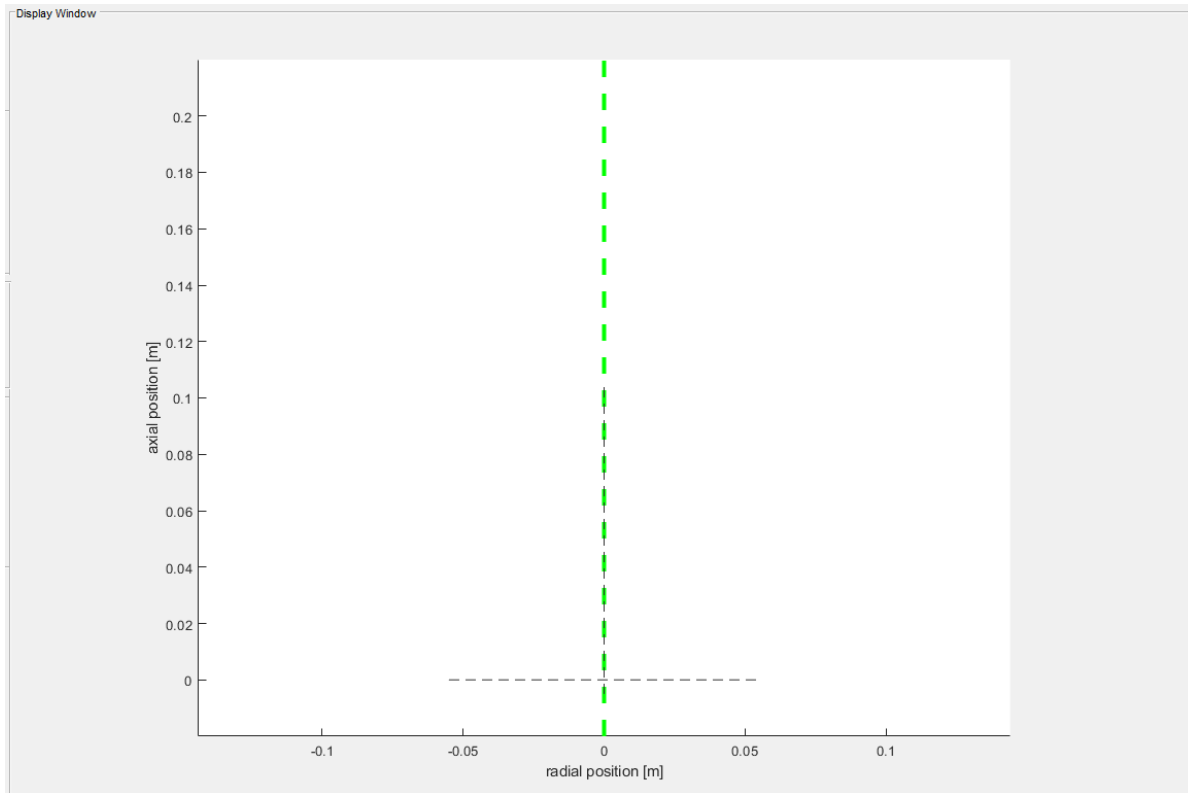


Figure 5.3: The “Display Window”. Source: Primary

5.2 File Controls

File controls allow the user to save and load models in the GUI, shown in Figure 5.4.

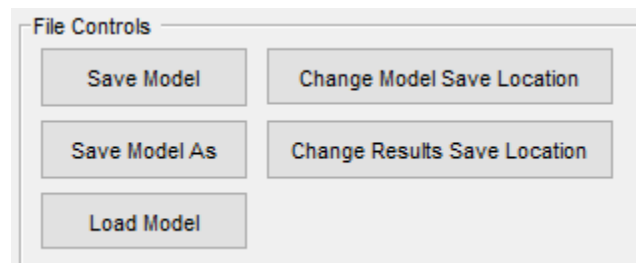


Figure 5.4: The “File Controls” buttons. Source: Primary

5.2.1 Save Model

The “Save Model” button saves the model object (variable) as a MATLAB “.mat” data file. After pressing the button, the program will save the model to the model save location with the current model name. The program will ask if the user wants to overwrite the currently saved model - if the user wants a version history, use the “Save Model As” command (5.2.2). If this is the first time the model is saved, the program will run the “Save As” command instead. WARNING: There is a limit of 256 characters in the full file path, including the file name - ensure the name of the model does not exceed this limit.

5.2.2 Save Model As

The “Save Model As” button performs the same function as the “Save Model” button but allows renaming of the file before saving. After pressing the button, the program will ask for the file name to

save the model as, this will also be the name of the model. The model will be saved in the model save location. If there is already a file with that name in the folder, the program will ask if the user wants to overwrite the old file. WARNING: There is a limit of 256 characters in the full file path including the file name -ensure the name of the model does not exceed this limit.

5.2.3 Load Model

The “Load Model” button loads a previously saved model into the GUI. After pressing the button, if there is a model currently loaded, the program will ask if the user wants to save the current model before loading the new one. Then, the program will bring up a File Explorer window, at the folder set as the model save location, and prompt the user to select a model save file. The selected file is then loaded into the program.

5.2.4 Change Model Save Location

The “Change Model Save Location” button changes the location the model files are saved. This location is stored in the “parameters.mat” file. After pressing the button, the program will bring up a File Explorer window at the folder set as the current model save location. The user can then select a new folder to use as the model save location. This location is then saved to the parameters file.

5.2.5 Change Results Save Location

The “Change Results Save Location” button changes the location the simulation results files are saved. This location is stored in the “parameters.mat” file. After pressing the button, the program will bring up a File Explorer window at the folder set as the current simulation results save location. The user can then select a new folder to use as the simulation results save location. This location is then saved to the parameters file.

5.3 Insert

This section contains buttons to insert objects into the model in the display window, shown in Figure 5.5. When inserting any object, any of the view buttons (other than “Update Positions”) can be used to move the displayed model.

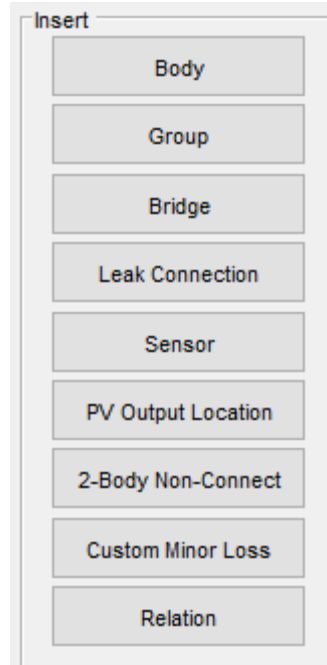


Figure 5.5: The “Insert” buttons. Source: Primary

5.3.1 Body

The “Body” button inserts a body (2.3) into the model. When entering a number manually, the positive radial is to the right and the positive axial direction is up. It is possible to enter a negative number to reverse the thickness/height of a body. Any body inserted below the axial zero will cause the entire model to be shifted to be above zero. To insert a body:

1. Select the “Body” button.
2. Click the first connection in the display window OR right click to enter the inner diameter of the body. Ensure the click is near the group you want to add the body to.
3. Select the second connection in the display window OR right click to enter the radial thickness of the body.
4. Select the third connection in the display window OR right click to enter the axial height of the bottom of the body.
5. Select the fourth connection in the display window OR right click to enter the axial thickness of the body.
6. Select the body material from the dialog.

MSPM automatically selects the nearest connection to the mouse clicks as the user creates a body. If two connections are very close and there are issues selecting the correct one, the user can zoom in with box zoom, or click a bit further away from the connections (as opposed to somewhere in between them). If the user selects an incorrect connection, click the insert body button again to deselect it and cancel the body creation, and then reselect the button to restart.

5.3.2 Group

The “Group” button adds a new group (2.2) to the model. To insert a group:

1. Select the “Group” button.

2. Click where you want to add the group. Only the radial position of the click matters, the axial position is ignored.
3. (Optional) After adding the group, press the “Recenter View” button to ensure the group appears in the display window.

5.3.3 Bridge

The “Bridge” button adds a new bridge to the model (2.6). To insert a bridge:

1. Select the “Bridge” button.
2. Click anywhere in the display window to start placing a bridge.
3. Using the crosshairs, select the group that will contain one of the bridge connections.
4. Click the first connection for the bridge.
5. Select the body associated with the selected connection. This will be the body that is connected through that connection. All the connections between that body and any others that are not part of the bridge will be removed for the side touching the selected connection.
6. Click anywhere in the display window to continue placing the bridge. Before doing this is a good time to move the display window to see the other body you want to bridge.
7. Using the crosshairs, select the group that will contain the other bridge connection.
8. Click the second connection for the bridge.
9. Select the body associated with the selected connection. This will be the body that is connected through that connection. All the connections between that body and any others that are not part of the bridge will be removed for the side touching the selected connection.
10. In the dialog, enter the appropriate offset for the bridge:
 - a. If both selected connections are horizontal, the parameter is the offset between the radial positions of the bridges.
 - b. If both selected connections are vertical, the parameter is the offset between the axial positions of the bridges. Reminder the radius of both connections must be the same.
 - c. If one of the selected connections is vertical and the other selected connection is horizontal the parameter is the axial offset from the bottom connection of the body associated with the vertical connection the horizontal connection will placed.

5.3.4 Leak Connection

The “Leak Connection” button adds a new leak connection (2.9) to the model. To insert a leak connection:

1. Select the “Leak Connection” button.
2. Click the first body of the leak connection.
3. Click the second body of the leak connection OR right click to select the environment.
4. In the dialog, enter the parameters for the leak equation.

5.3.5 Sensor

The “Sensor” button adds a new sensor (3.6) to the model. To insert a sensor:

1. Select the “Sensor” button.
2. Select the body to add the sensor to.
3. In the dialog, enter a name for the sensor.
4. In the dialog, select the sensor position.

5. In the dialog, select the independent variable.
6. In the dialog, select the dependent variable.
7. For a line sensor, in the dialog, select the number of points along the sensor.

5.3.6 PV Loop Output

The “PV Loop Output” button adds a new PV loop output plot (5.3.6) to the model. To insert a PV loop output:

1. Select the “PV Loop Output” button.
2. Select the gas body to add the PV output to.
3. In the dialog, enter the name for the PV output.

5.3.7 2-Body Non-Connection

The “2-Body Non-Connection” button adds a non-connection (3.5) to the model. To insert a non-connection:

1. Select the “2-Body Non-Connection” button.
2. Select the first body of the non-connection.
3. Select the second body of the non-connection OR click outside the model to select the environment.

5.3.8 Custom Minor Loss

The “Custom Minor Loss” button adds a custom minor loss (3.1) to the model. To insert a custom minor loss:

1. Select the “Custom Minor Loss” button.
2. Select the first body of the custom minor loss.
3. Select the second body of the custom minor loss.
4. In the dialog, enter the name and the two loss coefficients.

5.3.9 Relation

The “Relation” button adds a relation (2.10) between two connections. Selecting the “Relation” button will automatically turn on showing connections. To insert a relation:

1. Select the “Relation” button.
2. Select the first connection of the relation.
3. Select the second connection of the relation.
4. In the dialog, select the type of relation.
5. If this is the first relation of the selected type on the selected connections, in the dialog, enter the name of the relation.
6. (Optional) To add more connections to the relation, select a connection with a previously created relation then select the connection to add to the relation.

5.4 Calculations

This section contains the calculate volume button, shown in Figure 5.6, to calculate the volume of the loaded model.

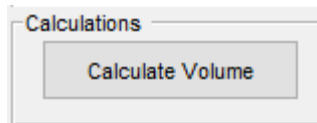


Figure 5.6: The “Calculations” buttons. Source: Primary

5.4.1 Calculate Volume

The “Calculate Volume” button calculates the volume of all gas bodies in the loaded model; except bodies that have the volume calculation flag turned off in their properties. Any bodies with a matrix will have their volume calculated by multiplying their shell volume by the porosity. All other bodies will just use the shell volume. The calculated volume is shown in the MATLAB Command Window.

5.5 Dynamics

This section contains the controls for creating and deleting motion in a model and for animating the model to preview the motion, shown in Figure 5.7.

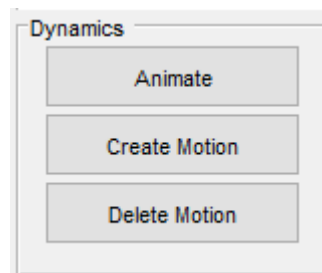


Figure 5.7: The “Dynamics” buttons. Source: Primary

5.5.1 Animate

The “Animate” button shows an animation of the motion of any moving bodies in the model. After the button is pressed, the GUI will show the motion of any moving bodies in the model. To stop the animation, press the “Animate” button again. It may take a while to stop the animation if you have faces turned on. It is possible to show faces in the animation.

5.5.2 Create Motion

The “Create Motion” button opens a new dialog to create a new linear to rotational mechanism (4.1) to add to the model, shown in Figure 5.8. Reminder all pistons must be drawn at the bottom of their stroke - use the phase or orientation to adjust their actual start position in the cycle.

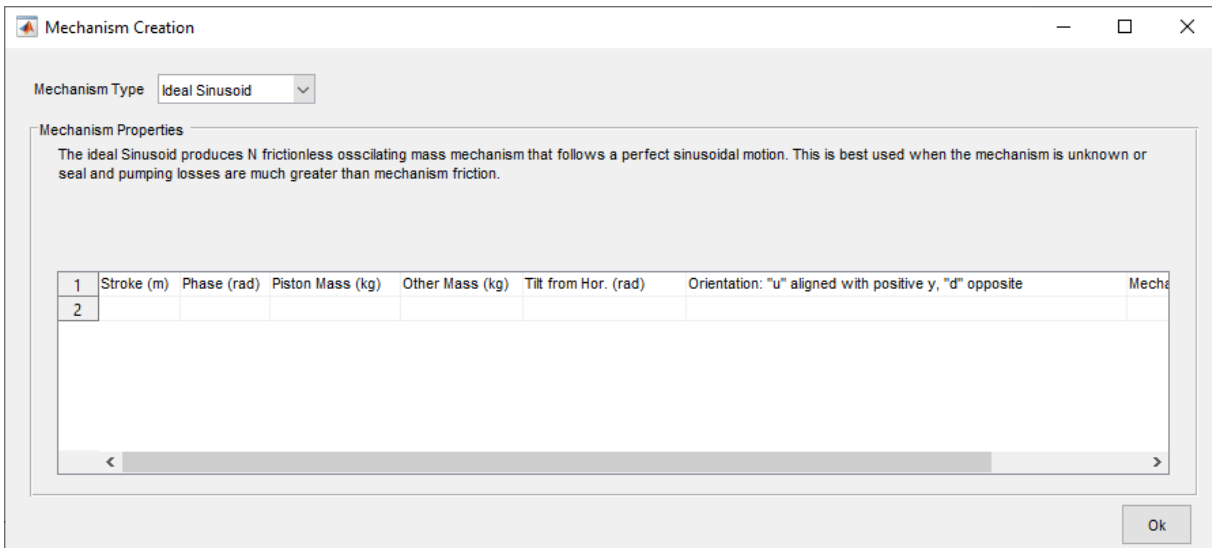


Figure 5.8: The dialog for creating a new motion profile after pressing the “Create Motion” button.
Source: Primary

5.5.2.1 Ideal Sinusoid

The “Ideal Sinusoid” profile is a perfect sine wave motion. The parameters for this motion are:

- Stroke: The stroke of the piston. This must be a positive value.
- Phase: The phase of the piston. This is used to determine the actual piston starting position; it is essentially the place in the cycle the piston is.
- Piston Mass: The mass of the piston.
- Other Mass: The mass of any other mechanism components associated with the piston.
- Tilt from Horizontal: The angle of the piston from the positive radial direction going counterclockwise.
- Orientation: Whether the piston moves up (“u”) or down (“d”) first. (“d” essentially adds 180° to the phase)
- Mechanical Efficiency: The overall efficiency of the mechanism associated with this piston.

5.5.2.2 Slider-Crank

The “Slider-Crank” profile is the modified sine wave produced by a slider-crank mechanism. The parameters for this motion are:

- Stroke: The stroke of the piston. This must be a positive value.
- Phase: The phase of the piston. This is used to determine the actual piston starting position; it is essentially the place in the cycle the piston is.
- Crank Mass: The mass of the crank arm.
- Crank COM Radius: The radius of the center of mass of the crank arm.
- Crank COM Radius: The angle of the center of mass of the crank arm.
- Crank Rotational Inertia: The rotational inertia of the crank arm.
- Crank-Connecting Rod Friction Function: The function used to define the friction between the crank arm and the connecting rod. Takes an input of normal force and outputs an opposing force.

- Connecting Rod Length: The length of the connecting rod.
- Connecting Rod Mass: The mass of the connecting rod.
- Connecting Rod Center of Mass Radius from Crank-Connecting Rod Pin: The radius of the center of mass of the connecting rod from the pin connecting the crank arm and the connecting rod.
- Connecting Rod Rotational Inertia: The rotational inertia of the connecting rod.
- Connecting Rod-Piston Friction Function: The function used to define the friction between the connecting rod and the piston. Takes an input of normal force and outputs an opposing force.
- Piston Mass: The mass of the piston.
- Piston Friction Function: The function used to define the friction of the piston. Takes an input of normal force and outputs an opposing force.
- Tilt from Horizontal: The angle of the piston from the positive radial direction going counterclockwise.
- Slider Offset: The offset for an offset slider crank.
- Orientation: Whether the piston moves up or down first. ("d" essentially adds 180° to the phase)

5.5.2.3 Custom Motion Profile

The motion of the "Custom" motion profile is defined by an external function created in MATLAB. The function interface is: `[pos] = Function_Name(Nang, Phase)` where:

- `pos`: Motion profile output vector. The values should be in the range of [0, 1]. Also, `pos(1) = pos(end)`.
- `Function_Name`: The name of the custom motion profile function.
- `Nang`: The number of angle increments per cycle.
- `Phase`: The phase shift.

This function is then input into the create motion dialog along with:

- Stroke: The stroke of the piston. This must be a positive value.
- Phase: The phase of the piston. This is used to determine the actual piston starting position; it is essentially the place in the cycle the piston is.
- Piston Mass: The mass of the piston.
- Other Mass: The mass of any other mechanism components associated with the piston.
- Mechanism Moment of Inertia: The moment of inertia for the mechanism associated with this piston.
- Tilt from Horizontal: The angle of the piston from the positive radial direction going counterclockwise.
- Custom Profile Function: The custom function defining the motion of the piston with the interface described above.
- Orientation: Whether the piston moves up or down first. ("d" essentially adds 180° to the phase)
- Mechanical Efficiency: The overall efficiency of the mechanism associated with this piston.

5.5.3 Delete Motion

The "Delete Motion" button brings up a dialog with all the created motion profiles and allows the user to delete any selected profiles. Select the profile to delete by clicking on it, then clicking ok. The user can hold shift or ctrl to select multiple profiles.

5.6 Select

This section contains buttons for selecting different parts of the model, shown in Figure 5.9.

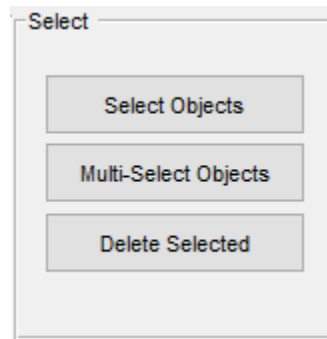


Figure 5.9: The “Select” buttons. Source: Primary

5.6.1 Select Objects

The “Select Objects” button allows the user to select an object in the display window. Clicking in the display window will bring up a dialog box with the objects near where the user clicked. The object selected in the dialog will be highlighted and populated in the list view. Selecting a second object will deselect all previously selected objects. The types of selectable objects are:

- Groups
- Bodies
- Connections
- Sensors

5.6.2 Multi-Select Objects

The “Multi-Select Objects” button allows the user to select more than one object at a time in the display window. The selection works identically to the “Select Objects” button but does not clear the old selection when additional objects are selected.

5.6.3 Delete Selected

The “Delete Selected” button will delete any objects that are currently selected. The button will delete bodies, connections, and sensors; however, groups must be deleted using the list view. If more than one object is selected, it will delete all selected objects sequentially in the order of selection.

5.7 Edit Model

This section contains buttons to edit properties of the entire model at once, shown in Figure 5.10.

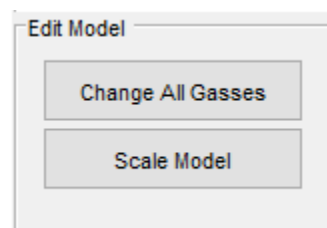


Figure 5.10: The “Edit Model” buttons. Source: Primary

5.7.1 Change All Gasses

The “Change All Gasses” button changes the material of all the gas bodies in the model. After pressing the button, the program will ask the user to select the gas material to apply to all the gas bodies.

5.7.2 Scale Model

The “Scale Model” button will scale the entire model geometrically. After pressing the button, the program will ask the user for the amount to scale the model in the axial and radial directions. Then, the program will ask for which groups to scale. The user can hold shift or ctrl to select multiple groups. The scaling is applied by modifying the location of all connections in the model corresponding to the scaling direction. Matrices are not scaled and thus the internal dimensions and parameters of heat exchangers and regenerators remain the same before and after scaling, however the external dimensions will change. Note that smart discretization parameters do not scale automatically, so you might want to scale those manually in the model’s mesher section in the tree view, if you don’t want to have an unreasonable number of nodes (which will slow MSPM to a crawl).

5.8 View

This section contains options for controlling the display window, shown in Figure 5.11.

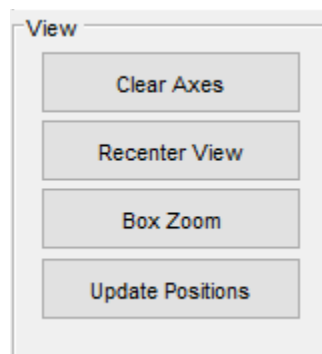


Figure 5.11: The “View” buttons. Source: Primary

5.8.1 Clear Axes

The “Clear Axes” button clears all rendered elements in the view window.

5.8.2 Recenter View

The “Recenter View” button changes the view to show the entire model centered in the window. Works the same as a “Home View” or “Center View” button in most other CAD software. If the user had cleared the axes with “Clear Axes”, this button will redraw everything.

5.8.3 Box Zoom

The “Box Zoom” button allows a user to zoom into different sections of the model. After pressing the button, the user selects two points in the view window using the crosshairs. The view window will then scale to fit the entire area within the box created by the two points while scaling both the radial and axial axes by the same amount to prevent squishing or stretching.

5.8.4 Update Positions

The “Update Positions” button will force update all elements of the model. Usually, objects are only updated if their internal “isChanged” flag is set to true. After pressing the button, the program will

update and validate all the objects in the model. This can fix any rendering and position issues and is strongly recommended to run after changing any elements in the list view.

5.9 Show Elements

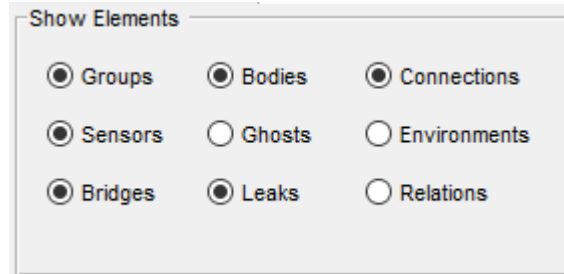


Figure 5.12: The “Show Elements” options. Source: Primary

This section contains toggles to show/hide the elements of a model. Disabling groups will hide the main $x=0$ and $y=0$ axes of the groups. Disabling bodies will hide all bodies from the display. Disabling connections will hide all connection lines from the display. Disabling sensors will hide all sensors from the display. Enabling ghosts plots an additional translucent body in the display window for each body that has a motion frame applied to it. This ghost body will appear at the opposite (top) end of the stroke of the piston. Enabling environments will display a blue line on the outside edge of bodies that touch the environment; the border between your model and the environment. It will also show a red dotted line if there is an empty environment space inside of your model that should not exist. Enabling bridges will display thin lines between the two bodies that are connected by a bridge. Enabling leaks is very similar to bridges. Enabling relations will highlight connections that have relations in a brown colour.

5.10 Face Controls

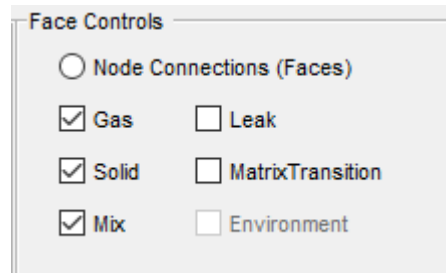


Figure 5.13: The “Face Controls” options. Source: Primary

This section contains options for showing the connections between the nodes (faces) in the discretized model. Showing the faces requires the model to be discretized; if the model is not discretized, it will automatically discretize when you enable showing faces. Each type of face can be shown individually. The environment faces cannot be shown as this is not implemented.

5.11 Node Controls

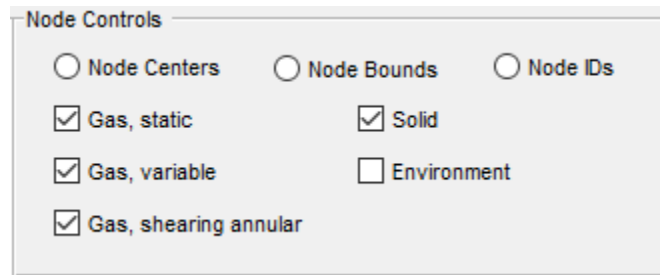


Figure 5.14: The “Node Controls” options. Source: Primary

This section contains options for showing the nodes in the discretized model. Showing any node options requires the model to be discretized; if the model is not discretized, it will automatically discretize when you enable showing nodes. It is possible to show the node bounds, the node centers, and the node IDs. Each type of node can be shown individually and will highlight in different colours.

5.12 Discretization Controls

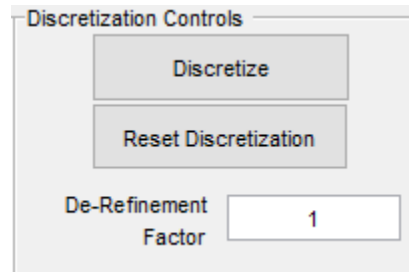


Figure 5.15: The “Discretization Controls” options. Source: Primary

This section contains controls for discretizing the model.

5.12.1 Discretize

The “Discretize” button discretizes the model. The button allows the user to manually discretize the model without running a simulation. This is also used (automatically) for showing faces and nodes.

5.12.2 Reset Discretization

The “Reset Discretization” button removes the discretization from the model. This can help fix rendering issues with showing nodes and faces.

5.12.3 Refinement Factor

The “Refinement Factor” controls the number of nodes when discretizing. A higher number will create more nodes and a smaller number will create less. This will affect the node and face displays; if you increase the number of nodes then you will see those nodes if you enable showing nodes. Before starting a simulation, ensure that you don’t have too many nodes, as this will make the simulation run very slowly.

5.13 Output Controls

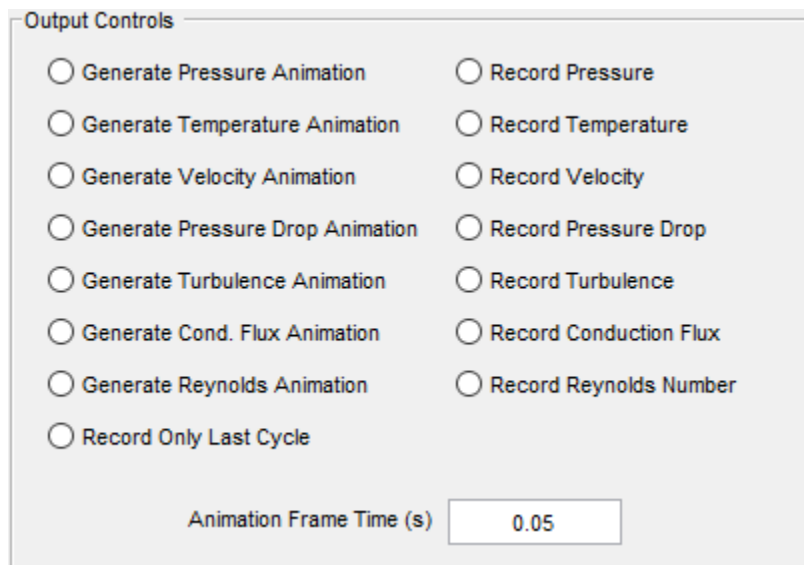


Figure 5.16: The “Output Controls” options. Source: Primary

This section contains the toggles for all the animations and recording the animation raw values to the model file. To generate the animation, the raw data must also be recorded. The “Animation Frame Time” is the in-simulation time between frames of the animation, this will determine the speed of the output animations.

5.14 Study Controls

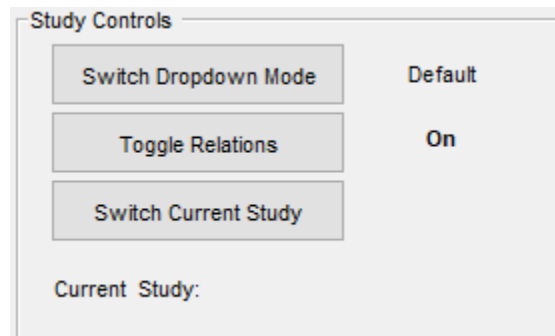


Figure 5.17: The “Study Controls” options. Source: Primary

5.14.1 Switch Dropdown Mode

The “Switch Dropdown Mode” button will change the effect of clicking an object in the display window or in the list view. In “Default” mode it behaves as described in the rest of this manual. In “Optimizer” mode, the program will attempt to add the selected object as a degree of freedom for the currently selected optimization study. Ensure a study is selected before trying to add a degree of freedom. If the current study is “Create New Study” the program will ask for the name of the study. It will then ask for the name of the degree of freedom. If there is no object selected, the program will just ask for the name of the study.

5.14.2 Toggle Relations

The “Toggle Relations” button will turn off/on any created relations in the model. This affects both moving connections within a study and manually changing the location of a connection in the list view.

5.14.3 Switch Current Study

The “Switch Current Study” button will switch between all created studies and “Create New Study” which allows for the creation of a new optimizations study. After the button is pressed, the current study will cycle to the next study in the list. If the current selected study is the last in the list, the next option displayed will be “Create New Study”. The study shown is the study any new degrees of freedom are added to.

5.15 Simulation Controls

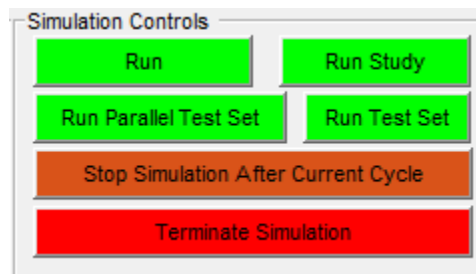


Figure 5.18: The “Simulation Controls” options. Source: Primary

This section contains the controls for running and stopping simulations.

5.15.1 Run

The “Run” button will run a single simulation run using the model loaded in the GUI. After pressing the button, the program will ask for the user to choose a snapshot then enter the running parameters for the simulation (see section 7.1 for more details).

5.15.2 Run Study

The “Run Study” button will run an optimization study on the currently loaded model. The user must use the “Switch Current Study” button to select the study to run before pressing the button. After pressing the button, the program will then ask the user for the “Study File” with the parameters for running the study (see section 7.5 for more details).

5.15.3 Run Test Set

The “Run Test Set” button runs a single-threaded sequential test set. After pressing the button, the program will ask the user for the “Test File” with the parameters for running the test set. (See section 7.2 for more details)

5.15.4 Run Parallel Test Set

The “Run Parallel Test Set” button runs a multi-threaded parallel test set. After pressing the button, the program will ask the user for the “Test File” with the parameters for running the test set. (See section 7.2 for more details)

5.15.5 Stop Simulation After Current Cycle

The “Stop Simulation After Current Cycle” button will stop the currently running simulation after the end of the current engine cycle. After pressing the button, after waiting for the current engine cycle to

complete, the program will produce all the selected and default outputs. This button will not work when running test sets.

5.15.6 Terminate Simulation

The “Terminate Simulation” button will immediately stop the currently running simulation. After pressing the button, the program will stop the simulation and close any open live plots and the progress bar. When terminating a simulation there will be no outputs produced. This button will not work when running test sets.

6 Simulation

In MSPM there are multiple styles of simulation to run. This section only includes simulations with set parameters. Optimizations are covered in the next section. Time in MSPM always refers to simulated time, not real time.

6.1 Single Run

This will run a single simulation using the model currently loaded into the GUI. The user is prompted to first select a snapshot then enter the following simulation parameters:

- Maximum Simulation Time (s)
- Engine Speed (rpm)
- Maximum Time Step (s)
- Engine Pressure (Pa)
- Converging Fourier Number
- Final Fourier Number
- Converging Courant Number
- Final Courant Number

This type of simulation will also show live plots of any PV outputs, power factor and power output. The entire simulation is run on a single thread.

6.2 Test Sets

A test set allows for running multiple simulations without having to start each one individually. The results of each simulation will end up in its own folder in the results save location. To run a test set, the model does not need to be loaded into the GUI beforehand as it is specified in the .mat “Test File” along with all the other simulation parameters. This file constructs a run conditions structure. The “Test File” can contain multiple values for each of these parameters:

- Model ['Model']
- Title (name of the output folder [needs to be unique for each run]) ['Title']
- Maximum Simulation Time (s) ['simTime']
- Engine Speed (rpm) ['rpm']
- Maximum Time Step (s) ['max_dt']
- Source Temperature (K) ['SourceTemp']
- Sink Temperature (K) ['SinkTemp']
- Engine Pressure (Pa) ['EnginePressure']
- De-Refinement Factor ['NodeFactor']
- Heat Exchanger Convection Factor ['HX_Convection']
- Regenerator Convection Factor ['Regen_Convection']
- Convection Outside Matrix Factor ['Outside_Matrix_Convection']
- Friction Factor ['Friction']
- Solid Conduction Factor ['Solid_Conduction']
- Axial Mixing Coefficient ['Axial_Mixing_Coefficient']
- Heat Exchanger Parameter C1 ['HX_C1']

- Heat Exchanger Parameter C2 ['HX_C2']
- Heat Exchanger Parameter C3 ['HX_C3']
- Heat Exchanger Parameter C4 ['HX_C4']
- Heat Exchanger Surface Area/Gas Volume ['HX_SA_V']
- Regenerator Parameter C1 ['Regen_C1']
- Regenerator Parameter C2 ['Regen_C2']
- Regenerator Parameter C3 ['Regen_C3']
- Regenerator Parameter C4 ['Regen_C4']
- Regenerator Surface Area/Gas Volume ['Regen_SA_V']
- Regenerator Porosity ['Regen_Porosity']

The C-parameters, Surface Area/Gas Volume ratios, and the “Regenerator Porosity” only affect custom heat exchangers and regenerators using the equation $N_{Nu} = C_1 N_{Re}^{C_2} N_{Pr}^{0.33}$ and $N_f = C_3 N_{Re}^{C_4}$. It is recommended to remove parameters from the test set file if they are not used, however the first nine are mandatory.

Running a test set does not require any model to be loaded into the GUI before running and will always output all optional outputs. There are two different styles of test-sets:

6.2.1 Single-Threaded Test Sets

A single-threaded test set will run each simulation one at a time, loading each model into the GUI and showing the live plots on the screen like in a single run, however, the animated outputs will be zoomed out to show the full extents of the model. This allows for getting similar outputs as a single run without having to babysit and manually change the parameters for each one. This essentially runs a bunch of single runs back-to-back on a single thread.

6.2.2 Multi-Threaded (Parallel) Test Sets

A multi-threaded test set will run as many simulations as the number of cores in your computer simultaneously (by default). This requires the MATLAB “Parallel Computing Toolbox”. In the toolbox settings you can change the number of parallel workers which is the same as the number of concurrent simulations. A multi-threaded test set will not show any of the live plots while running, or load the model into the open GUI; it just shows just an overall progress bar for the test set. The animated outputs will be cropped from 0-1 on both the radial and axial axes, but all other outputs are still produced. A parallel test set allows for faster running of simulations while sacrificing the animated outputs.

6.3 Live Plots

6.3.1 PV Loop Plots

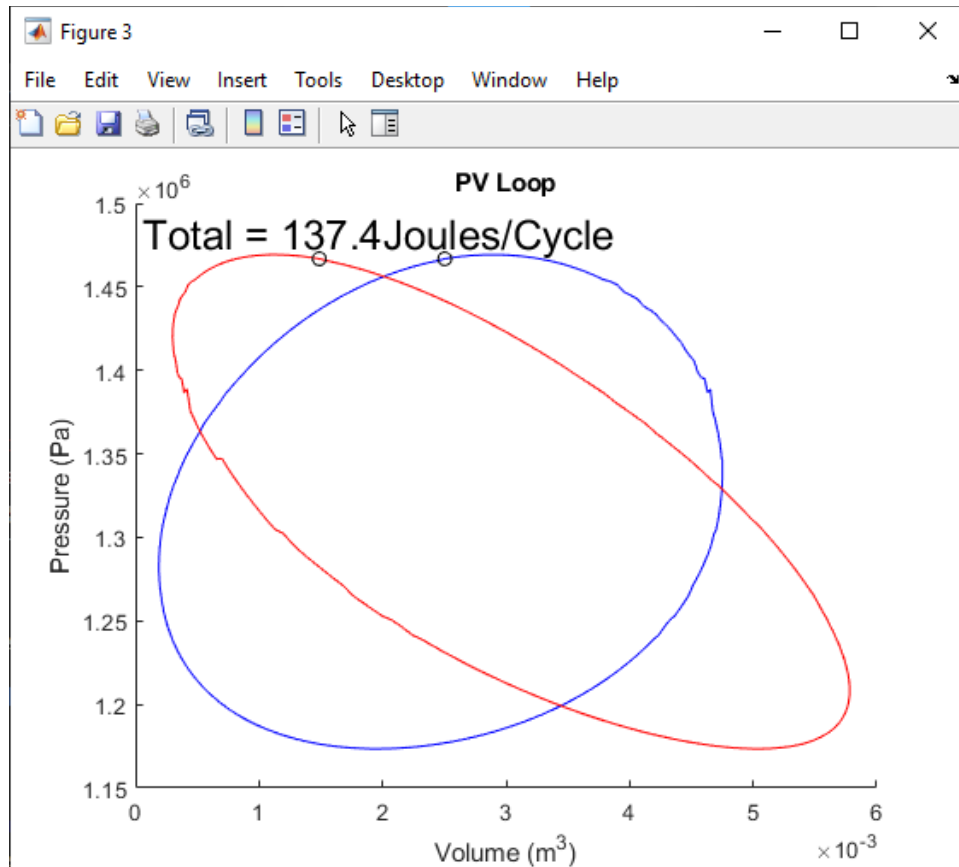


Figure 6.1: The live PV Output at a moment close to the end of a simulation. Source: Primary

When running a simulation, each PV loop will be plotted live as MSPM simulates each cycle of the engine. All connected gas bodies are grouped together and plotted in the same PV loop. From within that list of bodies, only gas bodies that change in volume over the cycle are plotted. Any PV loops that produce negative power are plotted in red, and PV loops that produce positive power are plotted in blue. The circle on each line is the start position of the cycle. The work per cycle on the plot is the sum of all the loops in that plot.

6.3.2 Power Output Plot

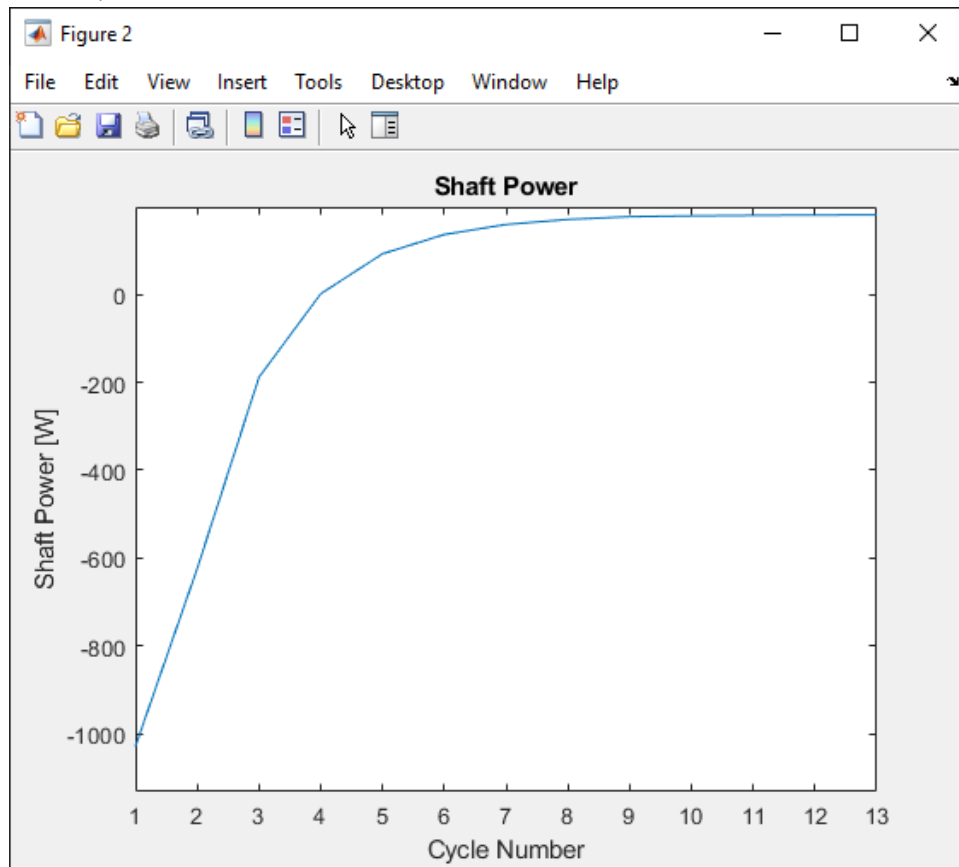


Figure 6.2: The live shaft power output at a moment close to the end of a simulation. Source: Primary

As a simulation runs, a plot of the power output for each cycle is plotted live. When starting from scratch (no snapshot), there will probably be a large difference in power in the first few cycles, then power output will converge to a final value. If starting from a good snapshot where the model had converged, then power output may not change much over the cycles, and this indicates that the simulation will converge quickly.

6.4 Outputs

After a simulation run, MSPM will output some raw data .mat files as well as processed GIFs and images.

6.4.1 Default Outputs

By default, MSPM will output both images and raw data files for:

- Any defined PV Outputs
- Any defined sensors
- A Statistics.mat file with other simulation parameters and output values.

6.4.2 Optional Outputs

Optionally, in the GUI, you can turn on optional GIF outputs and additionally record the raw data for the animation in the Model save file. The time between frames is set using the “Animation Frame Time (s)” parameter. The view of the animations is the same as the view of the model in the GUI at the time the simulation is run. These animations (and associated recorded values) are:

- Pressure
- Temperature
- Velocity
- Pressure Drop/Buildup
- Turbulence
- Conduction Flux
- Reynold's Number

6.4.3 Record Only Last Cycle

The “Record Only Last Cycle” option changes both the animations and the raw outputs to only have the data for the final simulated cycle rather than the entire transient simulation.

7 Optimization Studies

7.1 What is a Study?

A study allows for geometrical optimization of an engine using a gradient ascent algorithm to tune different parameters:

- Max Power
- Max Thermodynamic Power Per Unit Engine Volume
- Max Efficiency
- Max West Number

7.2 Gradient Ascent

MSPM makes small adjustments in the geometry in the direction of positive slope until it reaches the maximum number of iterations or a point at which the RMS of all gradients is below a set tolerance.

7.3 Creating a Study

To create a new study, press the “Switch Study” button until the current study reads “Create New Study”, then click the connection you want to set as a degree of freedom (DOF). This sets the study name as well as the DOF name.

7.4 Switching Studies

To switch which study you want to run press the “Switch Study” button until the study you want comes up. The button will only show studies that have already been created.

7.5 Running a Study

To run a study, the model to be run must be loaded into the display window using the “Load” button. Then ensure the study to be run is selected using the “Select Study” button. Then press the “Run Study” button. After pressing the button, the program will ask for the “Study File”. This file is very similar to the “Test File” for running test sets. It has all the same parameters as a “Test File” except for model and titles. It also has two extra parameters:

- Pressure Bounds (Pa) ['PressureBounds']
- Speed Bounds (RPM) ['SpeedBounds']

These two parameters allow for optimizing the speed and pressure of the model within set boundaries. A “Study File” also has the same mandatory parameters as a “Test File”, except for the model and title parameters.

8 References

- [1] S. M. W. Middleton, "A Modular Numerical Model for Stirling Engines and Single-Phase Thermodynamic Machines," M.S Thesis, Department of Mechanical Engineering, University of Alberta, Edmonton, Alberta, 2021. Available: <https://doi.org/10.7939/r3-x8qd-p159>