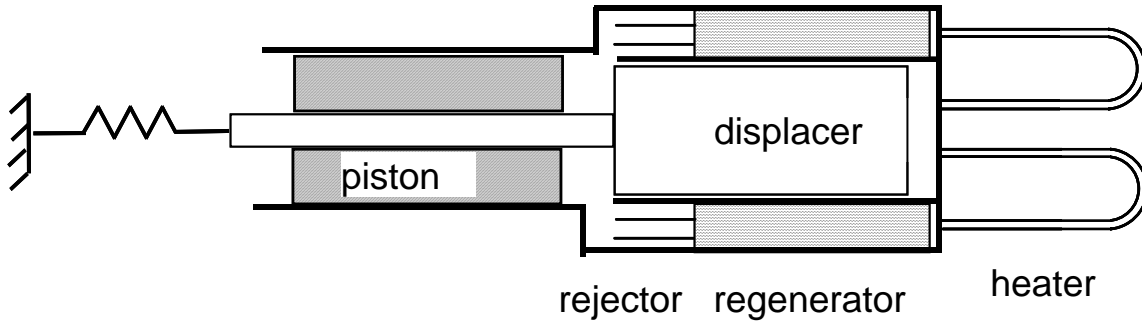


# Sage Model Notes

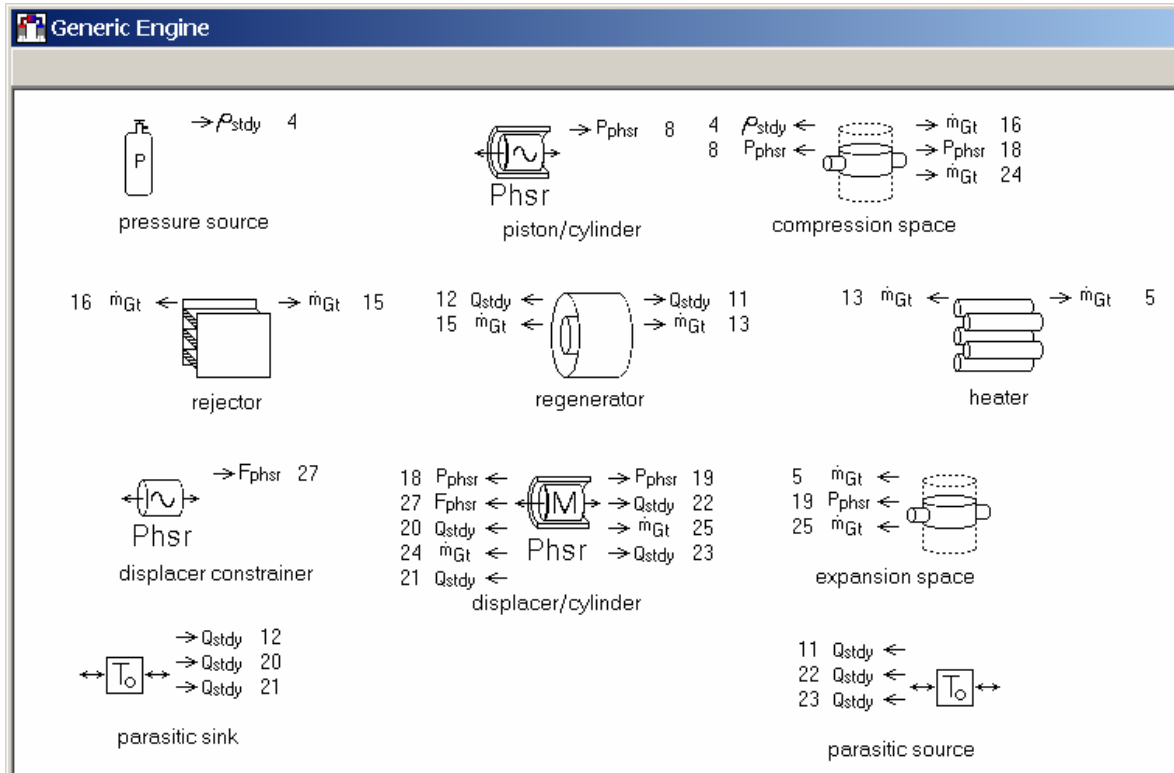
## FPSE.stl

D. Gedeon  
16 January 2009

A model for a displacer-type free-piston stirling engine with displacer arranged co-axially within an annular heat rejector and regenerator assembly and a tubular heater consisting of a number of bent tubes. Here is a rough schematic of the physical layout:



In the Sage model top level the components are arranged with cold parts on the left and hot parts on the right:



The *parasitic source* and *parasitic sink* components anchor the endpoints of the parasitic thermal conduction paths in the model. These are the regenerator pressure-wall, the

displacer and cylinder conduction, including shuttle heat transfer in the appendix gap. They do *not* establish the heater and cooler tube-wall temperatures.

## Piston/Cylinder

Within the *piston/cylinder* a *constrained piston* provides a volumetric displacement to the compression space via its motion (specified as input) and an area attachment connected to the compression space. That area  $A$  is an independent input by default but in this case recast to  $0.25 \cdot \pi \cdot (\text{Sqr}(D_{\text{shell}}) - \text{Sqr}(D_{\text{rod}}))$  so it automatically adjusts to the piston diameter  $D_{\text{shell}}$  input of the *piston/cylinder* and displacer rod diameter  $D_{\text{rod}}$  of the root model (see below) consistent with an annular piston with displacer rod passing through it as drawn in the schematic.

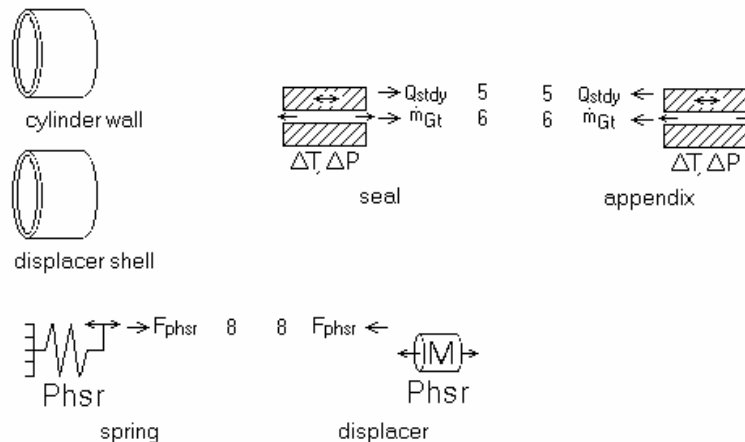
Even though the model claims to be a “free piston” model the piston motion is constrained by inputs  $X_{\text{amp}}$  and  $X_{\text{phase}}$  of the *constrained piston*. In principle it is possible to model the piston as a free *reciprocator*, responding to various forces acting upon it but experience suggests this is a better way. With a truly free piston you also need to model the load, and if not modeled properly the solution can go unstable. With a constrained piston the model is essentially asking the question, what happens if the load is adjusted so the piston amplitude is as specified. The piston phase  $X_{\text{phase}}$  is set to zero and provides the reference phase for the entire model solution.

## Regenerator

Physically, the displacer cylinder resides within the inner wall of the annular regenerator canister. In the Sage model they are two separate components and the physical arrangement is implemented by recasting certain input variables, as discussed below. The outer wall of the regenerator canister represents the pressure wall (important for modeling thermal conduction loss) and the inner wall is just a very thin wall (wall thickness  $W_{\text{in}}$  set very thin) because it is not physically present.

## Displacer/Cylinder

Here are the components inside the *displacer/cylinder*:



The *cylinder wall* and *displacer shell* represent the material and wall thicknesses of the fixed cylinder wall and the thin shell that insulates the displacer body from the hot expansion space. The seal models an annular clearance seal between the compression space and the *appendix*. It has an isothermal solid surface temperature inherited from input  $T_{\text{init}}$  of the parent component. The *appendix* models the annular gas gap between

the *displacer shell* and *cylinder wall*, including the combined conduction of the two walls and the shuttle heat transfer mechanism produced by the thermal interaction between them as they move relative to each other. The positive (right) end of the appendix gap connects to (gas flows to) the expansion space. The conductive surface within the *appendix* (not shown) gets wall thickness and material information from the built-in *cylinder wall* and *displacer shell* components in order to implement a combined wall conduction model.

The *displacer* models a free displacer with inputs for the reciprocating mass and frontal areas exposed to the *compression space* and *expansion space*. The difference between these two areas is often produced by a drive rod or other differential area. There is no drive rod area attachment in the present model since it is presumed to see a constant pressure space and therefore has no effect on the phasor solution of the free displacer. But the difference between the two frontal areas does drive the displacer in the sense that it delivers net PV power to the displacer. Another displacer drive force is supplied by the *spring* attached to the displacer. The combination of differential-area force and spring are what makes the displacer run at the desired phase relative to the piston. More on that later.

During preliminary engine design it is often convenient to forget about free displacer drive forces. For this purpose there is a constrained-motion *displacer constrainer* in the root model to which the internal *displacer* of the *displacer/cylinder* may be connected or not. When connected (current state) the *displacer constrainer* forces the displacer to move with the same amplitude and phase, regardless of any other forces. When disconnected the *displacer* runs free. Normally you would allow the displacer to run free as a test after you have designed the required drive rod and spring.

## Rejector

Physically the *rejector* is an annular structure around the displacer cylinder. In the Sage model the *rejector* does not look anything like an annular structure. But that does not matter because the only thing being modeled is the gas flow through rectangular gas passages with isothermal wall surface boundary conditions set by input *Tinit*. So as long as the total flow area does not exceed that of the regenerator, the model is consistent with physical reality. It is possible to add solid thermal conduction paths representing fins for more realism, in which case it is necessary to make sure the overall frontal areas of the rejector equals the regenerator frontal area. This can be done by appropriately setting the inputs of the thermal conduction paths (better yet recasting the inputs according to primary inputs). For more detailed instructions see the examples in the Stirling Model Class Guide for more information.

## Heater

In this model the heater is presumed to consist of a number of relatively long tubes emerging from the regenerator and bending back around to enter the expansion space. There is nothing to visually indicate this in the model and the same component could equally well model an annular packaging around the displacer cylinder as in the case of the rejector. Like the rejector, the only thing being modeled is the gas flow through rectangular gas passages with isothermal wall surface boundary conditions set by input *Tinit*.

## Annular Geometry Recasts

Three user-defined inputs at the top model level define key dimensions of the annular geometry:

IDpwall	pressure wall ID (m)	8.314E-02
---------	----------------------	-----------

IDregen	regen matrix ID (m)	7.041E-02
Wdiscyl	displacer cyl wall thickness (m)	5.000E-04

These may be assigned like any other inputs, even optimized, and are used to recast inputs of lower level model components. In the *regenerator*, the canister ID and OD are recast as:

**Recasts**

Din = IDregen - 2\*Win  
 Dout = IDpwall + 2\*Wout

**Outputs**

Din	inner-wall ID (m)	7.041E-02
Dout	outer-wall OD (m)	8.514E-02

In the *displacer/cylinder*, the displacer diameter is recast as:

**Recasts**

Dshell = IDregen - 2\*Wdiscyl

**Outputs**

Dliner	cylinder liner OD (m)	8.000E-02
Dshell	moving shell OD (m)	7.900E-02

In the *cylinder wall* within the *displacer/cylinder* the wall thickness is recast as:

**Recasts**

Wcan = Wdiscyl

**Outputs**

Wcan	wall thickness (m)	5.000E-04
------	--------------------	-----------

In effect, the key diameters of the annular geometry are defined at the root level eliminating the need to set them independently in the regenerator and displacer components.

## Axial Geometry Recasts

Dealing with axial geometry constraints is similar to the above but a bit different. A user-defined variable at the top level defines the displacer length to be the combined lengths of the *acceptor* and *regenerator*.

**Outputs**

Ldis	displacer length	1.000E-01
	Lrej + Lregen	

The definition could be modified depending on details of the hardware layout. The reason *Ldis* is a dependent variable instead of an independent input is that it gets its value from other inputs that remain independently set and are exported to the root level via user-defined variables *Lrej* and *Lregen*.

In the *displacer/cylinder* the total length is recast as:

**Recasts**

Length = Ldis

**Outputs**

Length	shell and liner length (m)	1.000E-01
--------	----------------------------	-----------

Physically speaking the lengths of the *seal* and *appendix* sub-components within the *displacer/cylinder* should be roughly the same as the lengths of the *rejector* and *regenerator*, respectively. In the model the *seal* and *appendix* lengths are specified relative to the overall *displacer/cylinder* length by inputs *XNeg* and *XPos*. Ideally these could be recast in terms of the *rejector* and *regenerator* lengths except that it is not possible to recast *XNeg* and *XPos* because they are constants rather than independent inputs (a technicality due to their being used to define the initial temperature distribution, which must be a constant). In this model *XPos* for the *seal* and *XNeg* for the *appendix* are set to 0.25, which is not quite consistent with the actual *rejector* fraction of *Ldis*, but maybe reasonable in terms of typical displacer hardware.

For a model of an actual engine (as opposed to a generic engine) the seal length should be the actual length of a clearance seal, which may be very short if you are using the seal to model the flow past a piston ring. The appendix length should be the length of the part of the displacer shell and cylinder wall where there is an active temperature gradient.

## Bottom-Line Outputs

Net PV power output is available in the root model as user-defined variable *Wnet*. It includes the work done on the *constrained piston* and *displacer constrainer* as provided by two lower level user-defined variables. When the displacer is properly tuned (will run freely at same amplitude and phase), the work done on the displacer constrainer is zero.

Also available in the root model are net heat input *Qin* and heat rejection *Qrej*. Net heat input includes that delivered to the heater gas plus heat flows down various thermal conduction paths. Net heat rejection includes that rejected by the gas in the *rejector* and *seal* plus the same conduction losses.

## Free-Displacer Dynamic Analysis

The model is set up to help you design the appropriate displacer drive rod and spring to make the displacer run free at the same amplitude and phase of the *displacer constrainer*. This is done with Sage's optimizer by specifying the drive rod diameter and spring stiffness as optimized variables and implementing two constraints.

To optimize drive rod diameter requires it be defined as another user-defined input at the root level:

### Inputs

Drod	displacer drive rod diameter (m)	1.591E-02
------	----------------------------------	-----------

To make this work the displacer frontal area exposed to the compression space as recast as  $0.25 \cdot \pi \cdot (\text{Sqr}(\text{Dshell}) - \text{Sqr}(\text{Drod}))$ , where *Dshell* is itself a recast input of the *displacer/cylinder* component as explained above. The displacer frontal area exposed to the expansion space is recast as  $0.25 \cdot \pi \cdot \text{Sqr}(\text{Dshell})$ .

The constraints that make the displacer run free are, in English, that both components of the phasor force supplied by the *displacer constrainer* must be zero. These constraints are implemented within the *displacer constrainer* in terms of the forcing function *FF*:

### Constraints

FF.Real = 0  
 FF.Imag = 0

After the optimization has converged the force connection between the *displacer constrainer* and *displacer* can be broken and the model should run in free-displacer mode at the same amplitude and phase.

## Optimization

In addition to solving free-displacer dynamics the optimizer also optimizes a few other key inputs at the same time. The objective is to minimize net heat input  $Q_{in}$ , subject to PV power equal to 1 kW ( $W_{net} = 1000$ ).

Optimized inputs are:

<i>IDpwall</i> and <i>IDregen</i>	regenerator annular dimensions
<i>Wchan</i>	Rejector channel width
<i>Dtube</i>	Heater tube diameter
<i>Xamp</i>	Displacer constrainer amplitude

In addition the optimizer maintains the dead volume in the *compression space* and *expansion space* by optimizing their *Volume* inputs subject to the constraint that they are 25% larger than the swept volume amplitude in each case. In model terminology, subject to:

$$\text{Volume} = 1.25 * \text{FV.Amp.1}$$

This accounts for the piston and displacer having some reserve stroke capacity and also for any dead volumes in the two spaces at the time when the swept volume is at a minimum.

You might think that it would be better to implement the above volume constraint by simply recasting the *Volume* input to explicitly depend on the relevant amplitudes and area attachments. While possible this is not recommended. The reason is because the actual component displacements of the solution are solved implicitly and can lag behind the values implied by current amplitude inputs immediately after those inputs change. Negative volume errors can result. Sometimes negative volume errors also occur as a result of the optimizer failing to meet the above volume constraint but it is possible to recover from those more easily by just manually increasing the *Volume* input. You can't do that if volume has been recast as a dependent variable.