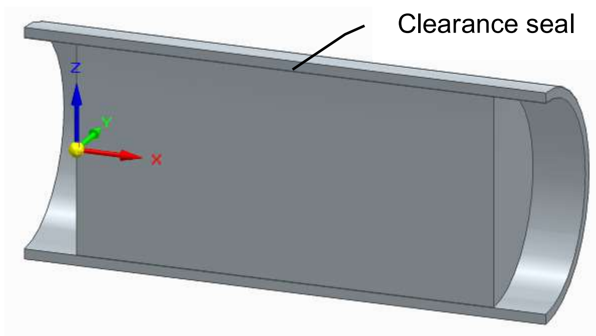
Sage Model Notes

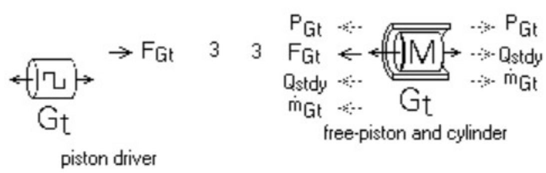
Piston-ClearanceSeal.scfn

D. Gedeon
4 December 2024

A stand-alone piston can be modeled with one of the piston components in the *Moving Parts* pages of the component palette. To model the combination of a piston with seal leakage, requires one of the components on the *Composite* page.



This model is based on the *free-piston and cylinder* component:



To make this model fully functional it must be copied into another model containing a *generic cylinder* (AKA variable volume space) with a mating P_{Gt} connector to connect to either of those emerging from the *free-piston and cylinder* (the other you can remove) Also required are some available mating $\dot{m}_{Gt}$ gas flow connectors and Q_{stdy} heat-flow connectors from heat exchangers and thermal sources or sinks.

The *piston driver* is just a renamed *constrained piston* from the Gt Moving Parts palette. Its purpose is to drive the *reciprocator* within the *free-piston and cylinder* via the F_{Gt} connector between the two. In effect, the *piston driver* produces whatever force is required to make the *reciprocator* move with identical motion.

> **Note:** when part of a free-piston model it is possible to drive the reciprocator with springs and actuator forces by optimizing appropriate model variables. To verify you have done this correctly you can disconnect the piston-driver force and make sure the model still runs the same, or close.

The *piston driver* contains a user-defined input :

```
Xamp             piston amplitude (m)                    5.000E-03
```

And recasts the piston motion as sinusoidal with that amplitude:

```
FX = 0.000E+00...
  (Xamp) Amp
  (0.000E+00) Arg
```

The *free-piston and cylinder* contains user-defined inputs:

```
Mpis            reciprocating mass (kg)                  5.000E-02
Dpis            piston diameter (m)                      2.500E-02
GapSeal         radial clearance gap (m)                 2.500E-05
```

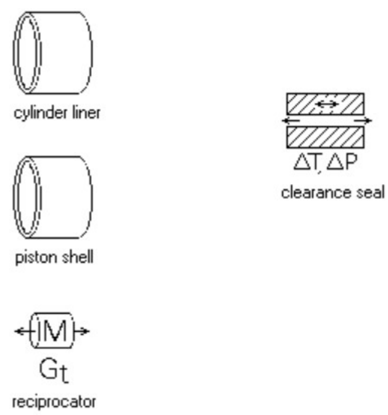
It recasts built-in input `Dshell` as

```
Dshell = Dpis
```

This makes more sense in a larger model where `Dpis` would likely be defined at a higher level. It also calculates a user-define output, useful below:

```
Apis            piston frontal area                      4.909E-04
  0.25*Pi * Sqr(Dpis)
```

Components inside the *free-piston and cylinder* are



The normal purpose of the *liner* and *shell* components is to model thermal conduction in a stationary *outer liner* and moving *piston shell* situated inside. But the presumption of this example is that there is no significant temperature gradient along the shell and liner so that thermal conduction is negligible. That being the case the shell and liner inputs are irrelevant and may be left at their default values.

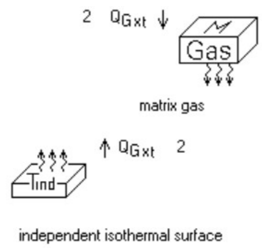
> **Note:** If you want to model thermal conduction you will have to drop in a *distributed conductor* component from the component palette inside either *liner* or *shell* and then thermal attachments from the *Thermal Attachments* page inside the *distributed conductor* as needed for connection to other heat sources or sinks within your model.

The *clearance seal* is a renamed version of the *annulus shuttle/seal/appendix* component from the *Inter-Gap* page of the component palette. Child components within this component model the actual gas leakage. The component itself establishes the radial gap between shell and liner and also models shuttle heat transfer between the *liner* and *shell*, which is implemented by the built Q_{stdy} heat-flow connector arrows. If there is a negligible temperature gradient along the seal these heat-flow connectors can be connected to a single point heat source. If there is a temperature gradient they must be connected to different heat sources, whose temperatures represent the endpoint temperatures of the seal. It recasts the clearance-gap input to

```
Gap = GapSeal
```

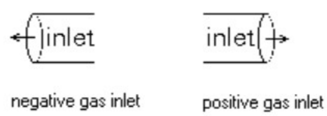
This too makes more sense in a larger model where `GapSeal` would likely be defined at a higher level.

Inside the *clearance seal* are, a *matrix gas* and an *independent isothermal surface* from the *Gas Domain* and *Cylinder Walls* pages of the component palette.

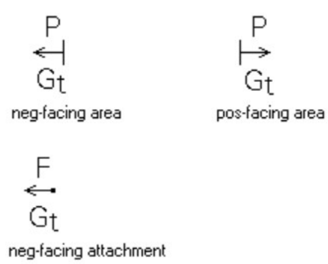


The *isothermal surface* anchors the gas temperature in the seal in this example. An isothermal surface is recommended because it is easiest to implement, well behaved from a solution convergence point of view and there is usually a negligible amount of heat transferred in the seal anyway. For more realism or other reasons you can instead anchor the gas with a *conductive surface*, following the guidelines of sample model *HeatExchangers-ThermalConductors.scfn*.

Within the *matrix gas* are negative and positive gas inlets, whose connectors have been bumped up to the *free-piston and cylinder level* for connection to other components.



Within the *reciprocator* are positive and negative facing area attachments.



Both have area inputs recast as

```
A = Apis
```

so that you don't forget about them, their P_{Gt} connector arrows have been bumped up to the *free-piston and cylinder* level for connection there to variable-volume gas domains, such as an expansion or compression space in a stirling-cycle model.

There is also a force connection, whose F_{Gt} connector arrow has also been bumped up to the *free-piston and cylinder* level for connection to the *piston driver*.