



Visit the mor4ansys web page

Author: Evgenii Rudnyii, <http://Evgenii.Rudnyi.Ru/>

■ Imtek`Post4MOR`: First Order System Example

The goal of the package is to make transient or harmonic simulations for a system of ordinary differential equations of the first or second order. I have written it to support simulation for a reduced model obtained by mor4ansys.

This notebook contains an example for a first order system. The manual is in Post4MORmanual.nb.

We will consider the Microthruster Benchmark.

■ Setting Up Directory and Loading Functions

You may need to change the directory name.

```
In[1]:= SetDirectory["."]
```

Loading functions (if file is in another directory, use the full file name).

```
In[2]:= << Post4MOR.m
```

An alternative way to load functions directly from my site (provided you are connected to Internet and starting *Mathematica* 5.1).

```
ToExpression[Import["http://Evgenii.Rudnyi.Ru/soft/Post4MOR/Post4MOR.m", "Text"]]
```

Path to examples (you may need to change it).

```
In[3]:= filePath = ToFileName[{ "ex" }];
```

Read in a reduced model

```
In[4]:= sys = ReadSystem[filePath <> "thruster"]
```

```
Out[4]= DynamicSystem[{25,1,7}, ...]
```

■ Transient Simulation

Read in ANSYS results for the transient simulation

```
In[5]:= ansys = ReadResult[filePath <> "thruster.transient"]
```

```
Out[5]= - SimulationResult -
```

Using NDSolve with TransientSolution Function

Perform a transient simulation for the reduced model (it has a dimension of 25)

```
In[6]:= res25 = TransientSolution[XSeries[ansys], sys, Verbose → True ]
```

```
NDSolve has made 735 steps for 1.236149 Second
```

```
Out[6]= - SimulationResult -
```

Perform a transient simulation for a reduced model of dimension of 5

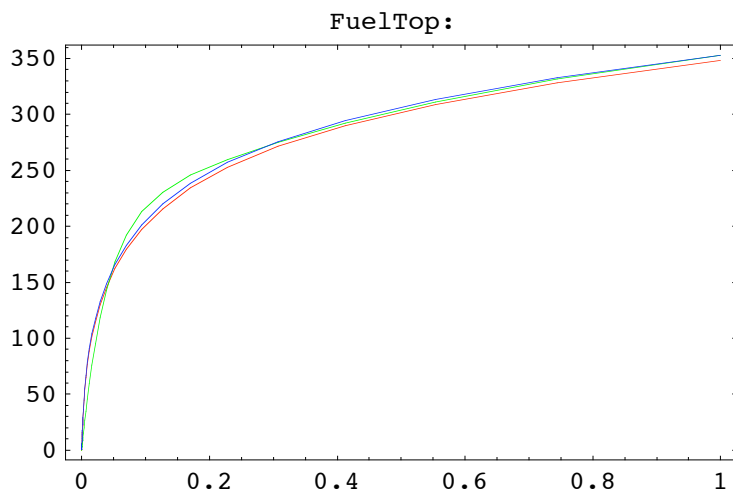
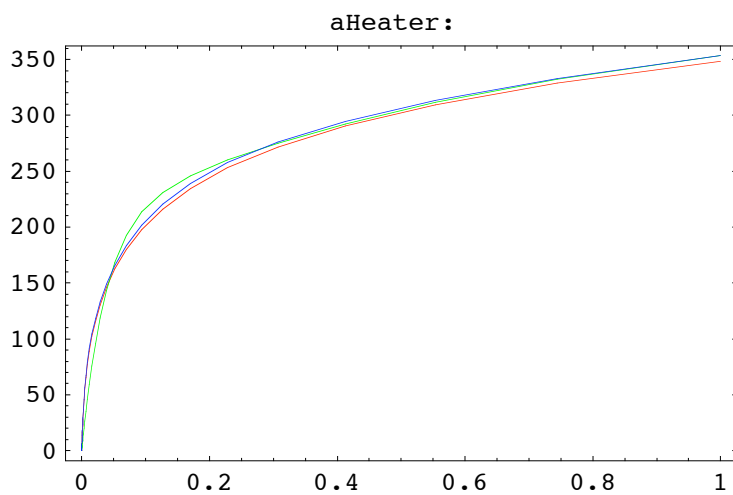
```
In[7]:= res5 = TransientSolution[XSeries[ansys], TakeSystem[sys, 5], Verbose → True ]
```

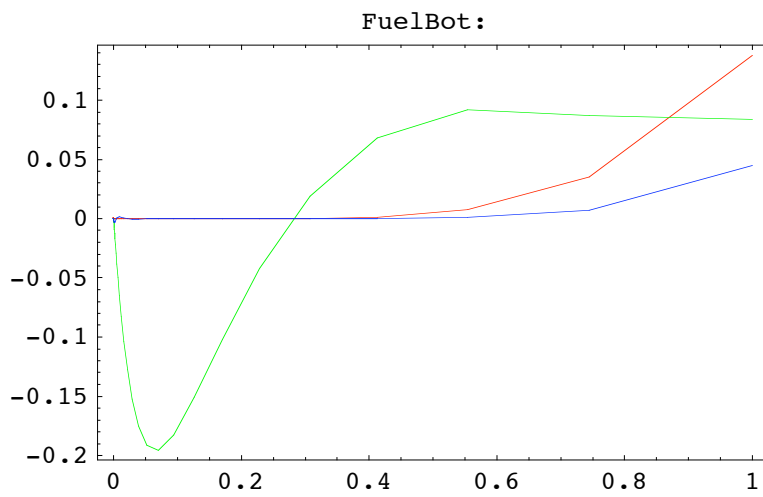
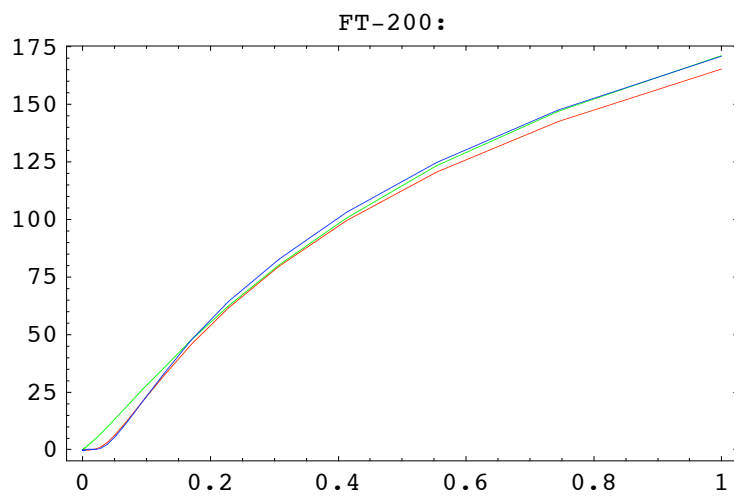
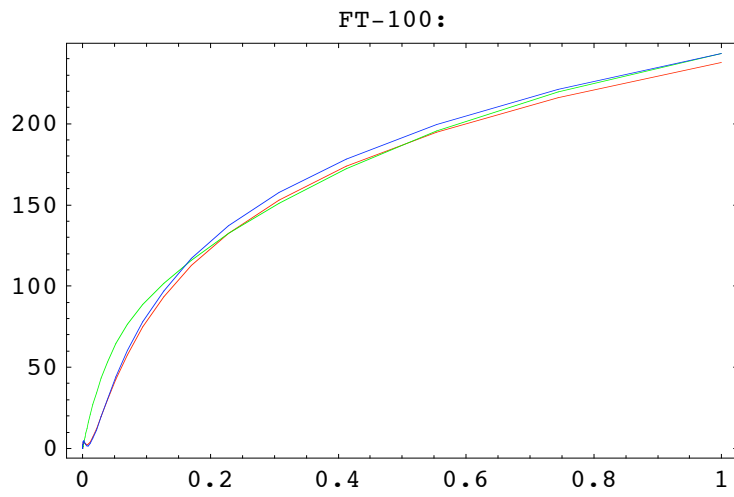
```
NDSolve has made 245 steps for 0.237637 Second
```

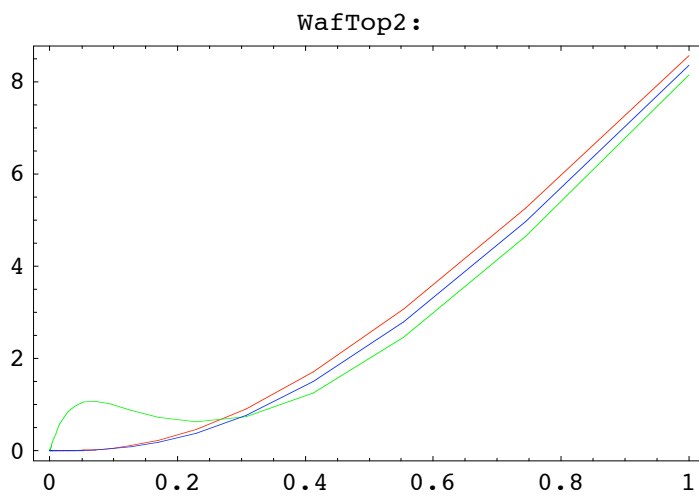
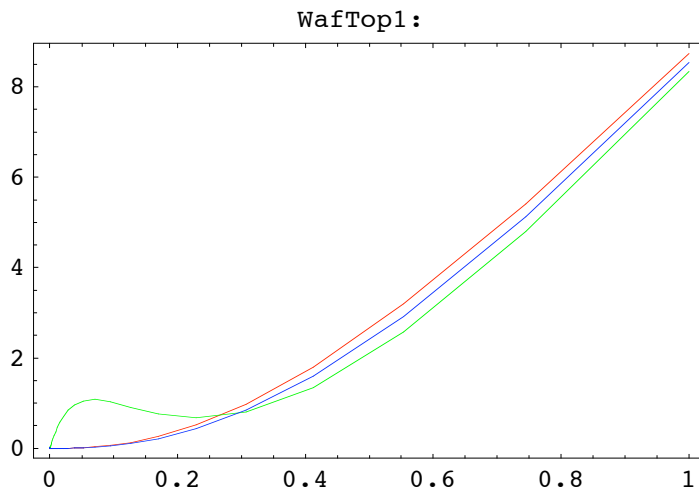
```
Out[7]= - SimulationResult -
```

Make a plot (ansys is red, res5 is green, res25 is blue)

```
In[8]:= PlotResult[{ansys, res5, res25},  
PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 1, 0], RGBColor[0, 0, 1]}];
```

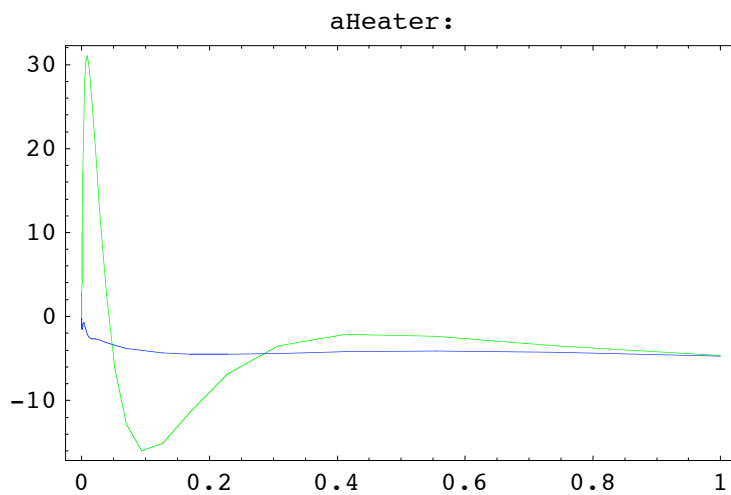


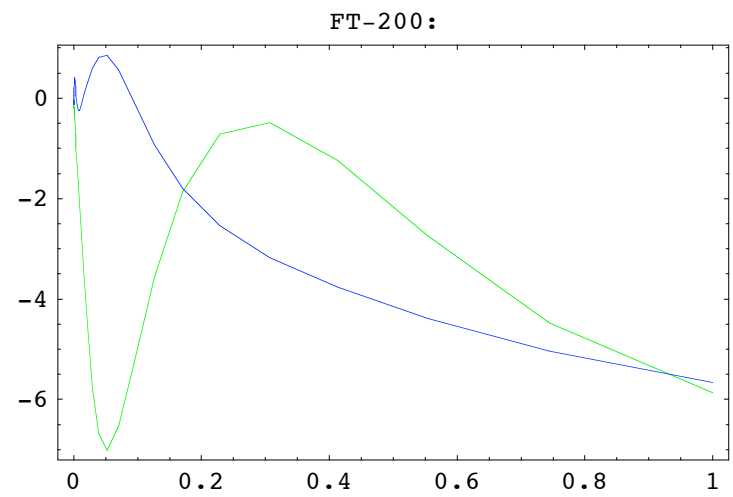
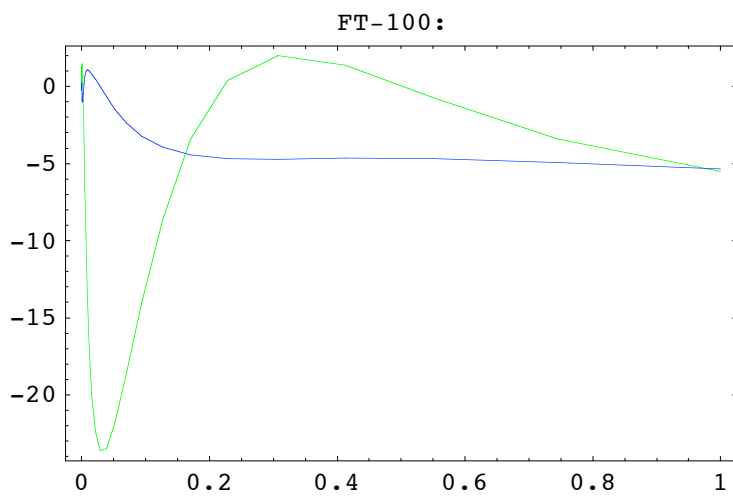
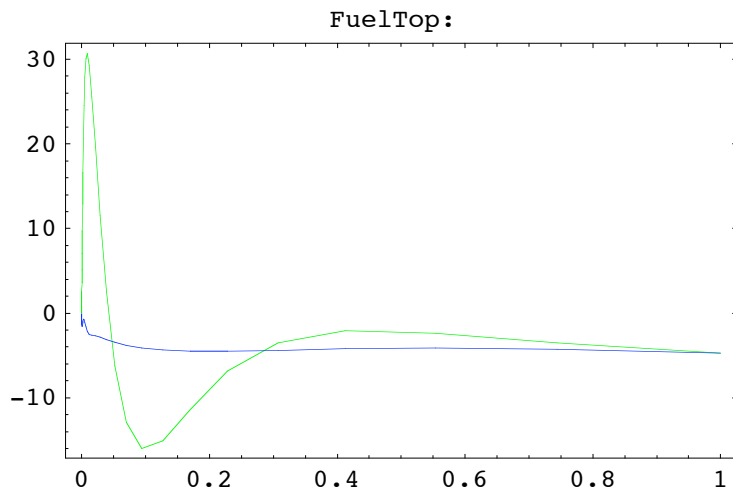


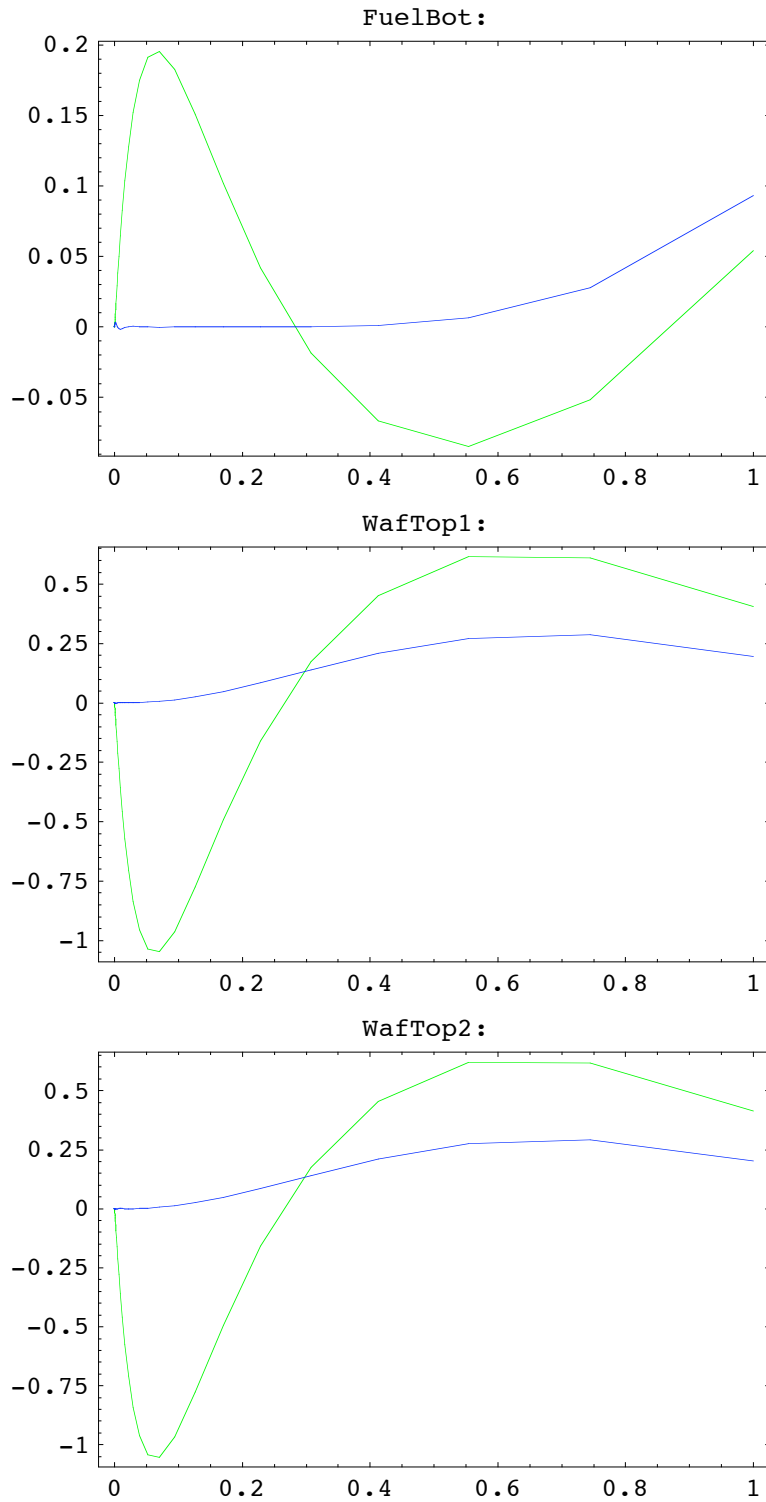


Make a difference plot

```
In[9]:= PlotResult[Difference[ansys, {res5, res25}],  
PlotStyle -> {RGBColor[0, 1, 0], RGBColor[0, 0, 1]}];
```







Using Ansys Integrator with AnsysTransientSolution Function

Perform a transient simulation for the reduced model (it has a dimension of 25)

```
In[10]:= res25 = AnsysTransientSolution[xSeries[ansys], sys]
```

```
Out[10]= - SimulationResult -
```

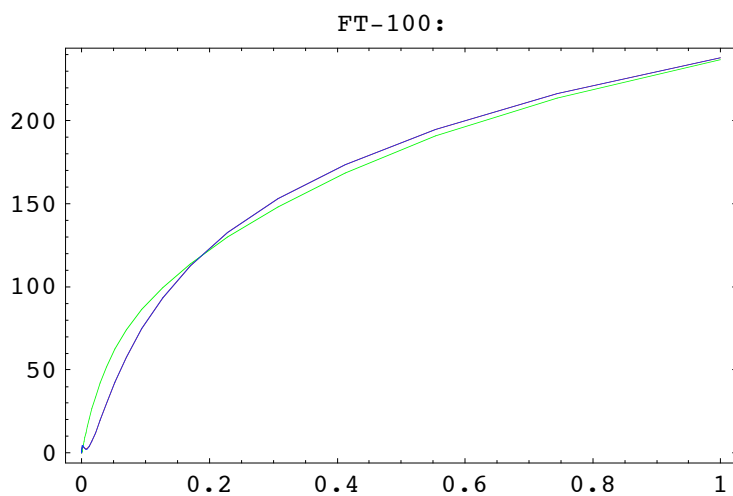
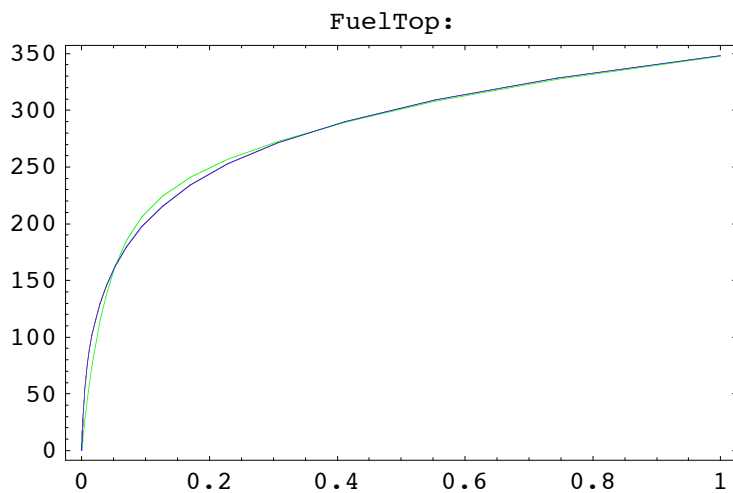
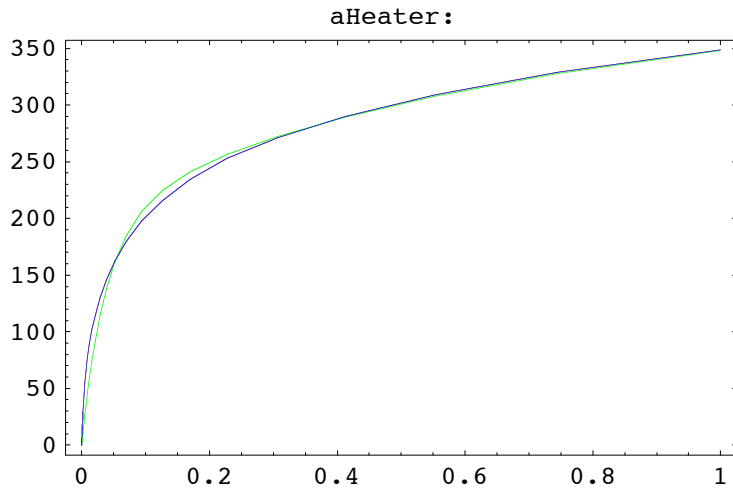
Perform a transient simulation for a reduced model of dimension of 5

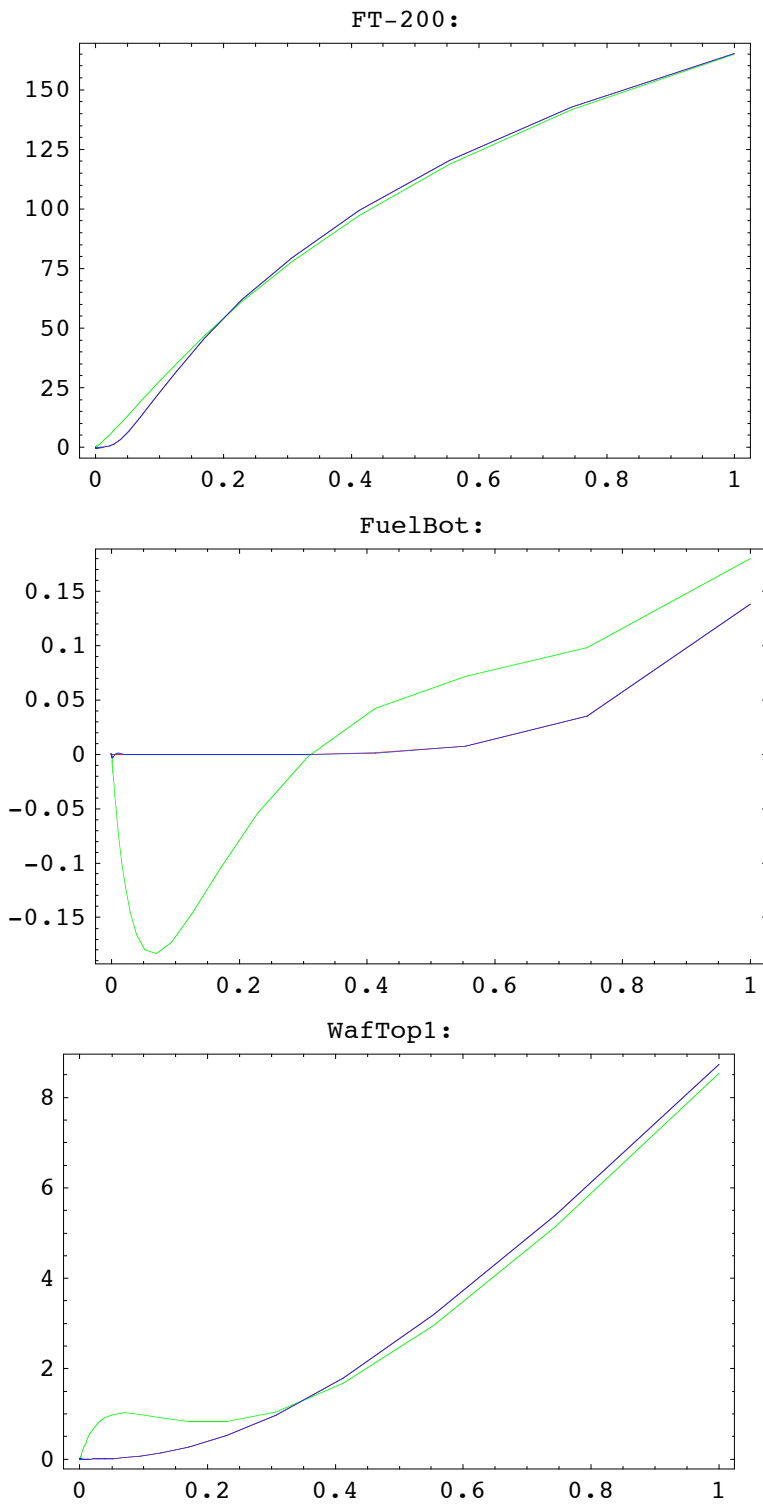
```
In[11]:= res5 = AnsysTransientSolution[XSeries[ansys], TakeSystem[sys, 5]]
```

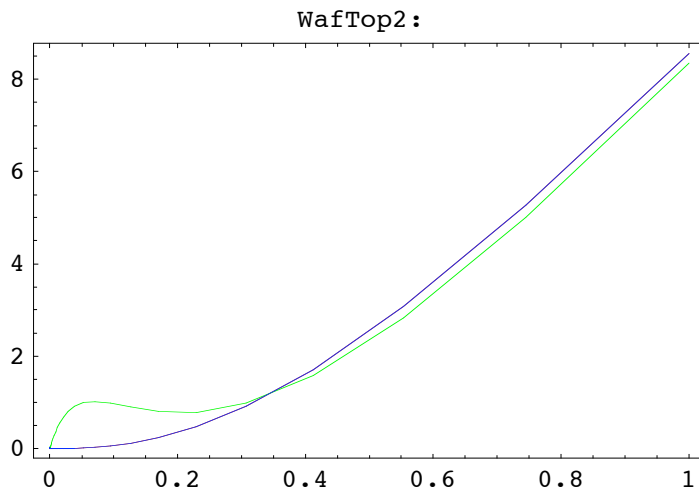
```
Out[11]= - SimulationResult -
```

Make a plot (ansys is red, res5 is green, res25 is blue)

```
In[12]:= PlotResult[{ansys, res5, res25},  
  PlotStyle -> {RGBColor[1, 0, 0], RGBColor[0, 1, 0], RGBColor[0, 0, 1]}];
```

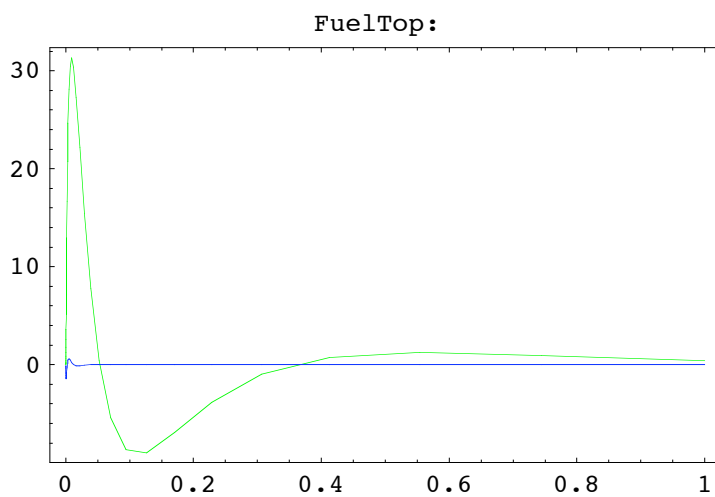
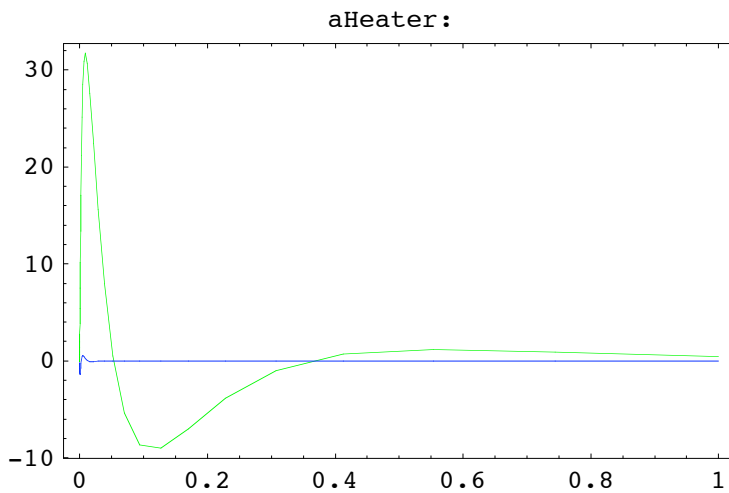


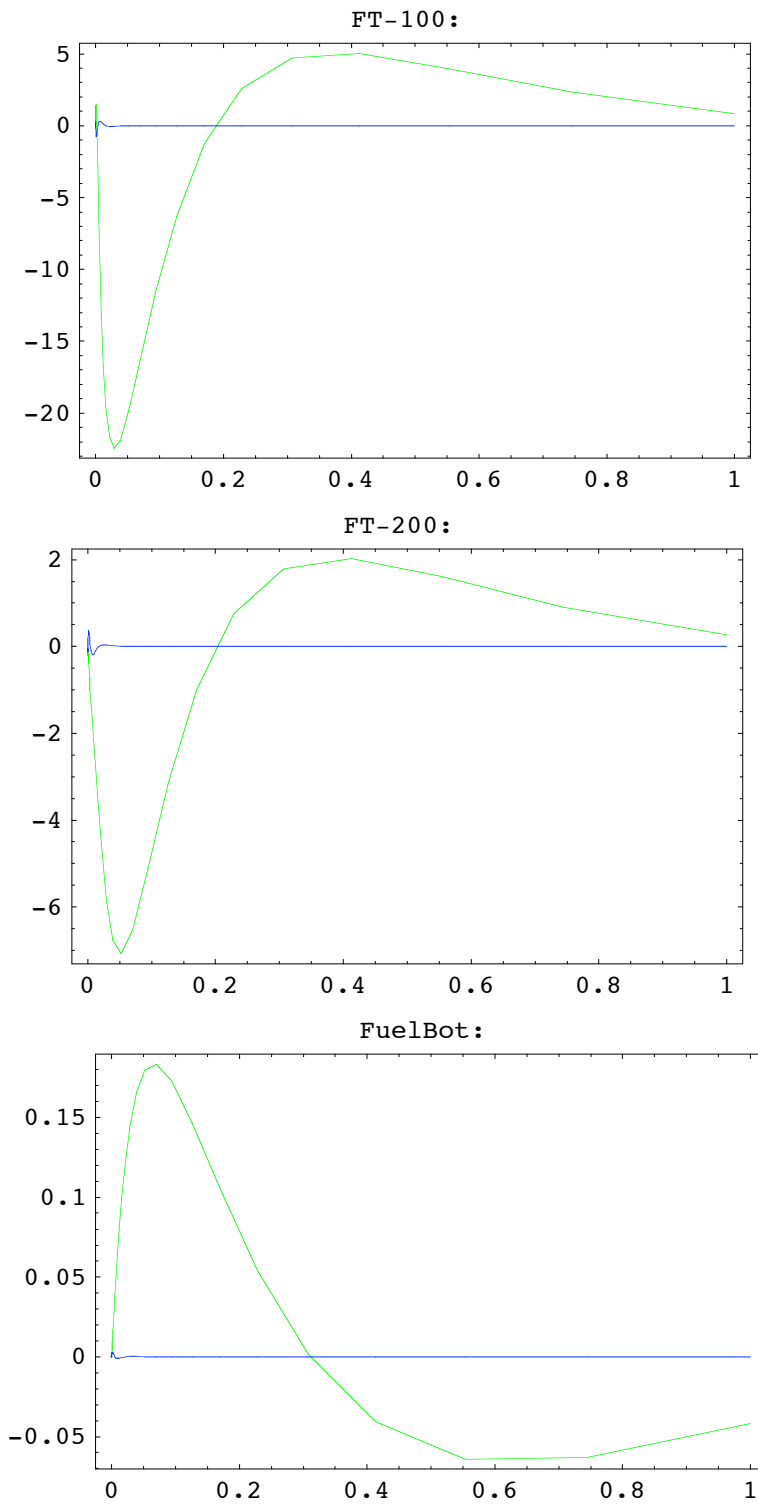


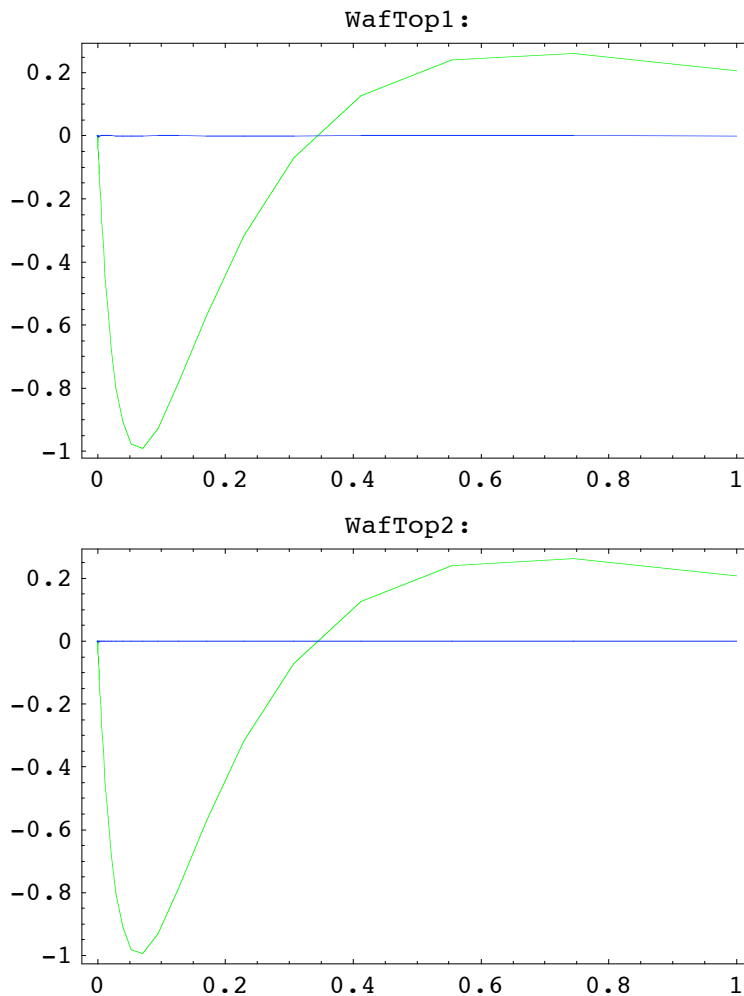


Make a difference plot

```
In[13]:= PlotResult[Difference[ansys, {res5, res25}],  
PlotStyle -> {RGBColor[0, 1, 0], RGBColor[0, 0, 1]}];
```







Discussion

You can see that AnsysTransientSolution is much faster and gives better agreement with ANSYS results. However, remember that results produced by TransientSolution is much accurate! AnsysTransientSolution makes the same integration errors as ANSYS and what you see is cancelation of error. As a result, it makes sense to use TransientSolution to check if the number of integration points chosen in ANSYS was good enough.

■ Local Error Indicator

We can estimate local error and choose an optimal dimension of the reduced system as described in paper Error indicators for fully automatic extraction of heat-transfer macromodels for MEMS.

Compute an error indicator for 1, 10, and 100 Hz.

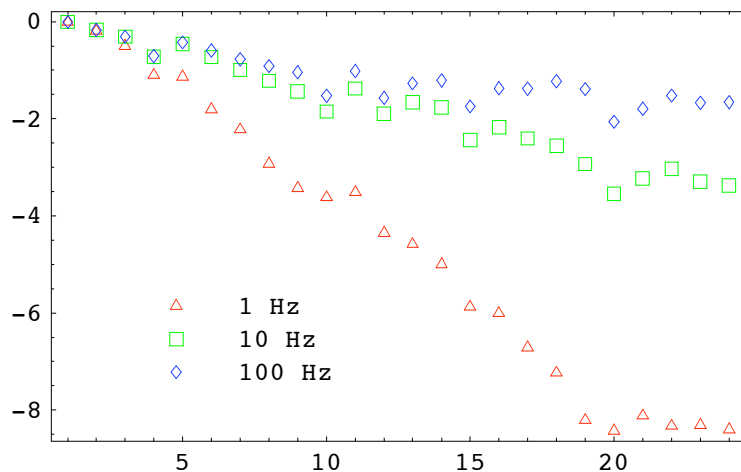
```
In[14]:= er1 = LocalErrorIndicator[FrequencyConvergence[1, sys, 1, 25]];
```

```
In[15]:= er10 = LocalErrorIndicator[FrequencyConvergence[10, sys, 1, 25]];
```

```
In[16]:= er100 = LocalErrorIndicator[FrequencyConvergence[100, sys, 1, 25]];
```

Make a plot Log10[Relative Error] vs. system dimension

```
In[17]:= MultipleListPlot[{er1, er10, er100},  
  SymbolShape -> {PlotSymbol[Triangle, Filled -> False],  
    PlotSymbol[Box, Filled -> False], PlotSymbol[Diamond, Filled -> False]},  
  SymbolStyle -> {RGBColor[1, 0, 0], RGBColor[0, 1, 0], RGBColor[0, 0, 1]},  
  PlotLegend -> {"1 Hz", "10 Hz", "100 Hz"}, LegendBorder -> {},  
  LegendPosition -> {-0.7, -0.4}, LegendSize -> {0.8, 0.28},  
  Evaluate[Imtek`Post4MOR`Private`defaultPlotOptions]];
```



Now you can choose the dimension of the reduced system based on your requirements.