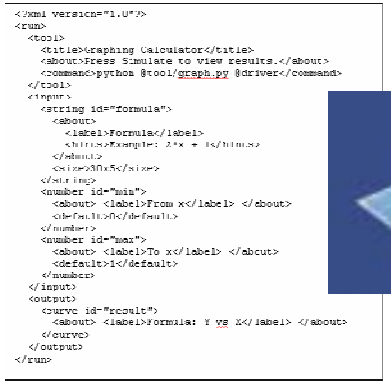
Adding Rappture to MATLAB Applications



Michael McLennan

HUBzero® Platform for Scientific Collaboration

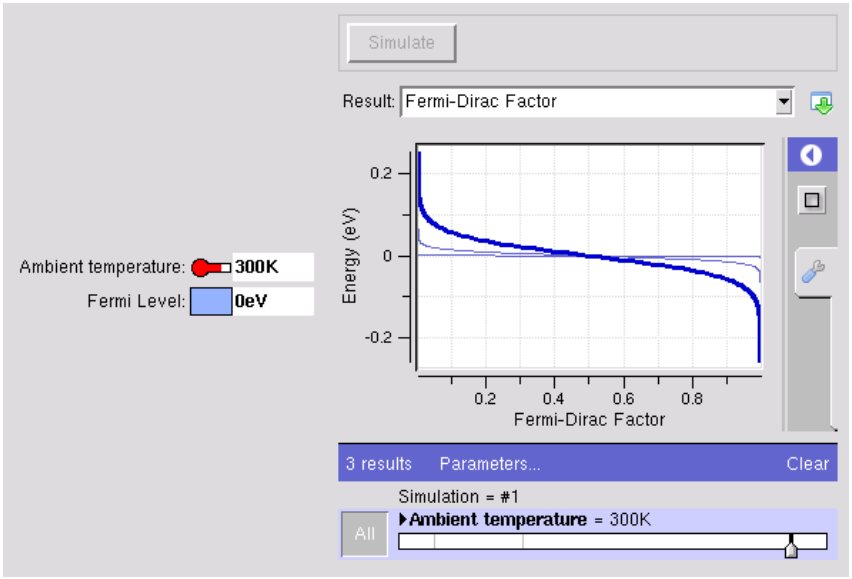
Purdue University

Example: Matlab/Octave Tool

The usual way…

```
% matlab -nodisplay -r fermi
Enter the Fermi level (eV):
 Ef = 2.4
Enter the temperature (K):
 T = 77

% more out.dat
FERMI-DIRAC FUNCTION F1/2

f1/2            Energy (eV)
------------    ------------
    0.999955        2.33365
     0.99995        2.33431
    0.999944        2.33498
```

The Rappture way…



https://nanohub.org/infrastructure/rappture

source code: rappture/examples/app-fermi

What is the interface?

```
disp('Enter the Fermi level (eV):');
Ef = input(' Ef = ');
```

```
disp('Enter the temperature (K):');
T = input(' T = ');
```

physics

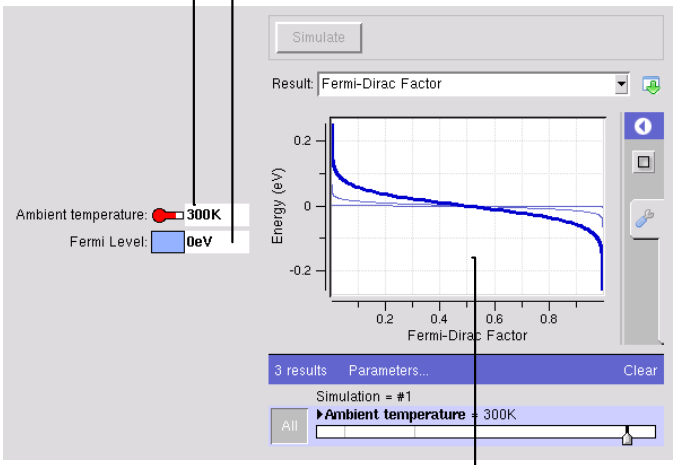
```
kT = 8.61734e-5 * T;
Emin = Ef - 10*kT;
Emax = Ef + 10*kT;

E = linspace(Emin,Emax,200);
f = 1.0 ./ (1.0 + exp((E - Ef)/kT));
```

```
fid = fopen('out.dat','w');
fprintf(fid,'FERMI-DIRAC FUNCTION
F1/2\n\n');
fprintf(fid,'f1/2          Energy (eV)\n');
fprintf(fid,'------------  ------------\n');
fprintf(fid,'%12g  %12g\n', [f; E]);
fclose(fid);
```

```
quit;
```

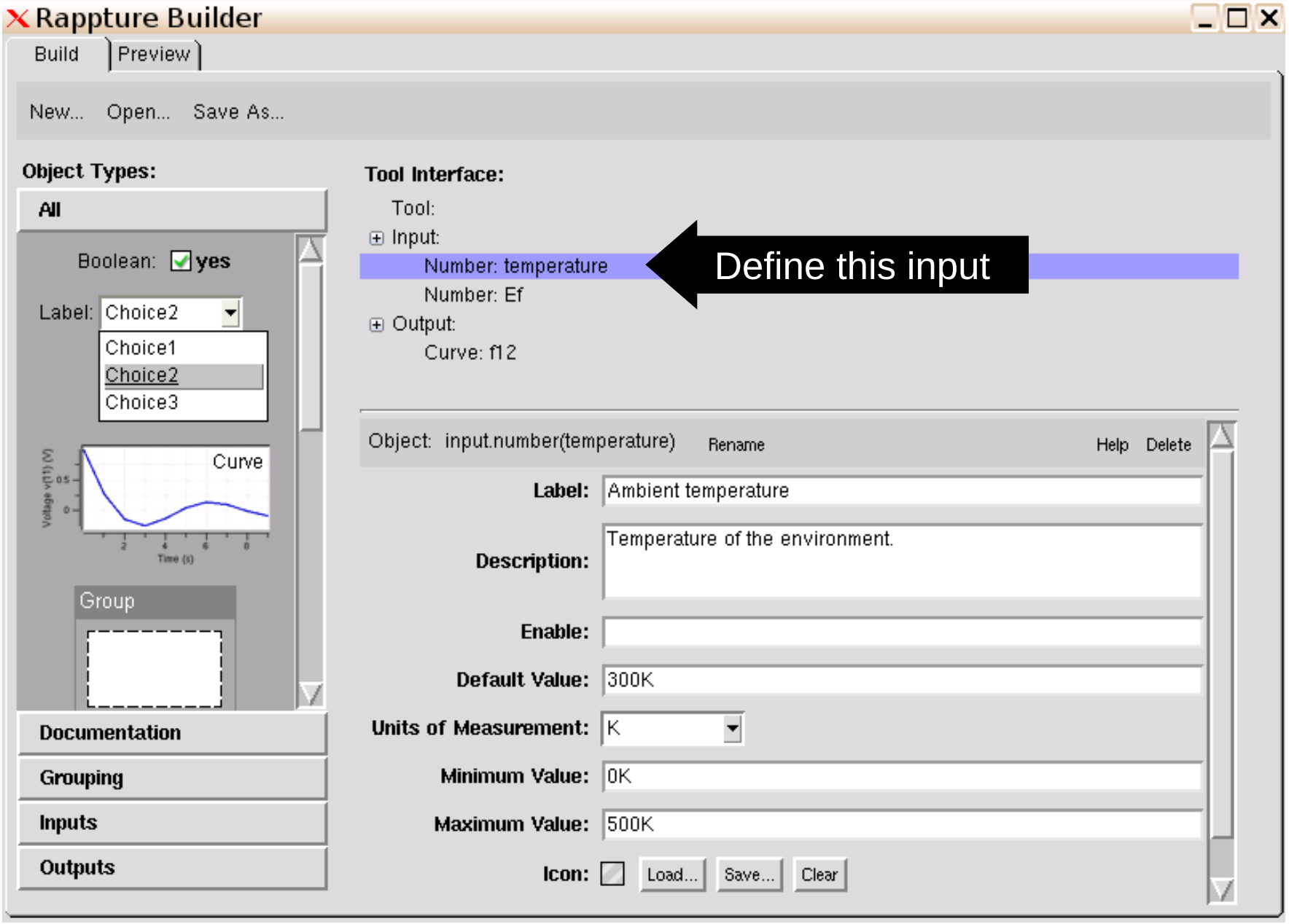
number

number

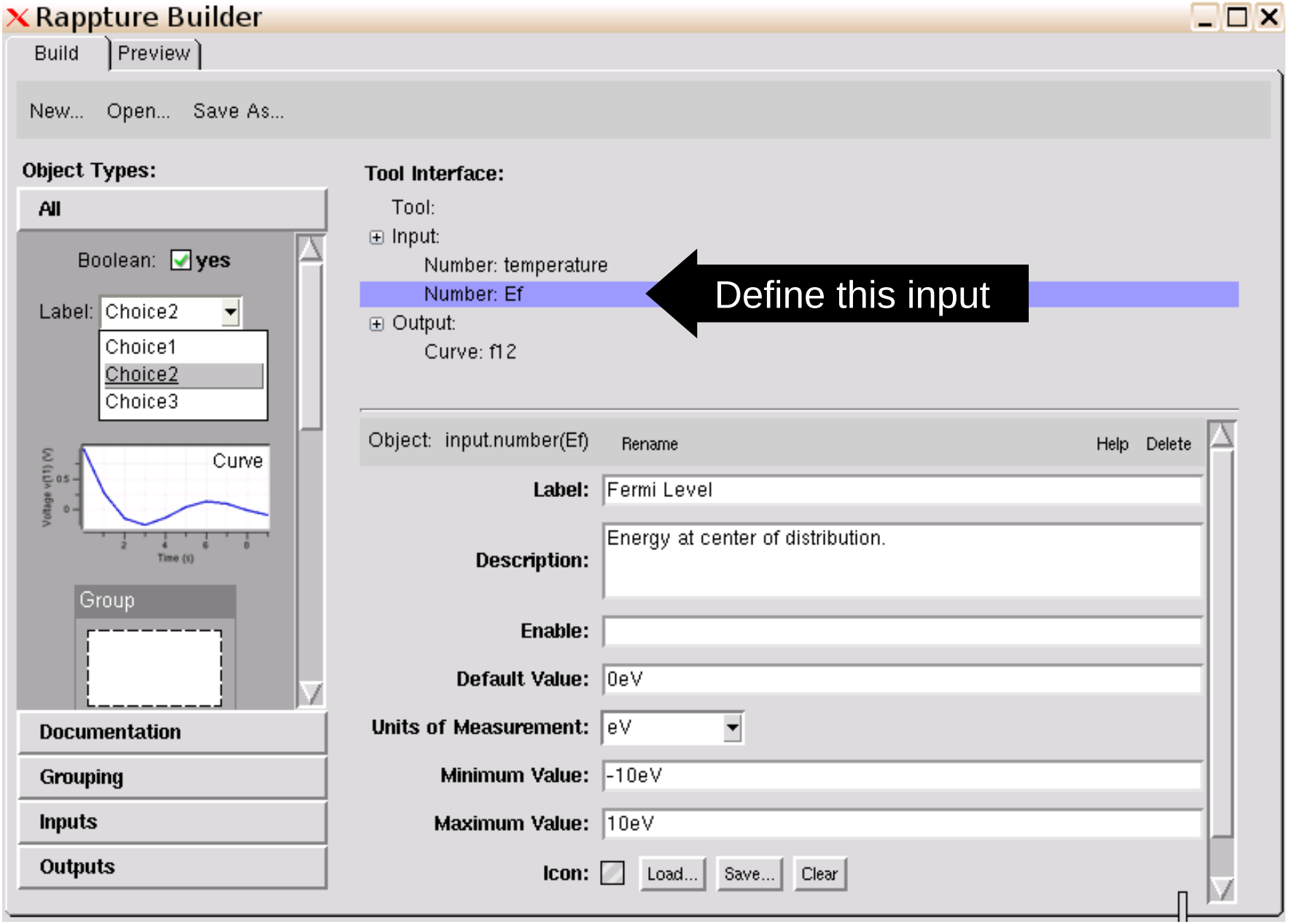


curve

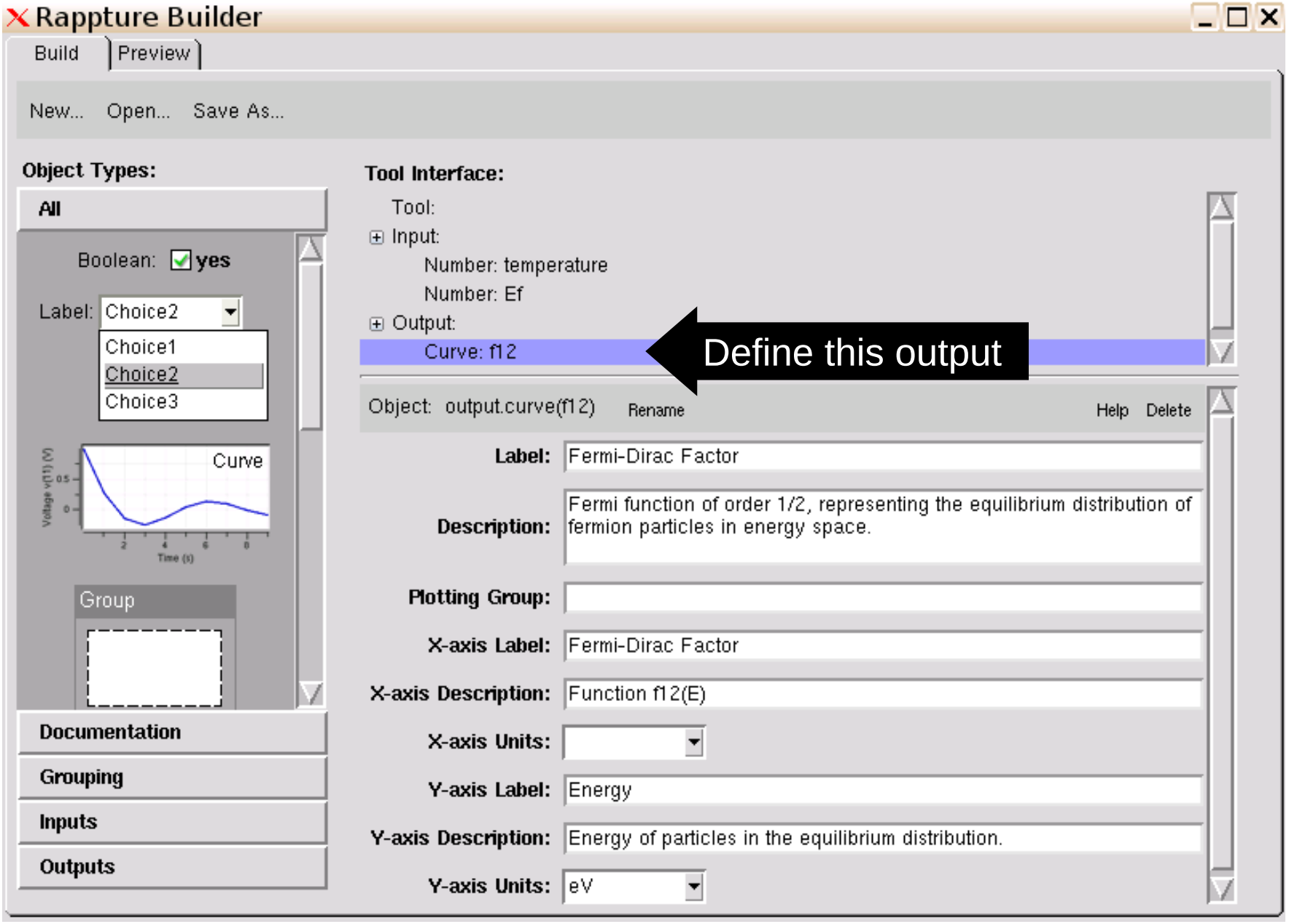
Build the interface: Temperature input



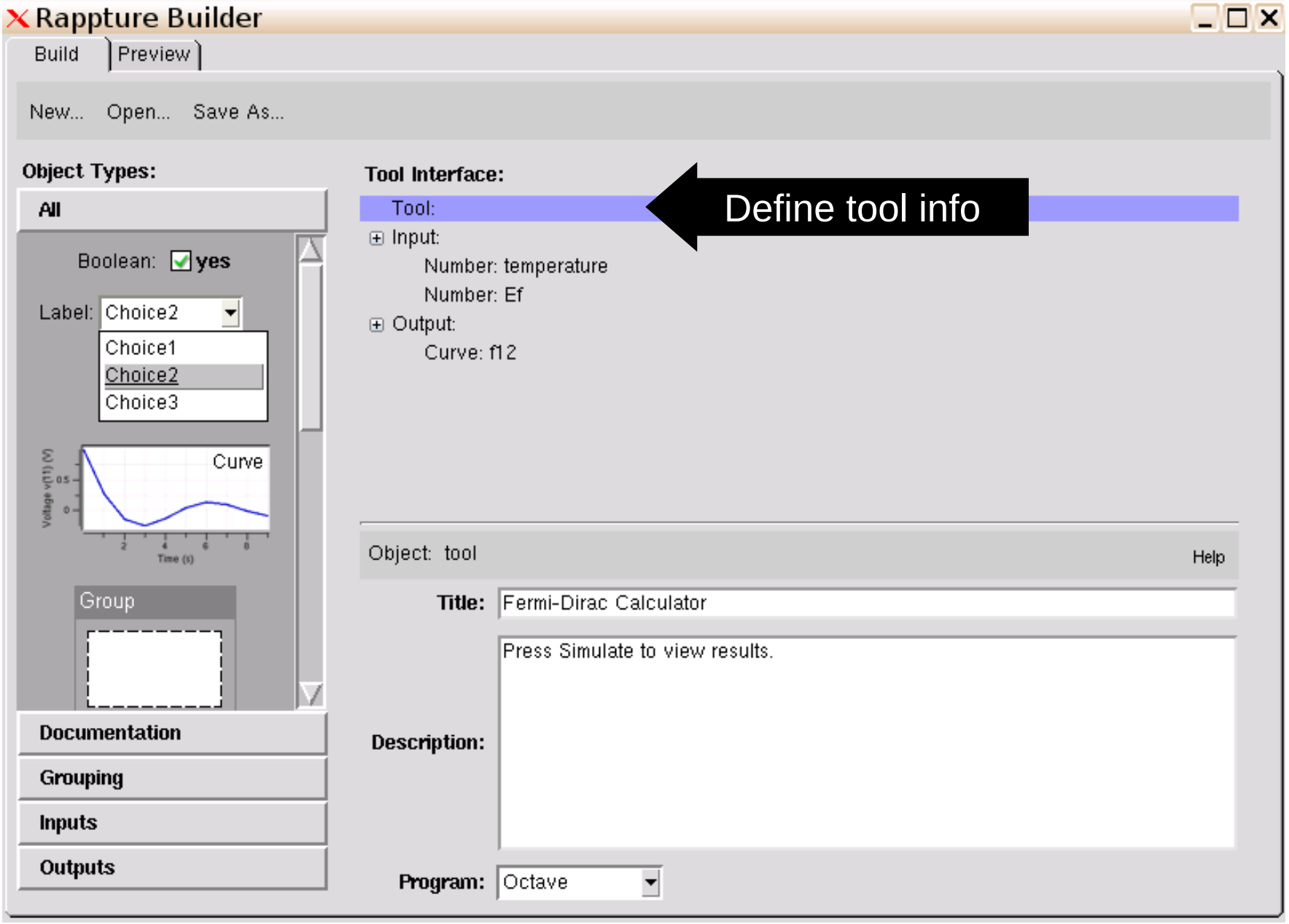
Build the interface: Ef input



Build the interface: Curve output



Build the interface: Use MATLAB/Octave



Add physics to the generated code

Generated script: main.m

```
…
% get input value for input.number(temperature) and convert to K
str = rpLibGetString(io,'input.number(temperature).current');
[temperature,err] = rpUnitsConvertDbl(str, 'K');

% get input value for input.number(Ef) and convert to eV
str = rpLibGetString(io,'input.number(Ef).current');
[Ef,err] = rpUnitsConvertDbl(str, 'eV');

%%%%%%%%%%%%%%%%%%%%%%%%
%  Add your code here f
%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%
% Save output values ba
%%%%%%%%%%%%%%%%%%%%%%%%

% save output value for output.curve(f12)
% this assumes a vector 'x' and a vector 'y'
xydata = [x;y];
str = sprintf('%12g %12g\n', xydata);
rpLibPutString(io,'output.curve(f12).component.xy',str,0);
…
```

physics

```
kT = 8.61734e-5 * T;
Emin = Ef - 10*kT;
Emax = Ef + 10*kT;

E = linspace(Emin,Emax,200);
f = 1.0 ./ (1.0 + exp((E - Ef)/kT));
```

Add physics to the generated code

Final script: main.m

```
…
% get input value for input.number(temperature) and convert to K
str = rpLibGetString(io,'input.number(temperature).current');
[temperature,err] = rpUnitsConvertDbl(str, 'K');

% get input value for input.number(Ef) and convert to eV
str = rpLibGetString(io,'input.number(Ef).current');
[Ef,err] = rpUnitsConvertDbl(str, 'eV');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
kT = 8.61734e-5 * temperature;
Emin = Ef - 10*kT;
Emax = Ef + 10*kT;

E = linspace(Emin,Emax,200);
f = 1.0 ./ (1.0 + exp((E - Ef)/kT));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% save output value for output.curve(f12)
% this assumes a vector 'x' and a vector 'y'
xydata = [f;E];
str = sprintf('%12g %12g\n', xydata);
rpLibPutString(io,'output.curve(f12).component.xy',str,0);
…
```

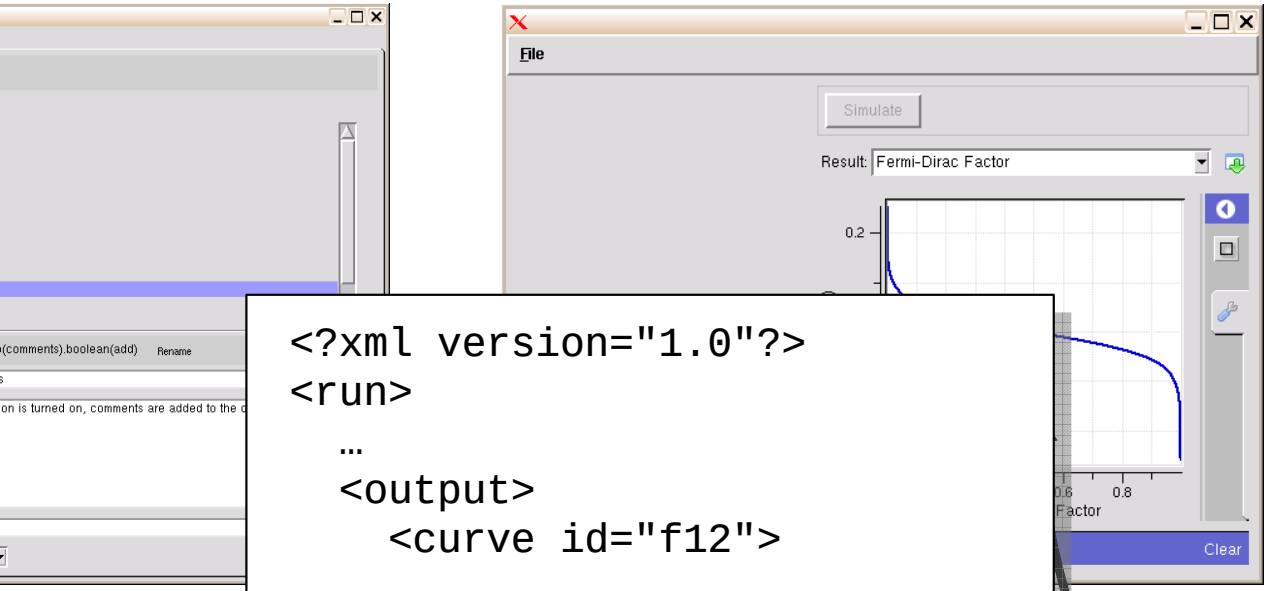
Running the code

Color xterm

```
$ rappture
```

Builder

Runner



<XML> driver1827.xml

MATLAB®

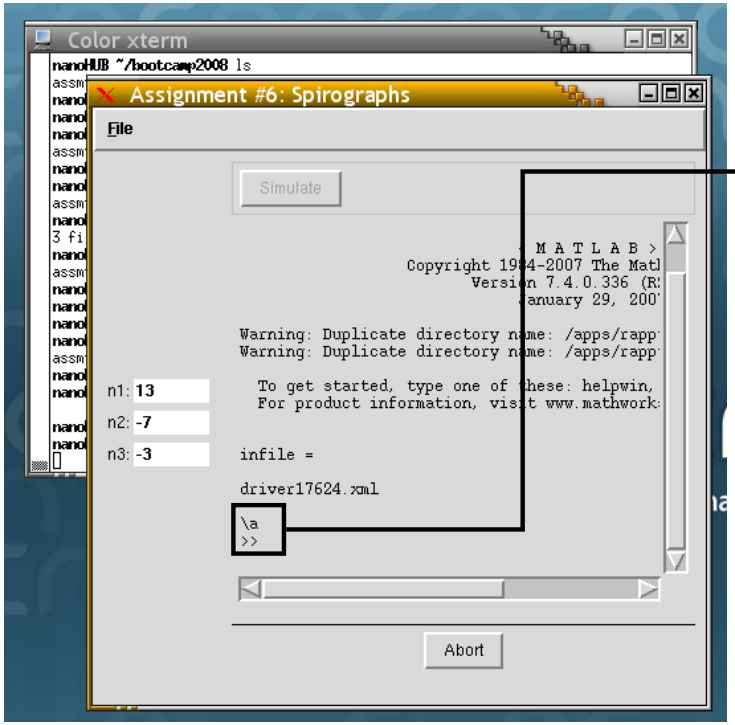
```
kT = 8.61734e-5 * temperature
Emin = Ef - 10*kT;
Emax = Ef + 10*kT;

E = linspace(Emin,Emax,200);
f = 1.0 ./ (1.0 + exp((E - Ef)/kT));
```

physics

```
<?xml version="1.0"?>
<run>
  …
  <output>
    <curve id="f12">
      …
      <component>
        <xy>
          0.999955  2.33365
          0.99995   2.33431
          0.999944  2.33498
          …
  </output>
</run>
```

<XML> run12703129102.xml

If something goes wrong, MATLAB goes into “debug” mode:




```
\a
>>
```

Waiting for you to type a MATLAB command

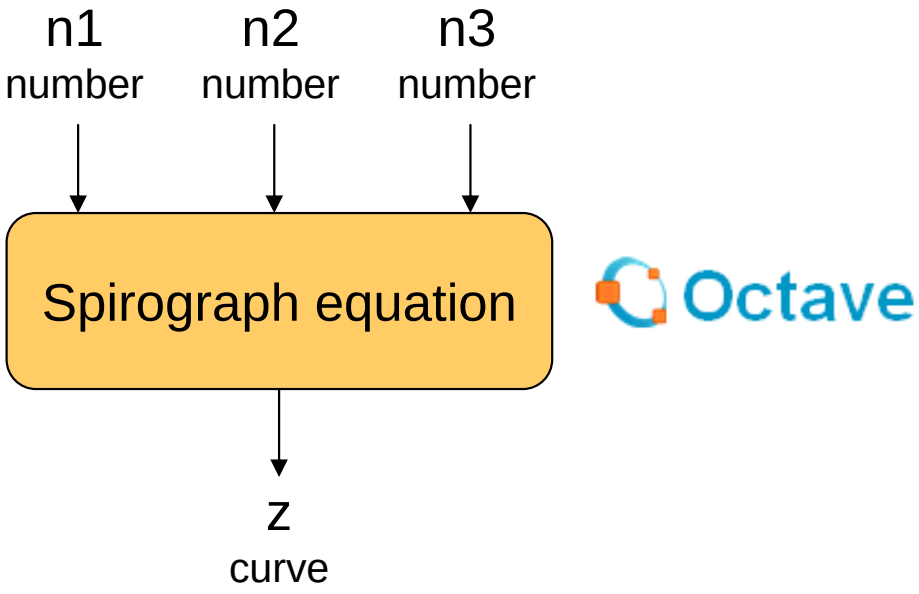
Click *Abort* instead

```
\a>>
??? Error using ==> mtimes
Inner matrix dimensions must agree.

Error in ==> spirograph at 15
z = exp(i*2*pi*n1*t) + exp(i*2*pi*n2*t) + exp
```

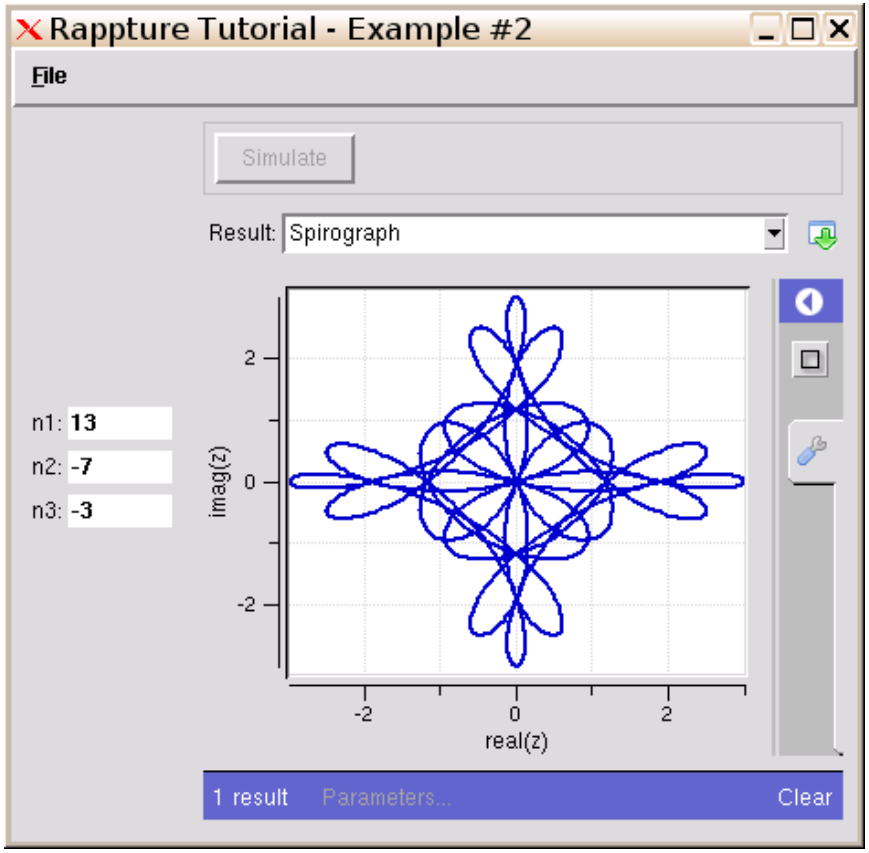
Run it by hand:

```
$ ls
driver1529.xml  fermi.m  tool.xml
$ use rappture
$ matlab -r infile=\'driver1529.xml\',main
```

Assignment #4: Build a simple Spirograph tool



Octave



In MATLAB/Octave:

```
t = linspace(0, 1, 1000);
z = exp(i*2*pi*n1*t) + exp(i*2*pi*n2*t) + exp(i*2*pi*n3*t);
plot(real(z),imag(z));
```