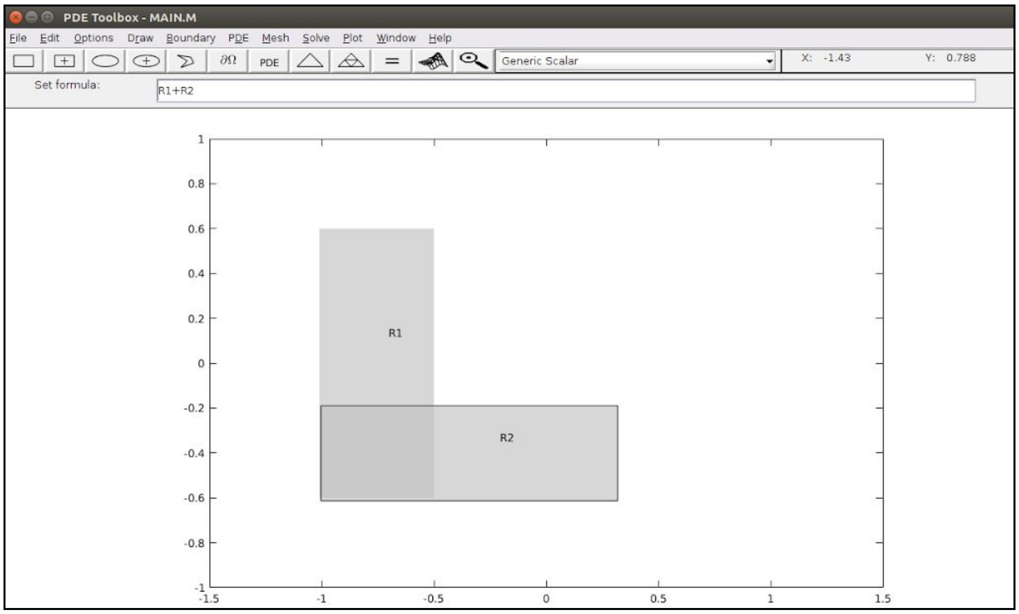
Seth Goodman 2018-02-15

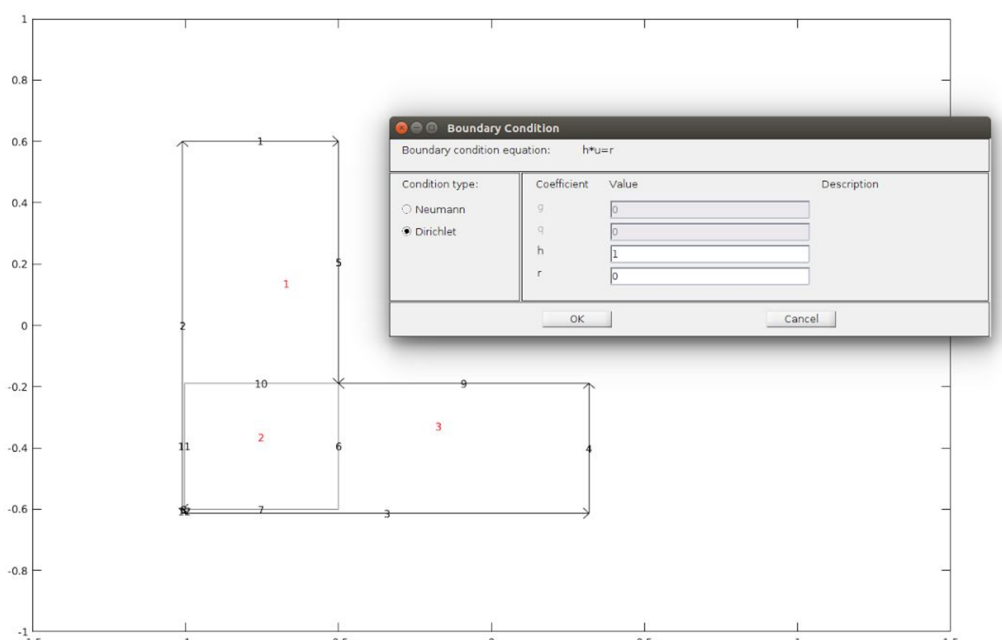
APSC 608 - Project 01

This project will explore the basic functionality of MATLAB's *pdetool* by solving the heat equation on an L-shaped domain (composed of two rectangles).

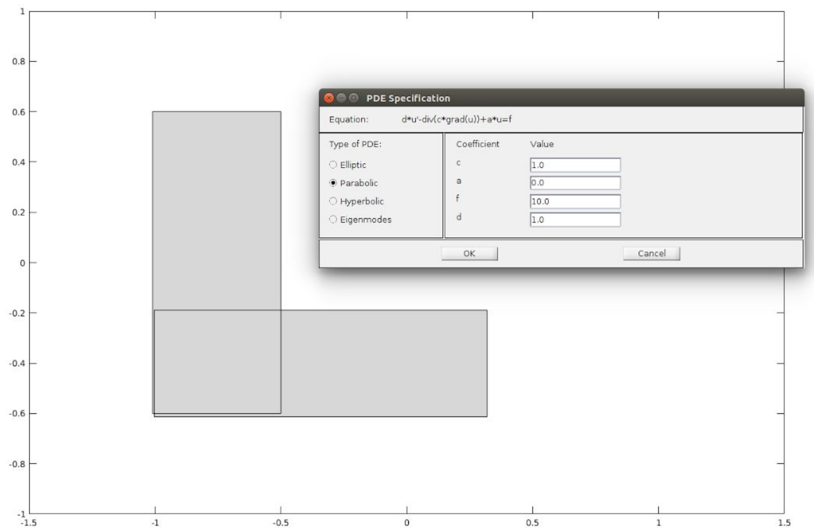
The first step is to draw the L-shaped region by intersecting two rectangles, R1 and R2.



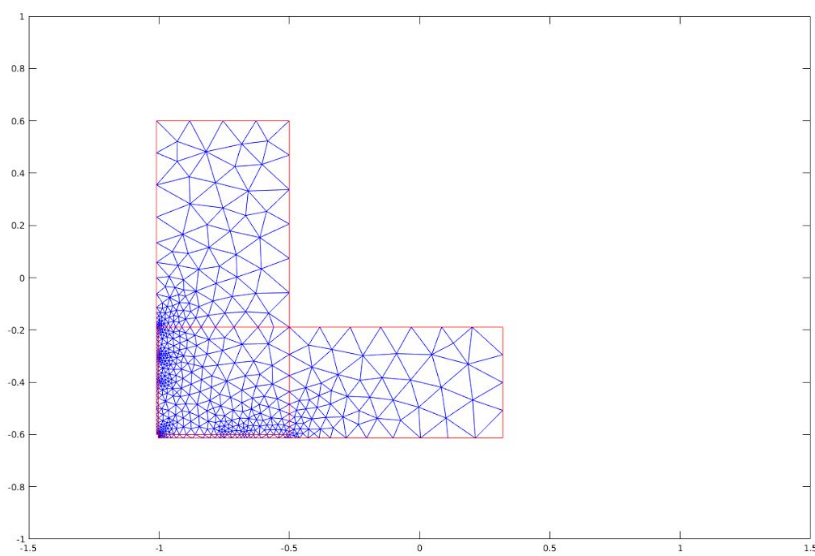
In boundary mode, edge and subdomain labels can be enabled, and boundary conditions can be defined.



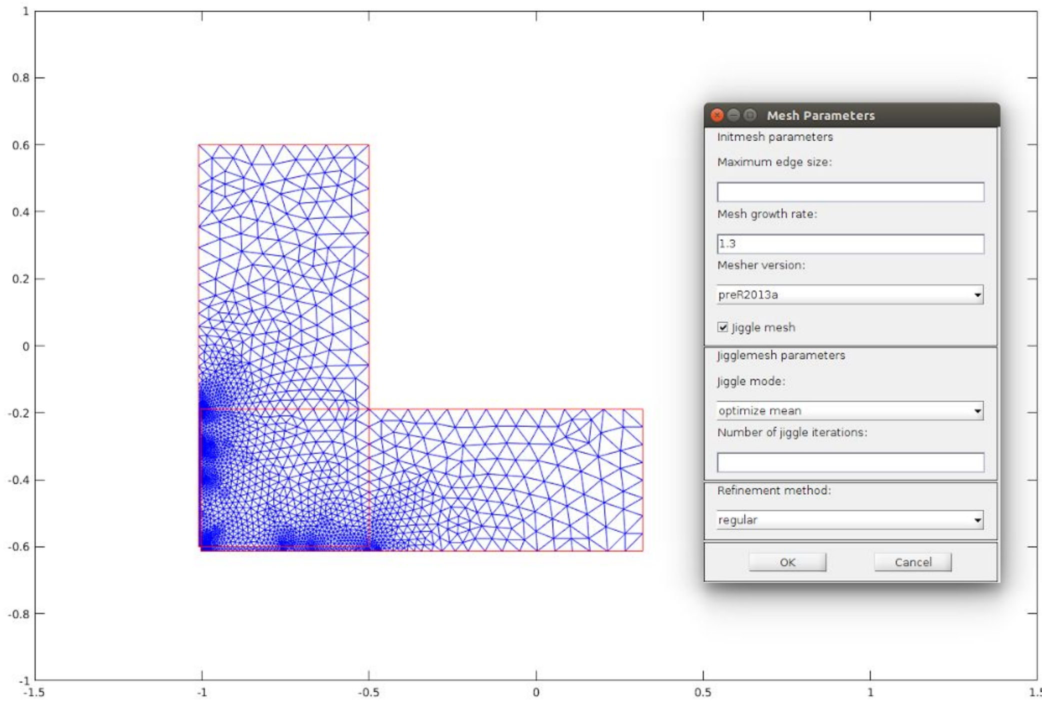
In PDE mode, the PDE specifications and coefficients can be assigned. The type of PDE should be set to “Parabolic” since this project is examining the heat equation.



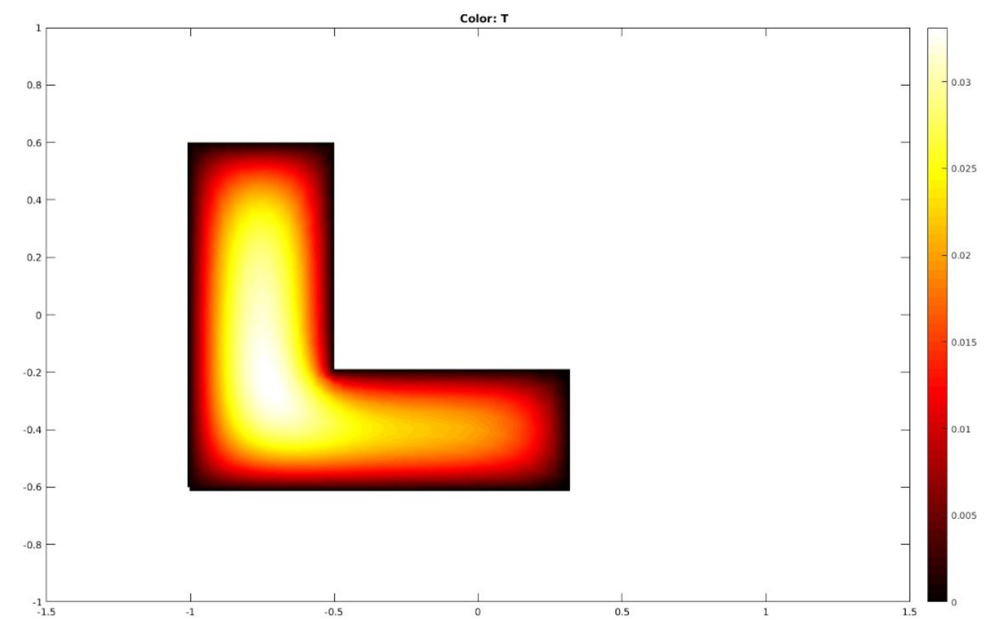
The initial mesh for the region can then be generated, and consists of 591 nodes and 1065 triangles.



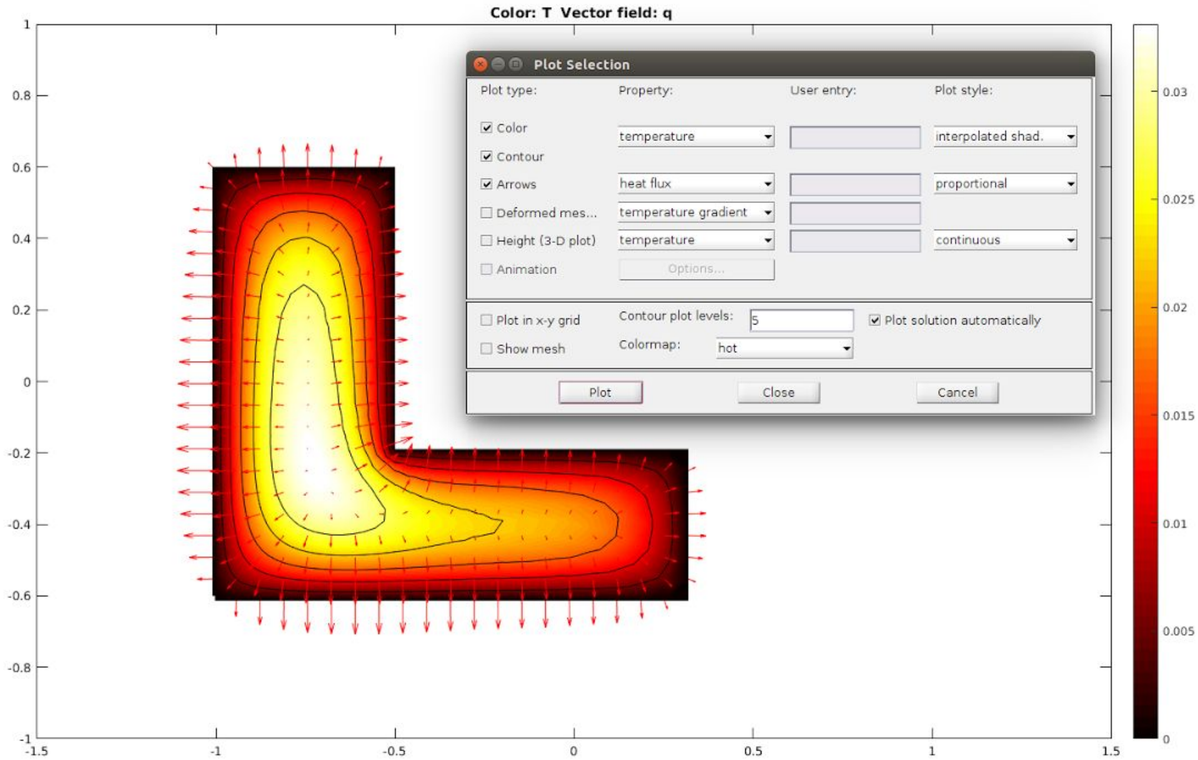
This mesh can be refined and additional parameters modified if needed. The refined mesh contains 2246 nodes and 4260 triangles.



With the region defined, boundary conditions set, PDE parameters defined, and mesh generated, the PDE can be solved and plotted. Make sure the Application Mode drop down has “Heat Transfer” (or for general diffusion processes, including the heat equation, “Diffusion” can be used since heat transfer is a diffusion process).



The plot can be adjusted with a variety of display settings, including contours and arrows indicating patterns of diffusion.



These plots visualize the transfer of heat from the center of the L-shaped domain towards the boundaries, based on the boundary conditions and PDE parameters specified.

References:

https://www.mathworks.com/help/pde/

https://edoras.sdsu.edu/doc/matlab/toolbox/pde/2examp20.html#8855

Appendix:

MATLAB m-file exported from pdetool

```
% This script is written and read by pdetool and should NOT be edited.
% There are two recommended alternatives:
 % 1) Export the required variables from pdetool and create a MATLAB script
 %    to perform operations on these.
 % 2) Define the problem completely using a MATLAB script. See
 %    http://www.mathworks.com/help/pde/examples/index.html for examples
 %    of this approach.
function pdemodel
[pde_fig,ax]=pdeinit;
pdetool('appl_cb',9);
set(ax,'DataAspectRatio',[1 1 1]);
set(ax,'PlotBoxAspectRatio',[1.5 1 1]);
set(ax,'XLim',[-1.5 1.5]);
set(ax,'YLim',[-1 1]);
set(ax,'XTickMode','auto');
set(ax,'YTickMode','auto');

% Geometry description:
pderect([-1.0098684210526316 -0.5 0.6003289473684208
-0.60032894736842146],'R1');
pderect([-1.0032894736842106 0.31907894736842124 -0.61348684210526372
-0.18914473684210553],'R2');
set(findobj(get(pde_fig,'Children'),'Tag','PDEEval'),'String','R1+R2')

% Boundary conditions:
pdetool('changemode',0)
pdesetbd(12,...
'dir',...
1,...
'1',...
'0')
pdesetbd(9,...
'dir',...
1,...
'1',...
'0')
pdesetbd(8,...
'dir',...
1,...
'1',...
'0')
pdesetbd(5,...
'dir',...
```

```
1,...
'1',...
'0')
pdesetbd(4,...
'dir',...
1,...
'1',...
'0')
pdesetbd(3,...
'dir',...
1,...
'1',...
'0')
pdesetbd(2,...
'dir',...
1,...
'1',...
'0')
pdesetbd(1,...
'dir',...
1,...
'1',...
'0')

% Mesh generation:
setappdata(pde_fig,'Hgrad',1.3);
setappdata(pde_fig,'refinemethod','regular');
setappdata(pde_fig,'jiggle',char('on','mean',''));
setappdata(pde_fig,'MesherVersion','preR2013a');
pdetool('initmesh')
pdetool('refine')

% PDE coefficients:
pdeseteq(1,...
'1.0',...
'1.0',...
'(1.0)+(1.0).*(0.0)',...
'(1.0).*(1.0)',...
'0:10',...
'0.0',...
'0.0',...
'[0 100]')
setappdata(pde_fig,'currparam',...
['1.0';...
'1.0';...
'1.0';...
'1.0';...
```

```
'1.0';...
'0.0'])

% Solve parameters:
setappdata(pde_fig,'solveparam',...
char('0','6390','10','pdeadworst',...
'0.5','longest','0','1E-4','','fixed','Inf'))

% Plotflags and user data strings:
setappdata(pde_fig,'plotflags',[1 1 1 1 2 1 6 1 0 0 0 1 1 0 0 0 0 1]);
setappdata(pde_fig,'colstring','');
setappdata(pde_fig,'arrowstring','');
setappdata(pde_fig,'deformstring','');
setappdata(pde_fig,'heightstring','');

% Solve PDE:
pdetool('solve')
```