Table of Contents

Embarrassing Parallel GPU Greens Function Linear Super Position 1
Simulate disjoint material/tissue types 1
Query the device 2
Setup Material Parameters 2
initialize data arrays 3
Compile and setup thread grid 3
Run on GPU 3
transfer device to host 3
plot temperature 3
global search and plot exhaustive search 4

Embarrassing Parallel GPU Greens Function Linear Super Position

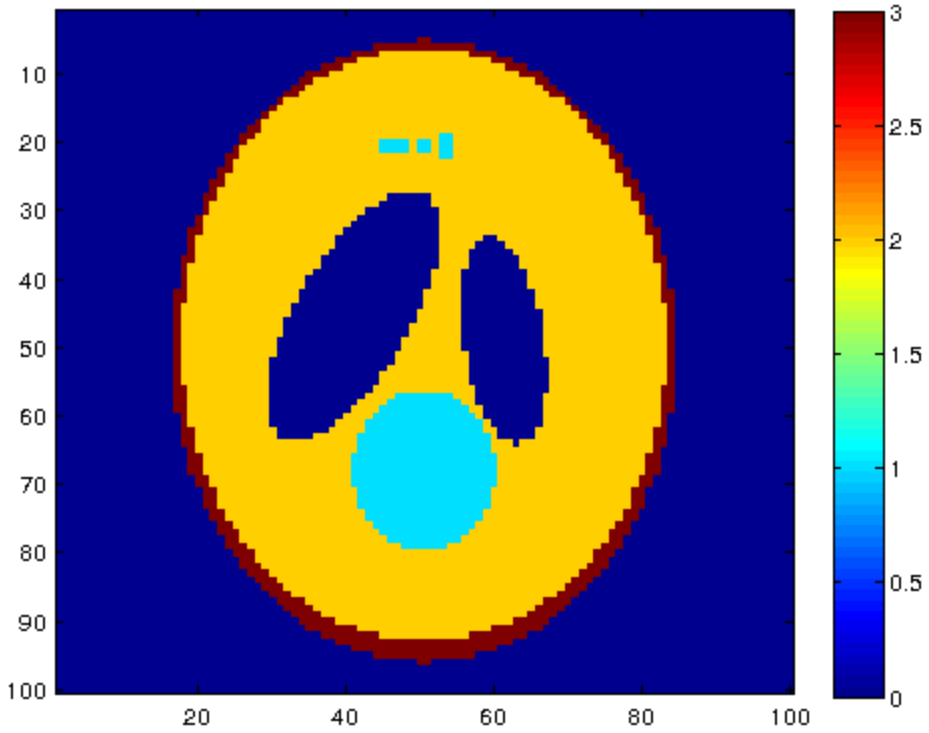
```
clear all
close all
format shortg
```

Simulate disjoint material/tissue types

create npixel^3 image

```
npixel   = 100;
materialID = int32(10*phantom3d('Modified Shepp-Logan',npixel));
materialID(materialID == 3  ) = 1;
materialID(materialID == 10 ) = 3;
handle1 = figure(1)
imagesc(materialID(:,:,npixel/2),[0 3])
colorbar
```

```
handle1 =

     1
```



Query the device

GPU must be reset on out of bounds errors reset(gpuDevice(1))

```
deviceInfo = gpuDevice(1);
numSMs = deviceInfo.MultiprocessorCount;

spacingX = 1.0e-3;
spacingY = 1.0e-3;
spacingZ = 1.0e-3;
```

Setup Material Parameters

```
ntissue = 4;
perfusion  = [5.e01 , 4.e01 , 3.e01, 6.e01];
conduction = [5.e-1 , 4.e-1 , 3.e-1, 6.e-1];
mueff      = [5.e02 , 4.e02 , 3.e02, 6.e02];
nsource    = 10;
xloc       = npixel/2*spacingX+spacingX*linspace(1,nsource ,nsource )+1.e-4;
yloc       = npixel/2*spacingY+spacingY*linspace(1,nsource ,nsource )+1.e-4;
zloc       = npixel/2*spacingZ+spacingZ*linspace(1,nsource ,nsource )+1.e-4;
u_artery   = 37.;
c_blood    = 3480.;
power      = 10.;
R1 = .001 ; % 1mm
```

```
R2 = .1   ; % 100mm
```

initialize data arrays

initialize on host and perform ONE transfer from host to device

```
h_temperature     = zeros(npixel,npixel,npixel);
d_temperature  = gpuArray( h_temperature  );
```

Compile and setup thread grid

grid stride loop design pattern, 1-d grid http://devblogs.nvidia.com/parallelforall/cuda-pro-tip-write-flexible-kernels-grid-stride-loops/

```
ssptx = parallel.gpu.CUDAKernel('steadyStatePennesLaser.ptx', 'steadyStatePennesLa
threadsPerBlock = 256;
ssptx.ThreadBlockSize=[threadsPerBlock  1];
ssptx.GridSize=[numSMs*32               1];
```

Run on GPU

```
[d_temperature ] = feval(ssptx,ntissue,materialID,perfusion,conduction, mueff, R1,
```

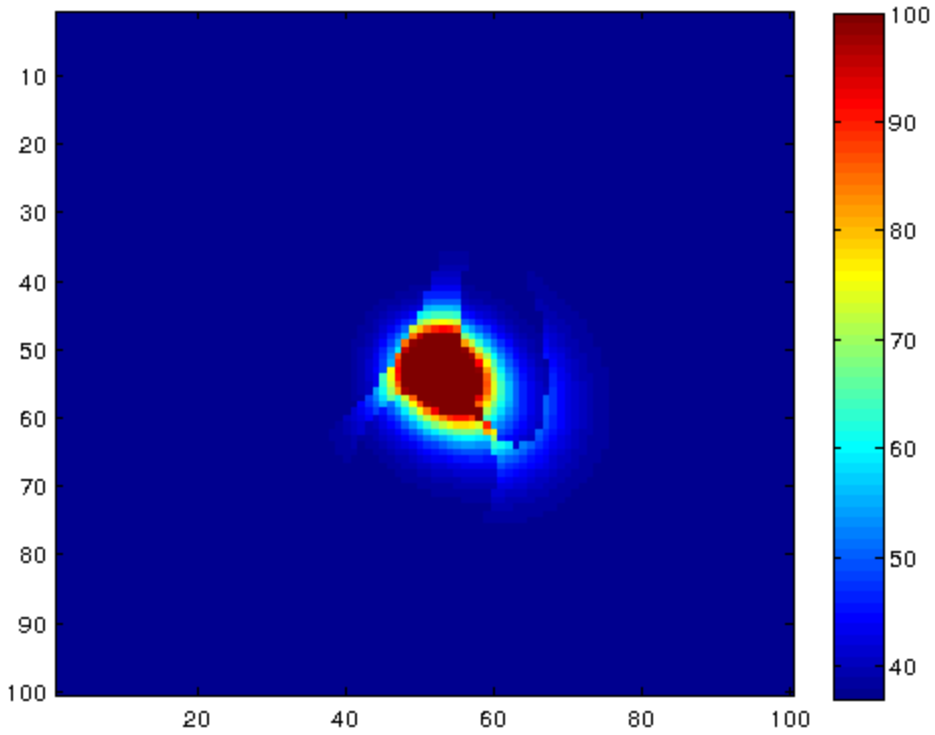
transfer device to host

```
h_temperature  = gather( d_temperature  );
```

plot temperature

```
handle2 = figure(2)
imagesc(h_temperature(:,:,50), [37 100]);
colormap default
colorbar
```

```
handle2 =

     2
```

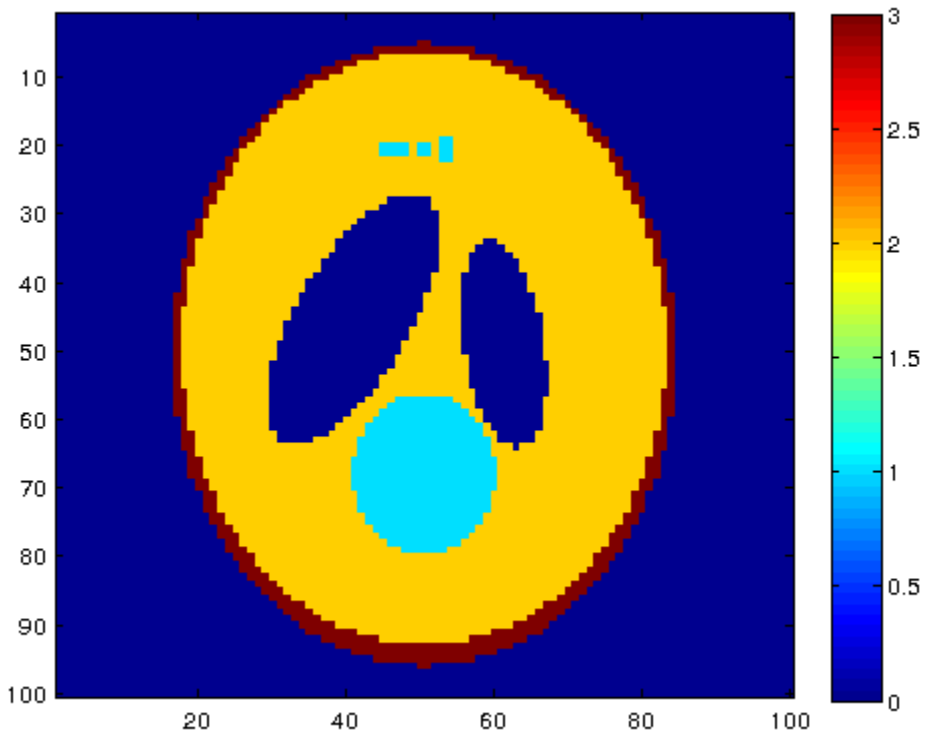


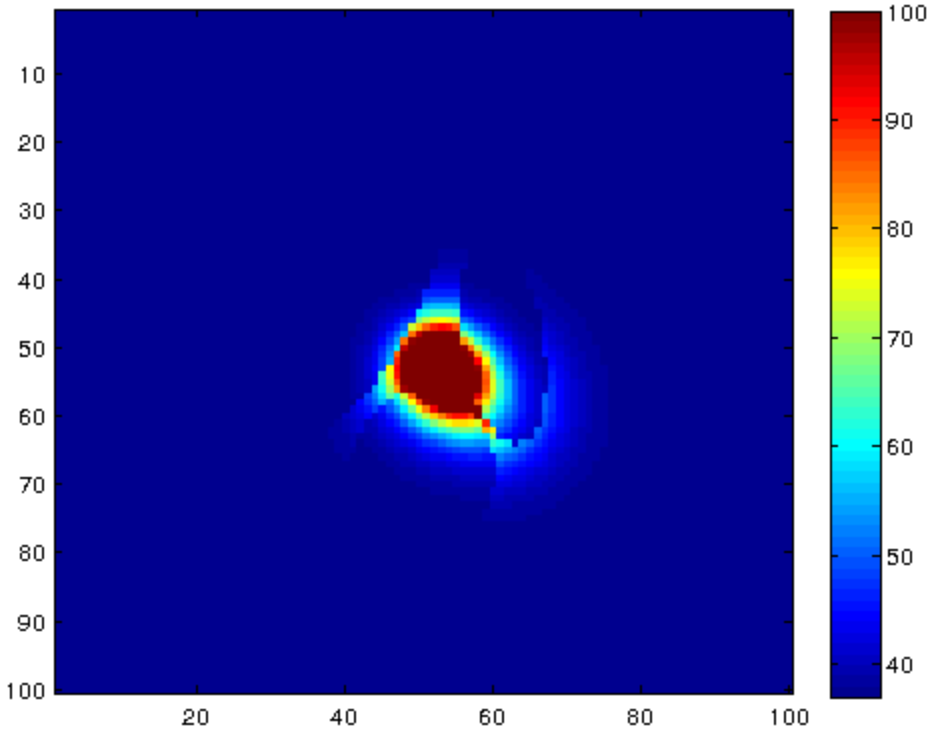
global search and plot exhaustive search

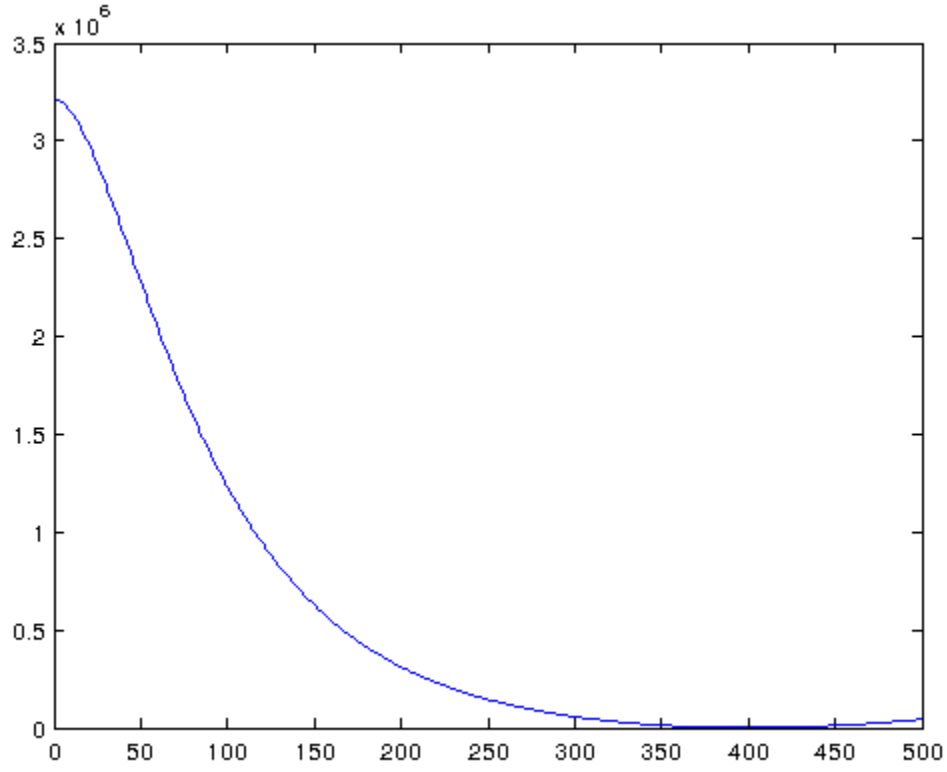
```
tic
sizesearch = 500;
objective =zeros(sizesearch,1);
for iii = 1:sizesearch
 if mod(iii,100 )==0
   disp(sprintf('iter %d',iii));
 end
 mueff(2) = 1. *iii;
 [p_temperature ] = feval(ssptx,ntissue,materialID,perfusion,conduction, mueff, R1
 objective(iii) = gather(sum((p_temperature(:) - d_temperature(:)).^2));
end
toc
handle3 = figure(3)
plot(objective)

saveas(handle1,'material','png')
saveas(handle2,'temperature','png')
saveas(handle3,'exhaustivesearch','png')
```

```
iter 100
iter 200
iter 300
iter 400
iter 500
```

```
Elapsed time is 204.271104 seconds.

handle3 =

     3
```




10
20
30
40
50
60
70
80
90
100
20
40
60
80
100
100
90
80
70
60
50
40


x 10^6
3.5
3
2.5
2
1.5
1
0.5
0
0
50
100
150
200
250
300
350
400
450
500

Published with MATLAB® R2013a