## Table of Contents



# Lab10Simons.m

Madilyn Simons

```
clear; clc;
```

# Problem 1

Find the inverse of a randomly generated matrix A

```
% n is a random integer between 4 and 6 (inclusive)
n = randi([4 6]);

% A is an n x n matrix of random numbers between 0 and 1 (exclusive)
A = rand(n);

% If the determinant of A isn't 0, then A is nonsingular
if det(A) ~= 0

    % create an augmented matrix represent [A | I]
    aug_A = horzcat(A, eye(n));

    % find the reduced row echelon form of [A | I]
    % the will represent [ I | A^-1 ]
    rref_aug_A = rref(aug_A);

    % extract A^-1 from [ I | A^-1 ]
    inv_A = rref_aug_A(1:n, n+1:2*n);

    % print the results
    fprintf('A\n');
    disp(A);

    fprintf('Inverse of A\n');
    disp(inv_A);

else
    % print the results
    fprintf('A is singular\n');
```

```
end
```

```
A
    0.4984    0.2238    0.6991    0.1386
    0.9597    0.7513    0.8909    0.1493
    0.3404    0.2551    0.9593    0.2575
    0.5853    0.5060    0.5472    0.8407

Inverse of A
    5.0123   -0.5663   -3.2870    0.2809
   -6.2026    2.4832    2.2805   -0.1167
   -0.2356   -0.1988    1.6437   -0.4293
    0.3969   -0.9708   -0.1541    1.3436
```

# Problem 2

Use Dootlittle factorization to solve the system Ax = b

```
% Given: A and b
A = [3 -6 9  3 ; ...
     2  1 4  1 ; ...
     1 -2 2 -1 ; ...
     1 -2 3  0 ];

 b = [1 ; 2 ; 3 ; 4];

 % Find the Doolittle factorization for A
 [L, U, P] = lu(A);

 % Use the Doolittle factorization of A to solve Ax = b
 y = front_substition(L, b, 4);
 x = back_substition(U, y, 4);

 % print the results
 fprintf('x = \n');
 disp(x);
```

```
x =
   -7.2000
    1.4000
    4.6667
   -3.6667
```

# Problem 3

Use Dootlittle factorization to solve the system Ax = b

```
% Given: A and b
A = [1  1 -1 0;...
     1  1  4 3;...
```

```
    2 -1  2 4;...
    2 -1  2 3];

b = [1 ; 2 ; 3 ; 4];

% Find the Doolittle facorization for A
[L, U, P] = lu(A);

% Use the Doolittle facorization of A to solve Ax = b
z = P*b;
y = front_substition(L, z, 4);
x = back_substition(U, y, 4);

% print the results
fprintf('x = \n');
disp(x);
```

```
x =
    2.4000
   -0.6000
    0.8000
   -1.0000
```

# Problem 4

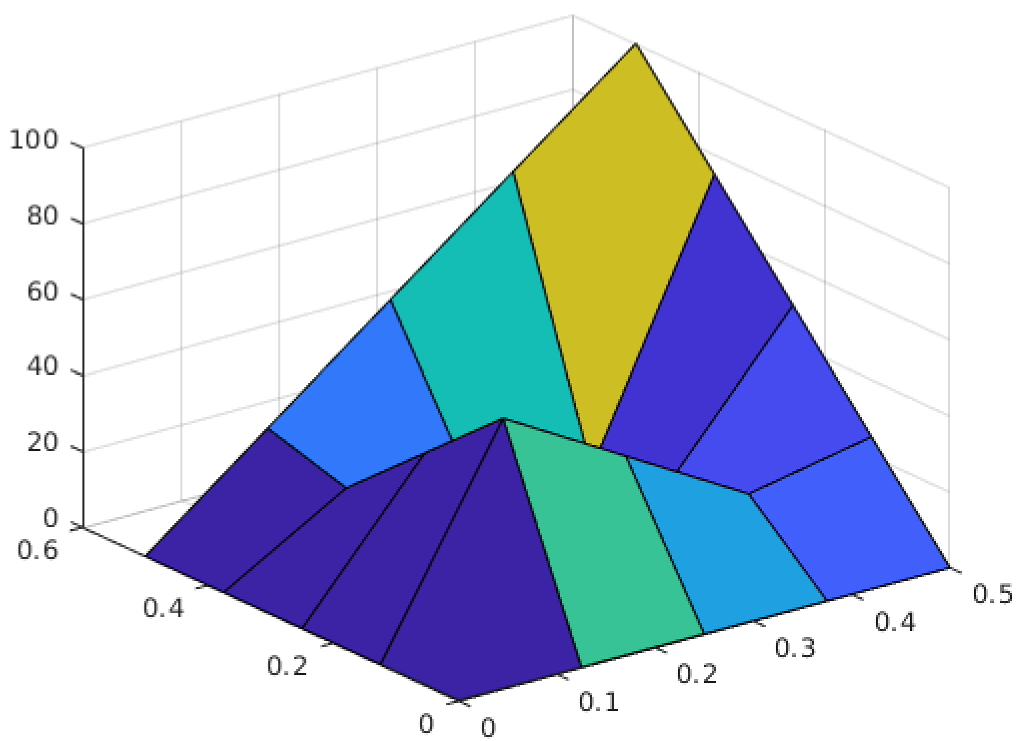
Find the steady-state heat distribution in a thin sqare metal

# Part a

Find the steady-state heat distribution in a thin sqare metal using n = 4 subintervals

```
% number of subintervals
n = 4;

% generate coefficient matrix
A = toeplitz([4 -1 zeros(1, n-3) -1 zeros(1, (n-2)*(n-1)-1)]);

for i=1:n-2
    A(i*(n-1), i*(n-1)+1) = 0;
    A(i*(n-1)+1, i*(n-1)) = 0;
end

% generate right-hand side vector
b = zeros((n-1)*(n-1), 1);

b(1:n-1) = b(1:n-1) + 100/n*[1:n-1]';
b(n-1:n-1:end) = b(n-1:n-1:end) + 100/n*[n-1:-1:1]';

% use Doolittle factoization to solve for w
[L, U, P] = lu(A);
```

```
y = front_substition(L, b, (n-1)*(n-1));
w = back_substition(U, y, (n-1)*(n-1));

% Part b

% resize w
W = zeros(n+1, n+1);
W(2:end-1, 2:end-1) = reshape(w, n-1, n-1);
W(1, :) = 100/n*[0:n];
W(:, end) = 100/n*[n:-1:0];
x= 0.5/n*[0:n];
y= 0.5/n*[n:-1:0];

[xx, yy] = meshgrid(x, y);

% graph the results
figure();
surf(xx, yy, W);
```

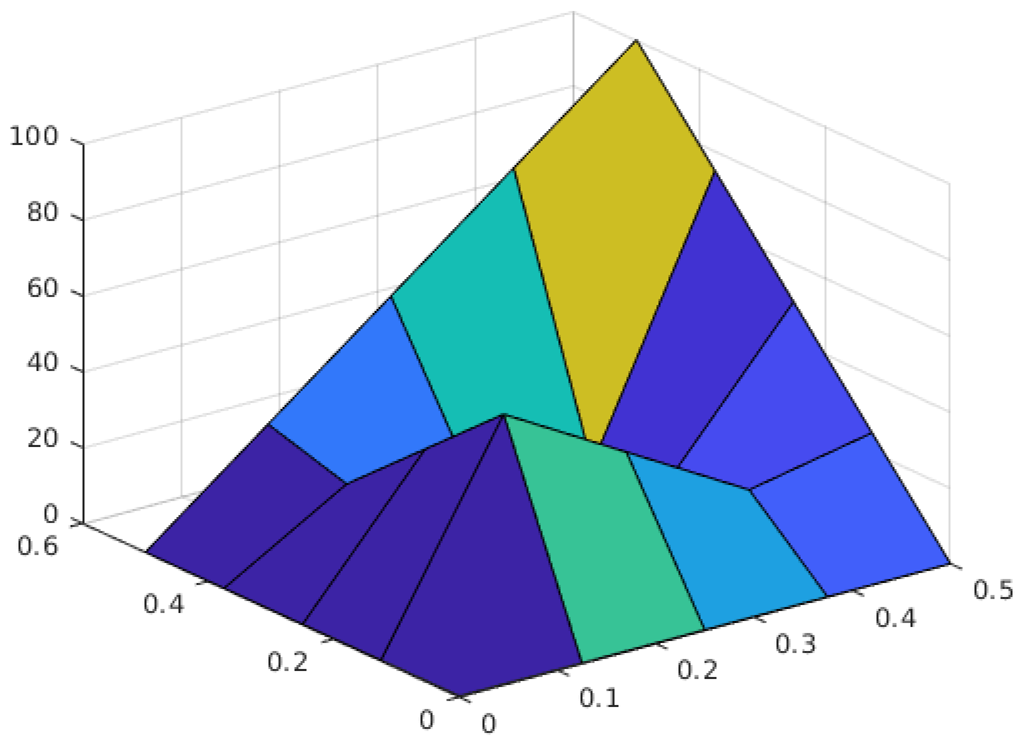


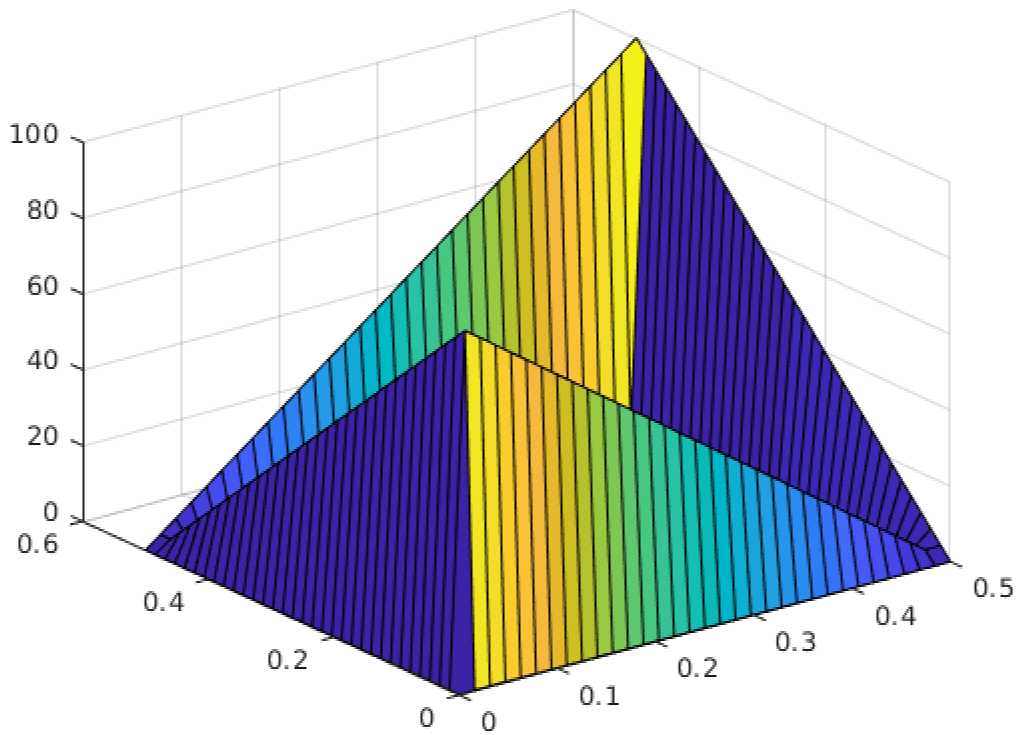

# Part c

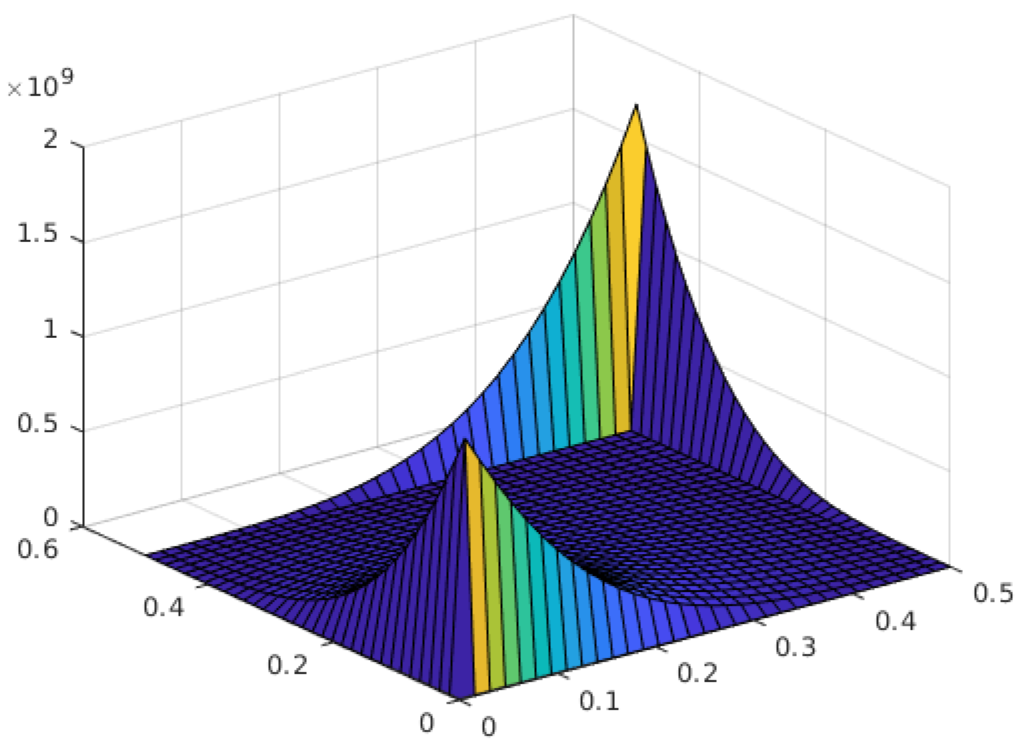
```
Find the steady-state heat distribution in a thin sqare metal
using n = 32 subintervals

% number of subintervals
```

```
n = 32;

% generate coefficient matrix
A = toeplitz([4 -1 zeros(1, n-3) -1 zeros(1, (n-2)*(n-1)-1)]);

for i=1:n-2
    A(i*(n-1), i*(n-1)+1) = 0;
    A(i*(n-1)+1, i*(n-1)) = 0;
end

% generate right-hand side vector
b = zeros((n-1)*(n-1), 1);

b(1:n-1) = b(1:n-1) + 100/n*[1:n-1]';
b(n-1:n-1:end) = b(n-1:n-1:end) + 100/n*[n-1:-1:1]';

% use Doolittle factorization to solve for w
[L, U, P] = lu(A);

y = front_substition(L, b, (n-1)*(n-1));
w = back_substition(U, y, (n-1)*(n-1));

% resize w
W = zeros(n+1, n+1);
W(2:end-1, 2:end-1) = reshape(w, n-1, n-1);
W(1, :) = 100/n*[0:n];
W(:, end) = 100/n*[n:-1:0];
x= 0.5/n*[0:n];
y= 0.5/n*[n:-1:0];

[xx, yy] = meshgrid(x, y);

% graph the results
figure();
surf(xx, yy, W);

% Part d
%  Find the steady-state heat distribution in a thin sqare metal
%  with the temperature distributions f(x) = 1600x^4 and f(y) =
 1600y^4

% number of subintervals
n = 32;

% update right-hand side vector
b = zeros((n-1)*(n-1), 1);

b(1:n-1) = b(1:n-1) + 1600.*([1:n-1].^4)';
b(n-1:n-1:end) = b(n-1:n-1:end) + 1600.*([n-1:-1:1].^4)';

% use Doolittle factorization to solve for w
y = front_substition(L, b, (n-1)*(n-1));
w = back_substition(U, y, (n-1)*(n-1));
```

```
% resize w
W = zeros(n+1, n+1);
W(2:end-1, 2:end-1) = reshape(w, n-1, n-1);
W(1, :) = 1600.*([0:n].^4);
W(:, end) = 1600.*([n:-1:0].^4);
x= 0.5/n*[0:n];
y= 0.5/n*[n:-1:0];

[xx, yy] = meshgrid(x, y);

% graph the results
figure();
surf(xx, yy, W);
```




100
80
60
40
20
0
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5


$\times 10^9$
2
1.5
1
0.5
0
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5

*Published with MATLAB® R2018b*