Tian Qiu

00265 35063

CS 314

Problem1:

1. For Ax

Multiplication operations: r \* s

Addition operations: r \* (s – 1)

1. DA is m \* n matrix. D is a diagonal matrix

So (DA)ij = Dii \* Aij

1. AD is m \* n matrix. D is a diagonal matrix

So (AD)ij = Djj \* Aij

Problem 2:

1.

function matmul(A, B)

Arow = size(A, 1)

Bcol = size(B,2)

Brow = size(B,1)

Acol = size(A,2)

C = zeros(Float64, Arow, Bcol)

For I = 1:Arow

For j=1:Bcol

For k=1:Acol

C[I,j] = c[I,j] + A[I,k] + b[k,j]

End

End

End

Return C

End

2. @show matmul([1 2], [2; 1])

[4.0]

@show matmul([1 2; -2 4; 0 3], [-1; 1])

1.0

6.0

3.0

3\*1 array

@show matmul([-1 1], [1 -2 0; 2 4 3])

[1.0 6.0 3.0]

@show matmul(3, 5) 15

t = pi/2

@show matmul([cos(t) -sin(t); sin(t) cos(t)], [cos(t) sin(t); -sin(t) cos(t)]);

[1.0 0.0 0.0 1.0]

3.

Pkg.add("Plots")

sample = [50,100,150,200,250,300,350,400,450,800,1000]

our = zeros(Float64,13)

buildin = zeros(Float64,13)

diff = zeros(Float64,13)

for i =1:10

for j = 1:50

A = randn(sample[i], sample[i])

B = randn(sample[i], sample[i])

our[i] += @elapsed C = matmul(A,B)

buildin[i] += @elapsed D = A\*B

diff[i] = vecnorm(C-D, )

end

our[i] = our[i]/50

buildin[i] /= 50

diff[i]/=50

end

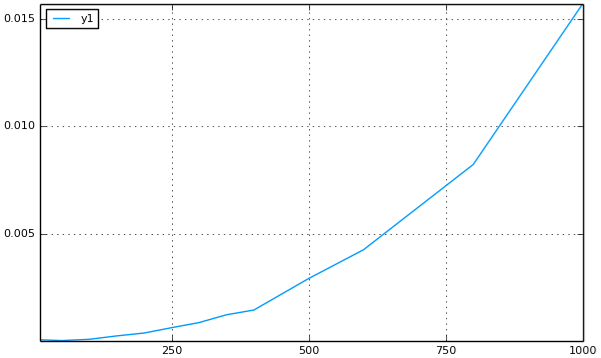
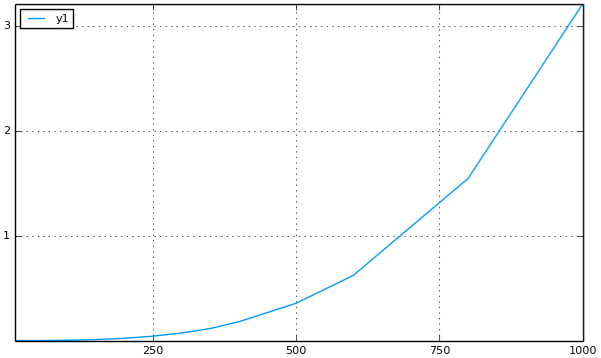
using Plots

plot(sample,our)

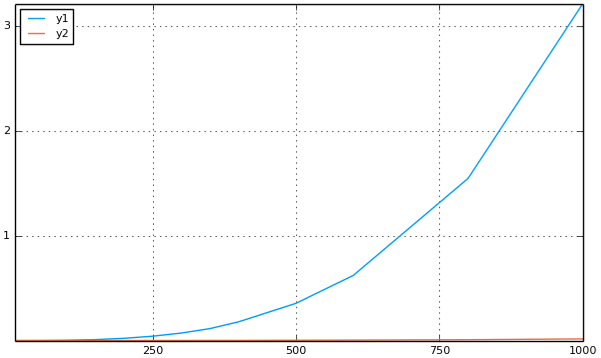
# plot(sample, buildin)

plot!(sample, buildin)

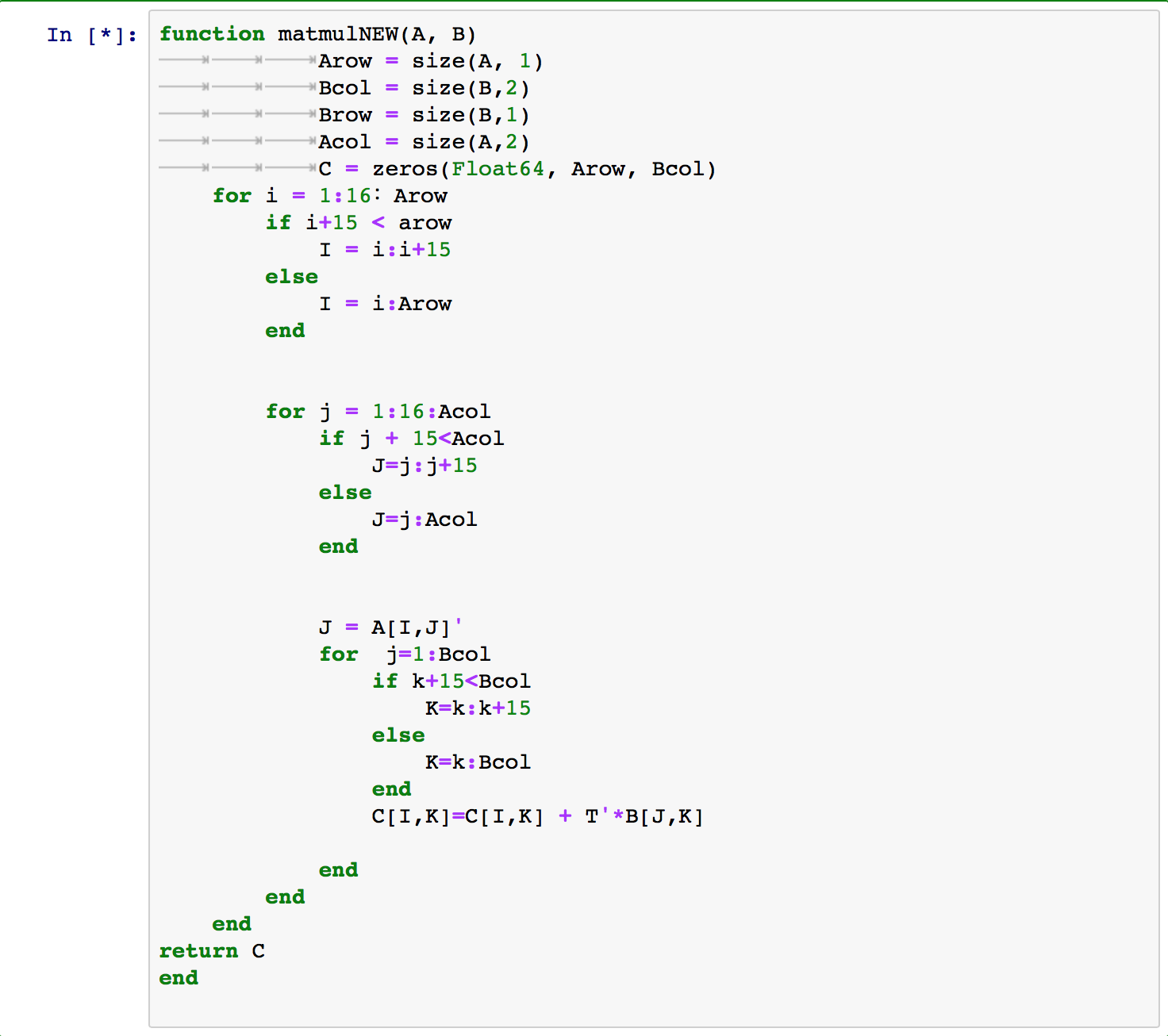
Julia builid-in function: Hand Write function:

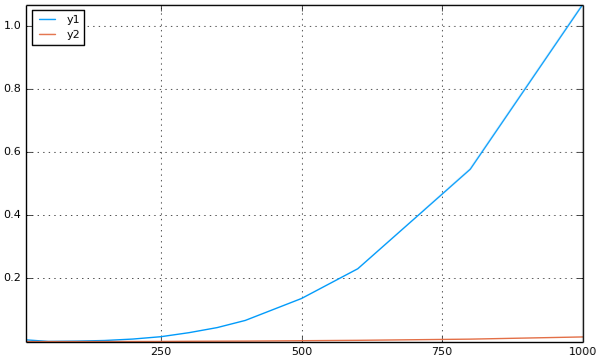
Together:







Show in Together:



Problem 3:



Code:

K = [5 6 4 5; 0 2 1 0]

Theta = pi /2

Plot(K[1,:]’, K[2,:]’, fill=(0, red))

R = [cos(Theta) – sin (Theta); sin(Theta) cos(Theta)];

V = R\*K

Plot(V[1,:]’, V[2,:]’, fill=(0, blue))

Theta += pi /2

R = [cos(Theta) – sin (Theta); sin(Theta) cos(Theta)];

V = R\*K

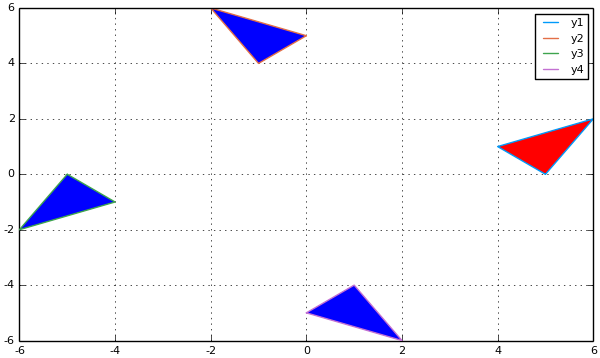
Plot(V[1,:]’, V[2,:]’, fill=(0, blue))

Theta += pi /2

R = [cos(Theta) – sin (Theta); sin(Theta) cos(Theta)];

V = R\*K

Plot(V[1,:]’, V[2,:]’, fill=(0, blue))





theta = pi/3;

R1 = [cos(theta) -sin(theta); sin(theta) cos(theta)];

theta = -pi/3;

R2 = [cos(theta) -sin(theta); sin(theta) cos(theta)];

R1\*R2

2x2 Array:

1 0

1 0

theta = pi/4;

R1 = [cos(theta) -sin(theta); sin(theta) cos(theta)];

theta = -pi/4;

R2 = [cos(theta) -sin(theta); sin(theta) cos(theta)];

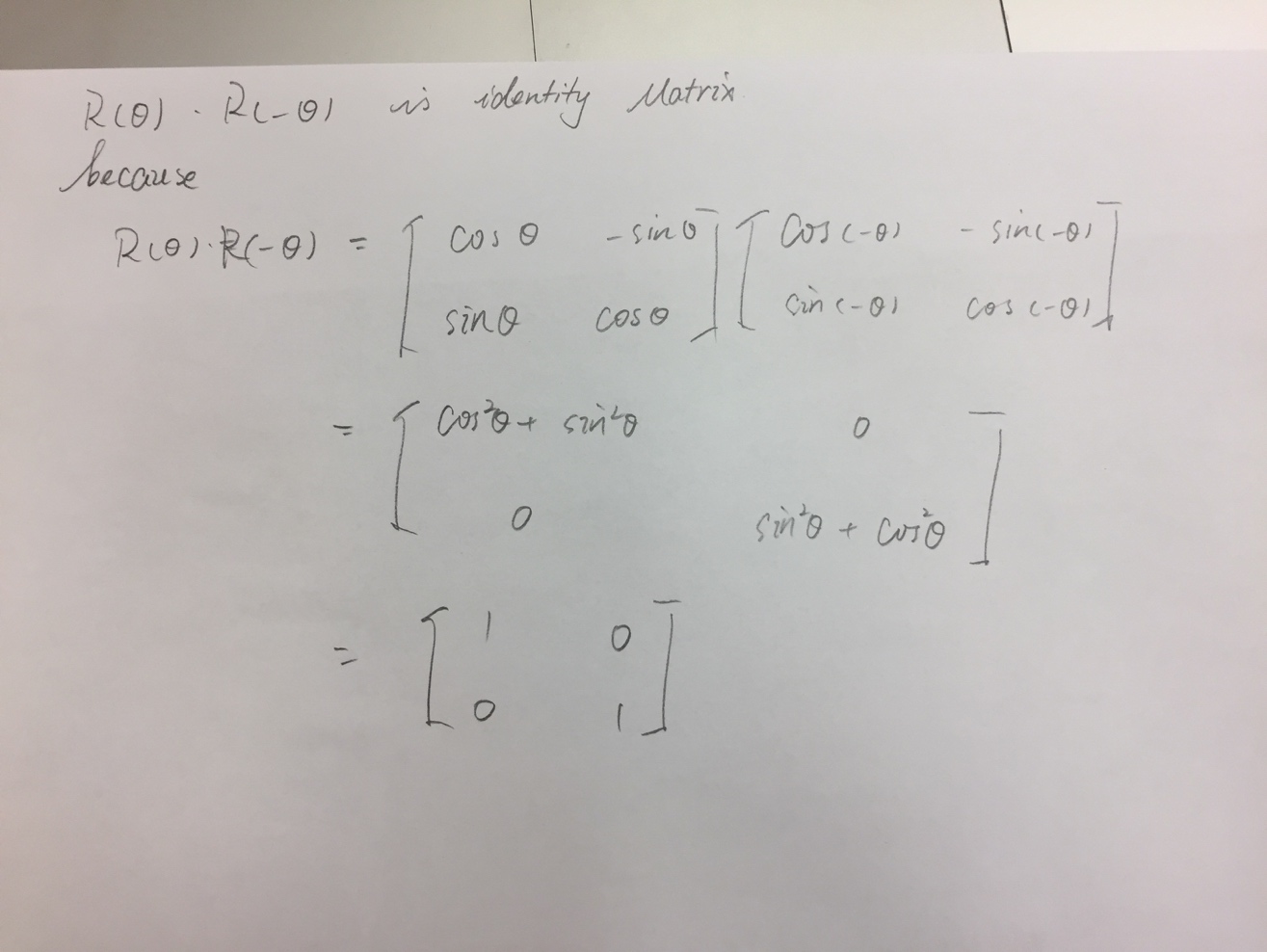
R1\*R2

2x2 Array:

1 0

1 0







# Julia 3\_4

K = [1 0 -1 0 1; 0 1 0 -1 0]

Theta = pi / 8

Plot(K[1,:]’, K[2,:]’, fill=(0, red))

For i=1:50

R = [cos(Theta) – sin (Theta); sin(Theta) cos(Theta)];

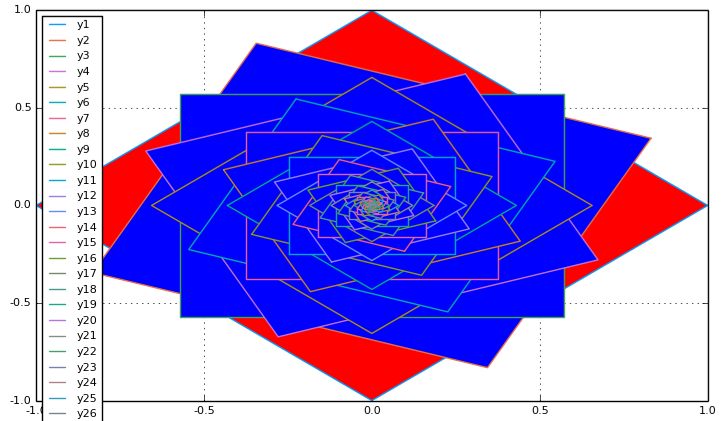
V = R\*K \* 0.9

Plot!(V[1,:]’, V[2,:]’, fill=(0, blue))

K=V;

End

Plot!(V[1,:]’, V[2,:]’, fill=(0, blue))



# Julia 3\_4

K = [1 0 -1 0 1; 0 1 0 -1 0]

Theta = pi / 8

Plot(K[1,:]’, K[2,:]’, fill=(0, red))

For i=1:50

R = 0.9 \* [cos(Theta) – sin (Theta); sin(Theta) cos(Theta)];

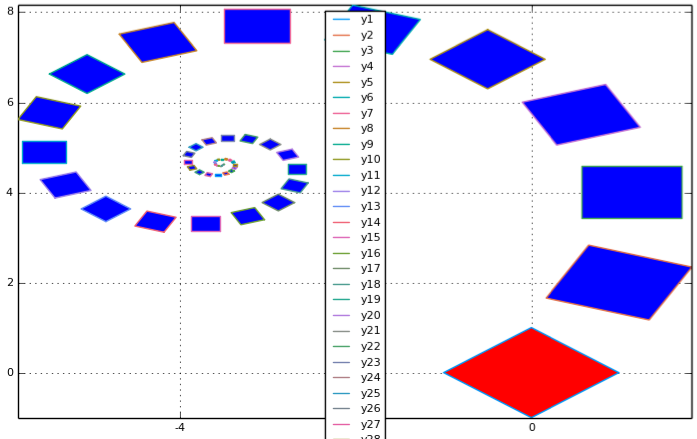
V = R\*K;

V += [1 1 1 1 1; 2 2 2 2 2]

Plot!(V[1,:]’, V[2,:]’, fill=(0, blue))

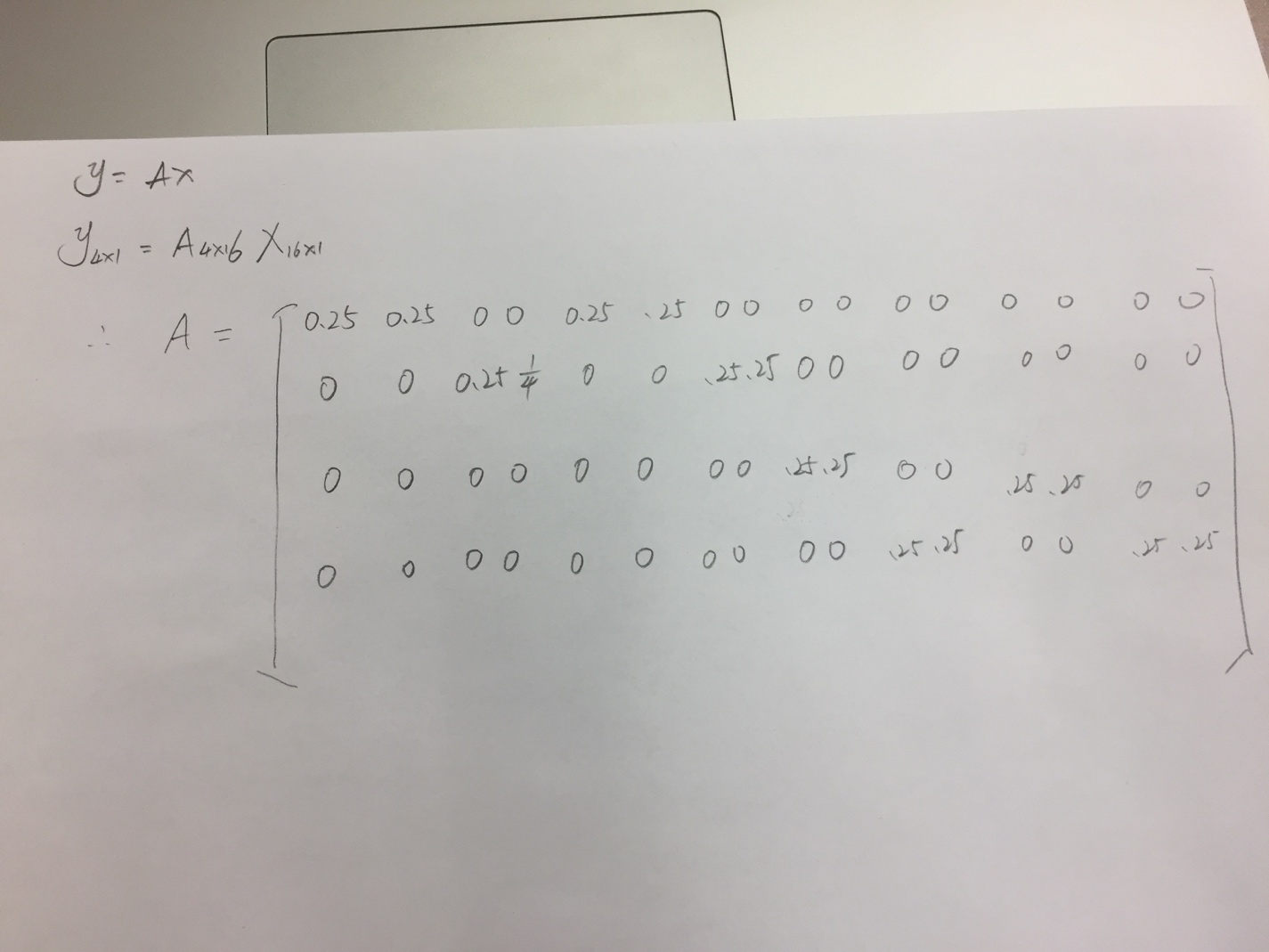
K=V;

End



Problem 4

1.



2.

download("http://www.cs.purdue.edu/homes/dgleich/cs314-2016/homeworks/smallicon.csv","smallicon.csv")

x = readcsv("smallicon.csv")

Sum(diag(x))

24.368627

3.

using FileIO

using ImageMagick

save("icon1.png",X)



4.

The reshape function changes a 1\*16 matrix into a 4\*4 matrix, which is shaping row by row. But without transpose, it will be reshaping column by column which is not what we want.

N = reshape(1:(4\*4), 4, 4)'

4×4 Array{Int64,2}:

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16

N(1,4)

N(3,2)

N = reshape(1:(4\*4), 4, 4)

4×4 Base.ReshapedArray{Int64,2,UnitRange{Int64},Tuple{}}:

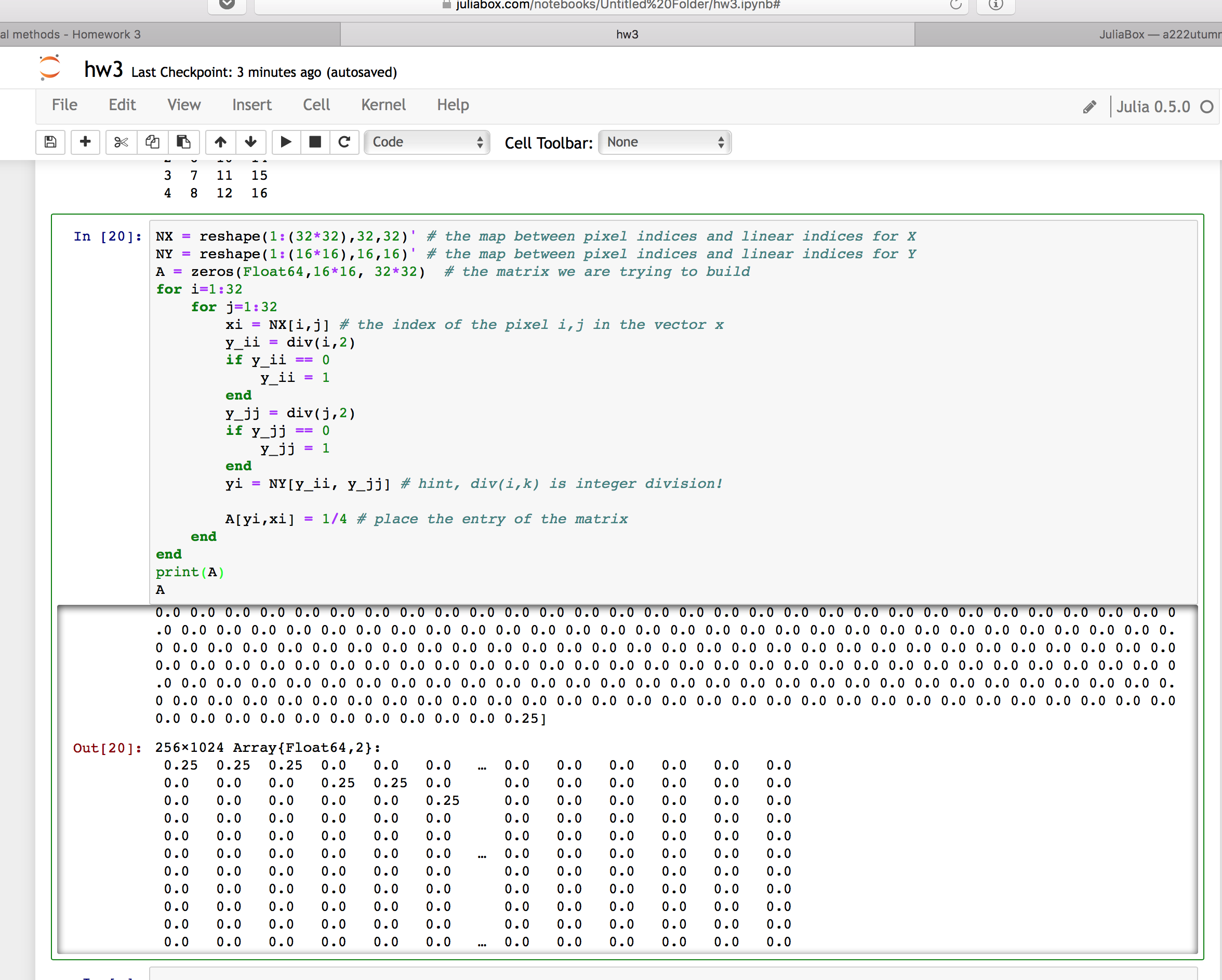
1 5 9 13

2 6 10 14

3 7 11 15

4 8 12 16

5.



6.

x = reshape(X',32\*32,1)

1024\*1 Array{Float64, 2}:

1

1

1

…

1

1

1

y = A\*x

256x1 Array{Float64, 2}

2.25

1.5

1.5

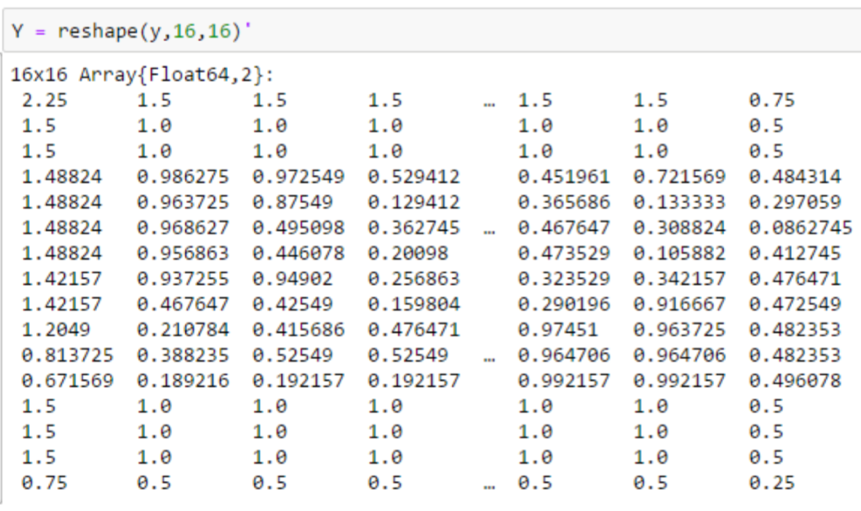
….

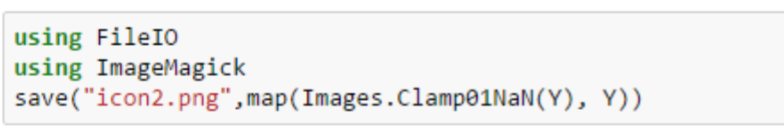
0.5

0.5

…

0.25







The resolution becomes very low for this output comparing to the previous one.