Aditya Patel

NCDS – Unit 2 HW

Q1.)

To solve, I modified Professor Gleich’s fixed-point.jl code to include a function to find the mean Euclidean distance between the sets of points (meanEucDist), and adding a for loop to iterate rotation between 1 and 359 degrees:

##

using Printf

using FixedPointNumbers

using FixedPointDecimals

using LinearAlgebra

##

x = FixedDecimal{Int,2}(5)

## Emulate the Mario Kart system to rotate a dinosaur

MKFP = Fixed{Int16,8} # use 16 bits for a signed integer and 8 bits for fraction

val = MKFP(5.25)

## Function to determine average square drift between 2 sets of equal length

function meanEucDist(act, ref)

med = sqrt(sum((act .- ref).^2))

return med

end

## Get some data off the internet and plot it

data = [24,18, 24,17, 22,16, 20,11, 19,6, 19,2, 17,2, 17,6,

16,5, 15,2, 13,2, 14,5, 14,6, 12,6, 12,2, 10,2, 10,4,

9,2, 7,2, 9,6, 7,6, 4,4, 2,3, 0,2, 1,3, 3,5, 5,9,

9,11, 17,11, 21,17, 23,18, 24,18]

P = reshape(data,2,div(length(data),2))

using Plots

driftDict = Dict{Int64, Float64}()

## Rotate 359 degrees

for deg in 1:359

local plt = plot()

global theta = deg/180\*pi

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

global Pp = R(theta)\*P # True

plot!(plt, Pp[1,:],Pp[2,:],linestyle = :solid,marker = :circle,color=2,lab="Float64")

# Rotate in fixed Point

global Pmk = MKFP.(R(MKFP(theta)))\*MKFP.(P)

plot!(plt, Pmk[1,:],Pmk[2,:],linestyle = :solid,marker = :circle,color=4,lab="Mario Kart")

# Calculate mean euclidean distance between points

local err = meanEucDist(Pp, Float64.(Pmk))

title!("ED between points at $deg degrees: $err")

push!(driftDict, deg => err)

display(plt)

end

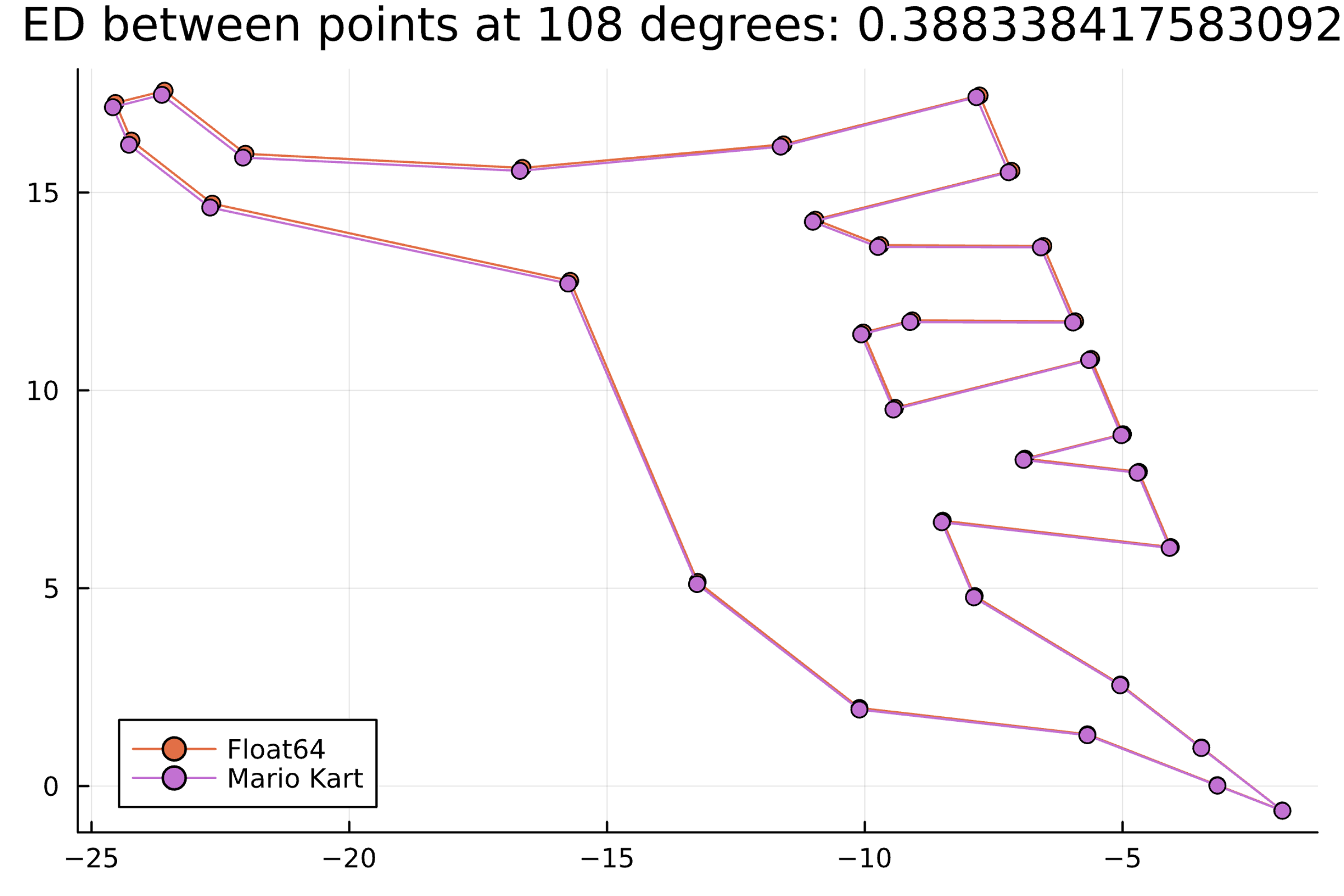
## Find the maximum drift value

maxDeg, maxDrift, = findmax(driftDict)

The output of this is: (0.3883384175830925, 108)

Translated, this means that the maximum mean Euclidean distance is 0.3883… at a 108-degree rotation.

Plotting these points, we see that this is a good estimate, as there are distinct visual differences between the points on the graph.



2a)

Three methods of calculating roots are:

1. Utilizing the quadratic formula
2. Utilizing the Newton-Rhapson formula and solving   
   For a quadratic function , we solve . We guess some value of x that we believe to be the 0, and then iterate until convergence.
3. Utilizing the completing the square method.   
   Using the form
4. Divide all terms by a:
5. Move the constant term to the other side:
6. Add and subtract from the left side:

2b.

Converting the methods to Julia code and running values of a, b, and c from -10 to +10:

##

using Printf

using LinearAlgebra

## Define functions for the three solution methods

function quadraticFormula(a, b, c)

if b^2 > 4\*a\*c

return (-b + sqrt(b^2 - 4\*a\*c)) / (2\*a), (-b - sqrt(b^2 - 4\*a\*c)) / (2\*a)

end

end

function completeTheSquare(a, b, c)

## Normalize all terms:

c = c/a

b = b/a

a = a/a

## Constant term to right side:

c = -c

## Add/Subtract b to make the perfect trinomial

b\_half\_sqr = (b/2)^2

## Squared Binomial

rhs = c + b\_half\_sqr

## Solve for X

if rhs > 0

## Real Roots

roots = -b/2 + (sqrt(rhs)), -b/2 - (sqrt(rhs))

return roots

end

end

function NewtonRaphsonMethod(a, b, c, itr, guess)

## Define f(x) and f'(x)

f(x, a, b, c) = a\*x^2 + b\*x + c

f\_prime(x, a, b) = 2\*a\*x + b

x = guess

for n in 1:itr

f\_x = f(x, a, b, c)

f\_xp = f\_prime(x, a, b)

if abs(f\_xp) < 1e-10

return x

end

x\_new = x - f\_x/f\_xp

if abs(x\_new - x) < 1e-10

return x\_new

end

x = x\_new

end

return x

end

## Execute with Float16 values for a, b, c, and x\_guess

for i in -10:10

for j in -10:10

for k in -10:10

local a = Float16(i)

local b = Float16(j)

local c = Float16(k)

local x\_guess = Float16((sum([i, j, k])/3))

quadRoot = quadraticFormula(a, b, c)

ctsRoot = completeTheSquare(a, b, c)

newtonRoot = NewtonRaphsonMethod(a, b, c, 1000, x\_guess)

if !isnothing(quadRoot) && !isnothing(ctsRoot) ## Only take real zeros

if quadRoot[1] != ctsRoot[1] != newtonRoot ## Check for where they are not equal

println("Root 1 with a=$i, b=$j, c=$k:\t", quadRoot[1], "\t\t\t", ctsRoot[1], "\t\t\t", newtonRoot)

end

end

end

end

end

We get the following results:

Root 1 with a=-10, b=-10, c=-2: -0.7236 -0.2764 -0.724

Root 1 with a=-10, b=-9, c=-2: -0.5 -0.3994 -0.4976

Root 1 with a=-10, b=-9, c=-1: -0.7705 -0.1296 -0.77

Root 1 with a=-10, b=-7, c=-1: -0.5 -0.1998 -0.5005

Root 1 with a=-10, b=-5, c=0: -0.5 0.0 -0.4998

Root 1 with a=-9, b=-10, c=-2: -0.8496 -0.2612 -0.8506

Root 1 with a=-9, b=-9, c=-2: -0.6665 -0.333 -0.667

Root 1 with a=-9, b=-9, c=-1: -0.8726 -0.1272 -0.873

Root 1 with a=-9, b=-9, c=0: -1.0 0.0 -0.9995

Root 1 with a=-9, b=-8, c=0: -0.8887 0.0 -0.889

Root 1 with a=-9, b=-7, c=-1: -0.5894 -0.1885 -0.589

Root 1 with a=-9, b=-4, c=0: -0.4443 0.0 -0.4446

Root 1 with a=-9, b=-2, c=0: -0.2222 0.0 -0.2223

Root 1 with a=-9, b=-1, c=0: -0.1111 0.0 -0.11115

Root 1 with a=-8, b=-10, c=-2: -1.0 -0.25 -0.9995

Root 1 with a=-8, b=-9, c=-2: -0.8203 -0.3047 -0.821

Root 1 with a=-8, b=-6, c=-1: -0.5 -0.25 -0.5005

Root 1 with a=-8, b=-6, c=0: -0.75 0.0 -0.7495

Root 1 with a=-7, b=-10, c=-3: -1.0 -0.4285 -1.001

Root 1 with a=-7, b=-8, c=-2: -0.7734 -0.3694 -0.773

Root 1 with a=-7, b=-7, c=-1: -0.827 -0.1726 -0.828

Root 1 with a=-7, b=-6, c=-1: -0.6304 -0.2266 -0.631

Root 1 with a=-6, b=-10, c=-4: -1.0 -0.665 -0.9995

Root 1 with a=-6, b=-10, c=-2: -1.435 -0.2324 -1.434

Root 1 with a=-6, b=-10, c=0: -1.667 0.0 -1.666

Root 1 with a=-6, b=-9, c=-3: -1.0 -0.5 -0.9995

Root 1 with a=-6, b=-9, c=-2: -1.2295 -0.2712 -1.23

Root 1 with a=-6, b=-8, c=-1: -1.194 -0.1396 -1.193

Root 1 with a=-6, b=-7, c=-2: -0.6665 -0.498 -0.666

Root 1 with a=-6, b=-5, c=-1: -0.5 -0.3325 -0.4995

Root 1 with a=-6, b=-5, c=0: -0.8335 0.0 -0.834

Root 1 with a=-6, b=-4, c=0: -0.6665 0.0 -0.667

Root 1 with a=-6, b=-2, c=0: -0.3333 0.0 -0.3335

Root 1 with a=-6, b=-1, c=0: -0.1666 0.0 -0.1667

Root 1 with a=-5, b=-10, c=-4: -1.447 -0.5527 -1.448

Root 1 with a=-5, b=-9, c=-4: -1.0 -0.799 -0.995

Root 1 with a=-5, b=-9, c=-3: -1.357 -0.4417 -1.358

Root 1 with a=-5, b=-9, c=-2: -1.541 -0.2593 -1.54

Root 1 with a=-5, b=-9, c=-1: -1.682 -0.11865 -1.681

Root 1 with a=-5, b=-8, c=-3: -1.0 -0.601 -1.001

Root 1 with a=-5, b=-8, c=-2: -1.29 -0.31 -1.289

Root 1 with a=-5, b=-8, c=-1: -1.463 -0.1367 -1.464

Root 1 with a=-5, b=-5, c=-1: -0.7236 -0.2764 -0.724

Root 1 with a=-4, b=-10, c=-5: -1.809 -0.691 -1.811

Root 1 with a=-4, b=-10, c=-4: -2.0 -0.5 -1.999

Root 1 with a=-4, b=-9, c=-5: -1.25 -1.0 -1.251

Root 1 with a=-4, b=-9, c=-3: -1.844 -0.4067 -1.842

Root 1 with a=-4, b=-7, c=-3: -1.0 -0.75 -0.9985

Root 1 with a=-3, b=-10, c=-2: -3.12 -0.2129 -3.121

Root 1 with a=-3, b=-10, c=-1: -3.229 -0.1035 -3.23

Root 1 with a=-3, b=-9, c=-6: -2.0 -1.0 -1.999

Root 1 with a=-3, b=-9, c=-4: -2.459 -0.5425 -2.457

Root 1 with a=-3, b=-8, c=-3: -2.217 -0.4512 -2.215

Root 1 with a=-3, b=-8, c=-2: -2.389 -0.2793 -2.387

Root 1 with a=-3, b=-7, c=-3: -1.769 -0.565 -1.77

Root 1 with a=-3, b=-7, c=-1: -2.18 -0.1523 -2.182

Root 1 with a=-3, b=-6, c=-2: -1.578 -0.4224 -1.576

Root 1 with a=-3, b=-5, c=-2: -1.0 -0.665 -1.001

Root 1 with a=-3, b=-2, c=0: -0.6665 0.0 -0.667

Root 1 with a=-2, b=-10, c=-10: -3.617 -1.382 -3.621

Root 1 with a=-2, b=-10, c=-9: -3.824 -1.177 -3.82

Root 1 with a=-2, b=-10, c=-2: -4.79 -0.209 -4.793

Root 1 with a=-2, b=-9, c=-10: -2.5 -2.0 -2.498

Root 1 with a=-2, b=-9, c=-9: -3.0 -1.5 -2.998

Root 1 with a=-2, b=-9, c=-7: -3.5 -1.0 -3.498

Root 1 with a=-2, b=-9, c=-4: -4.0 -0.5 -3.998

Root 1 with a=-2, b=-8, c=-3: -3.582 -0.419 -3.58

Root 1 with a=-2, b=-7, c=-6: -2.0 -1.5 -2.004

Root 1 with a=-2, b=-7, c=-5: -2.5 -1.0 -2.498

Root 1 with a=-2, b=-7, c=-2: -3.188 -0.3135 -3.186

Root 1 with a=-2, b=-6, c=-4: -2.0 -1.0 -2.002

Root 1 with a=-2, b=-6, c=-3: -2.367 -0.634 -2.365

Root 1 with a=-2, b=-6, c=-1: -2.824 -0.1768 -2.82

Root 1 with a=-2, b=-5, c=-3: -1.5 -1.0 -1.497

Root 1 with a=-2, b=-5, c=-2: -2.0 -0.5 -1.999

Root 1 with a=-2, b=-3, c=-1: -1.0 -0.5 -1.001

Root 1 with a=-2, b=-3, c=0: -1.5 0.0 -1.499

Root 1 with a=-1, b=-10, c=-4: -9.58 -0.418 -5.0

Root 1 with a=-1, b=-10, c=-3: -9.69 -0.3086 -0.3096

Root 1 with a=-1, b=-10, c=-2: -9.8 -0.2031 -0.2043

Root 1 with a=-1, b=-10, c=-1: -9.9 -0.10156 -0.10095

Root 1 with a=-1, b=-9, c=-10: -7.703 -1.299 -7.7

Root 1 with a=-1, b=-9, c=-9: -7.85 -1.146 -7.855

Root 1 with a=-1, b=-9, c=-5: -8.41 -0.5957 -8.4

Root 1 with a=-1, b=-9, c=-4: -8.53 -0.4688 -8.52

Root 1 with a=-1, b=-9, c=-3: -8.66 -0.3477 -0.3467

Root 1 with a=-1, b=-9, c=-2: -8.77 -0.2266 -0.228

Root 1 with a=-1, b=-9, c=-1: -8.89 -0.1133 -0.11255

Root 1 with a=-1, b=-8, c=-10: -6.45 -1.551 -6.445

Root 1 with a=-1, b=-8, c=-6: -7.164 -0.838 -7.16

Root 1 with a=-1, b=-8, c=-3: -7.605 -0.3945 -4.0

Root 1 with a=-1, b=-8, c=-2: -7.742 -0.2578 -0.2583

Root 1 with a=-1, b=-8, c=-1: -7.875 -0.127 -0.1271

Root 1 with a=-1, b=-7, c=-9: -5.305 -1.697 -5.3

Root 1 with a=-1, b=-7, c=-7: -5.79 -1.209 -5.797

Root 1 with a=-1, b=-7, c=-5: -6.195 -0.8066 -6.19

Root 1 with a=-1, b=-7, c=-3: -6.54 -0.459 -6.543

Root 1 with a=-1, b=-7, c=-2: -6.703 -0.2988 -0.2986

Root 1 with a=-1, b=-7, c=-1: -6.85 -0.1465 -0.1459

Root 1 with a=-1, b=-6, c=-6: -4.734 -1.268 -4.73

Root 1 with a=-1, b=-6, c=-4: -5.234 -0.7637 -5.24

Root 1 with a=-1, b=-6, c=-2: -5.65 -0.3535 -3.0

Root 1 with a=-1, b=-6, c=-1: -5.83 -0.1719 -0.1716

Root 1 with a=-1, b=-5, c=-6: -3.0 -2.0 -2.998

Root 1 with a=-1, b=-5, c=-5: -3.617 -1.382 -3.615

Root 1 with a=-1, b=-5, c=-4: -4.0 -1.0 -3.998

Root 1 with a=-1, b=-5, c=-1: -4.79 -0.209 -0.2087

Root 1 with a=-1, b=-4, c=-3: -3.0 -1.0 -2.998

Root 1 with a=-1, b=-4, c=-1: -3.732 -0.2676 -2.0

Root 1 with a=-1, b=-2, c=0: -2.0 0.0 -1.0

Root 1 with a=3, b=5, c=2: -0.6665 -0.665 -0.6675

Root 1 with a=3, b=7, c=1: -0.153 -0.1523 -0.1528

Root 1 with a=3, b=7, c=2: -0.3333 -0.333 -0.3335

Root 1 with a=3, b=9, c=1: -0.1159 -0.11523 -0.11554

Root 1 with a=3, b=9, c=2: -0.2416 -0.2412 -0.2417

Root 1 with a=3, b=10, c=1: -0.10284 -0.1035 -0.1032

Root 1 with a=3, b=10, c=2: -0.2135 -0.2129 -0.2136

Root 1 with a=3, b=10, c=3: -0.3333 -0.333 -0.3335

Root 1 with a=3, b=10, c=8: -1.333 -1.33 -1.336

Root 1 with a=5, b=7, c=1: -0.1614 -0.1616 -0.1615

Root 1 with a=5, b=7, c=2: -0.4 -0.3997 -0.4004

Root 1 with a=5, b=8, c=3: -0.6 -0.601 -0.5996

Root 1 with a=5, b=9, c=1: -0.11914 -0.11865 -0.119

Root 1 with a=5, b=9, c=4: -0.8 -0.799 -0.802

Root 1 with a=5, b=10, c=2: -0.2253 -0.2251 -0.2255

Root 1 with a=6, b=5, c=1: -0.3333 -0.3325 -0.3337

Root 1 with a=6, b=7, c=1: -0.1666 -0.1665 -0.1667

Root 1 with a=6, b=7, c=2: -0.5 -0.498 -0.4998

Root 1 with a=6, b=9, c=1: -0.1208 -0.1206 -0.12085

Root 1 with a=6, b=10, c=1: -0.10675 -0.10645 -0.1069

Root 1 with a=6, b=10, c=4: -0.6665 -0.665 -0.6655

Root 1 with a=7, b=9, c=1: -0.12274 -0.12256 -0.1228

Root 1 with a=7, b=9, c=2: -0.2856 -0.2854 -0.286

Root 1 with a=7, b=10, c=1: -0.1083 -0.1084 -0.1082

Root 1 with a=9, b=8, c=1: -0.1504 -0.1506 -0.1505

Root 1 with a=9, b=9, c=2: -0.3333 -0.333 -0.3335

Root 1 with a=10, b=7, c=1: -0.2 -0.1998 -0.2001

Root 1 with a=10, b=9, c=2: -0.4 -0.3994 -0.4001

Using Float64:

Root 1 with a=-10, b=-10, c=-1: -0.8872983346207416 -0.1127016653792583 -0.8872983346207417

Root 1 with a=-10, b=-9, c=-1: -0.7701562118716424 -0.12984378812835756 -0.7701562118716423

Root 1 with a=-10, b=-9, c=0: -0.9 0.0 -0.8999999999999999

Root 1 with a=-10, b=-7, c=-1: -0.5 -0.20000000000000004 -0.4999999999999999

Root 1 with a=-10, b=-6, c=0: -0.6 0.0 -0.5999999999999999

Root 1 with a=-10, b=-4, c=0: -0.4 0.0 -0.39999999999999997

Root 1 with a=-10, b=-2, c=0: -0.2 0.0 -0.19999999999999998

Root 1 with a=-10, b=-1, c=0: -0.1 0.0 -0.09999999999999999

Root 1 with a=-9, b=-9, c=-1: -0.872677996249965 -0.12732200375003505 -0.8726779962499648

Root 1 with a=-9, b=-8, c=-1: -0.73841681234051 -0.1504720765483788 -0.7384168123405102

Root 1 with a=-9, b=-7, c=-1: -0.5891972930813327 -0.188580484696445 -0.5891972930813326

Root 1 with a=-9, b=-4, c=0: -0.4444444444444444 0.0 -0.4444444444444445

Root 1 with a=-9, b=-1, c=0: -0.1111111111111111 0.0 -0.11111111111111112

Root 1 with a=-8, b=-10, c=-3: -0.75 -0.5 -0.7499999999999997

Root 1 with a=-8, b=-10, c=-1: -1.1403882032022077 -0.10961179679779243 -1.1403882032022075

Root 1 with a=-8, b=-9, c=-1: -1.0 -0.125 -0.9999999999999999

Root 1 with a=-8, b=-7, c=-1: -0.6951941016011038 -0.1798058983988962 -0.6951941016011037

Root 1 with a=-8, b=-6, c=0: -0.75 0.0 -0.7499999999999999

Root 1 with a=-8, b=-3, c=0: -0.375 0.0 -0.37500000000000006

Root 1 with a=-7, b=-10, c=-1: -1.3203772410170405 -0.10819418755438781 -1.3203772410170407

Root 1 with a=-7, b=-9, c=-1: -1.1628649920914655 -0.12284929362282004 -1.1628649920914653

Root 1 with a=-7, b=-9, c=0: -1.2857142857142858 0.0 -1.2857142857142856

Root 1 with a=-7, b=-6, c=-1: -0.6306019374818707 -0.2265409196609864 -0.6306019374818708

Root 1 with a=-7, b=-6, c=0: -0.8571428571428571 0.0 -0.8571428571428572

Root 1 with a=-7, b=-5, c=0: -0.7142857142857143 0.0 -0.7142857142857142

Root 1 with a=-6, b=-10, c=-4: -1.0 -0.6666666666666663 -1.0000000000000002

Root 1 with a=-6, b=-10, c=-3: -1.2742918851774319 -0.3923747814892348 -1.2742918851774314

Root 1 with a=-6, b=-10, c=-2: -1.434258545910665 -0.2324081207560017 -1.4342585459106654

Root 1 with a=-6, b=-10, c=-1: -1.5598164905901122 -0.10685017607655434 -1.5598164905901124

Root 1 with a=-6, b=-10, c=0: -1.6666666666666667 0.0 -1.666666666666667

Root 1 with a=-6, b=-9, c=-3: -1.0 -0.5 -0.9999999999999999

Root 1 with a=-6, b=-9, c=-1: -1.379152869605896 -0.12084713039410411 -1.3791528696058957

Root 1 with a=-6, b=-8, c=-1: -1.1937129433613967 -0.1396203899719367 -1.1937129433613969

Root 1 with a=-6, b=-7, c=-2: -0.6666666666666666 -0.4999999999999995 -0.666666666666667

Root 1 with a=-6, b=-7, c=-1: -1.0 -0.16666666666666663 -0.9999999999999999

Root 1 with a=-6, b=-6, c=-1: -0.7886751345948128 -0.21132486540518708 -0.788675134594813

Root 1 with a=-6, b=-5, c=-1: -0.5 -0.33333333333333315 -0.4999999999999998

Root 1 with a=-6, b=-3, c=0: -0.5 0.0 -0.49999999999999994

Root 1 with a=-5, b=-10, c=-4: -1.4472135954999579 -0.5527864045000421 -1.4472135954999577

Root 1 with a=-5, b=-10, c=-3: -1.632455532033676 -0.3675444679663241 -1.6324555320336758

Root 1 with a=-5, b=-10, c=-2: -1.7745966692414832 -0.2254033307585166 -1.7745966692414834

Root 1 with a=-5, b=-9, c=-4: -1.0 -0.7999999999999999 -1.0000000000000004

Root 1 with a=-5, b=-9, c=-3: -1.3582575694955838 -0.4417424305044159 -1.358257569495584

Root 1 with a=-5, b=-9, c=0: -1.8 0.0 -1.8000000000000003

Root 1 with a=-5, b=-8, c=0: -1.6 0.0 -1.5999999999999999

Root 1 with a=-5, b=-7, c=-2: -1.0 -0.4000000000000001 -0.9999999999999998

Root 1 with a=-5, b=-7, c=-1: -1.2385164807134506 -0.16148351928654958 -1.2385164807134503

Root 1 with a=-5, b=-5, c=-1: -0.7236067977499789 -0.27639320225002106 -0.723606797749979

Root 1 with a=-5, b=-2, c=0: -0.4 0.0 -0.39999999999999997

Root 1 with a=-4, b=-10, c=-6: -1.5 -1.0 -1.5000000000000004

Root 1 with a=-4, b=-10, c=-5: -1.8090169943749475 -0.6909830056250525 -1.809016994374947

Root 1 with a=-4, b=-10, c=-3: -2.1513878188659974 -0.3486121811340027 -2.151387818865997

Root 1 with a=-4, b=-10, c=-1: -2.3956439237389597 -0.10435607626104004 -2.39564392373896

Root 1 with a=-4, b=-10, c=0: -2.5 0.0 -2.4999999999999996

Root 1 with a=-4, b=-9, c=-5: -1.25 -1.0 -1.2500000000000009

Root 1 with a=-4, b=-9, c=-4: -1.6403882032022077 -0.6096117967977924 -1.6403882032022075

Root 1 with a=-4, b=-8, c=-3: -1.5 -0.5 -1.4999999999999998

Root 1 with a=-4, b=-8, c=-2: -1.7071067811865475 -0.2928932188134524 -1.7071067811865477

Root 1 with a=-4, b=-7, c=-3: -1.0 -0.75 -1.0000000000000002

Root 1 with a=-4, b=-7, c=-2: -1.3903882032022077 -0.3596117967977924 -1.3903882032022075

Root 1 with a=-4, b=-7, c=-1: -1.5930703308172536 -0.15692966918274642 -1.5930703308172534

Root 1 with a=-4, b=-6, c=-2: -1.0 -0.5 -1.0000000000000002

Root 1 with a=-4, b=-5, c=-1: -1.0 -0.25 -0.9999999999999999

Root 1 with a=-3, b=-10, c=-6: -2.5485837703548637 -0.7847495629784697 -2.5485837703548633

Root 1 with a=-3, b=-10, c=-5: -2.720759220056127 -0.6125741132772069 -2.7207592200561264

Root 1 with a=-3, b=-10, c=-4: -2.86851709182133 -0.4648162415120034 -2.8685170918213294

Root 1 with a=-3, b=-10, c=-3: -3.0 -0.33333333333333326 -2.9999999999999996

Root 1 with a=-3, b=-10, c=-2: -3.1196329811802244 -0.21370035215310867 -3.119632981180225

Root 1 with a=-3, b=-10, c=-1: -3.23013858660781 -0.10319474672552342 -3.2301385866078096

Root 1 with a=-3, b=-7, c=-4: -1.3333333333333333 -0.999999999999999 -1.3333333333333335

Root 1 with a=-3, b=-6, c=-2: -1.5773502691896255 -0.42264973081037416 -1.5773502691896257

Root 1 with a=-3, b=-6, c=0: -2.0 0.0 -1.9999999999999998

Root 1 with a=-3, b=-5, c=-1: -1.434258545910665 -0.2324081207560017 -1.4342585459106647

Root 1 with a=-3, b=-1, c=0: -0.3333333333333333 0.0 -0.33333333333333337

Root 1 with a=-2, b=-10, c=-10: -3.618033988749895 -1.381966011250105 -3.6180339887498953

Root 1 with a=-2, b=-10, c=-9: -3.8228756555322954 -1.1771243444677046 -3.822875655532295

Root 1 with a=-2, b=-10, c=-5: -4.436491673103708 -0.5635083268962915 -4.436491673103709

Root 1 with a=-2, b=-10, c=-4: -4.561552812808831 -0.4384471871911697 -4.56155281280883

Root 1 with a=-2, b=-10, c=-2: -4.7912878474779195 -0.20871215252208009 -4.79128784747792

Root 1 with a=-2, b=-9, c=-10: -2.5 -2.0 -2.4999999999999996

Root 1 with a=-2, b=-9, c=-5: -3.850781059358212 -0.6492189406417879 -3.8507810593582126

Root 1 with a=-2, b=-9, c=-3: -4.1374586088176875 -0.36254139118231254 -4.137458608817687

Root 1 with a=-2, b=-9, c=-1: -4.386000936329383 -0.1139990636706174 -4.386000936329382

Root 1 with a=-2, b=-8, c=-7: -2.7071067811865475 -1.2928932188134525 -2.707106781186548

Root 1 with a=-2, b=-8, c=-5: -3.224744871391589 -0.7752551286084111 -3.2247448713915894

Root 1 with a=-2, b=-8, c=-3: -3.58113883008419 -0.41886116991581024 -3.5811388300841895

Root 1 with a=-2, b=-8, c=-1: -3.8708286933869704 -0.12917130661302934 -3.870828693386971

Root 1 with a=-2, b=-7, c=-6: -2.0 -1.5 -1.9999999999999991

Root 1 with a=-2, b=-7, c=-5: -2.5 -1.0 -2.4999999999999996

Root 1 with a=-2, b=-7, c=-3: -3.0 -0.5 -2.9999999999999996

Root 1 with a=-2, b=-5, c=-3: -1.5 -1.0 -1.4999999999999998

Root 1 with a=-2, b=-5, c=-1: -2.2807764064044154 -0.21922359359558485 -2.280776406404415

Root 1 with a=-1, b=-10, c=-10: -8.872983346207416 -1.127016653792583 -8.872983346207418

Root 1 with a=-1, b=-10, c=-7: -9.242640687119284 -0.7573593128807152 -9.242640687119286

Root 1 with a=-1, b=-10, c=-4: -9.582575694955839 -0.41742430504416017 -5.0

Root 1 with a=-1, b=-10, c=-3: -9.69041575982343 -0.30958424017657027 -0.3095842401765705

Root 1 with a=-1, b=-10, c=-2: -9.79583152331272 -0.2041684766872809 -0.20416847668728044

Root 1 with a=-1, b=-10, c=-1: -9.898979485566356 -0.10102051443364424 -0.1010205144336438

Root 1 with a=-1, b=-10, c=0: -10.0 0.0 3.944304526105059e-30

Root 1 with a=-1, b=-9, c=-10: -7.701562118716424 -1.2984378812835757 -7.701562118716425

Root 1 with a=-1, b=-9, c=-5: -8.405124837953327 -0.594875162046673 -8.405124837953329

Root 1 with a=-1, b=-9, c=-2: -8.772001872658766 -0.2279981273412348 -0.2279981273412344

Root 1 with a=-1, b=-9, c=-1: -8.887482193696062 -0.1125178063039387 -0.11251780630393897

Root 1 with a=-1, b=-9, c=0: -9.0 0.0 2.8398992587956425e-29

Root 1 with a=-1, b=-8, c=-6: -7.16227766016838 -0.8377223398316205 -7.162277660168379

Root 1 with a=-1, b=-8, c=-3: -7.60555127546399 -0.3944487245360109 -4.0

Root 1 with a=-1, b=-8, c=-2: -7.741657386773941 -0.25834261322605867 -0.2583426132260586

Root 1 with a=-1, b=-8, c=-1: -7.872983346207417 -0.12701665379258298 -0.12701665379258312

Root 1 with a=-1, b=-8, c=0: -8.0 0.0 3.218552493301728e-28

Root 1 with a=-1, b=-7, c=-10: -5.0 -2.0 -4.999999999999999

Root 1 with a=-1, b=-7, c=-9: -5.302775637731995 -1.6972243622680054 -5.302775637731994

Root 1 with a=-1, b=-7, c=-2: -6.701562118716424 -0.29843788128357573 -0.2984378812835756

Root 1 with a=-1, b=-7, c=-1: -6.854101966249685 -0.1458980337503153 -0.14589803375031546

Root 1 with a=-1, b=-7, c=0: -7.0 0.0 7.169167906648555e-27

Root 1 with a=-1, b=-6, c=-6: -4.732050807568877 -1.2679491924311228 -4.732050807568878

Root 1 with a=-1, b=-6, c=-2: -5.645751311064591 -0.3542486889354093 -3.0

Root 1 with a=-1, b=-6, c=-1: -5.82842712474619 -0.1715728752538097 -0.17157287525380988

Root 1 with a=-1, b=-6, c=0: -6.0 0.0 4.493351716138883e-25

Root 1 with a=-1, b=-5, c=-5: -3.618033988749895 -1.381966011250105 -3.6180339887498953

Root 1 with a=-1, b=-5, c=-1: -4.7912878474779195 -0.20871215252208009 -0.20871215252208

Root 1 with a=-1, b=-5, c=0: -5.0 0.0 1.4431070514782548e-22

Root 1 with a=-1, b=-4, c=-2: -3.414213562373095 -0.5857864376269049 -3.4142135623730954

Root 1 with a=-1, b=-4, c=-1: -3.732050807568877 -0.2679491924311228 -2.0

Root 1 with a=-1, b=-3, c=0: -3.0 0.0 4.658949397362878e-25

Root 1 with a=-1, b=-2, c=0: -2.0 0.0 -1.0

Root 1 with a=3, b=5, c=1: -0.2324081207560018 -0.2324081207560017 -0.23240812075600176

Root 1 with a=3, b=7, c=2: -0.3333333333333333 -0.33333333333333326 -0.33333333333333337

Root 1 with a=3, b=7, c=4: -1.0 -0.999999999999999 -0.9999999999999994

Root 1 with a=3, b=8, c=1: -0.13148290817867028 -0.13148290817867014 -0.13148290817867023

Root 1 with a=3, b=8, c=2: -0.2792407799438735 -0.2792407799438734 -0.2792407799438736

Root 1 with a=3, b=9, c=2: -0.24169426078820835 -0.24169426078820822 -0.2416942607882084

Root 1 with a=3, b=10, c=5: -0.6125741132772068 -0.6125741132772069 -0.612574113277207

Root 1 with a=3, b=10, c=6: -0.7847495629784698 -0.7847495629784697 -0.7847495629784699

Root 1 with a=3, b=10, c=7: -1.0 -0.9999999999999999 -1.0000000000000002

Root 1 with a=3, b=10, c=8: -1.3333333333333333 -1.3333333333333326 -1.3333333333333337

Root 1 with a=5, b=7, c=1: -0.16148351928654964 -0.16148351928654958 -0.1614835192865496

Root 1 with a=5, b=8, c=1: -0.13667504192892005 -0.13667504192892 -0.13667504192892002

Root 1 with a=5, b=8, c=2: -0.31010205144336445 -0.31010205144336433 -0.3101020514433644

Root 1 with a=5, b=8, c=3: -0.6 -0.5999999999999996 -0.6000000000000001

Root 1 with a=5, b=9, c=1: -0.1189750324093346 -0.11897503240933449 -0.11897503240933456

Root 1 with a=5, b=9, c=4: -0.8 -0.7999999999999999 -0.8000000000000004

Root 1 with a=5, b=10, c=1: -0.10557280900008408 -0.10557280900008414 -0.10557280900008413

Root 1 with a=6, b=5, c=1: -0.3333333333333333 -0.33333333333333315 -0.3333333333333332

Root 1 with a=6, b=6, c=1: -0.21132486540518713 -0.21132486540518708 -0.2113248654051871

Root 1 with a=6, b=7, c=1: -0.16666666666666666 -0.16666666666666663 -0.16666666666666669

Root 1 with a=6, b=7, c=2: -0.5 -0.4999999999999995 -0.4999999999999999

Root 1 with a=6, b=8, c=1: -0.13962038997193674 -0.1396203899719367 -0.13962038997193676

Root 1 with a=6, b=9, c=1: -0.12084713039410418 -0.12084713039410411 -0.12084713039410419

Root 1 with a=6, b=10, c=2: -0.2324081207560018 -0.2324081207560017 -0.23240812075600176

Root 1 with a=6, b=10, c=4: -0.6666666666666666 -0.6666666666666663 -0.6666666666666665

Root 1 with a=7, b=9, c=1: -0.12284929362282014 -0.12284929362282004 -0.12284929362282011

Root 1 with a=7, b=9, c=2: -0.2857142857142857 -0.28571428571428564 -0.28571428571428575

Root 1 with a=7, b=10, c=1: -0.1081941875543879 -0.10819418755438781 -0.10819418755438784

Root 1 with a=9, b=9, c=2: -0.3333333333333333 -0.33333333333333326 -0.3333333333333334

Root 1 with a=10, b=7, c=1: -0.2 -0.20000000000000004 -0.19999999999999998

Root 1 with a=10, b=8, c=1: -0.15505102572168222 -0.15505102572168217 -0.1550510257216822

Root 1 with a=10, b=9, c=2: -0.4 -0.39999999999999997 -0.39999999999999974

2c)

We see that when Float64 arithmetic is used over Float16, such as with the case of

(Float16) Root 1 with a=-10, b=-9, c=-1: -0.7705 -0.1296 -0.77

vs.

(Float64) Root 1 with a=-10, b=-9, c=-1: -0.7701562118716424 -0.12984378812835756 -0.7701562118716423

We observe similar results, but with different degrees of accuracy. In the float16 example, we see that the A value and the C value have a difference of 0.0005, whereas with the Float64 sample, we see that there’s a difference of 0.0000000000000001. However, the interesting part is the comparison of the value of the completing the square value. Between Float16 and Float64, we observe that not only is there a difference of 0.0002…, but there is a consistent difference of 0.6… between the other methods as well.

3a)

* **Data** (Input) – The data in the random artists problem is the directed graph of musical artists
* **Model** (Mathematical model) – The model in the random artists problem is the PageRank model, or the linear system of equations and matrices defined by PageRank, including the out-edge adjacency matrix, the diagonal matrix D, and the system of equations defined by .
* **Algorithm** (Code and arithmetic) – The algorithm used in the random artists problem is the code to solve the system of equations.

3b)

Having two separate people implementing the same program and end up with different results stems from identifying error in the data, the model, and the algorithm.

Considering the problem, and under the assumption that the data is the same between the two implementations, the model used by both programs solve the same linear system of equations, and barring that the differences between the two rankings is the result of any bugs, we can then look to the algorithm to identify the issue. Three examples that I can think of are:

* Rounding issues – Despite Julia’s 10e-16 precision, we know that Julia is not immune from issues with working with floating-point numbers. There may have been differences with the way that both programs use values to solve the equation, which may have caused the
* Preprocessing Differences – If one program preprocessed the data (normalization, scaling, etc.,) prior to solving the system of equations, versus if the other program didn’t preprocess or preprocessed the data differently, could lead to implications in the results of the ranking.
* Tie handling – The method of handling ties could also impact the resultant ranking. One algorithm could break ties using alphabetical ranking, vs the other could break ties based on the list-order priority.

3c)

Having two separate people implementing the same program and end up with different results stems from identifying error in the data, the model, and the algorithm.

Considering the problem, and under the assumption that the data is the same between the two implementations and barring that the differences between the two rankings is the result of any bugs, we can then look to the model and the algorithm to identify the issue. Three examples that I can think of are:

* Model issue: Variability of Random Sampling – Because Monte Carlo processes are reliant on random sampling to estimate their results, the randomness of the model inherently introduces differences. If the scores for the 5th and 6th artists are close, the differences due to randomness could interfere with the ordering.
* Algorithm issue: Number of Trials affecting Convergence – Monte Carlo simulations grow more accurate with the number of trials that take place. The accuracy scores may not have converged fully if enough trials were not used, which leads to variability in the rankings.
* Algorithm issue: Seed differences – If the two programs are not using the same seed for the random operations, this directly affects the generation of random numbers that would be generated by the code.

4)

Julia code to solve a Linear System of equation for x given matrices A and b, and save b to a file

## import packages

using Random

using LinearAlgebra

using DelimitedFiles

## Function to create a square matrix of fractions

function ijmat(n::Integer)

A = zeros(n,n)

for j=1:n

for i=1:n

A[i,j] = 1/(i+j-1)

end

end

return A

end

## Function to generate a vector of random values

function randVec(n::Integer)

b = zeros(n,1)

for i=1:n

b[i, 1] = randn()

end

return b

end

## solve linear system

A = ijmat(10)

b = randVec(10)

x = A \ b

## Write b to a files

writedlm("b-matrix.csv", b, ",")

Python function to read the file containing b, create the A matrix, and solve for x

## Import statements

import numpy as np

## Function to create A matrix

def ijmat(n:int):

a = np.zeros((n, n))

for j in range(0, n):

for i in range(0, n):

a[i][j] = 1/(i+j+1)

return a

## Read Matrix from files

b = np.loadtxt("b-matrix.csv", delimiter=',')

a = ijmat(10)

## Solve Linear System of Equations

x = np.linalg.solve(a,b)

print("{}x{} Matrix[{}]".format(x.shape[0], 1, type(x[0])))

print(x)

**Results:**

|  |  |
| --- | --- |
| Julia | Python |
| [1.1679478871319711e7;  -9.922291147728866e8; 2.0880921003386864e10;  -1.8812416864386414e11; 8.91056570604955e11;  -2.435858305880686e12; 3.9783045458550254e12;  -3.830009203030495e12; 2.0042902293131746e12;  -4.395674439527591e11;] | [1.16796042e+07  -9.92239980e+08  2.08811531e+10  -1.88126284e+11  8.91066683e+11  -2.43588616e+12  3.97835031e+12  -3.83005349e+12  2.00431351e+12  -4.39572568e+11] |

We see that therefor the most part, both languages solve the system of equations to a similar accuracy, however there are some differences between the two, possibly due to the way that the files are read into python and stored. When looking at the b matrix in python we get the following values:

[-0.33403393 0.17410968 0.6433672 -0.93453359 0.52022645 -1.49928572

-0.18298165 0.22038015 0.8657362 -0.21727673]

Versus the values stored in the file:

[-0.3340339323081479 0.17410968487380635 0.6433672025329462 -0.9345335863585456

0.5202264501717059 -1.4992857198925658 -0.18298164669857492 0.22038014842594486

0.8657362023180324 -0.21727673171628686]

We see that the values encoded in the file have greater significant digits than the values read into python, which would impact rounding and the display of the number.