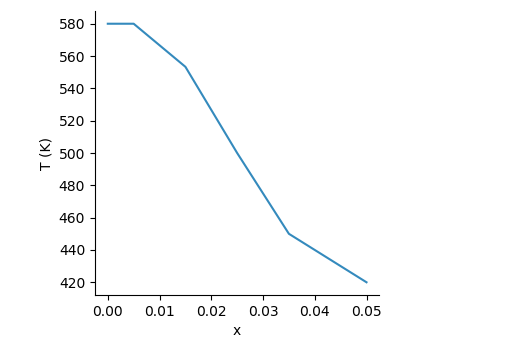
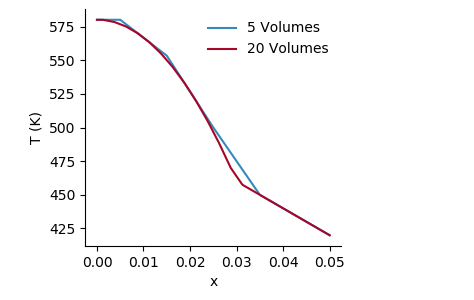
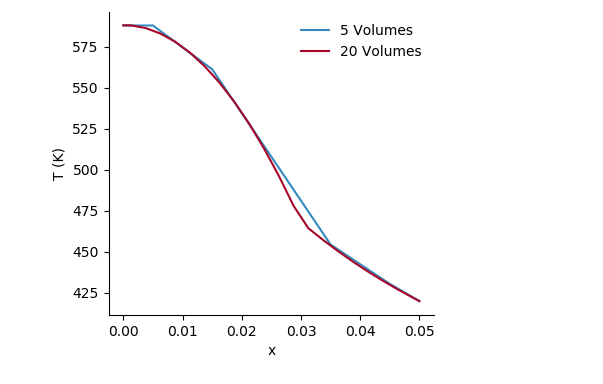
**3b/4a**) A and b matrices match, solved, the temperature profile is:



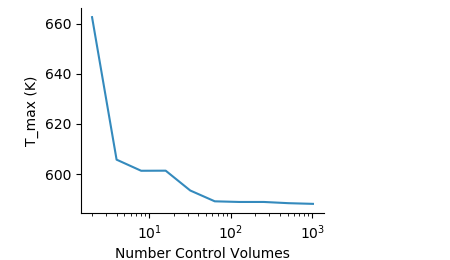
**4b)** More smooth as expected



**5a,b)**



**5c)** about 100 control volumes is grid independent



**Solver Source Code: Julia**

using PyPlot

#-------- Input Parameters --------#

function runsolver(N\_cv)

aw = zeros(N\_cv+2)

ap = zeros(aw)

ae = zeros(aw)

b = zeros(aw)

k = zeros(aw)

kw\_h = zeros(aw) #harmonic mean

ke\_h = zeros(aw)

Sc = zeros(N\_cv) #constant source term

Sp = zeros(N\_cv) #slope of source term, multiplied by temp

h = 1000

Tinf = 300

boundary = 0.03

cv\_x = collect(linspace(0,0.05,N\_cv+1))

cv\_dx = cv\_x[2:end] - cv\_x[1:end-1]

node\_x = [0;(cv\_x[2:end]+cv\_x[1:end-1])/2;cv\_x[end]]

node\_dx = node\_x[2:end] - node\_x[1:end-1]

for i = 1:length(k)

if node\_x[i]<boundary

k[i] = 15

else

k[i] = 137\*exp(25\*node\_x[i]-2)

end

end

for i = 1:length(Sc)

if cv\_x[i]<boundary

Sc[i] = 4E6

Sp[i] = 0

else

Sc[i] = 0

Sp[i] = 0

end

end

#-------- Apply harmonic mean --------#

node\_dx\_minus = cv\_x-node\_x[1:end-1]

node\_dx\_plus = node\_x[2:end]-cv\_x

for i = 1:length(kw\_h)-1

ke\_h[i] = node\_dx[i]\*k[i+1]\*k[i]/(k[i+1]\*node\_dx\_minus[i]+k[i]\*node\_dx\_plus[i])

kw\_h[i+1] = node\_dx[i]\*k[i]\*k[i+1]/(k[i]\*node\_dx\_plus[i]+k[i+1]\*node\_dx\_minus[i])

end

#-------- Assemble the coefficients --------#

# Apply Left BC

aw[1] = 0

ae[1] = ke\_h[1]/node\_dx[1]

ap[1] = ke\_h[1]/node\_dx[1]

b[1] = 0

# Apply Interior Points

for i = 2:N\_cv+1

aw[i] = kw\_h[i]/node\_dx[i-1]

ae[i] = ke\_h[i]/node\_dx[i]

ap[i] = aw[i]+ae[i]-Sp[i-1]\*cv\_dx[i-1]

b[i] = Sc[i-1]\*cv\_dx[i-1]

end

# Apply Right BC

aw[end] = kw\_h[end]/node\_dx[end]

ae[end] = 0

ap[end] = kw\_h[end]/node\_dx[end]+h

b[end] = h\*Tinf

#-------- Assemble the A\_matrix --------#

A = zeros(N\_cv+2,N\_cv+2)

# First Row

A[1,1] = ap[1]

A[1,2] = -ae[1]

# Interior Rows

for i = 2:N\_cv+1 #loop through the interior rows

A[i,i-1] = -aw[i]

A[i,i] = ap[i]

A[i,i+1] = -ae[i]

end

# Last Row

A[end,end-1] = -aw[end]

A[end,end] = ap[end]

#-------- Solve the System --------#

T = A\b

return T,node\_x

end

rc("figure", figsize=(4.5, 2.6))

rc("font", size=10.0)

rc("lines", linewidth=1.5)

rc("lines", markersize=3.0)

rc("legend", frameon=false)

rc("axes.spines", right=false, top=false)

rc("figure.subplot", left=0.18, bottom=0.18, top=0.97, right=0.72)

rc("axes", color\_cycle=["348ABD", "A60628", "009E73", "7A68A6", "D55E00", "CC79A7"])

# PyPlot.close("all")

# PyPlot.figure("5")

# N\_cv = [5,20]#linspace(5,100,7)

# N\_cv = round.(Int,N\_cv)

# for i = 1:length(N\_cv)

# T,node\_x = runsolver(N\_cv[i])

# PyPlot.plot(node\_x,T,label = "$(N\_cv[i]) Volumes")

# PyPlot.pause(0.01)

# end

# PyPlot.xlabel("x")

# PyPlot.ylabel("T (K)")

# PyPlot.legend(loc = "best")

PyPlot.figure("5c")

N\_cv = zeros(10)

for i = 1:length(N\_cv)

N\_cv[i] = 2^i

end

N\_cv2 = round.(Int,N\_cv)

T2 = zeros(length(N\_cv))

for i = 1:length(N\_cv)

# i = 1

T,node\_x = runsolver(N\_cv2[i])

T2[i] = T[1]

println(i)

end

PyPlot.semilogx(N\_cv,T2)

PyPlot.xlabel("Number Control Volumes")

PyPlot.ylabel("T\_max (K)")