galerkin

mike

November 5, 2017

Contents

import numpy as np import matplotlib.pyplot as plt from scipy.linalg import $\text{solve}_{\text{banded}}$ import math import seaborn as sns $\text{sns.set}_{\text{style}}$("darkgrid")

def tridiagsolver(K,F): ud = np.insert(np.diag(K,1), 0, 0) # upper diagonal d = np.diag(K) # main diagonal ld = np.insert(np.diag(K,-1), len(F)-1, 0) # lower diagonal ab = np.matrix([ud, d, ld]) # simplified matrix a = $\text{solve}_{\text{banded}}$((1, 1), ab, F) return a

def psi(j,x, dx): if x > (j+1)*dx or x < (j-1)*dx: return 0 elif x < j*dx: return (x - (j-1)*dx)/dx else: return ((j+1)*dx - x)/dx

def galerkin1d(nx): x = np.linspace(0,1,nx) dx = 1.0/(nx-1) K = np.zeros((nx,nx)) # Stiffness matrix for i in range(nx): if i == 0: K[i,i] = 1 K[i,i+1] = 0 elif i == len(K)-1: K[i,i] = 1 K[i,i-1] = 0 else: K[i,i] = 2/dx K[i,i-1] = -1/dx K[i,i+1] = -1/dx F = np.zeros(nx) # Load vector F [1] = 0 F[1:-1] = (-1.0/dx)*(2*np.cos(x[1:-1]) - np.cos(x[0:-2]) - np.cos(x[2:])) F[-1] = 1 a = tridiagsolver(K,F) # Solve system nxplot = 200 plot_{x} = np.linspace(0,1,nxplot) $\text{phi}_{\text{galerkin}}$ = np.zeros(nxplot) for i in range(len(plot_{x})): # Recombine phi from basis functions for j in range(len(a)): $\text{phi}_{\text{galerkin[i]}}$ += a[j]*psi(j, $\text{plot}_{\text{x[i]}}$, dx) return $\text{phi}_{\text{galerkin}}$

if name == "main": plot_{x} = np.linspace(0,1,200) # points for plotting $\text{phi}_{\text{galerkin5}}$ = galerkin1d(nx=5) $\text{phi}_{\text{galerkin9}}$ = galerkin1d(nx=9) $\text{phi}_{\text{analy}}$ = -np.cos(plot_{x}) + np.cos(1)*plot_{x} + 1

plt.figure(1) plt.plot(plot_{x}, $\text{phi}_{\text{analy}}$, label= "Analytical") plt.plot(plot_{x}, $\text{phi}_{\text{galerkin5}}$, label="Galerkin FE") plt.title("Galerkin FE 5 Nodes") plt.ylabel("ϕ") plt.xlabel("x") plt.legend()

plt.figure(2) plt.plot(plot_{x}, $\text{phi}_{\text{analy}}$, label= "Analytical") plt.plot(plot_{x}, $\text{phi}_{\text{galerkin9}}$, label="Galerkin FE") plt.title("Galerkin FE 9 Nodes") plt.ylabel("ϕ") plt.xlabel("x") plt.legend()

[1]DEFINITION NOT FOUND.

plt.figure(3) plt.plot($plot_x$, np.abs(phi_{analy}-$phi_{galerkin5}$), label= "5 Node num-analytical error") plt.plot($plot_x$, np.abs(phi_{analy}-$phi_{galerkin9}$), label= "9 Node num-analytical error") plt.title("Galerkin FE Error Compared to Analytic Solution") plt.ylabel("Error") plt.xlabel("x") plt.legend() plt.show()