Chapter 8

Boundary Value Problem

Boundary Value Problem (BVP) Example from Boyd - Collocation Method

$$P(x) = \sum a_k x^k = \sum L_k(\hat{x}) f(x_k)$$

Where:

- x_k: Find a_k
- L_k: Find L_k

Boundary conditions:

$$u(0) = 0, \quad u(1) = 1$$

Thus:

$$u = a_0 + a_1 x + a_2 x^2$$

We approximate the solution to a polynomial in the interval $[0, 1] \to [-1, 1]$.
Given equation:

$$(1 + u)\frac{d^2u}{dx^2} - \frac{du}{dx} = 0$$

At midpoint $t = \frac{1}{2}$:

$$a_2 = - \quad \text{(solve for } a_2\text{)}$$

Galerkin Method

$$\frac{\partial}{\partial a_k} \int R^2 \, dx$$

$$\Rightarrow \int R \frac{\partial R}{\partial a_k} = 0$$

Where:

$$J = \int R^2 \, w \, dx$$

J: Residual norm (Least Squares) with different weights w.

Note: Collocation method is easier for nonlinear problems.

BVP Transformation and Numerical Sine in Class

The BVP transforms the problem domain to $[0, 1]$ and then uses appropriate weights to do the Least Squares (LSQ) approximation.

$$\begin{bmatrix} u(x_0) \\ u(x_1) \\ \vdots \\ u(x_N) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$[D][a] = \sum D_{ij} u_j - \text{residual}$$

Numerical Sine in Class

Differential equation:

$$(1+u)\frac{d^2u}{dx^2} + \left(\frac{du}{dx}\right)^2 = 0$$

Boundary conditions:

$$u(0) = 0, \quad u(1) = 1$$

Time t relation:

$$t = c \quad \Rightarrow \quad \text{explicit approximation to solution}$$

$$P(x) = \sum L_k(x) f(x_k)$$

Transformation:

$$t = \frac{t-1}{c-1} = -1, \quad t = 1 \rightarrow \frac{2(c-t)}{t}$$

Jacobian determinant:

$$\det J = 2dx$$

Mapping for $u(t = 0) = 0$ and $u(t = 1) = 1$:

$$L_k(x_j) \Rightarrow [B] \Rightarrow L(x)$$

Matrix representation:

$$\begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} = \begin{pmatrix} u(t) & \text{differential operator} \end{pmatrix}$$

Question:

Which one is it?

Explanation of the Numerical Approach Using Chebyshev Collocation

To solve the nonlinear Boundary Value Problem (BVP) numerically, we use the Chebyshev collocation method. This approach leverages the properties of Chebyshev nodes for better numerical stability and accuracy. The process involves:

1. Chebyshev Nodes Calculation

We compute the Chebyshev nodes x_{cheby} in the domain $[-1, 1]$. These nodes are derived using:

$$x_{\text{cheby}} = \cos\left(\frac{\pi(2k-1)}{2n}\right), \quad k = 1, \ldots, n$$

These nodes are then transformed to the desired interval $[0, 1]$ using:

$$x_{\text{transformed}} = \frac{a+b}{2} + \frac{b-a}{2} \cdot x_{\text{cheby}}$$

2. Differentiation Matrix

The Chebyshev differentiation matrix D is obtained for these nodes. This matrix allows for the computation of derivatives at Chebyshev nodes.

3. Analytical Solution for Comparison

An analytical solution $u(x) = \sqrt{3x+1} - 1$ is defined and evaluated at the transformed Chebyshev nodes to compare against the numerical solution.

4. Initial Guess

We use the analytical solution to create an initial guess u_{initial} for the solver. Boundary conditions are applied to the initial guess:

$$u(0) = 1, \quad u(1) = 0$$

5. Solving the Nonlinear System

We solve the BVP using MATLAB's `fsolve` function, which minimizes the residuals defined by:

$$R = (1+u)\frac{d^2u}{dx^2} + \left(\frac{du}{dx}\right)^2$$

The function `residual_with_boundary` enforces the boundary conditions by fixing $u(0)$ and $u(1)$.

6. Plotting the Results

The numerical solution $u(x)$ is plotted alongside the analytical solution for comparison, illustrating the accuracy of the numerical method.

Detailed Code Walkthrough

Chebyshev Node Transformation

The Chebyshev nodes x_{cheby} are transformed to the interval $[0, 1]$ for this problem domain.

Initial Guess Setup

The initial guess u_{initial} is adjusted to match the boundary conditions $u(0) = 1$ and $u(1) = 0$.

Residual Function

The function `residual_with_boundary` defines the nonlinear residual R while ensuring the boundary conditions are met by setting:

$$\text{res}(1) = u(0) - 1, \quad \text{res}(n) = u(1) - 0$$

Solver Configuration

The nonlinear system is solved using `fsolve` with options set to display iteration steps for convergence monitoring.

Explanation of Boundary Conditions Enforcement

The boundary conditions $u(0) = 1$ and $u(1) = 0$ are crucial for ensuring that the numerical solution adheres to the physical requirements of the problem. These conditions are imposed directly within the residual function, keeping the first and last elements of u fixed to enforce $u(0)$ and $u(1)$.