1D Simulation to Semilinear Poisson Problem

Bachelor Thesis Presentation

SAN RADY

Supervisor: Dr. SEAM Ngonn

Faculty of Science
Royal University of Phnom Penh

Academic Year 2025–2026

Outline

1. Introduction
2. Approximation of Derivatives
3. Approximation of 1D Semilinear Problem
4. Numerical Simulation
5. Conlusion

Introduction

In this thesis, we solve the 1D semilinear Poisson problem:

Problem to Solve

$$-u''(x) = f(u(x)), \quad x \in (a, b),$$
with **Dirichlet boundary condition** $u(a) = g_0,\ u(b) = g_1$

Our approach:

- Discretize the problem using the **finite difference method**
- Solve the problem **nonlinear system** using the **Newton-Raphson method**, by computing the residual F and Jacobian J

Approximation of Derivatives

Forward Difference

forward difference quotient of u at x is defined by:

$$D_h^+ u(x) = \frac{u(x+h) - u(x)}{h}.$$

Backward Difference

backward difference quotient of u at x is defined by:

$$D_h^- u(x) = \frac{u(x) - u(x-h)}{h}.$$

Central Difference

central difference quotient of u at x is defined by:

$$D_h^0 u(x) = \frac{u(x+h) - u(x-h)}{2h}.$$

Three-Point Formula

Definition

three-point difference quotient of u at x is defined by:

$$D_h^2 u(x) = \frac{u(x+h) - 2\,u(x) + u(x-h)}{h^2}.$$

Continuous Problem

Let $[a, b]$ be an interval of $\mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$ be a given function. We consider the **non-homogeneous problem**:

$$-u''(x) = f(u(x)), \quad a < x < b$$

In this thesis, we take $[a, b]$ and study the **1D semilinear Poisson problem**:

$$-u''(x) = f(u(x)), \quad x \in (a, b)$$

with **Dirichlet boundary conditions**:

$$u(a) = g_0, \qquad u(b) = g_1$$

Mesh Partition

Let $[a, b]$ be divided into n equal subintervals with step size

$$\Delta x = \frac{b-a}{n}.$$

The grid points are defined by

$$x_i = a + i\,\Delta x, \quad i = 0, 1, \ldots, n.$$

and the subintervals are

$$I_i = [x_i, x_{i+1}].$$

Discrete Scheme

Denote $u_i \approx u(x_i)$ for $i = 0, 1, \dots, n$ Using the three-point formula from Chapter 2:

$$u''(x_i) \approx \frac{u_{i-1} - 2u_i + u_{i+1}}{(\Delta x)^2}$$

The discrete scheme at interior nodes is obtained by finite differences:

Discrete Scheme

$$-\frac{1}{(\Delta x)^2}u_{i-1} + \frac{2}{(\Delta x)^2}u_i - \frac{1}{(\Delta x)^2}u_{i+1} - f(u_i) = 0, \quad i = 1, \dots, n-1.$$

Nonlinear System of Equations

Using the finite difference approximation, we obtain the following **nonlinear system**:

$$\begin{cases} u_0 - g_0 = 0, \\ -\dfrac{1}{(\Delta x)^2} u_{i-1} + \dfrac{2}{(\Delta x)^2} u_i - \dfrac{1}{(\Delta x)^2} u_{i+1} - f(u_i) = 0, \quad i = 1, 2, \dots, n-1, \\ u_n - g_1 = 0. \end{cases}$$

We obtain a nonlinear system of $n+1$ equations for the $n+1$ unknowns, where $\mathbf{u}$ is the vector of unknowns

$$\mathbf{u} = (u_0, u_1, \dots, u_n)^T.$$

Nonlinear System of Equations

The system is **nonlinear** and cannot be solved directly.
We seek $\mathbf{u}$ such that $F(\mathbf{u}) = \mathbf{0}$, where:

$$F(\mathbf{u}) = \begin{pmatrix} F_0(\mathbf{u}) \\ F_1(\mathbf{u}) \\ \vdots \\ F_{n-1}(\mathbf{u}) \\ F_n(\mathbf{u}) \end{pmatrix} = \begin{pmatrix} u_0 - g_0 \\ -\frac{1}{(\Delta x)^2} u_0 + \frac{2}{(\Delta x)^2} u_1 - \frac{1}{(\Delta x)^2} u_2 - f(u_1) \\ \vdots \\ -\frac{1}{(\Delta x)^2} u_{n-2} + \frac{2}{(\Delta x)^2} u_{n-1} - \frac{1}{(\Delta x)^2} u_n - f(u_{n-1}) \\ u_n - g_1 \end{pmatrix}$$

This nonlinear problem is solved by the **Newton-Raphson algorithm**.

Newton-Raphson Algorithm

We solve the nonlinear system $F(\mathbf{u}) = \mathbf{0}$ using an iterative Newton-Raphson method.

Iteration Formula

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - J(\mathbf{u}^{(k)})^{-1} F(\mathbf{u}^{(k)}), \qquad k = 0, 1, 2, ...$$

Algorithm Steps

1. Start with an initial guess $\mathbf{u}^{(0)}$
2. Compute $F(\mathbf{u}^{(k)})$
3. Compute Jacobian matrix $J(\mathbf{u}^{(k)})$
4. Solve $J(\mathbf{u}^{(k)})\, \Delta\mathbf{u}^{(k)} = -F(\mathbf{u}^{(k)})$
5. Update the solution $\mathbf{u}^{(k+1)}$

Stopping Criterion

$$\|\Delta\mathbf{u}^{(k)}\|_\infty < \varepsilon$$

$\varepsilon > 0$ is a prescribed tolerance

The Jacobian Matrix J

Definition

Given a vector-valued function $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^m$, the Jacobian matrix element is defined by:

$$J_{i,j} = \frac{\partial F_i}{\partial u_j}, \quad i, j = 0, 1, \ldots, n$$

The Jacobian $J \in \mathbb{R}^{(n+1)\times(n+1)}$ is :

$$J = \begin{pmatrix} \frac{\partial F_0}{\partial u_0} & \frac{\partial F_0}{\partial u_1} & \frac{\partial F_0}{\partial u_2} & \cdots & \frac{\partial F_0}{\partial u_{n-1}} & \frac{\partial F_0}{\partial u_n} \\ \frac{\partial F_1}{\partial u_0} & \frac{\partial F_1}{\partial u_1} & \frac{\partial F_1}{\partial u_2} & \cdots & \frac{\partial F_1}{\partial u_{n-1}} & \frac{\partial F_1}{\partial u_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \frac{\partial F_{n-1}}{\partial u_0} & \frac{\partial F_{n-1}}{\partial u_1} & \cdots & \frac{\partial F_{n-1}}{\partial u_{n-2}} & \frac{\partial F_{n-1}}{\partial u_{n-1}} & \frac{\partial F_{n-1}}{\partial u_n} \\ \frac{\partial F_n}{\partial u_0} & \frac{\partial F_n}{\partial u_1} & \frac{\partial F_n}{\partial u_2} & \cdots & \frac{\partial F_n}{\partial u_{n-1}} & \frac{\partial F_n}{\partial u_n} \end{pmatrix}$$

Jacobian Matrix — Boundary Derivatives

Boundary node derivatives $(i = 0, n)$:

Differentiating the boundary residuals directly gives:

$$\frac{\partial F_0}{\partial u_j} = \frac{\partial}{\partial u_j}(u_0 - g_0) = \begin{cases} 1 & j = 0 \\ 0 & j \neq 0 \end{cases},$$

$$\frac{\partial F_n}{\partial u_j} = \frac{\partial}{\partial u_j}(u_n - g_1) = \begin{cases} 1 & j = n \\ 0 & j \neq n \end{cases}$$

Jacobian Matrix — Interior Derivatives

Interior node derivatives $(i = 1, \dots, n-1)$: The interior residual function is:

$$F_i(\mathbf{u}) \;=\; -\frac{1}{(\Delta x)^2}u_{i-1} + \frac{2}{(\Delta x)^2}u_i - \frac{1}{(\Delta x)^2}u_{i+1} - f(u_i).$$

- If $|i-j| = 1$:

$$\frac{\partial F_i}{\partial u_j} = -\frac{1}{(\Delta x)^2}.$$

- If $i = j$:

$$\frac{\partial F_i}{\partial u_j} \;=\; \frac{2}{(\Delta x)^2} - \frac{\partial f(u_i)}{\partial u_i}.$$

- Otherwise $|i-j| > 1$:

$$\frac{\partial F_i}{\partial u_j} \;=\; 0.$$

The Jacobian Matrix

We get the full Jacobian $J \in \mathbb{R}^{(n+1)\times(n+1)}$:

$$J = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} - \frac{\partial f(u_1)}{\partial u_1} & -\frac{1}{(\Delta x)^2} & \cdots & 0 & 0 \\ 0 & -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} - \frac{\partial f(u_2)}{\partial u_2} & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} - \frac{\partial f(u_{n-1})}{\partial u_{n-1}} & -\frac{1}{(\Delta x)^2} \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Example: Case 1 — $f(u(x)) = x(1-x)$

Since $f(u(x)) = x(1-x)$ does not depend on u, we write $f(u(x)) = f(x)$. We consider the **linear problem**:

$$-u''(x) = x(1-x),\ x \in (0,1),$$

The full Jacobian $J \in \mathbb{R}^{(n+1)\times(n+1)}$ becomes:

$$J = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} - \frac{\partial f(x)}{\partial u_1} & -\frac{1}{(\Delta x)^2} & \cdots & 0 & 0 \\ 0 & -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} - \frac{\partial f(x)}{\partial u_2} & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} - \frac{\partial f(x)}{\partial u_{n-1}} & -\frac{1}{(\Delta x)^2} \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Example: Case 1 — Simplified Jacobian

. **Then we get**:

$$J = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} & -\frac{1}{(\Delta x)^2} & \cdots & 0 & 0 \\ 0 & -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} & -\frac{1}{(\Delta x)^2} \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Example: Case 2 — $f(u(x)) = u(x)(1-u(x))$

We consider the **nonlinear problem**:

$$-u''(x) = u(x)(1-u(x)),\ x \in (0,1),$$

The full Jacobian $J \in \mathbb{R}^{(n+1)\times(n+1)}$ is:

$$J = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} - \frac{\partial f(u_1)}{\partial u_1} & -\frac{1}{(\Delta x)^2} & \cdots & 0 & 0 \\ 0 & -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} - \frac{\partial f(u_2)}{\partial u_2} & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} - \frac{\partial f(u_{n-1})}{\partial u_{n-1}} & -\frac{1}{(\Delta x)^2} \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Example: Case 2 — Simplified Jacobian

Then we get:

$$J = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} - (1 - 2u_1) & -\frac{1}{(\Delta x)^2} & \cdots & 0 & 0 \\ 0 & -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} - (1 - 2u_2) & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -\frac{1}{(\Delta x)^2} & \frac{2}{(\Delta x)^2} - (1 - 2u_{n-1}) & -\frac{1}{(\Delta x)^2} \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Numerical Simulation — Case 1 : $f(u(x)) = x(1-x)$

Problem

$$\begin{cases} -u''(x) = x(1-x), & x \in (0,1), \\ u(0) = 0, \quad u(1) = 0 \end{cases}$$

We solve this problem numerically for $n = 5, 20, 100, 500$ mesh points

Case 1 : $f\left(u(x)\right) = x(1-x)$

Figure: $n = 5$

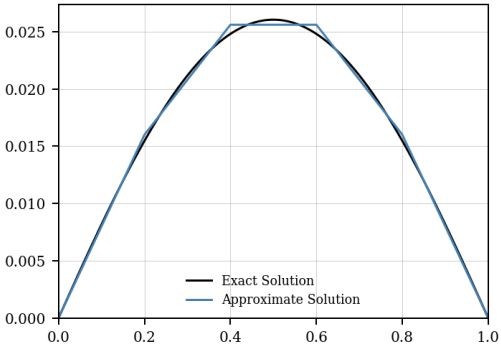
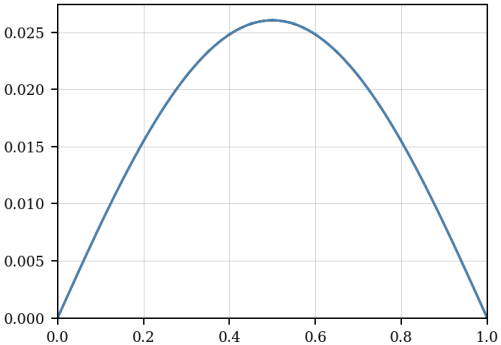


Figure: $n = 20$



Case 1 : $f\left(u(x)\right) = x(1-x)$

Figure: $n = 100$

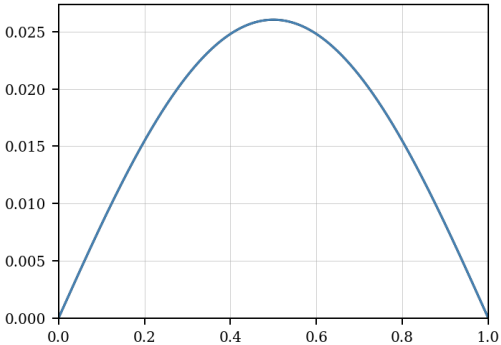
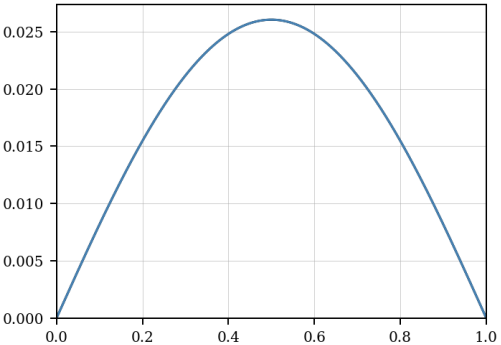


Figure: $n = 500$



Numerical Simulation — Case 2 : $f(u(x)) = u(x)(1-u(x))$

Problem

$$\begin{cases} -u''(x) = u(x)(1-u(x)), & x \in (0,1), \\ u(0) = 0, \quad u(1) = 1 \end{cases}$$

We solve this nonlinear problem numerically for $n = 5, 20, 100, 500$ mesh points using Newton-Raphson.

Case 2 : $f(u(x)) = u(x)(1 - u(x))$

Figure: $n = 5$

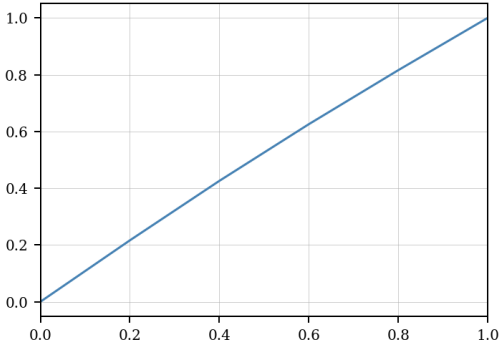
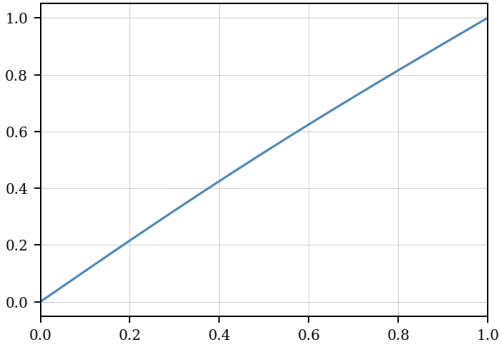


Figure: $n = 20$



Case 2 : $f(u(x)) = u(x)(1-u(x))$

Figure: $n = 100$

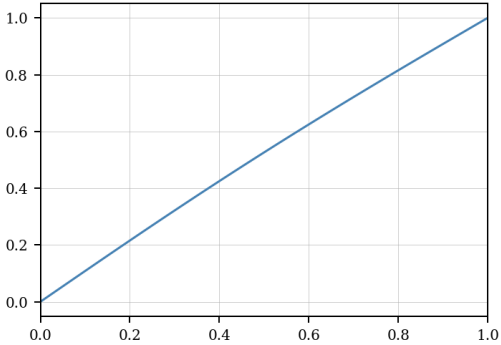
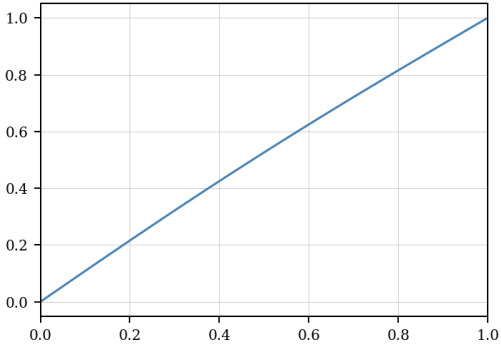


Figure: $n = 500$



Conclusion

In Summary

For

$$-u''(x) = f(u(x))$$

with Dirichlet boundary conditions, Newton-Raphson solves both linear and nonlinear cases.

Case 1 — Linear

- $f(u) = x(1-x)$
- Solved **directly**
- Matches exact solution

Case 2 — Nonlinear

- $f(u) = u(1-u)$
- Solved using the **Newton-Raphson**
- Converges, monotone profile

Thank You

Questions & Discussion