

Projekt 4: solving nonlinear PDEs

Goal of the project is to employ Newton’s method for a large nonlinear system of equations. Suitable stopping criteria and techniques for finding good starting values reduce the number of Newton steps (and thus the computing time).

Let $\Omega = [0, 1]^2$ und $\Gamma = \partial\Omega$. We consider the equation

$$-\Delta u + u^3 = f(x) \quad \text{for } x \in \Omega \tag{1}$$

$$u = 0 \quad \text{on } \Gamma \tag{2}$$

Here, $\Delta u := (\partial_x^2 + \partial_y^2)u$.

To approximate the solution u , we employ the *finite difference method*, where derivatives are approximated by difference quotients. For example, the first derivative $g'(x)$ can be approximated by

$$g'(x) \approx \frac{g(x+h) - g(x)}{h} \quad \text{or} \quad g'(x) \approx \frac{g(x) - g(x-h)}{h} \tag{3}$$

1. Think of a method to approximate by difference quotients the second derivative $g''(x)$. Based on your approximation of the second derivative, design an approximation $\Delta_h u(x, y)$ for $\Delta u(x, y) = \partial_x^2 u(x, y) + \partial_y^2 u(x, y)$ that only uses the values $u(x, y)$, $u(x \pm h, y \pm h)$.
2. Let $h = \frac{1}{2^p}$ for $p \in \mathbb{N}$ and $n = 1/h - 1$. The points $\{(ih, 1 - jh) \mid 0 \leq i, j \leq n + 1\}$ are in $[0, 1]^2$. We seek approximations to the values $u(ih, 1 - jh)$. Obviously, (2) yields the values for the cases $i \in \{0, n + 1\}$ or $j \in \{0, n + 1\}$. Hence, only approximations for the cases $(i, j) \in \{1, \dots, n\}$ have to be found. The total number of unknown values is therefore $N = n^2$.

In order to avoid “double indices” we introduce with $\nu : \{(i, j) \mid i = 1, \dots, n\} \rightarrow \{1, \dots, N\}$ a suitable numbering of the points $\{(ih, 1 - jh) \mid 1 \leq i, j \leq n\}$. The solution u of (1) in $(ih, 1 - jh)$ is approximated by $\bar{u}_{\nu(i,j)} \approx u(ih, 1 - jh)$. The vector $\bar{u} \in \mathbb{R}^N$ is obtained as the solution of the discrete equation

$$F(\bar{u}) = 0, \tag{4}$$

$$F : \mathbb{R}^N \rightarrow \mathbb{R}^N, \bar{u} \mapsto L_h \bar{u} + \bar{u}^3 - \bar{f}, \tag{5}$$

where L_h is the matrix that realizes the application of $-\Delta_h$ in each point, and $\bar{u}^3_{\nu(i,j)} := (\bar{u}_{\nu(i,j)})^3$ means to raise to the third power componentwise. Analogously, \bar{f} is the vector with entries $\bar{f}_{\nu(i,j)} = f(ih, 1 - jh)$.

- a) $(L_h \bar{u})_{\nu(i,j)}$ represents the approximation of $-\Delta u$ at $(ih, 1 - jh)$. What is $(L_h \bar{u})_{\nu(i,j)}$ for $i, j \in \{1, \dots, n\}$? Note that the cases $i \in \{1, n\}$ and $j \in \{1, n\}$ are special.
- b) The concrete representation of L_h depends on the chosen numbering of the unknowns. The *lexicographic numbering* orders them from left to right and top to bottom, see Fig. 1. Write down L_h with the lexicographic numbering. How many entries are non-zero depending on N ? What is the bandwidth of the matrix?
- c) Formulate Newton’s method for (4).
- d) Compute the derivative $DF(\bar{u}) \in \mathbb{R}^{N \times N}$ of F .

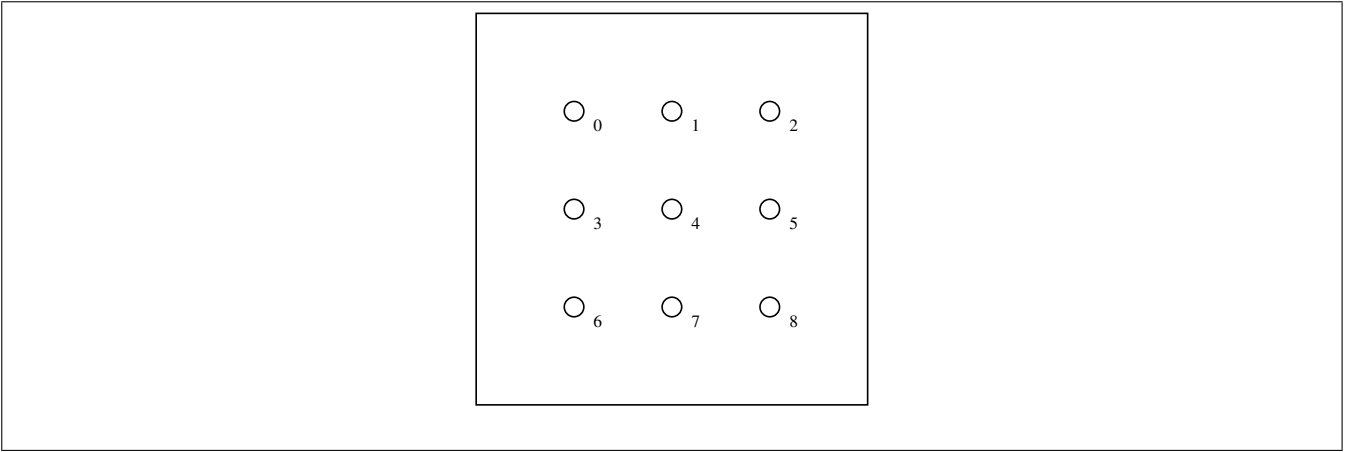


Figure 1: Example of a grid for $h = 1/4$ with lexicographic numbering

3. Write a MATLAB program¹ that solves the discrete problem (4) with Newton's method. Make sure that you set up the matrix L_h as a **sparse** matrix, e.g., by $L = \text{sparse}(I, J, A)$, where I, J, A are vectors that contain the indices i, j , and the values L_{ij} .

A step of Newton's method $\bar{u}^{(n+1)} := \bar{u}^{(n)} - (DF(\bar{u}))^{-1}F(\bar{u})$ consists of:

1. Compute $F(\bar{u}^{(n)})$ and $DF(\bar{u}^{(n)})$,
2. solve the linear system $DF(\bar{u}^{(n)})\bar{\delta} = -F(\bar{u}^{(n)})$
3. $\bar{u}^{(n+1)} := \bar{u}^{(n)} + \bar{\delta}$.

Document your program. To test it, you may use Problem 4.

4. The function $u(x, y) = \sin(\pi x) \sin(\pi y)$ solves (1) for $f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y) + \sin^3(\pi x) \sin^3(\pi y)$. The error of the discrete solution \bar{u}_h is defined as

$$e(\bar{u}_h) := \max_{i,j=1,\dots,n} |u(ih, 1 - jh) - (\bar{u}_h)_{\nu(i,j)}| \quad (6)$$

- a) Modify your program so that it also computes the error if the exact solution is provided.
 - b) Experiment with $h = 2^{-4}, 2^{-5}, \dots$ and plot the error in doubly logarithmic scale versus h . Make sure that the error of Newton's method does not dominate the discretization error of the finite difference method (i.e., stop Newton's method when the stopping criterion $\|\bar{u}^{(n+1)} - \bar{u}^{(n)}\|_\infty \leq \varepsilon$ for $\varepsilon = 10^{-10}$ is satisfied). What convergence behavior do you observe?
5. The convergence behavior of Newton's method depends on the initial value. To see this, consider the case $h = 1/256$ in Problem 4. Plot (semilogarithmically) the error $\|\bar{u}^{(n+1)} - \bar{u}^{(n)}\|_\infty$ versus the iteration index $n = 1, \dots, 15$. As starting vector consider the following 3 choices:
1. the zero vector
 2. a random vector with entries between 0 and 10
 3. a random vector with entries between 0 and 100

Compare.

¹You may write the program also in C/C++ —a direct solver for linear systems will then be provided

6. Let $\varepsilon > 0$. Several stopping conditions are possible:
1. Condition 1: $\|\bar{u}^{(n+1)} - \bar{u}^{(n)}\|_\infty < \varepsilon$
 2. Condition 2: $\|F(\bar{u}^{(n+1)})\|_\infty < \varepsilon$
 3. Condition 3: $\|DF(\bar{u}^{(n)})^{-1}F(\bar{u}^{(n+1)})\|_\infty < \varepsilon$
- a) Modify your program to include these three conditions.
 - b) Plot (semilogarithmically) for the same problem (e.g., $h = 1/128$ or $h = 1/256$) the values of the stopping condition as a function of the iteration index n .
 - c) Discuss the three conditions regarding their computational cost and the impact on the number of Newton steps. What is actually estimated by these three conditions?
7. A possibility to generate starting values is to generate them hierarchically. Let h be the step size of the mesh on which the solution is sought. The starting value for Newton's method is taken as the interpolation of the solution to the problem with mesh size $2h$. In turn, the starting value for this problem can be obtained as the interpolation of a problem for mesh size $4h$ etc.
- a) Write a function that yields the linear interpolation on a mesh of mesh size h from values of a mesh with mesh size $h/2$. For that, use that the solution is zero on Γ . Fig. 2 illustrates the procedure.
 - b) Implement variants with recursive generation of starting vector. The coarsest grid is the one for $h = 1/4$; the starting vector for Newton's method on the coarsest grid should be any of the three choice of Problem 5. Choose one of the stopping criteria "condition 1" or "condition 3" with $\varepsilon = 10^{-10}$ for all grids.
 - c) Solve the problem for $h = 1/256$ or $h = 1/512$ with the hierarchical construction of the starting vector. How that the hierarchical starting vector generation impact the total run-time and the number of Newton steps on the finest mesh? On the coarse mesh, you can use any of the three choices of starting vectors of Problem 5. Do you see a significant influence of the starting vector on the coarse mesh on the total run-time?

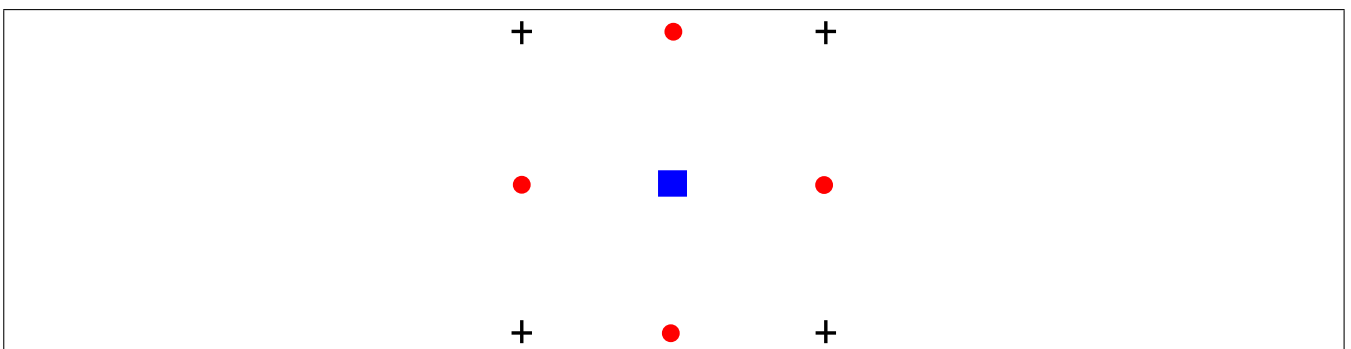


Figure 2: Idea of linear interpolation: The points (+) stand for nodes of the mesh with mesh size h , the nodes (•) and (■) for nodes of the mesh of mesh size $h/2$, that are added. The value at the new nodes (•) is obtained as the average of the 2 nearest nodes of type (+); the value at the new nodes (■) is the average of the 4 nearest nodes of type (+).