# Using Finite Difference Method to investigate the structure of flows in unbounded and confined geometries

Huynh Tuan Kiet Phan, supervised by Dr. Edward Hinton

2023-2024 Vacation Scholarship Program, The University of Melbourne

## Introduction

Many problems in fluid mechanics appear in the form of complex partial differential equations (PDE) which are hard to find the exact solutions, especially the non-linear PDE. Thus, one way to overcome the challenge is to solve it numerically using the finite difference method.

In this poster, we will provide some theoretical basis and solve numerically the biharmonic equation in different 2-dimension boundary conditions using Matlab. Two problems we will present include the lid-driven cavity problem, the stick-slip problem in a rectangular grid.

The Biharmonic equation: $\nabla^4\psi = \dfrac{\partial^4\psi}{\partial x^4} + \dfrac{\partial^4\psi}{\partial y^4} + 2\dfrac{\partial^4\psi}{\partial x^2 y^2} = 0$

## Method/Theoretical basis

The idea of the finite difference method is to divide the rectangular domain into a uniformly spaced m by n grid (the distance between 2 neighbor nodes is h) and try to evaluate the value at each node. By doing so with a small h, we can plot the solution approximately. In this poster, we denote $\psi_{i,j}$ the value at node in row i, column j.

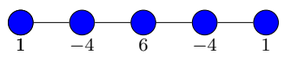
## 1. Using stencils to approximate linear operators

Our first task is to approximate linear operators such as $\dfrac{\partial^4}{\partial x^4}, \dfrac{\partial^4}{\partial y^4}, \dfrac{\partial^4}{\partial x^2 y^2}$.

To approximate the operator $T$, we find some coefficients $\alpha_{i+1,j}, \alpha_{i-1,j}, \cdots$ such that $T(\psi_{i,j}) = \alpha_{i+1,j}\psi_{i+1,j} + \alpha_{i-1,j}\psi_{i-1,j} + \cdots + error(h)$. This task can be done by setting the target error (eg. $O(h^4)$) and using the Taylor Series expansions (possibly in 2 dimensions). For example we have:

$$\frac{\partial^4\psi_{i,j}}{\partial x^4} = \frac{\psi_{i-2,j} - 4\psi_{i-1,j} + 6\psi_{i,j} - 4\psi_{i+1,j} + \psi_{i+2,j}}{h^4} + O(h^2)$$

Based on the coefficients and the positions of surrounding nodes, we can introduce the stencil $[1 \ -4 \ \ 6 \ -4 \ \ 1]$ to approximate $\dfrac{\partial^4}{\partial x^4}$.




Similarly, we can approximate $\dfrac{\partial^4}{\partial y^4}, \dfrac{\partial^4}{\partial x^2 y^2}$ and add together in the corresponding position to obtain the stencil for $\nabla^4$ as below with the error be $O(h^2)$

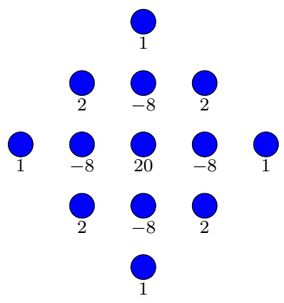



Figure 1. The 13-point stencil for $\nabla^4$.

## 2. Set up the system of equations for different boundary conditions

In the simplest case, when the values and their normal derivatives at boundary nodes are known, our task is to find all the inner nodes given the equations at each node by the above stencils. Hence, it comes naturally that we try to set up the linear systems $Ax = g$, where $x$ is a vector storing all boundary and inner values.

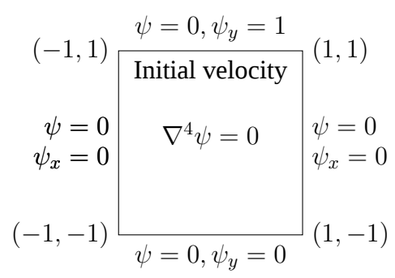



Figure 2. The lid driven cavity problem

However, the equations for nodes in the boundary and those for nodes near boundary and the central will be different. Hence, it will affect the run time of the program if we input coefficients using for loop (as MATLAB will prefer vectorization in stead of for loop in term of run time).
Hence, we use the idea of the tensor product to overcome the challenge:

- We will set up a matrix A such that $Ax$ will return the corresponding derivatives at each node (ie. in the boundary will return the original value, the inner boundary will return the normal derivatives of the boundary, and other nodes $(i, j)$ will return $\nabla^4\psi(i, j)$).
- To do so, we first set up some $m \times m$ matrices called $M_i$ to change every elements in one columns $y$ to $M_i y$, their corresponding derivatives wrt. $y$ by employing the stencils.
- Set up $n \times n$ matrices called $N_i$ the same way as $M$ but with the derivatives wrt. $x$.
- Represents $A = \sum_i N_i \otimes M_i$.

## 3. Solving the linear systems $Ax = b$ using LU factorization with partial pivoting (LUP)

- The target of the LUP is to find the lower and upper triangular matrices L,U and a permutation matrix P such that $PA = LU$.
- The core idea is at each step, we multiply A with $M_i$ in the left ($i = 1 : n-1$) to implement the Gauss-Jordan elimination at each column.
- Note that $M_i$ is easy to inverse just by changing signs of some entries and $M_i$'s are lower triangular.
- In the end we can obtain $L = M_1^{-1} \cdots M_{n-1}^{-1}$, U and P.

The Algorithm works as follow:
Initiation: $U = A, P = I_n$ (suppose $A$ is a square $n \times n$ matrix).

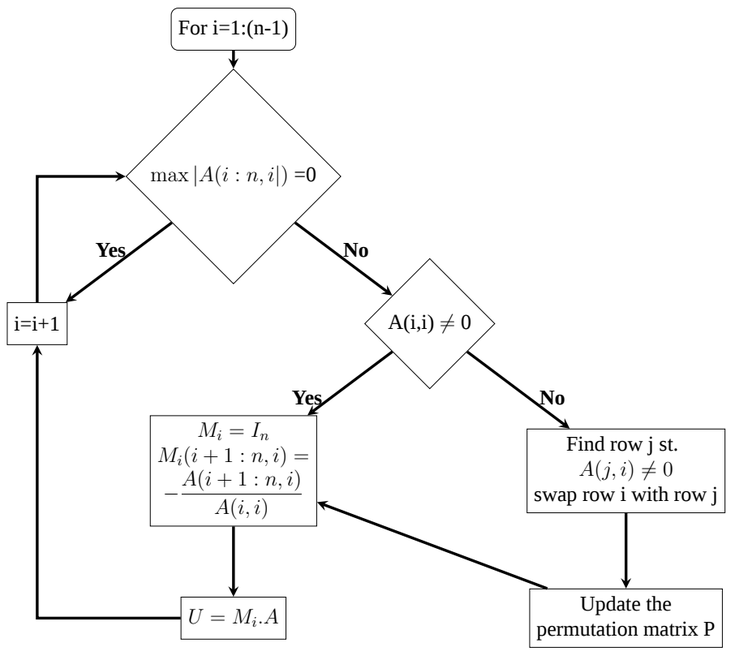



Figure 3. LU decomposition with partial pivoting

After getting $A = P^{-1}LU$, we can turn $Ax = b$ into $Ux = b'$ where $b' = L^{-1}Pb = M_{n-1} \cdots M_1 Pb$ and solve it by back-substitution (as $U$ is upper triangular).

## Numerical results

## The lid-driven cavity problem

- This problem describe a bucket of viscous fluid at the top boundary moving horizontally to cause flow.
- While the top boundary is moving, other sides of the square is stationary.
- The boundary conditions are described as in figure 2.

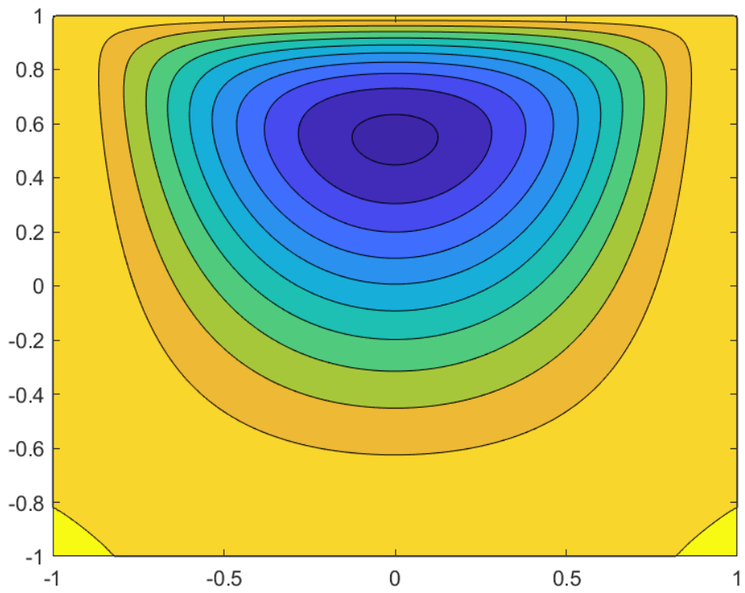



Figure 4. Contour plot of solution of the lid-driven cavity

## The stick-slip problem

- This problem describe a flow in an infinitely long tube with the walls (can think as river), go from very rough ($v = 0$) to very slippy ($\dfrac{dv}{dy} = 0$).
- The flow is made by a background pressure gradient from left to right.

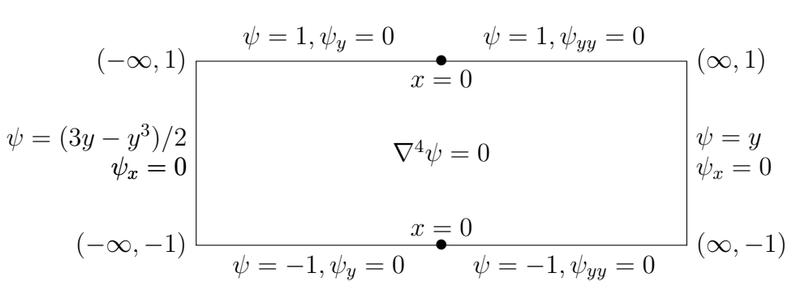



Figure 5. The stick-slip problem

- This problem is more complex than the lid-driven cavity problem as in the top and bottom boundary change from $\psi_y = 0$ to $\psi_{yy} = 0$ when $x$ moving from $-\infty$ to $\infty$.
- Further, the geometry is unbounded in the x-direction.

**Here is our method:**

- To overcome the geometry difficulty, we replace $\infty$ with relatively large number $n$ such as $n = 50, 1000$, plot the solution and consider if the difference is significant.
- For changing boundary conditions, we can split matrices $M_i = M_{i,1} + M_{i,2}$ to return different derivatives at different columns.

Here are the contour plot for $n = 50$ and $n = 100$:

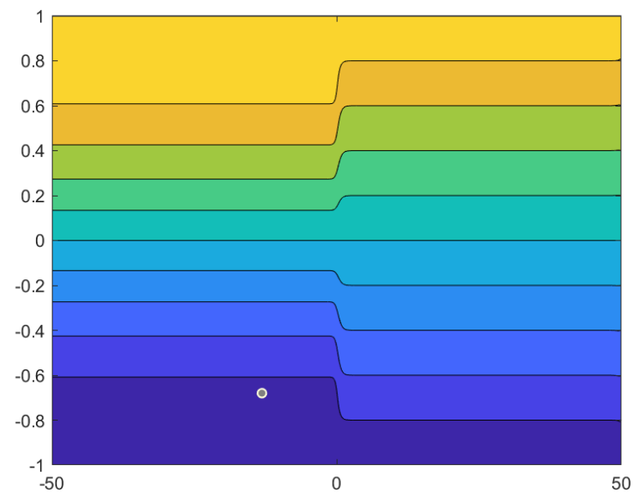



Figure 6. $n = 50$

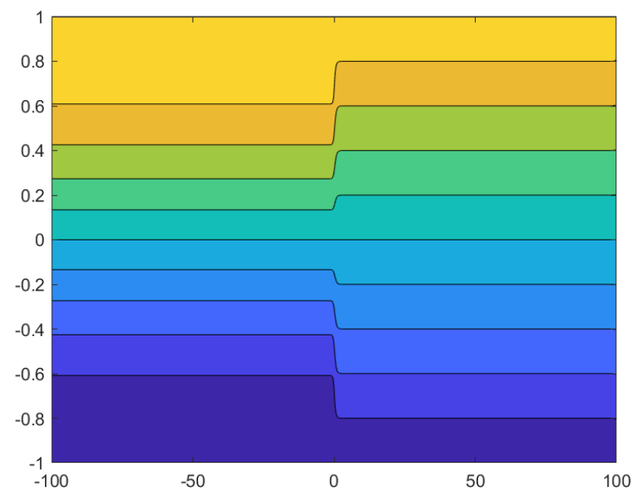



Figure 7. $n = 100$

- Comparing the two plot, we can see the only slight difference occurs at the mid point, where the flow begins to change significantly.
- Hence, we can expect the plot for $n = \infty$ to change more drastically at the mid point $x = 0$.

## Further Discussion

- Instead of using 13-point stencil in figure want, we can try to extend the stencils to a $7 \times 7$ grid for more accusation. However, the symmetry of the matrix $A$ may be destroy so we need to consider careful scaling to preserve the symmetry.
- Instead of using the LU method in figure 3, we can employ iterative methods such as Jacobi and Gauss-Seidel method to improve the runtime. However, we still need to balance the trade of between stability and convergence rate.
- By the same method, we can also create the solver for $\nabla^4\psi = g(x, y)$ for any known function $g$. This can be the basis step for solving non-linear PDE using iterative method. The idea is to come up with the solution $\psi_0$ for the linear problem, which is near to the non-linear one with small parameter, then keep solving $\nabla^4\psi_{n+1} = g(\psi)$ for closer solutions. To account for the convergence rate, the method of under-relaxation might be employed.

## Acknowledgement

- The Vacation scholarship program have offered me the chance to get the taste for research and explore further my interest in mathematics.
- I would like to thank my supervisors, Dr. Edward Hinton and Dr. Douglas Brumley for their patience, useful advice and discussion.

## References

[1] Long Chen.
Programming of finite element methods in matlab.
2018.

[2] John Lambert.
Direct methods for linear system solving.
2019.

[3] Jesse J Taylor West.
Flows of viscoplastic fluids.
2023.