Finite Difference Method

CENG6202: Advanced Computational Methods in Geotechnical Engineering

Yared W. Bekele, PhD

Fall Semester 2019

Contents

Contents

Introduction

The Finite Difference Method (FDM) is one of the most widely used computational methods in engineering.

Several problems are difficult or imposible to solve using analytical or other simple methods for various reasons:

- The governing mathematical equation is nonlinear and can't be linearized without affecting the result.
- The computational domain is complex.
- The boundary conditions are of mixed types.
- The boundary conditions are time-dependent.
- The material property is non-homogeneous or anisotropic.

Introduction

The Finite Difference Method (FDM) is one of the most widely used computational methods in engineering.

Several problems are difficult or imposible to solve using analytical or other simple methods for various reasons:

- The governing mathematical equation is nonlinear and can't be linearized without affecting the result.
- The computational domain is complex.
- The boundary conditions are of mixed types.
- The boundary conditions are time-dependent.
- The material property is non-homogeneous or anisotropic.

The Finite Difference Method (FDM) is one of the most widely used computational methods in engineering.

Several problems are difficult or imposible to solve using analytical or other simple methods for various reasons:

- The governing mathematical equation is nonlinear and can't be linearized without affecting the result.
- The computational domain is complex.
- The boundary conditions are of mixed types.
- The boundary conditions are time-dependent.
- The material property is non-homogeneous or anisotropic.

Introduction

The Finite Difference Method (FDM) is one of the most widely used computational methods in engineering.

Several problems are difficult or imposible to solve using analytical or other simple methods for various reasons:

- The governing mathematical equation is nonlinear and can't be linearized without affecting the result.
- The computational domain is complex.
- The boundary conditions are of mixed types.
- The boundary conditions are time-dependent.
- The material property is non-homogeneous or anisotropic.

Introduction

The Finite Difference Method (FDM) is one of the most widely used computational methods in engineering.

Several problems are difficult or imposible to solve using analytical or other simple methods for various reasons:

- The governing mathematical equation is nonlinear and can't be linearized without affecting the result.
- The computational domain is complex.
- The boundary conditions are of mixed types.
- The boundary conditions are time-dependent.
- The material property is non-homogeneous or anisotropic.

Introduction

The Finite Difference Method (FDM) is one of the most widely used computational methods in engineering.

Several problems are difficult or imposible to solve using analytical or other simple methods for various reasons:

- The governing mathematical equation is nonlinear and can't be linearized without affecting the result.
- The computational domain is complex.
- The boundary conditions are of mixed types.
- The boundary conditions are time-dependent.
- The material property is non-homogeneous or anisotropic.

Introduction

The Finite Difference Method (FDM) is one of the most widely used computational methods in engineering.

Several problems are difficult or imposible to solve using analytical or other simple methods for various reasons:

- The governing mathematical equation is nonlinear and can't be linearized without affecting the result.
- The computational domain is complex.
- The boundary conditions are of mixed types.
- The boundary conditions are time-dependent.
- The material property is non-homogeneous or anisotropic.

FDM is based on approximations that allow replacing differential equations by **finite difference equations**.

In general, solution using the FDM involves three steps:

1. Divide the computational domain into a **grid** of nodes.
2. Approximate the governing differential equation by an equivalent finite difference equation with respect to the grid points.
3. Solve the finite difference equations for the unknowns subject to the boundary and/or initial conditions.

FDM is based on approximations that allow replacing differential equations by **finite difference equations**.

In general, solution using the FDM involves three steps:

1. Divide the computational domain into a **grid** of nodes.
2. Approximate the governing differential equation by an equivalent finite difference equation with respect to the grid points.
3. Solve the finite difference equations for the unknowns subject to the boundary and/or initial conditions.

Introduction

FDM is based on approximations that allow replacing differential equations by **finite difference equations**.

In general, solution using the FDM involves three steps:

1. Divide the computational domain into a **grid** of nodes.
2. Approximate the governing differential equation by an equivalent finite difference equation with respect to the grid points.
3. Solve the finite difference equations for the unknowns subject to the boundary and/or initial conditions.

Introduction

FDM is based on approximations that allow replacing differential equations by **finite difference equations**.

In general, solution using the FDM involves three steps:

1. Divide the computational domain into a **grid** of nodes.
2. Approximate the governing differential equation by an equivalent finite difference equation with respect to the grid points.
3. Solve the finite difference equations for the unknowns subject to the boundary and/or initial conditions.

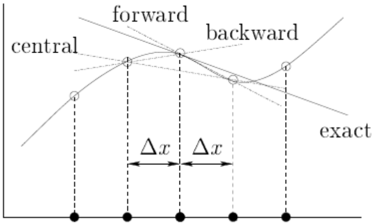
FDM is based on approximations that allow replacing differential equations by **finite difference equations**.

In general, solution using the FDM involves three steps:

1. Divide the computational domain into a **grid** of nodes.
2. Approximate the governing differential equation by an equivalent finite difference equation with respect to the grid points.
3. Solve the finite difference equations for the unknowns subject to the boundary and/or initial conditions.

Contents

Consider a real-valued function $f(x)$. The first derivative of f at $x = a$ may be approximated in different ways for $h = \Delta x$.



Forward Difference Equation:

$$f'(a) \approx \frac{f(a+h) - f(a)}{h}$$

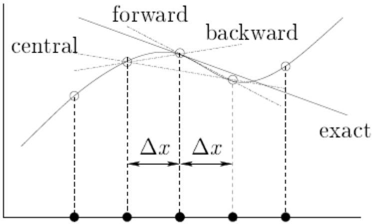
Central Difference Equation:

$$f'(a) \approx \frac{f(a+h) - f(a-h)}{2h}$$

Backward Difference Equation:

$$f'(a) \approx \frac{f(a) - f(a-h)}{h}$$

Consider a real-valued function $f(x)$. The first derivative of f at $x = a$ may be approximated in different ways for $h = \Delta x$.



Forward Difference Equation:

$$f'(a) \approx \frac{f(a+h) - f(a)}{h}$$

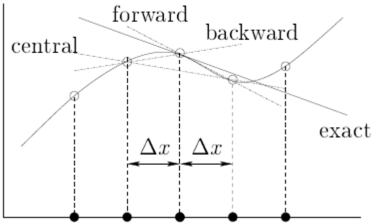
Central Difference Equation:

$$f'(a) \approx \frac{f(a+h) - f(a-h)}{2h}$$

Backward Difference Equation:

$$f'(a) \approx \frac{f(a) - f(a-h)}{h}$$

Consider a real-valued function $f(x)$. The first derivative of f at $x = a$ may be approximated in different ways for $h = \Delta x$.



Forward Difference Equation:

$$f'(a) \approx \frac{f(a+h) - f(a)}{h}$$

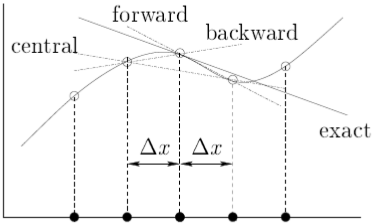
Central Difference Equation:

$$f'(a) \approx \frac{f(a+h) - f(a-h)}{2h}$$

Backward Difference Equation:

$$f'(a) \approx \frac{f(a) - f(a-h)}{h}$$

Consider a real-valued function $f(x)$. The first derivative of f at $x = a$ may be approximated in different ways for $h = \Delta x$.



Forward Difference Equation:

$$f'(a) \approx \frac{f(a+h) - f(a)}{h}$$

Central Difference Equation:

$$f'(a) \approx \frac{f(a+h) - f(a-h)}{2h}$$

Backward Difference Equation:

$$f'(a) \approx \frac{f(a) - f(a-h)}{h}$$

The accuracy of the three approximations can be compared by looking at the Taylor series expansion of f.

Taylor series expansion of $f(x)$ at a point a:

$$f(x) = \sum_{n=0}^{\infty} \frac{(x-a)^n}{n!} f^n(a) = f(a) + \frac{x-a}{1!} f'(a) + \frac{(x-a)^2}{2!} f''(a) + \cdots$$

Series expansions at $x = a + h$ and $x = a - h$:

$$f(a+h) = f(a) + hf'(a) + \frac{h^2}{2!} f''(a) + \frac{h^3}{3!} f'''(a) + \cdots$$
$$f(a-h) = f(a) - hf'(a) + \frac{h^2}{2!} f''(a) - \frac{h^3}{3!} f'''(a) + \cdots$$

The accuracy of the three approximations can be compared by looking at the Taylor series expansion of f.

Taylor series expansion of $f(x)$ at a point a:

$$f(x) = \sum_{n=0}^{\infty} \frac{(x-a)^n}{n!} f^n(a) = f(a) + \frac{x-a}{1!} f'(a) + \frac{(x-a)^2}{2!} f''(a) + \cdots$$

Series expansions at $x = a + h$ and $x = a - h$:

$$f(a+h) = f(a) + hf'(a) + \frac{h^2}{2!} f''(a) + \frac{h^3}{3!} f'''(a) + \cdots$$
$$f(a-h) = f(a) - hf'(a) + \frac{h^2}{2!} f''(a) - \frac{h^3}{3!} f'''(a) + \cdots$$

The accuracy of the three approximations can be compared by looking at the Taylor series expansion of f.

Taylor series expansion of $f(x)$ at a point a:

$$f(x) = \sum_{n=0}^{\infty} \frac{(x-a)^n}{n!} f^n(a) = f(a) + \frac{x-a}{1!} f'(a) + \frac{(x-a)^2}{2!} f''(a) + \cdots$$

Series expansions at $x = a + h$ and $x = a - h$:

$$f(a+h) = f(a) + hf'(a) + \frac{h^2}{2!} f''(a) + \frac{h^3}{3!} f'''(a) + \cdots$$
$$f(a-h) = f(a) - hf'(a) + \frac{h^2}{2!} f''(a) - \frac{h^3}{3!} f'''(a) + \cdots$$

Based on the previous series expansions, $f'(a)$ may be written as

$$f'(a) = \frac{f(a+h) - f(a)}{h} - \frac{h}{2!}f''(a) + \frac{h^2}{3!}f'''(a) + \cdots$$

$$f'(a) = \frac{f(a+h) - f(a-h)}{2h} - \frac{h^2}{3!}f'''(a) + \cdots$$

$$f'(a) = \frac{f(a) - f(a-h)}{h} + \frac{h}{2!}f''(a) - \frac{h^2}{3!}f'''(a) + \cdots$$

The orders of these approximations with respect to h:

$$f'(a) = \frac{f(a+h) - f(a)}{h} + \mathcal{O}(h)$$

$$f'(a) = \frac{f(a+h) - f(a-h)}{2h} + \mathcal{O}(h^2)$$

$$f'(a) = \frac{f(a) - f(a-h)}{h} + \mathcal{O}(h)$$

Based on the previous series expansions, $f'(a)$ may be written as

$$f'(a) = \frac{f(a+h) - f(a)}{h} - \frac{h}{2!}f''(a) + \frac{h^2}{3!}f'''(a) + \cdots$$
$$f'(a) = \frac{f(a+h) - f(a-h)}{2h} - \frac{h^2}{3!}f'''(a) + \cdots$$
$$f'(a) = \frac{f(a) - f(a-h)}{h} + \frac{h}{2!}f''(a) - \frac{h^2}{3!}f'''(a) + \cdots$$

The orders of these approximations with respect to h:

$$f'(a) = \frac{f(a+h) - f(a)}{h} + \mathcal{O}(h)$$
$$f'(a) = \frac{f(a+h) - f(a-h)}{2h} + \mathcal{O}(h^2)$$
$$f'(a) = \frac{f(a) - f(a-h)}{h} + \mathcal{O}(h)$$

Higher order derivatives are derived in a similar way from the Taylor series expansions.

For example, the second derivative is given by

$$f''(a) = \frac{f(a+h) - 2f(a) + f(a-h)}{h^2} + \mathcal{O}(h^2)$$

This approximation is sometimes referred to as the **Symmetric Difference Equation**.

Higher order derivatives are derived in a similar way from the Taylor series expansions.

For example, the second derivative is given by

$$f''(a) = \frac{f(a+h) - 2f(a) + f(a-h)}{h^2} + \mathcal{O}(h^2)$$

This approximation is sometimes referred to as the **Symmetric Difference Equation**.

Higher order derivatives are derived in a similar way from the Taylor series expansions.

For example, the second derivative is given by

$$f''(a) = \frac{f(a+h) - 2f(a) + f(a-h)}{h^2} + \mathcal{O}(h^2)$$

This approximation is sometimes referred to as the **Symmetric Difference Equation**.

Many physical quantities are functions of two or more variables.

Consider $f(t, x)$, a function of time and one-dimensional space. Parital derivative of f with respect to x for $a = x_\mathrm{o}$ and $h = \Delta x$:

$$f'(t, x_\mathrm{o}) = \frac{f(t, x_\mathrm{o} + \Delta x) - f(t, x_\mathrm{o})}{\Delta x}$$

For discrete values of $x = (x_1, x_2, \cdots, x_N)$, and discrete levels of $t = (t_\mathrm{o}, t_1, t_2, \cdots, t_n)$ the derivative at a point (t_n, x_i) is

$$f'(t_n, x_i) = \frac{f(t_n, x_{i+1}) - f(t_n, x_i)}{\Delta x} = \frac{f^n_{i+1} - f^n_i}{\Delta x}$$

Similarly, the partial derivative with respect to t:

$$\dot{f}(t_n, x_i) = \frac{f^{n+1}_i - f^n_i}{\Delta t}$$

Many physical quantities are functions of two or more variables.

Consider $f(t, x)$, a function of time and one-dimensional space. Parital derivative of f with respect to x for $a = x_o$ and $h = \Delta x$:

$$f'(t, x_o) = \frac{f(t, x_o + \Delta x) - f(t, x_o)}{\Delta x}$$

For discrete values of $x = (x_1, x_2, \cdots, x_N)$, and discrete levels of $t = (t_o, t_1, t_2, \cdots, t_n)$ the derivative at a point (t_n, x_i) is

$$f'(t_n, x_i) = \frac{f(t_n, x_{i+1}) - f(t_n, x_i)}{\Delta x} = \frac{f_{i+1}^n - f_i^n}{\Delta x}$$

Similarly, the partial derivative with respect to t:

$$\dot{f}(t_n, x_i) = \frac{f_i^{n+1} - f_i^n}{\Delta t}$$

Many physical quantities are functions of two or more variables.

Consider $f(t, x)$, a function of time and one-dimensional space. Parital derivative of f with respect to x for $a = x_o$ and $h = \Delta x$:

$$f'(t, x_o) = \frac{f(t, x_o + \Delta x) - f(t, x_o)}{\Delta x}$$

For discrete values of $x = (x_1, x_2, \cdots, x_N)$, and discrete levels of $t = (t_o, t_1, t_2, \cdots, t_n)$ the derivative at a point (t_n, x_i) is

$$f'(t_n, x_i) = \frac{f(t_n, x_{i+1}) - f(t_n, x_i)}{\Delta x} = \frac{f_{i+1}^n - f_i^n}{\Delta x}$$

Similarly, the partial derivative with respect to t:

$$\dot{f}(t_n, x_i) = \frac{f_i^{n+1} - f_i^n}{\Delta t}$$

Many physical quantities are functions of two or more variables.

Consider $f(t, x)$, a function of time and one-dimensional space. Parital derivative of f with respect to x for $a = x_\text{o}$ and $h = \Delta x$:

$$f'(t, x_\text{o}) = \frac{f(t, x_\text{o} + \Delta x) - f(t, x_\text{o})}{\Delta x}$$

For discrete values of $x = (x_1, x_2, \cdots, x_N)$, and discrete levels of $t = (t_\text{o}, t_1, t_2, \cdots, t_n)$ the derivative at a point (t_n, x_i) is

$$f'(t_n, x_i) = \frac{f(t_n, x_{i+1}) - f(t_n, x_i)}{\Delta x} = \frac{f^n_{i+1} - f^n_i}{\Delta x}$$

Similarly, the partial derivative with respect to t:

$$\dot{f}(t_n, x_i) = \frac{f^{n+1}_i - f^n_i}{\Delta t}$$

Many physical quantities are functions of two or more variables.

Consider $f(t, x)$, a function of time and one-dimensional space. Parital derivative of f with respect to x for $a = x_o$ and $h = \Delta x$:

$$f'(t, x_o) = \frac{f(t, x_o + \Delta x) - f(t, x_o)}{\Delta x}$$

For discrete values of $x = (x_1, x_2, \cdots, x_N)$, and discrete levels of $t = (t_o, t_1, t_2, \cdots, t_n)$ the derivative at a point (t_n, x_i) is

$$f'(t_n, x_i) = \frac{f(t_n, x_{i+1}) - f(t_n, x_i)}{\Delta x} = \frac{f_{i+1}^n - f_i^n}{\Delta x}$$

Similarly, the partial derivative with respect to t:

$$\dot{f}(t_n, x_i) = \frac{f_i^{n+1} - f_i^n}{\Delta t}$$

Partial Derivative	Difference Equation	Type	Accuracy
$f' = \dfrac{\partial f}{\partial x}$	$\dfrac{f_{i+1}^n - f_i^n}{\Delta x}$	Forward	$\mathcal{O}(\Delta x)$
$f' = \dfrac{\partial f}{\partial x}$	$\dfrac{f_i^n - f_{i-1}^n}{\Delta x}$	Backward	$\mathcal{O}(\Delta x)$
$f' = \dfrac{\partial f}{\partial x}$	$\dfrac{f_{i+1}^n - f_{i-1}^n}{2\Delta x}$	Central	$\mathcal{O}(\Delta x^2)$
$f'' = \dfrac{\partial^2 f}{\partial x^2}$	$\dfrac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\Delta x^2}$	Symmetric	$\mathcal{O}(\Delta x^2)$
$\dot{f} = \dfrac{\partial f}{\partial t}$	$\dfrac{f_i^{n+1} - f_i^n}{\Delta t}$	Forward	$\mathcal{O}(\Delta t)$
$\dot{f} = \dfrac{\partial f}{\partial t}$	$\dfrac{f_i^n - f_i^{n-1}}{\Delta t}$	Backward	$\mathcal{O}(\Delta t)$
$\dot{f} = \dfrac{\partial f}{\partial t}$	$\dfrac{f_i^{n+1} - f_i^{n-1}}{2\Delta t}$	Central	$\mathcal{O}(\Delta t^2)$
$\ddot{f} = \dfrac{\partial^2 f}{\partial t^2}$	$\dfrac{f_i^{n+1} - 2f_i^n + f_i^{n-1}}{\Delta t^2}$	Symmetric	$\mathcal{O}(\Delta t^2)$

Partial Derivative	Difference Equation	Type	Accuracy
$f' = \dfrac{\partial f}{\partial x}$	$\dfrac{f_{i+1}^n - f_i^n}{\Delta x}$	Forward	$\mathcal{O}(\Delta x)$
$f' = \dfrac{\partial f}{\partial x}$	$\dfrac{f_i^n - f_{i-1}^n}{\Delta x}$	Backward	$\mathcal{O}(\Delta x)$
$f' = \dfrac{\partial f}{\partial x}$	$\dfrac{f_{i+1}^n - f_{i-1}^n}{2\Delta x}$	Central	$\mathcal{O}(\Delta x^2)$
$f'' = \dfrac{\partial^2 f}{\partial x^2}$	$\dfrac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\Delta x^2}$	Symmetric	$\mathcal{O}(\Delta x^2)$
$\dot{f} = \dfrac{\partial f}{\partial t}$	$\dfrac{f_i^{n+1} - f_i^n}{\Delta t}$	Forward	$\mathcal{O}(\Delta t)$
$\dot{f} = \dfrac{\partial f}{\partial t}$	$\dfrac{f_i^n - f_i^{n-1}}{\Delta t}$	Backward	$\mathcal{O}(\Delta t)$
$\dot{f} = \dfrac{\partial f}{\partial t}$	$\dfrac{f_i^{n+1} - f_i^{n-1}}{2\Delta t}$	Central	$\mathcal{O}(\Delta t^2)$
$\ddot{f} = \dfrac{\partial^2 f}{\partial t^2}$	$\dfrac{f_i^{n+1} - 2f_i^n + f_i^{n-1}}{\Delta t^2}$	Symmetric	$\mathcal{O}(\Delta t^2)$

Partial Derivative	Difference Equation	Type	Accuracy
$f' = \frac{\partial f}{\partial x}$	$\frac{f_{i+1}^n - f_i^n}{\Delta x}$	Forward	$\mathcal{O}(\Delta x)$
$f' = \frac{\partial f}{\partial x}$	$\frac{f_i^n - f_{i-1}^n}{\Delta x}$	Backward	$\mathcal{O}(\Delta x)$
$f' = \frac{\partial f}{\partial x}$	$\frac{f_{i+1}^n - f_{i-1}^n}{2\Delta x}$	Central	$\mathcal{O}(\Delta x^2)$
$f'' = \frac{\partial^2 f}{\partial x^2}$	$\frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\Delta x^2}$	Symmetric	$\mathcal{O}(\Delta x^2)$
$\dot{f} = \frac{\partial f}{\partial t}$	$\frac{f_i^{n+1} - f_i^n}{\Delta t}$	Forward	$\mathcal{O}(\Delta t)$
$\dot{f} = \frac{\partial f}{\partial t}$	$\frac{f_i^n - f_i^{n-1}}{\Delta t}$	Backward	$\mathcal{O}(\Delta t)$
$\dot{f} = \frac{\partial f}{\partial t}$	$\frac{f_i^{n+1} - f_i^{n-1}}{2\Delta t}$	Central	$\mathcal{O}(\Delta t^2)$
$\ddot{f} = \frac{\partial^2 f}{\partial t^2}$	$\frac{f_i^{n+1} - 2f_i^n + f_i^{n-1}}{\Delta t^2}$	Symmetric	$\mathcal{O}(\Delta t^2)$

Partial Derivative	Difference Equation	Type	Accuracy
$f' = \dfrac{\partial f}{\partial x}$	$\dfrac{f_{i+1}^n - f_i^n}{\Delta x}$	Forward	$\mathcal{O}(\Delta x)$
$f' = \dfrac{\partial f}{\partial x}$	$\dfrac{f_i^n - f_{i-1}^n}{\Delta x}$	Backward	$\mathcal{O}(\Delta x)$
$f' = \dfrac{\partial f}{\partial x}$	$\dfrac{f_{i+1}^n - f_{i-1}^n}{2\Delta x}$	Central	$\mathcal{O}(\Delta x^2)$
$f'' = \dfrac{\partial^2 f}{\partial x^2}$	$\dfrac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\Delta x^2}$	Symmetric	$\mathcal{O}(\Delta x^2)$
$\dot{f} = \dfrac{\partial f}{\partial t}$	$\dfrac{f_i^{n+1} - f_i^n}{\Delta t}$	Forward	$\mathcal{O}(\Delta t)$
$\dot{f} = \dfrac{\partial f}{\partial t}$	$\dfrac{f_i^n - f_i^{n-1}}{\Delta t}$	Backward	$\mathcal{O}(\Delta t)$
$\dot{f} = \dfrac{\partial f}{\partial t}$	$\dfrac{f_i^{n+1} - f_i^{n-1}}{2\Delta t}$	Central	$\mathcal{O}(\Delta t^2)$
$\ddot{f} = \dfrac{\partial^2 f}{\partial t^2}$	$\dfrac{f_i^{n+1} - 2f_i^n + f_i^{n-1}}{\Delta t^2}$	Symmetric	$\mathcal{O}(\Delta t^2)$

Various problems in geotechnical engineering are mathematically described in terms of differential equations. Some examples include:

Groundwater Flow: General for confined aquifers

$$\frac{\partial}{\partial x}\left(k_x\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(k_y\frac{\partial h}{\partial y}\right)+\frac{\partial}{\partial z}\left(k_z\frac{\partial h}{\partial z}\right)=S_\mathrm{s}\frac{\partial h}{\partial t}$$

Various problems in geotechnical engineering are mathematically described in terms of differential equations. Some examples include:

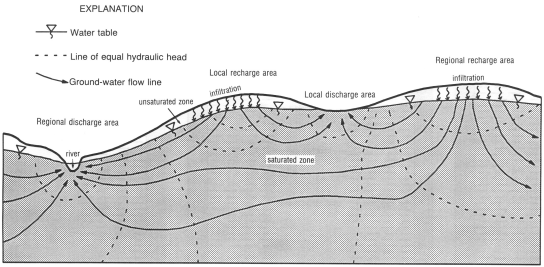
Groundwater Flow: General for confined aquifers

$$\frac{\partial}{\partial x}\left(k_x\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(k_y\frac{\partial h}{\partial y}\right)+\frac{\partial}{\partial z}\left(k_z\frac{\partial h}{\partial z}\right)=S_{\mathrm{s}}\frac{\partial h}{\partial t}$$

Various problems in geotechnical engineering are mathematically described in terms of differential equations. Some examples include:

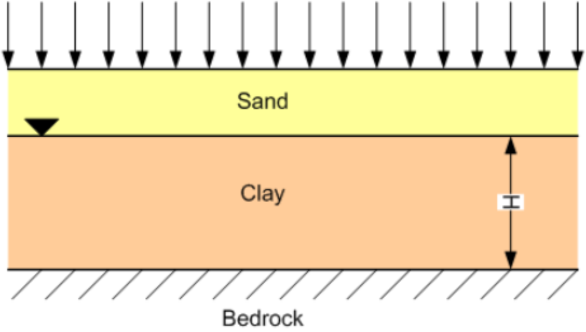
Groundwater Flow: General for confined aquifers

$$\frac{\partial}{\partial x}\left(k_x \frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial h}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z \frac{\partial h}{\partial z}\right) = S_\mathrm{s} \frac{\partial h}{\partial t}$$



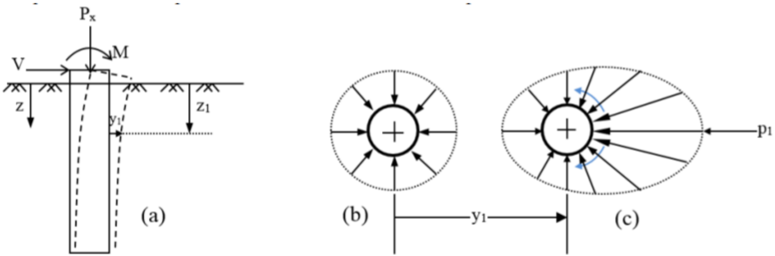
Consolidation:

$$\frac{\partial u}{\partial t} - c_{\mathrm{v}} \frac{\partial^2 u}{\partial z^2} = 0$$



Laterally Loaded Piles:

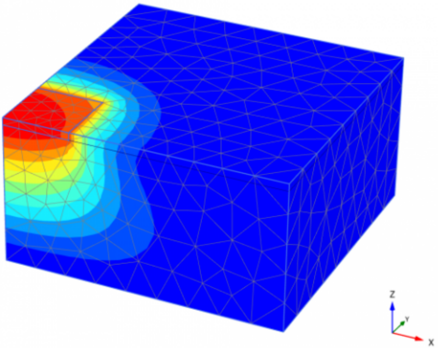
$$E_pI_p\frac{d^4y}{dx^4}+P_x\frac{d^2y}{dx^2}+E_{py}y-W=0$$



Coupled Deformation and Flow:

$$\nabla \cdot (\boldsymbol{\sigma}' + u\boldsymbol{I}) + \rho \boldsymbol{b} = \boldsymbol{0}$$

$$\nabla \cdot \dot{\boldsymbol{u}} + c\frac{\partial u}{\partial t} + \nabla \cdot \boldsymbol{v}_{\mathrm{w}} = 0$$



Initial Conditions (ICs):

- Specify the values of field variables at the start of a simulation.
- Usually specified for time dependent variables at $t = 0$.

Boundary Conditions (BCs):

- Define the values of field variables or functions of field variables at the boundaries of the computational domain.
- Boundary types depend on spatial dimension:
  1D - Point, 2D - Edge, 3D - Face

Initial Conditions (ICs):

- Specify the values of field variables at the start of a simulation.
- Usually specified for time dependent variables at $t = 0$.

Boundary Conditions (BCs):

- Define the values of field variables or functions of field variables at the boundaries of the computational domain.
- Boundary types depend on spatial dimension:
  1D - Point, 2D - Edge, 3D - Face

Three main types of boundary conditions:

1. **Dirichlet BCs**: specify the value of the unknown function on a given boundary in the form
$$f = \bar{f} \quad \text{on } \Gamma_{\mathrm{D}}$$
where Γ_{D} is the boundary on which the condition is specified.
2. **Neumann BCs**: specify the normal derivative of the unknown function on a given boundary i.e.
$$\frac{\partial f}{\partial n} = q \quad \text{on } \Gamma_{\mathrm{N}}$$
where n is the normal to the boundary Γ_{N}.
3. **Robin BCs**: define a linear combination of Dirichlet and Neumann conditions in the form
$$\alpha f + \beta \frac{\partial f}{\partial n} = r \quad \text{on } \Gamma_{\mathrm{R}}$$
where α and β are constants.

Three main types of boundary conditions:

1. **Dirichlet BCs**: specify the value of the unknown function on a given boundary in the form
$$f = \bar{f} \quad \text{on } \Gamma_\mathrm{D}$$
where Γ_D is the boundary on which the condition is specified.
2. **Neumann BCs**: specify the normal derivative of the unknown function on a given boundary i.e.
$$\frac{\partial f}{\partial n} = q \quad \text{on } \Gamma_\mathrm{N}$$
where n is the normal to the boundary Γ_N.
3. **Robin BCs**: define a linear combination of Dirichlet and Neumann conditions in the form
$$\alpha f + \beta \frac{\partial f}{\partial n} = r \quad \text{on } \Gamma_\mathrm{R}$$
where α and β are constants.

Three main types of boundary conditions:

1. **Dirichlet BCs**: specify the value of the unknown function on a given boundary in the form

$$f = \bar{f} \quad \text{on } \Gamma_{\mathrm{D}}$$

where Γ_{D} is the boundary on which the condition is specified.

2. **Neumann BCs**: specify the normal derivative of the unknown function on a given boundary i.e.

$$\frac{\partial f}{\partial n} = q \quad \text{on } \Gamma_{\mathrm{N}}$$

where n is the normal to the boundary Γ_{N}.

3. **Robin BCs**: define a linear combination of Dirichlet and Neumann conditions in the form

$$\alpha f + \beta \frac{\partial f}{\partial n} = r \quad \text{on } \Gamma_{\mathrm{R}}$$

where α and β are constants.

Three main types of boundary conditions:

1. **Dirichlet BCs**: specify the value of the unknown function on a given boundary in the form
$$f = \bar{f} \quad \text{on } \Gamma_{\mathrm{D}}$$
where Γ_{D} is the boundary on which the condition is specified.
2. **Neumann BCs**: specify the normal derivative of the unknown function on a given boundary i.e.
$$\frac{\partial f}{\partial n} = q \quad \text{on } \Gamma_{\mathrm{N}}$$
where n is the normal to the boundary Γ_{N}.
3. **Robin BCs**: define a linear combination of Dirichlet and Neumann conditions in the form
$$\alpha f + \beta \frac{\partial f}{\partial n} = r \quad \text{on } \Gamma_{\mathrm{R}}$$
where α and β are constants.

Spatial Discretization: The computational domain over which we aim to solve the PDE is divided into a grid mesh with the required grid spacing. Depending on the spatial dimension of the problem, the grid required may be 1D, 2D or 3D.

Temporal Discretization: If there are time dependent quantities in the differential equation, the next step is to perform temporal discretization of these to obtain difference equations.

Solving the Problem: The individual equations at all the grid points in the spatially discretized domain are assembled to form a system of equations of the form

$$Ax = b$$

which may be solved using the appropriate solution method.

Spatial Discretization: The computational domain over which we aim to solve the PDE is divided into a grid mesh with the required grid spacing. Depending on the spatial dimension of the problem, the grid required may be 1D, 2D or 3D.

Temporal Discretization: If there are time dependent quantities in the differential equation, the next step is to perform temporal discretization of these to obtain difference equations.

Solving the Problem: The individual equations at all the grid points in the spatially discretized domain are assembled to form a system of equations of the form

$$Ax = b$$

which may be solved using the appropriate solution method.

Spatial Discretization: The computational domain over which we aim to solve the PDE is divided into a grid mesh with the required grid spacing. Depending on the spatial dimension of the problem, the grid required may be 1D, 2D or 3D.

Temporal Discretization: If there are time dependent quantities in the differential equation, the next step is to perform temporal discretization of these to obtain difference equations.

Solving the Problem: The individual equations at all the grid points in the spatially discretized domain are assembled to form a system of equations of the form

$$\boldsymbol{Ax} = \boldsymbol{b}$$

which may be solved using the appropriate solution method.

Contents

One-dimensional Problems
↪Steady-state Flow

Governing equation in 3D:

$$k_x \frac{\partial^2 h}{\partial x^2} + k_y \frac{\partial^2 h}{\partial y^2} + k_z \frac{\partial^2 h}{\partial z^2} = Q$$

One-dimensional equivalent:

$$k \frac{\partial^2 h}{\partial x^2} = Q$$

Given k and Q, we aim to solve this equation for h on a 1D domain with length $x = L$:

Governing equation in 3D:

$$k_x \frac{\partial^2 h}{\partial x^2} + k_y \frac{\partial^2 h}{\partial y^2} + k_z \frac{\partial^2 h}{\partial z^2} = Q$$

One-dimensional equivalent:

$$k \frac{\partial^2 h}{\partial x^2} = Q$$

Given k and Q, we aim to solve this equation for h on a 1D domain with length $x = L$:

One-dimensional Problems
↪Steady-state Flow

Governing equation in 3D:

$$k_x \frac{\partial^2 h}{\partial x^2} + k_y \frac{\partial^2 h}{\partial y^2} + k_z \frac{\partial^2 h}{\partial z^2} = Q$$

One-dimensional equivalent:

$$k \frac{\partial^2 h}{\partial x^2} = Q$$

Given k and Q, we aim to solve this equation for h on a 1D domain with length $x = L$:

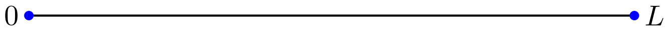
Governing equation in 3D:

$$k_x \frac{\partial^2 h}{\partial x^2} + k_y \frac{\partial^2 h}{\partial y^2} + k_z \frac{\partial^2 h}{\partial z^2} = Q$$

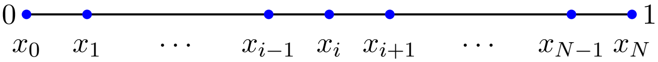
One-dimensional equivalent:

$$k \frac{\partial^2 h}{\partial x^2} = Q$$

Given k and Q, we aim to solve this equation for h on a 1D domain with length $x = L$:



Spatial discretization: Divide the domain into a desired number of sub-domains, say N:



The *grid spacing* in this case is

$$\Delta x = \frac{1}{N}$$

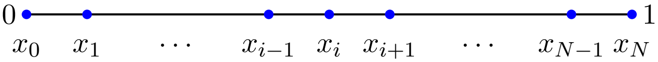
Hydraulic head values at grid points:

$$h(x_i) = h_i, \qquad i = 0, 1, \cdots, N$$

Assume the hydraulic heads are known at the end points (BCs):

$$h(x_0) = h_0$$
$$h(x_N) = h_N$$

Spatial discretization: Divide the domain into a desired number of sub-domains, say N:



The *grid spacing* in this case is

$$\Delta x = \frac{1}{N}$$

Hydraulic head values at grid points:

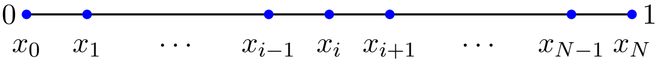
$$h(x_i) = h_i, \qquad i = 0, 1, \cdots, N$$

Assume the hydraulic heads are known at the end points (BCs):

$$h(x_0) = h_0$$
$$h(x_N) = h_N$$

One-dimensional Problems
↪Steady-state Flow

Spatial discretization: Divide the domain into a desired number of sub-domains, say N:



The *grid spacing* in this case is

$$\Delta x = \frac{1}{N}$$

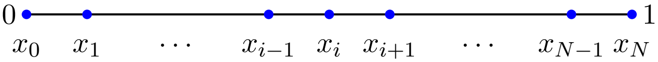
Hydraulic head values at grid points:

$$h(x_i) = h_i, \qquad i = 0, 1, \cdots, N$$

Assume the hydraulic heads are known at the end points (BCs):

$$h(x_0) = h_0$$
$$h(x_N) = h_N$$

Spatial discretization: Divide the domain into a desired number of sub-domains, say N:



The *grid spacing* in this case is

$$\Delta x = \frac{1}{N}$$

Hydraulic head values at grid points:

$$h(x_i) = h_i, \qquad i = 0, 1, \cdots, N$$

Assume the hydraulic heads are known at the end points (BCs):

$$h(x_0) = h_0$$
$$h(x_N) = h_N$$

Difference equation for partial derivative of hydraulic head:

$$\frac{\partial^2 h}{\partial x^2} = \frac{h_{i+1} - 2h_i + h_{i-1}}{\Delta x^2}$$

Discrete form of original equation, in terms of grid point values

$$k\,\frac{h_{i+1} - 2h_i + h_{i-1}}{(\Delta x)^2} = Q_i$$

Solution for $N = 2$:
Difference equation at internal grid point (assuming $Q_1 = 0$)

$$h_2 - 2h_1 + h_0 = 0$$

For known boundary values h_0 and h_2

$$h_1 = \frac{h_2 + h_0}{2}$$

Difference equation for partial derivative of hydraulic head:

$$\frac{\partial^2 h}{\partial x^2} = \frac{h_{i+1} - 2h_i + h_{i-1}}{\Delta x^2}$$

Discrete form of original equation, in terms of grid point values

$$k\,\frac{h_{i+1} - 2h_i + h_{i-1}}{(\Delta x)^2} = Q_i$$

Solution for $N = 2$:
Difference equation at internal grid point (assuming $Q_1 = 0$)

$$h_2 - 2h_1 + h_0 = 0$$

For known boundary values h_0 and h_2

$$h_1 = \frac{h_2 + h_0}{2}$$

Difference equation for partial derivative of hydraulic head:

$$\frac{\partial^2 h}{\partial x^2} = \frac{h_{i+1} - 2h_i + h_{i-1}}{\Delta x^2}$$

Discrete form of original equation, in terms of grid point values

$$k\,\frac{h_{i+1} - 2h_i + h_{i-1}}{(\Delta x)^2} = Q_i$$

Solution for $N = 2$:
Difference equation at internal grid point (assuming $Q_1 = 0$)

$$h_2 - 2h_1 + h_0 = 0$$

For known boundary values h_0 and h_2

$$h_1 = \frac{h_2 + h_0}{2}$$

Difference equation for partial derivative of hydraulic head:

$$\frac{\partial^2 h}{\partial x^2} = \frac{h_{i+1} - 2h_i + h_{i-1}}{\Delta x^2}$$

Discrete form of original equation, in terms of grid point values

$$k\,\frac{h_{i+1} - 2h_i + h_{i-1}}{(\Delta x)^2} = Q_i$$

Solution for $N = 2$:
Difference equation at internal grid point (assuming $Q_1 = 0$)

$$h_2 - 2h_1 + h_0 = 0$$

For known boundary values h_0 and h_2

$$h_1 = \frac{h_2 + h_0}{2}$$

Solution for arbitrary N:

Difference equations at unknown nodes $i = 1, 2, \cdots, N-1$:

$$
\begin{aligned}
i &= 1 & k\frac{h_2 - 2h_1 + h_0}{(\Delta x)^2} &= Q_1 \\
i &= 2 & k\frac{h_3 - 2h_2 + h_1}{(\Delta x)^2} &= Q_2 \\
&\vdots & &\vdots \\
i &= N-1 & k\frac{h_N - 2h_{N-1} + h_{N-2}}{(\Delta x)^2} &= Q_{N-1}
\end{aligned}
$$

Solution for arbitrary N:

In matrix form:

$$\frac{k}{(\Delta x)^2}\begin{bmatrix} 1 & -2 & 1 & & & & & \\ & 1 & -2 & 1 & & & & \\ & & 1 & -2 & 1 & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & 1 & -2 & 1 & \\ & & & & & 1 & -2 & 1 \end{bmatrix}\begin{Bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_{N-1} \\ h_N \end{Bmatrix} = \begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ \vdots \\ Q_{N-1} \end{Bmatrix}$$

Solution for arbitrary N:

Considering known boundary conditions:

$$\frac{k}{(\Delta x)^2}\begin{bmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix} \begin{Bmatrix} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_{N-1} \end{Bmatrix} = \begin{Bmatrix} Q_1 - \frac{k}{(\Delta x)^2} h_0 \\ Q_2 \\ Q_3 \\ \vdots \\ Q_{N-1} - \frac{k}{(\Delta x)^2} h_N \end{Bmatrix}$$

which is a linear equation system of the form

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Governing equation for one-dimensional consolidation:

$$\frac{\partial u}{\partial t} - c_{\mathrm{v}} \frac{\partial^2 u}{\partial z^2} = 0$$

Coefficient of consolidation

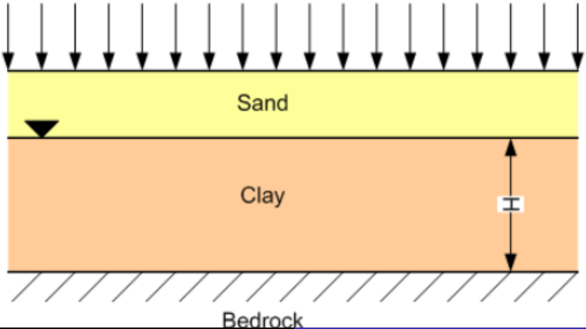
$$c_{\mathrm{v}} = \frac{k}{m_{\mathrm{v}} \gamma_{\mathrm{w}}}$$

Governing equation for one-dimensional consolidation:

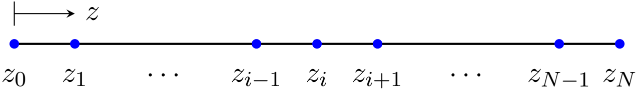
$$\frac{\partial u}{\partial t} - c_{\mathrm{v}} \frac{\partial^2 u}{\partial z^2} = 0$$

Coefficient of consolidation

$$c_{\mathrm{v}} = \frac{k}{m_{\mathrm{v}} \gamma_{\mathrm{w}}}$$



Spatial Discretization:
Consider a 1D domain sub-divided into N elements.



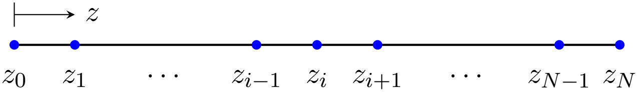
For a uniform grid spacing and domain depth/length d, we have

$$\Delta z = \frac{d}{N}$$

Temporal Discretization: For a simulation from time t_o to t_n, choosing a uniform time step Δt implies n time steps such that

$$t_n = n\Delta t$$

Spatial Discretization:
Consider a 1D domain sub-divided into N elements.



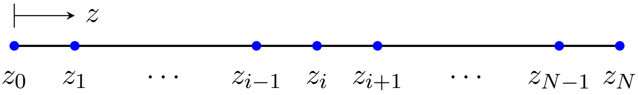
For a uniform grid spacing and domain depth/length d, we have

$$\Delta z = \frac{d}{N}$$

Temporal Discretization: For a simulation from time t_o to t_n, choosing a uniform time step Δt implies n time steps such that

$$t_n = n\Delta t$$

Spatial Discretization:
Consider a 1D domain sub-divided into N elements.

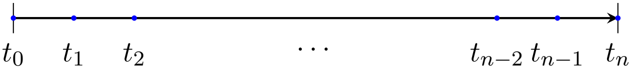


For a uniform grid spacing and domain depth/length d, we have

$$\Delta z = \frac{d}{N}$$

Temporal Discretization: For a simulation from time t_o to t_n, choosing a uniform time step Δt implies n time steps such that

$$t_n = n\Delta t$$



The excess pore pressure is a function of both time and space:

$$u = u(t, z)$$

The partial derivative with respect to time may be approximated as a difference equation in different ways, resulting in different solution methods.

1. Explicit Method: forward difference approximation in time.
2. Implicit Method: backward difference approximation in time.
3. Crank-Nicolson Method: central difference approximation in time.

The excess pore pressure is a function of both time and space:

$$u = u(t, z)$$

The partial derivative with respect to time may be approximated as a difference equation in different ways, resulting in different solution methods.

1. Explicit Method: forward difference approximation in time.
2. Implicit Method: backward difference approximation in time.
3. Crank-Nicolson Method: central difference approximation in time.

The excess pore pressure is a function of both time and space:

$$u = u(t, z)$$

The partial derivative with respect to time may be approximated as a difference equation in different ways, resulting in different solution methods.

1. Explicit Method: forward difference approximation in time.
2. Implicit Method: backward difference approximation in time.
3. Crank-Nicolson Method: central difference approximation in time.

Forward difference approximation for time derivative

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

Symmetric difference approximation for spatial derivative

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2}$$

Final explicit finite difference equation

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - c_{\mathrm{v}} \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2} = 0$$

Forward difference approximation for time derivative

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

Symmetric difference approximation for spatial derivative

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2}$$

Final explicit finite difference equation

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - c_\mathrm{v} \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2} = 0$$

Forward difference approximation for time derivative

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

Symmetric difference approximation for spatial derivative

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2}$$

Final explicit finite difference equation

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - c_{\mathrm{v}} \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2} = 0$$

Forward difference approximation for time derivative

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

Symmetric difference approximation for spatial derivative

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2}$$

Final explicit finite difference equation

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - c_{\mathrm{v}} \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2} = 0$$

Solving for pore water pressure at time step t_{n+1} gives

$$u_i^{n+1} = u_i^n + \kappa \left(u_{i+1}^n - 2u_i^n + u_{i-1}^n \right)$$

where

$$\kappa = c_{\mathrm{v}} \frac{\Delta t}{\Delta z^2}$$

Stencil for the explicit method

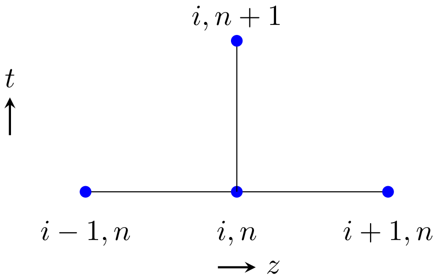
Solving for pore water pressure at time step t_{n+1} gives

$$u_i^{n+1} = u_i^n + \kappa \left(u_{i+1}^n - 2u_i^n + u_{i-1}^n \right)$$

where

$$\kappa = c_{\mathrm{v}} \frac{\Delta t}{\Delta z^2}$$

Stencil for the explicit method



Stability and Convergence:

The explicit method is numerically stable and convergent when

$$\kappa \leq \frac{1}{2}$$

Thus, the time step used must satisfy

$$\Delta t \leq \frac{\Delta z^2}{2c_{\mathrm{v}}}$$

Accuracy:

First-order in time, $\mathcal{O}(\Delta t)$, and second-order in space, $\mathcal{O}(\Delta z^2)$.

Backward difference approximation for time derivative

$$\frac{\partial u}{\partial t} = \frac{u_i^n - u_i^{n-1}}{\Delta t}$$

Symmetric difference approximation for spatial derivative

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2}$$

Final implicit finite difference equation

$$\frac{u_i^n - u_i^{n-1}}{\Delta t} - c_\text{v} \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2} = 0$$

Backward difference approximation for time derivative

$$\frac{\partial u}{\partial t} = \frac{u_i^n - u_i^{n-1}}{\Delta t}$$

Symmetric difference approximation for spatial derivative

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2}$$

Final implicit finite difference equation

$$\frac{u_i^n - u_i^{n-1}}{\Delta t} - c_{\mathrm{v}} \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2} = 0$$

Backward difference approximation for time derivative

$$\frac{\partial u}{\partial t} = \frac{u_i^n - u_i^{n-1}}{\Delta t}$$

Symmetric difference approximation for spatial derivative

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2}$$

Final implicit finite difference equation

$$\frac{u_i^n - u_i^{n-1}}{\Delta t} - c_{\mathrm{v}} \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2} = 0$$

Backward difference approximation for time derivative

$$\frac{\partial u}{\partial t} = \frac{u_i^n - u_i^{n-1}}{\Delta t}$$

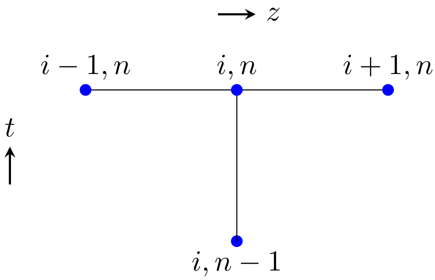
Symmetric difference approximation for spatial derivative

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2}$$

Final implicit finite difference equation

$$\frac{u_i^n - u_i^{n-1}}{\Delta t} - c_{\mathrm{v}} \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2} = 0$$

Stencil for the implicit method



The pore pressure at node i at time step t_n depends on the pore pressure values of the neighboring nodes at the same time step.
$\Rightarrow$ We can not solve for u_i^n directly.

Solution for $N = 3$:

Assume the pore pressures at the boundary nodes are known.

Finite difference equations at unknown interior nodes

$$\frac{u_1^n - u_1^{n-1}}{\Delta t} - c_v \frac{u_2^n - 2u_1^n + u_0^n}{\Delta z^2} = 0$$
$$\frac{u_2^n - u_2^{n-1}}{\Delta t} - c_v \frac{u_3^n - 2u_2^n + u_1^n}{\Delta z^2} = 0$$

Equations to be solved for unknowns u_1^n and u_2^n

$$(1 + 2\kappa)u_1^n - \kappa u_2^n = \kappa u_0^n + u_1^{n-1}$$
$$-\kappa u_1^n + (1 + 2\kappa)u_2^n = \kappa u_3^n + u_2^{n-1}$$

In matrix form

$$\begin{bmatrix} 1 + 2\kappa & -\kappa \\ -\kappa & 1 + 2\kappa \end{bmatrix} \begin{Bmatrix} u_1^n \\ u_2^n \end{Bmatrix} = \begin{Bmatrix} \kappa u_0^n + u_1^{n-1} \\ \kappa u_3^n + u_2^{n-1} \end{Bmatrix}$$

Solution for $N = 3$:

Assume the pore pressures at the boundary nodes are known.

Finite difference equations at unknown interior nodes

$$\frac{u_1^n - u_1^{n-1}}{\Delta t} - c_\mathrm{v} \frac{u_2^n - 2u_1^n + u_0^n}{\Delta z^2} = 0$$
$$\frac{u_2^n - u_2^{n-1}}{\Delta t} - c_\mathrm{v} \frac{u_3^n - 2u_2^n + u_1^n}{\Delta z^2} = 0$$

Equations to be solved for unknowns u_1^n and u_2^n

$$(1 + 2\kappa)u_1^n - \kappa u_2^n = \kappa u_0^n + u_1^{n-1}$$
$$-\kappa u_1^n + (1 + 2\kappa)u_2^n = \kappa u_3^n + u_2^{n-1}$$

In matrix form

$$\begin{bmatrix} 1 + 2\kappa & -\kappa \\ -\kappa & 1 + 2\kappa \end{bmatrix} \begin{Bmatrix} u_1^n \\ u_2^n \end{Bmatrix} = \begin{Bmatrix} \kappa u_0^n + u_1^{n-1} \\ \kappa u_3^n + u_2^{n-1} \end{Bmatrix}$$

Solution for $N = 3$:

Assume the pore pressures at the boundary nodes are known.

Finite difference equations at unknown interior nodes

$$\frac{u_1^n - u_1^{n-1}}{\Delta t} - c_\mathrm{v}\frac{u_2^n - 2u_1^n + u_0^n}{\Delta z^2} = 0$$
$$\frac{u_2^n - u_2^{n-1}}{\Delta t} - c_\mathrm{v}\frac{u_3^n - 2u_2^n + u_1^n}{\Delta z^2} = 0$$

Equations to be solved for unknowns u_1^n and u_2^n

$$(1+2\kappa)u_1^n - \kappa u_2^n = \kappa u_0^n + u_1^{n-1}$$
$$-\kappa u_1^n + (1+2\kappa)u_2^n = \kappa u_3^n + u_2^{n-1}$$

In matrix form

$$\begin{bmatrix} 1+2\kappa & -\kappa \\ -\kappa & 1+2\kappa \end{bmatrix} \begin{Bmatrix} u_1^n \\ u_2^n \end{Bmatrix} = \begin{Bmatrix} \kappa u_0^n + u_1^{n-1} \\ \kappa u_3^n + u_2^{n-1} \end{Bmatrix}$$

Solution for $N = 3$:

Assume the pore pressures at the boundary nodes are known.

Finite difference equations at unknown interior nodes

$$\frac{u_1^n - u_1^{n-1}}{\Delta t} - c_v \frac{u_2^n - 2u_1^n + u_0^n}{\Delta z^2} = 0$$
$$\frac{u_2^n - u_2^{n-1}}{\Delta t} - c_v \frac{u_3^n - 2u_2^n + u_1^n}{\Delta z^2} = 0$$

Equations to be solved for unknowns u_1^n and u_2^n

$$(1 + 2\kappa)u_1^n - \kappa u_2^n = \kappa u_0^n + u_1^{n-1}$$
$$-\kappa u_1^n + (1 + 2\kappa)u_2^n = \kappa u_3^n + u_2^{n-1}$$

In matrix form

$$\begin{bmatrix} 1 + 2\kappa & -\kappa \\ -\kappa & 1 + 2\kappa \end{bmatrix} \begin{Bmatrix} u_1^n \\ u_2^n \end{Bmatrix} = \begin{Bmatrix} \kappa u_0^n + u_1^{n-1} \\ \kappa u_3^n + u_2^{n-1} \end{Bmatrix}$$

Solution for arbitrary N:

Assume the pore pressures at the boundary nodes are known.

Finite difference equation

$$-\kappa u_{i-1}^{n} + (1+2\kappa)u_i^n - \kappa u_{i+1}^{n} = u_i^{n-1}$$

Solution for arbitrary N:

Assume the pore pressures at the boundary nodes are known.

Finite difference equation

$$-\kappa u_{i-1}^{n} + (1 + 2\kappa)u_i^n - \kappa u_{i+1}^n = u_i^{n-1}$$

Solution for arbitrary N:

Assume the pore pressures at the boundary nodes are known.

Finite difference equation

$$-\kappa u_{i-1}^{n} + (1+2\kappa)u_i^n - \kappa u_{i+1}^n = u_i^{n-1}$$

Finite difference equation at the unknown interior nodes

$$\begin{aligned}
i &= 1 && -\kappa u_0^n + (1+2\kappa)u_1^n - \kappa u_2^n = u_1^{n-1} \\
i &= 2 && -\kappa u_1^n + (1+2\kappa)u_2^n - \kappa u_3^n = u_2^{n-1} \\
&\vdots && \qquad\qquad\qquad \vdots \\
i &= N-2 && -\kappa u_{N-3}^n + (1+2\kappa)u_{N-2}^n - \kappa u_{N-1}^n = u_{N-2}^{n-1} \\
i &= N-1 && -\kappa u_{N-2}^n + (1+2\kappa)u_{N-1}^n - \kappa u_N^n = u_{N-1}^{n-1}
\end{aligned}$$

Solution for arbitrary N:

Assume the pore pressures at the boundary nodes are known.

Finite difference equation

$$-\kappa u_{i-1}^{n} + (1+2\kappa)u_i^n - \kappa u_{i+1}^n = u_i^{n-1}$$

In matrix form

$$\begin{bmatrix} -\kappa & 1+2\kappa & -\kappa & & & & & \\ & -\kappa & 1+2\kappa & -\kappa & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & -\kappa & 1+2\kappa & -\kappa & \\ & & & & & -\kappa & 1+2\kappa & -\kappa \end{bmatrix} \begin{Bmatrix} {\color{blue}u_0^n} \\ u_1^n \\ u_2^n \\ \vdots \\ u_{N-2}^n \\ u_{N-1}^n \\ {\color{blue}u_N^n} \end{Bmatrix} = \begin{Bmatrix} u_1^{n-1} \\ u_2^{n-1} \\ \vdots \\ u_{N-2}^{n-1} \\ u_{N-1}^{n-1} \end{Bmatrix}$$

Solution for arbitrary N:

Assume the pore pressures at the boundary nodes are known.

Finite difference equation

$$-\kappa u_{i-1}^{n} + (1+2\kappa)u_i^n - \kappa u_{i+1}^{n} = u_i^{n-1}$$

Matrix form considering boundary conditions

$$\begin{bmatrix} 1+2\kappa & -\kappa & & & & \\ -\kappa & 1+2\kappa & -\kappa & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\kappa & 1+2\kappa & -\kappa \\ & & & & -\kappa & 1+2\kappa \end{bmatrix} \begin{Bmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_{N-2}^n \\ u_{N-1}^n \end{Bmatrix} = \begin{Bmatrix} \kappa u_0^n + u_1^{n-1} \\ u_2^{n-1} \\ \vdots \\ u_{N-2}^{n-1} \\ \kappa u_N^n + u_{N-1}^{n-1} \end{Bmatrix}$$

Stability and Convergence:

The implicit method is numerically stable and convergent.

Accuracy:

First-order in time, $\mathcal{O}(\Delta t)$, and second-order in space, $\mathcal{O}(\Delta z^2)$.

Central difference approximation for time derivative at $(t_{n+\frac{1}{2}}, z_i)$

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{2\frac{\Delta t}{2}} = \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

Finite difference approximation for spatial derivative at $(t_{n+\frac{1}{2}}, z_i)$

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i+1}^{n+\frac{1}{2}} - 2u_i^{n+\frac{1}{2}} + u_{i-1}^{n+\frac{1}{2}}}{\Delta z^2}$$

Assume for all i that

$$u_i^{n+\frac{1}{2}} \approx \frac{1}{2}\left(u_i^n + u_i^{n+1}\right)$$

$$\Rightarrow \frac{\partial^2 u}{\partial z^2} = \frac{1}{2}\left(\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta z^2} + \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2}\right)$$

Central difference approximation for time derivative at $(t_{n+\frac{1}{2}}, z_i)$

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{2\frac{\Delta t}{2}} = \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

Finite difference approximation for spatial derivative at $(t_{n+\frac{1}{2}}, z_i)$

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i+1}^{n+\frac{1}{2}} - 2u_i^{n+\frac{1}{2}} + u_{i-1}^{n+\frac{1}{2}}}{\Delta z^2}$$

Assume for all i that

$$u_i^{n+\frac{1}{2}} \approx \frac{1}{2}\left(u_i^n + u_i^{n+1}\right)$$

$$\Rightarrow \frac{\partial^2 u}{\partial z^2} = \frac{1}{2}\left(\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta z^2} + \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2}\right)$$

Central difference approximation for time derivative at $(t_{n+\frac{1}{2}}, z_i)$

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{2\frac{\Delta t}{2}} = \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

Finite difference approximation for spatial derivative at $(t_{n+\frac{1}{2}}, z_i)$

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i+1}^{n+\frac{1}{2}} - 2u_i^{n+\frac{1}{2}} + u_{i-1}^{n+\frac{1}{2}}}{\Delta z^2}$$

Assume for all i that

$$u_i^{n+\frac{1}{2}} \approx \frac{1}{2}\left(u_i^n + u_i^{n+1}\right)$$

$$\Rightarrow \frac{\partial^2 u}{\partial z^2} = \frac{1}{2}\left(\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta z^2} + \frac{u_{i+1}^{n} - 2u_i^{n} + u_{i-1}^{n}}{\Delta z^2}\right)$$

Central difference approximation for time derivative at $(t_{n+\frac{1}{2}}, z_i)$

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{2\frac{\Delta t}{2}} = \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

Finite difference approximation for spatial derivative at $(t_{n+\frac{1}{2}}, z_i)$

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i+1}^{n+\frac{1}{2}} - 2u_i^{n+\frac{1}{2}} + u_{i-1}^{n+\frac{1}{2}}}{\Delta z^2}$$

Assume for all i that

$$u_i^{n+\frac{1}{2}} \approx \frac{1}{2}\left(u_i^n + u_i^{n+1}\right)$$

$$\Rightarrow \frac{\partial^2 u}{\partial z^2} = \frac{1}{2}\left(\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta z^2} + \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2}\right)$$

Central difference approximation for time derivative at $(t_{n+\frac{1}{2}}, z_i)$

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{2\frac{\Delta t}{2}} = \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

Finite difference approximation for spatial derivative at $(t_{n+\frac{1}{2}}, z_i)$

$$\frac{\partial^2 u}{\partial z^2} = \frac{u_{i+1}^{n+\frac{1}{2}} - 2u_i^{n+\frac{1}{2}} + u_{i-1}^{n+\frac{1}{2}}}{\Delta z^2}$$

Assume for all i that

$$u_i^{n+\frac{1}{2}} \approx \frac{1}{2}\left(u_i^n + u_i^{n+1}\right)$$

$$\Rightarrow \frac{\partial^2 u}{\partial z^2} = \frac{1}{2}\left(\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta z^2} + \frac{u_{i+1}^{n} - 2u_i^{n} + u_{i-1}^{n}}{\Delta z^2}\right)$$

Final finite difference equation

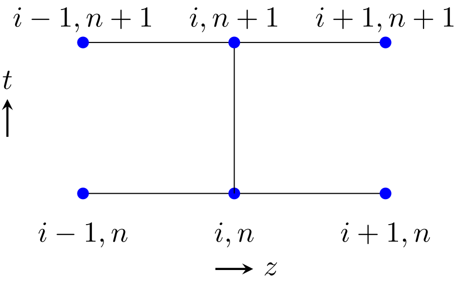
$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{c_\mathrm{v}}{2}\left(\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta z^2} + \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2}\right) = 0$$

Stencil for the Crank-Nicolson method

Final finite difference equation

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{c_\mathrm{v}}{2}\left(\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta z^2} + \frac{u_{i+1}^{n} - 2u_i^{n} + u_{i-1}^{n}}{\Delta z^2}\right) = 0$$

Stencil for the Crank-Nicolson method



Simplified finite difference equation with unknowns on the LHS

$$-\frac{\kappa}{2}u_{i-1}^{n+1} + (1+\kappa)u_i^{n+1} - \frac{\kappa}{2}u_{i+1}^{n+1} = \frac{\kappa}{2}u_{i-1}^{n} + (1-\kappa)u_i^{n} + \frac{\kappa}{2}u_{i+1}^{n}$$

Solution for arbitrary N:

Simplified finite difference equation with unknowns on the LHS

$$-\frac{\kappa}{2}u_{i-1}^{n+1} + (1+\kappa)u_i^{n+1} - \frac{\kappa}{2}u_{i+1}^{n+1} = \frac{\kappa}{2}u_{i-1}^{n} + (1-\kappa)u_i^{n} + \frac{\kappa}{2}u_{i+1}^{n}$$

Solution for arbitrary N:

Finite difference equation at the unknown interior nodes

$$\begin{aligned}
-\frac{\kappa}{2}u_0^{n+1} + (1+\kappa)u_1^{n+1} - \frac{\kappa}{2}u_2^{n+1} &= \frac{\kappa}{2}u_0^{n} + (1-\kappa)u_1^{n} + \frac{\kappa}{2}u_2^{n} \\
-\frac{\kappa}{2}u_1^{n+1} + (1+\kappa)u_2^{n+1} - \frac{\kappa}{2}u_3^{n+1} &= \frac{\kappa}{2}u_1^{n} + (1-\kappa)u_2^{n} + \frac{\kappa}{2}u_3^{n} \\
\vdots \qquad\qquad & \vdots \qquad\qquad \vdots \\
-\frac{\kappa}{2}u_{N-3}^{n+1} + (1+\kappa)u_{N-2}^{n+1} - \frac{\kappa}{2}u_{N-1}^{n+1} &= \frac{\kappa}{2}u_{N-3}^{n} + (1-\kappa)u_{N-2}^{n} + \frac{\kappa}{2}u_{N-1}^{n} \\
-\frac{\kappa}{2}u_{N-2}^{n+1} + (1+\kappa)u_{N-1}^{n+1} - \frac{\kappa}{2}u_N^{n+1} &= \frac{\kappa}{2}u_{N-2}^{n} + (1-\kappa)u_{N-1}^{n} + \frac{\kappa}{2}u_N^{n}
\end{aligned}$$

Simplified finite difference equation with unknowns on the LHS

$$-\frac{\kappa}{2}u_{i-1}^{n+1} + (1+\kappa)u_i^{n+1} - \frac{\kappa}{2}u_{i+1}^{n+1} = \frac{\kappa}{2}u_{i-1}^{n} + (1-\kappa)u_i^{n} + \frac{\kappa}{2}u_{i+1}^{n}$$

Solution for arbitrary N:

In matrix form

$$\begin{bmatrix} 1+\kappa & -\frac{\kappa}{2} & & & & \\ -\frac{\kappa}{2} & 1+\kappa & -\frac{\kappa}{2} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\frac{\kappa}{2} & 1+\kappa & -\frac{\kappa}{2} \\ & & & & -\frac{\kappa}{2} & 1+\kappa \end{bmatrix} \begin{Bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ \vdots \\ u_{N-2}^{n+1} \\ u_{N-1}^{n+1} \end{Bmatrix} = \begin{Bmatrix} b_1 + \frac{\kappa}{2}u_0^{n+1} \\ b_2 \\ \vdots \\ b_{N-2} \\ b_{N-1} + \frac{\kappa}{2}u_N^{n+1} \end{Bmatrix}$$

where

$$b_i = \frac{\kappa}{2}u_{i-1}^{n} + (1-\kappa)u_i^{n} + \frac{\kappa}{2}u_{i+1}^{n}$$

Consider a 1 m thick soil subjected to a surcharge load of 50 kPa.

Coefficient of consolidation:

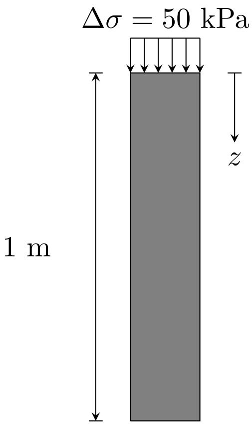
$$c_v = 2 \times 10^{-6}\ \mathrm{m}^2/\mathrm{s}$$

Dimensionless time

$$T = \frac{c_v t}{H^2}$$

Solve for the excess pore pressure as a function of time using the explicit, implicit and Crank-Nicolson methods.

Consider a 1 m thick soil subjected to a surcharge load of 50 kPa.



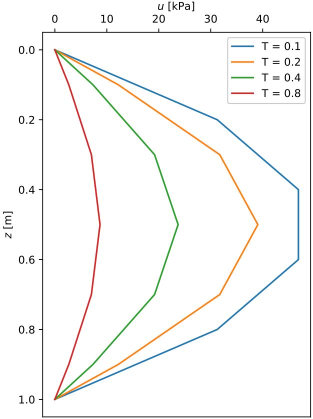
Coefficient of consolidation:

$$c_v = 2 \times 10^{-6}\ \mathrm{m}^2/\mathrm{s}$$

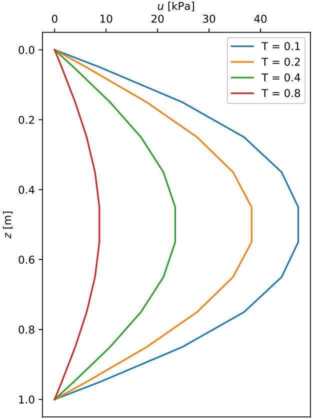
Dimensionless time

$$T = \frac{c_{\mathrm{v}} t}{H^2}$$

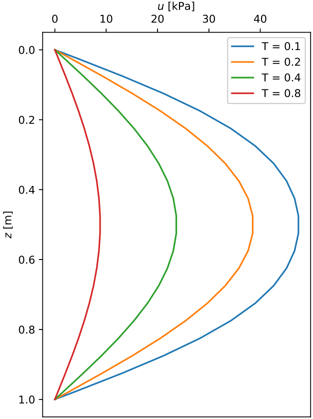
Solve for the excess pore pressure as a function of time using the explicit, implicit and Crank-Nicolson methods.



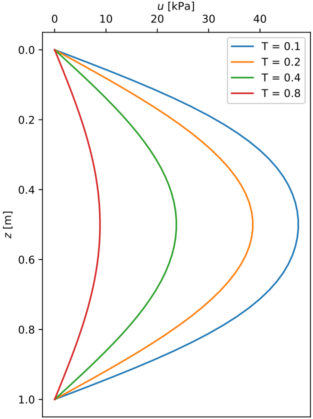
Explicit: $N = 10, \Delta t = 2500$



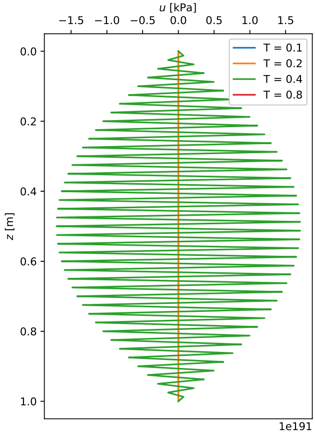
Explicit: $N = 20, \Delta t = 645$



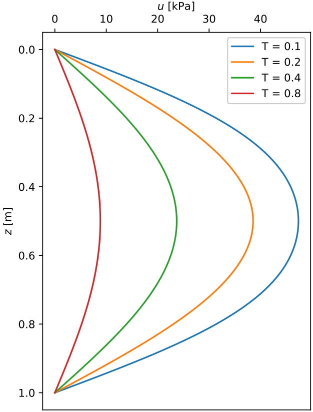
Explicit: $N = 40, \Delta t = 156$



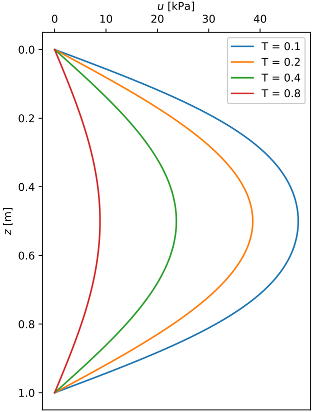
Explicit: $N = 80, \Delta t = 39$



Explicit: $N = 80, \Delta t = 50$



Implicit: $N = 80, \Delta t = 200$



Crank-Nicolson: $N = 80, \Delta t = 100$

Contents

Two-dimensional Problems

Steady-state groundwater flow in two-dimensions: The governing differential equation is

$$k_x \frac{\partial^2 h}{\partial x^2} + k_y \frac{\partial^2 h}{\partial y^2} = Q$$

For an isotropic material and neglecting source/sink terms

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = 0$$

Steady-state groundwater flow in two-dimensions: The governing differential equation is

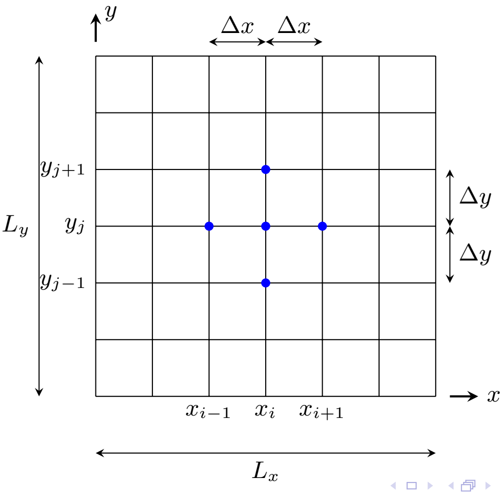
$$k_x \frac{\partial^2 h}{\partial x^2} + k_y \frac{\partial^2 h}{\partial y^2} = Q$$

For an isotropic material and neglecting source/sink terms

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = 0$$

Spatial Discretization:

N_x and N_y elements with grid spacing of Δx and Δy, respectively.



For a uniform grid we have

$$\Delta x = \Delta y$$

Grid spacings are calculated from

$$\Delta x = \frac{L_x}{N_x} \quad \text{and} \quad \Delta y = \frac{L_y}{N_y}$$

Notation for hydrauclic head at point (x_i, j_j)

$$h(x_i, y_j) = h_{i,j}$$

for $i = 0, 1, \cdots, N_x$ and $j = 0, 1, \cdots, N_y$.

Boundary conditions are specified on edges. For example

$$h_{0,j} = h_{N,j} = \bar{h}_1 \quad \text{and} \quad h_{i,0} = h_{i,N} = \bar{h}_2$$

For a uniform grid we have

$$\Delta x = \Delta y$$

Grid spacings are calculated from

$$\Delta x = \frac{L_x}{N_x} \quad \text{and} \quad \Delta y = \frac{L_y}{N_y}$$

Notation for hydrauclic head at point (x_i, j_j)

$$h(x_i, y_j) = h_{i,j}$$

for $i = 0, 1, \cdots, N_x$ and $j = 0, 1, \cdots, N_y$.

Boundary conditions are specified on edges. For example

$$h_{0,j} = h_{N,j} = \bar{h}_1 \quad \text{and} \quad h_{i,0} = h_{i,N} = \bar{h}_2$$

For a uniform grid we have

$$\Delta x = \Delta y$$

Grid spacings are calculated from

$$\Delta x = \frac{L_x}{N_x} \quad \text{and} \quad \Delta y = \frac{L_y}{N_y}$$

Notation for hydrauclic head at point (x_i, j_j)

$$h(x_i, y_j) = h_{i,j}$$

for $i = 0, 1, \cdots, N_x$ and $j = 0, 1, \cdots, N_y$.

Boundary conditions are specified on edges. For example

$$h_{0,j} = h_{N,j} = \bar{h}_1 \quad \text{and} \quad h_{i,0} = h_{i,N} = \bar{h}_2$$

For a uniform grid we have

$$\Delta x = \Delta y$$

Grid spacings are calculated from

$$\Delta x = \frac{L_x}{N_x} \quad \text{and} \quad \Delta y = \frac{L_y}{N_y}$$

Notation for hydrauclic head at point (x_i, j_j)

$$h(x_i, y_j) = h_{i,j}$$

for $i = 0, 1, \cdots, N_x$ and $j = 0, 1, \cdots, N_y$.

Boundary conditions are specified on edges. For example

$$h_{0,j} = h_{N,j} = \bar{h}_1 \quad \text{and} \quad h_{i,0} = h_{i,N} = \bar{h}_2$$

Finite difference approximations for spatial derivatives

$$\frac{\partial^2 h}{\partial x^2} = \frac{h_{i+1,j} - 2h_{i,j} + h_{i-1,j}}{\Delta x^2}$$
$$\frac{\partial^2 h}{\partial y^2} = \frac{h_{i,j+1} - 2h_{i,j} + h_{i,j-1}}{\Delta y^2}$$

Based on the governing equation, we get

$$\frac{h_{i+1,j} - 2h_{i,j} + h_{i-1,j}}{\Delta x^2} + \frac{h_{i,j+1} - 2h_{i,j} + h_{i,j-1}}{\Delta y^2} = 0$$

For a uniform grid where $\Delta x = \Delta y$

$$h_{i,j-1} + h_{i-1,j} - 4h_{i,j} + h_{i+1,j} + h_{i,j+1} = 0$$

Finite difference approximations for spatial derivatives

$$\frac{\partial^2 h}{\partial x^2} = \frac{h_{i+1,j} - 2h_{i,j} + h_{i-1,j}}{\Delta x^2}$$
$$\frac{\partial^2 h}{\partial y^2} = \frac{h_{i,j+1} - 2h_{i,j} + h_{i,j-1}}{\Delta y^2}$$

Based on the governing equation, we get

$$\frac{h_{i+1,j} - 2h_{i,j} + h_{i-1,j}}{\Delta x^2} + \frac{h_{i,j+1} - 2h_{i,j} + h_{i,j-1}}{\Delta y^2} = 0$$

For a uniform grid where $\Delta x = \Delta y$

$$h_{i,j-1} + h_{i-1,j} - 4h_{i,j} + h_{i+1,j} + h_{i,j+1} = 0$$

Finite difference approximations for spatial derivatives

$$\frac{\partial^2 h}{\partial x^2} = \frac{h_{i+1,j} - 2h_{i,j} + h_{i-1,j}}{\Delta x^2}$$
$$\frac{\partial^2 h}{\partial y^2} = \frac{h_{i,j+1} - 2h_{i,j} + h_{i,j-1}}{\Delta y^2}$$

Based on the governing equation, we get

$$\frac{h_{i+1,j} - 2h_{i,j} + h_{i-1,j}}{\Delta x^2} + \frac{h_{i,j+1} - 2h_{i,j} + h_{i,j-1}}{\Delta y^2} = 0$$

For a uniform grid where $\Delta x = \Delta y$

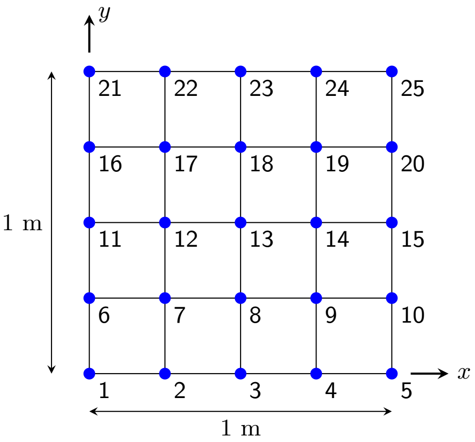
$$h_{i,j-1} + h_{i-1,j} - 4h_{i,j} + h_{i+1,j} + h_{i,j+1} = 0$$

Solution for $N_x = N_y = 4$:

Consider a $1\ \mathrm{m} \times 1\ \mathrm{m}$ grid divided into 4 elements.

Solution for $N_x = N_y = 4$:

Consider a $1\ \mathrm{m} \times 1\ \mathrm{m}$ grid divided into 4 elements.



Assume that the boundary conditions are

$$
\begin{aligned}
h &= 10 && \text{for } y = 1 \\
h &= 0 && \text{for } x = 0, x = 1 \text{ and } y = 0
\end{aligned}
$$

Finite difference equations for nodes 7, 8 and 9

$$
\begin{aligned}
-4h_7 + h_8 + h_{12} &= 0 \\
h_7 - 4h_8 + h_9 + h_{13} &= 0 \\
h_8 - 4h_9 + h_{14} &= 0
\end{aligned}
$$

Finite difference equations for nodes 12, 13 and 14

$$
\begin{aligned}
h_7 - 4h_{12} + h_{13} + h_{17} &= 0 \\
h_8 + h_{12} - 4h_{13} + h_{14} + h_{18} &= 0 \\
h_9 + h_{13} - 4h_{14} + h_{19} &= 0
\end{aligned}
$$

Assume that the boundary conditions are

$$
\begin{aligned}
h &= 10 && \text{for } y = 1 \\
h &= 0 && \text{for } x = 0, x = 1 \text{ and } y = 0
\end{aligned}
$$

Finite difference equations for nodes 7, 8 and 9

$$
\begin{aligned}
-4h_7 + h_8 + h_{12} &= 0 \\
h_7 - 4h_8 + h_9 + h_{13} &= 0 \\
h_8 - 4h_9 + h_{14} &= 0
\end{aligned}
$$

Finite difference equations for nodes 12, 13 and 14

$$
\begin{aligned}
h_7 - 4h_{12} + h_{13} + h_{17} &= 0 \\
h_8 + h_{12} - 4h_{13} + h_{14} + h_{18} &= 0 \\
h_9 + h_{13} - 4h_{14} + h_{19} &= 0
\end{aligned}
$$

Assume that the boundary conditions are

$$
\begin{aligned}
h &= 10 \qquad \text{for } y = 1 \\
h &= 0 \qquad \text{for } x = 0, x = 1 \text{ and } y = 0
\end{aligned}
$$

Finite difference equations for nodes 7, 8 and 9

$$
\begin{aligned}
-4h_7 + h_8 + h_{12} &= 0 \\
h_7 - 4h_8 + h_9 + h_{13} &= 0 \\
h_8 - 4h_9 + h_{14} &= 0
\end{aligned}
$$

Finite difference equations for nodes 12, 13 and 14

$$
\begin{aligned}
h_7 - 4h_{12} + h_{13} + h_{17} &= 0 \\
h_8 + h_{12} - 4h_{13} + h_{14} + h_{18} &= 0 \\
h_9 + h_{13} - 4h_{14} + h_{19} &= 0
\end{aligned}
$$

Finite difference equations for nodes 17, 18 and 19

$$h_{12} - 4h_{17} + h_{18} = -10$$
$$h_{13} + h_{17} - 4h_{18} + h_{19} = -10$$
$$h_{14} + h_{18} - 4h_{19} = -10$$

Assembling the equations into a matrix form gives

$$\left[\begin{array}{ccc|ccc|ccc} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & -4 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{array}\right] \begin{Bmatrix} h_7 \\ h_8 \\ h_9 \\ h_{12} \\ h_{13} \\ h_{14} \\ h_{17} \\ h_{18} \\ h_{19} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -10 \\ -10 \\ -10 \end{Bmatrix}$$

Finite difference equations for nodes 17, 18 and 19

$$\begin{aligned} h_{12} - 4h_{17} + h_{18} &= -10 \\ h_{13} + h_{17} - 4h_{18} + h_{19} &= -10 \\ h_{14} + h_{18} - 4h_{19} &= -10 \end{aligned}$$

Assembling the equations into a matrix form gives

$$\left[\begin{array}{ccc|ccc|ccc} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & -4 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{array}\right] \begin{Bmatrix} h_7 \\ h_8 \\ h_9 \\ h_{12} \\ h_{13} \\ h_{14} \\ h_{17} \\ h_{18} \\ h_{19} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -10 \\ -10 \\ -10 \end{Bmatrix}$$

The final equation system is of the form

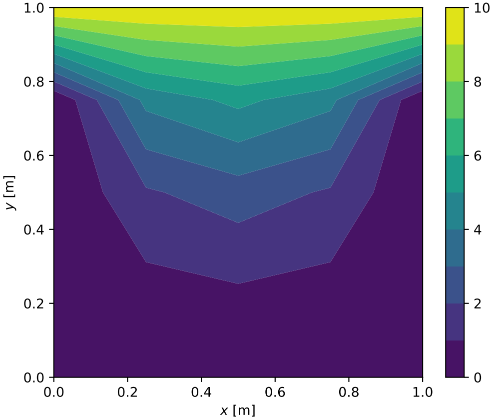
$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

where

$$\boldsymbol{A} = \left[\begin{array}{c|c|c} \boldsymbol{B} & \boldsymbol{I} & \mathbf{O} \\ \hline \boldsymbol{I} & \boldsymbol{B} & \boldsymbol{I} \\ \hline \mathbf{O} & \boldsymbol{I} & \boldsymbol{B} \end{array}\right]$$

and

$$\boldsymbol{B} = \begin{bmatrix} -4 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & -4 \end{bmatrix} \quad \text{and} \quad \boldsymbol{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Solution for $Nx = Ny = 4$.

Solution for an $N_x \times N_y$ grid:

Assume that the grid is uniform in both directions i.e. $\Delta x = \Delta y$.
Number of grid points in the computational domain:

$$(N_x + 1) \times (N_y + 1)$$

Coefficient matrix $\boldsymbol{A}$ with a similar structure:

$$\boldsymbol{A} = \left[\begin{array}{c|c|c|c|c} \boldsymbol{B} & \boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \hline \boldsymbol{I} & \boldsymbol{B} & \boldsymbol{I} & \ddots & \vdots \\ \hline \boldsymbol{0} & \boldsymbol{I} & \ddots & \ddots & \boldsymbol{0} \\ \hline \vdots & \ddots & \ddots & \boldsymbol{B} & \boldsymbol{I} \\ \hline \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{B} \end{array}\right]$$

Size of matrix $\boldsymbol{A}$:

$$(N_x - 1)^2 \times (N_y - 1)^2$$

Solution for an $N_x \times N_y$ grid:

Assume that the grid is uniform in both directions i.e. $\Delta x = \Delta y$.

Number of grid points in the computational domain:

$$(N_x + 1) \times (N_y + 1)$$

Coefficient matrix $\boldsymbol{A}$ with a similar structure:

$$\boldsymbol{A} = \left[\begin{array}{c|c|c|c|c} \boldsymbol{B} & \boldsymbol{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \boldsymbol{I} & \boldsymbol{B} & \boldsymbol{I} & \ddots & \vdots \\ \hline \mathbf{0} & \boldsymbol{I} & \ddots & \ddots & \mathbf{0} \\ \hline \vdots & \ddots & \ddots & \boldsymbol{B} & \boldsymbol{I} \\ \hline \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{I} & \boldsymbol{B} \end{array}\right]$$

Size of matrix $\boldsymbol{A}$:

$$(N_x - 1)^2 \times (N_y - 1)^2$$

Solution for an $N_x \times N_y$ grid:

Assume that the grid is uniform in both directions i.e. $\Delta x = \Delta y$. Number of grid points in the computational domain:

$$(N_x + 1) \times (N_y + 1)$$

Coefficient matrix A with a similar structure:

$$A = \left[\begin{array}{c|c|c|c|c} B & I & \mathbf{0} & \cdots & \mathbf{0} \\ \hline I & B & I & \ddots & \vdots \\ \hline \mathbf{0} & I & \ddots & \ddots & \mathbf{0} \\ \hline \vdots & \ddots & \ddots & B & I \\ \hline \mathbf{0} & \cdots & \mathbf{0} & I & B \end{array}\right]$$

Size of matrix A:

$$(N_x - 1)^2 \times (N_y - 1)^2$$

Solution for an $N_x \times N_y$ grid:

Assume that the grid is uniform in both directions i.e. $\Delta x = \Delta y$.
Number of grid points in the computational domain:

$$(N_x + 1) \times (N_y + 1)$$

Coefficient matrix $\boldsymbol{A}$ with a similar structure:

$$\boldsymbol{A} = \left[\begin{array}{c|c|c|c|c} \boldsymbol{B} & \boldsymbol{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \boldsymbol{I} & \boldsymbol{B} & \boldsymbol{I} & \ddots & \vdots \\ \hline \mathbf{0} & \boldsymbol{I} & \ddots & \ddots & \mathbf{0} \\ \hline \vdots & \ddots & \ddots & \boldsymbol{B} & \boldsymbol{I} \\ \hline \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{I} & \boldsymbol{B} \end{array}\right]$$

Size of matrix $\boldsymbol{A}$:

$$(N_x - 1)^2 \times (N_y - 1)^2$$

Matrices $\boldsymbol{B}$ and $\boldsymbol{I}$:

$$\boldsymbol{B} = \begin{bmatrix} -4 & 1 & 0 & \cdots & 0 \\ 1 & -4 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -4 & 1 \\ 0 & \cdots & 0 & 1 & -4 \end{bmatrix} \quad \text{and} \quad \boldsymbol{I} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

Sizes of matrices $\boldsymbol{B}$ and $\boldsymbol{I}$:

$$(N_x - 1) \times (N_y - 1)$$

RHS vector $\boldsymbol{b}$:

$$\boldsymbol{b} = \{0 \quad 0 \quad \cdots \quad 0 \quad -10 \quad \cdots \quad -10\}^{\mathsf{T}}$$

Matrices $\boldsymbol{B}$ and $\boldsymbol{I}$:

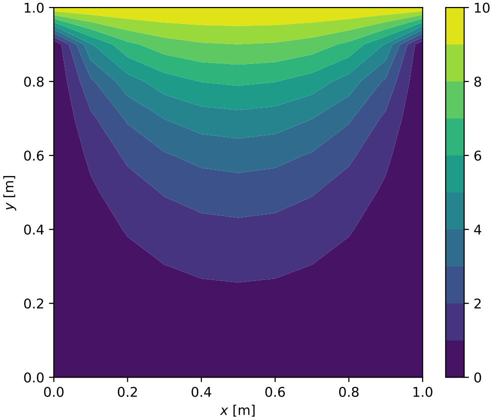
$$\boldsymbol{B} = \begin{bmatrix} -4 & 1 & 0 & \cdots & 0 \\ 1 & -4 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -4 & 1 \\ 0 & \cdots & 0 & 1 & -4 \end{bmatrix} \quad \text{and} \quad \boldsymbol{I} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

Sizes of matrices $\boldsymbol{B}$ and $\boldsymbol{I}$:

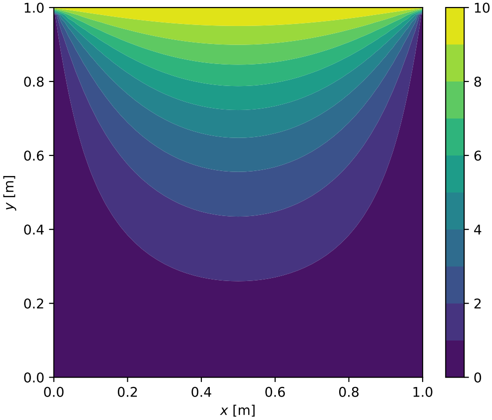
$$(N_x - 1) \times (N_y - 1)$$

RHS vector $\boldsymbol{b}$:

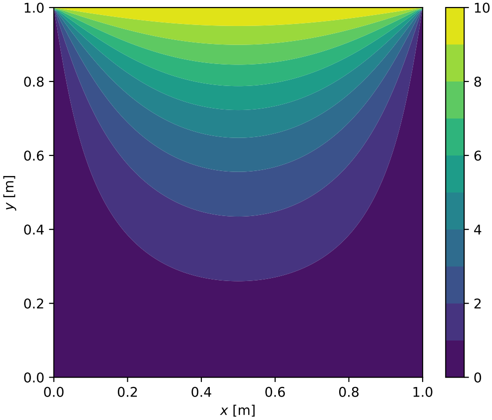
$$\boldsymbol{b} = \{0 \quad 0 \quad \cdots \quad 0 \quad -10 \quad \cdots \quad -10\}^\mathsf{T}$$



Solution for $Nx = Ny = 10$.



Solution for $Nx = Ny = 50$.



Solution for $Nx = Ny = 100$.