Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

Literature Study:
Study on deflation techniques and POD methods for the acceleration of Krylov subspace methods.

Report for the
Delft Institute of Applied Mathematics
as part of

MASTER OF SCIENCE
in
APPLIED MATHEMATICS

by

Jenny Tjan

Delft, Nederland
March 2018

MSc report APPLIED MATHEMATICS

Delft University of Technology

Jenny Tjan

Technische Universiteit Delft

Thesis advisors

Prof. dr. ir. C. Vuik Msc. G.B. Diaz Cortes

March, 2018 Delft

1 Abstract

The objective of this literature study report is to give an overview of the numerical methods used to model reservoir simulation. In particular, we focus on iterative linear solvers with preconditioner and deflation techniques. Simulating one-phase flow in a reservoir with heterogeneous porous media leads to a system of large linear equations after using discretization method. Those equations are derived from a mesoscopic model that uses mass-conservation law and Darcy's law to describe ground flow problem. The obtained linear equations are large and ill-conditioned, i.e. the matrix has high condition number. To solve this system is to use the Conjugate Gradient method. If the approach is insufficient, preconditioning techniques have to be used. Recently, Proper Orthogonal Decomposition (POD) based on system information has been found to be a good approach to accelerate the solving process further. The POD method constructs the basis matrix with the use of snapshots, known solutions of the linear system, to reduce the condition number. New POD based methods have been derived for this purpose. The first method is a deflation technique that uses the POD basis matrix is used as deflation-subspace matrix [1]. Another method to use the basis matrix is to construct a ROM-based preconditioner proposed by [2].

Preface

'Well begun is half done'

-Jenny Tjan, 2018

Contents

1 Abstract 4

2 Introduction 7

3 Preliminaries 8
- 3.1 Notation 8
- 3.2 Definition 8

4 Reservoir Simulation 9
- 4.1 Porous Media 9
- 4.2 Single-phase flow 9
  - 4.2.1 Mathematical Model 9
  - 4.2.2 Boundary Conditions 10
- 4.3 Incompressible Model 11
  - 4.3.1 Discretization 11
- 4.4 Compressible Model 12
  - 4.4.1 Constant Compressibility 12
- 4.5 Well Model 13

5 Iterative Numerical Methods 14
- 5.1 Newton-Raphson 14
- 5.2 Basic Iterative Method 16
- 5.3 Conjugate Gradient 16
- 5.4 Preconditioner 17
  - 5.4.1 Preconditioned Conjugate Gradient 17
- 5.5 Deflation Method 18
- 5.6 Proper Orthogonal Decomposition 19
- 5.7 Deflated Preconditioned Conjugated Gradient 20
- 5.8 ROM-based Preconditioner 22

6 Numerical Experiments 23
- 6.1 Incompressible model 23
- 6.2 Test Problems 24
  - 6.2.1 Constant Compressible model 24

7 Conclusion 26

8 Appendix 27

A List of Notation 27

2 Introduction

This report investigates one-phase flow through heterogeneous porous media on the mesoscopic scale. The mathematical general model is derived from mass-conservation law and Darcy's law. It is hard to obtain an accurate solution of the general problem and to simulate numerically. To model the flow problem and obtain a good approximation of the solution, it is sufficient to describe it with general trends in the reservoir flow pattern.

After discretizing the flow problem, we obtain a set of nonlinear equations. The equations will be linearized with Newton Rapson to obtain a linear system. The linear system is large and ill-conditioned, i.e. the matrix has high condition number. To solve the linear system, the iterative method Conjugate Gradient is used. The next step is to use preconditioning techniques to achieve faster convergence. The following step is to use deflation methods. For this method, the deflation subdomain matrix is needed and chosen so that the flow problem can be solved.

Recently, Proper Orthogonal Decomposition (POD) based on known information has been found to be a good approach to accelerate the solving process. The POD method requires snapshots, known solutions of the linear system, to constructs the basis matrix. This basis matrix is used as deflation subdomain matrix proposed by [1]. Also, the basis matrix can be used to construct a ROM-based preconditioner proposed by [2]. The different way to use the POD basis matrix is interesting to be investigated and compared.

The methods will be applied to incompressible case to get a basic idea about the convergence and amount of iterations are needed to solve the system. Thereafter, different test problems for the constant compressible case for the one-phase flow through porous media are given. These test problems will be more specified in the next report. In the end, the research question will be given in the conclusion.

3 Preliminaries

This section gives a brief introduction of linear algebra theory that will be used in this report.

3.1 Notation

The column vector $\mathbf{x} \in \mathbb{R}^n$ will be denoted as

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}. \tag{3.1}$$

The matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ will be denoted as

$$\mathbf{A} = \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{pmatrix}. \tag{3.2}$$

3.2 Definition

Definition 3.1. Let $\mathbf{A}$ be an $n \times n$ matrix. λ is called an eigenvalue of $\mathbf{A}$ if there exists an $\mathbf{v} \neq 0$ such that

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}. \tag{3.3}$$

The set of eigenvalues of $\mathbf{A}$ is given by

$$\lambda(\mathbf{A}) = \{\lambda_1, \dots, \lambda_n\}, \tag{3.4}$$

where λ_i is an eigenvalue of $\mathbf{A}$.

Definition 3.2. Let $\mathbf{A}$ be an $n \times n$ matrix, $\mathbf{A}$ is called symmetric positive definite (SPD) if for every $\mathbf{x} \in \mathbb{R}^n \backslash \{\mathbf{0}\}$

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} > 0. \tag{3.5}$$

$\mathbf{A}$ is called symmetric positive semi definite (SPSD) if for every $\mathbf{x} \in \mathbb{R}^n$

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} \geq 0. \tag{3.6}$$

Definition 3.3. Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, the inner product is defined as

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y}. \tag{3.7}$$

Definition 3.4. Let $\mathbf{A}$ be an $n \times n$ matrix, the 2-norm is defined as

$$\|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}(\mathbf{A}^\top \mathbf{A})}. \tag{3.8}$$

Definition 3.5. Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, and $\mathbf{A}$ is SPD, the $\mathbf{A}$-norm and $\mathbf{A}$-inner product is defined respectively as

$$\|\mathbf{x}\|_{\mathbf{A}} = \sqrt{\langle \mathbf{A}\mathbf{x}, \mathbf{x} \rangle} \quad \text{and} \quad \langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{A}} = \langle \mathbf{A}\mathbf{x}, \mathbf{y} \rangle. \tag{3.9}$$

Definition 3.6. Let $\mathbf{A}$ be an $n \times n$ matrix with eigenvalues $\lambda_1, \dots, \lambda_n$. The condition number of A is defined as

$$\kappa_2(\mathbf{A}) = \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2. \tag{3.10}$$

If $\mathbf{A}$ is SPD with real eigenvalues $\lambda_1, \dots, \lambda_n$, then

$$\kappa_2(\mathbf{A}) = \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}, \tag{3.11}$$

where $\lambda_{\max}(\mathbf{A}) = \max_{1 \leq i \leq n} \lambda_i$ and $\lambda_{\min}(\mathbf{A}) = \min_{1 \leq i \leq n} \lambda_i$.

4 Reservoir Simulation

Two models are needed to describe reservoir flow through porous media, which are the mathematical model and the geological model. The mathematical model consists of a set of partial differential equations to describe flow through porous media. The equations are derived from for mass-conservation law and Darcy's law, which will be more explained in Section 4.2.1.

4.1 Porous Media

The geological model describes the porous media rock formation and is constructed such that the model reproduce geological heterogeneity in the reservoir rock.The rock formation is defined by rock porosity φ, i.e. the ability of the rock to store fluid, and the rock permeability $\mathbf{K}$, i.e. the ability to transport fluid.
The porosity φ is defined as the percentage of void in the porous media and $1-\varphi$ is the percentage of solid material, i.e. rock matrix. There are interconnected pore space in the porous media where fluid can flow through and disconnected pores where fluid can only be stored. Since it is not interesting to look at disconnected pores, the effective porosity will be considered that only consider connected pores where fluid can flow through.
The rock permeability $\mathbf{K}$ describes the basic flow of porous media and it measures its ability to transmit a single fluid when the void space is filled with the fluid. Mathematically, the ability of a fluid to flow in a direction is described using a tensor.

This report will only consider a mesoscopic model of the problem. The fundamental equations of this model describe the fluid flow as continuity of fluid phases and uses Darcy's law to describe the speed of the fluid in porous media.

4.2 Single-phase flow

In this section, we will give a basic review of the mathematical model of a single-phase flow through porous media. The general model of the physical problem will be derived. However, obtaining an detailed solution of the general model requires a lot of computational time and the model is hard to simulate. To get a good approximation, it is sufficient to describe the physical problem with general trends in the reservoir flow pattern. Therefore, a few assumptions will be made such that it does not need a large amount of computer resources to model the problem.

4.2.1 Mathematical Model

The mathematical model of a single-phase in- and outflow though a porous medium is used to predict and analyse fluid flow while consider mass conservation.The equation is given in Equation (4.1):

$$\alpha\frac{\partial(\rho\varphi)}{\partial t}+\nabla(\alpha\rho\mathbf{v})=\alpha\rho q, \tag{4.1}$$

where $\rho(t,\mathbf{x})$ is the fluid density, $\alpha(\mathbf{x})$ is a geometric factor, g is the gravity constant and $q(t,\mathbf{x})$ is a source term. The geometric factor depends on the dimension of the problem. For 1D problem, we have $\alpha(x)=A(x)$, where A is the cross-sectional area. For 2D problem, the geometric factor $\alpha(\mathbf{x})=h(x,y)$, where h is the reservoir height. We only consider a 3D model of the problem, thus $\alpha(\mathbf{x})=1$. The mesoscopic model consider Darcy's velocity $\mathbf{v}(t,\mathbf{x})$ that is defined as

$$\mathbf{v}=-\frac{\mathbf{K}}{\mu}(\nabla p-\rho g\nabla d), \tag{4.2}$$

where $p(t, \mathbf{x})$ is the pressure, μ is the fluid viscosity, $\mathbf{K}(\mathbf{x})$ is the rock permeability and $d(\mathbf{x})$ is the reservoir depth. Combining (4.1) and (4.2) gives

$$\frac{\partial(\rho\varphi)}{\partial t} - \nabla\left(\rho\frac{\mathbf{K}}{\mu}(\nabla p - \rho g \nabla d)\right) = \rho q. \tag{4.3}$$

The fluid viscosity μ and rock permeability hardly depends on the pressure in our case, so they will be taken independent of the pressure and constant. Assuming the fluid density depends on the pressure, the liquid compressibility c_l can be defined as

$$c_l(p) := \frac{1}{\rho}\frac{\partial\rho}{\partial p}. \tag{4.4}$$

Similar, the relation between rock porosity and pressure can be defined with rock compressibility c_r

$$c_r(p) := \frac{1}{\varphi}\frac{\partial\varphi}{\partial p}. \tag{4.5}$$

Note that Equation (4.5) and (4.4) are first order ordinary differential equations. Let the total compressibility c_t be defined as

$$c_t = c_r + c_l. \tag{4.6}$$

Since fluid density and rock porosity depend on the pressure, the next relation can be obtained

$$\frac{\partial(\rho\varphi)}{\partial t} = \frac{\partial\rho}{\partial p}\frac{\partial p}{\partial t} + \frac{\partial\varphi}{\partial p}\frac{\partial p}{\partial t} = \rho\varphi\frac{\partial p}{\partial t}\left(\frac{1}{\rho}\frac{\partial\rho}{\partial p} + \frac{1}{\varphi}\frac{\partial\varphi}{\partial p}\right). \tag{4.7}$$

Then substitute Equation (4.6) in Equation (4.3) using Equation (4.7) to get the general result given in Equation (4.8).

The general nonlinear partial differential equation for the dependent variable pressure p is given by

$$c_t\rho\varphi\frac{\partial p}{\partial t} - \nabla\left(\rho\frac{\mathbf{K}}{\mu}(\nabla p - \rho g \nabla d)\right) = \rho q \tag{4.8}$$

The quantities and dimensions are given in appendix A.

4.2.2 Boundary Conditions

In reservoir simulation one would describe a closed flow system and provide boundary conditions to obtain an unique solution. For a closed flow system, the pressure related boundary conditions corresponds to Dirichlet boundary conditions. The homogeneous boundary condition is defined as:

$$p = 0 \quad \text{for } \mathbf{x} \in \partial\Omega, \tag{4.9}$$

where $\partial\Omega$ denotes the boundary of the Porous media Ω.
Another boundary condition that is often prescribed for this flow problem is in- and outflow related conditions, that corresponds to Neumann Boundary conditions:

$$\mathbf{v} \cdot \mathbf{n} = 0 \quad \text{for } \mathbf{x} \in \partial\Omega, \tag{4.10}$$

where $\mathbf{n}$ is defined as normal vector orthogonal to the boundary $\partial\Omega$.

The boundary conditions should be chosen such that the solution is well-posed.

4.3 Incompressible Model

The basic model of simulation one-phase flow through porous media is assuming the density and the porosity are pressure independent, i.e. $\frac{\partial \rho}{\partial p} = \frac{\partial \varphi}{\partial p} = 0$. Therefore, the incompressible model is also time-independent and Equation (4.8) becomes:

$$-\nabla\left(\rho\frac{\mathbf{K}}{\mu}(\nabla p - \rho g \nabla d)\right) = \rho q. \tag{4.11}$$

Assuming isotropic permeability, absence of gravity, fluid with constant velocity and density, Equation (4.11) becomes

$$-\frac{1}{\mu}\nabla(\mathbf{K}\nabla p) = q. \tag{4.12}$$

Equation (4.12) is an example of an elliptic equation with constant coefficients $\mu, \mathbf{K}$. This will be solved numerically and the numerical scheme will be given in the next section.

4.3.1 Discretization

The Method of lines is used to solve Equation (4.12). A finite difference scheme with cell central differences is used to approximate spatial derivatives. Assuming a uniform grid with grid size $\Delta x, \Delta y, \Delta z$ for the dimension x, y, z, respectively. Let (i, j, l) be the centre of the cell for the x-direction, y-direction and z-direction respectively. Also, the pressure in the cell (i, j, l) is defined as $p(x_i, y_j, z_l) = p_{i,j,l}$.

Equation 4.12 can be rewritten as

$$-\frac{1}{\mu}\left[\frac{\partial}{\partial x}\left(k\frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial p}{\partial y}\right) + \frac{\partial}{\partial z}\left(k\frac{\partial p}{\partial z}\right)\right] = q. \tag{4.13}$$

The first term in the equation in x-direction can be written as

$$\frac{\partial}{\partial x}\left(k\frac{\partial p}{\partial x}\right) \approx \frac{k_{i+\frac{1}{2},j,l}(p_{i+1,j,l} - p_{i,j,l}) - k_{i-\frac{1}{2},j,l}(p_{i,j,l} - p_{i-1,j,l})}{(\Delta x)^2} + \mathcal{O}\big((\Delta x)^2\big), \tag{4.14}$$

where $k_{i+\frac{1}{2},j,l}$ denote the harmonic averaging of grid-block permeabilities $(i+1, j, l)$ and (i, j, l) given by

$$k_{i+\frac{1}{2},j,l} = \frac{2}{\frac{1}{k_{i+1,j,l}} + \frac{1}{k_{i,j,l}}}. \tag{4.15}$$

Let the transmissibility between cell $(i+1, j, l)$ and (i, j, l) be given by

$$T_{i+\frac{1}{2},j,l} := \frac{1}{\mu}\frac{2\Delta y \Delta z}{\Delta x}k_{i+\frac{1}{2},j,l}. \tag{4.16}$$

Similarly expression can be obtained for the y, z-direction.

For a cell (i, j, l) the discretisation of Equation (4.13) is given by

$$\begin{aligned} &-p_{i-1,j,l}T_{i-\frac{1}{2},j,l} - p_{i,j-1,l}T_{i,j-\frac{1}{2},l} - p_{i,j,l-1}T_{i,j,l-\frac{1}{2}} \\ +p_{i-1,j,l}\Big(&T_{i-\frac{1}{2},j,l} + T_{i,j-\frac{1}{2},l} + T_{i,j,l-\frac{1}{2}} + T_{i+\frac{1}{2},j,l} + T_{i,j+\frac{1}{2},l} + T_{i,j,l+\frac{1}{2}}\Big) \\ &-p_{i+1,j,l}T_{i+\frac{1}{2},j,l} - p_{i,j+1,l}T_{i,j+\frac{1}{2},l} - p_{i,j,l+1}T_{i,j,l+\frac{1}{2}} \quad = \Delta x \Delta y \Delta z \; q_{i,j,l}. \end{aligned} \tag{4.17}$$

The transmissibility matrix $\mathbf{T}$ can be defined with the given boundary conditions. In the end, Equation (4.12) can be written as

$$\mathbf{T}\mathbf{p} = \mathbf{q}, \tag{4.18}$$

which is a system of linear equations.

4.4 Compressible Model

For the compressible model, it is not easy to derive the discretization to solve the problem numerically. The compressible model is, unlike the incompressible model, time-dependent. It means that it would take more computational time to get results. Like mentioned before, it would take a lot of computer resources to obtain a solution of the flow problem. Therefore, only the constant compressible model would be explained. More information can be found in [4].

4.4.1 Constant Compressibility

Assuming that the fluid density and rock porosity are constant compressible, i.e. $c_l, c_r \in \mathbb{R}$. The, the total compressibility is also constant. Therefore, the fluid density and porosity are linearly dependent on the pressure. The initial condition for the pressure is defined as $p\big|_{t=0} = p_0$, without loss of generality let $p_0 = 0$. Then, the initial conditions for rock porosity and fluid density are:

$$\rho\Big|_{p=p_0} = \rho_0 \quad \text{and} \quad \varphi\Big|_{p=p_0} = \varphi_0. \tag{4.19}$$

Inserting the initial conditions in Equation (4.5) and Equation (4.4) gives:

$$\varphi = \varphi_0 e^{c_r p} \quad \text{and} \quad \rho = \rho_0 e^{c_l p}. \tag{4.20}$$

For small values of the fluid compressibility, the fluid density can be written by using linearization

$$\rho \approx \rho_0(1 + c_l p). \tag{4.21}$$

If the rock porosity is pressure independent, Equation (4.8) can be written as

$$\varphi \frac{\partial \rho(p)}{\partial t} - \nabla\left(\rho(p)\frac{\mathbf{K}}{\mu}(\nabla p - \rho(p) g \nabla d)\right) = \rho(p) q. \tag{4.22}$$

Assuming isotropic permeability and absence of gravity and fluid with constant velocity results in

$$\varphi \frac{\partial \rho(p)}{\partial t} - \frac{\rho_0}{\mu}\nabla(\mathbf{K}\nabla p) - \frac{\rho_0 c_l}{\mu}\nabla(p\mathbf{K}\nabla p) = \rho(p) q. \tag{4.23}$$

If the fluid compressibility is sufficient small, in the sense of $c_l \nabla(p\mathbf{K}\nabla p) \ll \nabla(\mathbf{K}\nabla p)$, the term $c_l \nabla(p\mathbf{K}\nabla p)$ can be neglected. In the end, the result is

$$\varphi \frac{\partial \rho(p)}{\partial t} - \frac{\rho_0}{\mu}\nabla(\mathbf{K}\nabla p) = \rho(p) q. \tag{4.24}$$

Discretization

The difference between Equation (4.12) and Equation (4.24) is the time dependence. Equation (4.24) can be discretizated in the same manner as in the incompressible case, and written in the form:

$$\varphi \frac{\partial \boldsymbol{\rho}(\mathbf{p})}{\partial t} + \mathbf{T}\mathbf{p} = \bar{\mathbf{q}}(\mathbf{p}), \tag{4.25}$$

where the source term is defined as $\bar{\mathbf{q}}(\mathbf{p}) = \boldsymbol{\rho}(\mathbf{p})\mathbf{q}$ and $\mathbf{T}$ is the transmissibility matrix. In this case, Euler Backwards will be used to solve this system and will be rewritten as

$$\mathbf{V}\frac{\boldsymbol{\rho}(\mathbf{p}^{k+1}) - \boldsymbol{\rho}(\mathbf{p}^{k})}{\Delta t^k} + \mathbf{T}\mathbf{p}^{k+1} = \bar{\mathbf{q}}(\mathbf{p}^{k+1}), \tag{4.26}$$

where $\Delta t^k = t^{k+1} - t^k$ and $\mathbf{V}$ is the accumulation matrix defined as

$$\mathbf{V} = \Delta x \Delta y \Delta z \varphi \mathbf{I}_n, \tag{4.27}$$

where $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is the identity matrix.

4.5 Well Model

Usually, in reservoir simulation, the closed flow system is described in combination with a well model as source term. Fluids are injected or produced in a well at constant bottom-hole pressure or a constant rate. The inflow performance is defined by the bottom-hole pressure with surface flow rate. The simplest model is the Peaceman linear model is defined by [3, 4]

$$q_{i,j,l} = J(p_{i,j,l} - pbh_{i,j,l}), \tag{4.28}$$

where $pbh_{i,j,l}$ is the bottom-hole pressure in cell (i, j, l) and J is the productivity index.

5 Iterative Numerical Methods

The partial differential equation has been discretized in the following form:

$$\mathbf{V}\frac{\mathrm{d}\boldsymbol{\rho}(\mathbf{p})}{\mathrm{d}t} + \mathbf{T}\mathbf{p} = \mathbf{q}(\mathbf{p}), \tag{5.1}$$

where $\mathbf{V} \in \mathbb{R}^{n\times n}$ is the accumulation matrix which is strictly positive, $\mathbf{T}$ is the transmissibility matrix which is SPD, $\mathbf{p}$ is the pressure which is unknown and $\mathbf{q}$ is the source vector. The unknown time variable $\frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t}$ is approximated by using the Euler Backwards method. Let the time step size be defined by $\Delta t^k = t^{k+1} - t^k$. Equation (5.1) is discretized in time by

$$\mathbf{V}\frac{\boldsymbol{\rho}(\mathbf{p}^{k+1}) - \boldsymbol{\rho}(\mathbf{p}^k)}{\Delta t^k} + \mathbf{T}\mathbf{p}^{k+1} = \mathbf{q}(\mathbf{p}^{k+1}). \tag{5.2}$$

The equation is nonlinear and is solved to find the unknown pressure $\mathbf{p}$ by using linearization methods, i.e. Newton-Raphson. For every timestep, it can be written as a system of linear equations in the form of

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{5.3}$$

where $\mathbf{A}$ is a large SPD matrix, which makes it suitable to use iterative methods. This section will start with explaining Newton-Raphson and defining $\mathbf{A}$ more precisely. Thereafter, iteration methods like the Conjugate Gradient, preconditioner techniques and deflation methods will be explained. Hereafter, an overview of the POD method will be given and show that it could be used as deflation-subspace matrix and preconditioner.

5.1 Newton-Raphson

The Newton-Raphson method is used to linearize nonlinear equations. First, for an one-dimensional case, function $h(x)$ would be defined such that $h(x) = 0$. The iteration steps are found by using a Taylor expansion. Start with an initial guess x^0 and for each the iteration step, compute

$$x^{k+1} = x^k - \frac{h(x^k)}{h'(x^k)}, \tag{5.4}$$

while assuming $h'(x^k) \neq 0$ for every step k. Depending on the choice of the initial guess, this method will converge.

For the multidimensional case, the same process can be used. Let $\mathbf{f}(\mathbf{x})$ be an n-dimensional function. Assume $\mathbf{x}^* = \mathbf{x}^k + \delta\mathbf{x}$ where $\mathbf{f}(\mathbf{x}^*) = 0$, then the Taylor expansion around point $\mathbf{x}^k$ is

$$\mathbf{f}(\mathbf{x}^k + \delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}^k) + \mathbf{J}_{\mathbf{f}}(\mathbf{x}^k)\delta\mathbf{x}, \tag{5.5}$$

where $\mathbf{J}_{\mathbf{f}}$ is de Jacobian of $\mathbf{f}$. Recall, $\mathbf{f}(\mathbf{x}^*) = 0$, thus to find $\delta\mathbf{x}$ one need to solve the linear system

$$\mathbf{J}_{\mathbf{f}}(\mathbf{x}^k)\delta\mathbf{x} = -\mathbf{f}(\mathbf{x}^k). \tag{5.6}$$

Thereafter, update $\mathbf{x}^{k+1} = \mathbf{x}^k + \delta\mathbf{x}$. The algorithm for every iteration is defined as

Algorithm 1 Newton-Raphson

```
Initial: $\mathbf{p}^0, \varepsilon$
while $|\mathbf{p}^{k+1} - \mathbf{p}^k| > \varepsilon$ do
    Solve: $\mathbf{J}_{\mathbf{f}}(\mathbf{p}^k)\delta\mathbf{p} = -\mathbf{f}(\mathbf{p}^k)$
    Update: $\mathbf{p}^{k+1} = \mathbf{p}^k + \delta\mathbf{p}$
    $k = k + 1$
```

Example

The heat equation with nonlinear source term is defined as

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + T(T-1) \text{ for } 0 < x < 1,\ \ t > 0, \tag{5.7}$$

with homogeneous Dirichlet boundary conditions $T(0) = T(1) = 0$.
To illustrate how Newton-Raphson works, only the steady state of the problem will be solved, i.e. $\frac{\partial T}{\partial t} = 0$. The analytic solution for this problem is

$$T(x) = 0 \quad \text{for } 0 < x < 1. \tag{5.8}$$

To solve this problem numerically, we use a uniform gridsize $n = 4$ with $\Delta x = 0.25$. Hence, the function $\mathbf{f}(\mathbf{T})$ can be defined as

$$\begin{aligned} \mathbf{f}(\mathbf{T}) &= \frac{1}{\Delta x^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix} \\ &+ \begin{bmatrix} T_1(T_1-1) & & & & \\ & T_2(T_2-1) & & & \\ & & T_3(T_3-1) & & \\ & & & T_4(T_4-1) & \\ & & & & T_5(T_5-1) \end{bmatrix}. \end{aligned} \tag{5.9}$$

Newton-Raphson will be used to solve $\mathbf{f}(\mathbf{T})$. The Jacobian matrix is defined as

$$\mathbf{J_f}(\mathbf{T}) = \frac{1}{\Delta x^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix} + 2\begin{bmatrix} T_1 & & & & \\ & T_2 & & & \\ & & T_3 & & \\ & & & T_4 & \\ & & & & T_5 \end{bmatrix} - \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}. \tag{5.10}$$

Choose as initial condition

$$\mathbf{T}_{\text{int}} = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}^\top \tag{5.11}$$

with stop criteria 10^{-4}. After 6 iterations the steady state solution is

$$\mathbf{T}_{\text{ss}} \approx \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top, \tag{5.12}$$

with error $2.4559 \cdot 10^{-31}$. The solution found with the numerical scheme is close to the analytic solution with a small error.

For the original problem, Equation (5.2) is nonlinear and multidimensional. Define the function

$$\mathbf{f}(\mathbf{p}^{k+1}; \mathbf{p}^k) = \mathbf{V}\frac{\rho(\mathbf{p}^{k+1}) - \rho(\mathbf{p}^k)}{\Delta t^k} + \mathbf{T}\mathbf{p}^{k+1} - \bar{\mathbf{q}}(\mathbf{p}^{k+1}). \tag{5.13}$$

This will be used to find the solution for the pressure $\mathbf{p}$.

5.2 Basic Iterative Method

When matrices are very large, it is time-consuming to solve the system $\mathbf{Ax} = \mathbf{b}$ with direct solution methods. Therefore, another way to solve the system is by using iterative methods. The basic iterative method goes as follows: Split $\mathbf{A} = \mathbf{M} - \mathbf{N}$ such that $\mathbf{M}^{-1}$ exists. The iterative condition for $\mathbf{x}$ can be derived from

$$\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{Mx} = \mathbf{b} + \mathbf{Nx}. \tag{5.14}$$

Thus, the result is

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{M}^{-1}\mathbf{r}^k, \tag{5.15}$$

where $\mathbf{r}^k = \mathbf{b} - \mathbf{Ax}^k$ is the residual. The residual denotes the difference between the iterative solution and true solution.

There are different choices for $\mathbf{M}$. The Jacobi method uses $\mathbf{M} = diag(\mathbf{A})$ and Gauss-Seidel uses $\mathbf{M} = \mathbf{L}$, where $\mathbf{L}$ is the lower triangle of $\mathbf{A}$.

The iterative method goes as follows: Choose initial guess $\mathbf{x}^0$ and after k iterations the iterative solution can be written as.

$$\begin{aligned}
\mathbf{x}^0 &= \mathbf{x}^0 \\
\mathbf{x}^1 &= \mathbf{x}^1 + \mathbf{M}^{-1}\mathbf{r}^1 = \mathbf{x}^0 + \mathbf{M}^{-1}\mathbf{r}^0 \\
\mathbf{x}^2 &= \mathbf{x}^2 + \mathbf{M}^{-1}\mathbf{r}^2 \\
&= \ldots = \mathbf{x}^0 + \mathbf{M}^{-1}\mathbf{A}\mathbf{M}^{-1}\mathbf{r}^0 + 2\mathbf{M}^{-1}\mathbf{r}^0 \\
&\text{etc.}
\end{aligned}$$

It follows that the iterative solution can be written as

$$\mathbf{x}^k = \mathbf{x}^0 + \text{span}\Big\{\mathbf{M}^{-1}\mathbf{r}^0, \mathbf{M}^{-1}\mathbf{A}\mathbf{M}^{-1}\mathbf{r}^0, \ldots, (\mathbf{M}^{-1}\mathbf{A})^{k-1}\mathbf{M}^{-1}\mathbf{r}^0\Big\}. \tag{5.16}$$

The Krylov subspace of dimension k is defined as

$$\mathcal{K}_k(\mathbf{A}, \mathbf{r}) := \text{span}\Big\{\mathbf{Ar}, \mathbf{A}^2\mathbf{r}, \ldots, \mathbf{A}^{k-1}\mathbf{r}\Big\}. \tag{5.17}$$

Hence, the iterative solution can be written as

$$\mathbf{x}^k = \mathbf{x}^0 + \mathcal{K}_k(\mathbf{M}^{-1}\mathbf{A}, \mathbf{M}^{-1}\mathbf{r}^0). \tag{5.18}$$

The matrix $\mathbf{M}$ is also called a *preconditioner*, which will be explained later. In the following section, the Conjugate Gradient method will be explained by using $\mathbf{M} = \mathbf{I}$.

5.3 Conjugate Gradient

Conjugate Gradient (CG) is an iterative method that is used for SPD matrices. The purpose of CG is to construct a sequence $\{\mathbf{x}^k\}_k$ such that it minimizes the $\mathbf{A}$-norm of the error:

$$\min_{\mathbf{x}^k \in \mathcal{K}_k(\mathbf{A}, \mathbf{r}^0)} \|\mathbf{x} - \mathbf{x}^k\|_{\mathbf{A}}, \tag{5.19}$$

where $\mathbf{x}$ is the true solution. CG uses search vectors $\big\{\mathbf{p}^i\big\}_i$ that are defined such that $\big\langle \mathbf{Ap}^i, \mathbf{p}^j \big\rangle = 0$ for every $i \neq j$. Also, the residuals should be orthogonal hence $\big\langle \mathbf{r}^i, \mathbf{r}^j \big\rangle = 0$ for every $i \neq j$. With every iteration step, there will be updates for the solution and residual defined as

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k\mathbf{p}^k \quad \text{and} \quad \mathbf{r}^{k+1} = \mathbf{r}^k - \alpha^k\mathbf{Ap}^k, \tag{5.20}$$

respectively, where α^k is chosen such that it minimizes Equation (5.19). Therefore $\alpha^k = \frac{\langle \mathbf{r}^k, \mathbf{r}^k \rangle}{\langle \mathbf{Ap}^k, \mathbf{p}^k \rangle}$. The search vectors are updated as

$$\mathbf{p}^{k+1} = \mathbf{r}^{k+1} + \beta^k\mathbf{p}^k. \tag{5.21}$$

The method is summarized in Algorithm 2 and can be found in [5, 6].

Algorithm 2 Conjugate Gradient

```
Initial: $\mathbf{x}^0, \varepsilon$
Compute: $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0$ and $\mathbf{p}^0 = \mathbf{r}^0$
for $k = 0, \dots$ do
    while $\mathbf{r}^k > \varepsilon$ do
        $\alpha^k = \frac{\langle \mathbf{r}^k, \mathbf{r}^k \rangle}{\langle \mathbf{A}\mathbf{p}^k, \mathbf{p}^k \rangle}$
        $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{p}^k$
        $\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha^k \mathbf{A}\mathbf{p}^k$
        $\beta^k = \frac{\langle \mathbf{r}^{k+1}, \mathbf{r}^{k+1} \rangle}{\langle \mathbf{r}^k, \mathbf{r}^k \rangle}$
        $\mathbf{p}^{k+1} = \mathbf{r}^{k+1} + \beta^k \mathbf{p}^k$
```

Convergence

After k iterations, the error in the $\mathbf{A}$-norm is bounded by

$$\|\mathbf{x} - \mathbf{x}^k\|_{\mathbf{A}} \leq 2\|\mathbf{x} - \mathbf{x}^0\|_{\mathbf{A}} \left(\frac{\sqrt{\kappa_2(\mathbf{A})} - 1}{\sqrt{\kappa_2(\mathbf{A})} + 1} \right)^k . \tag{5.22}$$

The proof can be found in [7].

5.4 Preconditioner

The convergence depends on the condition number of the matrix. Preconditioners can be used to achieve a faster convergence by reducing the condition number. The preconditioner matrix $\mathbf{M}$ is applied to the system $\mathbf{A}\mathbf{x} = \mathbf{b}$ as

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}. \tag{5.23}$$

The system given in Equation (5.23), $\mathbf{M}^{-1}\mathbf{A}$ is not necessary SPD, thus the system is redefined to

$$\overline{\mathbf{A}}\overline{\mathbf{x}} = \overline{\mathbf{b}}, \tag{5.24}$$

where $\overline{\mathbf{A}} = \mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{M}^{-\frac{1}{2}}$, $\overline{\mathbf{x}} = \mathbf{M}^{\frac{1}{2}}\mathbf{x}$ and $\overline{\mathbf{b}} = \mathbf{M}^{-\frac{1}{2}}\mathbf{b}$. With extra conditions that $\mathbf{M}$ should be SPD and $\mathbf{M}^{-\frac{1}{2}}$ exists and is symmetric. It follows that $\overline{\mathbf{A}}$ is SPD, proof can be found in [7]. There are many choices that can be used as preconditioner. If $\mathbf{M} = \mathbf{I}$, then this is the iterative method from before and the condition number remain unchanged. If $\mathbf{M} = \mathbf{A}$, the condition number is equal to 1 and the solution can be found in one step. It is often hard to compute $\mathbf{A}^{-1}$, so it is often not chosen as preconditioner.

5.4.1 Preconditioned Conjugate Gradient

The new system using a preconditioner is defined as

$$\overline{\mathbf{A}}\overline{\mathbf{x}} = \overline{\mathbf{b}}, \tag{5.25}$$

where $\overline{\mathbf{A}}$ is SPD, thus Conjugate Gradient algorithm can be used. The derivation of this method, also called Preconditioned Conjugate Gradient (PCG), can also be found in [7] and is given in Algorithm 3.

To use preconditioned Conjugate Gradient method in practise, it is needed that $\mathbf{M}^{-1}$ is inexpensive to apply and cheap to compute.

Algorithm 3 Preconditioned Conjugate Gradient

```
Initial: $\mathbf{x}^0, \varepsilon$
Compute: $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0, \mathbf{z}^0 = \mathbf{M}^{-1}\mathbf{r}^0$ and $\mathbf{p}^0 = \mathbf{z}^0$
for $k = 0, \dots$ do
    while $\mathbf{r}^k > \varepsilon$ do
        $\mathbf{z}^{k+1} = \mathbf{M}^{-1}\mathbf{r}^k$
        $\alpha^k = \frac{\langle \mathbf{r}^k, \mathbf{z}^k \rangle}{\langle \mathbf{A}\mathbf{p}^k, \mathbf{p}^k \rangle}$
        $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{p}^k$
        $\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha^k \mathbf{A}\mathbf{p}^k$
        $\beta^k = \frac{\langle \mathbf{z}^{k+1}, \mathbf{r}^{k+1} \rangle}{\langle \mathbf{z}^k, \mathbf{r}^k \rangle}$
        $\mathbf{p}^{k+1} = \mathbf{z}^{k+1} + \beta^k \mathbf{p}^k$
```

Convergence

The error is bounded by the next inequality:

$$\|\mathbf{x} - \mathbf{x}^k\|_{\mathbf{A}} \leq 2\|\mathbf{x} - \mathbf{x}^0\|_{\mathbf{A}} \left(\frac{\sqrt{\kappa_2(\mathbf{M}^{-1}\mathbf{A})} - 1}{\sqrt{\kappa_2(\mathbf{M}^{-1}\mathbf{A})} + 1} \right)^k . \tag{5.26}$$

The advantages of choosing the right preconditioner ensures that the condition number is being reduced and a faster convergence is achieved.

Incomplete Decomposition

This report uses Incomplete Cholesky as preconditioner, that will be denoted by $\mathbf{M}_{IC0}$. An Incomplete Cholesky is an SPD approximation of the Cholesky factorization where the amount of fill in can be chosen. This entails a decomposition of the form $\mathbf{A} = \mathbf{L}\mathbf{L}^\top - \mathbf{A}_r$, where $\mathbf{L}$ is the lower triangle with the same zero pattern as matrix $\mathbf{A}$ and $\mathbf{A}_r$ is the residual or error of the factorization. The matrix $\mathbf{A}$ is approximated with $\mathbf{L}\mathbf{L}^\top$ and we define the Incomplete Cholesky as preconditioner matrix as $\mathbf{M}_{IC0} = \mathbf{L}\mathbf{L}^\top$. More information can be found in [6].

5.5 Deflation Method

Even when using a preconditioner, the eigenvalues of the system $\overline{\mathbf{A}}\overline{\mathbf{x}} = \overline{\mathbf{b}}$ are not always favorable. Thus, it would hardly make any difference in performance by using PCG. The deflation method reduces the condition number by setting the extreme eigenvalues equal to zero such that the convergence bound is small. The method is defined by using the next definition:

Definition 5.1. Given an $n \times n$-matrix $\mathbf{A}$ which is SPD, given a deflation-subspace matrix $\mathbf{Z}$ of size $n \times m$ where $m \ll n$. The deflation method is defined as

$$\mathbf{P} = \mathbf{I} - \mathbf{A}\mathbf{Q} \qquad \mathbf{P} \in \mathbb{R}^{n\times n}, \quad \mathbf{Q} \in \mathbb{R}^{n\times n}, \tag{5.27}$$

where $\mathbf{Q} = \mathbf{Z}\mathbf{E}^{-1}\mathbf{Z}^\top$ with $\mathbf{Z} \in \mathbb{R}^{n\times m}$, $\mathbf{E} \in \mathbb{R}^{m\times m}$ and $\mathbf{E} = \mathbf{Z}^\top\mathbf{A}\mathbf{Z}$.

The columns of the deflation-subspace matrix $\mathbf{Z}$ are called deflation vectors. The vectors are chosen such that the matrix $\mathbf{E}$, also known as coarse matrix, is nonsingular. In Section 5.6, more details will be given how matrix $\mathbf{Z}$ will be constructed. Note that $\mathbf{P}\mathbf{A}\mathbf{Z} = \mathbf{0}_{n,k}$.

Proof. Let $\mathbf{P}, \mathbf{Q}, \mathbf{Z}$ be defined as in Definition 5.1, then

$$\mathbf{P}\mathbf{A}\mathbf{Z} = (\mathbf{I} - \mathbf{A}\mathbf{Q})\mathbf{A}\mathbf{Z} = \mathbf{A}\mathbf{Z} - \mathbf{A}\mathbf{Q}\mathbf{A}\mathbf{Z} = \mathbf{A}\mathbf{Z} - \mathbf{A}\mathbf{Z} = \mathbf{0}_{n,k}. \tag{5.28}$$

This follows from

$$\mathbf{QAZ} = \mathbf{ZE}^{-1}\mathbf{Z}^\top\mathbf{AZ} = \mathbf{ZE}^{-1}\mathbf{E} = \mathbf{Z}. \tag{5.29}$$

□

Thus **PA** is a singular matrix since it contains zero eigenvalues. Hence, after using the deflation method, the system can be written as

$$\mathbf{PA}\hat{\mathbf{x}} = \mathbf{Pb}, \tag{5.30}$$

where $\hat{\mathbf{x}}$ is the non unique solution.
The solution of the original system $\mathbf{Ax} = \mathbf{b}$ is found by using

$$\mathbf{x} = \mathbf{Qb} + \mathbf{P}^\top\hat{\mathbf{x}}, \tag{5.31}$$

follows from [7].

5.6 Proper Orthogonal Decomposition

The objective of the report is to use the Proper Orthogonal Decomposition (POD) method to accelerate the solution of the linear system $\mathbf{Ax} = \mathbf{b}$. Hence, an overview will be given in this section. The POD method is a Model Order Reduction-based method (MOR), which reduces a large system into a smaller system such that it is easier to solve. The solution of the system can be approximated by

$$\mathbf{x} \approx \sum_{i=1}^{m} c_i \boldsymbol{\psi}_i, \tag{5.32}$$

where $c_i \in \mathbb{R}$ and $\{\boldsymbol{\psi}_i\}_i$ are basis vectors of the basismatrix $\boldsymbol{\Psi}$, which will be specified later on.

The basis is constructed from known solutions $\mathbf{x}_i$, also called snapshots, of the system $\mathbf{Ax}_i = \mathbf{b}_i$, where the right hand space $\mathbf{b}_i$ is changed. The correlation matrix is defined as follows:

$$\mathbf{R} = \frac{1}{m}\mathbf{XX}^\top, \tag{5.33}$$

where $\mathbf{X} = \begin{bmatrix}\mathbf{x}_1 & \dots & \mathbf{x}_m\end{bmatrix}$. Note that the correlation matrix $\mathbf{R} \in \mathbb{R}^{n\times n}$ is SPSD.

Proof. Let $\mathbf{y} \in \mathbb{R}^n$, then

$$\mathbf{y}^\top\mathbf{Ry} = \mathbf{y}^\top\frac{1}{m}\mathbf{XX}^\top\mathbf{y} = \frac{1}{m}\left(\mathbf{X}^\top\mathbf{y}\right)\left(\mathbf{X}^\top\mathbf{y}\right) = \frac{1}{m}\left(\mathbf{X}^\top\mathbf{y}\right)^2 \geq 0, \tag{5.34}$$

where $m > 0$. Also,

$$\mathbf{R}^\top = \left(\frac{1}{m}\mathbf{XX}^\top\right)^\top = \frac{1}{m}\mathbf{XX}^\top = \mathbf{R}. \tag{5.35}$$

Hence **R** is SPSD. □

The eigenvectors of **R** are used as vectors for the basis matrix $\boldsymbol{\Psi}$. Note that $\mathbf{R} \in \mathbb{R}^{n\times n}$, instead of computing the eigenvalues and eigenvectors of the correlation matrix **R**, it is easier to find the eigenvalues and eigenvectors of

$$\tilde{\mathbf{R}} := \frac{1}{m}\mathbf{X}^\top\mathbf{X}, \tag{5.36}$$

since the dimension is $m \times m$ and $m \ll n$. Assume $\tilde{\mathbf{R}}$ has eigenvalues defined as

$$\lambda_1 > \lambda_2 > \ldots > \lambda_m. \tag{5.37}$$

The relation between the eigenvectors of $\tilde{\mathbf{R}}$ and $\mathbf{R}$ is as follows. If $\mathbf{v}$ is an eigenvector of $\tilde{\mathbf{R}}$, then $\mathbf{Xv}$ is an eigenvector of $\mathbf{R}$. Not every eigenvector is used as basis vector, the dimension is chosen such that it only represents the l largest eigenvalues of $\mathbf{R}$, where $l \ll m \ll n$. The quantity l is chosen such that the next equality holds

$$\frac{\max\limits_{1\leq l\leq m} \sum\limits_{i=1}^{l} \lambda_i(\mathbf{R})}{\sum\limits_{i=1}^{m} \lambda_i(\mathbf{R})} \leq \alpha, \tag{5.38}$$

where $0 < \alpha \leq 1$ is close to 1. Therefore, the basis matrix $\mathbf{\Psi} \in \mathbb{R}^{n\times l}$ is defined as

$$\mathbf{\Psi} := \begin{bmatrix} \boldsymbol{\psi}_1 & \dots & \boldsymbol{\psi}_l \end{bmatrix}, \tag{5.39}$$

where $\{\boldsymbol{\psi}_i\}_i$ are eigenvectors of the matrix $\mathbf{R}$.

In the next two sections, this matrix is used in different ways. First, it will be used as deflation-subspace matrix before applying the preconditioner (section 5.7). Thereafter, it will be used to construct a preconditioner (section 5.8). The basis matrix will be denoted with $\mathbf{Z}$ instead of $\mathbf{\Psi}$.

5.7 Deflated Preconditioned Conjugated Gradient

This proposal was given by [1] and the method uses the basis matrix as deflation-subspace matrix to reduce the amount of iterations required to solve the system. As mentioned in Section 5.5 and Section 5.6, the deflation-subspace matrix is chosen as the eigenvectors corresponding to the largest eigenvalues of the matrix $\mathbf{R}$. It is important to choose good deflation vectors and since they contain relevant information about the spectral radius in a few vectors [1, 7]. The system $\mathbf{Ax} = \mathbf{b}$ is solved be defining the next system:

$$\tilde{\mathbf{P}}\tilde{\mathbf{A}}\tilde{\hat{\mathbf{x}}} = \tilde{\mathbf{P}}\tilde{\mathbf{b}}, \tag{5.40}$$

with

$$\tilde{\mathbf{A}} := \mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{M}^{-\frac{1}{2}}, \quad \tilde{\hat{x}} := \mathbf{M}^{\frac{1}{2}}\hat{\mathbf{x}}, \quad \tilde{\mathbf{b}} := \mathbf{M}^{-\frac{1}{2}}\mathbf{b} \tag{5.41}$$

and

$$\tilde{\mathbf{P}} := \mathbf{I} - \tilde{\mathbf{A}}\tilde{\mathbf{Q}}, \quad \tilde{\mathbf{Q}} := \tilde{\mathbf{Z}}\tilde{\mathbf{E}}^{-1}\tilde{\mathbf{Z}}^\top, \quad \tilde{\mathbf{E}} := \tilde{\mathbf{Z}}^\top\tilde{\mathbf{A}}\tilde{\mathbf{Z}}, \tag{5.42}$$

where $\tilde{\hat{\mathbf{x}}}$ is the nonunique deflation solution. The true solution of the deflation method can be found with:

$$\tilde{\mathbf{x}} := \tilde{\mathbf{Q}}\tilde{\mathbf{b}} + \tilde{\mathbf{P}}^\top\tilde{\hat{\mathbf{x}}}. \tag{5.43}$$

Therefore the true solution of the system $\mathbf{Ax} = \mathbf{b}$ is

$$\mathbf{x} = \mathbf{M}^{-\frac{1}{2}}\tilde{\mathbf{x}}. \tag{5.44}$$

This can be summarized and given in Algorithm 4.
Algorithm 4 is not being used since it is not practical. The more practical algorithm is given by [7] and is found in Algorithm 5.

Accuracy/Convergence

By using this method, the smallest eigenvalue will be equal to zero, thus another condition number will be defined for the convergence.

Algorithm 4 Deflated Preconditioned Conjugated Gradient

```
Initial: $\hat{\mathbf{x}}^0, \varepsilon$
Compute: $\tilde{\mathbf{r}}^0 = \tilde{\mathbf{b}} - \tilde{\mathbf{A}}\tilde{\mathbf{x}}^0, \hat{\tilde{\mathbf{r}}}^0 = \tilde{\mathbf{P}}\tilde{\mathbf{r}}^0$ and $\tilde{\mathbf{p}}^0 = \hat{\tilde{\mathbf{r}}}^0$
for $k = 0, \ldots$ do
    while $\tilde{\mathbf{r}}^k > \varepsilon$ do
        $\hat{\tilde{\mathbf{z}}}^k = \tilde{\mathbf{P}}\tilde{\mathbf{A}}\tilde{\mathbf{p}}^k$
        $\alpha^k = \frac{\left\langle \tilde{\mathbf{r}}^k, \tilde{\mathbf{r}}^k \right\rangle}{\left\langle \tilde{\mathbf{p}}^k, \hat{\tilde{\mathbf{z}}}^k \right\rangle}$
        $\hat{\tilde{\mathbf{x}}}^{k+1} = \hat{\tilde{\mathbf{x}}}^k + \alpha^k \tilde{\mathbf{p}}^k$
        $\beta^k = \frac{\left\langle \hat{\tilde{\mathbf{x}}}^{k+1}, \hat{\tilde{\mathbf{x}}}^{k+1} \right\rangle}{\left\langle \hat{\tilde{\mathbf{x}}}^k, \hat{\tilde{\mathbf{x}}}^k \right\rangle}$
        $\hat{\tilde{\mathbf{r}}}^{k+1} = \hat{\tilde{\mathbf{r}}}^k - \alpha^k \hat{\tilde{\mathbf{z}}}^k$
        $\tilde{\mathbf{p}}^{k+1} = \hat{\tilde{\mathbf{r}}}^{k+1} + \beta^k \tilde{\mathbf{p}}^k$
$\tilde{\mathbf{x}}^{k+1} := \tilde{\mathbf{Q}}\tilde{\mathbf{b}} + \tilde{\mathbf{P}}^\top \hat{\tilde{\mathbf{x}}}^{k+1}$
$\mathbf{x}^{k+1} = \mathbf{M}^{-\frac{1}{2}} \tilde{\mathbf{x}}^{k+1}$
```

Algorithm 5 Deflated Preconditioned Conjugated Gradient (Pracical version)

```
Initial: $\mathbf{x}^0, \varepsilon$
Compute: $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0, \hat{\mathbf{r}}^0 = \mathbf{P}\mathbf{r}^0, \mathbf{z}^0 = \mathbf{M}^{-1}\hat{\mathbf{r}}^0$ and $\mathbf{p}^0 = \mathbf{z}^0$
for $k = 0, \ldots$ do
    while $\mathbf{r}^k > \varepsilon$ do
        $\alpha^k = \frac{\left\langle \hat{\mathbf{r}}^k, \mathbf{z}^k \right\rangle}{\mathbf{p}^k, \mathbf{P}\mathbf{A}\mathbf{p}^k}$
        $\hat{\mathbf{x}}^{k+1} = \hat{\mathbf{x}}^k + \alpha^k \mathbf{p}^k$
        $\hat{\mathbf{r}}^{k+1} = \hat{\mathbf{r}}^k - \alpha^k \mathbf{P}\mathbf{A}\mathbf{p}^k$
        $\hat{\mathbf{z}}^{k+1} = \mathbf{M}^{-1}\hat{\mathbf{r}}^k$
        $\beta^k = \frac{\left\langle \hat{\mathbf{r}}^{k+1}, \mathbf{z}^{k+1} \right\rangle}{\left\langle \hat{\mathbf{r}}^k, \mathbf{z}^k \right\rangle}$
        $\mathbf{p}^{k+1} = \mathbf{z}^0 + \beta^k \mathbf{p}^k$
$\mathbf{x} = \mathbf{Q}\mathbf{b} = \mathbf{P}^\top \mathbf{x}^{k+1}$
```

Definition 5.2. Assume $\mathbf{A}$ is SPSD with eigenvalues $\lambda_1, \dots, \lambda_n$. The effective condition number is defined as

$$\kappa_{eff}(\mathbf{A}) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}, \tag{5.45}$$

where $\lambda_{\min}$ is the smallest nonzero eigenvalue.

The error in the $\mathbf{A}$-norm is given by

$$\|\mathbf{x} - \mathbf{x}^k\|_{\mathbf{A}} \leq 2\|\mathbf{x} - \mathbf{x}^0\|_{\mathbf{A}} \left(\frac{\sqrt{\kappa_{eff}(\mathbf{M}^{-1}\mathbf{PA})} - 1}{\sqrt{\kappa_{eff}(\mathbf{M}^{-1}\mathbf{PA})} + 1} \right)^k \tag{5.46}$$

5.8 ROM-based Preconditioner

The other method to use the POD basis matrix is to construct a preconditioner to accelerate the process based on an AMG approach. This has been investigated and proposed by [2]. The new preconditioner is similar to the inverse of $\mathbf{A}$. As mentioned before, if the inverse of the original system is found, the solution is found in one iteration. The proposed preconditioner constructed by using AMG approach given in algorithm 6 by using $\mathbf{v}^0 = 0, \nu_1 = 1, \nu_2 = 0$ is:

Algorithm 6 Two-grid AMG algorithm

```
Initial: $\nu_1, \nu_2, \mathbf{x}^0, \mathbf{M}, \mathbf{Z}$
for $k = 0, \dots, \nu_1 - 1$ do
    $\mathbf{x}^{k+1} = (\mathbf{I} - \mathbf{MA})\mathbf{x}^k + \mathbf{Mb}$
$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}^{\nu_1}$
$\tilde{\mathbf{r}} = \mathbf{Z}^\top \mathbf{r}$
$\tilde{\mathbf{e}} = \left(\mathbf{Z}^\top \mathbf{AZ}\right)^{-1} \tilde{\mathbf{r}}$
$\mathbf{e} = \mathbf{Z}\tilde{\mathbf{e}}$
$\mathbf{x}^{\nu_1} = \mathbf{x}^{\nu_1} + \mathbf{e}$
for $k = \nu_1, \dots, \nu_1 + \nu_2 - 1$ do
    $\mathbf{x}^{k+1} = (\mathbf{I} - \mathbf{MA})\mathbf{x}^k + \mathbf{Mb}$
$\mathbf{x} = \mathbf{x}^{\nu_1 + \nu_2}$
```

$$\mathbf{M}_{rom}^{-1} = \mathbf{M} + \mathbf{Q}(1 - \mathbf{AM}), \tag{5.47}$$

where $\mathbf{Q} = \mathbf{Z}^\top \mathbf{E}^{-1} \mathbf{Z}$ and $\mathbf{E} = \mathbf{Z}^\top \mathbf{AZ}$ as defined in Section 5.5.
Note that $\mathbf{M}_{rom}$ is not always symmetric, thus it is not SPD. To obtain an SPD variant, the preconditioner should be symmetric. The symmetric matrix is found by using the formula $\frac{\mathbf{A}+\mathbf{A}^\top}{2}$ and that $\mathbf{M}$ is symmetric. The symmetric version of the rom-based conditioner is defined as:

$$\mathbf{M}_{srom}^{-1} = \mathbf{M} + \mathbf{Q} - \frac{1}{2}(\mathbf{QAM} + \mathbf{MAQ}) \tag{5.48}$$

6 Numerical Experiments

In this section, we define a few test problems for the incompressible model and constant compressible model. To give a general impression on how the methods work, it will be tested on the incompressible model. For the numerical experiments, the matrices are small such that the true solution can be computed with direct methods. Then, for the constant compressible models, different flow test problems will be defined. In the next report, we will solve the flow problems and analyse the numerical method given in Section 5.

6.1 Incompressible model

The incompressible model given in (4.18) is defined as

$$\mathbf{Tp} = \mathbf{q} \tag{6.1}$$

with homogeneous Dirichlet boundary conditions on the x-axis. The source vector is zero except for the first entries, thus

$$\mathbf{q}(1:5) = \begin{bmatrix} 80000 & 160000 & 160000 & 160000 & 80000 \end{bmatrix}^\top. \tag{6.2}$$

We consider the grid cell $[-1,1] \times [-1,1]$ in 2D with 2 different layers. The number of gridpoints in de x-direction is $n_x = 10$ and y-direction is $n_y = 5$, since we have 2 layers. With homogeneous boundary conditions on the x-axis, we get $n = (n_x - 2)n_y$. For this problem we use the lexicographic ordering $(i,j) \mapsto i + (j-1)(n_x - 2)$ for $1 \leq i \leq n_x - 2$ and $1 \leq j \leq n_y$. The transmissibility matrix $\mathbf{T}$ can be found in Figure 1.

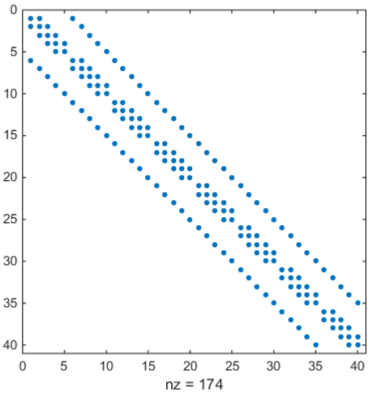

Figure 1: Nonzero structure of $\mathbf{T}$

Figure 1 shows that the matrix is sparse and the dimension of the transmissibility matrix $\mathbf{T}$ is 40 by 40. The exact solution of this problem is found with a direct method and compared to the iterative methods: Conjugate Gradient (CG), preconditioner Conjugate Gradient (PCG) and Deflated Conjugate Gradient (DCG). The preconditioner $\mathbf{M}$ is chosen as incomplete Cholesky decomposition with zero fill in. The deflation-subspace matrix $\mathbf{Z}$ consists of subdomain deflation vectors, that can be found in [7]. For this experiment, $\mathbf{Z}$ consists of 2 subdomain vectors.

Figure 2 shows the amount of iterations for each method and in Table 1 we give the general overview of the methods in terms of error, iteration and (effective) condition number κ_{eff}. The stopping criterion for this problem is defined as:

$$\epsilon = \frac{\|\mathbf{A}\mathbf{x}^k - b\|_2}{\|b\|_2} \leq 10^{-7}. \tag{6.3}$$

The error is defined as

$$\text{Error} = \|\mathbf{x} - \mathbf{x}^k\|_2, \tag{6.4}$$

where $\mathbf{x}$ is the true solution and $\mathbf{x}^k$ is the iterative solution after k steps.

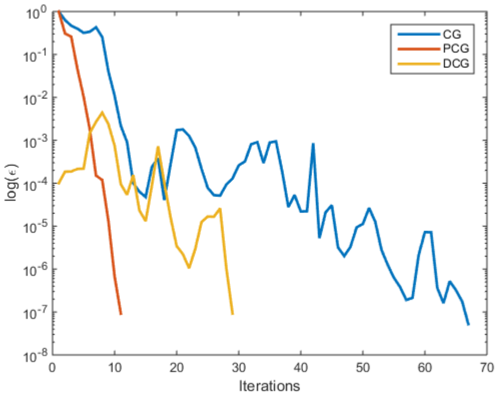

Figure 2: Iterations

Table 1: General overview

	CG	PCG	DCG
Iterations	67	11	29
Error	$4.6668 \cdot 10^{-13}$	$3.9139 \cdot 10^{-12}$	$6.0592 \cdot 10^{-10}$
κ_{eff}	$2.3591 \cdot 10^{5}$	5.4217	$2.1021 \cdot 10^{4}$

Table 1 show the problem of reservoir simulation is the high condition number. The (effective) condition number of preconditioner matrix $\mathbf{M}^{-1}\mathbf{A}$ and deflation matrix $\mathbf{PA}$ are smaller than that of the original system $\mathbf{A}$. Hence, it need less iterations to converge.
Recall, the dimension of $\mathbf{T}$ is 40 by 40 and the Conjugate Gradient method needed 67 iterations. The amount of iterations is more than the dimension of the system. This is presumably caused due to rounding errors and will be more investigated in the next report.

6.2 Test Problems

In this section, several test problems are given based on different assumptions. The previous model, incompressible model, is time independent. To investigate the methods further, we apply the methods to a 2D Laplace equation for better understanding of them. Then, we vary the amount the amount of layers to change the size of the system. Furthermore, the methods will be applied to the constant compressible model, which will be discussed in the next section.

6.2.1 Constant Compressible model

This model is time dependent the equation is nonlinear. To compute each time step, we need to linearize the model using the Newton-Raphson method. Then, for each iteration of this method, the large system that has the form of an incompressible model needs to be solved. Hence, more iterations are needed to solve the compressible model compared to the incompressible model.

The numerical constant compressible model defined in Section 4.4 is given by

$$\mathbf{V}\frac{\boldsymbol{\rho}(\mathbf{p}^{k+1}) - \boldsymbol{\rho}(\mathbf{p}^{k})}{\Delta t^k} + \mathbf{T}\mathbf{p}^{k+1} = \bar{\mathbf{q}}(\mathbf{p}^{k+1}), \tag{6.5}$$

where $\Delta t^k = t^{k+1} - t^k$ and $\mathbf{V}$ is the accumulation matrix defined as

$$\mathbf{V} = \Delta x \Delta y \Delta z \varphi \mathbf{I}_n, \tag{6.6}$$

where $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is the identity matrix.
The first test problem about the constant compressible model has a geological model with 2 different layers. For each time step, the obtained solutions are used to construct a POD basis. After that, the size of the problem is enlarged to see if the conclusions are the same.

The afore mentioned test problems will be used as numerical experiments for the numerical methods: Deflated preconditioner Conjugate Gradient and ROM-based preconditioner. These methods uses the POD method in their method and it is interesting to investigate and compare them in terms of complexity, error, amount of iterations and memory storage.

7 Conclusion

To simulate reservoir flow, two models are needed: the physical model, and the mathematical model. The mesoscopic model is used to derive the general mathematical model. Afterwards, assumptions are made to derive the simple model. The simple model is the incompressible case, i.e. where the porosity and density is pressure independent. The simple problem does not depend on time and therefore a time-independent problem. Then, the constant compressibility is assumed for the density to get the incompressible model. This problem is time-dependent and needs more computational time to solve numerically. The model has been discretized in a nonlinear system.

The nonlinear set of equations are linearized using Newton-Raphson method to get to the form $\mathbf{Ax} = \mathbf{b}$. The matrix $\mathbf{A}$ is very large and ill-conditioned. Therefore, it takes time to obtain the solution $\mathbf{x}$. Iterative numerical methods are used to solve large systems like this. Due to the property of $\mathbf{A}$ being SPD, it is convenient to use Conjugate Gradient method. The next step is to use preconditioner Conjugate Gradient, this reduces the condition number and ensures faster convergence. New POD based methods have been derived to further accelerate the process. The first method, Deflated Preconditioner Conjugate Gradient method, uses the POD basis matrix $\boldsymbol{\Psi}$ as the deflation-subspace matrix $\mathbf{Z}$ for the deflation based method. The other method uses the basis matrix to construct a ROM-based preconditioner.

These two afore mentioned methods are similar since they both uses the deflation-subspace matrix $\mathbf{Z}$ to solve the flow problem. Therefore, it is interesting to compare these two methods applied to different test problems. For the upcoming research, both methods will be implemented and analysed on 2D laplace equation before applied to flow problems. Then, the optimal POD based method will be given based on complexity, memory storage, convergence, and iterations.

8 Appendix

A List of Notation

The list of notation defined in Section 4 is given in this Appendix.

Table 2: Notation

Symbol	Quantity	SI Unit
ρ	Fluid density	kg/m^3
ϕ	Rock porosity	
q	Source term	
$\mathbf{v}$	Darcy's velocity	m/d
p	Pressure	Pa
K	Rock permeability	Darcy (D)
μ	Fluid viscosity	Pa
g	Gravity	m/s^2
d	reservoir depth	m
c_l	Liquid compressibility	Pa^{-1}
c_r	Rock compressibility	Pa^{-1}

References

[1] GB Diaz Cortes, C. Vuik, and J. D. Jansen. On POD-based Deflation Vectors for DPCG applied to porous media problems. *Journal of Computational and Applied Mathematics*, 330(Suplement C): 193-213, 2018.

[2] D. Pasetto, F. Massimiliano, and P. Mario. A reduced order modelbased preconditioner for the efficient solution of transient diffusion equations. *International Journal for Numerical Methods in Engineering* 1159-1179, 2017

[3] J. D. Jansen. A systems description of flow through porous media. *New York: Springer*, 2013.

[4] Knut-Andreas Lie. An introduction to reservoir simulation using MATLAB: user guide for the Matlab Reservoir Simulation Toolbox (MRST). *SINTEF ICT.* 2014.

[5] J.R. Shewhuk et al. An introduction to the conjugate gradient method without the agonizing pain. 1994.

[6] Y. Saad. Iterative methods for sparse linear systems. Vol. 82. *siam*, 2003.

[7] J. M. Tang. Two-level preconditioned conjugate gradient methods with applications to bubbly flow problems. *??*, 2008

[8] R.Haberman, Applied Partial Differential Equations, Fifth edition, *Pearson Prentice Hall*, New Jersey, 2013