Industrial Training

Reduced Order Models applied to hydraulic fracturing problems

Martí Burcet Rodríguez

MSc. in Numerical Methods in Engineering
Universitat Politècnica de Catalunya
Laboratòri de Càlcul Numèric (LaCàN)

Dec. 2016 - Mar. 2017

CIMNE

Industrial Training

Reduced Order Models applied to hydraulic fracturing problems

by

Martí Burcet Rodríguez

Industrial Training Tutor: Sergio Slotnik

CIMNE

Preface

The present report summarizes the work done by Martí Burcet Rodríguez during the Industrial Training of the MSc. in Numerical Methods in Engineering in *Laboratòri de Càlcul Numèric (LaCàN).*

Barcelona, 29th of may 2017

Contents

1

Industrial Training report

1.1. Introduction

Hydraulic Fracturing is a technique used in the oil industry to extract a type of non-conventional oil called *shale oil* from the subsurface. Shales are a type of rock with the particularity of very smalls permeabilities (in the order of $10^{-6} - 10^{-9}$ Darcy) what makes difficult for the hydrocarbon that may be occupying the porosity flow to the extraction wells. The idea of Hydraulic Fracturing is to intentionally fracture the rock to artificially increment the permeability and in this way allow the oil flow and pump it to the surface.

The process of triggering the fractures consists in injecting high pressure fluids to the rock that when starts flowing inside the rock, fractures starts opening. Then once the injection stops, the induced fractures remain opened due to sediments transported in the fracturing fluid and applying a negative gradient of pressure, the oil will flow to the extraction pump.

However this process of injecting high pressure fluids into the rock is sometimes dangerous in terms of environmental impact or even geothechnical risks. This is because the pressure at which fluid has to be injected or the time it should last for is unknown at the time of fluid injection. To improve this performance, numerical simulations have been proposed and used since 1950's to study the evolution of hydraulic fractures in shale reservoirs [1].

Full numerical simulations is a costly process because the domain of study is large and because this is a time dependent problem. In addition since the properties and geometry of the rock are in many cases quite uncertain (due to few geological tests or just because their confidence level is low) the pre-well performance simulations may be inaccurate and useless.

The aim of the present Industrial Training (and the future Master's Thesis) is to check the suitability of a model order reduction of the hydraulic fracturing problem. The final objective of the use of reduced order models in this kind of problem would be to run real-time simulations that could provide relevant data for the optimal and safety completion of hydraulic fracturing processes.

1.2. Work plan objectives

This Industrial Training has been thought as an introduction to the Master's Thesis that will be an extension of the present code to account for time evolving hydraulic fracture. To this aim the objectives fixed in the work plan for the present Industrial Training have been:

1. Produce a FE solution of the simple fracking model
2. Propose a scaling procedure for the fracking model
3. Obtain a reduced order representation of the FE solution via POD
4. Obtain a reduced order solution via PGD

5. Compare the POD and PGD solutions

1.3. Hydraulic Fracturing Model

The hydraulic fracturing problem is a solid-fluid coupled non-linear problem. In the model we have used the solid is assumed to be an elastic body that accomplish stress equilibium equations. The fluid is modeled as a laminar flow between two parallel plates, defined as Poiseuille flow [5].

The original domain of study is a rectangular domain with a punctual fluid injection in the middle of a predefined fracture, like the scheme of Figure 1.1.

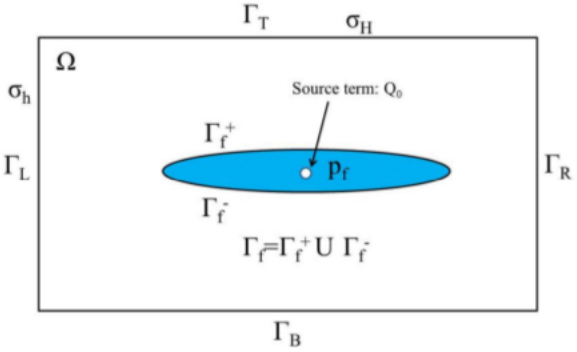

Figure 1.1: Scheme of the numerical model

The governing equations in strong form are:

- Solid elastic continuity equation and boundary conditions:

$$\nabla \cdot \sigma = 0 \qquad in \;\; \Omega_s \tag{1.1}$$

$$\sigma \cdot n_t = \sigma_H \qquad on \;\; \Gamma_L \cup \Gamma_R \tag{1.2}$$

$$\sigma \cdot n_t = \sigma_h \qquad on \;\; \Gamma_T \cup \Gamma_B \tag{1.3}$$

$$\sigma \cdot n_f = p(s,t) \qquad on \;\; \Gamma_f \tag{1.4}$$

$$u = 0 \qquad \text{on the four vertices of the outer boundary} \tag{1.5}$$

- Fluid flow equation: Poiseuille flow through two parallel plates

$$\frac{\partial w}{\partial t} = \frac{\partial}{\partial s}\left(\frac{w^3}{12\mu}\frac{\partial p}{\partial s}\right) + Q_0 \qquad in \;\; \Omega_f \tag{1.6}$$

$$w = 0 \qquad on \qquad \text{on the crack tip} \tag{1.7}$$

$$w(x, t=0) = w_0 \tag{1.8}$$

$$p(x, t=0) = p_0 \tag{1.9}$$

where $\boldsymbol{\sigma}$ is the stress tensor, p is the pressure and w is the fracture opening.

The corresponding weak forms are:

- Solid elastic:

$$\int_{\Omega_s} \delta\varepsilon^T \sigma d\Omega - \int_{\Gamma_{out}} \delta u^T \sigma_0 d\Gamma + \int_{\Omega_f} (\delta u^{+T} - \delta u^{-T}) p d\Omega_f = 0 \tag{1.10}$$

- Poiseuille (laminar) flow:

$$\int_{\Omega_f} \delta p \dot{w} d\Omega_f + \int_{\Omega_f} \nabla(\delta p) \frac{w^3}{12\mu} \nabla p d\Omega_f = \int_{\Omega_f} \delta p Q_0 d\Omega_f \tag{1.11}$$

Using Galerkin formulation for the FE discretization and finite differences approximation for the time derivative, the coupled system can be written in discrete form as:

$$\begin{bmatrix} K_s & -M_s \\ \frac{1}{\Delta t} M_f A & K_f \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} f_s \\ f_f + \frac{1}{\Delta t} M_f A u^n \end{bmatrix} \tag{1.12}$$

In equation (1.12) K_s is the stiffness matrix of the solid problem, M_s is the mass matrix of the solid problem, f_s is the force vector of the fluid problem; K_f is the stiffness matrix of the fluid problem, M_f is the mass matrix of the fluid problem, f_f is the force vector of fluid problem and A is a geometrical matrix that relates fluid and solid nodes.

This coupled system can be solved iteratively for u^{n+1} and p^{n+1} with a forward time integration scheme starting with an initial value of u^0.

1.4. Initial code

The starting point of this Industrial Training was a MATLAB code able to produce a Finite Elements (FE) solution of a simple hydraulic fracturing problem and also another code that was able to produce a Proper Generalized Decomposition (PGD) solution. However both codes once checked in detail had some limitations:

FE code

1. The code gave in some cases negative pressures
2. Negative pressures induce negative fracturing openings

PGD code

1. The code was able only to find one mode
2. Some terms of the PGD formulation were forgotten in the code without justification
3. Some elemental matrices were wrongly computer

1.5. Initial proper pressure

After realizing that the FE code produce negative pressures that in some cases produce negative fracture openings, we saw that the origin of these negative pressures was an initial pressure distribution which wasn't in equilibrium with the boundary conditions. In this way, high initial pressures had no problem but rather low pressures gave negative pressure nodes specially in the early state.

A formula for ensuring equilibrium of initial pressure with boundary conditions is proposed [4] :

$$\hat{q}^* \left(\frac{u_2^\sigma(0)}{u_2^*(0)} + \frac{\hat{p}}{\hat{p}^*} \right)^3 \left(\frac{\hat{p}}{\hat{p}^*} \right) - \frac{1}{2} Q_0 = 0 \tag{1.13}$$

In equation (1.13) we have to specify an initial displacement u_0^* (fracture opening), a flow rate $\hat{q}^*$ corresponding to the far-field stress pressure $\hat{p} = \hat{p}^*$, a fluid fracture pumping rate Q_0 and a displacement u_2^σ due to the response to solely far-field, in our case set to $u_2^\sigma = 0$.

The computation of $\hat{q}^*$ is done as:

$$\hat{q}^* = \frac{1}{\mu'} (u_2^*)^3 \frac{\hat{p}^*}{L^0}, \tag{1.14}$$

where L_0 is the initial fracture length, u_2^* is an initial fracture opening that has to be specified and $\hat{p}^*$ is an initial fracture pressure which we choose to be $\hat{p}^* = 2\sigma_0$.

Finally the proper initial pressure is found solving the non-linear equation (1.13) together with (1.14) which MATLAB can solve with its non-linear solver with a proper initial guess. A good initial guess is [4]:

$$\hat{p} = \hat{p}^* \left(\frac{\frac{1}{2} Q_0}{\hat{q}^*} \right)^{1/4} \tag{1.15}$$

This initial pressure distribution fixed the negative pressures found by the FE code.

1.6. Domain reduction

The model of study was a fracture with a punctual fluid injection in the middle. This model produce a symmetric solution respect to the horizontal and vertical axes. The symmetric solution can be seen in Figure 1.2, so it makes sense to study only a quarter of it.

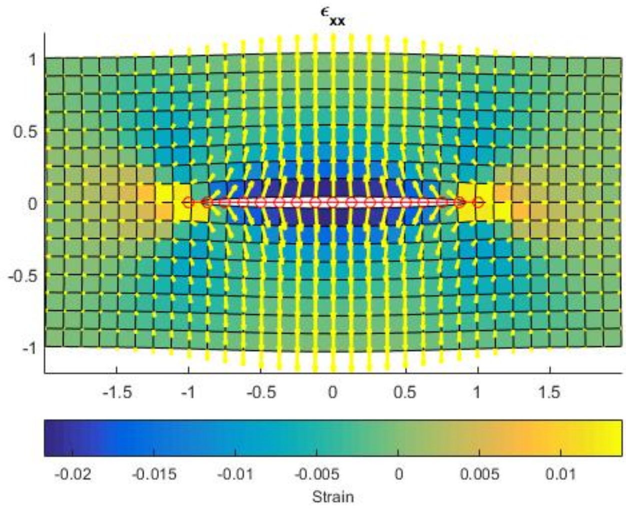

Figure 1.2: End time ϵ_{xx} strain distribution for the entire domain.

Taking advantage of the problem symmetries, a reduced domain of only a quarter of the original one has been adopted (Figure 1.3).

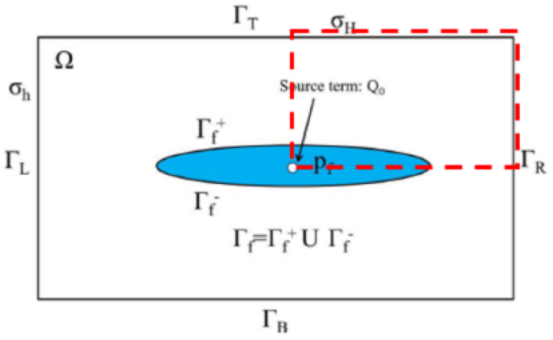

Figure 1.3: Domain reduction due to symmetries of the problem.

The implications in the formulation that this reduced domain requires are:

$$w = 2u_2 \text{ instead of } w = u_2^+ - u_2^- \tag{1.16}$$

and

$$Q_0 = \frac{Q_0}{2} \tag{1.17}$$

Also the Dirichlet boundary conditions of the symmetry edges are affected. The new conditions to impose are:

$$\begin{aligned} u_1 &= 0 \text{ in } \Gamma_L \\ u_2 &= 0 \text{ in } \Gamma_B \end{aligned} \tag{1.18}$$

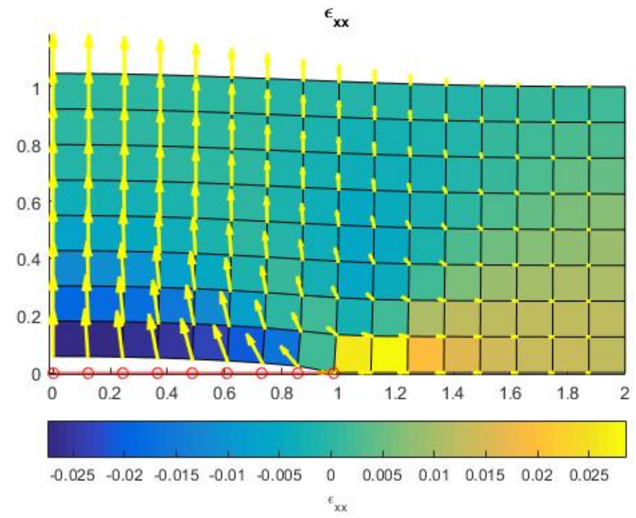

Figure 1.4: End time ϵ_{xx} strain distribution for the reduced domain.

Finally the only change in Neumann boundary conditions is in the bottom stress, which only affects the length that is not fracture:

$$\sigma \cdot n_b = t_b \text{ in } \Gamma_B \setminus \Gamma_f \tag{1.19}$$

With all these modifications we are able to compute the model in a reduced domain (Figure 1.4).

1.7. Discretizations

The spatial discretizations used to simulate the opening of hydraulic fractures were very simple. It consists in a 2D quadrilateral mesh to model the solid domain with vertical and horizontal degrees of freedom, and a 1D mesh for the fluid pressure unknown in the predefined fracture.

To avoid that the boundary conditions affects the fracture we have doubled the length of study to $\Omega_S = [0,4] \times [0,1]$ because as seen in Figure 1.4 the strain at the right boundary is significant.

In order to evaluate if the spatial and time discretization used were fine enough, several meshes were run and compared to see if the solutions were stable. The first test performed was a patch test to find the optimal spatial mesh size in terms of accuracy and computing resource. From this analysis (Figure 1.5) a mesh size of 0.125m was adopted because it gives nodal results very close to the ones obtained with finner meshes with much less computational time.

The time discretization has been tested to see whether more or less time steps affects the end time solution. As seen in Figure 1.6 there is no big difference in the final opening profile with one or another time mesh. However since the objective is to get a separable representation and to have the solution at many time instants, a time step of $\Delta t = 0.01s$ has been adopted and $t \in [0,1]$ so 101 time instants.

1.7.1. Centered pressure approximation

To obtain a better coupling between pressure and displacement, the codes have been adapted to interpolate pressures at the center of the displacement nodes. To get these centered meshes, a program for creating fluid and solid meshes with fluid nodes at the center of the solid nodes has been coded.

Note that the centered mesh doesn't start in 0 (Figure 1.7). This has an effect on the injection term Q_0 since we consider that the fluid is injected with a pipe of radius r_{pipe} placed in the region $(0, r_{pipe})$, and if we don't consider the fluid injected in the half element $(0, \frac{h}{2})$, we will be considering less injection than in reality. To fix this we have to compute an *effective* injection flow that equals the effect of the inflow (Q_{eff}) in the node-to-node and centered meshes. This *effective* injection flow is related to the initial value of source term as (1.20):

$$Q_{eff} = \frac{Q_0 r_{pipe}}{r_{pipe} - \frac{h}{2}} \tag{1.20}$$

1.8. Proper Orthogonal Decomposition

Before starting to develop a code that computes a separable solution of the hydraulic fracture problem with Proper Generalized Decomposition (PGD), the Proper Orthogonal Decomposition (POD) have been computed to see if the solution could be represented as a product of functions depending only on space and functions depending only on time.

We used Singular Value Decomposition (SVD) applied to a matrix storing the FE solution in all nodes (per rows) and for all times (per columns). The SVD of a matrix ($I_1 \times I_2$) containing the FE solution, gives as a result the following matrices [3]:

$$svd(A) = U \cdot S \cdot V^T \tag{1.21}$$

where U is a square matrix ($I_1 \times I_1$) that represent per columns the *modes* of the first dimension of A, V is also a square matrix ($I_2 \times I_2$) that represent per rows the *modes* of the second dimension of A and S is a rectangular diagonal matrix of size ($I_1 \times I_2$) where the elements of the diagonal represent the amplitudes σ_i of the modes.

The S matrix of the SVD have the following properties [3]:

- **Pseudodiagonality**: the only non zero elements of the matrix are those in the main diagonal of the rectangular matrix.

$$S = diag(\sigma_1, \sigma_2, \ldots, \sigma_{min(I_1, I_2)}) \tag{1.22}$$

- **Ordering**: the amplitude of the diagonal components decreases.

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{min} \geq 0 \tag{1.23}$$

Thanks to properties (1.22) and (1.23) it is possible to approximate the solution with only the firsts terms of POD because contains almost all the relevant information for the separable representation of the solution, which suppose savings in memory resources and also helps reducing the computation time in the post-process solution.

The fact that to get the separable representation with POD we need previously to compute the FE solution is why this method is considered an *a posteriori* method. The SVD of the matrix of the solution was easy done since MATLAB has a command that applied to a matrix does it automatically.

1.9. Proper Generalized Decomposition

The next task of the Industrial Training was to get an *a priori* separable representation of the solution via PGD. In order to obtain a scheme that could be used to generate a PGD solution of our problem we need to reduce the coupled system to a single equation that expresses the fluid-solid interaction. This is done isolating the displacement unknown $u^n + 1$ from the matrix FE formulation 1.12 and replacing w in the fluid equation as a function of u which is also a function of p. In this way we get an equation where the only unknown is the pressure p.

This pressure has to be rewritten as the product of pairs of modes depending only on space and time respectively. In this way the new expression for the pressure is written as:

$$p_{PGD}^{m+1} = \sum_{i=1}^{m+1} X_i(x) T_i(t) = X^{m+1}(x) T^{m+1}(t) + \sum_{i=1}^{m} X_i(x) T_i(t) = X^{m+1}(x) T^{m+1}(t) + p^m \tag{1.24}$$

what makes possible to construct the new approximation p^m as a correction of the previous solution p^{m-1}.

Finally we are able to compute the displacement field and the fracture opening as a transformation of the pressure field:

$$u^{m+1} = K_s^{-1} M_s p^{m+1} + K_s^{-1} f_s = B \cdot p^{m+1} + b \tag{1.25}$$

$$w^{m+1} = 2A \cdot u^{m+1} = 2A(Bp^{m+1} + b) \tag{1.26}$$

where A is a geometrical matrix that extracts the vertical component of the displacement of the fracture nodes and the 2 in equation (1.26) appears due to the domain reduction (Equation (1.16)).

The separable expression of the solution as the product of space and time modes allows the construction of the solution for all nodes and for all time. Therefore and since we have expressed the coupled system only in terms of pressure, the governing equation of the system is derived from the fluid weak form. This implies that the integrals of the weak form now are not only over space but also over time (1.27).

$$\int_{\Omega_f \times \Omega_t} \delta p \dot{w} dt d\Omega_f + \int_{\Omega_f \times \Omega_t} \nabla(\delta p) \frac{w^3}{12\mu} \nabla p dt d\Omega_f = \int_{\Omega_f \times \Omega_t} \delta p Q_0 dt d\Omega_f \tag{1.27}$$

Now to find the space and time modes we use an Alternating Direction Scheme (greedy algorithm) that consists in first solve for space X^{m+1} assuming time T^{m+1} known, and then solve for time T^{m+1} with the just-computed space mode X^{m+1} and iterate this process till the new computed modes stabilizes or a maximum of iterations.

Then the equation for finding the space mode X^{m+1} is obtained defining δp as the variations of p, namely $X^* T^{m+1}$ in (1.27).

$$\int_{\Omega_f \times \Omega_t} (X^* T^{m+1}) \dot{w} dt d\Omega_f + \int_{\Omega_f \times \Omega_t} \nabla(X^* T^{m+1}) \frac{w^3}{12\mu} \nabla p dt d\Omega_f = \int_{\Omega_f \times \Omega_t} (X^* T^{m+1}) Q_0 dt d\Omega_f \tag{1.28}$$

and the equation for finding the time mode T^{m+1} is obtained defining δp as the variations of p, namely $X^{m+1} T^*$ in (1.27).

$$\int_{\Omega_f \times \Omega_t} (X^{m+1} T^*) \dot{w} dt d\Omega_f + \int_{\Omega_f \times \Omega_t} \nabla(X^{m+1} T^*) \frac{w^3}{12\mu} \nabla p dt d\Omega_f = \int_{\Omega_f \times \Omega_t} (X^{m+1} T^*) Q_0 dt d\Omega_f \tag{1.29}$$

with

$$\dot{w} = 2AB\dot{p} = 2ABX^{m+1} \dot{T}^{m+1} + \sum_i^m X^i \dot{T}^i \tag{1.30}$$

Finally replacing terms and operating it leaves for the X^{m+1} mode:

$$\begin{aligned}
&\int_{\Omega_f} \left(2X^* ABX^m \int_{\Omega_t} T^{m+1} \dot{T}^{m+1} dt \right) d\Omega_f + \int_{\Omega_f} \left(\nabla(X^*) \frac{w^3}{12\mu} \nabla(X^{m+1}) \int_{\Omega_t} T^{m+1} T^{m+1} dt \right) d\Omega_f \\
&= \int_{\Omega_f} \left(X^* Q_0 \int_{\Omega_t} T^{m+1} dt \right) d\Omega_f - \sum_i^m \int_{\Omega_f} \left(2X^* ABX^i \int_{\Omega_t} T^{m+1} \dot{T}^i dt \right) d\Omega_f \\
&- \sum_i^m \int_{\Omega_f} \left(\nabla(X^*) \frac{w^3}{12\mu} \nabla(X^i) \int_{\Omega_t} T^{m+1} T^i dt \right) d\Omega_f
\end{aligned} \tag{1.31}$$

and for the T^{m+1} mode:

$$\begin{aligned}
&\int_{\Omega_t} \left(T^* \dot{T}^{m+1} \int_{\Omega_f} 2X^{m+1} ABX^{m+1} d\Omega_f \right) dt + \int_{\Omega_t} \left(T^* T^{m+1} \int_{\Omega_f} \nabla(X^{m+1}) \frac{w^3}{12\mu} \nabla(X^{m+1}) d\Omega_f \right) dt \\
&= \int_{\Omega_t} \left(T^* Q_0 \int_{\Omega_f} X^{m+1} d\Omega_f \right) dt - \sum_i^m \int_{\Omega_t} \left(T^* \dot{T}^i \int_{\Omega_f} 2X^{m+1} ABX^i d\Omega_f \right) dt \\
&- \sum_i^m \int_{\Omega_t} \left(T^* T^i \int_{\Omega_f} \nabla(X^{m+1}) \frac{w^3}{12\mu} \nabla(X^i) d\Omega_f \right) dt
\end{aligned} \tag{1.32}$$

To solve the time problem (1.32) we have to localize and return to the strong form (1.33) which is a first order IVP that we solve with the Runge-Kutta fourth-order method (RK4):

$$\begin{aligned} &\dot{T}^m \int_{\Omega_f} 2X^m ABX^m d\Omega f + T^m \int_{\Omega_f} \nabla(X^m) \frac{w^3}{12\mu} \nabla(X^m) d\Omega_f = Q_T \int_{\Omega_f} X^m Q_X d\Omega_f \\ &- \sum_i^{m-1} \dot{T}^i \int_{\Omega_f} 2X^m ABX^i d\Omega_f - \sum_i^{m-1} T^i \int_{\Omega_f} \nabla(X^m) \frac{w^3}{12\mu} \nabla(X^i) d\Omega_f \end{aligned} \tag{1.33}$$

The remaining equations (1.31) and (1.32) are still non-linear due to the cubic opening term $\left(\frac{w^3}{12\mu}\right)$ which also has to be linearized and separated. To linearize it there are two possibilities: Newton's method, that consists in linearize the governing equation before applying PGD, or Picard's method, where the non-linear term is evaluated from the previous enrichment step [2]. In our case we chose the later.

To separate the cubic term $\frac{w^3}{12\mu}$ in modes depending only on space and modes depending only on time we use Singular Value Decomposition (SVD), so we end up with a decomposition:

$$svd\left(\frac{w^3}{12\mu}\right) = \theta(x) \cdot \alpha \cdot \phi(t)^T \approx \sum_l^L \theta^l(x) \alpha^l (\phi^l(t))^T \tag{1.34}$$

where θ has the space modes, α represents the amplitudes of the modes and ϕ expresses the time modes. In this way we have obtained a linearized and separable representation of $\frac{w^3}{12\mu}$.

The injection term Q_0 also has to be expressed as the product of a function depending only on space and a function depending only on time (1.35). These functions are basically the injection rate in the nodes inside the pipe and zero elsewhere for space and all ones for the time since we inject during all the time domain at the same rate.

$$\int_{\Omega_f} \int_{\Omega_t} Q_0 dt d\Omega_f = \int_{\Omega_f} Q_0 d\Omega_f \int_{\Omega_t} dt = Q_X Q_T \tag{1.35}$$

For the matrix formulation of the problem we define:

$$\begin{aligned} K_{w^3} &= \sum_l K^l_{w^3} & K^l_{w^3}(x) &= \int_{\Omega_f} \nabla N \theta(x) (\nabla N)^T d\Omega_f \\ M_{w^3} &= \sum_l M^l_{w^3} & M^l_{w^3}(t) &= \int_{\Omega_f} \varphi \phi(t) \varphi^T dt \\ C(t) &= \int_{\Omega t} \varphi (\nabla \varphi)^T dt & M(x) &= \int_{\Omega_f} N N^T d\Omega_f \end{aligned}$$

And finally the matrix expression for the alternating direction strategy ends being:

$$[T^m(2MABC)T^m + \sum_l \alpha^l K^l_{w^3} T^m M^l_{w^3} T^m] X^{m+1} = \tag{1.36}$$

$$X^m Q_X T^m Q_T - \sum_i^m X^m (2MAB) X^i T^m C T^i - \sum_i^m X^m (\alpha^i M T^m C T^m - \sum_l K^l_{w^3} T^m M^l_{w^3} T^i) X^i \tag{1.37}$$

$$X^{m+1}(2MAB)X^{m+1}\dot{T}^{m+1} + \sum_l X^{m+1} K^l_{w^3} \phi^l X^{m+1} T^{m+1} = \tag{1.38}$$

$$X^{m+1} Q_X Q_T - \sum_i^m X^{m+1}(2MAB) X^i \dot{T}^i - \sum_i^m \sum_l X^{m+1} K^l_{w^3} \phi^l X^i T^i \tag{1.39}$$

The alternating direction strategy defined by equations (1.36) and (1.38) is solved sequentially first for X^{m+1} with (1.36) and then for T^{m+1} with (1.38) and this process is iterated some times. The alternating direction scheme defined above finds modes in a sequential way, and then the procedure is iterated to ensure stability of the non-linear term. However since the model require a proper initial pressure distribution to avoid negative pressures, the first mode is devoted to impose this initial pressure distribution. This is done likewise in FEM with equation (1.13).

The resulting alternating direction scheme is formed by equation (1.36) which is an algebraic equation that is solved with a proper linear system of equation solver.

On the other hand equation (1.38) is a first order ordinary differential equation that we solve with the fourth order Runge-Kutta method with an appropriate initial value. To use this method we have to define the expression for the time derivative $\dot{T}^{m+1}$:

$$F(T(t)) = \dot{T}^{m+1} = \frac{X^{m+1} Q_X Q_T - \sum_i^m X^{m+1}(2MAB) X^i \dot{T}^i(t) - \sum_i^{m+1} \sum_l X^{m+1} K^i_{w^3} X^i \phi^l T^i(t)}{X^{m+1}(2MAB) X^{m+1}} \tag{1.40}$$

For the time derivatives of the already computed modes $\dot{T}^i$ we use an explicit finite differences approach:

$$\dot{T}^i(n) = \frac{T^i(n) - T^i(n-1)}{\Delta t} \tag{1.41}$$

The Runge-Kutta fourth order method is defined as:

$$T(n+1) = T(n) + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{1.42}$$

$$k_1 = F(t, T(n)) \qquad k_3 = F(t + \frac{\Delta t}{2}, T(n) + \frac{\Delta t}{2} k_2)$$

$$k_2 = F(t + \frac{\Delta t}{2}, T(n) + \frac{\Delta t}{2} k_1) \qquad k_4 = F(t + \Delta t, T(n) + \Delta t k_3)$$

which needs an initial condition T^0 to start the loop, in our case we have chosen $T^0 = 0$.

1.9.1. PGD algorithm

As we have discussed and presented, the model order reduction of the hydraulic fracturing problem is a non-linear problem. The Picard's scheme that we use to linearize it consists in iterate the separable expression of the non-linear term computed in the previous iteration till some equilibrium is reached.

Our algorithm is formed out of three loops. Before initializing the loops the firsts space and time modes are computed to accomplish with initial conditions. In this way the initial pressure is computed likewise in FEM with Equation 1.13. The space mode is assigned to be a normalized constant vector. The time mode is also assigned to be a constant vector with amplitude the computed initial pressure. The amplitude is assigned to the time mode and not to the space one because in the alternating direction strategy the space modes will be normalized while the time ones will have the amplitude.

The most external loop is a non-linear loop that updates the Singular Value Decomposition of the cubic opening term $\left(\frac{w^3}{12\mu}\right)$ with the updated PGD solution. The stopping criteria adopted for this process is (1.43):

$$max(p^i_{PGD} - p^{i-1}_{PGD}) < nonLinearTol \tag{1.43}$$

Inside this non-linear loop there is a second loop that controls the addition of new modes. This loop starts with the initialization of new modes in case in the previous non-linear iteration we had less modes. If we need to add new modes these are initialized with a vector of ones for both space X and time T. Since unlike in POD we don't have an *a priori* solution, our criteria cannot be based on an error with respect to the precomputed solution. Instead of this we use a criteria based on the importance of the new computed mode with respect to the already computed separable solution p^{i-1}. The stopping criteria is (1.44):

$$\frac{\|X^j T^j\|}{\|p^{j-1}_{PGD}\|} < tolPGD \tag{1.44}$$

This stopping criteria is sustained in the fact that, since we compute the $m-th$ PGD solution as an improve of the $m-1$ solution, the algorithm flats when the equation balance and no further improve is possible.

```
initialize $X^1$ and $T^1$;
$i = 1$;
while $max(p^i_{PGD} - p^{i-1}_{PGD}) > nonLinearTol\ or\ i < maxNonLinearIter)$ do
    compute $K_{w^3}$ and $M_{w^3}$ of $w^{i-1}_{PGD}$;
    $j = 2$;
    while $\|X^j T^j\| / \|p^{j-1}_{PGD}\| > tolPGD\ or\ j < maxModes$ do
        if $j > modes(p^{i-1}_{PGD})$ then
            initialize $X^j$ and $T^j$;
            compute $K_{w^3}$, $M_{w^3}$ and $\phi$ of the $j - th$ mode;
        end
        $k = 1$;
        while $\|X^k T^k - X^{k-1} T^{k-1}\| > alternateTol\ or\ k < maxIter$ do
            compute $X^{k+1}$ from $T^k$;
            compute $T^{k+1}$ from $X^{k+1}$;
            $k = k + 1$;
        end
        $j = j + 1$;
    end
    $i = i + 1$;
end
```

Algorithm 1: Summarized PGD algorithm.

Finally the most internal loop is the alternating direction strategy where the stopping criteria is:

$$\|X^k T^k - X^{k-1} T^{k-1}\| < alternateTol \tag{1.45}$$

The summarized algorithm is outlined in Algorithm 1. In the cases we have run we have used as algorithm parameters:

Parameter	$nonLinearTol$	$maxNonLinearIter$	$tolPGD$	$maxModes$	$alternateTol$	$maxIter$
Value	1×10^{-3}	5	1×10^{-3}	10	1×10^{-2}	10

1.10. Examples

Some example cases have been run with FEM, POD and PGD methods to compare their performances. The model parameters used for these examples are [6]:

Parameter	**Value**
Young's modulus	1440
Poisson's ratio	0.2
Viscosity	1×10^{-3}
Injection flow	0.2
Confining stress	27.7579

1.10.1. POD results

The amplitude of the POD modes have been plotted in Figure 1.9 and it clearly shows a decay in the amplitude of the modes. This indicates that with only few of them we are be able to represent with quite good accuracy and in a cheap way the separable solution.

The results of the POD decomposition (Figure 1.9) show that a separable representation of the hydraulic fracturing problem can be found for both displacement and pressure unknowns. However we are mode interested in the separable representation of the pressure problem because in the PGD scheme we will express our coupled problem only as function of pressure and displacement will be included as an elastic linearized and precomputed response to the fluid injection.

If now the number of modes required to obtain a similar level of precision with different space discretization is studied, we observe (Figure 1.10) that the finner the mesh, the more modes are needed to get the same level of error. We can also observe that the two finner meshes require the same number of elements to get the same level of error.

The time modes resulting from the Proper Orthogonal Decomposition (Figure 1.11) constitute a base for expressing the evolution of the fracture in the interval of interest.

The base of space pressure modes sequentially captures higher frequencies as seen in Figure 1.12. The $p-th$ space mode correspond to the $p-1$ order polynomial centered in the fracture domain.

1.10.2. PGD results

We have run the PGD algorithm to solve the hydraulic fracturing problem with different mesh sizes. Then the norm of the error with respect with the FEM solution has been computed. In Figure 1.13 we can see how all the mesh sizes after 2 modes flats to an almost constant error level and doesn't improve any more. The norm of the error increases as we refine the mesh as a consequence of integrating more nodes, not because the punctual error increases. We can see this looking at Figure 1.14 what shows a tendency of decreasing maximum error as we refine the mesh, which is the expectable behavior since it is known that the PGD error is proportional to the number of modes, the mesh size as well as the order of interpolation.

The fact that with only 2 modes the solution stops improving is due to the fact that we are computing the separable solution of the pressure distribution which inside the fracture is almost constant and the problem evolves linearly. Therefore after the first mode which is devoted to impose the initial conditions, the second space mode just has to approximate the constant distribution of pressures in space, which doesn't need of higher frequencies to capture it (Figure 1.15).

On the other hand the time evolution of the problem is linear because the problem is elliptical, so the second time mode is just a first order polynomial which translates the initial pressure distribution in time as we inject fluid (Figure 1.16). The next two time modes doesn't contribute in anything new to the solution. That is why they are the mirror of the other one, what cancels its effect.

1.11. Discussion

It has been checked that is possible to obtain a separable solution of the hydraulic fracturing problem although beeing coupled and non-linear from an *a posterior* method like POD and from an *a priori* method like PGD. Both methods present its pros and cons that we outline next.

POD requires finding a discrete solution before starting the Singular Value Decomposition, what in our case is not a big problem because the domain and time of study can be treated with a desktop computer in a maximum of few minutes (Table 1.1). Also standard SVD is limited to only a separation in two variables, although it is true that with High Order Singular Value Decomposition it is possible to extend this to more dimensions. Another disadvantage of POD with respect to POD is that for a time depending problem where at each time step we need to solve a linear system of equations, if instead of a maximum of tens of unknowns we had hundreds or thousands of them, this method blows up in terms of run-time and becomes much computationally costly than PGD.

On the other hand since POD is an *a posteriori*, we had in advance the FEM solution and using the Singular Value Decomposition we get an optimal base [2] which allows to obtain a separated representation of the solution with the desired accuracy up to machine precision (Figure 1.10). This choice in the desired level of accuracy is not possible with PGD because the construction of the solution is done sequentially as on improve of the previous one and the modes obtained are not necessarily optimum. This is an obvious pro in favor of POD against which PGD cannot compete.

Mesh size	**POD pre-process**	**POD computation (FEM+SVD)**	**PGD pre-process**	**PGD computation**
0.25	0.530460 s	0.157590 s	0.213657 s	1.929621 s
0.125	0.899074 s	0.443952 s	0.569684 s	1.664666 s
0.0625	7.813784 s	0.665579 s	7.433254 s	1.753949 s
0.03125	440.989199 s	9.489935 s	401.490810 s	13.173495 s

Table 1.1: Execution times of the examples.

PGD has no limitations in the number of variables of the problem, a property not only useful when trying to obtain a separable representation for a model order reduction but also to be able to get a solution with fine meshes for multidimensional models. This last property responds to the fact that the discretization required for a FEM solution is the cross product (1.46) while for the PGD solution the number of nodes grows as (1.47).

$$nodes_{FEM} = \Pi_{i}^{n_{dim}} d_i \tag{1.46}$$

$$nodes_{PGD} = \sum_{i}^{n_{d}im} d_i \tag{1.47}$$

However the fact of directly construct the solution as a product of modes depending only in one dimension limits the precision of the solution. This limitation in the accuracy is augmented in the case of time depending problems by the fact that initially we have to impose initial conditions devoting arbitrary chosen modes to this end. This intervention of the user in the construction of the solution affects all the further modes that have to adapt to the imposed condition. In addition in the computation of the time modes in the alternating direction strategy also an initial condition is required, what increases the affection of the choice in the final solution. Nevertheless we have been able to approximate the FEM solution with PGD with a maximum relative punctual error of around 4% (Figure 1.14).

In terms of computational performance, the PGD method is costly and more in our case of a non-linear problem where we have to iterate the alternating direction strategy with new updates of the separable representation of the non-linear term till the process stabilizes (1). This can be clearly seen looking at the POD and PGD execution times of Table 1.1 that shows that the PGD ones are substantially higher, although the difference reduces as we refine the meshes.

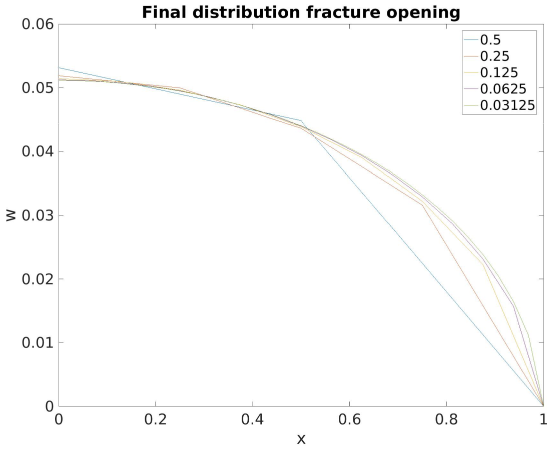

Figure 1.5: End time opening profile for different space discretizations

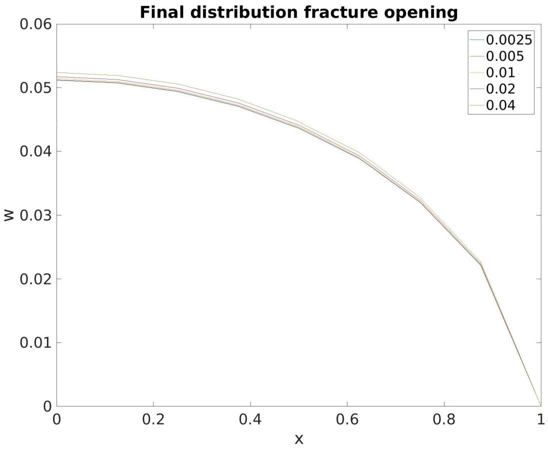

Figure 1.6: End time opening profile for different time discretizations

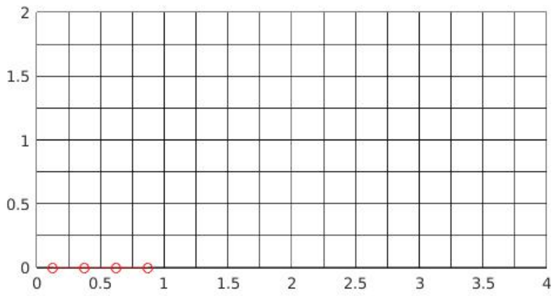

Figure 1.7: Centered fluid mesh

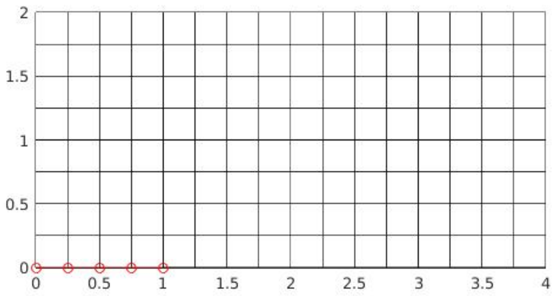

Figure 1.8: Node-to-node fluid mesh

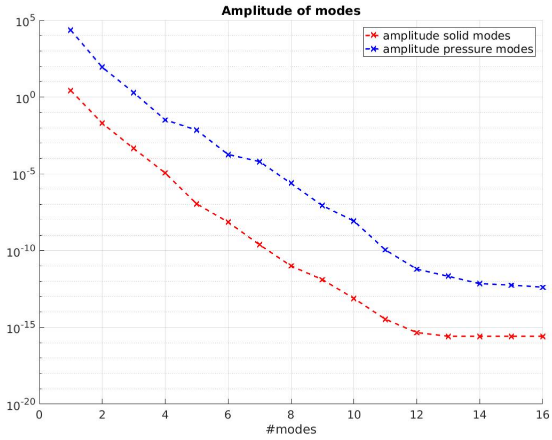

Figure 1.9: Amplitude of the pressure modes with a space discretization of $h = 0.0625$.

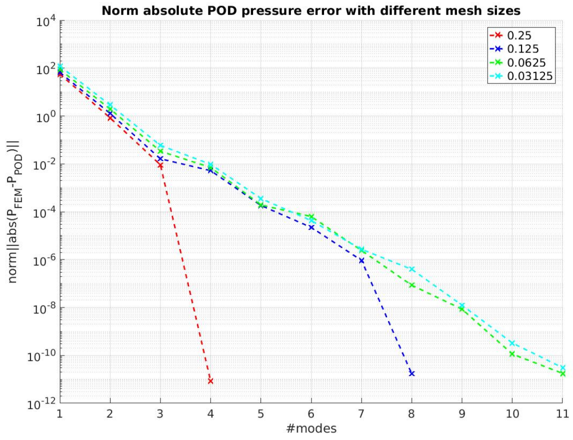

Figure 1.10: POD norm of the pressure errors with different space discretizations keeping the modes with amplitude bigger than 1×10^{-12}.

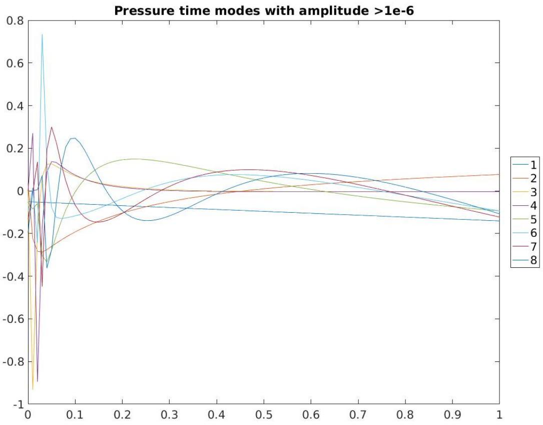

Figure 1.11: Firsts eight time modes of the POD solution with a time discretization of $t \in (0,1)$ with 101 time steps of $\Delta t = 0.01$.

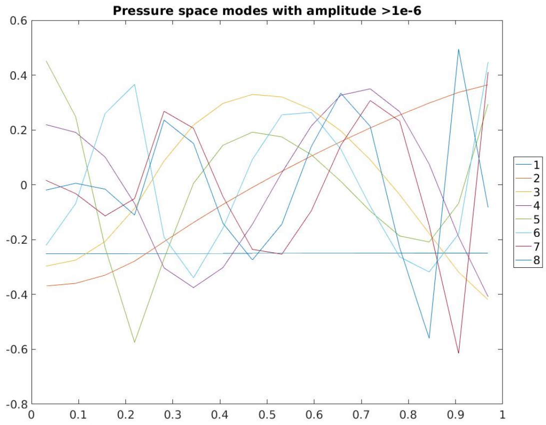

Figure 1.12: Firsts eight space modes of the POD solution with a space discretization of $h = 0.0625$.

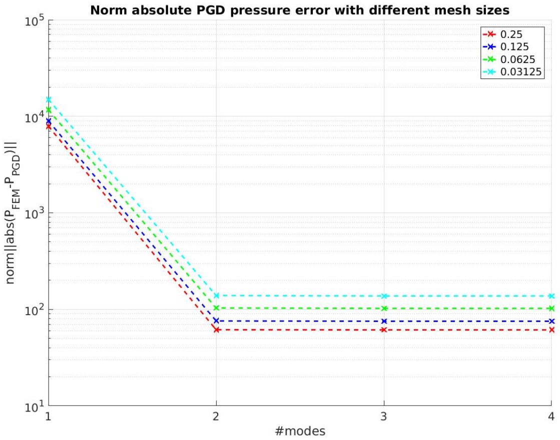

Figure 1.13: Norm of the absolute error for the pressure unknown with increasing number of modes and different mesh sizes.

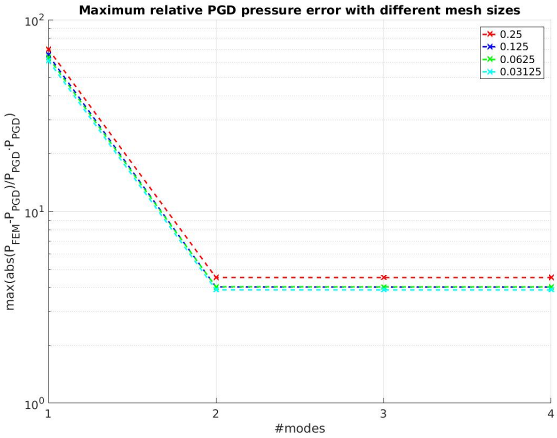

Figure 1.14: Maximum relative error for the pressure unknown with increasing number of modes and different mesh sizes.

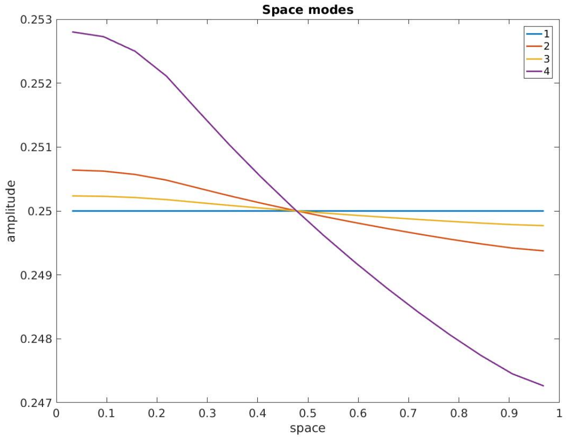

Figure 1.15: Space PGD modes for a fracture space discretization of $X \in [0,1]$ with $h = 0.0625$.

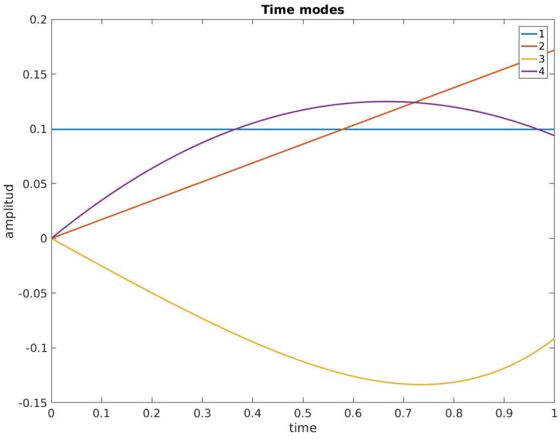

Figure 1.16: Time PGD modes for a time discretization of $t \in [0,1]$ with $\Delta t = 0.01$.

Bibliography

[1] J. Adachi, E. Siebrits, A. Peirce, and J. Desroches. Computer simulation of hydraulic fractures. *International Journal of Rock Mechanics and Mining Sciences*, 44(5):739–757, 2007. ISSN 13651609. doi: 10.1016/j.ijrmms.2006.11.006.

[2] Francisco Chinesta, R. Keunings, and Adrien Leygue. *The proper generalized decomposition for advanced numerical simulations : a primer*. ISBN 9783319028651. URL `http://cataleg.upc.edu/record=b1453206{~}S1*cat`.

[3] L De Lathauwer, B De Moor, and J Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000. ISSN 0895-4798. doi: 10.1137/S0895479896305696. URL `papers2://publication/uuid/1F9C4A1E-A3EC-4F48-BB7A-5C2EA0D5E82F`.

[4] Huabei Liu, Degao Zou, and Jiangmao Liu. A finite element approach to the simulation of hydraulic fractures with lag. *International Journal for Numerical and Analytical Methods in Geomechanics*, 2012. ISSN 03639061. doi: 10.1002/nag.

[5] J M L Poiseuille. Experimental investigations on the flow of liquids in tubes of very small diameter. *Rheological Memoirs*, 1940. ISSN 0097-1529.

[6] Ching H. Yew and Xiaowei Weng. *Mechanics of Hydraulic Fracturing*. ISBN 9780124200036.