Gas flow in a channel with elastic walls

Final project INF9620 Autumn 2012

Andreas Hafver

December 21, 2012

Abstract

This project considers the flow of a gas in a fracture which is being held open by the gas pressure.

Contents

1 Background

Primary migration is the process whereby hydrocarbons (oil and gas) is expelled from source rock. My group has developed the following analog 2D experiment which we hope will give some insight to the coupling between fluid production, fracturing and

drainage: Gelatine containing yeast and sugar is confined between two glass plates (Hele-Shaw cell). When the yeast consumes sugar it produces CO_2. We observe nucleation of bubbles that later may propagate as long, relatively straight fractures. There is no fracture branching, and the resulting fractures open and close as gas is expelled through the fracture network.

One aspect of these experiments is the coupling between the flow and dynamic opening and closing of fractures. This is what I set out to investigate in this project.

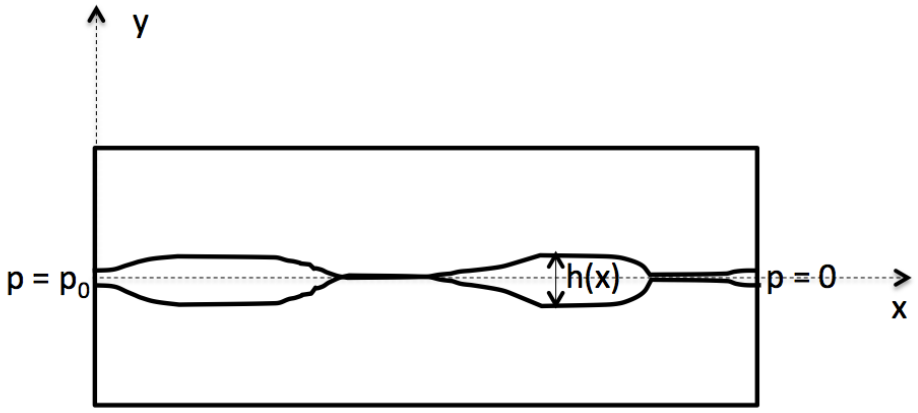

Figure 1: Model setup

2 Mathematical model

2.1 Derivation of the flow equation

The given problem is illustrated in figure 1. The point of departure for deriving a flow equation is the conservation of mass, stated mathematically as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0, \tag{1}$$

where $\rho = \frac{Nm}{V}$ is the mass density (N = number of particles, m = mass of one particle, V = volume) and $\vec{v}$ is the flow velocity. We consider a fracture in the xy-plane and define the mass per unit area $\Delta x \Delta y$ of a fracture as $\eta(x,y,t) = \int_0^{h(x,y,t)} \rho(x,y,z,t)dz$, where $h(x,y,t)$ is the fracture aperture at the point (x,y). $\eta(x,y,t)$ vary with time according to

$$\frac{d\eta}{dt} = \int_0^{h(x,y,t)} \frac{d}{dt}\rho(x,y,z,t)dz + \rho(x,y,h,t)\frac{dh}{dt}. \tag{2}$$

We assume the ideal gas law as the equation of state, so that the density at a point is proportional to the pressure,

$$\rho = \frac{Nm}{V} = \frac{NkT}{V}\frac{m}{kT} = \frac{m}{kT}P. \tag{3}$$

Note that the density is dependent on h trough the volume $V = \Delta x \Delta y h$, and therefore

$$\frac{d\rho}{dt} = \frac{\partial \rho}{\partial t} + \frac{\partial \rho}{\partial h}\frac{\partial h}{\partial t} = \frac{\partial \rho}{\partial t} - \frac{\rho}{h}\frac{\partial h}{\partial t}. \tag{4}$$

If we now assume that the density is constant over the fracture height, (2) in combination with (1) reduces to

$$\frac{d}{dt}(h\rho) \quad = \quad -\int_0^{h(x,y,t)} \nabla \cdot (\rho \vec{v})\, dz. \tag{5}$$

Finally we will assume Poiseuille flow (i.e. the flow is relatively steady so that the velocity profile is parabolic in the z direction), i.e.

$$\vec{v} = \frac{z(z-h)}{2\mu}\nabla P, \tag{6}$$

where μ is the gas viscosity. Inserting this in (5) yields

$$\frac{d}{dt}(h\rho) \quad = \quad \nabla \cdot \left(\frac{h^3 \rho}{12\mu} \nabla P \right). \tag{7}$$

We can rewrite this in terms of h and P as

$$\begin{aligned} \frac{d}{dt}(hP) &= 2k\nabla \cdot \left(h^3 P \nabla P \right) \\ &= k\nabla \cdot \left(h^3 \nabla P^2 \right), \end{aligned} \tag{8}$$

where $k = \frac{1}{24\mu}$. The final equation (8) makes intuitive sense: The only contribution to mass flux is the flux in the fracture plane due to pressure gradients. Changes in h does not affect the amount of mass contained at a point because ρ will change accordingly.

2.2 The coupling of gas pressure to the fracture aperture

Instead of solving the full elastic problem, the deformable host medium is modelled as an elastic membrane, i.e. the aperture $h(x,t)$ is related to the pressure P by the equation

$$h(x,y) \quad = \quad aP + b\nabla^2 h. \tag{9}$$

For $b = 0$ the aperture would be proportional to P, and a finite b accounts for the restoring force resulting from the the curvature of the wall. The significance of the last term is that the fracture can remain open at a point even when the pressure goes to zero, because the pressure in surrounding points can keep the fracture open. In

principle it should be possible to relate the constants a and b to material properties such as Young's modulus and Poisson ratio.

Based on the above, we end up with a set of coupled equations,

$$\begin{aligned} \frac{d}{dt}(hP) &= k\nabla\cdot\left(h^3\nabla P^2\right), \\ h &= aP + b\nabla^2 h. \end{aligned} \tag{10}$$

3 The 1D problem with constant aperture $h(x,t) = h$

For a constant aperture in one dimension the problem simplifies to

$$\frac{dP}{dt} = 2kh^2\frac{d}{dx}\left(P\frac{dP}{dx}\right), \tag{11}$$

or alternatively

$$\frac{dP}{dt} = kh^2\frac{d^2}{dx^2}(P^2). \tag{12}$$

Even this simplified problem is nonlinear, so it is instructive to study this problem first before returning to the more complicated problem with variable aperture. For the purpose of the next sections we will take the solution domain to be the unit interval and assume Dirichlet boundary conditions

$$P(0,t) = 1 \qquad P(1,t) = 0. \tag{13}$$

3.1 Finite difference approximation

The two forms (11) and (12) are mathematically equivalent, so let us check if they lead to the same discretizations: The finite difference disretization of the right hand side of (11) is

$$[2kh^2 D_x P D_x P]_i^n = \frac{2kh^2}{(\Delta x)^2}(P_{i+1/2}^n(P_{i+1}^n - P_i^n) - P_{i-1/2}^n(P_i^n - P_{i-1}^n)), \tag{14}$$

and the finite difference disretization of the right hand side of (12) is

$$[kh^2 D_x D_x P^2]_i^n = \frac{kh^2}{(\Delta x)^2}((P_{i+1}^n)^2 - 2(P_i^n)^2 + (P_{i-1}^n)^2)). \tag{15}$$

We see that the two discretizations are equivalent if we use the arithmetic mean for the midpoint vaues of P, i.e.

$$P_{i\pm1/2}^n = \frac{1}{2}(P_i^n + P_{i\pm1}^n) \tag{16}$$

The explicit forward Euler method, yields the following update scheme for our problem:

$$\begin{aligned} & [D_t^+ P = kh^2 D_x D_x P^2]_i^n \\ \rightarrow \quad & P_i^{n+1} = P_i^n + \frac{kh^2\Delta t}{(\Delta x)^2}((P_{i+1}^n)^2 - 2(P_i^n)^2 + (P_{i-1}^n)^2)). \end{aligned} \tag{17}$$

3.2 Finite element approximation

In the finite element method we start by reformulating the partial differential equation problem in variational form. In the Galerkin formulation this is achieved by approximating the unknown function $P(x,t_n)$ by a projection onto a vector space V and requiring that the residual $R = \frac{dP}{dt} - 2kh^2\frac{d}{dx}\left(P\frac{dP}{dx}\right)$ (from (11)) is orthogonal to V. More precisely we express $P(x,t_n)$ as a linear combination of basis function $\psi_i \in V$,

$$P(x,t_n) = \sum_i P_i^n \psi_i, \tag{18}$$

and require

$$(\psi_i, R) = \int_0^1 dx \psi_i(x) R(x,t) = 0 \qquad \forall i. \tag{19}$$

Let us chose the backward Euler scheme for time discretisation. Then we can expand the latter as

$$\int_0^1 dx \frac{1}{\Delta t}\psi_i \sum_j (P_j^n - P_j^{n-1})\psi_j - 2kh^2\psi_i(x)\frac{d}{dx}\left(\sum_j P_j^n \psi_j \frac{d}{dx}\sum_j P_j^n \psi_j\right) = 0. \tag{20}$$

The first term is equivalent to the matrix-vector expression

$$\frac{1}{\Delta t} M(P^n - P^{n-1}), \tag{21}$$

with

$$M_{ij} = \int_0^1 dx \psi_i(x)\psi_j(x). \tag{22}$$

The last term of (20) can be rewritten using integration by parts:

$$\begin{aligned}\int_0^1 dx \psi_i \tfrac{d}{dx}\left(\textstyle\sum_j P_j^n\psi_j \tfrac{d}{dx}\sum_j P_j^n\psi_j\right) = &-\int_0^1 dx \tfrac{d\psi_i}{dx}\textstyle\sum_j P_j^n\psi_j \tfrac{d}{dx}\sum_j P_j^n\psi_j \\ &+\left[\psi_i \tfrac{d}{dx}\textstyle\sum_j P_j^n\psi_j \tfrac{d}{dx}\sum_j P_j^n\psi_j\right]_0^1\end{aligned}, \tag{23}$$

and the boundary term vanish because we require the test functions to be zero at the Dirichlet boundaries. We may therefore express the last part of (20) in matrix vector form as

$$K(P^n)P^n, \tag{24}$$

with

$$K(P^n)_{ij} = \int_0^1 dx 2kh^2 \sum_k P_k^n \psi_k \frac{d\psi_i}{dx}\frac{d\psi_j}{dx}. \tag{25}$$

The full problem can then be stated as

$$\left[M + \Delta t K(P^n)\right] P^n = M P^{n-1}. \tag{26}$$

Because K depends on P^n this is not a linear system, but there are various strategies to proceed solving for P^n, and two of these are outlined below.

3.2.1 Alternative formulation

Suppose we had started from (12) and the residual $R = \frac{dP}{dt} - kh^2 \frac{d^2}{dx^2}(P^2)$. Then we would end up with

$$K(P^n)_{ij} = \int_0^1 dx kh^2 \sum_k P_k^n \frac{d\psi_i}{dx} \frac{d}{dx} (\psi_j \psi_k) . \tag{27}$$

Unlike (25), (27) is not a symmetric matrix. To see this explicitly, let us calculate the two different matrices K for an element of length s with linear shape functions,

$$\psi_1(x) = 1 - \frac{x}{s}, \qquad \psi_2(x) = \frac{x}{s}. \tag{28}$$

Inserting these in (25) and integrating yields (by means of Mathematica)

$$K_e = \frac{kh^2}{2s}(P_1^n + P_2^n) \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \tag{29}$$

Using (27) yields (by means of Mathematica)

$$K_e = \frac{kh^2}{s} \begin{bmatrix} P_1^n & -P_2^n \\ -P_1^n & P_2^n \end{bmatrix}. \tag{30}$$

Symmetry is an advantage and a property that may be exploited for efficient solution of linear systems, as well as for efficient storage (one only needs to store the upper or lower half matrix) and therefore we will stick to the form (25).

3.2.2 Relation to finite difference

It as clear that after assembling the matrix K and multiplying with the P^n vector one ends up with terms of the form $(P_{i+1}^n)^2 - 2(P_i^n)^2 - (P_{i-1}^n)^2)$ just like for the finite difference method. However, the two schemes will not be equivalent because of the matrix M which is not diagonal. Using Mathematica we obtain for linear elements

$$M_e = \frac{s}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}. \tag{31}$$

By lumping the M matrix (i.e. putting everything on the diagonal) and using the forward Euler time discretisation one could retain the finite difference scheme (17) derived above.

3.2.3 Picard iterations

The idea of this approach is to turn the problem $[M + \Delta t K(P^n)]\, P^n = M P^{n-1}$ into a linear problem by approximating the matrix $K(P^n)$ by $K(\tilde{P}^n)$, where $\tilde{P}^n$ is a vector of previously known values. One may solve the nonlinear problem by iteratively solving the linearised problem according to the following procedure at every time step n:

1. Solve $\left[M + \Delta t K(P^{n-1})\right] \tilde{P}^n = M P^{n-1}$;
2. Solve $\left[M + \Delta t K(\tilde{P}^{n-1})\right] P^n = M P^{n-1}$;
3. If $|\tilde{P}_i^n - P_i^n| > tol$ set $\tilde{P}^n = P^n$ and return to step 2.

3.2.4 Newton's method

We may derive Newton's method at the level of the PDE,

$$\frac{dP}{dt} = 2kh^2 \frac{d}{dx}\left(P\frac{dP}{dx}\right) \tag{32}$$

Suppose that the unknown P^n at time t_n can be expressed as a small deviation from a previously known $\tilde{P}^n$ i.e.

$$P^n \approx \tilde{P}^n + \delta P. \tag{33}$$

After applying the backward Euler scheme to discetize in time and inserting (33) we get

$$\tilde{P}^n + \delta P - P^{n-1} = 2\Delta t k h^2 \frac{d}{dx}\left(\tilde{P}^n + \delta P \frac{d}{dx}(\tilde{P}^n + \delta P)\right) \tag{34}$$

Collecting linear terms in δP on one side and ignoring higher order terms in δP we have

$$\delta P - C\left[\frac{d}{dx}\left(\tilde{P}^n \frac{d(\delta P)}{dx}\right) + \frac{d}{dx}\left(\delta P \frac{d\tilde{P}^n}{dx}\right)\right] = P^{n-1} - \tilde{P}^n + C\frac{d}{dx}\left(\tilde{P}^n \frac{d\tilde{P}^n}{dx}\right) \tag{35}$$

with $C = 2\Delta t k h^2$. We can now state our problem in variational from as

$$a(v, \delta P) = L(v), \tag{36}$$

with

$$a(v, \delta P) = \int_0^1 dx \delta P v + C\frac{dv}{dx}\left[\tilde{P}^n \frac{d(\delta P)}{dx} + \delta P \frac{d\tilde{P}^n}{dx}\right], \tag{37}$$

and

$$L(v) = \int_0^1 dx v(P^{n-1} - \tilde{P}^n) - C\frac{dv}{dx}\tilde{P}^n \frac{d\tilde{P}^n}{dx}. \tag{38}$$

The linear variational problem (36) can now be used to solve the nonlinear PDE iteratively in the same way as outlined for the Picard method. The variational form renders this problem suitable for isolation with the FEniCS package.

3.3 Numerical comparison

Figures 2 shows examples where the flow in a static channel is solved by respectively the finite difference forward Euler and finite element backward Euler with the Picard and Newton methods. For $c = \frac{kh^2\Delta t}{(\Delta x)^2} = 0.45$ we see large oscillations for the finite difference onward Euler methods, whereas the other two give almost indistinguishable results with no oscillations. By halving c all three methods collapse on one curve. For $c = 5$ the forward Euler finite difference method produce an error message whereas

Newton's method seem to switch the sign of the solution. It could be that there are two mathematically correct solutions and the method is not able to pick out the physical one. Given that the simple finite difference method is 30-70 times faster to execute than the other two methods one could afford to run this scheme with smaller time steps to get better accuracy and still save computation time. If long time steps are required the Picard scheme seems to be most reliable.

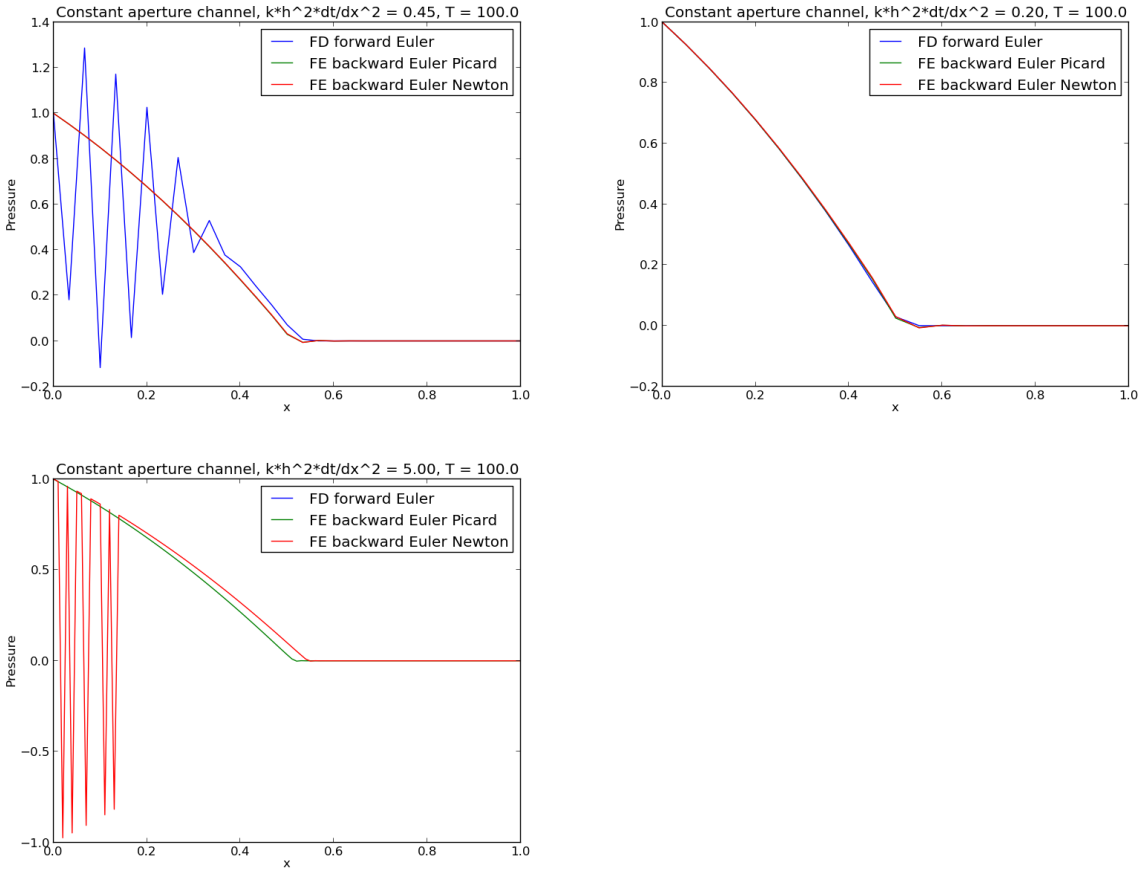

Figure 2: Examples with static channel aperture

A general observation is that the pressure moves like a wave and at there are some numerical artefacts (small fluctuations) at the front of the wave. This is well known even for linear diffusion problems near discontinuities/ sharp features.

Figure 3 shows that all three methods reach the correct steady state solution $P = \sqrt{1-x}$.

The code used to generate the above examples is found in 'FracFlow_static1D.py'.

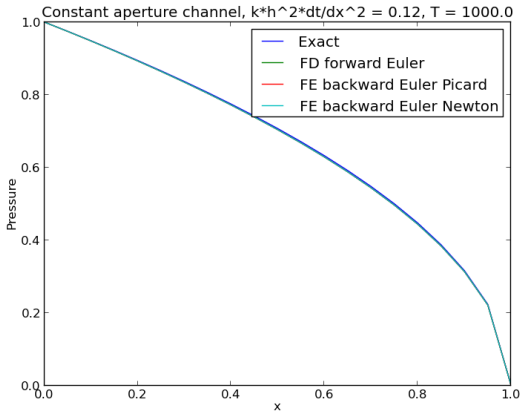

Figure 3: Steady state solution for static channel

4 The decoupled problem

The full problem

$$\begin{aligned} \frac{d}{dt}(hP) &= k\nabla\cdot\left(h^3\nabla P^2\right), \\ h &= aP + b\nabla^2 h. \end{aligned} \tag{39}$$

is decoupled for the special case when $b = 0$. In this case the opening is simply proportional to the pressure, which means that locally the pressure is evenly distributed on a plane so that all forces are perpendicular to the fracture plane. Inserting $h = aP$ in (39) gives the pressure equation

$$\frac{d}{dt}(P^2) = ka^2\nabla\cdot\left(P^3\nabla P^2\right) = \frac{2ka^2}{5}\nabla^2 P^5 = 2ka^2\nabla\cdot\left(P^4\nabla P\right), \tag{40}$$

or, in terns of the auxiliary variable $Q = P^2$,

$$\frac{dQ}{dt} = ka^2\nabla\cdot\left(Q^{3/2}\nabla Q\right) = \frac{2ka^2}{5}\nabla^2 Q^{5/2}. \tag{41}$$

If we again consider the 1D problem on a unit interval with boundary conditions $P(0,t) = 1$ and $P(1,t) = 0$ the exact steady state solution is $P = (1-x)^{1/5}$ (alternatively $Q = (1-x)^{2/5}$).

In principle only small adjustments should be necessary to solve this problem with the code used for the static problem. However, the finite element implementations keep giving '*nan*' values and I decided to not spend more time on it and rather focus on the finite difference implementation. The modified code is found in 'FracFlow_decoupled1D.py'.

Figure 4 shows a time sequence as the pressure is equilibrated on the unit interval using the forward Euler finite difference method. The blue curve is the exact steady state solution. The numerical result does converge towards this solution. The solution propagates like a wavefront through the domain. The stronger nonlinearity makes the front sharper than it was for the static aperture problem.

5 The coupled problem

For the coupled problem two alternative methods were attempted. The first is a FEniCS implementation where the pressure and aperture were solved iteratively in a Picard like manner. The variational form of the problem, phrased in FEniCS language, is:

```
a_p = (h_k + h0)*p*v*dx +
 dt*k*h_k**3*p_k*inner(nabla_grad(p), nabla_grad(v))*dx
L_p = (h_1 + h0)*p_1*v*dx
a_h = h*v*dx + b*inner(nabla_grad(h), nabla_grad(v))*dx
L_h = a*p_k*v*dx
```

(A small opening h_0 was included as a precautionary measure to ensure that the fracture always has capacity to hold some gas, even if it is closed, otherwise the pressure could diverge.) The algorithm is as follows at every time step:

1. Find new opening given a pressure;
2. Solve the flow equation using the updated opening;
3. Iterate until the opening and pressure converge before proceeding to the next time step.

The code is found in 'FracFlow_coupled1D.py'.

After much work and frustration the program is not doing what I want. For some strange reason the linear form L_p always assembles to a zero vector, and therefore it gives the wrong pressure.

Another solver was written based on one of the examples provided on the FEniCS project webste [1]. In this version the inbuilt nonlinear solver was used and the coupled set of equations were combined and solved simultaneously. The solution method attempted was the Newton method. The program also allows the user chose the time discretisation scheme and mesh (it works in 1D and 2D). Because of problems with implementing Dirichlet noundary conditions this program uses natural boundary condition and looks at the evolution of an initial pressure and aperture configuration. The program runs, but not much seems to happen. Unfortunately there was not enough time to resolve these problems, but the code is submitted anyway in . 'FracFlow_coupled.py'.

6 Final remarks and conclusions

The ultimate aim of this project was to look at how bubbles escape from a fracture in 1D and 2D. Unfortunately I was not able to do all that I set out to do, but I intend

[1] http://fenicsproject.org/documentation/dolfin/dev/python/demo/pde/cahn-hilliard/python/documentation.html

to continue working on this problem. Much time was spent on trying to solve the full coupled problem, and as a result there was too little time to do more simple numerical experiments with the simplified models.

This is what has been achieved and learnt so far:

- Derived finite difference and finite element schemes for the simplified problem with a static fracture aperture in 1D and used it to show that the correct steady state is reached.
- For the simplified static problem it appears that a finite element scheme with Picard iterations is the most reliable scheme. When computational cost are accounted for one might consider using more time steps and a simpler forward Euler finite difference scheme.
- The decoupled problem (where the fracture opening is assumed to be proportional to the pressure) behaves similar to the problem with static aperture. The result is again a wave front moving across the interval, but this time the front is sharper. The stronger nonlinearity makes the problem more sensitive, i.e. smaller time steps are required.
- Numerical artefacts, i.e. fluctuations, are observed close to the wavefront. This is a feature that is carried over from linear diffusion problems near discountinuities.
- The coupled problem has been cast in a variational form suitable for solution with FEniCS. One of the programs that was written is trivially extendable to 2D.

Some final thoughts outstanding challenges:

- The first challenge is to get the FEniCS implementations to work properly.
- One may ask whether the given mathematical formulation is the adequate. For example, the equation for the aperture in terms of pressure would not work if the fracture walls are in contact. As long as a small opening always remain this might never become a problem.
- It is possible that the nonlinearity is not going to be sufficient to give bubbles that move through a fracture that opens and closes. To get such intermittent behaviour one might need to add some more physics. One aspect is wall cohesion: when the fracture walls are close together they may collapse and hinder flow. This could be implemented with a threshold value for the opening below which there is no flow. Unfortunately it appears that FEniCS does not allow discontinuous functions in the variational forms.
- Given that the finite difference model was much faster for the simpler problems one should consider implementing the coupled problem with this scheme. This may also give some more flexibility since one is not restricted to phrase everything in a form recognised by FEniCS (It would maybe be easier to include discontinuous functions, as suggested above).

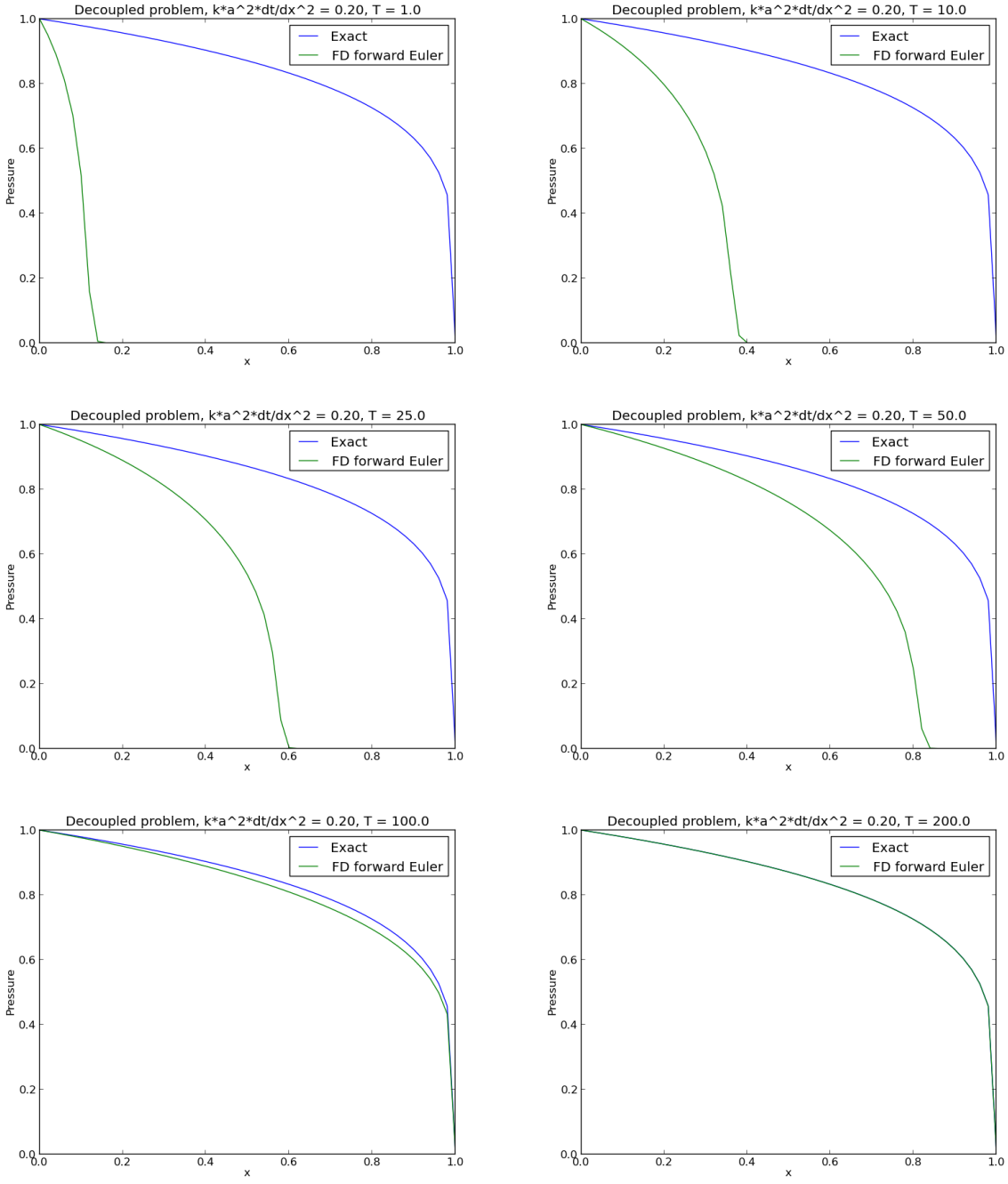

Figure 4: Pressure equilibration in the decoupled model