Aberystwyth University

Faculty of Business and Physical Sciences

Department of Mathematics

PhD thesis:

Effective time-space adaptive algorithm for hydraulic fracturing

Author:
Gaspare Da Fies

Supervisor:
Prof. Gennady Mishuris

Academic year 2019-2020

Declaration

Word count of thesis:

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Candidate Name: Date: Signed:

This thesis is the result of my own investigations, except where otherwise stated. Where correction services have been used, the extent and nature of the correction is clearly marked in a footnote(s). Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Candidate Name: Date: Signed:

I hereby give consent for my thesis, if accepted, to be available for photocopying and for interlibrary loan, and for the title and summary to be made available to outside organisations.

Candidate Name: Date: Signed:

Acknowledgements

This thesis is dedicated in particular to Maryam and to my parents, who have always been supporting me during the period of this PhD. A thought goes also to my friends, the old ones and the new ones that I have had the chance to meet in Aberystwyth.

I am thankful to Prof. Gennady Mishuris and to Aberystwyth University for giving me the great opportunity of doing this PhD. I also acknowledge the support of the EU project H2020-MSCA-RISE-2014-644175-MATRIXASSAY, which enabled me to undertake this research project and to attend numerous conferences and workshops:

- Wales Mathematics Colloquium, Gregynog, Wales, May 2016;
- Matrixassay Workshop, Dublin, Ireland, June 2016;
- IMSE 2016, Padua, Italy, July 2016;
- BAMC 2017, Surrey, UK, April 2017 (poster);
- Matrixassay Workshop, Loughborough, UK, May 2017 (talk);
- Wales Mathematics Colloquium, Gregynog, Wales, May 2017;
- ISMMS 2017, Augustow, Poland, June 2017 (talk);
- APM 2017, Saint Petersburg, Russia, June 2017 (talk);
- SPAN 2018, Padua, Italy, May 2018 (talk);
- ESMC 2018, Bologna, Italy, July 2018;
- Wales Mathematics Colloquium, Gregynog, Wales, May 2019 (talk).

Summary of the thesis

In this thesis we construct an accurate and effective numerical algorithm to solve the three classic 1D Hydraulic fracturing (HF) models: PKN, KGD and radial (also called penny-shaped). The solver works with power-law fluids (and Newtonian as a special case) and with impermeable rocks or with fluid leak-off modelled by classic Carter's formula. In KGD and radial models we also include the possibility to consider the effects of fluid induced shear stress on rock deformation and fracture propagation. The solver is adaptive in space and time, allowing at every stage of the computation to keep under control the error of the solution and to use the smallest possible number of approximation points.

A brief description of the general HF problem is presented, and the three classic 1D HF models are described in detail. A simplified, time independent self-similar version of the 1D models is also considered. HF models are characterised by a moving boundary, for this reason we use a suitable normalisation of the variables to move the problem to a fixed interval.

HF models are also characterised by an irregular (possibly singular), asymptotic behaviour of the solution at the crack mouth and at the crack tip, for this reason we use a carefully chosen change of variable to smooth the solution at the boundaries. Once the solution has been made smoother, it can be effectively approximated on the Chebyshev nodes with polynomials multiplied by the Jacobi weight function. Interpolation on Chebyshev nodes of a smooth function guarantees fast convergence and can be implemented efficiently using discrete cosine transform (or discrete Fourier transform). In the case of KGD and radial models we must also evaluate an integral operator that has an irregular kernel, this requires the use of another smoothing transformation. A fast and accurate way to evaluate the kernels of the integral operators using symmetric elliptic integrals is also proposed.

The solution of the problem is further complicated by the fact that the asymptotic behaviour of the solution can change when passing from a regime to another: storage or leak-off dominated and toughness or viscosity dominated. Therefore, the function spaces of the approximants must be chosen even more carefully to be able to keep into account the behaviour of the solution at any regime.

The self-similar and the time-dependent solvers that we propose are based on multigrid methods, where the solution is first found on a coarse grid and successively refined on denser grids, until the error is satisfactory.

Time discretisation is done using implicit Runge Kutta methods, that allow high order of convergence while remaining stable in stiff problems. Time step strategy is also adaptive and step length is chosen dynamically depending on the solution.

The solution is validated comparing it with some semi-analytical benchmarks present in literature and its convergence is thoroughly tested. Extensive computations prove that the numerical scheme is stable and efficient. It provides accurate results for all the three classic 1D hydraulic fracturing models, at all the regimes and with or without fluid leak-off. The algorithm can also be easily modified to work with different fluid and leak-off models.

Finally, we run several simulations with oscillating pumping rate, oscillating leak-off coefficient and oscillating toughness, to see how this affects the behaviour of the solution compared to the case with constant parameters.
In addition we investigate the effects of fluid induced shear stress on the fracture walls. We compare the results obtained with the modified KGD and radial models with those coming from the classic versions.

Contents

Chapter 1

Introduction

Hydraulic fracturing (HF) is a process in which large quantities of fluid, possibly containing small solid particles, is pumped into a fracture inside a rock layer. The high pressure exerted by the fluid subsequently causes the fracture to expand and extend. Fractures are usually very thin, usually some millimeters, but can be very long, even several hundred meters. This complicated solid-fluid interaction can be found in technological processes, like in fracking, geothermal energy or carbon-storage, but also in nature, like in magma intrusion in lithosphere, subglacial drainage of water or formation of mineral veins.

1.1 Short historical note on mathematical modelling of HF

The mathematical studies on fracture models started in the 1940s with Muskhelishvili [52] and Sneddon [70, 33]. In the 1950s their results were used to produce the first mathematical models of HF. In 1955 Khristianovic and Zheltov [39] studied the problem in the case of a planar fracture. In 1961 Perkins and Kern [59] created the so-called PK model. The crack is assumed to be very long and with constant height, therefore plane strain condition is assumed perpendicular to crack propagation direction, resulting in an elliptical cross section. In 1969 Geertsma and de Klerk [27] improved the model introduced by Khristianovic and Zheltov, creating the so-called KGD model. The fracture is assumed to be very short and with constant height, therefore the fracture can be considered as planar. In the same paper they also considered the case of a radial (or penny-shaped fracture). In this case the rock formation is assumed to be uniform and therefore the crack is axis-symmetric. In 1972 Nordgren [53] modified the PK-model adding the leak-off effect using Carter's formula, resulting in the so-called PKN model. The PKN, KGD and radial models constitute the three classic 1D models. Variations of them were generally used for designing treatments until the 1990s and are sometimes still used today.

While being effective in several cases, classic 1D models are not good with layered reservoirs where the height of the fracture cannot be assumed constant. The first attempts to address the problem were made in late 1970s [67] and led to the creation and development of the pseudo 3D (P3D) models during the 1980s. The main limitation of the P3D models is that the fluid flow is still 1D in space and fracture propagates in a fixed direction.

In the 1980s the development of planar 3D (PL3D) models [16] also started. In this case the fluid flow direction is no longer predetermined and the full elasticity equation is used to describe the relation between fluid pressure and fracture width. PL3D models are more accurate than P3D but computationally much more expensive. They are needed for example when the rock layer in the

middle of the fracture is stiffer than the ones above or below, resulting in a crack that does not have a convex shape.
There have also been attempts to model fully 3D HF models. See for example discrete elements method [19], finite elements method [79], boundary elements method [77]. The task is still computationally very heavy and there are unsolved questions related to the propagation of fractures in complex modes.

Since the early 2000s technological advances have made application of HF to shale formations more economic, this has led to a sharp increase in HF treatments and to a renewed interest in mathematical modelling. There has also been a comeback to the study of the fundamentals of HF. Much effort has been made to fully understand the classic models rather than to try to develop new models that keep into account more complex environmental conditions. The behaviour of the KGD and radial problems near the crack tip has been thoroughly studied using asymptotic theory. This was done for different combinations of toughness and leak-off regimes [47, 48]. Several numerical solutions for KGD and radial models have been computed for different combinations of levels of toughness (zero, small, finite, large) and leak-off (impermeable rocks, permeable). The studies carried on led to the understanding of the dominant factors in HF: fractures are controlled by leak-off or storage and by toughness, or viscosity. During its evolution, a fracture moves among these regimes.

1.2 Motivation

Huge progresses have been made since the beginning of numerical modelling of HF but, despite complex 2D and 3D models are now available, the 1D case still remains challenging. In this work we will consider the three classic 1D HF models: PKN, KGD and radial (or penny-shaped). Indeed it is important to get full understanding of these simplified models, because they can provide useful information about the more general cases. For instance the near tip behaviour of KGD and radial models is the same as the one of the planar 2D case. The 1D problem can also be used to see the effect of changing the problem parameters, or to test different fluid models, modified elasticity operators and fracture propagation conditions. Moreover a new solving method for the 1D problem might be generalisable to higher dimensional cases or offer new ideas to improve what already existing.

Although PKN, KGD and radial are 1D models, the unique features associated prevents the use of standard approaches. Indeed the main difficulties arise from:

- presence of a moving boundary;
- strong non-linearity coming from the moving boundary and the fluid flow equation;
- irregular asymptotic behaviour of the variables at the crack tip and mouth;
- change in the asymptotic behaviour of the variables when changing regime;
- presence of an integral operator (in KGD and radial).

Much research has been done on HF numerical simulation in the recent years [2, 22, 28, 29, 30, 31, 41, 42, 51, 54, 57, 58, 65, 82, 83]. In particular we will take as a starting point for our work the solver for PKN, KGD and radial models, with power-law fluid rheology, developed in [56, 61, 80].This solver in many situations produces remarkable performances, however the case of small toughness still remains a challenge and Carter's model for fluid leak-off has not yet been implemented. It also shows some limitations in the case of the time-dependent problem, when difficult boundary conditions are supplied.

The aim of our work is to build an updated solver that overcomes these problems and is able to provide a fast and accurate solution in all conditions. In addition our solver will be also adaptive in time and space, allowing to keep the error under control at every step, while at the same time minimising the amount of computations required.
To reach our goal we will start from the main points presented in the aforementioned papers:

- normalisation of the problem to pass from a moving to a fixed boundary;
- appropriate treatment of the asymptotic behaviour of the solution at the boundary;
- modified formulation of the elasticity integral operator.

We underline that special attention must be given to the asymptotics, because of their irregularity. A good solver cannot be built without keeping this into account properly, in particular if we want to use effectively high order approximation. Indeed we will use high order approximation both in time and space, allowing us to obtain high accuracy with few discretisation points.
Finally we will implement in the KGD and radial models the modified elasticity equation and fracture propagation condition proposed in [81], that keeps into account the effect of shear stress.

1.3 Structure of the thesis

In chapter 2 we will discuss HF models. We will start stating the general equations that describe the physical phenomena involved in HF, then we will pass to the classic 1D models. For each PKN, KGD and radial models we will see which assumptions are taken to reduce the general problem to 1D and then write the relative governing equations. We will also see what is the asymptotic behaviour of the solution, which plays a fundamental role in the numerical solution of the problem, and finally describe the simplified self-similar problem, that does not depend on time.

In chapter 3 we will discuss the numerical methods we will use to approximate the variables of the problem in space. We will do it using polynomial interpolation on Chebyshev nodes, a method that allows a very good accuracy with few nodes if applied properly. We will see how to compute the interpolant, evaluate it and integrate it in a fast and stable way using the discrete cosine transform or in alternative the discrete Fourier transform. However in HF the variables have a non-smooth at the interval endpoints, therefore Chebyshev interpolation cannot be applied directly. We will see how to deal with the problem smoothing the solution through a suitable change of variable. In KGD and radial models an additional challenge is presented by the evaluation of the integral operator that appears in the elasticity equation. The task is made easier by the modified formulation given in [56, 80], but it still requires a special approach to be tackled effectively. Finally we will see how to compute both the kernels of KGD and radial in a fast and accurate way, expressing them in terms of symmetric elliptic integrals.

In chapter 4 we will describe our solver for the self-similar problem. We will see how to apply the methods described in chapter 3 to discretise the problem. Particular attention is needed in the KGD and radial models, where the asymptotic behaviour of the approximant must be carefully chosen to have a solver capable to work with all values of toughness (large, small and 0). Then the non-linear system resulting from discretisation will be solved with a fixed point iteration, like proposed in [56, 80] and in alternative using Newton's method, the latter implementation being preparatory for the solver of the time-dependent problem. In the implementation of Newton's method we will study the advantage of solving the arising linear systems using an iterative method instead of a direct one. Our solver will also use the idea of multigrid methods, where a solution is found on a coarse mesh and then is used as an initial guess on a denser one and so on. This allows to spare computational time performing a good part of the iterations with coarser grids while only few with the denser ones. Finally we will validate our solution comparing it with the semi-analytical benchmarks provided in [55, 56, 60, 61] and then evaluate the convergence properties of the solver.

In chapter 5 we will describe the solver for the time-dependent problem. The spacial discretisation is essentially the same that the one we used for the self-similar problem, apart from the case of PKN model with Carter leak-off that needs a special treatment for the asymptotics. For the temporal discretisation we will use some implicit Runge Kutta methods, that allow high order convergence while remaining stable in case of stiff problems. As in the self-similar case, the non-linear system resulting from discretisation will be solved with Newton's iteration and a multigrid technique. The time step will be chosen dynamically. Finally we will validate our solution and evaluate the convergence properties of the solver comparing it to the self-similar solution, and with some long time asymptotes in the case of Carter leak-off.

In chapter 6 we will test the solver for the 3 classic 1D HF model with different values of parameters and boundary conditions. We will see the effect of oscillating pumping rate, oscillating leak-off coefficient and oscillating toughness, compared to the case with constant parameters.

In chapter 7 we will discuss the modified versions of the classic KGD and radial models proposed in [55, 81]. In these new models elasticity equation and fracture propagation condition are modified to keep into account the effect of shear stress. Finally we will compare the solutions obtained with the modified and the classic models.

In chapter 8 we will discuss some ideas about possible future implementations, that include different fluid and leak-off models and a different change of variable to smooth the asymptotics.

Chapter 2

Hydraulic fracturing models

In Hydraulic fracturing (HF) several physical processes take place at the same time: fluid flow, rock deformation, rock fracture and, if the rocks are permeable, fluid leakage through the fracture walls. Each of these processes can be modelled through a specific set of equations.
The solution of the most general formulation of the problem is extremely challenging, but under some assumptions the general set of equations can be simplified producing some simpler models that it is possible to tackle effectively.

2.1 General model

We state here the equations that describe each process happening in HF. All formulas in this section are written in Cartesian coordinates and using Einstein summation convention. Einstein summation convention assumes that when an index variable appears twice in a single term, it implies the summation of that term over all the values of the index. For example, if the indices can range over the set $\{1, 2, 3\}$, then the equation

$$y = \sum_{i=1}^{3} c_i x_i \tag{2.1}$$

is simplified by the convention to

$$y = c_i x_i. \tag{2.2}$$

We will also use the following shorthands: ∂_t for $\partial/\partial t$, ∂_{tt} for $\partial^2/\partial t^2$ and $(\bullet)_{,i}$ for $\partial/\partial x_i$.

Physical constants

Here we list all the physical constants that will be used in the following.

- Rocks (elasticity and fracture):
  ρ_e - mass density
  E - Young's modulus
  ν - Poissons's ratio
  G - shear modulus
  K - bulk modulus
  λ_e - Lamé's first parameter

μ_e - Lamé's second parameter
K_{IC} - toughness.

– Fluid:
ρ_f - mass density
μ_n - Newtonian fluid viscosity
K_{pl} - Power-law fluid consistency index
n_{pl} - Power-law fluid behaviour index
$\mu_{c,0}$ - Carreau fluid viscosity at 0 shear rate
$\mu_{c,\infty}$ - Carreau fluid viscosity at infinte shear rate
λ_c - Carreau fluid time parameter
n_c - Carreau fluid power-law parameter.

– Rocks (porous medium):
ϕ_p - porosity
k_p - permeability.

Elasticity

In general in HF, linear elasticity is used to describe the rock deformation under load. Linear elasticity is based on the following sets of partial differential equations [68].
– Equations of motion (Newton's second law):

$$\sigma_{ij,j} + F_i = \rho_e \partial_{tt} u_i, \tag{2.3}$$

where $\sigma_{ij} = \sigma_{ji}$ is the Cauchy stress tensor, F_i are the body forces, ρ_e is the mass density and u_i is the displacement (3 equations).
– Strain-displacement equations for small deformation:

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}), \tag{2.4}$$

where $\varepsilon_{ij} = \varepsilon_{ji}$ is the strain (6 equations).
– Constitutive equations (Hooke's law) for an isotropic and homogeneous material:

$$\sigma_{ij} = \lambda_e \delta_{ij} \varepsilon_{kk} + 2\mu_e \varepsilon_{ij}, \tag{2.5}$$

where δ_{ij} is the Kronecker delta, λ_e and μ_e are Lamé's first and second parameters (6 equations). In particular $\mu_e = G$ where G is the shear modulus and $\lambda_e = K - \frac{2}{3}\mu_e$ where K is the bulk modulus. Boundary conditions can be given in terms of displacement, stress, or a combination of them.

Substituting (2.4) into (2.5) and then everything into (2.3) one obtains the displacement equation of elastodynamics (3 equations), also known as **Navier-Cauchy equation**:

$$\mu_e u_{i,jj} + (\mu_e + \lambda_e) u_{j,ij} + F_i = \rho_e \partial_{tt} u_i. \tag{2.6}$$

In general the HF process is very slow, much slower than the speed of the waves in the rocks, therefore the inertial term can be neglected, reducing to a static problem (note that this might not hold if the crack is very small and the pumping rate is very high). Moreover for small deformations in rocks body forces (gravity) can be neglected. We obtain

$$\mu_e u_{i,jj} + (\mu_e + \lambda_e) u_{j,ij} = 0. \tag{2.7}$$

Fracture

In HF, linear elastic fracture mechanics (LEFM) is used to describe fracture propagation. It is based on Griffith's criterion [34] that says that for a crack to propagate, the energy used to create new surface must be equal to the potential energy released from the elastic material. However, at the beginning Griffith's idea only worked for brittle materials, but later Irwin [37] was able to extend it to ductile materials, considering the contribution in energy dissipation of the plastic deformation near the crack tip. Irwin also introduced the concept of stress intensity factor K_I, a quantity that describes the asymptotic behaviour of the stresses at the crack tip in a linear elastic material. In case of brittle materials the elastic energy released during crack propagation, and consequently Griffith's criterion, can be expressed in terms of the stress intensity factor.
We consider the case when plane strain is assumed, fracture is mode I and the material is brittle (no plastic deformation), linear elastic, isotropic and homogeneous. Under these conditions [5] the fracture propagates when

$$K_I \geq K_{IC}, \tag{2.8}$$

where K_{IC} is the toughness of the material. In addition the stress intensity factor (therefore also the elastic energy released) can be also expressed in terms of the J-integral

$$K_I = \sqrt{\frac{EJ}{1-\nu^2}}, \tag{2.9}$$

where E is the Young's modulus, ν the Poisson's ratio. The J-integral [63] is defined as

$$J = \int_\Gamma \left(W dx_2 - t_i \frac{\partial u_i}{\partial x_1} ds \right) \tag{2.10}$$

$$W = \int_0^{\varepsilon_{ij}} \sigma_{ij} d\varepsilon_{ij} \tag{2.11}$$

$$t_i = \sigma_{ij} n_j, \tag{2.12}$$

where x_1 is the crack propagation direction, x_2 is the normal to the crack plane, Γ is an arbitrary path on the plane (x_1, x_2) clockwise around the apex of the crack, W is the density of strain energy, t_i is the surface traction. For linear elasticity [68] we have that

$$W = \frac{1}{2}\sigma_{ij}\varepsilon_{ij}. \tag{2.13}$$

Fluid flow

In HF the fluid is most commonly modelled as Newtonian, however to better describe the fluid behaviour, non-Newtonian models can be used. In particular we consider here the case of generalised Newtonian fluids, for which the viscosity is expressed as a function of the shear rate [36] whereas in the Newtonian case the fluid viscosity is assumed to be constant.
· Newtonian fluid:

$$\mu_{\text{eff}}(\dot\gamma) \equiv \mu_n, \tag{2.14}$$

where μ_{eff} is the effective viscosity, $\dot\gamma$ is the shear rate and μ_n is the viscosity (constant viscosity).
· Power-law fluid:

$$\mu_{\text{eff}}(\dot\gamma) = K_{pl}\dot\gamma^{n_{pl}-1}, \tag{2.15}$$

where K_{pl} is the flow consistency index and n_{pl} is the flow behaviour index. If $K_{pl} = \mu_n$ and $n_{pl} = 1$ we obtain the Newtonian fluid.
· Carreau fluid:

$$\mu_{\text{eff}}(\dot{\gamma}) = \mu_{c,\infty} + (\mu_{c,0} - \mu_{c,\infty}) \left(1 + (\lambda_c \dot{\gamma})^2\right)^{\frac{n_c - 1}{2}}, \tag{2.16}$$

where $\mu_{c,0}$ is the viscosity at zero shear rate, $\mu_{c,\infty}$ is the viscosity at infinite shear rate, λ_c is a time parameter and n_c is a power-law parameter. If $\mu_{c,0} = \mu_{c,\infty} = \mu_n$ we obtain the Newtonian fluid.

Fluid dynamics can be described by the following set of partial differential equations [1]:
– Equations of motion (Cauchy momentum equation):

$$\rho_f(\partial_t v_i + v_{i,j} v_j) = -p_{,i} + \tau_{ij,j} + F_i, \tag{2.17}$$

where ρ_f is the mass density, v is the flow velocity, p is the pressure, $\tau_{ij} = \tau_{ji}$ is the deviatoric stress tensor, F_i are the body forces (3 equations).
– Strain-flow velocity equations:

$$\varepsilon_{ij} = \frac{1}{2}(v_{i,j} + v_{j,i}), \tag{2.18}$$

where $\varepsilon_{ij} = \varepsilon_{ji}$ is the strain (6 equations).
– Shear rate-strain equation:

$$\dot{\gamma} = \sqrt{2\varepsilon_{ij}\varepsilon_{ji}}, \tag{2.19}$$

where $\dot{\gamma}$ is the shear rate (1 equation).
– Constitutive equations for an isotropic, homogeneous, incompressible generalised Newtonian fluid:

$$\tau_{ij} = 2\mu_{\text{eff}}(\dot{\gamma})\varepsilon_{ij}, \tag{2.20}$$

where μ_{eff} is the effective viscosity (6 equations).
– Mass conservation for an incompressible fluid (1 equation):

$$v_{i,i} = 0. \tag{2.21}$$

Boundary conditions can be given in terms of flow velocity, stress and pressure or as a combination of them.

Substituting (2.20), (2.19), (2.18), (2.21) into (2.17), one obtains the flow velocity equation of fluid dynamics (3 equations plus continuity):

$$\rho_f(\partial_t v_i + v_{i,j} v_j) = -p_{,i} + \mu_{\text{eff}} v_{i,jj} + \mu_{\text{eff},j}(v_{i,j} + v_{j,i}) + F_i \tag{2.22}$$

$$v_{i,i} = 0. \tag{2.23}$$

In the case of a Newtonian fluid $\mu_{\text{eff}} \equiv \mu_n$ and one obtains the **Navier-Stokes equation**:

$$\rho_f(\partial_t v_i + v_{i,j} v_j) = -p_{,i} + \mu_n v_{i,jj} + F_i \tag{2.24}$$

$$v_{i,i} = 0 \tag{2.25}$$

In general in HF Reynolds number is very small (creeping flow), therefore the inertial term can be neglected. The fracture is very thin, therefore body forces (gravity) can be neglected. We obtain

$$-p_{,i} + \mu_{\text{eff}} v_{i,jj} + \mu_{\text{eff},j}(v_{i,j} + v_{j,i}) = 0 \tag{2.26}$$

$$v_{i,i} = 0. \tag{2.27}$$

Fluid flow in a porous medium

In HF the leakage of the fluid into the surrounding rocks is modelled as a fluid flow in a porous medium, that can be described using the following partial differential equations [7]:
– Flow through an isotropic, homogeneous porous medium for an isotropic, homogeneous, incompressible, generalised Newtonian fluid [66]:

$$\mu_{\text{eff}}(\dot\gamma_{\text{eff}})\phi_p v_i = -k_p p_{,i}, \tag{2.28}$$

where μ_{eff} is the effective viscosity of the fluid, $\dot\gamma_{\text{eff}}$ is the effective shear rate, ϕ_p is the porosity of the medium, v_i the velocity of the fluid, k_p the permeability of the medium and p the pressure of the fluid (3 equations).
– Mass conservation for an incompressible fluid (1 equation):

$$v_{i,i} = 0. \tag{2.29}$$

Boundary conditions can be given in terms of fluid velocity, pressure or as a combination of them.

The flow through a porous medium can be also expressed in terms of the fluid flux using the relation $q_i = \phi_p v_i$ and one obtains (3 equations plus continuity):

$$\mu_{\text{eff}} q_i = -k_p p_{,i} \tag{2.30}$$

$$q_{i,i} = 0. \tag{2.31}$$

In the case of a Newtonian fluid $\mu_{\text{eff}} \equiv \mu_n$ and one obtains **Darcy's law**:

$$\mu_n q_i = -k_p p_{,i} \tag{2.32}$$

$$q_{i,i} = 0. \tag{2.33}$$

Usually in HF the fluid leakage flow is slower than crack propagation, so the fluid does not invade the area in front of the crack. Actually in some cases it has been seen that the fluid does not reach the tip at all leaving a small gap, the so called fluid lag. Also the pressure at the fracture walls changes slowly in space, therefore the leak-off can be considered locally and in the direction perpendicular to the fracture plane. So we can write $q = q_{n_f} n_f$, where n_f is the normal to the fracture plane, and the equations reduce to:

$$\mu_{\text{eff}} q_{n_f} = -k_p p_{,n_f} \tag{2.34}$$

$$q'_{n_f} = 0. \tag{2.35}$$

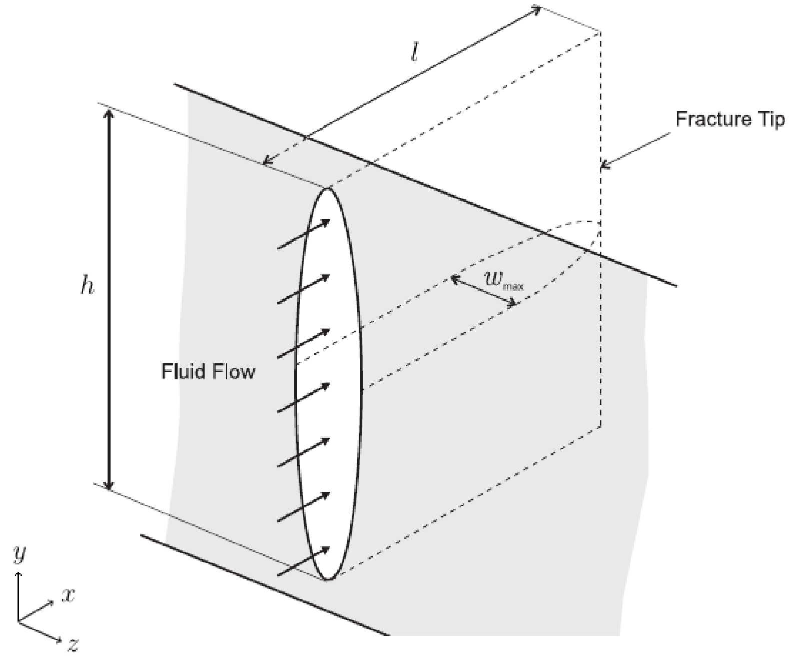
2.2 PKN

In 1961 Perkins and Kern [59] introduced the PK model, where they used the classical result for a plane strain crack by Sneddon [69]. Later in 1972 Nordgren [53] adapted the model adding fluid leak-off producing the PKN model.
We describe here the PKN model, we will state all the equations and boundary conditions. Then we will normalise the problem and describe the asymptotic behaviour of the variables at the crack tip. Finally we will give a simplified self-similar formulation that holds under some special assumptions.

Coordinates and variables

We define coordinates and variables for the PKN model.

Figure 2.1: Sketch of the PKN model, half fracture.
The height h is constant and the cross-section is elliptical. Adapted picture from [2].



- Coordinates (see figure 2.1):
  t time
  x-axis direction of propagation of the fracture
  y-axis direction of the height of the fracture
  z-axis direction of the width of the fracture.

- Variables (see figure 2.1):
  h crack height
  $l(t)$ crack length
  C_x cross section of the fracture (ellipse on the y, z-plane of centre $(x, 0, 0)$)
  $w(t, x, y)$ crack width
  $w_{max}(t, x) := \max\limits_{-h/2 \leq y \leq h/2} w(t, x, y) \equiv w(t, x, 0)$ maximum crack width over y
  $\bar{w}(t, x) := 1/h \int_{-h/2}^{h/2} w(t, x, y)dy \equiv \pi/4 \ w_{max}(t, x)$ crack width averaged over y
  $p(t, x)$ fluid absolute pressure (constant on C_x)
  $v(t, x, y, z)$ fluid velocity in the x direction
  $\bar{v}(t, x) := 1/A(C_x) \int_{C_x} v(t, x, y, z)dydz$ fluid velocity averaged over the cross section C_x
  $q(t, x) \equiv \int_{C_x} v(t, x, y, z)dydz = h\bar{w}(t, x)\bar{v}(t, x)$ fluid flow through the cross section C_x
  $\bar{q}(t, x) := q(t, x)/h = \bar{w}(t, x)\bar{v}(t, x)$ fluid flow per unit of y
  $q_l(t, x)$ fluid leak-off through the boundary of the cross section C_x
  $\bar{q}_l(t, x) := q_l(t, x)/h$ fluid leak-off per unit of y
  $Q_*(t)$ fluid flow at the wellbore.

We will express the problem in terms of the variables $\bar{w}(t,x), p(t,x), \bar{v}(t,x), \bar{q}(t,x), \bar{q}_l(t,x)$ that depend only on the coordinate $x \in [-l(t), l(t)]$. In the PKN model the crack is assumed to be symmetric with respect to the x-axis, so we can just consider $x \in [0, l(t)]$.

Elasticity

The elasticity equation relates the crack width to the fluid pressure. To simplify the problem the following assumptions are made:

- the material is linear elastic, homogeneous and isotropic
- the process is quasi-static, therefore inertial forces are neglected
- body forces (gravity) are neglected
- the crack is confined between two layers of tougher material, therefore height is constant
- crack length is much greater than crack height and width, therefore plane strain on the y,z-plane is assumed
- pressure is constant on the y,z-plane.

Under these conditions Sneddon's result for a plane strain crack can be used, therefore the cross section of the crack is elliptical and its width can be expressed as [78]

$$w(t,x,y) = \frac{2h(1-\nu^2)}{E}(p(t,x)-\sigma_c)\sqrt{1-\left(\frac{2y}{h}\right)^2}, \tag{2.36}$$

where E is Young's modulus, ν is Poisson's ratio and σ_c is the confining stress. We can express the width in terms of the average width

$$w(t,x,y) = \frac{4}{\pi}\bar{w}(t,x)\sqrt{1-\left(\frac{2y}{h}\right)^2} \tag{2.37}$$

and we obtain that

$$\bar{w}(t,x) = k_e(p(t,x)-\sigma_c), \tag{2.38}$$

where the constant k_e is

$$k_e = \frac{\pi h(1-\nu^2)}{2E}. \tag{2.39}$$

We note that for PKN model the relation between $\bar{w}$ and p is local: the value of $\bar{w}$ in x depends only on the value of p in x.

Fracture propagation

In the PKN model there is not an explicit fracture propagation criterion, in fact we assume that:

- the material is linear elastic, homogeneous and isotropic
- the material is brittle
- the fracture is confined between two layers of much higher toughness than the layer where the fracture is propagating, therefore crack height remains constant.
- the fluid is pumped in at a sufficient rate/pressure, so that fracture is always propagating and the effects of toughness at the crack tip can be ignored.

Indeed if the fluid is pumped at a sufficient rate, the fluid pressure far from the crack tip is high and it acts almost only on the confining layers and not on the crack tip. Only a small negligible area close to the tip interacts with crack propagation.
On the opposite, if the fluid pumping is stopped, the fluid pressure gets low along all the fracture. The action of the pressure on the confining layers decreases, while the area where the pressure interacts with fracture propagation is no longer negligible. Indeed it can be seen that with no pumping, a PKN fracture with no leak-off would go on propagating to infinite length, that is clearly not physical.

Continuity

The continuity equation describes mass conservation under the following assumptions:

- fluid is incompressible
- crack height is constant
- crack cross section is elliptical (from elasticity).

Under these conditions mass conservation can be expressed as [78]

$$\frac{\partial \bar{w}}{\partial t} + \frac{\partial \bar{q}}{\partial x} + \bar{q}_l = 0. \tag{2.40}$$

Fluid flow

The fluid flow equation relates the crack width, the fluid flow and the fluid pressure. To simplify the problem the following assumptions are made:

- the fluid is incompressible, homogeneous and isotropic
- the fluid is power-law (or Newtonian as a special case)
- Reynolds number is very small, therefore the flow is creeping (inertial forces are neglected)
- body forces (gravity) are neglected
- crack cross section is elliptical (from elasticity)
- crack height is much greater than width (eccentricity tends to 1)
- crack length is much greater than width, therefore crack width is quasi-constant
- leak-off is small, therefore fluid flow is quasi-constant
- pressure is constant on the y, z-plane.

Under these assumptions lubrication theory can be applied and fluid flow can be described by Poiseuille equation for a pipe with elliptical cross section with eccentricity that tends to 1 [43]

$$\frac{\partial p}{\partial x} = -k_f \frac{\bar{q}^{n_{pl}}}{\bar{w}^{2n_{pl}+1}}, \tag{2.41}$$

where the constant k_f is

$$k_f = 2K_{pl} \left(\frac{\pi(1 + \pi n_{pl} - n_{pl})}{2n_{pl}} \right)^{n_{pl}} \tag{2.42}$$

and where K_{pl} is the flow consistency index and n_{pl} is the flow behaviour index for a power-law fluid.
In particular in HF we are interested in Newtonian fluids ($n_{pl} = 1$ and $K_{pl} = \mu_n$) and in pseudo-plastic (or shear-thinning) fluids ($n_{pl} < 1$). Newtonian fluids have constant viscosity, pseudo-plastic fluids have a lower viscosity at higher shear rates.

Leak-off

The leak-off equation describes the fluid loss through the fracture walls. To simplify the problem the following assumptions are made:

- leak-off process is slower than fracture propagation, therefore the fluid remains in the area $|x| \leq l(t)$.
- the fracture is confined, therefore the fluid remains in the area $|y| > h/2$
- pressure at the crack surface changes slowly in space
- pressure at the crack surface changes slowly in time
- fluid is Newtonian.

The first three assumptions imply that leak-off can be considered perpendicular to the fracture plane and a 1D formula can be applied locally. The last two assumptions applied to the 1D flow in a porous medium give Carter's formula for the leak-off

$$\bar{q}_l = 2\frac{k_{cl}}{\sqrt{t - t_0(x)}}, \tag{2.43}$$

where k_{cl} is a constant that depends on the fluid and on the rock formation (usually estimated experimentally) and $t_0(x)$ is the time when the fluid has reached the point x. Carter's leak off formula has been derived for Newtonian fluids, nevertheless in HF it is commonly used also in combination with power-law fluid flow equation [2]. In our work we will do the same.

Carter leak-off is singular at the fluid front, if the fluid front coincides with the crack tip then $t_0^{-1}(t) = l(t)$ and the asymptotic expansion $(x \to l(t))$ can be written as

$$\frac{1}{\sqrt{t - t_0(x)}} = \sqrt{l'(t)}\frac{1}{\sqrt{l(t) - x}} + \frac{1}{4}\frac{l''(t)}{l'(t)^{\frac{3}{2}}}\sqrt{l(t) - x} + O((l(t) - x)^{\frac{3}{2}}) \tag{2.44}$$

and the derivation of (2.44) can be found in [40]. If instead the fluid front and the crack tip do not coincide, it follows immediately that the asymptotic expansion $(x \to t_0^{-1}(t))$ is

$$\frac{1}{\sqrt{t - t_0(x)}} = \sqrt{(t_0^{-1})'(t)}\frac{1}{\sqrt{t_0^{-1}(t) - x}} + \frac{1}{4}\frac{(t_0^{-1})''(t)}{(t_0^{-1})'(t)^{\frac{3}{2}}}\sqrt{t_0^{-1}(t) - x} + O((t_0^{-1}(t) - x)^{\frac{3}{2}}). \tag{2.45}$$

Fluid front

In general in HF the fluid front might not coincide with the crack tip, in this case we say that there is fluid lag. This happens if the fluid pressure $p(t, x)$ tends to get smaller than the pore pressure p_p close to the crack tip. In this case the fluid cannot reach the tip because of the pore pressure and therefore there is fluid lag.

In the PKN model from the elasticity equation (2.38) and because $\bar{w}(t, x) \geq 0$, we have that

$$p(t, x) \geq \sigma_c. \tag{2.46}$$

In general in HF we have that $\sigma_c > p_p$ (otherwise it would not be necessary to perform HF), therefore we get

$$p(t, x) > p_p, \tag{2.47}$$

so there is no need to introduce lag between the crack tip and the fluid and the fluid front coincides with the crack tip.

Boundary and initial conditions

To have a unique solution we need to set some boundary and initial conditions.
– The crack closes at the tip:

$$\bar{w}(t, l(t)) = 0. \tag{2.48}$$

– There is no outflow at the tip:

$$\bar{q}(t, l(t)) = 0. \tag{2.49}$$

– The inflow at the mouth is Q_* (the inflow splits in two because the crack is symmetric):

$$\bar{q}(t, 0) = \bar{q}_*(t) := Q_*(t)/(2h). \tag{2.50}$$

– The initial crack opening and length are:

$$\bar{w}(t_*, x) = \bar{w}_*(x), \quad l(t_*) = l_*. \tag{2.51}$$

We note that if the pumping rate is high enough for the given fracture volume, then the general assumptions for HF do not hold. Indeed in this case the inertial terms cannot be neglected because crack propagation speed might reach the speed of the waves in the rock and the fluid flow might become turbulent. As a consequence in this situation the model does not produce a meaningful result.
We also note that sensitivity analysis to the initial condition [44, 46] has shown that the influence of the initial conditions for the problem of hydraulic fracturing decays rapidly over time. Therefore a difference in the initial condition is rapidly absorbed at the beginning of the process and after that the evolution of the fracture depends only on the pumping rate. As a consequence the choice of the initial condition is not crucial for the final quality of the simulations.
For this reason, one possible is to start the simulation from a crack with small volume compared to the pumping rate, no matter the shape. Indeed despite the fact that at the beginning the simulation might not be meaningful, very rapidly the solution enters the area of validity of the model and at that point the effect of the shape of the initial condition has already been reabsorbed. A common way to set the initial conditions is to take the self-similar solution (see below) at a small value of time and with a compatible pumping rate to the one we want to apply.

Normalisation

The problem can be normalised [61], through some rescaling:

$$\begin{gathered}
\tilde{x} = \frac{x}{l(t)}, \quad \tilde{t} = \frac{t}{t_r}, \quad \tilde{t}_* = \frac{t_*}{t_r}, \quad t_r = (k_e k_f)^{\frac{1}{n_{pl}}}, \quad \tilde{t}_0(L(\tilde{t})\tilde{x}) = \frac{t_0(x)}{t_r}, \\
L(\tilde{t}) = l(t), \quad L_* = l_*, \quad \tilde{w}(\tilde{t}, \tilde{x}) = \bar{w}(t, x), \quad \tilde{w}_*(\tilde{x}) = \bar{w}_*(x), \quad \tilde{p}(\tilde{t}, \tilde{x}) = k_e(p(t, x) - \sigma_c), \\
\tilde{q}(\tilde{t}, \tilde{x}) = t_r \bar{q}(t, x), \quad \tilde{q}_*(\tilde{t}) = t_r \bar{q}_*(t), \quad \tilde{q}_l(\tilde{t}, \tilde{x}) = t_r \bar{q}_l(t, x), \quad \tilde{k}_{cl} = 2\sqrt{t_r} k_{cl},
\end{gathered} \tag{2.52}$$

where the parameter $\tilde{k}_{cl}$ can assume values greater or equal to zero. One advantage of the normalisation is that the domain is now time independent and $\tilde{x} \in [0, 1]$.

The equations in the normalised variables become:
– **elasticity equation**

$$\tilde{w} = \tilde{p} \tag{2.53}$$

– **continuity equation**

$$\frac{\partial \tilde{q}}{\partial \tilde{x}} = L'\tilde{x}\frac{\partial \tilde{w}}{\partial \tilde{x}} - L\left(\frac{\partial \tilde{w}}{\partial \tilde{t}} + \tilde{q}_l\right) \tag{2.54}$$

– **fluid flow equation**

$$\frac{\partial \tilde{p}}{\partial \tilde{x}} = -L\frac{\tilde{q}^{n_{pl}}}{\tilde{w}^{2n_{pl}+1}} \tag{2.55}$$

– **leak-off equation**

$$\tilde{q}_l = \frac{\tilde{k}_{cl}}{\sqrt{\tilde{t} - L^{-1}(L\tilde{x})}} \tag{2.56}$$

– **boundary conditions**

$$\tilde{q}(\tilde{t}, 0) = \tilde{q}_*(\tilde{t}), \quad \tilde{q}(\tilde{t}, 1) = 0, \quad \tilde{w}(\tilde{t}, 1) = 0 \tag{2.57}$$

– **initial conditions**

$$L(\tilde{t}_*) = L_*, \quad \tilde{w}(\tilde{t}_*, \tilde{x}) = \tilde{w}_*(\tilde{x}). \tag{2.58}$$

From here on we will only consider the normalised PKN problem and we will omit the $\sim$ symbol for simplicity. Also we will simply call n the power-law fluid parameter n_{pl}.

Asymptotics

Doing some asymptotic analysis it is possible to find the asymptotic behaviour of the solution at the crack mouth and at the crack tip. First we note that the asymptotic behaviour of Carter leak-off is independent from the other equations and is:

$$q_l(t, x) \sim \sum_{i=0}^{\infty} q_{l_{0,i}}(t)x^i \quad \text{as} \quad x \to 0^+ \tag{2.59}$$

$$q_l(t, x) \sim \sum_{i=0}^{\infty} q_{l_{1,i}}(t)(1 - x)^{-\frac{1}{2}+i} \quad \text{as} \quad x \to 1^-, \tag{2.60}$$

where applying the normalisation to (2.44) we get

$$q_{l_{1,0}}(t) = \sqrt{\frac{L'(t)}{L(t)}}, \quad q_{l_{1,1}}(t) = \frac{L(t)^{\frac{1}{2}}L''(t)}{4L'(t)^{\frac{3}{2}}}. \tag{2.61}$$

We state now the asymptotic behaviour of the other variables. (Note that here and in the following we will use the notation $dp_{i,j}$ to refer to the asymptotic coefficients of $\frac{\partial p(t,x)}{\partial x}$. The name of the variable $dp_{i,j}$ has just been chosen to be recognisable and must not be intended as a differential).

Crack mouth, when $(x \to 0^+)$ the variables are well behaved:

$$w(t,x) \sim \sum_{i=0}^{\infty} w_{0,i}(t)x^i \tag{2.62}$$

$$q(t,x) \sim \sum_{i=0}^{\infty} q_{0,i}(t)x^i \tag{2.63}$$

$$p(t,x) \sim \sum_{i=0}^{\infty} p_{0,i}(t)x^i \tag{2.64}$$

$$\frac{\partial p(t,x)}{\partial x} \sim \sum_{i=0}^{\infty} dp_{0,i}(t)x^i. \tag{2.65}$$

Crack tip, when $(x \to 1^-)$ the asymptotic behaviour depends on the leak-off.
– In the case of no leak-off we have that:

$$w(t,x) \sim \sum_{i=0}^{\infty} w_{1,i}(t)(1-x)^{\frac{1}{n+2}+i} \tag{2.66}$$

$$q(t,x) \sim \sum_{i=0}^{\infty} q_{1,i}(t)(1-x)^{\frac{1}{n+2}+i} \tag{2.67}$$

$$p(t,x) \sim \sum_{i=0}^{\infty} p_{1,i}(t)(1-x)^{\frac{1}{n+2}+i} \tag{2.68}$$

$$\frac{\partial p(t,x)}{\partial x} \sim \sum_{i=0}^{\infty} dp_{1,i}(t)(1-x)^{\frac{1}{n+2}+i-1}. \tag{2.69}$$

– In the case of Carter leak-off we have that:

· when $n = 0$ the asymptotic behaviour is the same as for no leak-off,

· when $n > 0$ the leading term remains the same, but the others change:

$$w(t,x) = w_{1,0}(t)(1-x)^{\frac{1}{n+2}} + w_{1,1}(t)(1-x)^{\frac{1}{2}} + \text{h.o.t.} \tag{2.70}$$

$$q(t,x) = q_{1,0}(t)(1-x)^{\frac{1}{n+2}} + q_{1,1}(t)(1-x)^{\frac{1}{2}} + \text{h.o.t.} \tag{2.71}$$

$$p(t,x) = p_{1,0}(t)(1-x)^{\frac{1}{n+2}} + p_{1,1}(t)(1-x)^{\frac{1}{2}} + \text{h.o.t.} \tag{2.72}$$

$$\frac{\partial p(t,x)}{\partial x} = dp_{1,0}(t)(1-x)^{\frac{1}{n+2}-1} + dp_{1,1}(t)(1-x)^{-\frac{1}{2}} + \text{h.o.t.}, \tag{2.73}$$

where h.o.t. stands for higher order term. Here for simplicity we have reported only the first two terms, that is what we need to know for our implementation, nevertheless more terms can be derived, see for example [40] for the case $n = 1$.
– In the case of Carter leak-off at the long time asymptote we have that:

· when $n = 0$ the asymptotic behaviour does not change,

· when $n > 0$ the asymptotic behaviour completely changes:

$$w(t,x) \sim \sum_{i=0}^{\infty} w_{1,i}(t)(1-x)^{\frac{n+2}{4n+4}+i} \tag{2.74}$$

$$q(t,x) \sim \sum_{i=0}^{\infty} q_{1,i}(t)(1-x)^{\frac{1}{2}+i} \tag{2.75}$$

$$p(t,x) \sim \sum_{i=0}^{\infty} p_{1,i}(t)(1-x)^{\frac{n+2}{4n+4}+i} \tag{2.76}$$

$$\frac{\partial p(t,x)}{\partial x} \sim \sum_{i=0}^{\infty} dp_{1,i}(t)(1-x)^{\frac{n+2}{4n+4}+i-1}. \tag{2.77}$$

This happens because for large values of time [53], if the pumping rate does not grow too fast ($q_*(t) = o\left(t^{\frac{2n+1}{2n+2}}\right)$ as $t \to \infty$), the leak-off term $L\tilde{q}_l$ dominates the right hand side of the continuity equation (2.54) and therefore the two terms that depend on $\tilde{w}$ can be neglected.

Self-similar problem

In the case of no leak-off it is possible to find some self-similar solutions to the problem separating the time and space components. The variables can be written as

$$w(t,x) = \hat{w}(x)\phi_w(t), \quad p(t,x) = \hat{p}(x)\phi_p(t), \quad q(t,x) = \hat{q}(x)\phi_q(t), L(t) = \hat{L}\phi_L(t), \quad q_*(t) = \hat{q}_*\phi_q(t),$$

where the time-dependent functions are exponential or power-law, that depends on the parameter γ as shown in table 2.1.

Table 2.1: Behaviour in time of the self-similar solution for PKN model.

	$\phi_w(t)$	$\phi_p(t)$	$\phi_q(t)$	$\phi_L(t)$	ρ
exponential	$e^{\gamma t}$	$e^{\gamma t}$	$e^{(\gamma+\rho)t}$	$e^{\rho t}$	$\gamma + \frac{\gamma}{n+1}$
power-law	t^{γ}	t^{γ}	$t^{\gamma+\rho-1}$	t^{ρ}	$\gamma + \frac{n+\gamma}{n+1}$

The equations become time independent and can be rewritten as follows:

– **elasticity equation**

$$\hat{w} = \hat{p} \tag{2.78}$$

– **continuity equation**

$$\frac{\partial \hat{q}}{\partial x} = \hat{L}\left(\rho x \frac{\partial \hat{w}}{\partial x} - \gamma \hat{w}\right) \tag{2.79}$$

– **fluid flow equation**

$$\frac{\partial \hat{p}}{\partial x} = -\hat{L}\frac{\hat{q}^n}{\hat{w}^{2n+1}} \tag{2.80}$$

– **boundary conditions**

$$\hat{q}(0) = \hat{q}_*, \quad \hat{q}(1) = 0, \quad \hat{w}(1) = 0. \tag{2.81}$$

The asymptotic behaviour of the self-similar solution remains the same as the one of the time-dependent solution.
Interesting cases are the power-law solution when $\gamma = \frac{1}{2n+3}$ that has constant fluid flow, and the power-law solution when $\gamma = 0$ that has constant pressure, see table 2.1.
In general it is not possible to include Carter leak-off in the self-similar solution. However it is possible to include a predefined self-similar leak-off term, provided that the time-dependent component is chosen properly, see [61].

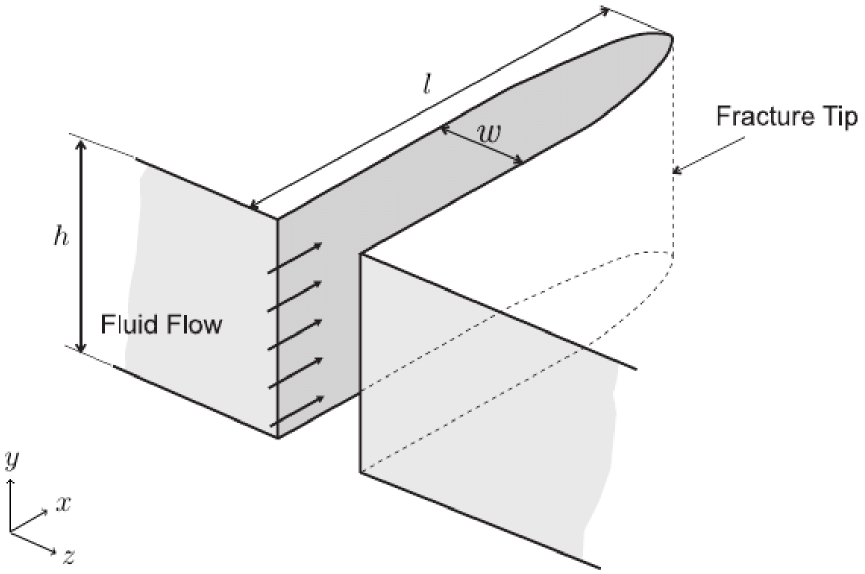
2.3 KGD

In 1955 Khristianovic and Zheltov [39] and independently in 1969 Geertsma and De Klerk [27] introduced the KGD model.
As we have done for PKN model, we describe here the KGD model, normalise it, give the asymptotic behaviour and the self-similar formulation.

Coordinates and variables

We define coordinates and variables for the KGD model.

Figure 2.2: Sketch of the KGD model, half fracture.
The height h is constant and the cross-section is rectangular. Adapted picture from [2].



- Coordinates (see figure 2.2):
  t time
  x-axis direction of propagation of the fracture
  y-axis direction of the height of the fracture
  z-axis direction of the width of the fracture.

- Variables (see figure 2.2):
  h crack height
  $l(t)$ crack length

C_x cross section of the fracture (rectangle on the y, z plane of centre $(x, 0, 0)$)
$w(t, x)$ crack width (does not depend on y)
$p(t, x)$ fluid absolute pressure (constant on C_x)
$v(t, x, z)$ fluid velocity in the x direction (does not depend on y)
$\bar{v}(t, x) := 1/w(t, x) \int_{-w(t,x)/2}^{w(t,x)/2} v(t, x, z)dz$ fluid velocity averaged over the cross section C_x
$q(t, x) \equiv h \int_{-w(t,x)/2}^{w(t,x)/2} v(t, x, z)dz = hw(t, x)\bar{v}(t, x)$ fluid flow through the cross section C_x
$\bar{q}(t, x) := q(t, x)/h = w(t, x)\bar{v}(t, x)$ fluid flow per unit of y
$q_l(t, x)$ fluid leak-off through the boundary of the cross section C_x
$\bar{q}_l(t, x) := q_l(t, x)/h$ fluid leak-off per unit of y
$Q_*(t)$ fluid flow at the wellbore.

We will express the problem in terms of the variables $w(t, x), p(t, x), \bar{v}(t, x), \bar{q}(t, x), \bar{q}_l(t, x)$ that depend only on the coordinate $x \in [-l(t), l(t)]$. In the KGD model the crack is assumed to be symmetric with respect to the x-axis, so we can just consider $x \in [0, l(t)]$.

Elasticity

The elasticity equation relates the crack width to the fluid pressure. To simplify the problem the following assumptions are made:

- the material is linear elastic, homogeneous and isotropic
- the process is quasi-static, therefore inertial forces are neglected
- body forces (gravity) are neglected
- the crack is symmetric with respect to the y, z-plane
- the crack is confined between two layers of tougher material, therefore height is constant
- crack height is much greater than crack length and width (plane strain is assumed on the x, z-plane)
- pressure is constant on the y, z-plane.

Under these assumptions, from linear elasticity theory comes the following hyper-singular integral equation that relates crack width and fluid pressure [80]

$$p(t, x) - \sigma_c = -\frac{E}{4\pi(1-\nu^2)} \int_{-l(t)}^{l(t)} \frac{\partial w(t, s)}{\partial s} \frac{ds}{s - x}. \tag{2.82}$$

Because of the symmetry of the crack the equation can be rewritten as

$$p(t, x) - \sigma_c = -\frac{E}{2\pi(1-\nu^2)} \int_{0}^{l(t)} \frac{\partial w(t, s)}{\partial s} \mathcal{I}(x, s) ds \tag{2.83}$$

$$\mathcal{I}(x, s) = \frac{s}{s^2 - x^2} \tag{2.84}$$

and we note that for a symmetric crack to have a finite value of $p(t, 0)$ the condition $\frac{\partial w(t,0)}{\partial x} = 0$ must be fulfilled. It is possible to invert (2.83) obtaining [71] the following weakly-singular integral equation [80]

$$-\frac{E}{2\pi(1-\nu^2)} \int_{x}^{l(t)} \frac{\partial w(t, s)}{\partial s} ds = \frac{4}{\pi^2} \int_{0}^{l(t)} (p(t, s) - \sigma_c) \mathcal{J}(t, x, s) ds \tag{2.85}$$

$$\mathcal{J}(t, x, s) = \frac{1}{2} \ln \left| \frac{\sqrt{l^2(t) - x^2} + \sqrt{l^2(t) - s^2}}{\sqrt{l^2(t) - x^2} - \sqrt{l^2(t) - s^2}} \right|, \tag{2.86}$$

so we can express w as a function of p

$$w(t,x) = k_e \int_0^{l(t)} (p(t,s) - \sigma_c)\mathcal{J}(t,x,s)ds, \tag{2.87}$$

where the constant k_e is

$$k_e = \frac{8(1-\nu^2)}{\pi E}. \tag{2.88}$$

It is also possible to separate the leading asymptotic term in (2.87) and obtain an alternative formulation where the kernel of the integral operator is continuous [80]

$$\int_0^{l(t)} (p(t,s) - \sigma_c)\mathcal{J}(t,x,s)ds = \int_0^{l(t)} \frac{\partial p(t,s)}{\partial s}\mathcal{K}(t,x,s)ds + \sqrt{l^2(t) - x^2}\int_0^{l(t)} \frac{p(t,s) - \sigma_c}{\sqrt{l^2(t) - s^2}}ds \tag{2.89}$$

$$\mathcal{K}(t,x,s) = \frac{x-s}{2}\ln\left|\frac{\sqrt{l^2(t)-x^2} + \sqrt{l^2(t)-s^2}}{\sqrt{l^2(t)-x^2} - \sqrt{l^2(t)-s^2}}\right| - \frac{x}{2}\ln\left(\frac{l^2(t) + xs + \sqrt{l^2(t)-x^2}\sqrt{l^2(t)-s^2}}{l^2(t) + xs - \sqrt{l^2(t)-x^2}\sqrt{l^2(t)-s^2}}\right), \tag{2.90}$$

so we can express w as

$$w(t,x) = k_e \int_0^{l(t)} \frac{\partial p(t,s)}{\partial s}\mathcal{K}(t,x,s)ds + k_e\sqrt{l^2(t) - x^2}\int_0^{l(t)} \frac{p(t,s) - \sigma_c}{\sqrt{l^2(t) - s^2}}ds. \tag{2.91}$$

We note that, differently from PKN model, in KGD model the relation between w and p is non-local: the values of w in x depend on the value of p along all the crack.

Fracture propagation

In the KGD model a fracture propagation criterion is combined with the elasticity equation. We assume that:

- the material is linear elastic, homogeneous and isotropic
- the material is brittle
- the crack is symmetric with respect to the y, z-plane
- crack height is much greater than crack length and width (plane strain is assumed on the x, z-plane)
- pressure is constant on the y, z-plane.

It is then possible to apply the criterion that the fracture propagates when the stress intensity factor K_I equals the toughness of the material K_{IC}. We consider only regimes in which the crack is always propagating, so we have

$$K_I = K_{IC}. \tag{2.92}$$

The stress intensity factor can be expressed as [80]

$$K_I(l(t)) = 2\sqrt{\frac{l(t)}{\pi}}\int_0^{l(t)} \frac{p(t,s) - \sigma_c}{\sqrt{l^2(t) - s^2}}ds. \tag{2.93}$$

Continuity

As for PKN, mass conservation is applied under the following assumptions:

- fluid is incompressible
- crack height is constant
- crack cross section is rectangular.

Under these conditions mass conservation can be expressed as [80]

$$\frac{\partial w}{\partial t} + \frac{\partial \bar{q}}{\partial x} + \bar{q}_l = 0. \tag{2.94}$$

Fluid flow

The fluid flow equation relates the crack width, the fluid flow and the fluid pressure. To simplify the problem the following assumptions are made:

- the fluid is incompressible, homogeneous and isotropic
- the fluid is power-law (or Newtonian as a special case)
- Reynolds number is very small, therefore the flow is creeping (inertial forces are neglected)
- body forces (gravity) are neglected
- crack cross section is rectangular
- crack height is much greater than width (aspect ratio tends to infinity)
- crack length is much greater than width, therefore crack width is quasi-constant
- leak-off is small, therefore fluid flow is quasi-constant
- pressure is constant on the y, z-plane.

Under these assumptions lubrication theory can be applied and fluid flow can be described by Poiseuille equation for a pipe with rectangular cross section and aspect ratio that tends to infinity [61]

$$\frac{\partial p}{\partial x} = -k_f \frac{\bar{q}^{n_{pl}}}{w^{2n_{pl}+1}} \tag{2.95}$$

where the constant k_f is

$$k_f = 2K_{pl} \left(\frac{2(2n_{pl}+1)}{n_{pl}} \right)^{n_{pl}} \tag{2.96}$$

and where K_{pl} is the flow consistency index and n_{pl} is the flow behaviour index for a power-law fluid.

Leak-off

What previously said in section 2.2, in the case of PKN model, holds also for KGD model and therefore the leak-off formula is the same.

Fluid front

In the case of a power-law fluid, if the presence of some fluid lag at the tip is not taken into account, it can be seen from asymptotic analysis that the KGD model admits solution only when the fluid parameter n is smaller than 2. Indeed from (2.122,2.126,2.130) it can be seen that if $n \geq 2$ then (2.105) is not integrable, therefore the problem does not admit solution. Anyway in HF we usually have $n \leq 1$, in which case a solution always exists also with no fluid lag.
However, assuming no fluid lag and $n \leq 1$, it might still happen that the value of $p(t,x) - \sigma_c$ at the crack tip becomes negative or even goes to $-\infty$. This contradicts the fact that the pressure of the fluid must always be greater or equal to the pore pressure $p(t,x) \geq p_p \geq 0$.
Therefore in the following we will consider HF problems where the confining stress is much greater than the pore pressure

$$\sigma_c \gg p_p \tag{2.97}$$

and as a consequence $p(t,x) \leq p_p$ only in a very small area close to the crack tip. In this case fluid lag can be neglected and the fluid front can be set to coincide with the crack tip. On the contrary, in case of small toughness and highly pressurised reservoir the influence of fluid lag might become relevant, for more details see for instance [20, 26].

Boundary and initial conditions

As for PKN model, to have a unique solution we need to set some boundary and initial conditions.
– The crack closes at the tip:

$$w(t, l(t)) = 0. \tag{2.98}$$

– There is no outflow at the tip:

$$\bar{q}(t, l(t)) = 0. \tag{2.99}$$

– The inflow at the mouth is Q_* (the inflow splits in two because the crack is symmetric):

$$\bar{q}(t, 0) = \bar{q}_*(t) := Q_*(t)/(2h). \tag{2.100}$$

– The initial crack opening and length are:

$$w(t_*, x) = w_*(x), \quad l(t_*) = l_*. \tag{2.101}$$

What previously said on the choice of the initial condition in section 2.2, in the case of PKN model, holds also for KGD.

Normalisation

In the same way as for PKN, the model can be normalised [61], through some rescaling.

$$\begin{gathered}\tilde{x} = \frac{x}{l(t)}, \quad \tilde{t} = \frac{t}{t_r}, \quad \tilde{t}_* = \frac{t_*}{t_r}, \quad t_r = (k_e k_f)^{\frac{1}{n_{pl}}}, \quad \tilde{t}_0(L(\tilde{t})\tilde{x}) = \frac{t_0(x)}{t_r}, \\ L(\tilde{t}) = l(t), \quad L_* = l_*, \quad \tilde{w}(\tilde{t}, \tilde{x}) = w(t,x), \quad \tilde{w}_*(\tilde{x}) = w_*(x), \quad \tilde{p}(\tilde{t}, \tilde{x}) = k_e(p(t,x) - \sigma_c), \quad (2.102) \\ \tilde{q}(\tilde{t}, \tilde{x}) = t_r \bar{q}(t,x), \quad \tilde{q}_*(\tilde{t}) = t_r \bar{q}_*(t), \quad \tilde{q}_l(\tilde{t}, \tilde{x}) = t_r \bar{q}_l(t,x), \quad \tilde{k}_{cl} = 2\sqrt{t_r} k_{cl}, \quad \tilde{K}_{IC} = \frac{\sqrt{\pi}}{2} k_e K_{IC},\end{gathered}$$

where the parameter $\tilde{k}_{cl}$ can assume values greater or equal to zero. The domain is now time independent and $\tilde{x} \in [0, 1]$.

The equations in the normalised variables become:
– **elasticity equation**

$$\tilde{w}(\tilde{t}, \tilde{x}) = L(\tilde{t}) \int_0^1 \frac{\partial \tilde{p}(\tilde{t}, \tilde{s})}{\partial \tilde{s}} \tilde{\mathcal{K}}(\tilde{x}, \tilde{s}) d\tilde{s} + \tilde{K}_{IC} \sqrt{L(\tilde{t})} \sqrt{1 - \tilde{x}^2} \tag{2.103}$$

$$\tilde{\mathcal{K}}(\tilde{x}, \tilde{s}) = \frac{\tilde{x} - \tilde{s}}{2} \ln \left| \frac{\sqrt{1 - \tilde{x}^2} + \sqrt{1 - \tilde{s}^2}}{\sqrt{1 - \tilde{x}^2} - \sqrt{1 - \tilde{s}^2}} \right| - \frac{\tilde{x}}{2} \ln \left(\frac{1 + \tilde{x}\tilde{s} + \sqrt{1 - \tilde{x}^2}\sqrt{1 - \tilde{s}^2}}{1 + \tilde{x}\tilde{s} - \sqrt{1 - \tilde{x}^2}\sqrt{1 - \tilde{s}^2}} \right) \tag{2.104}$$

– **fracture propagation**

$$\tilde{K}_{IC} = \sqrt{L(\tilde{t})} \int_0^1 \frac{\tilde{p}(\tilde{t}, \tilde{s})}{\sqrt{1 - \tilde{s}^2}} d\tilde{s} \tag{2.105}$$

– **continuity equation**

$$\frac{\partial \tilde{q}}{\partial \tilde{x}} = L' \tilde{x} \frac{\partial \tilde{w}}{\partial \tilde{x}} - L \left(\frac{\partial \tilde{w}}{\partial \tilde{t}} + \tilde{q}_l \right) \tag{2.106}$$

– **fluid flow equation**

$$\frac{\partial \tilde{p}}{\partial \tilde{x}} = -L \frac{\tilde{q}^{n_{pl}}}{\tilde{w}^{2n_{pl}+1}} \tag{2.107}$$

– **leak-off equation**

$$\tilde{q}_l = \frac{\tilde{k}_{cl}}{\sqrt{\tilde{t} - L^{-1}(L\tilde{x})}} \tag{2.108}$$

– **boundary conditions**

$$\tilde{q}(\tilde{t}, 0) = \tilde{q}_*(\tilde{t}), \quad \tilde{q}(\tilde{t}, 1) = 0, \quad \tilde{w}(\tilde{t}, 1) = 0 \tag{2.109}$$

– **initial conditions**

$$L(\tilde{t}_*) = L_*, \quad \tilde{w}(\tilde{t}_*, \tilde{x}) = \tilde{w}_*(\tilde{x}). \tag{2.110}$$

From here on we will only consider the normalised KGD problem and we will omit the $\sim$ symbol for simplicity. Also we will simply call n the power-law fluid parameter n_{pl}.

Asymptotics

As previously done for PKN, it is possible to find the asymptotic behaviour of the solution at the crack mouth and at the crack tip. We start with the asymptotic expansion of the kernel:

$$\mathcal{K}(x, s) \sim \sum_{i=0}^{\infty} \frac{B_{1-x^2}\left(\frac{3}{2}, i\right)}{2x^{2i}} \frac{\left(\frac{1}{2}\right)^{(i)}}{i!} s^{2i+1} \quad \text{as} \quad s \to 0^+ \quad \text{when} \quad 0 < x \leq 1 \tag{2.111}$$

$$\mathcal{K}(x, s) \sim \sum_{i=0}^{\infty} k_{1,i}(x)(1 - s)^{\frac{1}{2}+i} \quad \text{as} \quad s \to 1^- \quad \text{when} \quad 0 \leq x < 1 \tag{2.112}$$

$$\mathcal{K}(x, s) = x \log\left(\frac{1}{x}\right) + \frac{1}{2}(x - s)\log(|x - s|) + \text{h.o.t.} \quad \text{as} \quad s \to x \quad \text{when} \quad 0 \leq x \leq 1, \tag{2.113}$$

where $B_x(a,b)$ is the incomplete beta function [21, Eq. 8.17.1] and $m^{(i)}$ is the rising factorial and

$$k_{1,0}(x) = \sqrt{2-2x^2}, \quad k_{1,1}(x) = -\frac{7+x^2}{6\sqrt{2-2x^2}}. \tag{2.114}$$

The correctness of expression (2.111) has been verified numerically.

We have already seen in the PKN section the asymptotic expansion of the leak-off term, we state now the asymptotic behaviour of the other variables. For simplicity we state here only the asymptotic terms that we will need in our implementation but more terms can be derived if needed. For more details about the derivation of the asymptotics see [60, 61]. Note that in the following formulas the notation $[P]$ refers to the Iverson bracket, which in our case means that

$$[n=m] = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases} \tag{2.115}$$

Crack mouth, when $(x \to 0^+)$ the variables behave like:

$$w(t,x) = w_{0,0}(t) + w_{0,1}(t)x^2\log(x) + \text{h.o.t.} \tag{2.116}$$
$$q(t,x) = q_{0,0}(t) + q_{0,1}(t)x + \text{h.o.t.} \tag{2.117}$$
$$p(t,x) = p_{0,0}(t) + p_{0,1}(t)x + \text{h.o.t.} \tag{2.118}$$
$$\frac{\partial p(t,x)}{\partial x} = dp_{0,0}(t) + dp_{0,1}(t)x + \text{h.o.t.}\,. \tag{2.119}$$

Crack tip, when $(x \to 1^-)$ the asymptotic behaviour of the solution depends on the leak-off and on the toughness of the material.
– In the case $K_{IC} > 0$, with no leak-off or with Carter leak-off, we have that:

$$w(t,x) = w_{1,0}(t)(1-x)^{\frac{1}{2}} + w_{1,1}(t)(1-x)^{\frac{3-n}{2}}\log^{[n=0]}(1-x) + \text{h.o.t.} \tag{2.120}$$
$$q(t,x) = q_{1,0}(t)(1-x)^{\frac{1}{2}} + q_{1,1}(t)(1-x)^{\frac{3-n}{2}}\log^{[n=0]}(1-x) + \text{h.o.t.} \tag{2.121}$$
$$p(t,x) = p_{1,0}(t)(1-x)^{\frac{1-n}{2}}\log^{[n=1]}(1-x) + p_{1,*}(t) + p_{1,1}(t)(1-x)^{\frac{3-2n}{2}}\log^{[n=0]+[n=\frac{3}{2}]}(1-x) + \text{h.o.t.} \tag{2.122}$$
$$\frac{\partial p(t,x)}{\partial x} = dp_{1,0}(t)(1-x)^{-\frac{n+1}{2}} + dp_{1,1}(t)(1-x)^{\frac{1-2n}{2}}\log^{[n=0]}(1-x) + \text{h.o.t.}\,, \tag{2.123}$$

note that in (2.122) the order of the terms depends on n and the pressure is singular when $n \geq 1$.
– In the case $K_{IC} = 0$ with no leak-off we have that:

$$w(t,x) = w_{1,0}(t)(1-x)^{\frac{2}{n+2}} + \text{h.o.t.} \tag{2.124}$$
$$q(t,x) = q_{1,0}(t)(1-x)^{\frac{2}{n+2}} + \text{h.o.t.} \tag{2.125}$$
$$p(t,x) = p_{1,0}(t)(1-x)^{-\frac{n}{n+2}}\log^{[n=0]}(1-x) + p_{1,*}(t) + \text{h.o.t.} \tag{2.126}$$
$$\frac{\partial p(t,x)}{\partial x} = dp_{1,0}(t)(1-x)^{-\frac{2n+2}{n+2}} + \text{h.o.t.}\,, \tag{2.127}$$

note that in this case the pressure is always singular.
– In the case $K_{IC} = 0$ With Carter leak-off we have that:

$$w(t,x) = w_{1,0}(t)(1-x)^{\frac{n+4}{4n+4}} + \text{h.o.t.} \tag{2.128}$$

$$q(t,x) = q_{1,0}(t)(1-x)^{\frac{1}{2}} + \text{h.o.t.} \tag{2.129}$$

$$p(t,x) = p_{1,0}(t)(1-x)^{-\frac{3n}{4n+4}} \log^{[n=0]}(1-x) + p_{1,*}(t) + \text{h.o.t.} \tag{2.130}$$

$$\frac{\partial p(t,x)}{\partial x} = dp_{1,0}(t)(1-x)^{-\frac{7n+4}{4n+4}} + \text{h.o.t.}\,, \tag{2.131}$$

note that in this case both the pressure and the fluid velocity $v = q/w$ are always singular.

Self-similar problem

As for PKN, in the case of no leak-off it is possible to find some self-similar solutions to the problem separating the time and space components. The variables can be written as

$$\begin{aligned} w(t,x) = \hat{w}(x)\phi_w(t), \quad p(t,x) = \hat{p}(x)\phi_p(t), \quad q(t,x) = \hat{q}(x)\phi_q(t), \\ L(t) = \hat{L}\phi_L(t), \quad K_{IC}(t) = \hat{K}_{IC}\phi_{K_{IC}}(t), \quad q_*(t) = \hat{q}_*\phi_q(t), \end{aligned} \tag{2.132}$$

where the time-dependent functions are exponential or power-law, that depend on the parameter γ as shown in table 2.2.

Table 2.2: Behaviour in time of the self-similar solution for KGD model.

	$\phi_w(t)$	$\phi_p(t)$	$\phi_q(t)$	$\phi_L(t)$	$\phi_{K_{IC}}(t)$	ρ
exponential	$e^{\gamma t}$	$e^{(\gamma-\rho)t}$	$e^{(\gamma+\rho)t}$	$e^{\rho t}$	$e^{(\gamma-\rho/2)t}$	γ
power-law	t^{γ}	$t^{\gamma-\rho}$	$t^{\gamma+\rho-1}$	t^{ρ}	$t^{\gamma-\rho/2}$	$\gamma + \frac{n}{n+2}$

the equations become time independent and can be rewritten as follows:
– **elasticity equation**

$$\hat{w}(x) = \hat{L}\int_0^1 \frac{\partial \hat{p}(s)}{\partial s}\mathcal{K}(x,s)ds + \hat{K}_{IC}\sqrt{\hat{L}}\sqrt{1-x^2} \tag{2.133}$$

– **fracture propagation**

$$\hat{K}_{IC} = \sqrt{\hat{L}}\int_0^1 \frac{\hat{p}(s)}{\sqrt{1-s^2}}ds \tag{2.134}$$

– **continuity equation**

$$\frac{\partial \hat{q}}{\partial x} = \hat{L}\left(\rho x\frac{\partial \hat{w}}{\partial x} - \gamma\hat{w}\right) \tag{2.135}$$

– **fluid flow equation**

$$\frac{\partial \hat{p}}{\partial x} = -\hat{L}\frac{\hat{q}^n}{\hat{w}^{2n+1}} \tag{2.136}$$

– **boundary conditions**

$$\hat{q}(0) = \hat{q}_*, \quad \hat{q}(1) = 0, \quad \hat{w}(1) = 0. \tag{2.137}$$

Again the asymptotic behaviour of the self-similar solution remains the same as the one of the time-dependent solution.
Interesting cases are the power-law solution when $\gamma = \frac{1}{n+2}$ that has constant fluid flow and the power-law solution when $\gamma = \frac{n}{n+2}$ that has constant toughness, see table 2.1.
As for PKN it is not possible to include Carter leak-off in the self-similar solution. However it is possible to include a predefined self-similar leak-off term, provided that the time-dependent component is chosen properly, see [61].

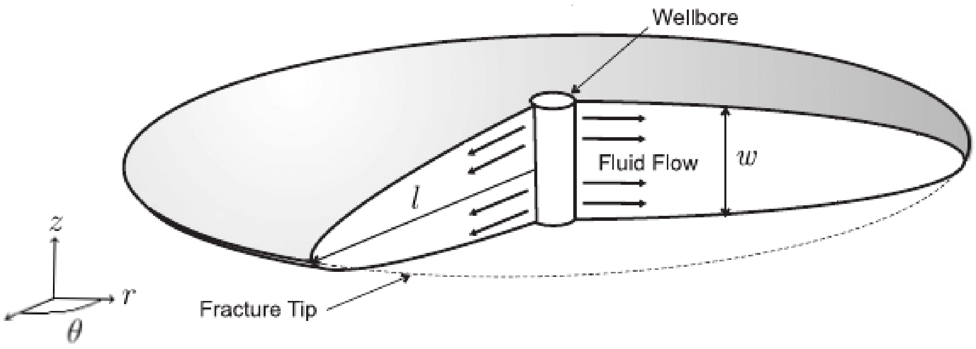
2.4 Radial

In 1969 Geertsma and De Klerk [27] introduced the radial model together with the KGD.
As we have done for PKN and KGD models, we describe here the radial model, normalise it, give the asymptotic behaviour and the self-similar formulation.

Coordinates and variables

We define coordinates and variables for the radial model.

Figure 2.3: Sketch of the radial or penny-shaped model.
The fracture is axis-symmetric. Adapted picture from [2].



- Coordinates (see figure 2.3):
  t time
  r-axis direction of propagation of the fracture
  θ-axis angle around the axis of radial symmetry
  z-axis direction of the width of the fracture.

- Variables (see figure 2.3):
  $l(t)$ crack length
  C_r cross section of the fracture (cylinder of centre $(0, 0, 0)$, axis z and radius r)
  $w(t, r)$ crack width (does not depend on θ)
  $p(t, r)$ fluid absolute pressure (constant on C_r)
  $v(t, r, z)$ fluid velocity in the r direction (does not depend on θ)
  $\bar{v}(t, r) := 1/w(t, r) \int_{-w(t,r)/2}^{w(t,r)/2} v(t, r, z)dz$ fluid velocity averaged over the cross section C_r
  $q(t, r) \equiv 2\pi r \int_{-w(t,r)/2}^{w(t,r)/2} v(t, r, z)dz = 2\pi r w(t, r)\bar{v}(t, r)$ fluid flow through the cross section C_r

$\bar{q}(t,r) := q(t,r)/(2\pi r) = w(t,r)\bar{v}(t,r)$ fluid flow per unit of θ
$q_l(t,r)$ fluid leak-off through the boundary of the cross section C_r
$\bar{q}_l(t,r) := q_l(t,r)/(2\pi r)$ fluid leak-off per unit of θ
$Q_*(t)$ fluid flow at the wellbore.

We will express the problem in terms of the variables $w(t,r), p(t,r), \bar{v}(t,r), \bar{q}(t,r), \bar{q}_l(t,r)$ that depend only on the coordinate $r \in [0, l(t)]$.

Elasticity

The elasticity equation relates the crack width to the fluid pressure. To simplify the problem the following assumptions are made:

- the material is linear elastic, homogeneous and isotropic
- the process is quasi-static, therefore inertial forces are neglected
- body forces (gravity) are neglected
- the crack is axis-symmetric
- the pressure is axis-symmetric.

Under these assumptions, from linear elasticity theory comes the following hyper-singular integral equation that relates crack width and fluid pressure [56]

$$p(t,r) - \sigma_c = -\frac{E}{2\pi(1-\nu^2)} \int_0^{l(t)} \frac{\partial w(t,s)}{\partial s} \mathcal{I}(r,s) ds \tag{2.138}$$

$$\mathcal{I}(r,s) = \begin{cases} \frac{1}{r} K\left(\frac{s^2}{r^2}\right) + \frac{r}{s^2-r^2} E\left(\frac{s^2}{r^2}\right), & s < r \\ \frac{s}{s^2-r^2} E\left(\frac{r^2}{s^2}\right), & s > r. \end{cases}, \tag{2.139}$$

where E and K are the complete elliptic integrals of the first and second kind. It is possible to invert (2.138) obtaining [71] the following weakly-singular integral equation [56]

$$-\frac{E}{2\pi(1-\nu^2)} \int_r^{l(t)} \frac{\partial w(t,s)}{\partial s} ds = \frac{4}{\pi^2} \int_0^{l(t)} (p(t,s) - \sigma_c) \mathcal{J}(t,r,s) ds \tag{2.140}$$

$$\mathcal{J}(t,r,s) = \begin{cases} \frac{s}{r} F\left(\arcsin\left(\sqrt{\frac{l^2(t)-r^2}{l^2(t)-s^2}}\right), \frac{s^2}{r^2}\right), & s < r \\ F\left(\arcsin\left(\sqrt{\frac{l^2(t)-s^2}{l^2(t)-r^2}}\right), \frac{r^2}{s^2}\right), & s > r, \end{cases} \tag{2.141}$$

where F is the incomplete elliptic integral of the first kind. So we can express w as a function of p

$$w(t,r) = k_e \int_0^{l(t)} (p(t,s) - \sigma_c) \mathcal{J}(t,r,s) ds, \tag{2.142}$$

where the constant k_e is the same as for KGD (2.88)

$$k_e = \frac{8(1-\nu^2)}{\pi E}. \tag{2.143}$$

It is possible to separate the leading asymptotic term in (2.140) and obtain an alternative formulation where the kernel of the integral operator is continuous [56]

$$\int_0^{l(t)} (p(t,s)-\sigma_c)\mathcal{J}(t,r,s)ds = \int_0^{l(t)} \frac{\partial p(t,s)}{\partial s}\mathcal{K}(t,r,s)ds + \sqrt{l^2(t)-r^2}\int_0^{l(t)} \frac{p(t,s)-\sigma_c}{\sqrt{l^2(t)-s^2}}\frac{s}{l(t)}ds \tag{2.144}$$

$$\mathcal{K}(t,r,s) = \begin{cases} s\left[E\left(\arcsin\left(\frac{s}{l(t)}\right)|\frac{r^2}{s^2}\right) - E\left(\arcsin\left(\frac{s}{r}\right)|\frac{r^2}{s^2}\right)\right], & s<r \\ s\left[E\left(\arcsin\left(\frac{s}{l(t)}\right)|\frac{r^2}{s^2}\right) - E\left(\frac{r^2}{s^2}\right)\right], & s>r, \end{cases} \tag{2.145}$$

where E is the incomplete elliptic integral of the second kind. So we can express w as

$$w(t,r) = k_e\int_0^{l(t)} \frac{\partial p(t,s)}{\partial s}\mathcal{K}(t,r,s)ds + k_e\sqrt{l^2(t)-r^2}\int_0^{l(t)} \frac{p(t,s)-\sigma_c}{\sqrt{l^2(t)-s^2}}\frac{s}{l(t)}ds. \tag{2.146}$$

We note that, as in KGD model, the relation between w and p is non-local: the values of w in x depend on the value of p along all the crack.

Fracture propagation

In the radial model a fracture propagation criterion is combined with the elasticity equation. We assume that:

- the material is linear elastic, homogeneous and isotropic
- the material is brittle
- crack is axis-symmetric
- pressure is axis-symmetric.

As for KGD it is then possible to apply the criterion that the fracture propagates when the stress intensity factor K_I equals the toughness of the material K_{IC}. We consider only regimes in which the crack is always propagating, so we have

$$K_I = K_{IC}. \tag{2.147}$$

The stress intensity factor can be expressed as [56]

$$K_I(l(t)) = 2\sqrt{\frac{l(t)}{\pi}}\int_0^{l(t)} \frac{p(t,s)-\sigma_c}{\sqrt{l^2(t)-s^2}}\frac{s}{l(t)}ds. \tag{2.148}$$

Continuity

As for PKN and KGD, mass conservation is applied under the following assumptions:

- fluid is incompressible
- crack is axis-symmetric

Under these conditions mass conservation can be expressed as [56]

$$\frac{\partial w}{\partial t} + \frac{1}{r}\frac{\partial(r\bar{q})}{\partial r} + \bar{q}_l = 0. \tag{2.149}$$

Fluid flow

The fluid flow equation relates the crack width, the fluid flow and the fluid pressure. To simplify the problem the following assumptions are made:

- the fluid is incompressible, homogeneous and isotropic
- the fluid is power-law (or Newtonian as a special case)
- Reynolds number is very small, therefore the flow is creeping (inertial forces are neglected)
- body forces (gravity) are neglected
- fluid flow is axis-symmetric
- crack length is much greater than width, therefore crack width is quasi-constant
- leak-off is small, therefore fluid flow is quasi-constant
- pressure is constant on the r, θ-cylinder.

Under these assumptions lubrication theory can be applied and fluid flow can be described by Poiseuille equation for a radial flow [56]

$$\frac{\partial p}{\partial x} = -k_f \frac{\bar{q}^{n_{pl}}}{w^{2n_{pl}+1}}, \tag{2.150}$$

where the constant k_f is the same as for KGD (2.96)

$$k_f = 2K_{pl}\left(\frac{2(2n_{pl}+1)}{n_{pl}}\right)^{n_{pl}} \tag{2.151}$$

and where K_{pl} is the flow consistency index and n_{pl} is the flow behaviour index for a power-law fluid.

Leak-off

What previously said in sections 2.2 and 2.3, in the case of PKN and KGD models, similarly holds also for radial model and therefore the leak-off formula is the same.

Fluid lag

What previously said in section 2.3 in the case of KGD model, similarly holds also for radial model and again we will consider only the case when there is no fluid lag.

Boundary and initial conditions

As for PKN and KGD models, to have a unique solution we need to set some boundary and initial conditions.
– The crack closes at the tip:

$$w(t, l(t)) = 0. \tag{2.152}$$

– There is no outflow at the tip:

$$\bar{q}(t, l(t)) = 0. \tag{2.153}$$

– The inflow at the mouth is Q_*:

$$\lim_{r\to 0}(r\bar{q}(t,r)) = \bar{q}_*(t) := Q_*(t)/(2\pi). \tag{2.154}$$

– The initial crack opening and length are:

$$w(t_*, r) = w_*(r), \quad l(t_*) = l_*. \tag{2.155}$$

What previously said on the choice of the initial condition in sections 2.2 and 2.3, in the case of PKN and KGD models, holds also for radial.

Normalisation

In the same way as for PKN and KGD, the model can be normalised [56], through some rescaling.

$$\begin{gathered}\tilde{r} = \frac{r}{l(t)}, \quad \tilde{t} = \frac{t}{t_r}, \quad \tilde{t}_* = \frac{t_*}{t_r}, \quad t_r = (k_e k_f)^{\frac{1}{n_{pl}}}, \quad \tilde{t}_0(L(\tilde{t})\tilde{r}) = \frac{t_0(r)}{t_r}, \\ L(\tilde{t}) = l(t), \quad L_* = l_*, \quad \tilde{w}(\tilde{t},\tilde{r}) = w(t,r), \quad \tilde{w}_*(\tilde{r}) = w_*(r), \quad \tilde{p}(\tilde{t},\tilde{r}) = k_e(p(t,r) - \sigma_c), \\ \tilde{q}(\tilde{t},\tilde{r}) = t_r\bar{q}(t,r), \quad \tilde{q}_*(\tilde{t}) = t_r\bar{q}_*(t), \quad \tilde{q}_l(\tilde{t},\tilde{r}) = t_r\bar{q}_l(t,r), \quad \tilde{k}_{cl} = 2\sqrt{t_r}k_{cl}, \quad \tilde{K}_{IC} = \frac{\sqrt{\pi}}{2}k_e K_{IC},\end{gathered} \tag{2.156}$$

where the parameter $\tilde{k}_{cl}$ can assume values greater or equal to zero. The domain is now time independent and $\tilde{x} \in [0,1]$.

The equations in the normalised variables become:
– **elasticity equation**

$$\tilde{w}(\tilde{t},\tilde{r}) = L(\tilde{t})\int_0^1 \frac{\partial\tilde{p}(\tilde{t},\tilde{s})}{\partial\tilde{s}}\tilde{\mathcal{K}}(\tilde{r},\tilde{s})d\tilde{s} + \tilde{K}_{IC}\sqrt{L(\tilde{t})}\sqrt{1-\tilde{r}^2} \tag{2.157}$$

$$\tilde{\mathcal{K}}(\tilde{r},\tilde{s}) = \begin{cases} \tilde{s}\left[E\left(\arcsin\left(\tilde{s}\right)|\frac{\tilde{r}^2}{\tilde{s}^2}\right) - E\left(\arcsin\left(\frac{\tilde{s}}{\tilde{r}}\right)|\frac{\tilde{r}^2}{\tilde{s}^2}\right)\right], & \tilde{s} < \tilde{r} \\ \tilde{s}\left[E\left(\arcsin\left(\tilde{s}\right)|\frac{\tilde{r}^2}{\tilde{s}^2}\right) - E\left(\frac{\tilde{r}^2}{\tilde{s}^2}\right)\right], & \tilde{s} > \tilde{r} \end{cases} \tag{2.158}$$

– **fracture propagation**

$$\tilde{K}_{IC} = \sqrt{L(\tilde{t})}\int_0^1 \frac{\tilde{p}(\tilde{t},\tilde{s})\tilde{s}}{\sqrt{1-\tilde{s}^2}}d\tilde{s} \tag{2.159}$$

– **continuity equation**

$$\frac{\partial(\tilde{r}\tilde{q})}{\partial\tilde{r}} = L'\tilde{r}^2\frac{\partial\tilde{w}}{\partial\tilde{r}} - L\tilde{r}\left(\frac{\partial\tilde{w}}{\partial\tilde{t}} + \tilde{q}_l\right) \tag{2.160}$$

– **fluid flow equation**

$$\frac{\partial\tilde{p}}{\partial\tilde{r}} = -L\frac{\tilde{q}^{n_{pl}}}{\tilde{w}^{2n_{pl}+1}} \tag{2.161}$$

– **leak-off equation**

$$\tilde{q}_l = \frac{\tilde{k}_{cl}}{\sqrt{\tilde{t} - \tilde{t}_0(L\tilde{r})}} \tag{2.162}$$

– **boundary conditions**

$$L(\tilde{t})\lim_{\tilde{r}\to 0}(\tilde{r}\tilde{q}(\tilde{t},\tilde{r})) = \tilde{q}_*(\tilde{t}), \quad \tilde{q}(\tilde{t},1) = 0, \quad \tilde{w}(\tilde{t},1) = 0 \tag{2.163}$$

– **initial conditions**

$$L(\tilde{t}_*) = L_*, \quad \tilde{w}(\tilde{t}_*,\tilde{r}) = \tilde{w}_*(\tilde{r}). \tag{2.164}$$

From here on we will only consider the normalised radial problem and we will omit the $\sim$ symbol for simplicity. Also we will simply call n the power-law fluid parameter n_{pl}.

Asymptotics

As previously done for PKN and KGD, it is possible to find the asymptotic behaviour of the solution at the crack mouth and at the crack tip. We start with the asymptotic expansion of the kernel:

$$\mathcal{K}(x,s) \sim \sum_{i=0}^{\infty} \frac{B_{1-x^2}\left(\frac{3}{2}, i+\frac{1}{2}\right)}{2x^{2i+1}} \frac{\left(\frac{1}{2}\right)^{(i)}}{i!} s^{2i+2} \quad \text{as} \quad s \to 0^+ \quad \text{when} \quad 0 < x \leq 1 \tag{2.165}$$

$$\mathcal{K}(x,s) \sim \sum_{i=0}^{\infty} k_{1,i}(x)(1-s)^{\frac{1}{2}+i} \quad \text{as} \quad s \to 1^- \quad \text{when} \quad 0 \leq x < 1 \tag{2.166}$$

$$\mathcal{K}(x,s) = x(1-x) + \frac{1}{2}(x-s)\log(|x-s|) + \text{h.o.t.} \quad \text{as} \quad s \to x \quad \text{when} \quad 0 \leq x \leq 1, \tag{2.167}$$

where $B_x(a,b)$ is the incomplete beta function, $m^{(i)}$ is the rising factorial and

$$k_{1,0}(x) = \sqrt{2-2x^2}, \quad k_{1,1}(x) = \frac{1-9x^2}{6\sqrt{2-2x^2}}. \tag{2.168}$$

The correctness of expression (2.165) has been verified numerically. Note that the kernels of KGD and radial have the same kind of asymptotic behaviour when $s \to 1^-$ and when $s \to x$ but the coefficients are different.

We have already seen in the PKN section the asymptotic expansion of the leak-off term, we state now the asymptotic behaviour of the other variables. For simplicity we state here only the asymptotic terms that we will need in our implementation but more terms can be derived if needed. For more details about the derivation of the asymptotics see [55, 56].
Crack mouth, when $(x \to 0^+)$ the variables behave like:

$$w(t,x) = w_{0,0}(t) + w_{0,1}(t)x^{2-n}\log^{[n=0]}(x) + \text{h.o.t.} \tag{2.169}$$

$$q(t,x) = q_{0,0}(t)x^{-1} + q_{0,1}(t)x + \text{h.o.t.} \tag{2.170}$$

$$p(t,x) = p_{0,*}(t) + p_{0,0}(t)x^{1-n}\log^{[n=1]}(x) + p_{0,1}(t)x^{3-2n}\log^{[n=0]+[n=\frac{3}{2}]}(x) + \text{h.o.t.} \tag{2.171}$$

$$\frac{\partial p(t,x)}{\partial x} = dp_{0,0}(t)x^{-n} + dp_{0,1}(t)x^{2-2n}\log^{[n=0]}(x) + \text{h.o.t.}. \tag{2.172}$$

note that in (2.171) the order of the terms depends on n and the pressure becomes singular when $n \geq 1$.
Crack tip, when $(x \to 1^-)$ the asymptotic behaviour is of the same kind of that of KGD.

Self-similar problem

As for PKN and KGD, in the case of no leak-off it is possible to find some self-similar solutions to the problem separating the time and space components. The variables can be written as

$$\begin{gathered} w(t,r) = \hat{w}(r)\phi_w(t), \quad p(t,r) = \hat{p}(r)\phi_p(t), \quad q(t,r) = \hat{q}(r)\phi_q(t), \\ L(t) = \hat{L}\phi_L(t), \quad K_{IC}(t) = \hat{K}_{IC}\phi_{K_{IC}}(t), \quad q_*(t) = \hat{q}_*\phi_q(t)\phi_L(t), \end{gathered} \tag{2.173}$$

Table 2.3: Behaviour in time of the self-similar solution for radial model.

	$\phi_w(t)$	$\phi_p(t)$	$\phi_q(t)$	$\phi_L(t)$	$\phi_{K_{IC}}(t)$	ρ
exponential	$e^{\gamma t}$	$e^{(\gamma-\rho)t}$	$e^{(\gamma+\rho)t}$	$e^{\rho t}$	$e^{(\gamma-\rho/2)t}$	γ
power-law	t^{γ}	$t^{\gamma-\rho}$	$t^{\gamma+\rho-1}$	t^{ρ}	$t^{\gamma-\rho/2}$	$\gamma+\frac{n}{n+2}$

where the time-dependent functions are exponential or power-law, that depend on the parameter γ as shown in table 2.3.

the equations become time independent and can be rewritten as follows:
– **elasticity equation**

$$\hat{w}(r)=\hat{L}\int_0^1 \frac{\partial \hat{p}(s)}{\partial s}\mathcal{K}(r,s)ds+\hat{K}_{IC}\sqrt{\hat{L}}\sqrt{1-r^2} \tag{2.174}$$

– **fracture propagation**

$$\hat{K}_{IC}=\sqrt{\hat{L}}\int_0^1 \frac{\hat{p}(s)s}{\sqrt{1-s^2}}ds \tag{2.175}$$

– **continuity equation**

$$\frac{\partial(r\hat{q})}{\partial r}=\hat{L}r\left(\rho r\frac{\partial \hat{w}}{\partial r}-\gamma\hat{w}\right) \tag{2.176}$$

– **fluid flow equation**

$$\frac{\partial \hat{p}}{\partial r}=-\hat{L}\frac{\hat{q}^n}{\hat{w}^{2n+1}} \tag{2.177}$$

– **boundary conditions**

$$\hat{L}\lim_{r\to 0}(r\hat{q}(r))=\hat{q}_*,\quad \hat{q}(1)=0,\quad \hat{w}(1)=0. \tag{2.178}$$

Again the asymptotic behaviour of the self-similar solution remains the same as the one of the time-dependent solution.
Interesting cases are the power-law solution when $\gamma=\frac{2-n}{3(n+2)}$ that has constant fluid input and the power-law solution when $\gamma=\frac{n}{n+2}$ that has constant toughness, see table 2.1.
As for PKN and KGD it is not possible to include Carter leak-off in the self-similar solution. However it is possible to include a predefined self-similar leak-off term, provided that the time-dependent component is chosen properly, see [56].

Chapter 3

Numerical approximation methods

We recall here some theoretical results about polynomial approximation. In hydraulic fracturing, the presence of singularities at the boundaries makes it impossible to approximate the functions using global polynomials straight out of the box. We will also have to deal with the kernels of the integral operators of KGD and radial models that are not smooth and will need a special approach as well. Through the use of some suitable changes of variable we will see how to use effectively Chebyshev interpolation despite these additional difficulties.
Finally we will discuss about the accurate computation of the kernels of the integral operators of KGD and radial models. This will be done through the use of symmetric elliptic integrals.

3.1 Chebyshev interpolation

General theory

For more details about this subsection see [75]. In general the interpolation of a function f with a polynomial p_n of degree n is not guaranteed to converge as $n \to \infty$, even for analytical functions. A famous example was given by Runge for the interpolation on equally spaced nodes. This problem can be overcome with a suitable choice of the interpolation nodes, like the Chebyshev nodes, that cluster at the boundary of the interval.
Two families of Chebyshev nodes are mainly used, one is formed by the zeros and the other by the extrema in $[-1, 1]$ of the Chebyshev polynomials of first kind. They share similar properties and both of them have some pros and cons in the implementation. We will use the extrema Chebyshev nodes, that are defined as

$$x_j = -\cos(\pi j/n), \quad j = 0, \dots, n \tag{3.1}$$

The main advantage to use the extrema instead of the zeros is that they include the endpoints, that makes easier to apply the boundary conditions in the differential equations. It must be also noted that the sets of $n = 2^m + 1$ nodes are nested.
Now we recall here a theorem about Chebyshev interpolation, that describes its convergence properties for differentiable functions.

Theorem 1. *For an integer $\nu > 0$, let f and its derivatives through $f^{(\nu-1)}$ be absolutely continuous on $[-1, 1]$ and suppose that the νth derivative is of bounded variation V. Then for any $n > \nu$ its interpolant on the extrema Chebyshev nodes satisfies*

$$\|f - p_n\| \leq \frac{4V}{\pi\nu(n-\nu)^{\nu}}. \tag{3.2}$$

As a consequence we have for example that the convergence rate for the function $|x|$ is $O(n^{-1})$ and for $x|x|$ is $O(n^{-2})$. More generally it can be seen that the result holds also for non-integer powers, indeed for example the convergence for the function $|x|^{1/2}$ is $O(n^{-1/2})$ and for $|x|^{4/3}$ is $O(n^{-4/3})$.
If the irregularity is at the endpoint of the interval one can verify that the rate of convergence is quadratically faster than if it is inside the interval. For example the convergence rate for $(1-x)^{1/2}$ is $O(n^{-1})$ and for $(1-x)^{4/3}$ is $O(n^{-8/3})$.
Finally a theorem that describes convergence for analytic functions.

Theorem 2. *Let a function f analytic in $[-1,1]$ be analytically continuable to the open Bernstein Ellipse E_p, where it satisfies $|f(x)| \leq M$. Then for each $n > 0$ its interpolant on the extrema Chebyshev nodes satisfies*

$$\|f - p_n\| \leq \frac{4M}{\rho^n(\rho - 1)}. \tag{3.3}$$

Summarising, the smoother the function, the faster the convergence. On the other hand, convergence for non-continuous functions is not guaranteed: for example it can be seen that for the function $\operatorname{sign}(x)$ there is indeed no convergence. The interpolant shows an oscillating behaviour around the jump, with the amplitude decaying at a slow rate $O(k^{-1})$, where k is the distance in nodes from the discontinuity.
To apply effectively Chebyshev interpolation is therefore crucial to have some regularity.

Computation of the polynomial interpolant

We give here the definitions of discrete Fourier transform (DFT), type-I discrete cosine and sine transforms (DCT-I,DST-I) and type-I discrete Chebyshev transforms (DTT-I,DUT-I).
We will extensively use DTT-I, DUT-I and their inverses to perform polynomial interpolation and integration, therefore it is important to have a fast implementation.
DST-I, DTT-I, DUT-I and their inverses can be all expressed in terms of DFT or DCT-I for which fast and accurate algorithms are widely available.

Discrete Fourier transform

The discrete fourier transform, (DFT) of a vector of complex numbers $x = (x_0, x_1, \ldots, x_{N-1})$ is another vector of complex numbers $X = (X_0, X_1, \ldots, X_{N-1})$ defined as follows:

$$X = \mathcal{F}(x), \quad \text{where} \quad X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i n k/N} = \sum_{n=0}^{N-1} x_n \left[\cos(2\pi nk/N) - i\sin(2\pi nk/N)\right]. \tag{3.4}$$

Its inverse can be expressed in terms of the DFT itself

$$\mathcal{F}^{-1}(X) = \frac{1}{N}\mathcal{F}(X^*)^*. \tag{3.5}$$

The DFT is widely used in digital signal processing and fast algorithms to perform it (FFT) with complexity $O(N\log(N))$ have been developed since the 1960s [17]. Libraries with the implementation of the FFT are widely available for any programming language and a direct implementation is present in most computational software.

Type I discrete cosine transform

In general, eight different types of DCT can be defined, we are interested only in the type I, so we will not treat the others. The type I discrete cosine transform (DCT-I) of a vector of real numbers $x = (x_0, x_1, \ldots, x_{N-1})$ is another vector of real numbers $X = (X_0, X_1, \ldots, X_{N-1})$ defined as follows:

$$X = \mathcal{C}(x), \quad \text{where} \quad X_k = \sum_{n=0}^{N-1}{}' x_n \cos\left(\pi n \frac{k}{N-1}\right) \tag{3.6}$$

$$= \frac{x_0}{2} + \sum_{n=1}^{N-2} x_n \cos\left(\pi n \frac{k}{N-1}\right) + \frac{x_{N-1}}{2}(-1)^k, \tag{3.7}$$

where the symbol $'$ after the sum means that the first and last terms are divided by two. Its inverse can be expressed in terms of the DCT-I itself

$$\mathcal{C}^{-1}(X) = \frac{2}{N-1}\mathcal{C}(X). \tag{3.8}$$

The DCT was first introduced in the 1970s [3]. A few years later algorithms based on the same idea of the FFT but using real instead of complex arithmetics, started to be developed [15]. Like for FFT, DCT complexity is $O(N\log(N))$ and libraries and implementations are widely available. In case a good direct implementation of DCT-I is not available, DCT-I can be expressed in terms of FFT, see Appendix A.1.

Type I discrete sine transform

As for the DCT, eight different types of DST can be defined, we will treat only the type I. The type I discrete sine transform (DST-I) of a vector of real numbers $x = (x_0, x_1, \ldots, x_{N-1})$ is another vector of real numbers $X = (X_0, X_1, \ldots, X_{N-1})$ defined as follows:

$$X = \mathcal{S}(x), \quad \text{where} \quad X_k = \sum_{n=0}^{N-1} x_n \sin\left(\pi(n+1)\frac{k+1}{N+1}\right). \tag{3.9}$$

Its inverse can be expressed in terms of the DST-I itself

$$\mathcal{S}^{-1}(X) = \frac{2}{N+1}\mathcal{S}(X). \tag{3.10}$$

Introduced together with the DCT, the DST can be computed with similar algorithms and same complexity $O(N\log(N))$.
If a good direct implementation of DST-I is not available, a N-terms DST-I can be expressed through a $N+2$-terms DCT-I as

$$\mathcal{S}(x_0, x_1, \ldots, x_{N-1}) = (Y_0 - Y_2, Y_1 - Y_3, \ldots, Y_{N-1} - Y_{N+1}), \quad \text{where} \tag{3.11}$$

$$(Y_0, Y_1, \ldots, Y_{N+1}) = \mathcal{C}\left(0, \frac{(x_0, x_1, \ldots, x_{N-1})}{2\sin\left(\pi(1, \ldots, N)/(N+1)\right)}, 0\right), \tag{3.12}$$

or alternatively

$$\mathcal{S}(x_0, x_1, \ldots, x_{N-1}) = 2\sin\left(\pi(1,\ldots,N)/(N+1)\right)(Y_1, Y_2, \ldots, Y_N), \quad \text{where} \tag{3.13}$$

$$(Y_0, Y_1, \ldots, Y_{N+1}) = \mathcal{C}(y_0, y_1, \ldots, y_{N-1}, 0, 0), \quad \text{and} \quad y_j = \sum_{\substack{i \bmod 2 = j \bmod 2 \\ i \geq j}} x_i, \tag{3.14}$$

where the y_j can be computed using two cumulative sums.
In case a good direct implementation of DCT-I is not available, DST-I can also be expressed in terms of FFT, see Appendix A.1.

Type I discrete T Chebyshev transform

As for the DCT and DST, different types of DTT can be defined, we will consider only what we will call the type I discrete T Chebyshev transform (DTT-I) for its similarity to DCT-I. The DTT-I of a vector of real numbers $x = (x_0, x_1, \ldots, x_{N-1})$ is another vector of real numbers $X = (X_0, X_1, \ldots, X_{N-1})$ defined as follows:

$$X = \mathcal{T}(x), \quad \text{where} \quad X_k = \sum_{n=0}^{N-1} x_n T_n(z_k). \tag{3.15}$$

In the definition above $T_n(z)$ are the Chebyshev polynomials of the first kind. They form a basis of the polynomial space and when $z \in [-1, 1]$ thay can be represented as

$$T_n(z) = \cos(n \arccos(z)). \tag{3.16}$$

The z_k are the extrema Chebyshev nodes (they are the extrema in the interval $[0, 1]$ of $T_{N-1}(z)$)

$$z_k = -\cos\left(\pi \frac{k}{N-1}\right), \quad k = 0, 1, \ldots, N-1. \tag{3.17}$$

This means that the computation of the transform $\mathcal{T}(x)$ corresponds to the evaluation of the polynomial $\sum_{n=0}^{N-1} x_n T_n(z)$ on the extrema nodes $z_0, \ldots, z_{N-1}$. On the opposite, the inverse transform $\mathcal{T}^{-1}(X)$ will give the coefficients in the Chebyshev basis of the interpolation polynomial that assumes values $X_0, \ldots, X_{N-1}$ on the extrema nodes $z_0, \ldots, z_{N-1}$.
Since the introduction of the DCT [3] the close relation between the DCT and Chebyshev interpolation was clear. Indeed it can be easily seen that a N-terms DTT-I can be expressed through a N-terms DCT-I as

$$\mathcal{T}(x_0, x_1, \ldots, x_{N-1}) = (Y_{N-1}, Y_{N-2}, \ldots, Y_0), \quad \text{where} \tag{3.18}$$

$$(Y_0, Y_1, \ldots, Y_{N-1}) = \mathcal{C}(2x_0, x_1, x_2, \ldots, x_{N-2}, 2x_{N-1}) \tag{3.19}$$

and for the inverse, that a N-terms IDTT-I can be expressed through a N-terms IDCT-I as

$$\mathcal{T}^{-1}(X_0, X_1, \ldots, X_{N-1}) = (y_0/2, y_1, y_2, \ldots, y_{N-2}, y_{N-1}/2), \quad \text{where} \tag{3.20}$$

$$(y_0, y_1, \ldots, y_{N-1}) = \mathcal{C}^{-1}(X_{N-1}, X_{N-2}, \ldots, X_0). \tag{3.21}$$

Therefore, knowing the values of a function on the Chebyshev nodes, it is possible to perform Chebyshev interpolation in a fast and stable way, that does not require matrix-vector products and has complexity $O(N \log(N))$. This, combined to the small sensitivity to the Runge phenomenon of the interpolation on the Chebyshev nodes, makes possible to use effectively high order polynomials for approximation.

Type I discrete U Chebyshev transform

As we did for the Chebyshev polynomial of the first kind, we do the same for the Chebyshev polynomials of the second kind and consider what we will call the type I discrete U Chebyshev transform (DUT-I). The DUT-I of a vector of real numbers $x = (x_0, x_1, \dots, x_{N-1})$ is another vector of real numbers $X = (X_0, X_1, \dots, X_{N-1})$ defined as follows:

$$X = \mathcal{U}(x), \quad \text{where} \quad X_k = \sum_{n=0}^{N-1} x_n U_n(z_k). \tag{3.22}$$

In the definition above $U_n(z)$ are the Chebyshev polynomials of the second kind. They form a basis of the polynomial space and when $z \in [-1, 1]$ thay can be represented as

$$U_n(z) = \frac{\sin((n+1)\arccos(z))}{\sin(\arccos(z))}. \tag{3.23}$$

The z_k are again the extrema Chebyshev nodes. As for the case above, the DUT-I corresponds to the evaluation of a polynomial, while the inverse computes the interpolation coefficients. Using the following relations between Chebyshev polynomials of the first and secon kind:

$$2T_n(z) = U_n(z) - U_{n-2}(z) \tag{3.24}$$

$$U_n + m \bmod 2 = \sum_{\substack{n \bmod 2 = m \bmod 2 \\ m \le n}} 2T_m, \tag{3.25}$$

it can be easily verified that a N-terms DUT-I can be expressed through a N-terms DTT-I as

$$\mathcal{U}(x_0, x_1, \dots, x_{N-1}) = 2\mathcal{T}(y_0/2, y_1, y_2, \dots, y_{N-1}), \quad \text{where} \tag{3.26}$$

$$y_j = \sum_{\substack{i \bmod 2 = j \bmod 2 \\ i \ge j}} x_i, \tag{3.27}$$

where the y_j above can be computed using two cumulative sums. In the case of the inverse, a N-terms IDUT-I can be expressed through a N-terms IDTT-I as

$$\mathcal{U}^{-1}(X_0, X_1, \dots, X_{N-1}) = (2y_0 - y_2, y_1 - y_3, y_2 - y_4, \dots, y_{N-3} - y_{N-1}, y_{N-2}, y_{N-1})/2, \quad \text{where} \tag{3.28}$$

$$(y_0, y_1, \dots, y_{N-1}) = \mathcal{T}^{-1}(X_0, X_1, \dots, X_{N-1}). \tag{3.29}$$

Using both DTT-I and DUT-I combined will be useful later to simplify the computation of the antiderivative of a polynomial.

Evaluation of the interpolant

We want to evaluate the sum of a polynomial of degree $N-1$ expressed in terms of the Chebyshev basis on a set of M generic points $z_k \in [-1, 1]$,

$$X = \mathcal{NT}(x, z), \quad \text{where} \quad X_k = \sum_{n=0}^{N-1} x_n T_n(z_k), \quad k = 0, \dots, M-1 \tag{3.30}$$

If one simply evaluates the sum doing a matrix vector product, this will take $O(MN)$ operations, in alternative one can reconduct to the non-uniform fast Fourier transform (NUFFT) or to the non-uniform discrete cosine transform (NUDCT), that allow to get the result in $O(\max(M, N\log(N)))$ operations.
Three types of non-uniform transforms can be defined [32]: type I where the frequences are non-uniform, type II where the sample points are non-uniform, type III where both frequences and sample points are non-uniform. We will consider here NUFFT-II and NUDCT-II.
NUFFT-II is defined as

$$X = \mathcal{NF}(x,\theta), \quad \text{where} \quad X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i n\theta_k}, \quad k = 0,\dots,M-1 \tag{3.31}$$

and NUDCT-II is defined as

$$X = \mathcal{NC}(x,\theta), \quad \text{where} \quad X_k = \sum_{n=0}^{N-1}{}' x_n \cos(\pi n\theta_k), \quad k = 0,\dots,M-1, \tag{3.32}$$

where the symbol $'$ after the sum means that the first and last terms are divided by two. We can express Chebyshev interpolation in terms of NUFFT-II as

$$\mathcal{NT}(x,z) = \sum_{n=0}^{N-1} x_n \cos(n\arccos(z_k)) = \Re\left(\sum_{n=0}^{N-1} x_n e^{-2\pi i n\theta_k}\right) = \mathcal{NF}(x,\theta), \tag{3.33}$$

$$\text{where} \quad \theta_k = -\frac{\arccos(z_k)}{2\pi}, \tag{3.34}$$

or in terms of NUDCT-II as follows

$$\mathcal{NT}(x,z) = \sum_{n=0}^{N-1} x_n \cos(n\arccos(z_k)) = \sum_{n=0}^{N-1}{}' x'_n \cos(\pi n\theta_k) = \mathcal{NC}(x',\theta), \tag{3.35}$$

$$\text{where} \quad \theta_k = \frac{\arccos(z_k)}{\pi} \quad \text{and} \quad x' = (2x_0, x_1, x_2, \dots, x_{n-2}, 2x_{n-1}). \tag{3.36}$$

The advantage of using the NUDCT instead of the NUFFT is basically to avoid complex arithmetic, dealing just with the real part, resulting in a better performance. We have implemented the NUDCT-II following the idea used in [6] to implement the NUFFT-II. We start writing

$$\mathcal{NC}(x,\theta_k) = \sum_{n=0}^{N-1}{}' x_n \cos(\pi n\theta_k) = \sum_{n=0}^{N-1}{}' x_n \cos\left(\pi n\frac{j_k}{N-1} + \pi n\Delta\theta_k\right), \tag{3.37}$$

$$\text{where} \quad j_k = \lfloor (N-1)\theta_k \rceil \quad \text{and} \quad \Delta\theta_k = \theta_k - \frac{j_k}{N-1}, \tag{3.38}$$

where with $\lfloor\rceil$ we denote the closest integer, therefore $0 \le j_k \le N-1$ is the integer for which $\frac{j_k}{N-1}$ is closest to θ_k. We can now apply the addition formula

$$= \sum_{n=0}^{N-1}{}' x_n \left[\cos\left(\pi n\frac{j_k}{N-1}\right)\cos(\pi n\Delta\theta_k) - \sin\left(\pi n\frac{j_k}{N-1}\right)\sin(\pi n\Delta\theta_k)\right] \tag{3.39}$$

and then approximate $\cos(\pi n \Delta\theta_k)$ and $\sin(\pi n \Delta\theta_k)$ using the Taylor expansion of cosine and sine

$$= \sum_{n=0}^{N-1}{}' x_n \cos\left(\pi n \frac{j_k}{N-1}\right) \sum_{r=0}^{R} (-1)^r \frac{(\pi n \Delta\theta_k)^{2r}}{(2r)!} - \sum_{n=0}^{N-1}{}' x_n \sin\left(\pi n \frac{j_k}{N-1}\right) \sum_{r=0}^{R} (-1)^r \frac{(\pi n \Delta\theta_k)^{2r+1}}{(2r+1)!} \tag{3.40}$$

$$= \sum_{r=0}^{R} \left[\sum_{n=0}^{N-1}{}' (-1)^r \frac{(\pi n)^{2r}}{(2r)!} x_n \cos\left(\pi n \frac{j_k}{N-1}\right) \right] \Delta\theta_k^{2r} - \sum_{r=0}^{R} \left[\sum_{n=0}^{N-1}{}' (-1)^r \frac{(\pi n)^{2r+1}}{(2r+1)!} x_n \sin\left(\pi n \frac{j_k}{N-1}\right) \right] \Delta\theta_k^{2r+1}. \tag{3.41}$$

If N is big enough (this case however is not relevant for the purposes of this thesis) the term $(\pi(N-1))^{2R+1}$ might suffer overflow and be numerically rounded to infinity and at the same time the term $\Delta\theta_k^{2r+1}$ might suffer underflow and be numerically rounded to 0. Therefore it is a good practice to divide $\Delta\theta$ by $\Theta = \max_k |\Delta\theta_k|$. Moreover we note that the first and last terms of the sine sum are zero, therefore

$$= \sum_{r=0}^{R} \left[\sum_{n=0}^{N-1}{}' (-1)^r \frac{(\pi n \Theta)^{2r}}{(2r)!} x_n \cos\left(\pi n \frac{j_k}{N-1}\right) \right] (\Delta\theta_k/\Theta)^{2r} + \tag{3.42}$$

$$\sum_{r=0}^{R} \left[\sum_{n=1}^{N-2} (-1)^{r+1} \frac{(\pi n \Theta)^{2r+1}}{(2r+1)!} x_n \sin\left(\pi n \frac{j_k}{N-1}\right) \right] (\Delta\theta_k/\Theta)^{2r+1}. \tag{3.43}$$

Now we can rewrite everything in terms of DCT-I and DST-I

$$= \sum_{r=0}^{R} \mathcal{C}_{j_k} \left(\dots, (-1)^r \frac{(\pi n \Theta)^{2r}}{(2r)!} x_n, \dots \right) (\Delta\theta_k/\Theta)^{2r} + \tag{3.44}$$

$$\sum_{r=0}^{R} \mathcal{S}'_{j_k} \left(\dots, (-1)^{r+1} \frac{(\pi n \Theta)^{2r+1}}{(2r+1)!} x_n, \dots \right) (\Delta\theta_k/\Theta)^{2r+1}, \tag{3.45}$$

$$\text{where} \quad \mathcal{S}'(x) = (0, \mathcal{S}(x_1, x_2, \dots, x_{N-2}), 0), \tag{3.46}$$

where the input vectors of $\mathcal{C}_{j_k}$ and $\mathcal{S}_{j_k}$ vary on n, and where with $\mathcal{C}_{j_k}$ and $\mathcal{S}_{j_k}$ we refer to the j_k-th term of the output vector of $\mathcal{C}$ and $\mathcal{S}$ respectively. We can put all together in a more compact form

$$\mathcal{NC}(x, \theta_k) = \sum_{s=0}^{S-1} Y_{j_k}^s (\Delta\theta_k/\Theta)^s, \quad \text{where} \tag{3.47}$$

$$Y^0, Y^1, \dots = \mathcal{C}(X^0), \mathcal{S}'(X^1), \mathcal{C}(X^2), \mathcal{S}'(X^3), \dots \tag{3.48}$$

$$X_n^s = (-1)^{\lceil \frac{s}{2} \rceil} \frac{(\pi n \Theta)^s}{s!} x_n \qquad n = 0, \dots, N-1, \tag{3.49}$$

where finally S doesn't have to be necessarily even. The computation of the NUDCT-II is then reduced to the computation of S transforms DCT-I/DST-I (3.48) and of the sum of the series in (3.47). The truncation error of the Taylor expansions can be estimated as

$$\max_{n,k} \left(\frac{(\pi n |\Delta\theta_k|)^S}{S!} \right) \le \frac{(\pi(N-1)\Theta)^S}{S!} \le \frac{(\pi/2)^S}{S!}. \tag{3.50}$$

This means for example that we can get an error smaller than 10^{-16} if we choose $S = 22$.
If the number of evaluation points M is much bigger than the number of sample points N, then the cost of computing (3.47) becomes much bigger then the cost of computing (3.48). In this case it can be useful to perform some oversampling to reduce computational time. Oversampling means adding some zeros at the end of the sequence x_n so that the grid where cosine and sine transforms are computed becomes denser. As a consequence the Taylor expansions in (3.40) will be evaluated on a smaller interval and therefore will require less terms. More in detail if we oversample to $N' > N$ we get

$$\mathcal{NC}(x,\theta) = \sum_{n=0}^{N-1} x_n \cos(\pi n \theta_k) = \sum_{n=0}^{N'-1} x_n \cos(\pi n \theta_k), \tag{3.51}$$

$$\text{where} \quad x_N = x_{N+1} = \ldots = x_{N'-1} = 0. \tag{3.52}$$

Then, proceeding as above with N' in place of N, we obtain

$$\mathcal{NC}(x,\theta_k) = \sum_{s=0}^{S-1} Y^s_{j'_k} (\Delta\theta'_k/\Theta')^s, \quad \text{where} \tag{3.53}$$

$$j'_k = \lfloor (N'-1)\theta_k \rceil, \quad \Delta\theta'_k = \theta_k - \frac{j'_k}{N'-1} \quad \text{and} \quad \Theta' = \max_k |\Delta\theta'_k| \tag{3.54}$$

$$Y^0, Y^1, \ldots = \mathcal{C}(X^0), \mathcal{S}'(X^1), \mathcal{C}(X^2), \mathcal{S}'(X^3), \ldots \tag{3.55}$$

$$X^s_n = (-1)^{\lceil \frac{s}{2} \rceil} \frac{(\pi n \Theta')^s}{s!} x_n \qquad n = 0, \ldots, N-1. \tag{3.56}$$

We can now estimate the truncation error as

$$\max_{n,k} \left(\frac{(\pi n |\Delta\theta'_k|)^S}{S!} \right) \le \frac{(\pi(N-1)\Theta')^S}{S!} \le \frac{\left(\frac{\pi}{2}\frac{N-1}{N'-1}\right)^S}{S!} = \left(\frac{N-1}{N'-1}\right)^S \frac{(\pi/2)^S}{S!}, \tag{3.57}$$

therefore incereasing the value of N' allows to take a smaller value of S. For example we can get an error smaller than 10^{-16} if we choose $N' = 8N$ and $S = 12$.

In the derivation above we have used the Taylor expansion to approximate the functions cosine and sine. Better results can be obtained if we use the Chebyshev expansion, similarly to what has been done in [64] for the NUFFT. Cosine and sine can be then expressed as

$$\cos(ax) = {\sum_{r=0}^{\infty}}' c_{2r}(a) T_{2r}(x) \quad x \in [-1,1] \tag{3.58}$$

$$\sin(ax) = \sum_{r=0}^{\infty} c_{2r+1}(a) T_{2r+1}(x) \quad x \in [-1,1], \tag{3.59}$$

where the coefficients c_r can be computed using the orthogonality of the Chebyshev basis

$$c_{2r}(a) = \frac{2}{\pi} \int_{-1}^{1} \frac{\cos(ax) T_{2r}(x)}{\sqrt{1-x^2}} dx = (-1)^r 2 J_{2r}(a) \tag{3.60}$$

$$c_{2r+1}(a) = \frac{2}{\pi} \int_{-1}^{1} \frac{\sin(ax) T_{2r+1}(x)}{\sqrt{1-x^2}} dx = (-1)^r 2 J_{2r+1}(a), \tag{3.61}$$

where $J_\alpha(x)$ is the Bessel function of the first kind. We can truncate the expansion to the first R terms and then express it in terms of the monomial basis

$$\cos(ax) \approx {\sum_{r=0}^{R}}' c_{2r}(a) T_{2r}(x) = \sum_{r=0}^{R} d_{2r}(a) x^{2r} \tag{3.62}$$

$$\sin(ax) \approx \sum_{r=0}^{R} c_{2r+1}(a) T_{2r+1}(x) = \sum_{r=0}^{R} d_{2r+1}(a) x^{2r+1} \tag{3.63}$$

In our case we have $ax = \pi n \Delta\theta'_k$ and we have that a must not depend on n and $\Delta\theta'_k$ to be able to apply the same procedure used previously, while x must cover all the interval $[-1, 1]$ to maximise the quality of the approximation. Therefore we have

$$\cos(\pi n \Delta\theta'_k) \approx \sum_{r=0}^{R} d_{2r}\left(\pi(N-1)\Theta'\right) \left(\frac{n}{N-1}\frac{\Delta\theta'}{\Theta'}\right)^{2r} \tag{3.64}$$

$$\sin(\pi n \Delta\theta'_k) \approx \sum_{r=0}^{R} d_{2+1r}\left(\pi(N-1)\Theta'\right) \left(\frac{n}{N-1}\frac{\Delta\theta'}{\Theta'}\right)^{2r+1}, \tag{3.65}$$

$$\text{where} \quad \frac{n}{N-1}\frac{\Delta\theta'}{\Theta'} \in [-1, 1] \tag{3.66}$$

and following the same steps used above, but using this new approximation, we finally get

$$\mathcal{NC}(x, \theta_k) = \sum_{s=0}^{S-1} Y^s_{j'_k} (\Delta\theta'_k/\Theta')^s, \quad \text{where} \tag{3.67}$$

$$Y^0, Y^1, \ldots = \mathcal{C}(X^0), \mathcal{S}'(X^1), \mathcal{C}(X^2), \mathcal{S}'(X^3), \ldots \tag{3.68}$$

$$X^s_n = d_s\left(\pi(N-1)\Theta'\right) \left(\frac{n}{N-1}\right)^s x_n \qquad n = 0, \ldots, N-1. \tag{3.69}$$

Using the inequality [21, Eq. 10.14.4] for the Bessel function, the truncation error of the Chebyshev series can be estimated as

$$\sum_{s=S}^{\infty} \left|c_s\left(\pi(N-1)\Theta'\right)\right| = 2\sum_{s=S}^{\infty} \left|J_s\left(\pi(N-1)\Theta'\right)\right| \leq 2\sum_{s=S}^{\infty} \frac{\left(\frac{\pi}{2}(N-1)\Theta'\right)^s}{s!} \tag{3.70}$$

$$= 2\frac{\left(\frac{\pi}{2}(N-1)\Theta'\right)^S}{S!} \sum_{s=0}^{\infty} \frac{\left(\frac{\pi}{2}(N-1)\Theta'\right)^s}{(S+1)^{(s)}} \leq 2\frac{\left(\frac{\pi}{2}(N-1)\Theta'\right)^S}{S!} \sum_{s=0}^{\infty} \left(\frac{\pi}{2}(N-1)\Theta' \frac{1}{S+1}\right)^s \tag{3.71}$$

$$\leq 2\frac{\left(\frac{\pi}{2}(N-1)\Theta'\right)^S}{S!} \sum_{s=0}^{\infty} \left(\frac{1}{S+1}\right)^s = 2\frac{S+1}{S}\frac{\left(\frac{\pi}{2}(N-1)\Theta'\right)^S}{S!} \leq 2\frac{S+1}{S}\frac{\left(\frac{\pi}{4}\frac{N-1}{N'-1}\right)^S}{S!}, \tag{3.72}$$

where $m^{(i)}$ is the rising factorial. Comparing (3.57) and (3.72) we can see that using Chebyshev approximation basically corresponds to adding a factor 2 of oversampling in the Taylor approximation. Therefore if we want to get an error smaller than 10^{-16} using Chebyshev approximation we must choose $S = 17$ with no oversampling ($N' = N$) or $S = 10$ if we choose $N' = 8N$.

3.2 Integral of a weighted polynomial

In section 3.1 we have seen how to find the representation in the Chebyshev bases of first and second kind of the polynomial interpolant on the set of extrema Chebyshev nodes. Here we will see how to integrate a polynomial $p_n(x)$ of degree n multiplied by the Jacobi weight function

$$F(x) = \int p_n(x)w(x)dx \quad \text{where} \quad w(x) = (1-x)^\alpha(1+x)^\beta. \tag{3.73}$$

We will compute $F(x)$ using the propeties of Chebyshev polynomials, in particular the recurrence relations

$$T_{n+1} = 2xT_n - T_{n-1} \tag{3.74}$$

$$U_{n+1} = 2xU_n - U_{n-1}, \tag{3.75}$$

the differentiation of Chebyshev polynomials of first kind

$$\frac{dT_n}{dx}(x) = nU_{n-1}(x) \tag{3.76}$$

and the relations between Chebyshev polynomials of first and second kind

$$T_n(x) = 1/2(U_n(x) - U_{n-2}(x)) \tag{3.77}$$

$$U_n = 2\sum_{\text{odd}j}^{n} T_j(x) \quad \text{for odd n} \tag{3.78}$$

$$U_n = 2\sum_{\text{even}j}^{n} T_j(x) - 1 \quad \text{for even n.} \tag{3.79}$$

This kind of approach has been used several times in the literature for the computation of definite integrals of weighted polynomials, see for example [62].

Asymptotics on one side We want to compute the integral

$$F(x) = \int p_n(x)w(x)dx, \quad \text{where} \quad w(x) = (1-x)^\alpha. \tag{3.80}$$

It is possible to express $F(x)$ as

$$F(x) = P_n(x)(1-x)^{\alpha+1} + c, \tag{3.81}$$

where $P_n(x)$ is a polynomial of degree n and c is an arbitrary constant. Then if we differentiate we get

$$F'(x) = \frac{dP_n(x)}{dx}(1-x)^{\alpha+1} - P_n(x)(\alpha+1)(1-x)^\alpha \tag{3.82}$$

$$= \left(\frac{dP_n(x)}{dx}(1-x) - P_n(x)(\alpha+1)\right)w(x) \tag{3.83}$$

$$= q_n(x)w(x), \tag{3.84}$$

where $q_n(x)$ is a polynomial of degree n. Therefore we need to find $P_n(x)$ such that $q_n(x) = p_n(x)$. We start writing $P_n(x)$ in terms of the Chebyshev basis of first kind and $p_n(x)$ in terms of the Chebyshev basis of second kind

$$P_n(x) = \sum_{i=0}^{n} a_i T_i(x) \quad \text{and} \quad p_n(x) = \sum_{i=0}^{n} b_i U_i(x). \tag{3.85}$$

Using the properties of Chebyshev polynomials listed above, it can be seen that

$$\frac{d(T_i(x)(1-x)^{\alpha+1})}{dx} = \left(-\frac{\alpha+1+i}{2} U_i(x) + iU_{i-1}(x) + \frac{\alpha+1-i}{2} U_{i-2}(x) \right) w(x), \quad i \geq 2 \tag{3.86}$$

$$\frac{d(T_1(x)(1-x)^{\alpha+1})}{dx} = \left(-\frac{\alpha+2}{2} U_1(x) + U_0(x) \right) w(x) \tag{3.87}$$

$$\frac{d(T_0(x)(1-x)^{\alpha+1})}{dx} = -(\alpha+1) U_0(x) w(x). \tag{3.88}$$

This means that we have a relation between a_i and b_i in the form of a linear system

$$\begin{pmatrix} d_{1,0} & d_{2,0} & d_{3,0} & & \\ & \ddots & \ddots & \ddots & \\ & & d_{1,n-2} & d_{2,n-2} & d_{3,n-2} \\ & & & d_{1,n-1} & d_{2,n-1} \\ & & & & d_{1,n} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}, \tag{3.89}$$

that has an upper triangular band matrix and can be solved in $O(n)$ operation using back substitution. This method of solution is always stable when $\alpha > -1$.

When $\alpha > -1$ we also have that the integral

$$\int_1^x p_n(x) w(x) dx = P_n(x)(1-x)^{\alpha+1} \tag{3.90}$$

is defined and can be computed without loss of accuracy even when x is very close to 1.

Asymptotics on both sides We want to compute the integral

$$F(x) = \int p_n(x) w(x) dx, \quad \text{where} \quad w(x) = (1-x)^{\alpha}(1+x)^{\beta}. \tag{3.91}$$

It is possible to express $F(x)$ as

$$F(x) = P_{n-1}(x)(1-x)^{\alpha+1}(1+x)^{\beta+1} + a_0 g(x), \tag{3.92}$$

where

$$g(x) = \int T_0(x) w(x) dx = \int w(x) dx \tag{3.93}$$

and $P_{n-1}(x)$ is a polynomial of degree $n-1$. Then if we differentiate we get

$$F'(x) = \frac{dP_{n-1}(x)}{dx}(1-x)^{\alpha+1}(1+x)^{\beta+1} - \tag{3.94}$$

$$P_{n-1}(x)\left((\alpha+1)(1-x)^{\alpha}(1+x)^{\beta+1} - (\beta+1)(1-x)^{\alpha+1}(1+x)^{\beta}\right) + a_0 g'(x) \tag{3.95}$$

$$= \left(\frac{dP_{n-1}(x)}{dx}(1-x)(1+x) - P_{n-1}(x)((\alpha+1)(1+x) - (\beta+1)(1-x)) + a_0\right) w(x) \tag{3.96}$$

$$=q_n(x)w(x), \tag{3.97}$$

where $q_n(x)$ is a polynomial of degree n. Therefore we need to find $P_{n-1}(x)$ and a_0 such that $q_n(x) = p_n(x)$. We start writing $P_{n-1}(x)$ and $p_n(x)$ in terms of the Chebyshev basis of first kind

$$P_{n-1}(x) = \sum_{i=1}^{n} a_i T_{i-1}(x) \quad \text{and} \quad p_n(x) = \sum_{i=0}^{n} b_i T_i(x). \tag{3.98}$$

Using the properties listed above it can be seen that

$$\frac{d(T_i(x)(1-x)^{\alpha+1}(1+x)^{\beta+1})}{dx} = \tag{3.99}$$

$$\left(-\frac{\alpha+\beta+2+i}{2}T_{i+1}(x) - (\alpha-\beta)T_i(x) + \frac{\alpha+\beta+2-i}{2}T_{i-1}(x)\right) w(x), \quad i \geq 2 \tag{3.100}$$

$$\frac{d(T_1(x)(1-x)^{\alpha+1}(1+x)^{\beta+1})}{dx} = \left(-\frac{\alpha+\beta+3}{2}T_2(x) - (\alpha-\beta)T_1(x) - \frac{\alpha+\beta+1}{2}T_0(x)\right) w(x) \tag{3.101}$$

$$\frac{d(T_0(x)(1-x)^{\alpha+1}(1+x)^{\beta+1})}{dx} = \left(-(\alpha+\beta+2)T_1(x) - (\alpha-\beta)T_0(x)\right) w(x) \tag{3.102}$$

with in addition

$$\frac{dg(x)}{dx} = T_0(x)w(x) \tag{3.103}$$

This means that we have a relation between a_i and b_i in the form of a linear system

$$\begin{pmatrix} d_{1,0} & d_{2,0} & d_{3,0} & & \\ & \ddots & \ddots & \ddots & \\ & & d_{1,n-2} & d_{2,n-2} & d_{3,n-2} \\ & & & d_{1,n-1} & d_{2,n-1} \\ & & & & d_{1,n} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}, \tag{3.104}$$

that has an upper triangular band matrix and can be solved in $O(n)$ operation using back substitution. When $\alpha > -1$ and $\beta > -1$ the function $g(x)$ can be expressed in terms of the incomplete beta function as

$$g(x) = -2^{\alpha+\beta+1} B_{\frac{1-x}{2}}(\alpha+1, \beta+1) + c, \tag{3.105}$$

where c is an arbitrary constant. This method of solution is always stable when $\alpha > -1$ and $\beta > -1$. It can be also used to solve the case of the asymptotic on one side setting $\beta = 0$.

When $\alpha > -1$ and $\beta > -1$ we also have that the integral

$$\int_1^x p_n(x)w(x)dx = P_{n-1}(x)(1-x)^{\alpha+1}(1+x)^{\beta+1} - a_0 2^{\alpha+\beta+1} B_{\frac{1-x}{2}}(\alpha+1, \beta+1) \tag{3.106}$$

is defined and can be computed without loss of accuracy even when x is very close to 1.

3.3 Approximation

We have seen how to compute the Chebyshev interpolant in an efficient way and how to evaluate it at arbitrary points. However, if the function that we want to approximate is not smooth at the endpoints (see the beginning of section 3.1), Chebyshev interpolation is not that effective. Even worse, if the function is singular at the endpoints, Chebyshev approximation cannot be directly applied at all. To address the problem we can proceed as follows.

Reformulation

Instead of approximating the function directly, we will apply a change of variable to smooth the function at the boundary and then put the leading asymptotic term into a weight function.

Asymptotics on one side Let us consider a function f that is smooth inside the interval $[-1, 1]$, but has the following asymptotic behaviour on the right side

$$f(x) = a_0(1-x)^{\alpha} + a_1(1-x)^{\alpha+1} + a_2(1-x)^{\alpha+2} + \dots \quad x \to 1, \tag{3.107}$$

then we can define g as

$$g(x) = \frac{f(x)}{w(x)}, \quad \text{where} \tag{3.108}$$

$$w(x) = (1-x)^{\alpha}, \tag{3.109}$$

so $f(x) = g(x)w(x)$ and the asymptotic behaviour of g is

$$g(x) = a_0(1-x)^0 + a_1(1-x)^1 + a_2(1-x)^2 + \dots \quad x \to 1, \tag{3.110}$$

therefore g is C^{∞} also at the right endpoint. If instead f behaves like

$$f(x) = a_0(1-x)^{\alpha} + a_1(1-x)^{\alpha+1d} + a_2(1-x)^{\alpha+2d} + \dots \quad x \to 1, \tag{3.111}$$

then we can define g as

$$g(y) = \frac{f(s(y))}{w(y)}, \quad \text{where} \tag{3.112}$$

$$s(y) = 1 - 2\left(\frac{1-y}{2}\right)^{\frac{1}{d}} \tag{3.113}$$

$$w(y) = (1-y)^{\frac{\alpha}{d}}, \tag{3.114}$$

so $f(x) = g(s^{-1}(x))w(s^{-1}(x))$ and the asymptotic behaviour of g is

$$\tag{3.115}$$

$$g(y) = \hat{a}_0(1-y)^0 + \hat{a}_1(1-y)^1 + \hat{a}_2(1-y)^2 + \dots \quad y \to 1, \tag{3.116}$$

therefore g is C^{∞} also at the right endpoint. More in general if f behaves like

$$f(x) = a_0(1-x)^{\alpha_0} + a_1(1-x)^{\alpha_1} + \text{h.o.t.} \quad x \to 1 \tag{3.117}$$

and we proceed as above with

$$
\begin{aligned}
g(y) &= \frac{f(s(y))}{w(y)}, \quad \text{where} && (3.118)\\
s(y) &= 1 - 2\left(\frac{1-y}{2}\right)^{\frac{m}{\alpha_1-\alpha_0}} && (3.119)\\
w(y) &= (1-y)^{\frac{m\alpha_0}{\alpha_1-\alpha_0}}, && (3.120)
\end{aligned}
$$

then $f(x) = g(s^{-1}(x))w(s^{-1}(x))$ and the asymptotic behaviour of g is

$$
g(y) = \hat{a}_0(1-y)^0 + \hat{a}_1(1-y)^m + \text{h.o.t.} \quad y \to 1, \tag{3.121}
$$

therefore, if m is a positive integer, the function g is at least C^m at the right endpoint. Increasing m will make g smoother at the boundary but at the same time g will become worse inside the interval. The value of m can be chosen to get the best compromise depending on the problem.

Asymptotics on both sides Let us consider now a function f that has a bad asymptotic behaviour at both the endpoints of the interval

$$
\begin{aligned}
f(x) &= a_0(1-x)^{\alpha_0} + a_1(1-x)^{\alpha_1} + \text{h.o.t.} \quad x \to 1 && (3.122)\\
f(x) &= b_0(1+x)^{\beta_0} + b_1(1+x)^{\beta_1} + \text{h.o.t.} \quad x \to -1, && (3.123)
\end{aligned}
$$

then we can generalise the case seen above and define g as

$$
\begin{aligned}
g(y) &= \frac{f(s(y))}{w(y)}, \quad \text{where} && (3.124)\\
s(y) &= 1 - 2\frac{\int_1^y (1-t)^{\frac{m_a}{\alpha_1-\alpha_0}-1}(1+t)^{\frac{m_b}{\beta_1-\beta_0}-1}\,dt}{\int_1^{-1} (1-t)^{\frac{m_a}{\alpha_1-\alpha_0}-1}(1+t)^{\frac{m_b}{\beta_1-\beta_0}-1}\,dt} && (3.125)\\
&= 1 - 2I_{\frac{1-y}{2}}\left(\frac{m_a}{\alpha_1-\alpha_0}, \frac{m_b}{\beta_1-\beta_0}\right) && (3.126)\\
w(x) &= (1-y)^{\frac{m_a\alpha_0}{\alpha_1-\alpha_0}}(1+y)^{\frac{m_b\beta_0}{\beta_1-\beta_0}}, && (3.127)
\end{aligned}
$$

where $I_x(a,b)$ is the regularised incomplete beta function. So $f(x) = g(s^{-1}(x))w(s^{-1}(x))$ and the asymptotic behaviour of g is

$$
\begin{aligned}
g(y) &= \hat{a}_0(1-y)^0 + \hat{a}_1(1-y)^1 + \ldots + \hat{a}_{m_a}(1-y)^{m_a} + \text{h.o.t.} \quad y \to 1 && (3.128)\\
g(y) &= \hat{b}_0(1+y)^0 + \hat{b}_1(1+y)^1 + \ldots + \hat{b}_{m_a}(1+y)^{m_b} + \text{h.o.t.} \quad y \to -1. && (3.129)
\end{aligned}
$$

Therefore, if m_a, m_b are positive integers, the function g is at least C^{m_a} at the right and C^{m_b} at the left endpoint. Increasing m_a, m_b will make g smoother at the boundary but at the same time g will become worse inside the interval. It can be easily seen that if we take $\beta_0 = 0$, $\beta_1 = 1$, $m_b = 1$ we recover the case with the singularity only on one endpoint.

Approximation

Now that the function g is smoother, we can effectively interpolate it with a polynomial p_n at the Chebyshev nodes y_i. The interpolation nodes expressed in the original variable become $x_i = s(y_i)$, that means that, if $\alpha_1 - \alpha_0 < 1$ and $\beta_1 - \beta_0 < 1$, the nodes are pushed towards the endpoints. As we increase m_a and m_b the nodes get even more packed at the sides of the interval.
At the same time, when we approximate g with a polynomial, we are actually approximating f in the function space

$$f(x) \approx \tilde{f}(x) := p_n(s^{-1}(x))w(s^{-1}(x)), \tag{3.130}$$

a polynomial p_n composed with a smooth change of variables, multiplied by a weight function. The asymptotic behaviour of $\tilde{f}$ is

$$\tilde{f}(x) = \tilde{a}_0(1-x)^{\alpha_0} + \tilde{a}_1(1-x)^{\alpha_0+1\frac{\alpha_1-\alpha_0}{m_a}} + \tilde{a}_2(1-x)^{\alpha_0+2\frac{\alpha_1-\alpha_0}{m_a}} + \dots \quad x \to 1 \tag{3.131}$$

$$\tilde{f}(x) = \tilde{b}_0(1+x)^{\beta_0} + \tilde{b}_1(1+x)^{\beta_0+1\frac{\beta_1-\beta_0}{m_b}} + \tilde{b}_2(1+x)^{\beta_0+2\frac{\beta_1-\beta_0}{m_b}} + \dots \quad x \to -1 \tag{3.132}$$

and as we increase m_a and m_b the asymptotic terms get closer to each other, the (m_a+1)-th and (m_b+1)-th having exponent α_1 and β_1 respectively.
The final effect of the change of variable is therefore to put more nodes close to the interval endpoints and at the same time more asymptotic terms close to the leading ones, in the attempt of better capturing the behaviour of f at the endpoints.

Remarks

The use of the incomplete beta function as a smoothing transformation, can be found for example in [49], where it has been used to perform definite integrals. The authors considered, along with other changes of variable, the polynomial case of the beta transformation, that is when the coefficient a, b of $I_x(a,b)$ are positive integers. The use of the incomplete beta function with non integer powers generalises the polynomial case allowing to obtain better results. The beta function, as well as its inverse, is already implemented in most computational software therefore easy to implement and fast to compute.

3.4 Integration

We have treated the approximation case above, now we proceed in a similar way for indefinite integration and the evaluation of the integral operator.

Indefinite integration

Reformulation

As done above, we will apply a change of variables and then put the leading asymptotic term into a weight function. The difference this time is that if we apply a change of variables s to the integral we get

$$\int f(x)dx = \int f(s(y))s'(y)dy \tag{3.133}$$

and therefore also the derivative of s must be taken into account. We will discuss only the case with bad asymptotic behaviour on both sides, the one side case can be obtained taking $\beta_0 = 0$, $\beta_1 = 1$, $m_b = 1$.

Asymptotics on both sides If f is defined like in (3.122,3.123), then we can define g as

$$g(y) = \frac{f(s(y))s'(y)}{w(y)}, \quad \text{where} \tag{3.134}$$

$$s(y) = 1 - 2I_{\frac{1-y}{2}}\left(\frac{m_a}{\alpha_1 - \alpha_0}, \frac{m_b}{\beta_1 - \beta_0}\right) \tag{3.135}$$

$$s'(y) = \frac{\left(\frac{1-y}{2}\right)^{\frac{m_a}{\alpha_1-\alpha_0}-1}\left(\frac{1+y}{2}\right)^{\frac{m_b}{\beta_1-\beta_0}-1}}{B\left(\frac{m_a}{\alpha_1-\alpha_0}, \frac{m_b}{\beta_1-\beta_0}\right)} \tag{3.136}$$

$$w(y) = (1-y)^{\frac{m_a(\alpha_0+1)}{\alpha_1-\alpha_0}-1}(1+y)^{\frac{m_b(\beta_0+1)}{\beta_1-\beta_0}-1}, \tag{3.137}$$

so $\int f(x)dx = \int g(y)w(y)dy$ and the asymptotic behaviour of g is

$$g(y) = \hat{a}_0(1-y)^0 + \hat{a}_1(1-y)^1 + \ldots + \hat{a}_{m_a}(1-y)^{m_a} + \text{h.o.t.} \quad y \to 1 \tag{3.138}$$

$$g(y) = \hat{b}_0(1+y)^0 + \hat{b}_1(1+y)^1 + \ldots + \hat{b}_{m_b}(1+y)^{m_b} + \text{h.o.t.} \quad y \to -1. \tag{3.139}$$

Therefore, if m_a, m_b are positive integers, the function g is at least C^{m_a} at the right and C^{m_b} at the left endpoint.

Approximation

We can now effectively approximate g with a polynomial, obtaining

$$\int f(x)dx \approx \int p_n(y)w(y)dy, \tag{3.140}$$

that is the integral of a polynomial multiplied by the Jacobi weight function. As in the previous section the polynomial p_n can be obtained interpolating the function $g(y)$ at the Chebyshev nodes y_i, that in the original variable become $x_i = s(y_i)$. We are approximating f in the function space

$$f(x) \approx \tilde{f}(x) = \frac{p_n(s^{-1}(x))w(s^{-1}(x))}{s'(s^{-1}(x))}, \tag{3.141}$$

a polynomial p_n composed with a smooth change of variables, multiplied by a weight function. The asymptotic behaviour of $\tilde{f}$ is like in (3.131) and the choice of m_a and m_b has the same effect as previously discussed.

Integral operator

We want now to integrate over the interval $[-1, 1]$ a function f that has a bad asymptotic behaviour at both endpoints of the interval like in (3.122,3.123) and in addition is not smooth (or possibly weakly singular) in a point $x_0 \in [-1, 1]$. In particular we consider the case where

$$f(x) = a_0(1-x)^{\alpha_0} + a_1(1-x)^{\alpha_1} + \text{h.o.t.} \quad x \to 1 \tag{3.142}$$

$$f(x) = b_0(1+x)^{\beta_0} + b_1(1+x)^{\beta_1} + \text{h.o.t.} \quad x \to -1 \tag{3.143}$$

$$f(x) = c_0 + c_1(x - x_0)\log(|x - x_0|) + \text{h.o.t.} \quad x \to x_0, \tag{3.144}$$

but a similar procedure can be used also in different situations.

Reformulation

We start working at the endpoints with the same smoothing transformation used above, and we define the function g as

$$g(y) = f(s(y)), \quad \text{where} \tag{3.145}$$

$$s(y) = 1 - 2I_{\frac{1-y}{2}}\left(\frac{m_a}{\alpha_1-\alpha_0}, \frac{m_b}{\beta_1-\beta_0}\right), \tag{3.146}$$

that has the following asymptotic behaviour

$$g(y) = \hat{a}_0(1-y)^{\frac{m_a\alpha_0}{\alpha_1-\alpha_0}} + \hat{a}_1(1-y)^{\frac{m_a\alpha_0}{\alpha_1-\alpha_0}+1} + \ldots + \hat{a}_{m_a}(1-y)^{\frac{m_a\alpha_0}{\alpha_1-\alpha_0}+m_a} + \text{h.o.t.} \quad y \to 1 \tag{3.147}$$

$$g(y) = \hat{b}_0(1+y)^{\frac{m_b\beta_0}{\beta_1-\beta_0}} + \hat{b}_1(1+y)^{\frac{m_b\beta_0}{\beta_1-\beta_0}+1} + \ldots + \hat{b}_{m_b}(1+y)^{\frac{m_b\beta_0}{\beta_1-\beta_0}+m_b} + \text{h.o.t.} \quad y \to -1 \tag{3.148}$$

$$g(y) = c_0 + \hat{c}_1(y-y_0)\log(|y-y_0|) + \text{h.o.t.} \quad y \to y_0 = s^{-1}(x_0). \tag{3.149}$$

Now we work on the point y_0 with another smoothing transformation h, and we define the function h as

$$h(z) = g(t(z)), \quad \text{where} \tag{3.150}$$

$$t(z) = y_0 + \left((1-y_0)^{\frac{1}{m_c}}\frac{(1+z)}{2} - (1+y_0)^{\frac{1}{m_c}}\frac{(1-z)}{2}\right)^{m_c} = \tag{3.151}$$

$$= y_0 + \left(\frac{(1+y_0)^{\frac{1}{m_c}} + (1-y_0)^{\frac{1}{m_c}}}{2}\right)^{m_c}(z-z_0)^{m_c} \tag{3.152}$$

$$z_0 = t^{-1}(y_0) = \frac{(1+y_0)^{\frac{1}{m_c}} - (1-y_0)^{\frac{1}{m_c}}}{(1+y_0)^{\frac{1}{m_c}} + (1-y_0)^{\frac{1}{m_c}}}, \tag{3.153}$$

where m_c is an odd positive integer. Note that being m_c an integer the transformation is polynomial. Note also that it is not possible to define a suitable smooth transformation when m_c is even. The asymptotic behaviour of h is

$$h(z) = \bar{a}_0(1-z)^{\frac{m_a\alpha_0}{\alpha_1-\alpha_0}} + \bar{a}_1(1-z)^{\frac{m_a\alpha_0}{\alpha_1-\alpha_0}+1} + \ldots + \bar{a}_{m_a}(1-z)^{\frac{m_a\alpha_0}{\alpha_1-\alpha_0}+m_a} + \text{h.o.t.} \quad z \to 1 \tag{3.154}$$

$$h(z) = \bar{b}_0(1+z)^{\frac{m_b\beta_0}{\beta_1-\beta_0}} + \bar{b}_1(1+z)^{\frac{m_b\beta_0}{\beta_1-\beta_0}+1} + \ldots + \bar{b}_{m_b}(1+z)^{\frac{m_b\beta_0}{\beta_1-\beta_0}+m_b} + \text{h.o.t.} \quad z \to -1 \tag{3.155}$$

$$h(z) = c_0 + \bar{c}_{m_c}(z-z_0)^{m_c}\log(|z-z_0|) + \text{h.o.t.} \quad z \to z_0 = t^{-1}(s^{-1}(x_0)). \tag{3.156}$$

Finally we consider the integral

$$\int_{-1}^{1} f(x)dx = \int_{-1}^{1} g(y)s'(y)dy = \int_{-1}^{1} h(z)s'(t(z))t'(z)dz = \int_{-1}^{1} k(z)w(z)dz, \tag{3.157}$$

where k and w are

$$k(z) = \frac{h(z)s'(t(z))t'(z)}{w(z)} \tag{3.158}$$

$$w(z) = (1-z)^{\frac{m_a(\alpha_0+1)}{\alpha_1-\alpha_0}-1}(1+z)^{\frac{m_b(\beta_0+1)}{\beta_1-\beta_0}-1}. \tag{3.159}$$

The asymptotic behaviour of k is

$$k(z) = \hat{a}_0(1-z)^0 + \hat{a}_1(1-z)^1 + \ldots + \hat{a}_{m_a}(1-z)^{m_a} + \text{h.o.t.} \quad z \to 1 \tag{3.160}$$

$$k(z) = \hat{b}_0(1+z)^0 + \hat{b}_1(1+z)^1 + \ldots + \hat{b}_{m_b}(1+z)^{m_b} + \text{h.o.t.} \quad z \to -1 \tag{3.161}$$

$$k(z) = \hat{c}_0(z-z_0)^{m_c-1} + \hat{c}_1(z-z_0)^{m_c} + \ldots + \hat{c}_{m_c}(z-z_0)^{2m_c-1}\log(|z-z_0|) + \text{h.o.t.} \quad z \to z_0 \tag{3.162}$$

and with a suitable choice of m_a, m_b positive integers and m_c odd positive integer, the function k can be made smooth, at least C^{m_a} at the right and C^{m_b} at the left endpoint, and C^{2m_c-2} at z_0.

Approximation

We can now effectively approximate k with a polynomial, obtaining

$$\int_{-1}^{1} f(x)dx \approx \int_{-1}^{1} p_n(z)w(z)dz, \tag{3.163}$$

the integral of a polynomial multiplied by the Jacobi weight function. Again the polynomial p_n can be obtained interpolating the function $k(z)$ at the Chebyshev nodes z_i, that in the original variable become $x_i = s(t(z_i))$. We are approximating f in the function space

$$f(x) \approx \tilde{f}(x) := \frac{p_n(t^{-1}(s^{-1}(x)))w(t^{-1}(s^{-1}(x)))}{s'(s^{-1}(x))t'(t^{-1}(s^{-1}(x)))}, \tag{3.164}$$

a polynomial p_n composed with a smooth change of variables, multiplied by a weight function. The asymptotic behaviour of $\tilde{f}$ at the boundary is like in (3.131), while at x_0 we have

$$\tilde{f}(x) = \tilde{c}_0(x-x_0)^{-1+\frac{1}{m_c}} + \tilde{c}_1(x-x_0)^{-1+\frac{2}{m_c}} + \tilde{c}_2(x-x_0)^{-1+\frac{3}{m_c}} + \ldots \quad x \to x_0 \tag{3.165}$$

The choice of m_a and m_b has the same effect as previously discussed. In a similar way, increasing m_c puts more nodes close to x_0 and more asymptotic terms with exponent smaller than 1, with the $2m_c$-th having exponent 1;

Remarks

The use of the polynomial transformation (3.152) can be found for example in [50] to evaluate an integral operator, and other similar transformation can be found in literature.
We would also like to point out that in our case the most natural choice for a transformation that does simultaneously the job of s and t, would be the transformation u defined as

$$u(z) = 1 - 2\frac{\int_1^z (1-t)^{\frac{m_a}{\alpha_1-\alpha_0}-1}(t-z_0)^{m_c-1}(1+t)^{\frac{m_b}{\beta_1-\beta_0}-1}dt}{\int_1^{-1} (1-t)^{\frac{m_a}{\alpha_1-\alpha_0}-1}(t-z_0)^{m_c-1}(1+t)^{\frac{m_b}{\beta_1-\beta_0}-1}dt}, \tag{3.166}$$

$$\text{where} \quad u(z_0) = x_0. \tag{3.167}$$

Indeed by definition $u(1) = 1$, $u(-1) = -1$, $u'(z) > 0$ ($m_c - 1$ is an even positive integer) and u behaves like s (3.126) at the boundaries and like t (3.152) in z_0. However for simplicity of implementation we chose to use s and t combined, the main problem of u being the computation of z_0.

3.5 Numerical remarks

Numerical problems can arise in the implementation of the formulas defined in the previous sections. Indeed we have seen that applying the changes of variables (3.126,3.135,3.146) we cluster the nodes at the interval endpoints to mitigate the bad asymptotic behaviour of the functions that we want to approximate. But as the nodes get closer to the endpoints, the potential loss of information due to finite precision arithmetic increases. For example consider the function

$$f(x) = 1 + (1-x)^{\frac{1}{6}} \tag{3.168}$$

and the change of variable s to make it regular

$$s(y) = 1 - 2\left(\frac{1-y}{2}\right)^6 . \tag{3.169}$$

Indeed applying s to f we obtain the function g

$$g(y) := f(s(y)) = 1 + 2^{-\frac{5}{6}}(1-y), \tag{3.170}$$

that is C^∞. Let us take now 25 Chebyshev nodes

$$y_j = -\cos\left(\frac{\pi j}{24}\right), \quad j = 0, \dots, 24, \tag{3.171}$$

if we consider the last but one node y_{23} and we compute it using double precision arithmetic, then we get $y_{23} \approx 0.992114701314478$ and after the change of variable $x_{23} \approx 0.999999999999993$. Now if we want to evaluate the function $f(x)$ on this node and we do it using (3.168) we get $f(x_{23}) \approx 1.004429144579495$, while instead if we use (3.170) we have $g(y_{23}) \approx 1.004425474257041$. With the first formulation we have lost 10 digits! This is first due to cancellation error when we compute $1-x$, and then further amplified when raising to the power $\frac{1}{6}$ because the derivative of $z^{\frac{1}{6}}$ goes to infinity when z goes to 0. The situation gets worse as the number of Chebyshev nodes increases and as the exponent of the change of variables gets bigger (in practical cases it can be greater than 10).

To overcome this problem, instead of using the variables x and y, we will work with the couple of variables $y_1 := 1-y$ and $y_{-1} := 1+y$ and the corresponding $x_1 := 1-x$ and $x_{-1} := 1+x$. Where necessary, we will reformulate the expressions in a way to avoid numerical error close to the interval endpoints and to do so we will tend to use x_1 and y_1 when x and y are close to 1 and x_{-1} and y_{-1} when x and y are close to -1. Therefore the Chebyshev nodes can be expressed as

$$y_{1,j} = 1 - y_j = 2\sin\left(\frac{\pi j}{2(n-1)}\right)^2, \quad j = 0, \dots, n-1 \tag{3.172}$$

$$y_{-1,j} = 1 + y_j = 2\sin\left(\frac{\pi(n-1-j)}{2(n-1)}\right)^2, \quad j = 0, \dots, n-1, \tag{3.173}$$

the transformation $x = s(y)$ in (3.126,3.135,3.146) can be expressed as

$$x_1 = 1 - s(y) = 2I_{\frac{y_1}{2}}\left(\frac{m_a}{\alpha_1 - \alpha_0}, \frac{m_b}{\beta_1 - \beta_0}\right) \tag{3.174}$$

$$x_{-1} = 1 + s(y) = 2 - 2I_{\frac{1-y}{2}}\left(\frac{m_a}{\alpha_1 - \alpha_0}, \frac{m_b}{\beta_1 - \beta_0}\right) = 2I_{\frac{y_{-1}}{2}}\left(\frac{m_b}{\beta_1 - \beta_0}, \frac{m_a}{\alpha_1 - \alpha_0}\right), \tag{3.175}$$

the weight function $w(x)$ in (3.127) can be expressed as

$$w(x) = (1-x)^{\alpha_0}(1+x)^{\beta_0} = x_1^{\alpha_0} x_{-1}^{\beta_0} \tag{3.176}$$

and similarly for $w(y)$ in (3.137) and $w(z)$ in (3.151). The transformation $y = t(z)$ in (3.152), that is a polynomial because m_c is an odd positive integer, can be reformulated as

$$y_1 = 1 - t(z) = -\sum_{n=1}^{m_c} \binom{m_c}{n} y_{0,1}^{\frac{m_c-n}{m_c}} \left(-\frac{\left(y_{0,1}^{\frac{1}{m_c}} + y_{0,-1}^{\frac{1}{m_c}} \right) z_1}{2} \right)^n \tag{3.177}$$

$$y_{-1} = 1 + t(z) = -\sum_{n=1}^{m_c} \binom{m_c}{n} y_{0,-1}^{\frac{m_c-n}{m_c}} \left(-\frac{\left(y_{0,1}^{\frac{1}{m_c}} + y_{0,-1}^{\frac{1}{m_c}} \right) z_{-1}}{2} \right)^n \tag{3.178}$$

and in a similar way also its derivative $t'(z)$. The function $g(x)$ in (3.105) can be expressed as

$$g(x) = -2^{\alpha+\beta+1} B_{\frac{x_1}{2}}(\alpha+1, \beta+1) + c. \tag{3.179}$$

In general it might be necessary to apply the same kind of procedure when we evaluate any function, if we see the possibility that some numerical error can arise. For example in the case seen above (3.168) if we write

$$f(x) = 1 + (1-x)^{\frac{1}{6}} = 1 + x_1^{\frac{1}{6}}, \tag{3.180}$$

then the accuracy of the computations is preserved.

3.6 Changing interval

In sections 3.3 and 3.4 we have described some numerical methods to approximate and integrate functions in $[-1,1]$, the standard interval for approximation theory. Our HF problem instead is defined in $[0,1]$, therefore to use the previous results it will be necessary to rescale the interval. This is done using the simple linear transformation $\bar{x} = (x+1)/2$, where in $x \in [-1,1]$ and $\bar{x} \in [0,1]$.

Approximation

Consider a function $\bar{f}(\bar{x})$ in $[0,1]$ that behaves like

$$\bar{f}(\bar{x}) = \bar{a}_0(1-\bar{x})^{\alpha_0} + \bar{a}_1(1-\bar{x})^{\alpha_1} + \text{h.o.t.} \quad \bar{x} \to 1 \tag{3.181}$$

$$\bar{f}(\bar{x}) = \bar{b}_0 \bar{x}^{\beta_0} + \bar{b}_1 \bar{x}^{\beta_1} + \text{h.o.t.} \quad \bar{x} \to 0, \tag{3.182}$$

then the function $f(x) := \bar{f}((x+1)/2)$ in $[-1,1]$ behaves like

$$f(x) = a_0(1-x)^{\alpha_0} + a_1(1-x)^{\alpha_1} + \text{h.o.t.} \quad x \to 1 \tag{3.183}$$

$$f(x) = b_0(1+x)^{\beta_0} + b_1(1+x)^{\beta_1} + \text{h.o.t.} \quad x \to -1, \tag{3.184}$$

where the asymptotic coefficients have changed but the exponents have remained the same. We can now apply the results from section 3.3 to approximate $f(x)$.

Integration

Consider the same function $\bar{f}$ and f as above, integrating it and passing to $[-1,1]$ we get that

$$\int \bar{f}(\bar{x})d\bar{x} = \frac{1}{2}\int f(x)dx \quad \text{and} \quad \int_0^x \bar{f}(\bar{x})d\bar{x} = \frac{1}{2}\int_{-1}^x f(x)dx. \tag{3.185}$$

We can now apply the results from section 3.4 to integrate $f(x)$.

Integral operator

Consider a function $\bar{f}(\bar{x})$ that behaves like

$$\bar{f}(\bar{x}) = \bar{a}_0(1-\bar{x})^{\alpha_0} + \bar{a}_1(1-\bar{x})^{\alpha_1} + \text{h.o.t.} \quad \bar{x} \to 1 \tag{3.186}$$

$$\bar{f}(\bar{x}) = \bar{b}_0\bar{x}^{\beta_0} + \bar{b}_1\bar{x}^{\beta_1} + \text{h.o.t.} \quad \bar{x} \to 0 \tag{3.187}$$

$$\bar{f}(\bar{x}) = c_0 + \bar{c}_1(\bar{x}-\bar{x}_0)\log(|\bar{x}-\bar{x}_0|) + \text{h.o.t.} \quad \bar{x} \to \bar{x}_0, \tag{3.188}$$

then the function $f(x) := \bar{f}((x+1)/2)$ in $[-1,1]$ behaves like

$$f(x) = a_0(1-x)^{\alpha_0} + a_1(1-x)^{\alpha_1} + \text{h.o.t.} \quad x \to 1 \tag{3.189}$$

$$f(x) = b_0(1+x)^{\beta_0} + b_1(1+x)^{\beta_1} + \text{h.o.t.} \quad x \to -1 \tag{3.190}$$

$$f(x) = c_0 + c_1(x-x_0)\log(|x-x_0|) + \text{h.o.t.} \quad x \to x_0, \tag{3.191}$$

where the asymptotic coefficients have changed but the exponents and logarithm have remained the same. Integrating it and passing to $[-1,1]$ we have that

$$\int_0^1 \bar{f}(\bar{x})d\bar{x} = \frac{1}{2}\int_{-1}^1 f(x)dx \tag{3.192}$$

and we can again apply the results from section 3.4 to integrate $f(x)$.

Remarks

To perform approximation and integration in sections 3.3 and 3.4, we needed to have the values of $f(x)$ on the nodes $x_i = s(y_i)$ in the case of approximation and integration and on the nodes $x_i = s(t(z_i)))$ in the case of the integral operator, where y_i and z_i are the Chebyshev nodes on $[-1,1]$. Therefore when passing to $[0,1]$ we will need the values of $\bar{f}(\bar{x})$ on the corresponding nodes $\bar{x}_i = (x_i+1)/2$.

We have also seen that in the interval $[-1,1]$ we needed to use the couple of variables $x_1 := 1-x$ and $x_{-1} := 1+x$ in place of x, to avoid numerical problems when nodes cluster at the endpoints of $[-1,1]$. Similarly, in the interval $[0,1]$, we will need to use $\bar{x}_0 := \bar{x}$ and $\bar{x}_1 := 1-\bar{x}$ for the same reason. When passing from one interval to the other we get the relations $\bar{x}_0 = x_{-1}/2$ and $\bar{x}_1 = x_1/2$

3.7 Computation of the kernel

In our implementation we need to have the kernels of the integral operators computed with a small relative error at any point. If we use the formulations provided above (2.86,2.90,2.141,2.145), the

absolute error can be kept small, but when the value of the kernel approaches zero, the relative error can get arbitrarily large because of numerical cancelation.
In the case of the KGD model, to get a small relative error, the kernel can be for example reformulated in terms of simple functions in some different ways that have different numerical behaviours. Each formulation works well for a specific set of points but not for all the domain. For most of the points, the relative error can be maintained small choosing the appropriate formulation, but for some other points it is necessary to use series expansions cancelling the leading asymptotic terms. The same approach can be attempted also for radial but it turns out to be more challenging.
We propose instead to proceed in a different way, where we express both KGD and radial kernels in terms of symmetric elliptic integrals, that can be directly computed using a fast and accurate algorithm.

Elliptic integrals

We recall here some definitions and some identities related to elliptic integrals.
Incomplete elliptic integrals of first and second kind:

$$F(\phi|m) = \int_0^\phi \frac{d\theta}{\sqrt{1-m\sin^2\theta}}, \qquad E(\phi|m) = \int_0^\phi \sqrt{1-m\sin^2\theta}d\theta \tag{3.193}$$

and associate incomplete elliptic integrals of first and second kind [23, 24]:

$$B(\phi|m) = \int_0^\phi \frac{\cos^2\theta d\theta}{\sqrt{1-m\sin^2\theta}}, \qquad D(\phi|m) = \int_0^\phi \frac{\sin^2\theta d\theta}{\sqrt{1-m\sin^2\theta}}, \tag{3.194}$$

that are related together by the following equations:

$$F(\phi|m) = B(\phi|m) + D(\phi|m), \qquad E(\phi|m) = B(\phi|m) + (1-m)D(\phi|m). \tag{3.195}$$

Incomplete elliptic integral of third kind:

$$\Pi(\phi, n|m) = \int_0^\phi \frac{d\theta}{(1-n\sin^2\theta)\sqrt{1-m\sin^2\theta}} \tag{3.196}$$

and associate incomplete elliptic integral of third kind [25]:

$$J(\phi, n|m) = \int_0^\phi \frac{\sin^2\theta d\theta}{(1-n\sin^2\theta)\sqrt{1-m\sin^2\theta}}, \tag{3.197}$$

that are related together by the following equations:

$$J(\phi, n|m) = \frac{\Pi(\phi, n|m) + F(\phi, m)}{n}. \tag{3.198}$$

Symmetric elliptic integrals [12]:

$$R_F(x,y,z) = \tfrac{1}{2}\int_0^\infty \frac{dt}{\sqrt{(t+x)(t+y)(t+z)}}, \qquad R_C(x,y) = R_F(x,y,y) \tag{3.199}$$

$$R_J(x,y,z,p) = \tfrac{3}{2}\int_0^\infty \frac{dt}{(t+p)\sqrt{(t+x)(t+y)(t+z)}}, \qquad R_D(x,y,z) = R_J(x,y,z,z). \tag{3.200}$$

The incomplete elliptic integrals can be expressed in terms of the symmetric elliptic integrals

$$F(\phi|m) = \sin\phi R_F\left(\cos^2\phi, 1 - m^2\sin^2\phi, 1\right) \tag{3.201}$$

$$D(\phi|m) = \tfrac{1}{3}\sin^3\phi R_D\left(\cos^2\phi, 1 - m^2\sin^2\phi, 1\right) \tag{3.202}$$

$$J(\phi, n|m) = \tfrac{1}{3}\sin^3\phi R_J\left(\cos^2\phi, 1 - m^2\sin^2\phi, 1, 1 - n\sin^2\phi\right) \tag{3.203}$$

and the expressions for $E(\phi|m), B(\phi|m), \Pi(\phi, n|m)$ can be derived by combining these equations with the equations above.

Symmetric elliptic integrals can be computed using the fast and accurate algorithm proposed by Carlson [12, 13, 14].

KGD

The kernel of the integral operator that relates the crack width to the pressure (2.86) can be expressed in different formulations (see Appendix A.2):

$$\mathcal{J}(x, s) = \frac{1}{2}\ln\left|\frac{\sqrt{1-x^2}+\sqrt{1-s^2}}{\sqrt{1-x^2}-\sqrt{1-s^2}}\right| \tag{3.204}$$

$$= \begin{cases} \operatorname{artanh}\left(\sqrt{\dfrac{1-x^2}{1-s^2}}\right), & s < x \\ \operatorname{artanh}\left(\sqrt{\dfrac{1-s^2}{1-x^2}}\right), & s > x \end{cases} \tag{3.205}$$

$$= \begin{cases} F\left(\arcsin\left(\sqrt{\dfrac{1-x^2}{1-s^2}}\right)\middle|1\right), & s < x \\ F\left(\arcsin\left(\sqrt{\dfrac{1-s^2}{1-x^2}}\right)\middle|1\right), & s > x \end{cases} \tag{3.206}$$

$$= \begin{cases} \sqrt{\dfrac{x(1-x^2)}{x^2-s^2}}R_C\left(\dfrac{x(1-s^2)}{x^2-s^2}, x\right), & s < x \\ \sqrt{\dfrac{s(1-s^2)}{s^2-x^2}}R_C\left(\dfrac{s(1-x^2)}{s^2-x^2}, s\right), & s > x. \end{cases} \tag{3.207}$$

Not all these formulations are numerically stable at all points of the domain, we now give an idea of the underlying difficulties.

The first formulation (3.204) has numerical problems when $x \approx 1$ or when $s \approx 1$, because in this case the argument in input to the logarithm is close to 1. The problem arises because computing $\ln(z)$ when z is close to 1 greatly amplifies the relative error. Indeed if we take $\bar{z}$ close to z close to 1 we have that

$$\left|\frac{\ln(\tilde{z})-\ln(z)}{\tilde{z}-z}\right| \approx |\ln'(1)| = 1 \quad \text{and} \quad \left|\frac{\ln(z)}{z}\right| \approx \left|\frac{\ln(1)}{1}\right| = 0 \tag{3.208}$$

and as a consequence

$$\left|\frac{\ln(\tilde{z})-\ln(z)}{\tilde{z}-z}\right| \gg \left|\frac{\ln(z)}{z}\right|. \tag{3.209}$$

Rearranging the terms we get

$$\left|\frac{\ln(\bar{z})-\ln(z)}{\ln(z)}\right| \gg \left|\frac{\bar{z}-z}{z}\right|, \tag{3.210}$$

that means that in this case the relative error of the logarithm is much greater than the error of the input.

The second formulation (3.205) has again numerical problems for a similar reason when $x \approx s$, that means that the argument in input to the inverse hyperbolic tangent is close to 1. Indeed if we take $\bar{z}$ close to z close to 1 we have that

$$\left|\frac{\operatorname{artanh}(\tilde{z})-\operatorname{artanh}(z)}{\tilde{z}-z}\right| \approx |\operatorname{artanh}'(z)| \sim \frac{1}{1-z} \quad \text{and} \quad \left|\frac{\operatorname{artanh}(z)}{z}\right| \sim \ln(1-z) \tag{3.211}$$

and as a consequence

$$\left|\frac{\operatorname{artanh}(\tilde{z})-\operatorname{artanh}(z)}{\tilde{z}-z}\right| \gg \left|\frac{\operatorname{artanh}(z)}{z}\right|. \tag{3.212}$$

Rearranging the terms we get

$$\left|\frac{\operatorname{artanh}(\tilde{z})-\operatorname{artanh}(z)}{\operatorname{artanh}(z)}\right| \gg \left|\frac{\tilde{z}-z}{z}\right|, \tag{3.213}$$

that means that in this case the relative error of the inverse hyperbolic tangent is much greater than the error of the input.

Again for the third formulation (3.206) there are similar numerical problems when $x \sim s$, that means that the argument in input to the elliptic integral tends to $\pi/2$. Indeed if we take $\bar{z}$ close to z close to $\pi/2$ we have that

$$\left|\frac{F(\tilde{z}|1)-F(z|1)}{\tilde{z}-z}\right| \approx \left|\frac{\partial F(z|1)}{\partial z}\right| \sim \frac{1}{\frac{\pi}{2}-z} \quad \text{and} \quad \left|\frac{F(z|1)}{z}\right| \sim \ln\left(\frac{\pi}{2}-z\right) \tag{3.214}$$

and as a consequence

$$\left|\frac{F(\tilde{z}|1)-F(z|1)}{\tilde{z}-z}\right| \gg \left|\frac{F(z|1)}{z}\right|. \tag{3.215}$$

Rearranging the terms we get

$$\left|\frac{F(\tilde{z}|1)-F(z|1)}{F(z|1)}\right| \gg \left|\frac{\tilde{z}-z}{z}\right|, \tag{3.216}$$

that means that in this case the relative error of the incomplete elliptic integral of the first kind is much greater than the error of the input.
The last formulation instead does not have numerical problems, provided that $1-s^2, 1-x^2, x^2-s^2, s^2-x^2$ are computed accurately. Note also that the last formulation has been obtained thanks to the homogeneity of the symmetric elliptic integrals [21, Eq. 19.20.1.2,19.20.6.2] and it has been chosen in a way that minimises to chance of overflow.

Similarly the kernel of the integral operator that relates the crack width to the derivative of the pressure (2.90) can be expressed as (see Appendix A.2):

$$\mathcal{K}(x,s) = \frac{x-s}{2}\ln\left|\frac{\sqrt{1-x^2}+\sqrt{1-s^2}}{\sqrt{1-x^2}-\sqrt{1-s^2}}\right| - \frac{x}{2}\ln\left(\frac{1+xs+\sqrt{1-x^2}\sqrt{1-s^2}}{1+xs-\sqrt{1-x^2}\sqrt{1-s^2}}\right) \tag{3.217}$$

$$= \begin{cases} x\,\operatorname{artanh}\left(\dfrac{s}{x}\sqrt{\dfrac{1-x^2}{1-s^2}}\right) - s\,\operatorname{artanh}\left(\sqrt{\dfrac{1-x^2}{1-s^2}}\right), & s<x \\[2ex] x\,\operatorname{artanh}\left(\dfrac{x}{s}\sqrt{\dfrac{1-s^2}{1-x^2}}\right) - s\,\operatorname{artanh}\left(\sqrt{\dfrac{1-s^2}{1-x^2}}\right), & s>x \end{cases} \tag{3.218}$$

$$= s\begin{cases} \dfrac{s^2-x^2}{x^2}J\left(\arcsin\left(\sqrt{\dfrac{1-x^2}{1-s^2}}\right), \dfrac{s^2}{x^2}\middle|1\right), & s<x \\[2ex] \dfrac{x^2-s^2}{s^2}\Pi\left(\arcsin\left(\sqrt{\dfrac{1-s^2}{1-x^2}}\right), \dfrac{x^2}{s^2}\middle|1\right), & s>x \end{cases} \tag{3.219}$$

$$= -\begin{cases} \dfrac{1}{3}\dfrac{s(1-x^2)^{\frac{3}{2}}}{\sqrt{x(x^2-s^2)}}R_J\left(\dfrac{x(1-s^2)}{x^2-s^2}, x, x, \dfrac{1}{x}\right), & s<x \\[2ex] \sqrt{\dfrac{(1-s^2)(s^2-x^2)}{s}}\times \\ \left[R_C\left(\dfrac{s(1-x^2)}{s^2-x^2}, s\right) + \dfrac{1}{3}\dfrac{x^2(1-s^2)}{s(s^2-x^2)}R_J\left(\dfrac{s(1-x^2)}{s^2-x^2}, s, s, \dfrac{1}{s}\right)\right], & s>x. \end{cases} \tag{3.220}$$

For similar reasons to the case discussed above, the first three formulations have numerical problems but the last one does not and is written in a form that is less sensitive to overflow, provided that $1-s^2, 1-x^2, x^2-s^2, s^2-x^2$ are computed accurately.

Radial

The kernel of the integral operator that relates the crack width to the pressure (2.141) can be expressed in different formulations:

$$\mathcal{J}(r,s) = \begin{cases} F\left(\arcsin\left(\sqrt{\frac{1-r^2}{1-s^2}}\right)\middle|\frac{s^2}{r^2}\right)\frac{s}{r}, & s<r \\ \\ F\left(\arcsin\left(\sqrt{\frac{1-s^2}{1-r^2}}\right)\middle|\frac{r^2}{s^2}\right), & s>r \end{cases} \tag{3.221}$$

$$= \begin{cases} \sqrt{\frac{r(1-r^2)}{r^2-s^2}}R_F\left(\frac{r(1-s^2)}{r^2-s^2}, r, \frac{1}{r}\right)\frac{s}{r}, & s<x \\ \\ \sqrt{\frac{s(1-s^2)}{s^2-r^2}}R_C\left(\frac{s(1-r^2)}{s^2-r^2}, s, \frac{1}{s}\right), & s>x. \end{cases} \tag{3.222}$$

As for KGD the last formulation is written in a form that is less sensitive to overflow and it is numerically stable provided that $1-s^2, 1-r^2, r^2-s^2, s^2-r^2$ are computed accurately.

Similarly the kernel of the integral operator that relates the crack width to the derivative of the pressure (2.145) can be expressed as (see Appendix A.2):

$$\mathcal{K}(r,s) = s\begin{cases} \left[E\left(\arcsin(s)\middle|\frac{r^2}{s^2}\right) - E\left(\arcsin\left(\frac{s}{r}\right)\middle|\frac{r^2}{s^2}\right)\right] = \left[B\left(\arcsin(r)\middle|\frac{s^2}{r^2}\right) - B\left(\frac{s^2}{r^2}\right)\right]\frac{s}{r}, & s<r \\ \\ \left[E\left(\arcsin(s)\middle|\frac{r^2}{s^2}\right) - E\left(\frac{r^2}{s^2}\right)\right], & s>r \end{cases} \tag{3.223}$$

$$= -s\begin{cases} \frac{r}{\sqrt{r^2-s^2}}D\left(\arcsin(\sqrt{1-r^2})\middle|\frac{s^2}{s^2-r^2}\right)\frac{s}{r}, & s<r \\ \\ \frac{\sqrt{s^2-r^2}}{s}E\left(\arcsin(\sqrt{1-s^2})\middle|\frac{r^2}{r^2-s^2}\right), & s>r \end{cases} \tag{3.224}$$

$$= -\begin{cases} \frac{1}{3}\frac{s(1-r^2)^{\frac{3}{2}}}{\sqrt{r(r^2-s^2)}}R_D\left(\frac{r(1-s^2)}{r^2-s^2}, r, \frac{1}{r}\right)\frac{s}{r}, & s<r \\ \\ \sqrt{\frac{(1-s^2)(s^2-r^2)}{s}}\times \\ \left[R_F\left(\frac{s(1-r^2)}{s^2-r^2}, s, \frac{1}{s}\right) + \frac{1}{3}\frac{r^2(1-s^2)}{s(s^s-r^2)}R_D\left(\frac{s(1-r^2)}{s^2-r^2}, s, \frac{1}{s}\right)\right], & s>r. \end{cases} \tag{3.225}$$

As for KGD the last formulation is written in a form that is less sensitive to overflow and it is numerically stable provided that $1-s^2, 1-r^2, r^2-s^2, s^2-r^2$ are computed accurately.

Chapter 4

Solver for the self-similar problem

We have seen in chapter 3 how to approximate and integrate functions with non smooth behaviour at the boundaries and possibly also inside the interval. Using these results we will see how to discretise and solve the self-similar problem. First we will describe an algorithm that follows the idea proposed in [56, 61, 80], that consists in the use of two nested fixed point iterations to solve the non linear system arising from the discretisation. Then we will describe another algorithm based instead on Newton's method. Finally we will verify our results against some benchmarks already present in literature and study the quality and convergence of our numerical solutions.

4.1 General formulation

The method we will use to solve all three PKN, KGD and radial models is essentially the same, therefore it is useful to state the equations of the problem in a general way. Then we will just have to describe the solver for the general formulation and it will be valid for all the models.

– **elasticity equation:**

$$w(x) = \mathcal{E}(\partial p/\partial x, L), \quad \text{where} \tag{4.1}$$

$$\mathcal{E}(\partial p/\partial x, L) = \begin{cases} \int_1^x \dfrac{\partial p(x)}{\partial x} dx & \text{if PKN} \\ L\int_0^1 \dfrac{\partial p(s)}{\partial s}\mathcal{K}(x,s)ds + K_{IC}\sqrt{L}\sqrt{1-x^2} & \text{if KGD, radial,} \end{cases} \tag{4.2}$$

the kernel of the integral operator depends on the model.

– **fracture propagation**

$$K_{IC} = \mathcal{T}(p, L), \quad \text{where} \tag{4.3}$$

$$\mathcal{T}(p, L) = \begin{cases} \sqrt{L}\int_0^1 \dfrac{p(s)}{\sqrt{1-s^2}} ds & \text{if KGD} \\ \sqrt{L}\int_0^1 \dfrac{p(s)s}{\sqrt{1-s^2}} ds & \text{if radial,} \end{cases} \tag{4.4}$$

it is only present for KGD and radial models.

– **continuity equation**

$$q(x) = \mathcal{C}(w, L), \quad \text{where} \tag{4.5}$$

$$\mathcal{C}(w, L) = \begin{cases} L\rho w(x)x + L(\gamma+\rho)\int_x^1 w(x)dx & \text{if PKN, KGD} \\ \left(L\rho w(x)x^2 + L(\gamma+2\rho)\int_x^1 w(x)xdx\right)/x & \text{if radial,} \end{cases} \tag{4.6}$$

this formulation can be derived integrating continuity equation in the differential form.
– **fluid balance equation**

$$L = \mathcal{B}(w, q_*), \quad \text{where} \tag{4.7}$$

$$\mathcal{B}(w, q_*) = \begin{cases} \dfrac{q_*}{(\gamma+\rho)\int_0^1 w(x)dx} & \text{if PKN, KGD} \\ \sqrt{\dfrac{q_*}{(\gamma+2\rho)\int_0^1 w(x)xdx}} & \text{if radial,} \end{cases} \tag{4.8}$$

this formulation can be derived evaluating (4.6) at the crack mouth and equating it to the boundary condition on the fluid input.
– **fluid flow equation**

$$\frac{\partial p}{\partial x} = \mathcal{F}(q, w, L), \quad \text{where} \tag{4.9}$$

$$\mathcal{F}(q, w, L) = -L\frac{q^n}{w^{2n+1}}, \tag{4.10}$$

the same for all models.
In the next section we will see how to discretise the problem and from now on, when we will refer to the operators just defined, we will mean their discretised version.

4.2 Discretisation

We will see here how we can use the methods described in chapter 3 to approximate and integrate the variables of the self-similar problem and to evaluate the integral operators. This depends on the asymptotic behaviour of the solution, that differs among the models, therefore we will pass through all of them one by one.

PKN

Approximation and integration

In the PKN model with no leak-off, we know from asymptotic analysis in section 2.2, that every function of the problem has the following asymptotic behaviour

$$f(x) = a_0(1-x)^{\alpha_0} + a_1(1-x)^{\alpha_1} + a_2(1-x)^{\alpha_2} + \dots \quad x \to 1, \tag{4.11}$$

$$\text{where} \quad \alpha_i = \alpha_0 + i \quad i = 1, 2, \dots \tag{4.12}$$

$$f(x) = b_0 x^{\beta_0} + b_1 x^{\beta_1} + b_2 x^{\beta_2} + \dots \quad x \to 0, \tag{4.13}$$

$$\text{where} \quad \beta_i = i \quad i = 0, 1, 2, \dots \tag{4.14}$$

and the coefficients a_i, b_i, α_0 depend on the function and on the input parameters.
We can then approximate or integrate every function of the problem proceeding as seen in section 3.6 and then back to 3.3 and 3.4. The process is based on the changes of variable s (3.126) and (3.135), which we can choose to be the same for all the functions. Indeed they depend on $\alpha_1 - \alpha_0 = \beta_1 - \beta_0 = 1$ that is the same for all the functions and on the parameters m_a, m_b, that can be taken both equal to 1. There is in fact no need to increase m_a, m_b because g in (3.124) and (3.134) is already C^∞. The interpolation nodes are also provided in section 3.6 and they depend on the aforementioned changes of variable s. Since the changes of variable are always the same for all the functions, the same is true also for the nodes.

KGD

In the KGD model with no leak-off, we know from asymptotic analysis in section 2.3, that the asymptotic behaviour of the functions at the crack tip is different when $K_{IC} = 0$ and when $K_{IC} > 0$. Indeed we recall that the asymptotic behaviour of the crack opening is:

$$\text{if } K_{IC} = 0, \quad w(x) = w_{1,0}(1-x)^{\frac{2}{n+2}} + \text{h.o.t.} \quad x \to 1, \tag{4.15}$$

$$\text{if } K_{IC} > 0, \quad w(x) = w_{1,0}(1-x)^{\frac{1}{2}} + w_{1,1}(1-x)^{\frac{3-n}{2}} \log^{[n=0]}(1-x) + \text{h.o.t.} \quad x \to 1. \tag{4.16}$$

As a consequence if one tried to directly approximate the variable as done for PKN model, it would be necessary to change discretisation in each case and in addition further problems would arise when K_{IC} is very small. Let us see more in detail.
In the case $K_{IC} > 0$ the leading asymptotic coefficient depends only on the toughness term of elasticity equation, indeed it can be seen from [61] that $w_{1,0} = K_{IC}\sqrt{2L}$ and that $w_{1,0}$ and $w_{1,1}$ are inversely proportional. Therefore when K_{IC} is close to 0 then $w_{1,0}$ is also close to 0 and as a consequence $w_{1,1}$ grows to infinity. This corresponds to the fact that the leading asymptotic term of the limiting case $K_{IC} = 0$ stays between the first and second term of the case $K_{IC} > 0$, that is $\frac{1}{2} < \frac{2}{n+2} < \frac{3-n}{2}$ when $0 \leq n < 2$. Indeed when K_{IC} is close to 0 the solution is trying to mimic the different asymptotic of the limiting case, the leading term goes to zero while the second term goes to infinity.
Similarly also the derivative of the pressure $\frac{\partial p}{\partial x}(x)$ has different asymptotic behaviour when $K_{IC} = 0$ and when $K_{IC} > 0$:

$$\text{if } K_{IC} = 0, \quad \frac{\partial p}{\partial x}(x) = dp_{1,0}(1-x)^{\frac{2}{n+2}-2} + \text{h.o.t.} \quad x \to 1, \tag{4.17}$$

$$\text{if } K_{IC} > 0, \quad \frac{\partial p}{\partial x}(x) = dp_{1,0}(1-x)^{\frac{3-n}{2}-2} + \text{h.o.t.} \quad x \to 1. \tag{4.18}$$

In the case $K_{IC} > 0$ we have from [61] that $dp_{1,0}$ is directly proportional to $w_{1,1}$, therefore when K_{IC} is close to 0 then $dp_{1,0}$ grows to infinity. This makes again impossible to use the approach of PKN model in the case of small toughness.
To address this problem we must keep into account at the same time the asymptotics of both cases $K_{IC} = 0$ and $K_{IC} > 0$. In this way we will be able to tackle both cases with the same approach and also to solve the problem of small toughness. To do so, for the crack opening $w(x)$ we will consider the following more general asymptotic behaviour

$$w(x) = w_{1,0}(1-x)^{\frac{1}{2}} + w_{1,1}(1-x)^{\frac{2}{n+2}} + \text{h.o.t.} \quad x \to 1 \tag{4.19}$$

and the same for the fluid flow $q(x)$, that has the same characteristics. The situation with the derivative of the pressure is instead different, because this time the leading term of the case $K_{IC} = 0$ is stronger than the leading term of $K_{IC} > 0$. Therefore it turns out that the best thing to do for $\frac{\partial p}{\partial x}(x)$, is to add a fictitious leading term and consider the more general asymptotic behaviour

$$\frac{\partial p}{\partial x}(x) = dp_{1,0}(1-x)^{-\frac{3}{2}} + dp_{1,1}(1-x)^{\frac{2}{n+2}-2} + \text{h.o.t.} \quad x \to 1, \tag{4.20}$$

where $dp_{1,0}$ is set to 0. One advantage of such choice is that in this case the distance between the first and second term is again $\frac{2}{n+2} - \frac{1}{2}$, the same as for $w(x)$ and $q(x)$ and as a consequence the change of variable s and the interpolation nodes are the same as for the other functions. Another advantage of adding the extra term $-\frac{3}{2}$ is that when we apply the procedure in section 3.6, then the functions g in (3.124) and (3.134) become more regular and therefore easier to approximate. On the other hand the drawback is that the approximation of $\frac{\partial p}{\partial x}(x)$ becomes inaccurate right close to the crack tip because we are not imposing the right asymptotic behaviour. Anyway this does not affect the overall quality of the solution of the self-similar problem on the rest of the interval. Finally the same idea can be applied to the pressure $p(x)$, for which we will consider the following asymptotic behaviour

$$p(x) = p_{1,0}(1-x)^{-\frac{1}{2}} + p_{1,1}(1-x)^{\frac{2}{n+2}-1} + \text{h.o.t.} \quad x \to 1, \tag{4.21}$$

where $p_{1,0}$ is set to 0. In the case of p we have the same advantages and drawbacks that we have seen for $\frac{\partial p}{\partial x}(x)$.

Passing to the crack mouth we note that also there the functions are not C^∞, indeed it can be seen from asymptotic analysis in section 2.3 that the third term of the crack opening $w(x)$ contains a logarithm

$$w(x) = w_{0,0} + w_{0,1}x + w_{0,2}x^2 \log(x) + \text{h.o.t.} \quad x \to 0, \tag{4.22}$$

where in this case $w_{0,1} = 0$. The presence of the logarithm is also propagated to the other functions and for this reason it will be useful to increase the regularity at 0 during approximation, with a suitable choice of the parameters in the changes of variable.

Approximation and integration

After the previous discussion, for every function of the KGD self-similar problem, we will consider the following asymptotic behaviour

$$f(x) = a_0(1-x)^{\alpha_0} + a_1(1-x)^{\alpha_1} + \text{h.o.t.} \quad x \to 1, \tag{4.23}$$

$$\text{where} \quad \alpha_1 = \alpha_0 + \frac{2}{n+2} - \frac{1}{2} \tag{4.24}$$

$$f(x) = b_0 x^{\beta_0} + b_1 x^{\beta_1} + \text{h.o.t.} \quad x \to 0, \tag{4.25}$$

$$\text{where} \quad \beta_0 = 0, \quad \beta_1 = 1 \tag{4.26}$$

and the coefficients a_i, b_i, α_0 depend on the function and on the input parameters. In particular we take $\alpha_0 = 1/2$ for $w(x)$ and $q(x)$, $\alpha_0 = -3/2$ for $\frac{\partial p}{\partial x}(x)$, $\alpha_0 = -1/2$ for $p(x)$ and $\alpha_0 = -1$ for $\frac{p(x)}{\sqrt{1-x^2}}$. Note that, as said previously, in the cases of $\frac{\partial p}{\partial x}(x)$, $p(x)$ and $\frac{p(x)}{\sqrt{1-x^2}}$ the term a_0 must be set to 0.

Now we can proceed as we have done for PKN and approximate or integrate every function of the problem proceeding as seen in section 3.6 and then back to 3.3 and 3.4. For all the functions we have that $\alpha_1 - \alpha_0 = \frac{2}{n+2} - \frac{1}{2}$, $\beta_1 - \beta_0 = 1$ and we can take the parameters m_a, m_b equal to 2, to increase the regularity of g (3.124) and (3.134) near 0 and 1. For this reason the changes of variable s (3.126) and (3.135) are the same for all the functions and as a consequence the interpolation nodes are the same as well.

Integral operator

Having seen how to approximate and integrate the functions, now we will see how to evaluate the integral operator $\int_0^1 \frac{\partial p}{\partial y}(y)\mathcal{K}(x,y)dy$. In this case we must also take into account the behaviour of the kernel $\mathcal{K}(x,y)$ (see section 2.3), the additional complication comes from the fact that the kernel is not regular also inside the interval.
Let us see now in detail what to do at the boundaries and inside the interval. When y tends to 1 we can consider $\frac{\partial p}{\partial y}(y)$ as in (4.20) and simply add the asymptotic behaviour of the kernel. When y tends to x we have that $\frac{\partial p}{\partial y}(y)$ is smooth and the irregularity comes only from the kernel $\mathcal{K}(x,y)$. When y tends to 0 we can use a stronger asymptotic approximation, that also provides more regularity. Therefore we can consider the asymptotic behaviour of $\frac{\partial p}{\partial y}(y)\mathcal{K}(x,y)$ as

$$f(x,y) = a_0(x)(1-y)^{\alpha_0} + a_1(x)(1-y)^{\alpha_1} + \text{h.o.t.} \quad y \to 1, \tag{4.27}$$

$$\text{where} \quad \alpha_0 = -1, \quad \alpha_1 = \frac{2}{n+2} - \frac{3}{2} \tag{4.28}$$

$$f(x,y) = b_0(x)y^{\beta_0} + b_1(x)y^{\beta_1} + \text{h.o.t.} \quad y \to 0, \tag{4.29}$$

$$\text{where} \quad \beta_0 = 0, \quad \beta_1 = 1 \tag{4.30}$$

$$f(x,y) = c_0 + c_1(x)(y-x)\log(|y-x|) + \text{h.o.t.} \quad y \to x \tag{4.31}$$

and the coefficients a_i, b_i, c_i depend on the input parameters. The terms a_0 and b_0 must be set to 0 and we can proceed as seen in section 3.6 and 3.4.
The changes of variable s (3.146) and t (3.152) are defined by $\alpha_1 - \alpha_0 = \frac{2}{n+2} - \frac{1}{2}$, $\beta_1 - \beta_0 = 1$ and by the parameters m_a, m_b, m_c that can be taken equal to 2, 2, 3 respectively, to increase the regularity of the function k in (3.158). However this time the interpolation nodes on y depend on the variable x, therefore if we want to evaluate the integral operator on the set of nodes x_i, we need to evaluate the derivative of the pressure and the kernel on the corresponding set of nodes $y_{i,j}$, that is a different set of nodes for every x_i.

We have seen in section 3.7 that to have an accurate computation of the kernel $\mathcal{K}(x,y)$ we need to have $1-y^2, 1-x^2, x^2-y^2, y^2-x^2$ accurately computed. If we use the variables y_0, y_1, x_0, x_1 we can get the values of $1-y^2 = y_1(1+y_0)$ and $1-x^2 = x_1(1+x_0)$ without introducing numerical error. About x^2-y^2, in our implementation we do not actually need to have it always computed with a small relative error. The reason is related to the approximation of $\frac{\partial p}{\partial y}(y)$ and $w(x)$, where we cluster the nodes at the boundaries to compensate the fact that the function changes more rapidly there. Where the nodes are closer, we must also be more precise in their computation, while where they are further apart the accuracy is less crucial. This means that when computing x^2-y^2 it is enough to keep the error small relatively to the values of $y, 1-y, x, 1-x$ and for this purpose we can use the formulation $x^2-y^2 = (x_0-y_0)(x_0+y_0)$ when y is close to 0 and $x^2-y^2 = (y_1-x_1)(x_0+y_0)$ when y is close to 1.

Radial

In the radial model with no leak-off, what has been already said for KGD about the asymptotic behaviour of the solution at the crack tip remains true and the same approach can be used. Therefore we will just focus on what happens at the crack mouth.

Approximation and integration

For every function of the radial self-similar problem, we will consider the following asymptotic behaviour at the crack mouth.

$$f(x) = b_0 x^{\beta_0} + b_1 x^{\beta_1} + \text{h.o.t.} \quad x \to 0, \tag{4.32}$$
$$\text{where} \quad \beta_1 = \beta_0 + \min(2-n, 1) \tag{4.33}$$

and the coefficients b_i, β_0 depend on the function and on the input parameters. In particular we take $\beta_0 = 0$ for $w(x)$, $\beta_0 = -1$ for $q(x)$, $\beta_0 = -2$ for $\frac{\partial p}{\partial x}(x)$, $\beta_0 = -1$ for $p(x)$ and $\beta_0 = -\frac{1}{2}$ for $\frac{p(x)x}{\sqrt{1-x^2}}$. We remark that in the case of $\frac{\partial p}{\partial x}(x)$, $p(x)$ and $\frac{p(x)x}{\sqrt{1-x^2}}$, the leading asymptotic term that we have chosen is stronger than the actual one, therefore in these cases b_0 must be set to 0, similarly to what we have seen previously for the crack tip.

We have made this choice because (see section 2.3) in the asymptotics of the pressure $p(x)$ a constant term and in some cases a logarithm appear. As a consequence the two leading terms can become very close to each other making impossible to use the methods described above. Instead, choosing the leading term in a suitable way, we can easily get a good approximation of $p(x)$ using the same approach and discretisation already used for the other functions. The drawback of this approach, similarly to what we have seen at the crack tip, is that in this way $\frac{\partial p}{\partial x}(x)$ and $p(x)$ are not very well approximated right close to the crack mouth. Nevertheless we underline that this doesn't affect the overall quality of the solution.

As we have done for PKN and KGD, we can now proceed as seen in section 3.6 and then back to 3.3 and 3.4. For all the functions we have that $\alpha_1 - \alpha_0 = \frac{2}{n+2} - \frac{1}{2}$, $\beta_1 - \beta_0 = \min(2-n, 1)$ and we can take the parameters m_a, m_b equal to 2 and 4 respectively, to increase the regularity of g (3.124) and (3.134) near 0 and 1. For this reason the changes of variable s (3.126) and (3.135) are the same for all the functions and as a consequence the interpolation nodes are the same as well.

Integral operator

For the evaluation of the integral operator $\int_0^1 \frac{\partial p}{\partial y}(y)\mathcal{K}(x,y)dy$, at the crack mouth we can simply add together the asymptotic behaviour of $\frac{\partial p}{\partial y}(y)$, as we have just chosen it above, and the asymptotic behaviour of the kernel. Therefore we can consider the asymptotic behaviour of $\frac{\partial p}{\partial y}(y)\mathcal{K}(x,y)$ as

$$f(x,y) = b_0(x) y^{\beta_0} + b_1(x) y^{\beta_1} + \text{h.o.t.} \quad y \to 0, \tag{4.34}$$
$$\text{where} \quad \beta_0 = 0, \quad \beta_1 = \min(2-n, 1) \tag{4.35}$$

and the coefficients b_i depend on the input parameters. The term b_0 must be set to 0 and we can

proceed as seen in section 3.6 and 3.4.
The changes of variable s (3.146) and t (3.152) are defined by $\alpha_1 - \alpha_0 = \frac{2}{n+2} - \frac{1}{2}$, $\beta_1 - \beta_0 = \min(2-n, 1)$ and by the parameters m_a, m_b, m_c that can be taken equal to 2, 4, 3 respectively, to increase the regularity of the function k in (3.158). Like for KGD, the set of interpolation nodes $y_{i,j}$ to evaluate the integral operator depends on the set of nodes x_i where we want to evaluate it.

What said above for KGD, about the accurate computation of the kernel $\mathcal{K}(x, y)$, holds for radial as well.

Implementation details

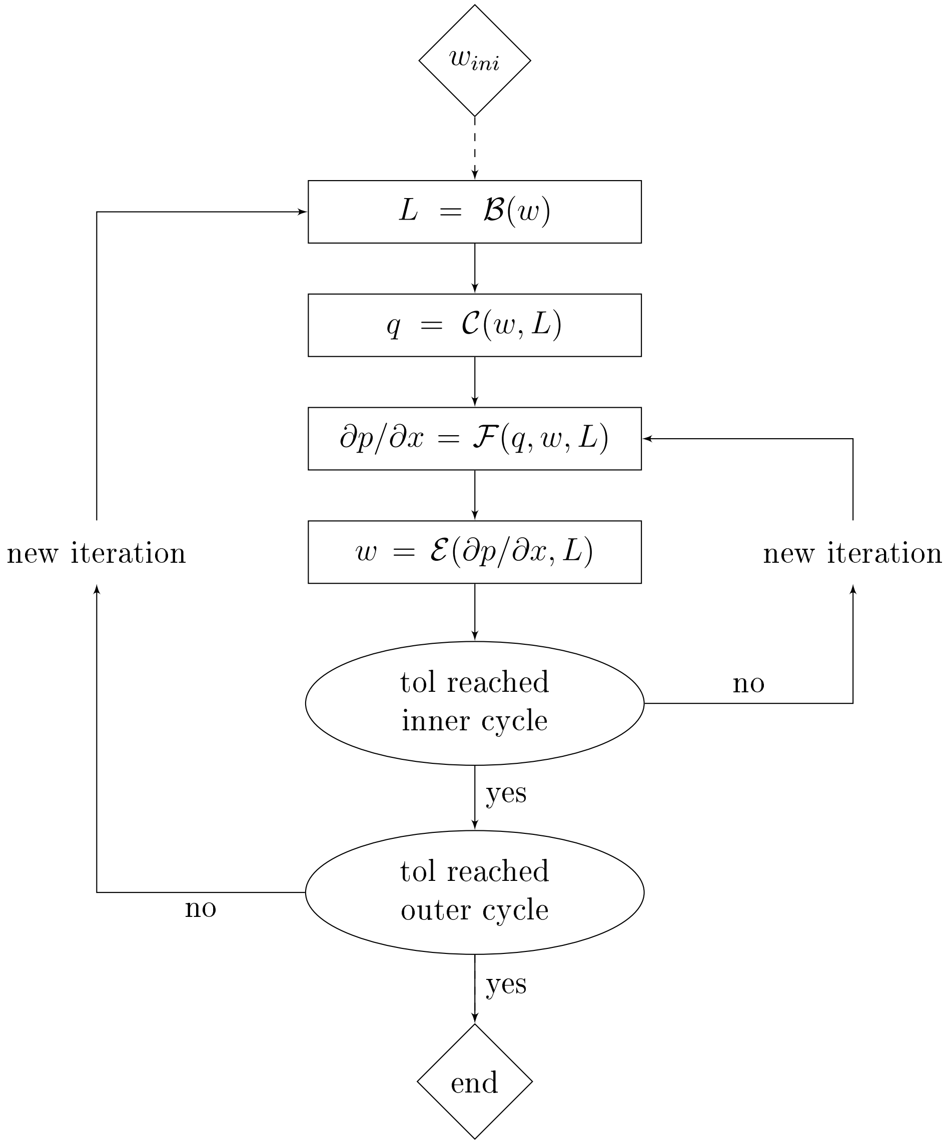
We have seen that for every model we have a set of nodes x_i where to approximate and integrate all the functions. Therefore for each function $f(x)$ we just need to have the values at the nodes $f_i := f(x_i)$. However to simplify the implementation it is better to work with the variables $f_i^{asy} := f(x_i)/((1-x_i)^{\alpha_0} x_i^{\beta_0})$. This is useful because in this way the leading asymptotic coefficients of $f(x)$ at 0 and 1 are already included in f_i^{asy} and therefore do not need to be treated separately. The same holds true for the integral operators where we have the set of nodes $y_{i,j}$ and we will work with the variables $f_{i,j}^{asy} := f(x_i, y_j)/((1-y_i)^{\alpha_0} y_i^{\beta_0})$.
We also point out that for simplicity we are writing all the formulas in terms of x, y, but as said in section 3.6, in the implementation of the solver we must represent the nodes using the variables $x_{0_i}, x_{1_i}, y_{0_i}, y_{1_i}$. The appropriate representation must be chosen each time to avoid numerical problems at the interval endpoints.

4.3 Fixed point solver

We describe here the fixed point solver, that is based on [56, 61, 80]. We start with an initial guess of the crack opening $w_{ini}(x)$ that has an asymptotic behaviour compatible with the problem, for example

$$w_{ini}(x) = \begin{cases} a(1-x)^{\frac{1}{n+2}} & \text{if PKN} \\ a(1-x)^{\frac{1}{2}} & \text{if KGD, radial.} \end{cases}, \tag{4.36}$$

where a depends on the input parameters. Then a solution is found through an iterative process made of two nested cycles in which the values of the variables are updated step by step as in (4.37).



(4.37)

The inner cycle is repeated until the error of the inner problem defined by fluid flow and elasticity equations goes under the tolerance. Similarly the outer cycle is repeated until the error of the solution of the self-similar problem goes under a desired tolerance.
To estimate the error we use the following criterion both for the inner and the outer cycle. We call w_i the value of w at the i-th iteration and define $\Delta w_i := w_i - w_{i-1}$. Because fixed point iteration has linear convergence, we have that the ratio $\frac{\|\Delta w_i\|_\infty}{\|\Delta w_{i-1}\|_\infty}$ will remain approximately constant for every i. Therefore the error at the i-th iteration can be approximated with a geometric series obtaining

$$err_{w_i} := \frac{\|w - w_i\|_\infty}{\|w\|_\infty} = \frac{\| \sum_{j=0}^{\infty} \Delta w_{i+j+1} \|_\infty}{\|w\|_\infty} \leq \frac{\sum_{j=0}^{\infty} \|\Delta w_{i+j+1}\|_\infty}{\|w\|_\infty} = \frac{\|\Delta w_i\|_\infty}{\|w\|_\infty} \sum_{j=0}^{\infty} \frac{\|\Delta w_{i+j+1}\|_\infty}{\|\Delta w_i\|_\infty} \tag{4.38}$$

$$\approx \frac{\|\Delta w_i\|_\infty}{\|w_i\|_\infty} \sum_{j=0}^{\infty} \left(\frac{\|\Delta w_i\|_\infty}{\|\Delta w_{i-1}\|_\infty} \right)^{j+1} = \frac{\|\Delta w_i\|_\infty}{\|w_i\|_\infty} \frac{\|\Delta w_i\|_\infty}{\|\Delta w_{i-1}\|_\infty} \sum_{j=0}^{\infty} \left(\frac{\|\Delta w_i\|_\infty}{\|\Delta w_{i-1}\|_\infty} \right)^{j} \tag{4.39}$$

$$= \frac{1}{\|w_i\|_\infty} \frac{\|\Delta w_i\|_\infty^2}{\|\Delta w_{i-1}\|_\infty - \|\Delta w_i\|_\infty}. \tag{4.40}$$

Note that we do not ask for a small relative error of the solution at any point of the domain, but just in infinity norm. This basically means that it is acceptable that the relative error of w close to the crack tip (where w goes to 0) might be relatively big. This is necessary to avoid convergence problems due to numerical cancellation in the case of small or zero toughness.
After the iterative process has bee completed, the pressure p is computed integrating $\frac{\partial p}{\partial x}$ and imposing a condition to determine the unknown constant coming from the integration. In the case of PKN model we simply have $p = w$ or equivalently the boundary condition $p(1) = w(1) = 0$. Instead in the case of KGD and radial models we do not have an explicit boundary condition for p, but the constant has to be chosen so that the pressure satisfies $\mathcal{T}(p, L) = K_{IC}$.

Fixed point iterations in general do not converge to a solution. Indeed this is true also in our case, where if the algorithm is directly applied as stated above, the inner cycle will be divergent in most of the cases. To address the problem a standard method is to change the length of the step done at each iteration hoping for better results. More in detail, let us consider a generic fixed point iteration

$$x_{i+1} = F(x_i), \tag{4.41}$$

where we are trying to solve the equation $x = F(x)$. If the iterations do not converge or if the convergence is slow we can define x_{i+1} as

$$x_{i+1} = \lambda F(x_i) + (1 - \lambda) x_i \tag{4.42}$$

and choose λ to get better results. In our case, we found through some numerical tests, that a conservative choice to have convergence for all the models when $0 \leq n \leq 2$, is to choose λ equal to 3/4 for the outer cycle and equal to 1/6 for the inner cycle. Better results in terms of speed can be however obtained choosing λ depending on the model and on the input parameters.

Computational complexity

We discuss here the computational complexity of one iteration of the fixed point solver, depending on N the number of nodes used for approximation.
In the case of PKN one iteration requires only evaluations of simple functions ($O(N)$) and integrations ($O(N \log(N))$). Therefore the overall computational complexity is $O(N \log(N))$.
In the case of KGD and radial models we have in addition to evaluate the integral operator. One evaluation requires the computation of the kernel ($O(N^2)$), the evaluation of the derivative of the pressure at the integration points ($O(N^2)$) and the integration itself ($O(N^2 \log(N))$). Therefore the overall computational complexity is $O(N^2 \log(N))$. However the integral operator is linear and does not depend on the solution, therefore to save time it can be precomputed and stored in form of a matrix. To do so it is necessary to evaluate it on a complete basis, resulting in a computational complexity of $O(N^3 \log(N))$.
Once the integral operator is available as a matrix, for its evaluation it is sufficient a matrix vector product ($O(N^2)$). Therefore for KGD and radial the cost of one iteration is dominated by the evaluation of the integral operator and the overall computational complexity is $O(N^2)$ operations.

4.4 Newton solver

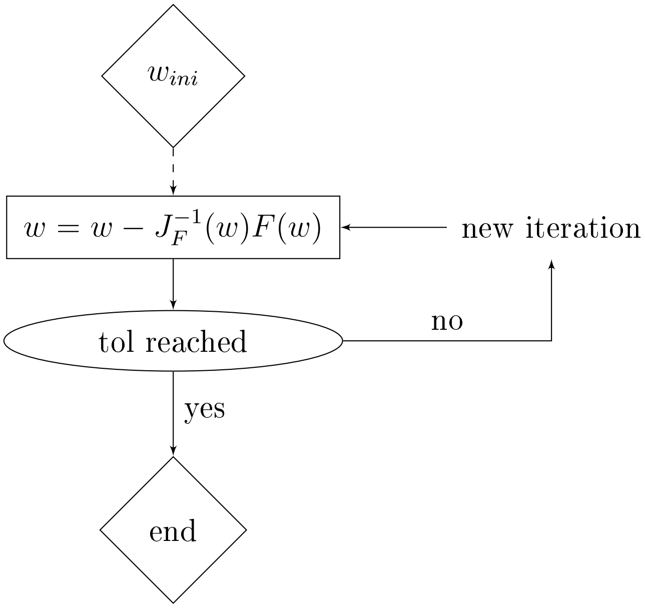
In alternative to the fixed point iteration, we can reformulate the problem as a single equation and use Newton's method to solve the non-linear system that results from the discretisation:

$$F(w) := \mathcal{E}(\mathcal{F}(\mathcal{C}(w, \mathcal{B}(w)), w, \mathcal{B}(w)), \mathcal{B}(w)) - w = 0. \tag{4.43}$$

Again we start with an initial guess of the solution, but Newton's method is more sensitive to the choice of the initial guess than fixed point iteration. Indeed if we want to have convergence we will have to provide a better starting point, like

$$w_{ini}(x) = \begin{cases} a(1-x)^{\frac{1}{n+2}} & \text{if PKN} \\ a(1-x)^{\frac{1}{2}} + b(1-x)^{\frac{2}{n+2}} & \text{if KGD, radial,} \end{cases} \tag{4.44}$$

where a and b depend on the input parameters. In general, for KGD and radial models, in case of small or zero toughness a should be taken much smaller than b, while on the contrary in case of large toughness b should be taken much smaller than a. After choosing the initial guess we update w using Newton's iteration as in (4.45)



(4.45)

where we can compute J_F, the Jacobian of F, as a composition of the Jacobians of the functions that constitute F. The cycle is repeated until the error of the solution of the self-similar problem goes under a desired tolerance.

To estimate the error we use a criterion similar to the one used for fixed point iteration, but this time we also consider the residue $F(w)$. Again we call w_i the value of w at the i-th iteration and define $\Delta w_i := w_i - w_{i-1}$, with in addition $F_i := F(w_i)$. Because Newton's method has quadratic convergence, the ratios $\frac{\|\Delta w_i\|_\infty}{\|\Delta w_{i-1}\|_\infty^2}$ and $\frac{\|F_i\|_\infty}{\|F_{i-1}\|_\infty^2}$ will remain approximately constant for every i. Therefore the error at the i-th iteration can be estimated with

$$err_i := \max(err_{w_i}, err_{F_i}), \quad \text{where} \tag{4.46}$$

$$err_{w_i} := \frac{\|w - w_i\|_\infty}{\|w\|_\infty} = \frac{\|\sum_{j=0}^{\infty} \Delta w_{i+j+1}\|_\infty}{\|w\|_\infty} \leq \frac{\sum_{j=0}^{\infty} \|\Delta w_{i+j+1}\|_\infty}{\|w\|_\infty} \tag{4.47}$$

$$\approx \frac{\|\Delta w_{i+1}\|_\infty}{\|w_i\|_\infty} \approx \frac{1}{\|w_i\|_\infty} \frac{\|\Delta w_i\|_\infty^3}{\|\Delta w_{i-1}\|_\infty^2} \tag{4.48}$$

$$err_{F_i} := \frac{\|F_i\|_\infty}{\|w\|_\infty} \approx \frac{\|F_i\|_\infty}{\|w_i\|_\infty}. \tag{4.49}$$

As we have done for the fixed point iterator, also in Newton's method we do not ask for a small relative error of the solution at any point of the domain, but just in infinity norm.
Again, after the iterative process has been completed, the pressure p is computed integrating $\frac{\partial p}{\partial x}$ and imposing the appropriate condition.

At every iteration of Newton's method it is necessary to solve the linear system $J_F(w_{i-1})\Delta w_i = -F(w_{i-1})$. To solve the linear system it is possible to use a direct method, for example the backslash command in Matlab that uses LU decomposition.
In alternative one can consider an iterative method like for example GMRES. The use of an iterative solver has the advantage that it does not require the computation of the full Jacobian matrix. Indeed in practical cases it is usually necessary to evaluate it only in a small number of directions, therefore reducing the computational complexity.
If we use GMRES, the iterative process must be repeated until the error goes under a desired tolerance. Therefore the error at the j-th GMRES iteration of the i-th Newton's iteration can be estimated as

$$err_{i,j} := \frac{\|J_{F^{asy}}(w_{i-1})\Delta w_{i,j} + F^{asy}(w_{i-1})\|_\infty}{\|w^{asy}\|_\infty} \approx \frac{\|J_{F^{asy}}(w_{i-1})\Delta w_{i,j} + F^{asy}(w_{i-1})\|_\infty}{\|w_{i-1}^{asy}\|_\infty}, \tag{4.50}$$

$$\text{where} \tag{4.51}$$

$$w^{asy} = \frac{w}{(1-x)^\alpha}, \quad F^{asy} = \frac{F}{(1-x)^\alpha}, \quad \alpha = \begin{cases} \frac{1}{n+2} & \text{if PKN} \\ \frac{1}{2} & \text{if KGD or radial.} \end{cases} \tag{4.52}$$

Note that for GMRES we have used a different error estimate compared to fixed point and Newton's method, indeed this time we must be more careful. In fact in our solver when we approximate w, we actually approximate w^{asy} (with an additional change of variable) and therefore if the new value of w^{asy} provided by GMRES has some uncontrolled error close to the crack tip, also the rest of the function will be affected. For this reason the error in GMRES is measured on w^{asy}. On the contrary we did not need to be so strict when we tested the solution of fixed point or Newton's solvers. Indeed in those cases the values close to the crack tip had already been given so that the rest of the function was not badly affected.

Computational complexity

We discuss here the computational complexity of one iteration of Newton's solver, depending on N the number of nodes used for approximation.
In addition to the computations needed for the fixed point solver, Newton's solver requires also

the evaluation of the Jacobian matrix. It can be seen that the complexity of the evaluation of the Jacobian in one direction is the same than the complexity of one evaluation of the function, therefore, similarly to the fixed point case, it is $O(N\log(N))$ for PKN and $O(N^2)$ for KGD and radial.
If Newton's method is implemented using a direct solver for the linear system, the complete Jacobian matrix has to be computed requiring $O(N^3)$ operations. The solution of the system using for example the LU factorisation would require also $O(N^3)$ operations. Therefore the overall complexity is $O(N^3)$.
If instead one uses an iterative solver for the linear system, like for example GMRES, the Jacobian matrix has to be evaluated only on one direction for each GMRES iteration and the cost of one GMRES iteration itself is $O(N^2)$. If the number of GMRES iterations required remains quite small, as it turns out to be in our case, it is clear that iterative solver offers the best performance. As a result, the overall computational complexity using GMRES is $O(N^2)$.

4.5 Multigrid method

Multigrid methods can be used successfully to accelerate the convergence in the solution of boundary value problems, see for instance [74]. The main idea is to work with several grids of different refinement, and use the computations from the coarser grids to speed up the computations for the finer grids. Methods based on different techniques and with different levels of complexity have been developed, we will just use a basic implementation of the main idea. Nevertheless this will allow us to speed up the execution of the algorithm while at the same time to estimate the error of the solution due to the discretisation of the problem.

We describe now our multigrid procedure. We start solving the self-similar problem on a grid with $2^3+1=9$ nodes. After that, when we have a solution w^m on the grid x^m with 2^m+1 nodes, we use it as initial guess to find the solution w^{m+1} on the grid x^{m+1} with $2^{m+1}+1$ nodes. Then we compare the solutions w^m and w^{m+1} on the grid x^m and estimate the error as $\frac{\|w^m-w^{m+1}\|_\infty}{\|w^{m+1}\|_\infty}$. If the error is under the required tolerance we stop, otherwise we refine the grid and repeat the procedure.
We chose to use the grids x^m with 2^m+1 elements because, as we have seen in section 3.1, they are nested. This allows to pass directly from the finer grids to the coarser ones just keeping the values of the functions at the relevant points, while the opposite process can be obtained with just one IDTT-1 and one DTT-1. For example if we want to pass between the set x^m and the set x^{m+1} we can just do

$$f(x^M) = f(x_0^{m+1}, x_2^{m+1}, \dots, x_{2^{m+1}}^{m+1}) \tag{4.53}$$

$$f(x^{m+1}) = \mathcal{T}(\mathcal{T}^{-1}(f(x^m)), 0, \dots, 0), \tag{4.54}$$

where $\mathcal{T}(x)$ is the DTT-1, $\mathcal{T}^{-1}(x)$ is the IDTT-1 and in (4.54) we have added 2^m zeros. Another advantage for using sets with 2^m+1 elements is that DTT-1, IDTT-1, DUT-1, IDUT-1 are optimised for this number of nodes, being based on a DCT-1 of the same size. This is due to the fact that the algorithms generally used to perform the DCT-1 are optimised for 2^m+1 elements, equivalently to what happens to the FFT of 2^m elements.

The multigrid procedure that we have just described allows us to estimate the error of the solution at any point of the algorithm, so we can decide when to stop without ending up using a grid

that is too fine for our needs. Moreover, each time we increase the nodes, we have a good initial guess from the previous solution to start the solver of the non-linear system and therefore we will need to do less iterations. In this way most part of the job is done using coarser meshes and only few iterations are needed with the finest ones.
Note that the speed-up also depends on the complexity of a single fixed point or a Newton iteration. If the complexity of one iteration on N nodes is for example $O(N^2)$, then the advantage of reducing the number of nodes is clearly greater compared to the case when the complexity is $O(N)$.

We have just seen that to approximate the functions of PKN, KGD and radial models we work with the sets of nodes x^m with $2^m + 1$ elements. However we recall that in KGD and radial models we also have another set of nodes to evaluate the integral operator. Due to the additional difficulty coming from the presence of a non smooth kernel, it turns out that for the evaluation of the integral operator it is a good choice to use a number of nodes that is roughly double than for the approximation of the functions. Therefore, to each node of x_i^m will correspond $(2^{m+1} + 1)$ nodes for the integral operator, forming the set $y^{m,2m}$ of $(2^m + 1)(2^{m+1} + 1)$ elements. However as previously mentioned, the integral operator can be precomputed and stored as a matrix, and independently of how many nodes are used internally for its computation, the matrix will always have size $(2^m + 1) \times (2^m + 1)$.

4.6 Comparison with benchmark

To verify the correctness of our solver, we compare it with some semi-analytical benchmarks for the self-similar problem that have been proposed in [60, 61] for PKN and KGD and in [55, 56] for radial. We will use our solver with a number of points N big enough to allows us to get a better approximation of the solution than the benchmark itself, therefore actually measuring the error of the benchmark. We will consider the relative error of the crack opening and length

$$err_w := \frac{|w_{bm} - w_N|}{|w_N|}, \quad err_L := \frac{|L_{bm} - L_N|}{|L_N|}, \tag{4.55}$$

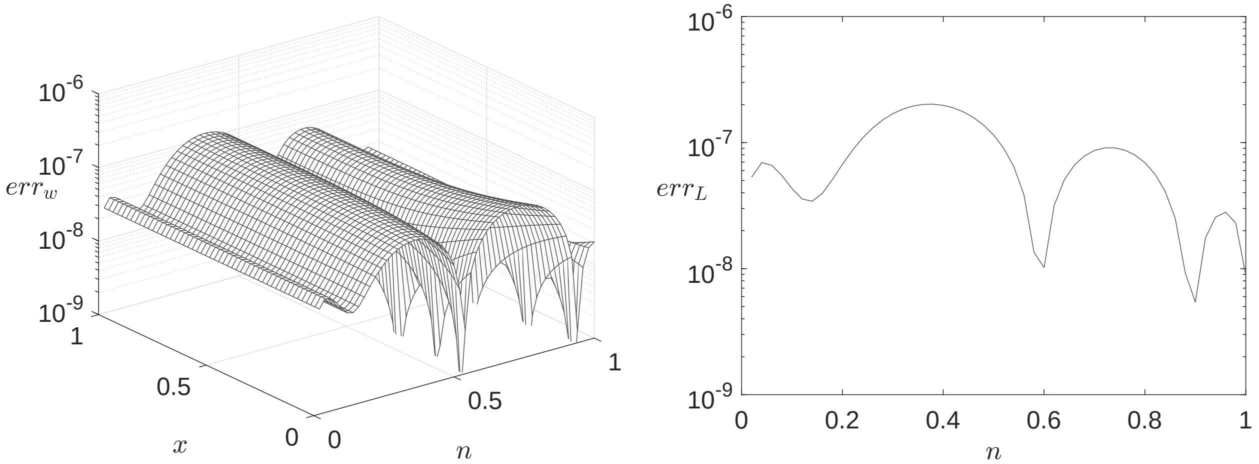
where w_N, L_N is our solution with N nodes and w_{bm}, L_{bm} is the semi analytical benchmark. In some plots we will also use the average error $err_{w_{avg}}$. Similar tests have been done in the aforementioned references when building the benchmarks, therefore the results obtained here can be verified.
Note that in all the tests of the self-similar solver we will take $q_* = 1$. The choice of q_* indeed is not important because in PKN model a change in q_* simply corresponds to a rescaling of the solution. In KGD and radial models instead the situation is more complicated, nevertheless for every combination q_*, K_{IC} it is still possible to find a new value of K_{IC} that, if combined together with $q_* = 1$, gives a solution that differs form the original solution only for a rescaling. For this reason in our tests we will use a range of values of toughness, in such a way to cover all possible situations.

PKN

The semi-analytical benchmark for PKN provided in [60, 61] is valid in the case of constant flow, that corresponds to $\gamma = \frac{1}{2n+3}$. We solve the problem using $N = 2^3 + 1 = 9$ nodes and use the numerical solution to measure the error of the benchmark.

– In figure 4.1 we plot the relative error of w and L for different values of the fluid parameter n. The plots confirm the correctness of our solver.

Figure 4.1: Error of the PKN semi-analytical benchmark, varying n.



KGD

The semi-analytical benchmark for KGD provided in [60, 61] is valid in the case of constant flow when $K_{IC} = 0$, that corresponds to $\gamma = \frac{1}{n+2}$ and in the case of constant toughness when $K_{IC} > 0$, that corresponds to $\gamma = \frac{n}{n+2}$. We solve the problem using $N = 2^4 + 1 = 17$ nodes and use the numerical solution to measure the error of the benchmark.
– In figure 4.2 we plot the relative error of w and L for $K_{IC} = 1$ and different values of n.
– In figure 4.3 we plot the relative error of w and L for $n = 1$ and different values of K_{IC}.
– In figure 4.4 we plot the relative error of w and L for different values of K_{IC} and n.
The plots confirm the correctness of our solver.

Figure 4.2: Error of the KGD semi-analytical benchmark, $K_{IC} = 1$ varying n.

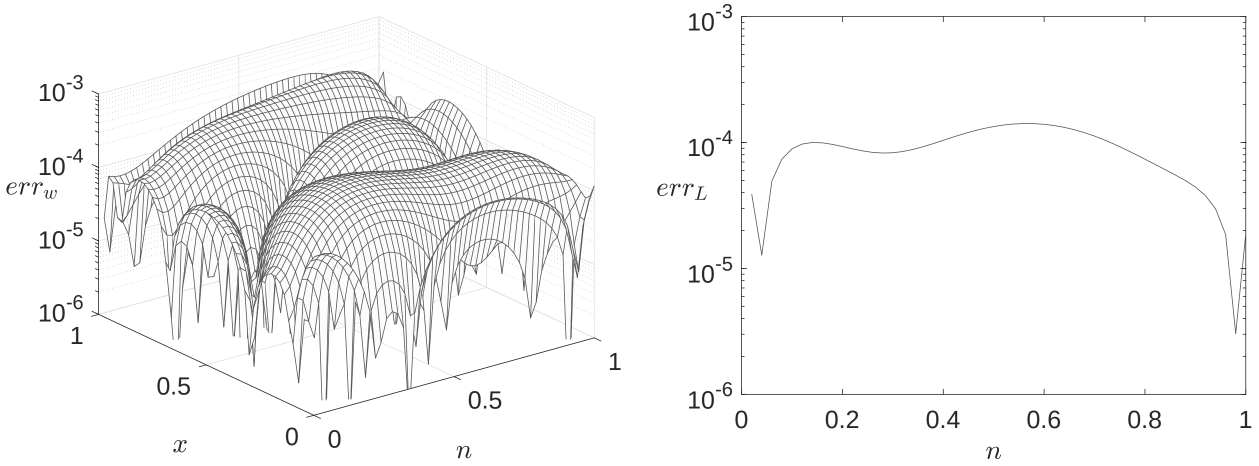


Figure 4.3: Error of the KGD semi-analytical benchmark, $n = 1$ varying K_{IC}.

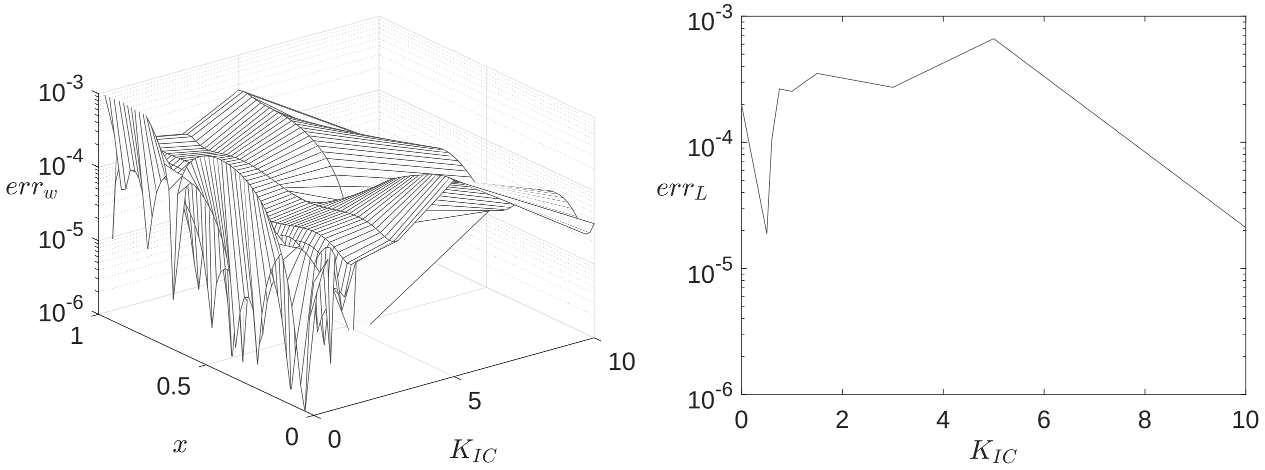
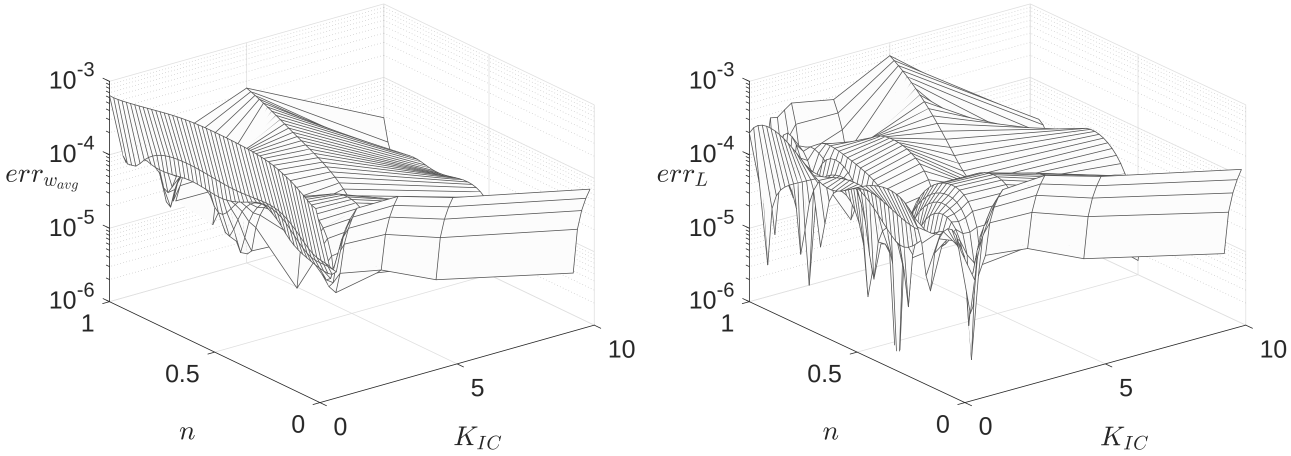


Figure 4.4: Error of the KGD semi-analytical benchmark, varying K_{IC} and n.



Radial

The semi-analytical benchmark for radial provided in [55, 56] is valid in the case of $\gamma = \frac{2-n}{3(n+2)}$, that corresponds to constant input. We solve the problem using $N = 2^4 + 1 = 17$ nodes and use the numerical solution to measure the error of the benchmark.
– In figure 4.5 we plot the relative error of w and L for $K_{IC} = 1$ and different values of n.
– In figure 4.6 we plot the relative error of w and L for $n = 1$ and different values of K_{IC}.
– In figure 4.7 we plot the relative error of w and L for different values of K_{IC} and n.
The plots confirm the correctness of our solver.

Figure 4.5: Error of the radial semi-analytical benchmark, $K_{IC} = 1$ varying n.

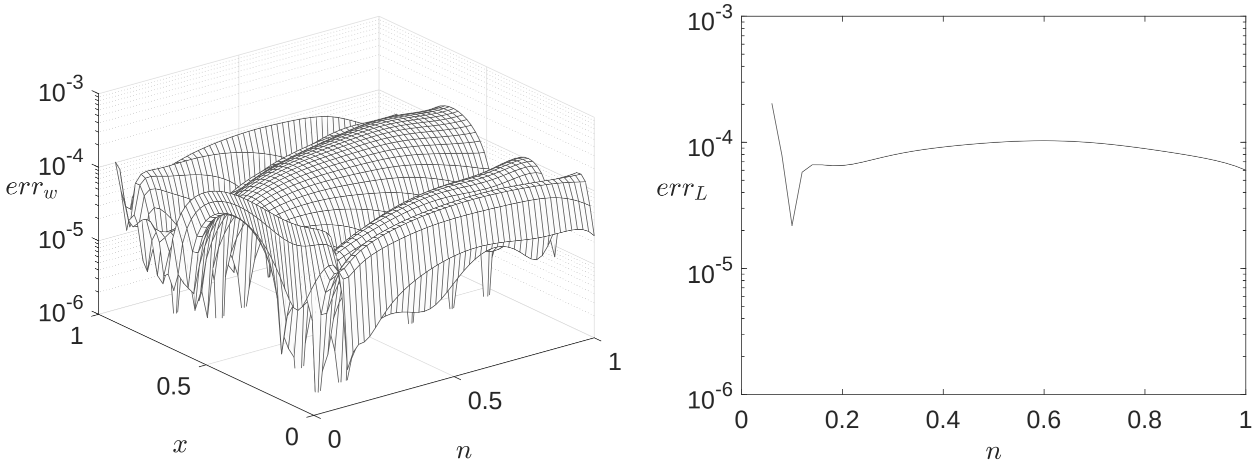


Figure 4.6: Error of the radial semi-analytical benchmark, $n = 1$ varying K_{IC}.

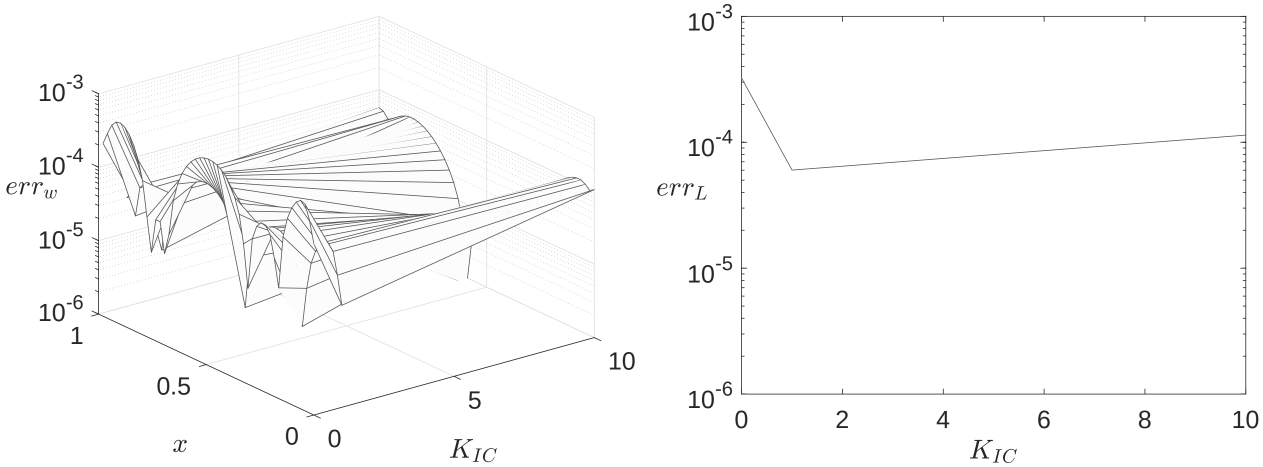
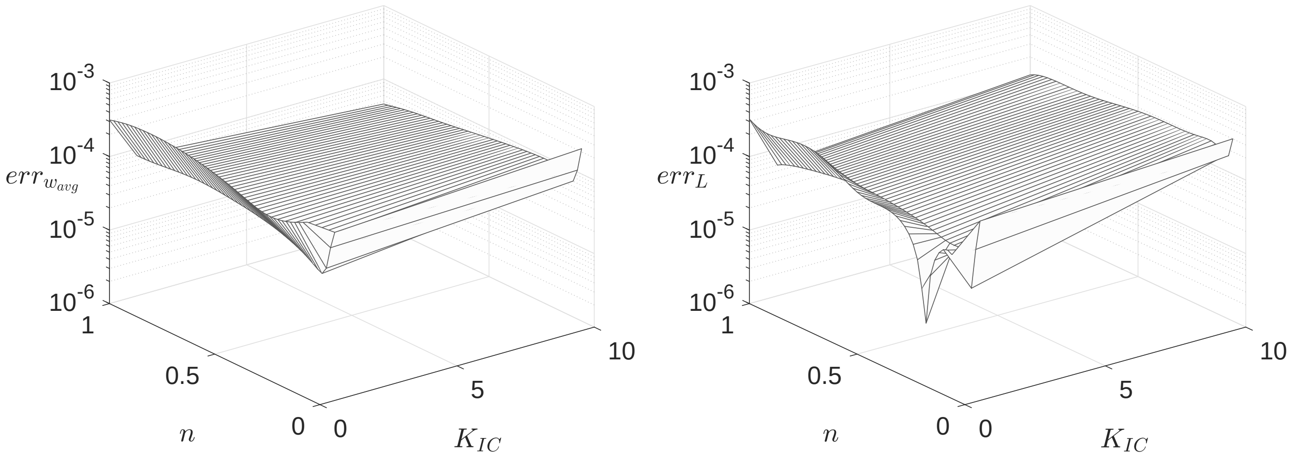


Figure 4.7: Error of the radial semi-analytical benchmark, varying K_{IC} and n.



4.7 Convergence analysis

We want to verify the accuracy and the convergence properties of our solver. To do so we will compute a numerical benchmark running our solver with a high number of nodes N_{bm} and then compare it to the solutions obtained using smaller values of N. We will consider the relative error of the crack opening and the crack length

$$err_w := \frac{|w_N - w_{N_{bm}}|}{|w_{N_{bm}}|}, \quad err_L := \frac{|L_N - L_{N_{bm}}|}{|L_{N_{bm}}|}, \tag{4.56}$$

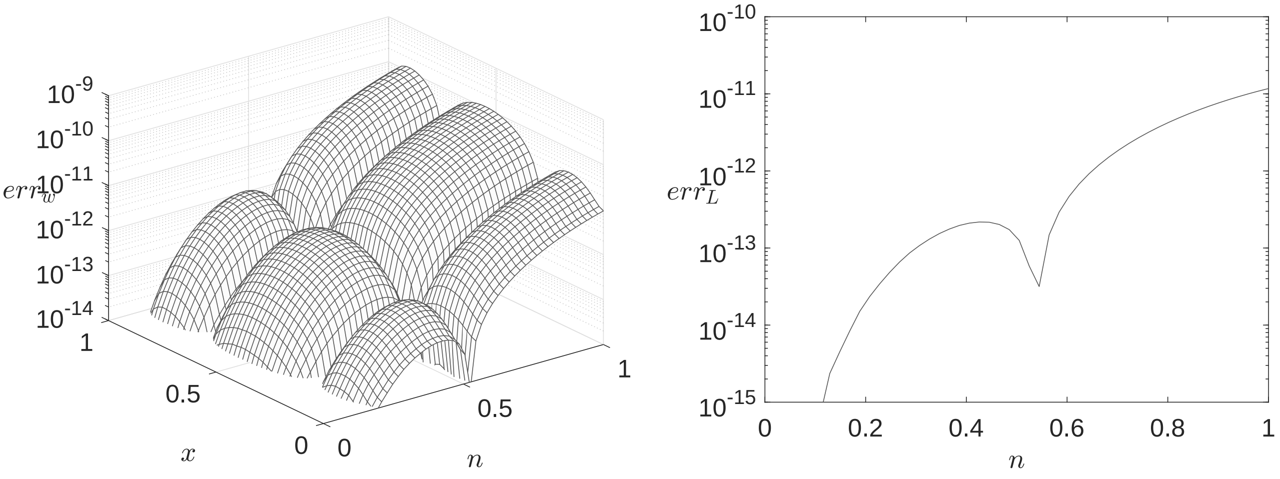
where w_N, L_N is the solution with N nodes and $w_{N_{bm}}, L_{N_{bm}}$ is the numerical benchmark. In some plots we will also use the average error $err_{w_{avg}}$.

PKN

For the PKN model we consider the case with constant flow, that corresponds to $\gamma = \frac{1}{2n+3}$ and take the solution with $N_{bm} = 2^4 + 1 = 17$ nodes as a numerical benchmark.

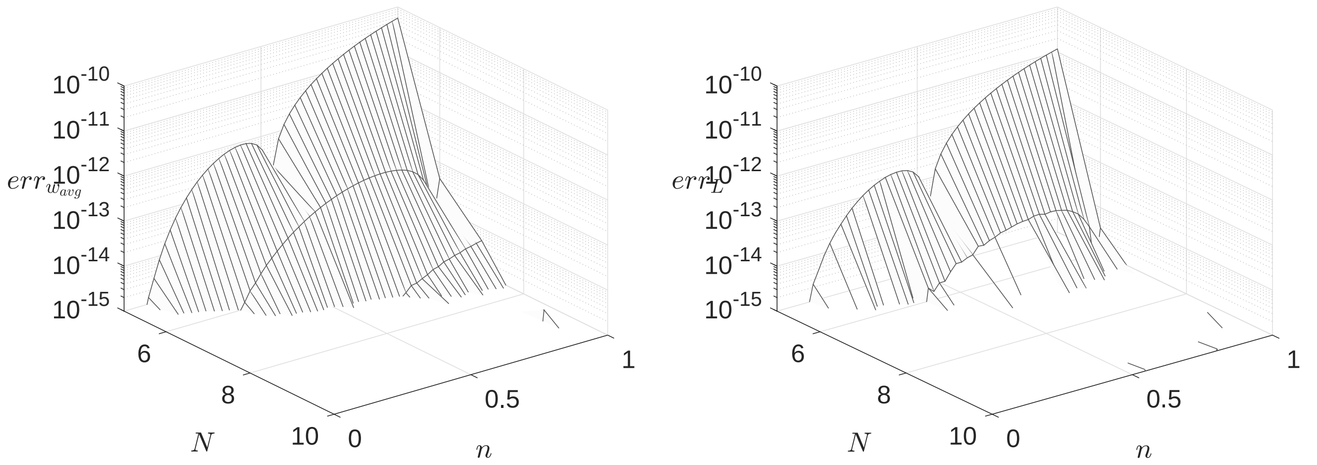
Here we consider the solution obtained using $N = 2^2 + 1 = 5$ nodes.
– In figure 4.8 we show the relative error of w and L for different values of n.
With only 5 nodes the self-similar problem is already solved with an extremely good accuracy.

Figure 4.8: Error of the PKN numerical solution with $N = 5$, varying n.



Now we look at the convergence of the solution varying N the number of nodes.
– In figure 4.9 we show the relative error of w and L for different values of n and N.
With 10 nodes we are already at machine precision.

Figure 4.9: Error of the PKN numerical solution, varying n and N.



KGD

For the KGD model we consider the case with constant toughness, that corresponds to $\gamma = \frac{n}{n+2}$ and take the solution with $N_{bm} = 2^8 + 1 = 257$ nodes as a numerical benchmark.

Here we consider the solution obtained using $N = 2^4 + 1 = 17$ nodes.
– In figure 4.10 we show the relative error for $K_{IC} = 1$ and different values of n.
– In figure 4.11 we show the relative error for $n = 1$ and different values of K_{IC}.
– In figure 4.12 we show the relative error for different values of K_{IC} and n.
With only 17 nodes the self-similar problem is already solved with a good accuracy. We can see from the plots that the error does not change very much with different values of x, instead when n is small the problem becomes a little bit easier. It is also evident that the small toughness case is much harder to solve than the large toughness case.

Figure 4.10: Error of the KGD numerical solution with $N = 17$, $K_{IC} = 1$ varying n.

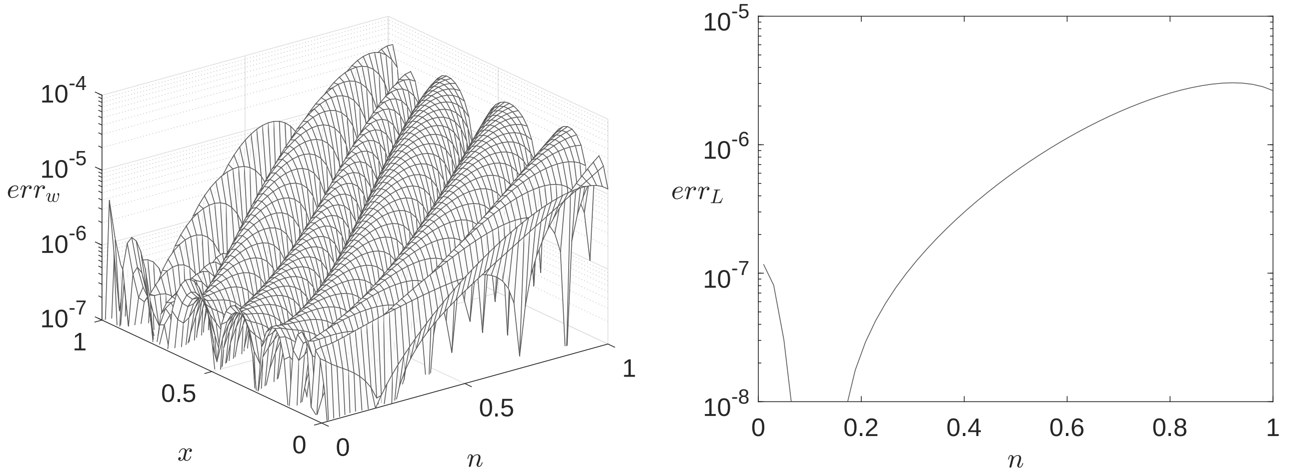


Figure 4.11: Error of the KGD numerical solution with $N = 17$, $n = 1$ varying K_{IC}.

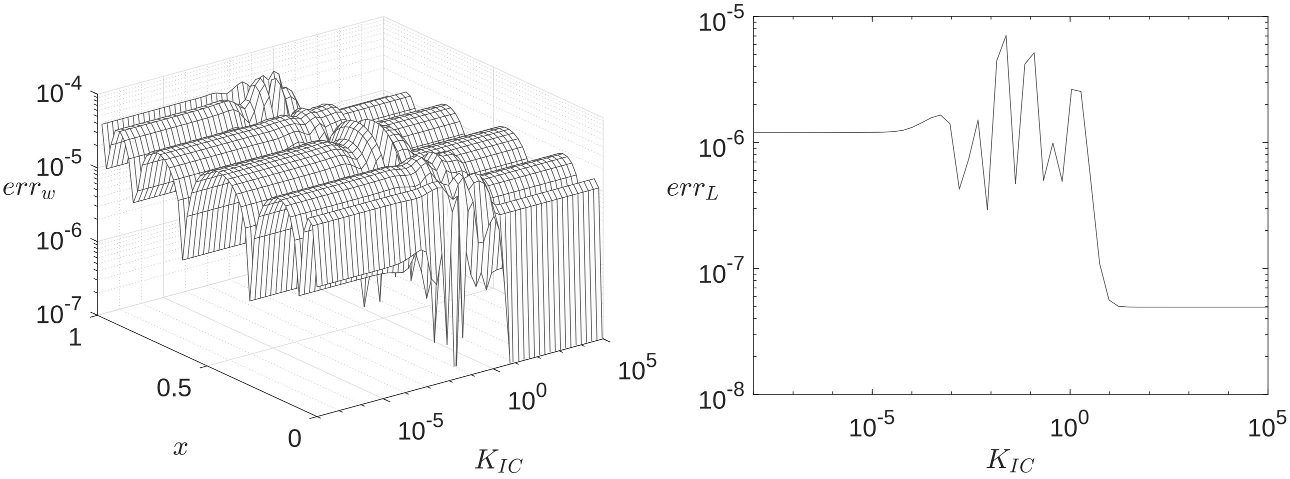
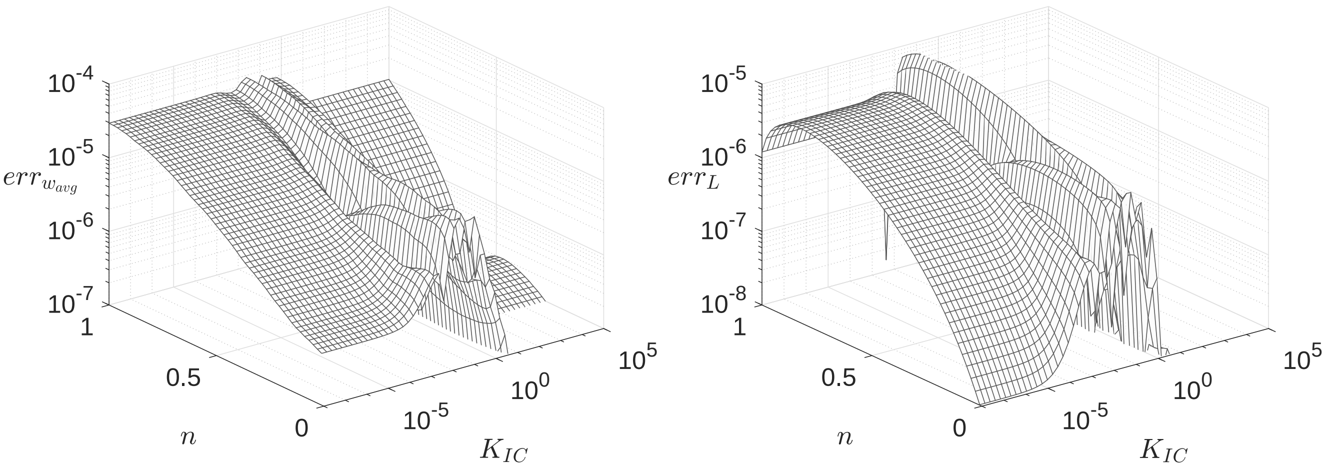


Figure 4.12: Error of the KGD numerical solution with $N = 17$, varying K_{IC} and n.



Now we look at the convergence of the solution varying N the number of nodes.
– In figure 4.13 we show the relative error for $K_{IC} = 1$ and different values of n and N.
– In figure 4.14 we show the relative error for $n = 1$ and different values of K_{IC} and N.
The convergence is much slower compared to PKN, but when $K_{IC} = 1$ with 50 nodes the error is already close to 10^{-10} for any value of n. It is again confirmed that the convergence is faster in the case of large toughness, while in the case of small toughness it gets slower as the number of nodes increases. The problem is particularly difficult in the transition area when toughness is small but not yet close enough to 0, due to the bad asymptotic behaviour of the solution.

Figure 4.13: Error of the KGD numerical solution with $K_{IC} = 1$, varying n and N.

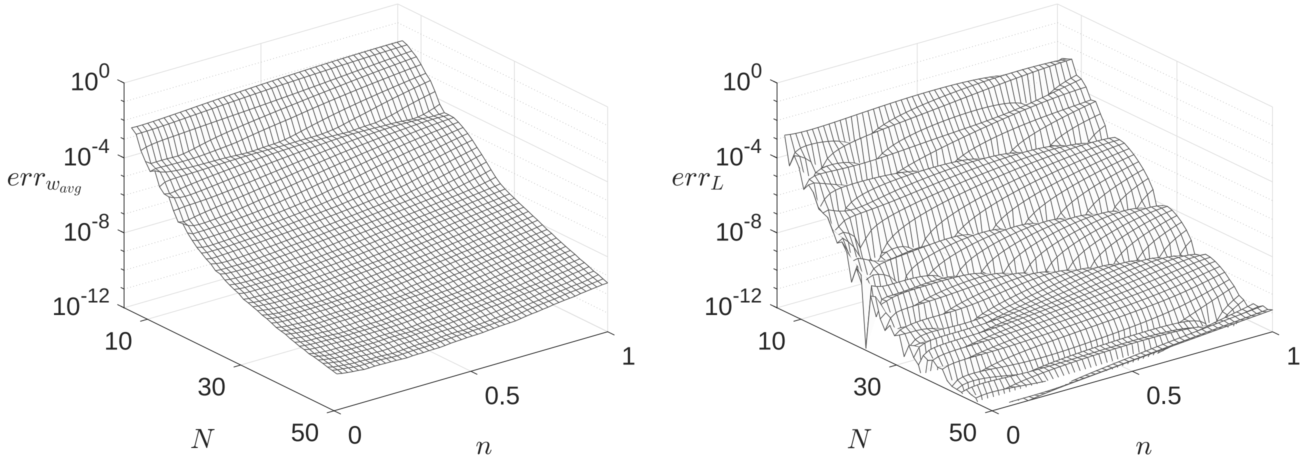
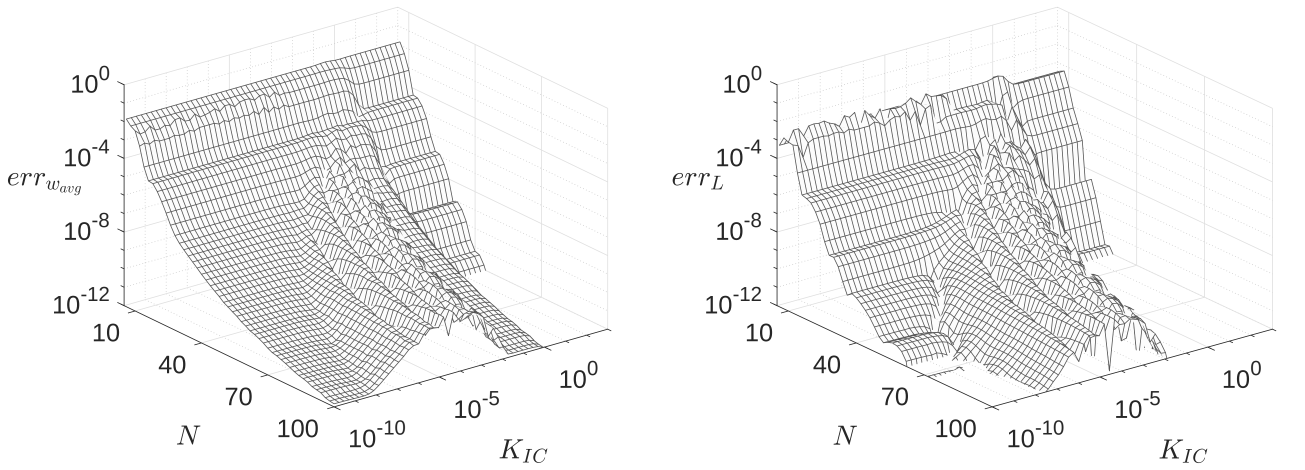


Figure 4.14: Error of the KGD numerical solution with $n = 1$, varying K_{IC} and N.



Compared to the results previously obtained in [60, 61], we provide a solver capable to work for every value of toughness, while with the other solver this was not possible. The accuracy of the solution using the same number of nodes is also improved.

Radial

For the radial model we proceed as for KGD. We consider the case with constant toughness, that corresponds to $\gamma = \frac{n}{n+2}$ and take the solution with $N_{bm} = 2^8 + 1 = 257$ nodes as a numerical benchmark.

Here we consider the solution obtained using $N = 2^4 + 1 = 17$ nodes.
– In figure 4.15 we show the relative error for $K_{IC} = 1$ and different values of n.
– In figure 4.16 we show the relative error for $n = 1$ and different values of K_{IC}.
– In figure 4.17 we show the relative error for different values of K_{IC} and n.
As for KGD, with only 17 nodes the self-similar problem is already solved with a good accuracy.

We can see from the plots that the error does not change very much with different values of x. When n is small the problem becomes a bit easier. Instead it is again evident that the small toughness case is harder to solve than the large toughness case.

Figure 4.15: Error of the radial numerical solution with $N = 17$, $K_{IC} = 1$ varying n.

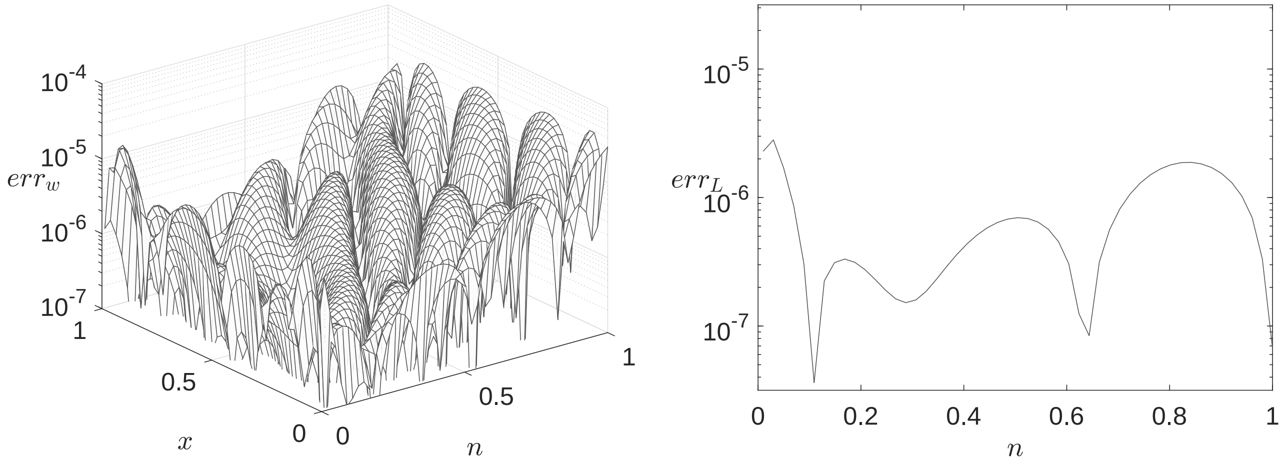


Figure 4.16: Error of the radial numerical solution with $N = 17$, $n = 1$ varying K_{IC}.

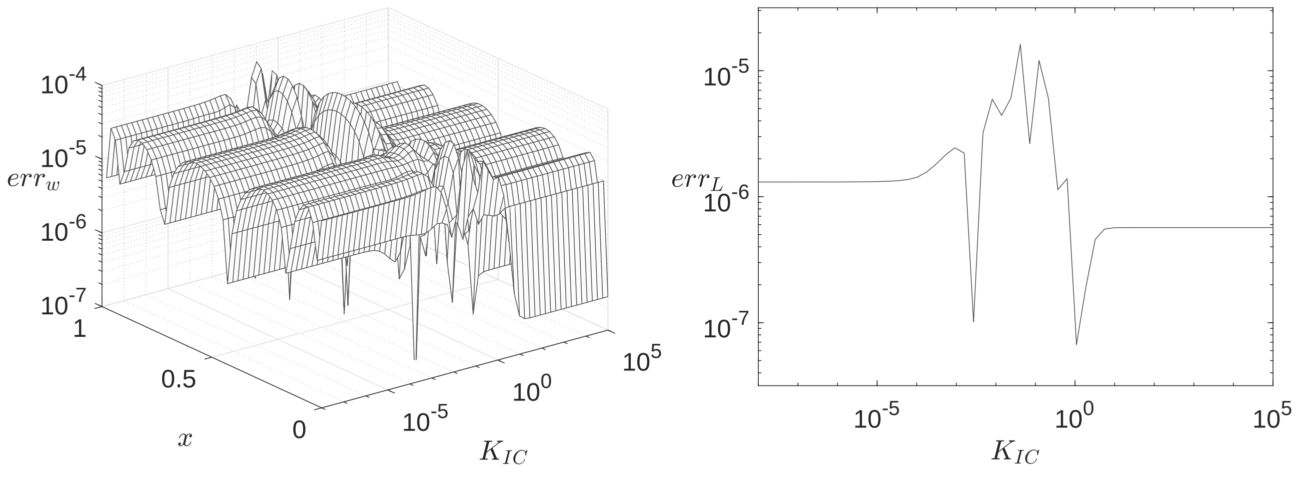
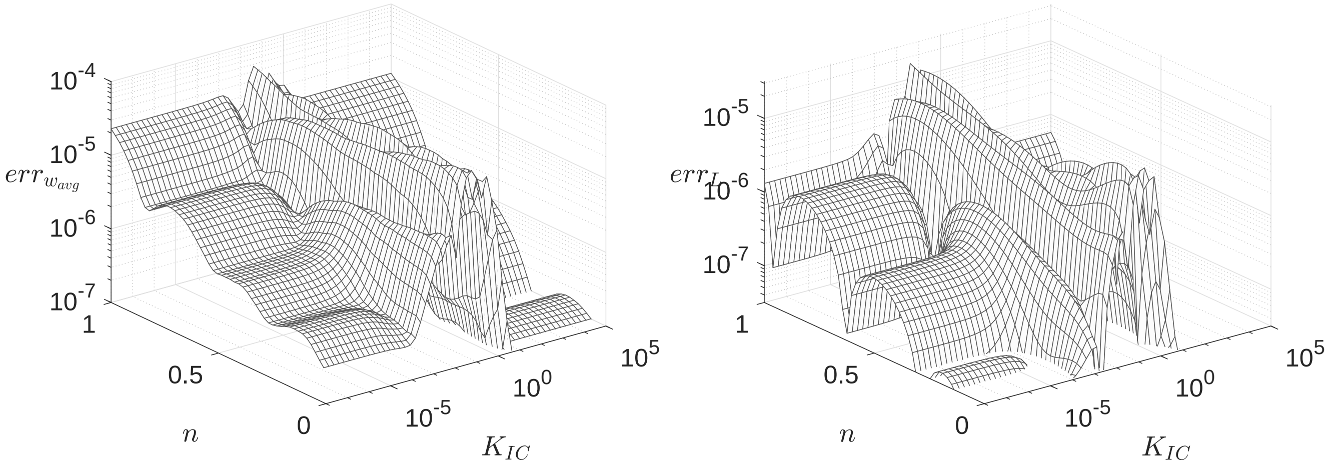


Figure 4.17: Error of the radial numerical solution with $N = 17$, varying K_{IC} and n.



Now we look at the convergence of the solution varying N the number of nodes.
– In figure 4.18 we show the relative error for $K_{IC} = 1$ and different values of n and N.
– In figure 4.19 we show the relative error for $n = 1$ and different values of K_{IC} and N.
The convergence is much slower compared to PKN, but when $K_{IC} = 1$ with 50 nodes the error is already close to 10^{-10} for any value of n. It is again confirmed that the convergence is faster in the case of large toughness, while in the case of small toughness it gets slower as the number of nodes increases. The problem is particularly difficult in the transition area when toughness is small but not yet close enough to 0, due to the bad asymptotic behaviour of the solution.

Figure 4.18: Error of the radial numerical solution with $K_{IC} = 1$, varying n and N.

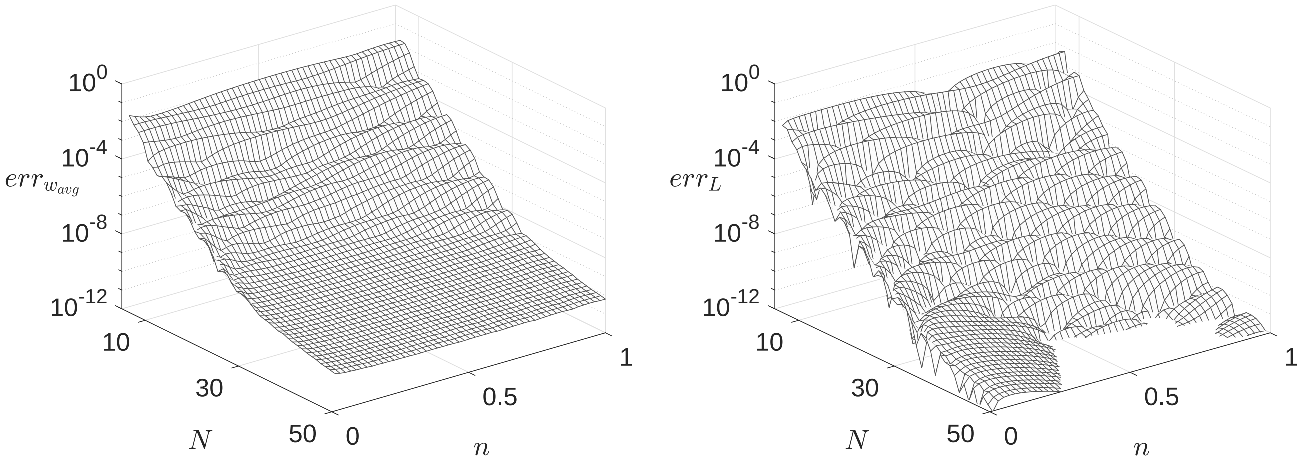
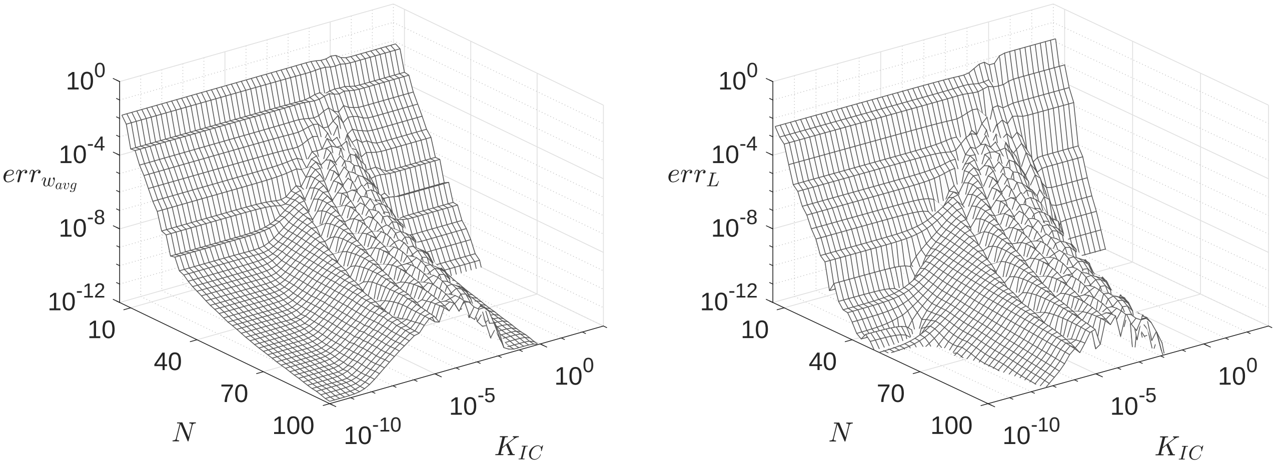


Figure 4.19: Error of the radial numerical solution with $n = 1$, varying K_{IC} and N.



Like for KGD, compared to the results previously obtained in [55, 56], we provide a solver capable to work for every value of toughness, while with the other solver this was not possible. The accuracy of the solution using the same number of nodes is also improved.

4.8 Conclusion

In this chapter we have built an effective solver for the 1D HF self-similar problem.

- The solver has been validated comparing the solutions with some self-similar benchmarks present in literature.
- The quality of the solution is basically independent from the values of fluid parameter n.
- For PKN model machine precision is reached with very few nodes.
- For KGD and radial models it is possible to obtain an acceptable solution already with very few nodes.
- For KGD and radial models it is possible to obtain a very accurate solution with a moderate amount of nodes, even in the numerically difficult small toughness case.

Chapter 5

Solver for the time-dependent problem

In the previous chapter we have seen how to solve the self-similar problem, now we pass to the time-dependent problem. For the spatial discretisation we will use the same approach of the self-similar solver, while we will use an implicit Runge Kutta method for the temporal discretisation. The resulting non-linear system will be solved using Newton's method. In addition to the impermeable fracture case, we will also consider Carter leak-off. Finally we will verify our results comparing them with the self-similar solutions from the previous chapter and, in case of Carter leak-off, with some long time asymptotes.

5.1 General formulation

As we have done for the self-similar problem, we will write the equations in a generic way to allow us to treat PKN, KGD and radial models all together. The elasticity equation, the fracture propagation condition and the fluid flow equation remain the same as for the self-similar problem (see section 4.1), the only difference being that now the functions depend also on time. On the contrary we need to modify the continuity equation, add Carter leak-off and integrate L' and $\frac{\partial w}{\partial t}$ over time.

– **crack length**

$$L(t) = \mathcal{L}(L'), \quad \text{where} \tag{5.1}$$

$$\mathcal{L}(L') = L(t_*) + \int_{t_*}^{t} L'(t)dt. \tag{5.2}$$

– **crack width**

$$w(t,x) = \mathcal{W}\left(\frac{\partial w}{\partial t}\right), \quad \text{where} \tag{5.3}$$

$$\mathcal{W}\left(\frac{\partial w}{\partial t}\right) = w(t_*,x) + \int_{t_*}^{t} \frac{\partial w}{\partial t}(t,x)dt. \tag{5.4}$$

– **continuity equation**

$$q(t,x) = \mathcal{C}\left(\frac{\partial w}{\partial t}, w, L', L, q_l\right), \quad \text{where} \tag{5.5}$$

$$\mathcal{C}\left(\frac{\partial w}{\partial t}, w, L', L, q_l\right) = \begin{cases} L'(t)xw(x) + \int_x^1 L'(t)w(x) + L(t)\left(\frac{\partial w}{\partial t}(t,x) + q_l(t,x)\right)dx \\ \text{if PKN, KGD} \\ \\ \left(L'(t)x^2 w(x) + \int_x^1 2L'(t)xw(x) + L(t)x\left(\frac{\partial w}{\partial t}(t,x) + q_l(t,x)\right)dx\right)/x \\ \text{if radial,} \end{cases} \tag{5.6}$$

this formulation can be derived integrating the continuity equation in the differential form.

– **fluid balance equation**

$$L'(t) = \mathcal{B}\left(\frac{\partial w}{\partial t}, w, L, q_l, q_*\right), \quad \text{where} \tag{5.7}$$

$$\mathcal{B}\left(\frac{\partial w}{\partial t}, w, L, q_l, q_*\right) = \begin{cases} \dfrac{q_*(t) - L(t)\int_0^1 \frac{\partial w}{\partial t}(t,x) + q_l(t,x)dx}{\int_0^1 w(x)dx} & \text{if PKN, KGD} \\ \\ \dfrac{\frac{q_*(t)}{L(t)} - L(t)\int_0^1 \frac{\partial w}{\partial t}(t,x) + q_l(t,x)dx}{2\int_0^1 w(x)xdx} & \text{if radial,} \end{cases} \tag{5.8}$$

this formulation can be derived integrating the continuity equation in the differential form and applying the boundary condition on the fluid input at the crack mouth.

– **Carter leak-off**

$$q_l(t,x) = \mathcal{Q}(L), \quad \text{where} \tag{5.9}$$

$$\mathcal{Q}(L) = \frac{k_{cl}}{\sqrt{t - L^{-1}(L(t)x)}}, \tag{5.10}$$

the same for all models.

In the next sections we will see how to discretise the problem and from now on, when we will refer to the operators just defined, we will mean their discretised version where for the time we consider a single time step starting from t_*.

5.2 Spatial discretisation

The spatial discretisation in case of no leak-off remains the same as for the self-similar problem. Instead with Carter leak-off in some cases there is a change in the asymptotic behaviour at the crack tip, therefore it will be necessary to do some modifications.

In the case of PKN model when $n = 0$ the asymptotic behaviour is the same with or without leak-off, therefore no changes are needed. Instead when $n > 0$ the solution has a different asymptotic behaviour, that further changes when time goes to infinity, see section 2.2.

If we ignore what happens for large values of time, we can consider the asymptotic behaviour of the functions as

$$f(x) = a_0(1-x)^{\alpha_0} + a_1(1-x)^{\alpha_1} + \text{h.o.t.} \quad x \to 1, \tag{5.11}$$

$$\text{where} \quad \alpha_1 = \alpha_0 + \frac{1}{2} - \frac{1}{n+2} = \alpha_0 + \frac{n}{2(n+2)}, \tag{5.12}$$

where α_0 remains the same as for no leak-off. The changes of variables s (3.126) and (3.135) depend on $\alpha_1 - \alpha_0 = \frac{1}{2} - \frac{1}{n+2}$ and the parameter m_a that can be chosen equal to 2. In this way, if n is not too small, the solver works well for small and medium values of time, while it struggles when approaching the long time asymptote. This happens for the same reason previously seen in the case of small toughness, indeed when time goes to infinity a_0 tends to 0 while a_1 tends to infinity, trying to mimic the behaviour of the long time asymptote. Nevertheless if we do not ask for very small error the solver still works, even though for large values of time it requires a lot of discretisation points. An additional problem instead happens when we decrease n, in fact in this case we can see that $\alpha_1 - \alpha_0$ also gets smaller and tends to 0 when n tends to 0. Therefore in this situation it becomes practically impossible to use this approach because the change of variables s (3.126) and (3.135) becomes too extreme and the approximation nodes are all moved close to the crack tip.
If instead we want to include in the approximation also the behaviour of the long time asymptote, we should then add also its asymptotic terms. Therefore we can consider the asymptotic behaviour of the functions as

$$f(x) = a_0(1-x)^{\alpha_0} + a_1(1-x)^{\alpha_1} + \text{h.o.t.} \quad x \to 1, \tag{5.13}$$

$$\text{where} \quad \alpha_1 = \alpha_0 + \frac{n+2}{4n+4} - \frac{1}{n+2} = \alpha_0 + \frac{n^2}{4(n+1)(n+2)} \tag{5.14}$$

where again α_0 remains the same as for no leak-off. We can immediately see that in this way the transformation is practically never usable, because for $n = 1$ we have already that $\alpha_1 - \alpha_0 = \frac{1}{24}$ and if n is smaller the situation gets even worse.
What we can do instead is to take inspiration from what we have previously done for KGD and radial in case of small toughness and take the approximant with a stronger leading term the one of the actual solution. Therefore we can consider the asymptotic behaviour of the functions as

$$f(x) = a_0(1-x)^{\alpha_0} + a_1(1-x)^{\alpha_1} + \text{h.o.t.} \quad x \to 1, \tag{5.15}$$

$$\text{where} \quad \alpha_1 = \alpha_0 + \frac{1}{n+2} \tag{5.16}$$

and $\alpha_0 = 0$ for w, q and p and $\alpha_0 = -1$ for $\frac{\partial p}{\partial x}$. We note that the coefficient a_0 must be forced to 0, instead the value of m_a can be chosen equal to 3 to make the approximation more effective. Proceeding in this way we are able to approximate accurately the solution with a small number of nodes, for both small and large values of time and for every $n < 2$. The only drawback is that the solution is no longer approximated very accurately right close to the crack tip, but however this does not affect the overall quality of the solution on the rest of the interval. We remark that this spatial discretisation can be used effectively also in the case with no leak-off, but the performance is obviously reduced with respect to the discretisation proposed and tested in the previous chapter.

In the case of KGD and radial models the asymptotic behaviour with no leak-off of with Carter leak-off remains the same if $K_{IC} > 0$, while it changes when $K_{IC} = 0$. To obtain a solver that works for every value of K_{IC} with Carter leak-off, we will follow the same idea used in the no leak-off case, trying to keep into account the asymptotic behaviour of both cases, see sections 2.3 and 2.4. Therefore we will consider the asymptotic behaviour functions of the problem as

$$f(x) = a_0(1-x)^{\alpha_0} + a_1(1-x)^{\alpha_1} + \text{h.o.t.} \quad x \to 1, \tag{5.17}$$

$$\text{where} \quad \alpha_1 = \alpha_0 + \frac{n+4}{4n+4} - \frac{1}{2}, \tag{5.18}$$

where α_0 remains the same as for no leak-off. The changes of variables s (3.126) and (3.135) depend on $\alpha_1 - \alpha_0 = \frac{n+4}{4n+4} - \frac{1}{2}$ and the parameter m_a that can be chosen equal to 2 as in the case with no leak-off.

5.3 Temporal discretisation

The problem of HF is stiff, therefore the use of a method suitable for this situation is usually better. To determine if a method performs well on stiff problems the concepts of A-stability and L-stability have been introduced [35].

We recall that a method is said A-stable if when applied to solve the differential equation

$$f'(x) = kf(x) \tag{5.19}$$

where $k \in \mathbb{C}$ and $\Re(k) < 0$, then the approximated solution tends to 0 as the number of steps tends to infinity, for any given step length. A method is instead said L-stable if when applied to solve the same differential equations, then the solution tends to 0 after a single step as the step length tends to infinity.

To be considered good for stiff problems, a method should be at least A-stable, or preferably L-stable (L-stability is a stronger property and implies A-stability). The main groups of methods for temporal discretisation in integro-differential equations are linear multistep methods and Runge Kutta methods. In both cases the explicit methods are never A-stable, so the choice for a stiff problem is restricted to the implicit ones. Moreover linear multistep methods of order greater than 2 can never be A-stable [18], therefore if one wants to use an high order method the choice must fall on Runge Kutta. Among the implicit Runge Kutta methods, we will use the Gauss Radau IIA, a family that is L-stable for any order.

In general Runge Kutta methods to solve a differential equation $\frac{\partial y}{\partial t} = f(t, y)$ take the form

$$y_{n+1} = y_n + h \sum_{i=1}^{S} b_i k_i \tag{5.20}$$

$$k_i = f\left(t_n + c_i h,\ y_n + h \sum_{j=1}^{S} a_{ij} k_j\right), \quad i = 1, \dots, S, \tag{5.21}$$

where the parameters A, $\mathbf{b}$, $\mathbf{c}$ are usually arranged in the Butcher tableau

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b}^{\mathbf{T}} \end{array} \tag{5.22}$$

The methods of the Radau IIA family are collocation methods based on the Radau quadrature formula (for more details about Radau IIA family see [11]). Indeed, the vector $\mathbf{c}$ is formed by the nodes of Radau quadrature rescaled in $[0,1]$, that are the roots of $P_S(2x-1)-P_{S-1}(2x-1)$ where P_S is the Legendre polynomial of degree S. In a collocation method, the matrix A and the vector $\mathbf{b}$ must satisfy the relation

$$\sum_{j=1}^{S} a_{ij}p(c_j) = \int_0^{c_i} p(x)dx, \quad i=1,\dots,S \qquad\qquad \sum_{j=1}^{S} b_j p(c_j) = \int_0^1 p(x)dx \tag{5.23}$$

for every polynomial p of degree less than S. This actually means that the solution is approximated with a polynomial and indeed $\mathbf{b}$ are the Radau quadrature weights rescaled in $[0,1]$ (i.e. divided by 2). We write here the Butcher tableau of the first three members of the family, that can be expressed in a closed form. The one stage method (backward Euler)

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \tag{5.24}$$

the two stages method

$$\begin{array}{c|cc} 1/3 & 5/12 & -1/12 \\ 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4 \end{array} \tag{5.25}$$

the three stages method

$$\begin{array}{c|ccc} \frac{2}{5}-\frac{\sqrt{6}}{10} & \frac{11}{45}-\frac{7\sqrt{6}}{360} & \frac{37}{225}-\frac{169\sqrt{6}}{1800} & -\frac{2}{225}+\frac{\sqrt{6}}{75} \\ \frac{2}{5}+\frac{\sqrt{6}}{10} & \frac{37}{225}+\frac{169\sqrt{6}}{1800} & \frac{11}{45}+\frac{7\sqrt{6}}{360} & -\frac{2}{225}-\frac{\sqrt{6}}{75} \\ 1 & \frac{4}{9}-\frac{\sqrt{6}}{36} & \frac{4}{9}+\frac{\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{4}{9}-\frac{\sqrt{6}}{36} & \frac{4}{9}+\frac{\sqrt{6}}{36} & \frac{1}{9} \end{array} \tag{5.26}$$

The other members of the family instead can only be computed numerically.
The order of a S stages Radau IIA is $2S-1$, because it is based on a Gaussian-like quadrature. In practice however, if the solution is not very well behaved, these methods often suffer of order reduction and become of order S.
We have seen that Radau IIA is a collocation method, and for this reason we have immediately the values of the function and its derivative at the discretisation points

$$\frac{\partial y}{\partial t}(t_n + c_i h) = k_i \tag{5.27}$$

$$y(t_n + c_i h) = y_n + h\sum_{j=1}^{S} a_{ij}k_j. \tag{5.28}$$

Therefore the values of $\frac{\partial y}{\partial t}$ at for value of time in the interval $[t_n, t_n+h]$ can be computed interpolating $\frac{\partial y}{\partial t}(t_n+c_ih)$ and the values of y taking the integral of the polynomial.
We underline that the order of the solver cannot in any case be $2S-1$ inside the interval, because the order gain in Gaussian quadrature happens only when performing the integral on the whole interval and anyway the order of the interpolation would still be S. Indeed in our solver we will

need to know accurately the values of the crack length for any possible value of time for the computation of Carter leak-off, and to do so we will have to interpolate the temporal discretisation. This means that the order of the approximation of the leak-off is anyway S, therefore in the following we will simply consider an S stage Radau IIA method as having order S.

The main disadvantage of using a fully implicit Runge Kutta method like Gauss Radau IIA is that if S is the number of stages and N is the number of unknowns of the spatial discretisation, then the resulting non-linear system has SN unknowns. Therefore if one for example wants to solve it using Newton's method, it will require the solution of a linear system with SN unknowns, which will take $O(S^3N^3)$ operations with a solver based on LU factorisation, or in alternative, in the best case scenario of fast convergence, $O(S^2N^2)$ operations using an iterative solver. This makes evident that the overall computational complexity grows rapidly when increasing the number of stages S.
Although for simplicity we will just stick to the basic implementation, it is worth to mention that a possible approach to overcome this problem can be found in [76], where instead of solving directly the original linear system with SN unknowns, the solution is approximated through the iterated solution of S linear systems with N unknowns. An alternative way to reach the same goal can be also found in [9], where the Newton iteration is modified and, with the use of a suitable transformation, reverted to the solution of S linear systems with N unknowns.
Directly related to this last approach are SIRK methods [10], where the Runge Kutta matrix A is carefully chosen to simplify the computations. It is also worth to mention DIRK methods [4], less accurate of the previous ones, but of easier implementation. Indeed in this last case the Runge Kutta matrix A is lower triangular and therefore the solution can be directly found solving S non-linear system with N unknowns.

5.4 Carter leak-off

In the case of Carter leak-off we also need to evaluate the function

$$q_l(t,x) = \frac{k_{cl}}{\sqrt{t - L^{-1}(L(t)x)}} = \frac{k_{cl}}{\sqrt{t - t_0(L(t)x)}}. \tag{5.29}$$

To compute $t_0(L)$ we will not invert the approximation of $L(t)$, but instead we will interpolate directly the inverse function $t_0(L)$ itself. However this computation can be affected by numerical problems close to the crack tip and we have already underlined that it is important to have an accurate approximation of the functions on the whole interval. Indeed when x is close to 1 then $t - t_0(L(t)x)$ is close to 0, but t can be arbitrarily large introducing huge numerical error when performing the subtraction. Let us see now what to do to avoid this problem.

We consider the temporal discretisation as seen in the previous section and call t_* the beginning of the actual time step. We consider the nodes $t_i = t_* + hc_i$ from the previous section and define $\Delta_i t = t_i - t$, therefore we have $\Delta_i t_* = hc_i$ and $\Delta_i t_j = h(c_i - c_j)$ that are accurately computed.
For simplicity we describe now only the cases when $L(t_i)x > L(t_*)$, that means we are working on the part of the function $t_0(L)$ currently processed in the new time step. In the opposite case instead, when we deal with the values stored from the previous time steps, the way to proceed remains similar.

We define $\Delta_i L(t) = L(t_i) - L(t)$, therefore from the previous section we have $\Delta_i L(t_j) = L(t_i) - \sum_{k=1}^{S} a_{jk} L'(t_k) = \sum_{k=1}^{S}(a_{ik} - a_{jk})L'(t_k)$ that in the rightmost formulation is accurately computed. Finally we define the function $\Delta_i t_0(\Delta_i L) := t_i - t_0(L(t_i) - \Delta L(t))$ and we approximate it accurately, interpolating the values of the function $\Delta_i t_*, \Delta_i t_1, \dots, \Delta_i t_S$ at the nodes $\Delta_i L(t_*), \Delta_i L(t_1), \dots, \Delta_i L(t_S)$. Finally we can express the leak-off at time t_i as

$$q_l(t_i, x) = \frac{k_{cl}}{\sqrt{t_i - t_0(L(t_i)x)}} = \frac{k_{cl}}{\sqrt{t_i - (t_i - \Delta_i t_0(L(t_i) - L(t_i)x))}} \tag{5.30}$$

$$= \frac{k_{cl}}{\sqrt{\Delta_i t_0(L(t_i)(1-x))}}. \tag{5.31}$$

We note that when we interpolate the inverse function we are working on some nodes that are not predetermined, indeed while the original nodes are the points t_i, in the case of the inverse the nodes become $L(t_i)$. However this should not be a big problem just for the fact that we are not using very high order polynomials for time discretisation and in this case the choice of the nodes is less important.

Nevertheless we remark that the nodes generally used in Gaussian quadrature, in our case Gauss Radau nodes, share the same stability properties of the Chebyshev nodes, because they all tend to cluster at the interval endpoints. This stability property is also preserved when we want to interpolate the inverse function. Indeed if we evaluate an invertible function on nodes that cluster at the endpoints also the values of the function on the same nodes will tend to cluster at the maximum and minimum. These values are actually the interpolation nodes for the inverse function and therefore they are clustered at the endpoints. To be precise this might not be true in general, but it is for sure true if the starting interval is short enough, in which case an invertible function can be approximated as linear. This means that in the worst case it is always possible to have a stable interpolation of the inverse taking a time step short enough.

Long time asymptote

We state here some long time asymptotes for the case of Carter leak-off, that we will use later to test the correctness of our solver. More general results can be found but we will consider here just the case of Newtonian fluid $n = 1$, constant input q_* and for KGD and radial constant toughness K_{IC}.

PKN

In the case of PKN model with constant input, Newtonian fluid and Carter leak-off, it can be seen from [53] that when $t \to \infty$ the solution can be expressed in a closed form. Indeed it is easy to verify that

$$L(t) \sim \frac{2q_*}{\pi k_{cl}} t^{\frac{1}{2}} \tag{5.32}$$

$$q_l(t, x) \sim \frac{k_{cl}}{\sqrt{1 - x^2}} t^{-\frac{1}{2}} \tag{5.33}$$

$$q(x) \sim \frac{2q_*}{\pi} \arcsin(\sqrt{1 - x^2}) \tag{5.34}$$

$$w(t, x) \sim \frac{2\sqrt{q_*}}{\sqrt{\pi} k_{cl}^{\frac{1}{4}}} (\sqrt{1 - x^2} - x \arcsin(\sqrt{1 - x^2}))^{\frac{1}{4}} t^{\frac{1}{8}}. \tag{5.35}$$

KGD

In the same way in the case of KGD model with constant input, constant toughness, Newtonian fluid and Carter leak-off, it is easy to verify that when $t \to \infty$ we have

$$L(t) \sim \frac{2q_*}{\pi k_{cl}} t^{\frac{1}{2}} \tag{5.36}$$

$$q_l(t,x) \sim \frac{k_{cl}}{\sqrt{1-x^2}} t^{-\frac{1}{2}} \tag{5.37}$$

$$q(x) \sim \frac{2q_*}{\pi} \arcsin(\sqrt{1-x^2}) \tag{5.38}$$

$$w(t,x) \sim w_\infty(x) t^{\frac{1}{4}}. \tag{5.39}$$

where this time $w_\infty(x)$ depends on the input parameters and cannot be expressed in a simple form but can be approximated numerically.

Radial

Finally in the case of radial model with constant input, constant toughness, Newtonian fluid and Carter leak-off, it is easy to verify that when $t \to \infty$ we have

$$L(t) \sim 2\sqrt{\frac{q_*}{\pi k_{cl}}} t^{\frac{1}{4}} \tag{5.40}$$

$$q_l(t,x) \sim \frac{k_{cl}}{\sqrt{1-x^4}} t^{-\frac{1}{2}} \tag{5.41}$$

$$q(x) \sim \sqrt{\frac{q_* k_{cl}}{\pi}} \frac{\arcsin \sqrt{1-x^4}}{x} t^{-\frac{1}{4}} \tag{5.42}$$

$$w(t,x) \sim K_{IC} \sqrt{L(t)} (1-x^2)^{\frac{1}{2}}. \tag{5.43}$$

In this case when time goes to infinity the fracture passes to toughness dominated regime.

5.5 Newton solver

To solve the time-dependent problem a generalised version of the fixed point solver used for the self-similar problem could be used [80]. However while it works well with slowly changing boundary conditions, in more difficult situations it turns out to be unstable. To overcome these limitations, we will instead implement a generalised version of the Newton's solver used for the self-similar problem. Therefore, as done previously, we reformulate the problem as a single equation and then use Newton's method to solve the non-linear system that results from the discretisation:

$$F\left(\frac{\partial w}{\partial t}, L'\right) := \begin{pmatrix} F_{\frac{\partial w}{\partial t}}\left(\frac{\partial w}{\partial t}, L'\right) \\ F_{L'}\left(\frac{\partial w}{\partial t}, L'\right) \end{pmatrix} = 0, \tag{5.44}$$

where

$$F_{\frac{\partial w}{\partial t}}\left(\frac{\partial w}{\partial t}, L'\right) := \tag{5.45}$$

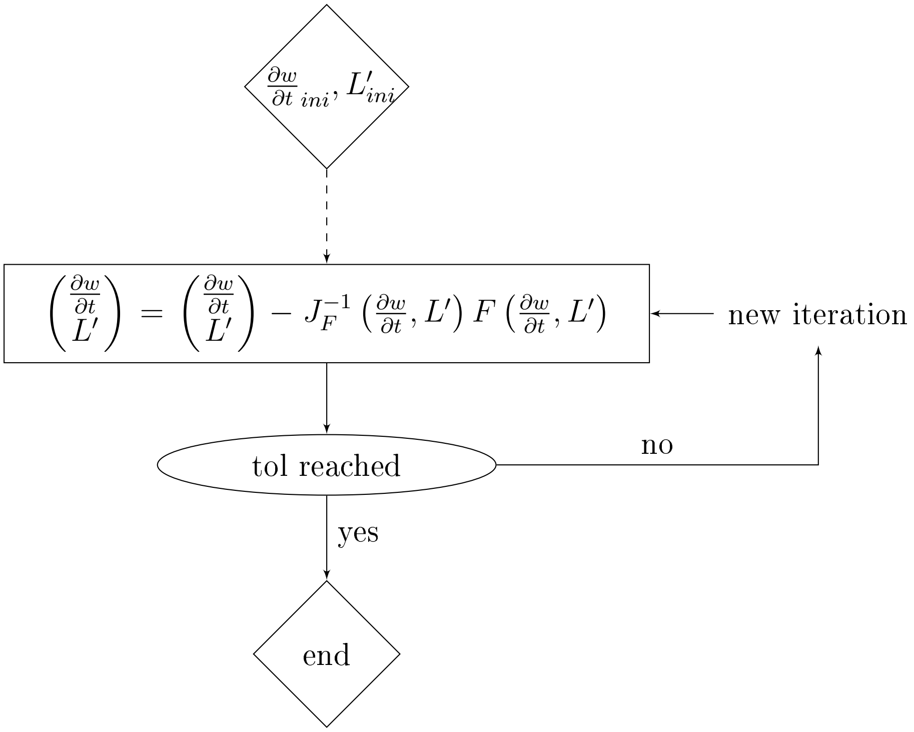
$$\mathcal{W}^{-1}\left(\mathcal{E}\left(\mathcal{F}\left(\mathcal{C}\left(\frac{\partial w}{\partial t}, \mathcal{W}\left(\frac{\partial w}{\partial t}\right), L', \mathcal{L}(L'), \mathcal{Q}(\mathcal{L}(L'))\right), \mathcal{W}\left(\frac{\partial w}{\partial t}\right), \mathcal{L}(L')\right), \mathcal{L}(L')\right)\right) - \frac{\partial w}{\partial t} \tag{5.46}$$

$$F_{L'}\left(\frac{\partial w}{\partial t}, L'\right) := \mathcal{B}\left(\frac{\partial w}{\partial t}, w, L, \mathcal{Q}(\mathcal{L}(L')), q_*\right) - L'. \tag{5.47}$$

We start with an initial guess of the solution, a good choice can be something like

$$\frac{\partial w}{\partial t}_{ini}(t, x) = 0, \quad L'_{ini}(t) = L'(t_*). \tag{5.48}$$

After choosing the initial guess we update $\frac{\partial w}{\partial t}$ and L' using Newton's iteration as in 5.49


$\frac{\partial w}{\partial t}_{ini}, L'_{ini}$
$\begin{pmatrix} \frac{\partial w}{\partial t} \\ L' \end{pmatrix} = \begin{pmatrix} \frac{\partial w}{\partial t} \\ L' \end{pmatrix} - J_F^{-1}\left(\frac{\partial w}{\partial t}, L'\right) F\left(\frac{\partial w}{\partial t}, L'\right)$
new iteration
no
tol reached
yes
end

(5.49)

where we can compute J_F, the Jacobian of F, as a composition of the Jacobians of the functions that constitute F. The cycle is repeated until the error of the solution of the problem goes under a desired tolerance.

To estimate the error we use a criterion similar to the one used in the self-similar case. We call $w_i := \mathcal{W}\left(\frac{\partial w_i}{\partial t}\right)$ and $L_i := \mathcal{L}(L'_i)$ the values of w and L at the i-th iteration and define $\Delta w_i := w_i - w_{i-1}$ and $\Delta L_i := L_i - L_{i-1}$, with in addition $F_{w,i} := \mathcal{W}\left(F_{\frac{\partial w}{\partial t}}\left(\frac{\partial w_i}{\partial t}, L'_i\right)\right)$ and $F_{L,i} := \mathcal{L}\left(F_{L'}\left(\frac{\partial w_i}{\partial t}, L'_i\right)\right)$. This is done because we are not actually interested in having a small error in $\frac{\partial w}{\partial t}$ and L', but instead we just want to control w and L. Finally, as we have done for the self-similar solver, the error at the i-th iteration can be estimated with

$$err_i := \max(err_{w_i}, err_{L_i}, err_{F_{w,i}}, err_{F_{L,i}}), \quad \text{where} \tag{5.50}$$

$$err_{w_i} := \max_t \left(\frac{\|w - w_i\|_\infty}{\|w\|_\infty} \right) \approx \max_t \left(\frac{1}{\|w_i\|_\infty} \frac{\|\Delta w_i\|_\infty^3}{\|\Delta w_{i-1}\|_\infty^2} \right) \tag{5.51}$$

$$err_{L_i} := \max_t \left(\frac{|L - L_i|}{|L|} \right) \approx \max_t \left(\frac{1}{|L_i|} \frac{|\Delta L_i|^3}{|\Delta L_{i-1}|^2} \right) \tag{5.52}$$

$$err_{F_{w,i}} := \max_t \left(\frac{\|F_{w,i}\|_\infty}{\|w\|_\infty} \right) \approx \max_t \left(\frac{\|F_{w,i}\|_\infty}{\|w_i\|_\infty} \right) \tag{5.53}$$

$$err_{F_{L,i}} := \max_t \left(\frac{|F_{L,i}|}{|L|} \right) \approx \max_t \left(\frac{|F_{L,i}|}{|L_i|} \right). \tag{5.54}$$

As we have done for the self-similar problem, we do not ask for a small relative error of the solution at any point of the domain, but just in infinity norm.
After the iterative process has been completed, the pressure p is computed integrating $\frac{\partial p}{\partial x}$ and imposing the appropriate condition.

If we use GMRES to solve the linear system $J_F \left(\frac{\partial w_{i-1}}{\partial t}, L'_{i-1} \right) \begin{pmatrix} \Delta \frac{\partial w_i}{\partial t} \\ \Delta L'_i \end{pmatrix} = -F \left(\frac{\partial w_{i-1}}{\partial t}, L'_{i-1} \right)$, the iterative process must be repeated until the error goes under a desired tolerance. Therefore the error at the j-th GMRES iteration of the i-th Newton's iteration can be estimated as

$$err_{i,j} := \max(err_{w,i,j}, err_{L,i,j}), \quad \text{where} \tag{5.55}$$

$$err_{w,i,j} = \max_t \left(\frac{\left\| J_{F^{asy}_{\frac{\partial w}{\partial t}}} (\frac{\partial w_{i-1}}{\partial t}, L'_{i-1}) \begin{pmatrix} \Delta \frac{\partial w_{i,j}}{\partial t} \\ \Delta L'_{i,j} \end{pmatrix} + F^{asy}_{\frac{\partial w}{\partial t}} (\frac{\partial w_i}{\partial t}_{i-1}, L'_{i-1}) \right\|_\infty}{\|w^{asy}\|_\infty} \right) \tag{5.56}$$

$$\approx \max_t \left(\frac{\left\| J_{F^{asy}_{\frac{\partial w}{\partial t}}} (\frac{\partial w_{i-1}}{\partial t}, L'_{i-1}) \begin{pmatrix} \Delta \frac{\partial w_{i,j}}{\partial t} \\ \Delta L'_{i,j} \end{pmatrix} + F^{asy}_{\frac{\partial w}{\partial t}} (\frac{\partial w_i}{\partial t}_{i-1}, L'_{i-1}) \right\|_\infty}{\|w^{asy}_{i-1}\|_\infty} \right) \tag{5.57}$$

$$err_{L,i,j} = \max_t \left(\frac{\left| J_{F_L} (\frac{\partial w_{i-1}}{\partial t}, L'_{i-1}) \begin{pmatrix} \Delta \frac{\partial w_{i,j}}{\partial t} \\ \Delta L'_{i,j} \end{pmatrix} + F_L (\frac{\partial w_i}{\partial t}_{i-1}, L'_{i-1}) \right|}{|L|} \right) \tag{5.58}$$

$$\approx \max_t \left(\frac{\left| J_{F_L} (\frac{\partial w_{i-1}}{\partial t}, L'_{i-1}) \begin{pmatrix} \Delta \frac{\partial w_{i,j}}{\partial t} \\ \Delta L'_{i,j} \end{pmatrix} + F_L (\frac{\partial w_i}{\partial t}_{i-1}, L'_{i-1}) \right|}{|L_{i-1}|} \right), \tag{5.59}$$

where

$$\frac{\partial w^{asy}}{\partial t} = \frac{\frac{\partial w}{\partial t}}{(1-x)^\alpha}, \quad F^{asy}_{\frac{\partial w}{\partial t}} = \frac{F_{\frac{\partial w}{\partial t}}}{(1-x)^\alpha}, \quad \alpha = \begin{cases} \frac{1}{n+2} & \text{if PKN and no leak-off} \\ 0 & \text{if PKN and Carter leak-off} \\ \frac{1}{2} & \text{if KGD or radial.} \end{cases} \tag{5.60}$$

We note that when we have built $F_{\frac{\partial w}{\partial t}}$, at the end we have applied $\mathcal{W}^{-1}$ and then subtracted $\frac{\partial w}{\partial t}$. However we could have simply defined a function F_w, stopping right after the elasticity equation and then subtracting w instead. Yet the reason for which we decided to do this additional step and use $F_{\frac{\partial w}{\partial t}}$ is because it offers better performances in the application of GMRES reducing the number of iterations.

Computational complexity

As we have done for the self-similar problem, if N is the number of nodes of the spatial discretisation and S the number of Runge Kutta stages, we can evaluate the computational complexity of the time-dependent solver. As for the self-similar problem, also this time we can precompute the integral operator and store it in form of a matrix. In this case the precomputation will be even more beneficial considering that we can reuse it at every time step.
There are two main differences between the self-similar and the time-dependent problem in the evaluation of the function F. The first is that in the time-dependent problem we need to perform integration/differentiation in time ($O(S^2N)$). The second is that all the other operations must be repeated for all S stages. Therefore, recalling what we have said for the self-similar case, for time-dependent problem the computational complexity of one evaluation of F is $O(\max(S^2N, SN\log(N)))$ for the PKN model and $O(\max(S^2N, SN^2))$ for KGD and radial.
As in the case of the self-similar problem also this time the complexity of one evaluation of F is the same of the complexity of the evaluation of the Jacobian J_F in one direction. Using an iterative solver for the linear system like GMRES, the complexity of one GMRES iteration itself is $O(S^2N^2)$ plus the evaluation of the Jacobian in one direction. Therefore if the number of iterations is limited, the overall complexity will be $O(S^2N^2)$ for all the models.

5.6 Adaptive algorithm

For the time-dependent problem we will use the same multigrid approach that we used for the self-similar problem. In addition we will estimate the error coming from the temporal discretisation using together an S stage and an $S+1$ stage Runge Kutta method and comparing the results.

Multigrid method

After choosing the time step length we start working with S stages for the temporal discretisation and proceed with the spatial discretisation like in section 4.5. Therefore we start by solving the problem with $2^3+1=9$ nodes and then, when we have a solution $(\partial w/\partial t, L')^{S,m}$ with 2^m+1 nodes, we use it as initial guess to find a solution $(\partial w/\partial t, L')^{S,m+1}$ with $2^{m+1}+1$ nodes. We repeat the process until we reach an M such that the error coming from the spatial discretisation

$$err_{space} = \max\left(\max_t\left(\frac{\|w^{S,M-1}-w^{S,M}\|_\infty}{\|w^{S,M}\|_\infty}\right), \max_t\left(\frac{|L^{S,M-1}-L^{S,M}|}{|L^{S,M}|}\right)\right) \tag{5.61}$$

goes under the desired tolerance.
Now we take the last result $(\partial w/\partial t, L')^{S,m+1}$, we interpolate it in time and use it as initial guess to solve again the problem on the same time step, with again $2^{M+1}+1$ nodes but this time with

$S+1$ stages. Therefore we get a solution $(\partial w/\partial t, L')^{S+1,m+1}$ and we can estimate the error coming from the temporal discretisation as

$$err_{time} = \max\left(\max_t \left(\frac{\|w^{S,M+1} - w^{S+1,M+1}\|_\infty}{\|w^{S+1,M+1}\|_\infty} \right), \max_t \left(\frac{|L^{S,M+1} - L^{S+1,M+1}|}{|L^{S+1,M+1}|} \right) \right). \tag{5.62}$$

If the error is above the tolerance the solution is rejected and we try again with a shorter time step, otherwise the solution is accepted and we can proceed to the next time step.
We note that in general it is possible to embed an error estimate in Runge Kutta methods and this is true also in our case, see for example [35]. However if we run an embedded $S+1$ stage method, we have to solve the whole problem using $S+1$ stages, but anyway the error estimate is equivalent to the result of an S stage method. Instead with our approach first we solve the whole problem with S stages and then we only need few iterations with $S+1$ stages because we are using the S stage solution as an initial guess. Therefore the big part of the job is done with S stages and the computational cost is reduced.

Time step choice

After we have solved the problem for a given time step, we need to choose the length of the following one, we can proceed as suggested in [35]. In our case to estimate the error we are using an S stage Runge Kutta method that has order S, therefore the error goes like $err = Ch^S$ where h is the time step length.
In case of a rejected time step we can simply estimate the constant as $C = err/h^S$ and in order to get an error close to the tolerance we can choose the new step as

$$h_{new} = \left(\frac{fac \cdot tol}{C} \right)^{\frac{1}{S}} = h \left(\frac{fac \cdot tol}{err} \right)^{\frac{1}{S}}, \tag{5.63}$$

where fac is a safety factor.
In case of an accepted time step instead, if we call C_k the constant at the k-th time step, it is better to assume the the ratio C_k/C_{k-1} remains constant, therefore we get that $C_{k+1} \approx C_k^2/C_{k-1} = err_k^2/err_{k-1}(h_{k-1}/h_k^2)^S$. Then the choice for the new time step is

$$h_{new} = \left(\frac{fac \cdot tol}{C_{k+1}} \right)^{\frac{1}{S}} = \frac{h_k^2}{h_{k-1}} \left(\frac{fac \cdot tol \cdot err_{k-1}}{err_k^2} \right)^{\frac{1}{S}}. \tag{5.64}$$

Following [35], the safety factor fac can be taken equal to about 0.5. Indeed choosing a value of fac smaller than 1 is equivalent to aim for a smaller tolerance than required, that means that the actual error is more likely to stay below it. As a consequence the step is taken a bit smaller, but the risk of step rejection is greatly reduced.

5.7 Comparison with the self-similar solution

To verify the correctness of our solver for the time-dependent problem, we will compare it to the self-similar solver presented in the previous chapter. The spatial discretisation that we use in the self-similar and in the time-dependent solvers is essentially the same and we have already analysed it. For this reason in this section we will only analyse the temporal discretisation and we will not

perform again the tests for several different values of the fluid parameter n and rock toughness K_{IC}. We will simply consider the same self-similar solution already tested in section 4.7 in the case of $n = 1$ (Newtonian fluid) and $K_{IC} = 1$ (for KGD and radial), and solve the corresponding time-dependent problem.
We will set the start time of the simulation as 10^{-5}, while the end time as 10^{5}. We will require an error smaller than 10^{-5} and we will use together the Runge Kutta methods of order 3 and 4 (RK34).

PKN

For the PKN model, the self-similar problem in the case of constant input and $n = 1$ corresponds to $\gamma = \frac{1}{5}$.
– In figure 5.1 we plot the relative error of w and L.
– In figure 5.2 we show the time step length.
We can see that the error does not depend much on x and the reason is that here the error from the temporal discretisation dominates the error from the spatial discretisation. This happens because in the spatial discretisation, when we refine the grid, we double the points of the grid and therefore the higher order solution is much better than the lower order one. On the other hand the same difference is not present in the temporal discretisation where we are using Runge Kutta methods of order 3 and 4, so the error that we are actually seeing is just coming from temporal discretisation. The error remains consistently below the required tolerance of 10^{-5} confirming the good performance of the time step prediction strategy. The time step length increases exponentially in time.

Figure 5.1: Error of the PKN numerical solution with RK34 and required tolerance 10^{-5}.

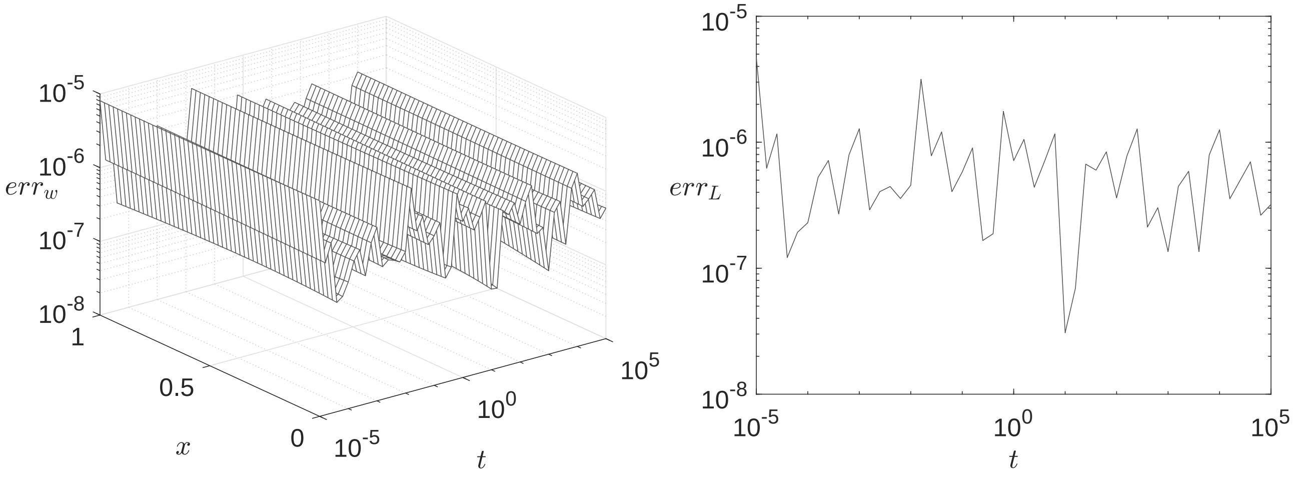
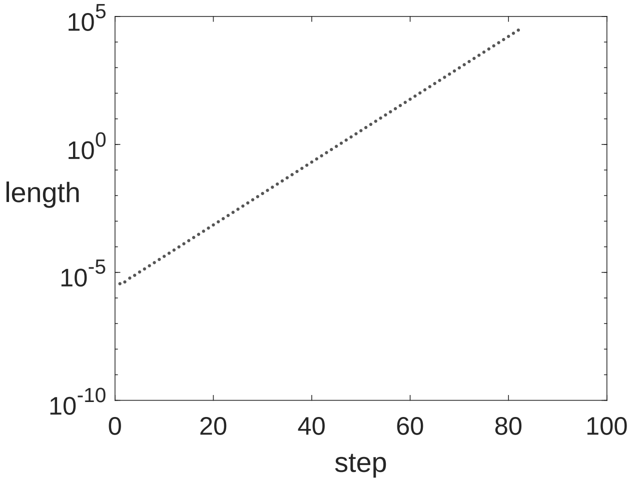


Figure 5.2: Time step length for PKN with RK34 and required tolerance 10^{-5}.



KGD

For the KGD model, the self-similar problem in the case of constant input, $n = 1$ and constant toughness corresponds to $\gamma = \frac{1}{3}$.
– In figure 5.3 we plot the relative error of w and L.
– In figure 5.4 we show the time step length.
As for PKN the error remains consistently below the required tolerance and the time step length increases exponentially in time.

Figure 5.3: Error of the KGD numerical solution with RK34 and required tolerance 10^{-5}.

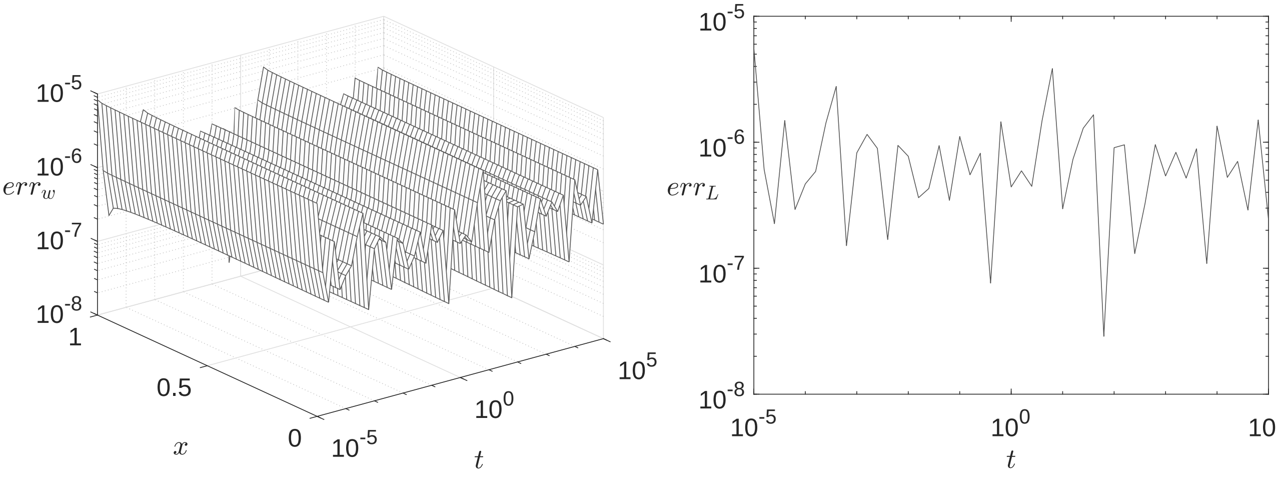
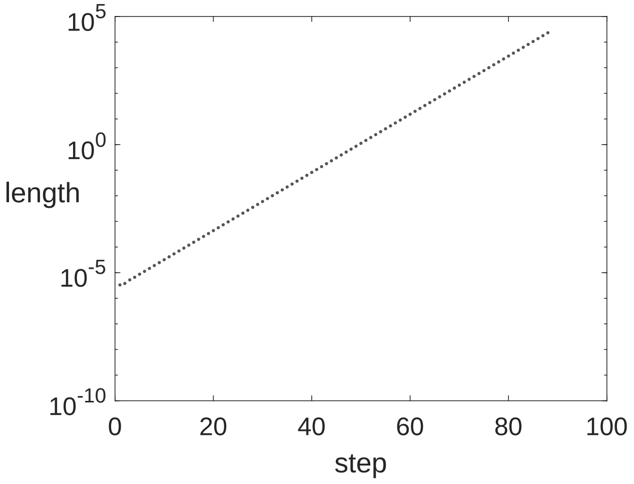


Figure 5.4: Time step length for KGD with RK34 and required tolerance 10^{-5}.



Radial

For the radial model, the self-similar problem in the case of $n = 1$ and constant toughness corresponds to $\gamma = \frac{1}{3}$.
– In figure 5.5 we plot the relative error of w and L.
– In figure 5.6 we show the time step length.
As for PKN and KGD the error remains consistently below the required tolerance and the time step length increases exponentially in time.

Figure 5.5: Error of the radial numerical solution with RK34 and required tolerance 10^{-5}.

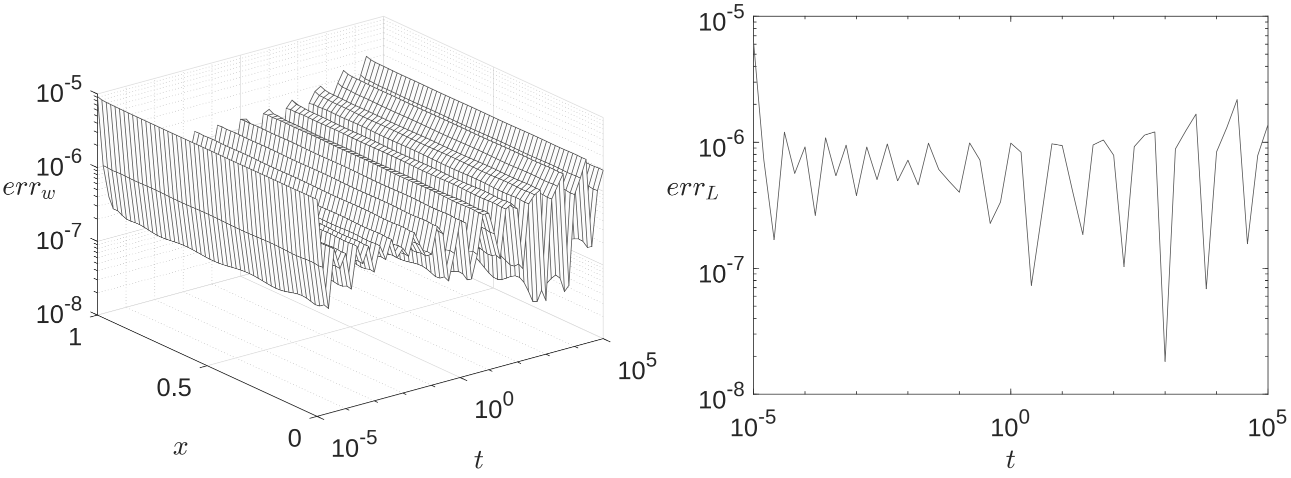
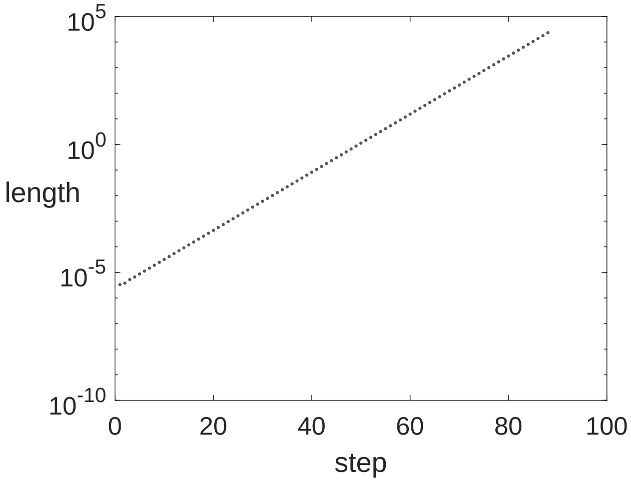


Figure 5.6: Time step length for KGD with RK34 and required tolerance 10^{-5}.



5.8 Convergence analysis

We want now to analyse the convergence properties of the temporal discretisation. We will run the solver using different orders of Runge Kutta and asking for different values of tolerance to see how many time steps are necessary to get the desired solution. We consider the same problems of the previous section and we set the start time of the simulation as 10^{-5}, while instead the end time is 10^{0}.
– In figures 5.7 5.8 5.9 we show the number of time steps necessary to solve the problem with the required tolerance.

Figure 5.7: PKN, number of time steps with start time 10^{-5}, end time 10^{0} and tolerance 10^{-5}.

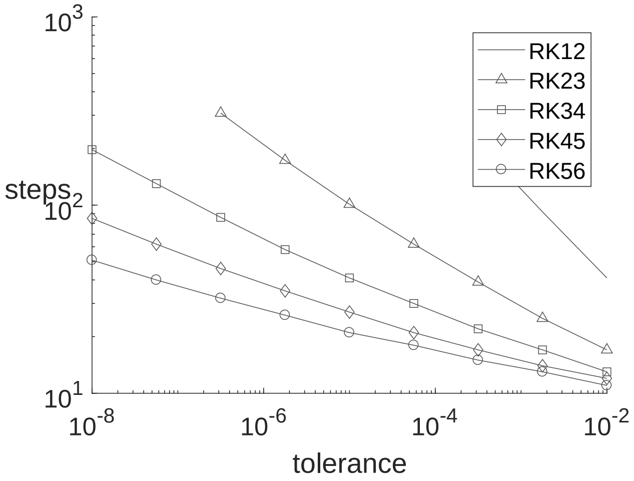


Figure 5.8: KGD, number of time steps with start time 10^{-5}, end time 10^{0} and tolerance 10^{-5}.

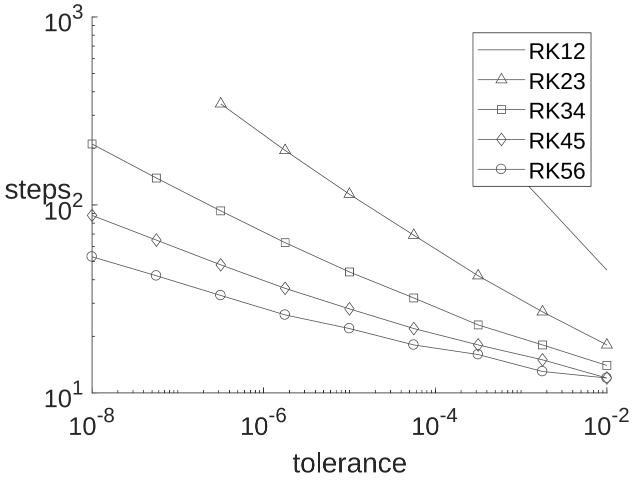
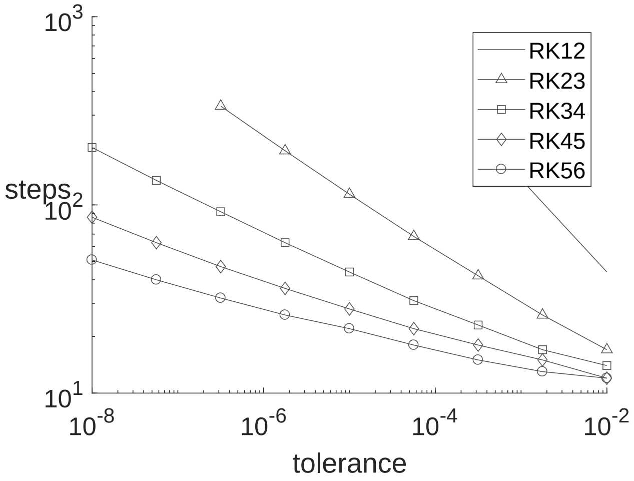


Figure 5.9: Radial, number of time steps with start time 10^{-5}, end time 10^{0} and tolerance 10^{-5}.



It is clear that the higher order methods need far fewer steps when the required tolerance is very small, while the difference is less noticeable if the required tolerance is larger. RK12 can be discarded because it always requires many more time steps compared to the others. On the other hand if we consider the extra cost of an higher order step, and take a reasonable tolerance, also RK45 and RK56 do not seem very interesting. The best choice therefore appear to be between RK23 and RK34, both working well with large tolerance, but RK34 has an edge because of a better performance with smaller tolerances.

5.9 Convergence to long time asymptote with Carter leak-off

To verify the correctness of our solver for Carter leak-off, we test its convergence to the long time asymptotes presented in section 5.4. Indeed we expect the solution to converge to these asymptotes if the end time is large enough. Here we consider the case with $n = 1$, constant input $q_* = 1$ and

for the KGD and radial models constant toughness $K_{IC} = 1$. We start the simulation at $t = 10^{-5}$ from the self-similar solutions already seen in the previous section. We will use the Runge Kutta methods of order 3 and 4 (RK34) with required error smaller than 10^{-5}.
– In figures 5.10 5.11 5.12 we show the relative distance between the numerical solution and the long time asymptote

$$\Delta w := \left| \frac{w - w_{lta}}{w_{lta}} \right|, \quad \Delta L := \left| \frac{L - L_{lta}}{L_{lta}} \right|, \tag{5.65}$$

where w_{lta} and L_{lta} are the long time asymptotes.
As time gets larger, we can see from the plots that the numerical solution indeed converges to the asymptote until the distance reaches the numerical error asked to the solver.

Figure 5.10: Distance between the PKN numerical solution and the long time asymptote with required tolerance 10^{-5}.

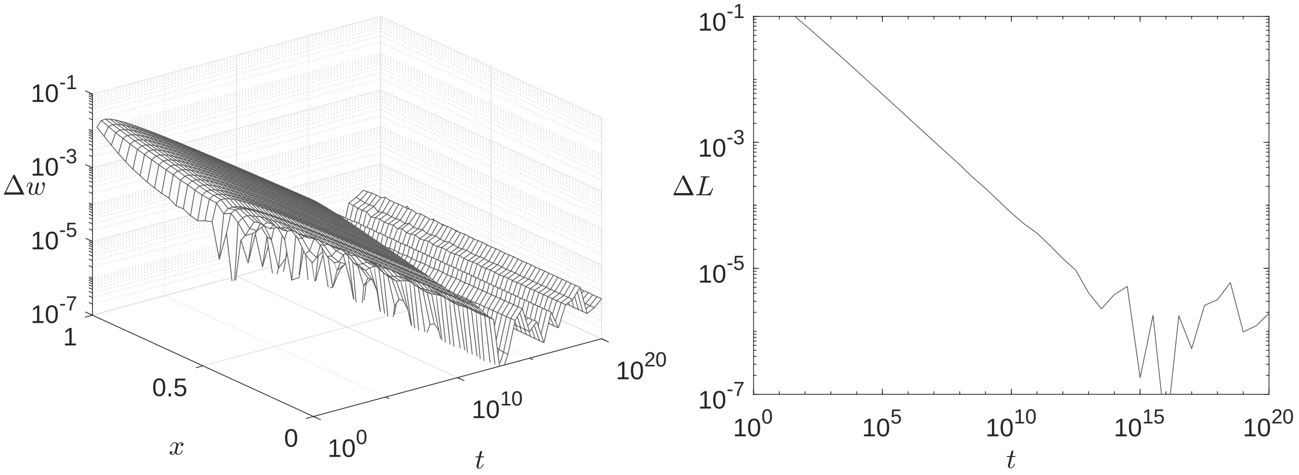


Figure 5.11: Distance between the KGD numerical solution and the long time asymptote with required tolerance 10^{-5}.

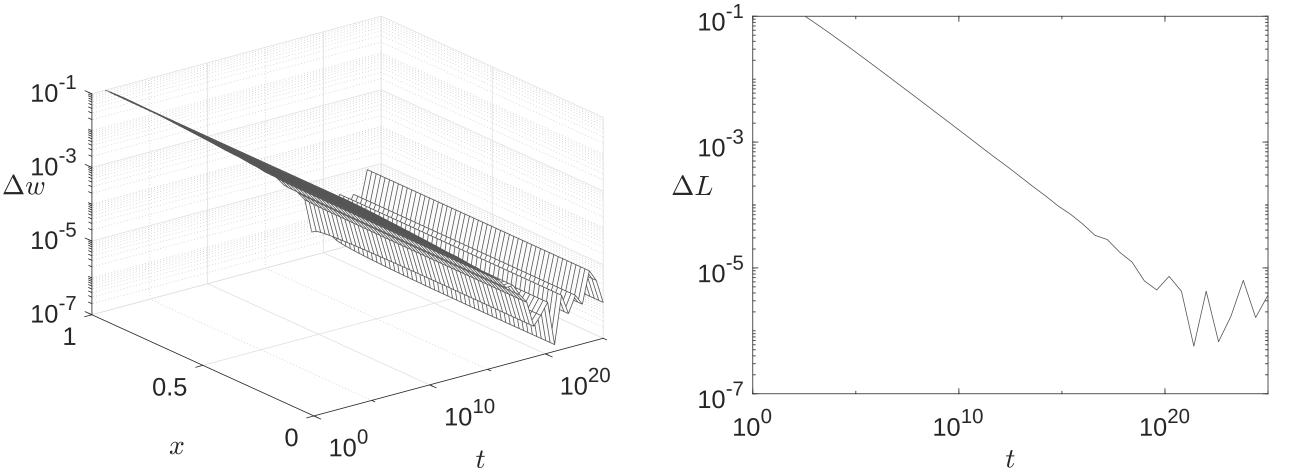
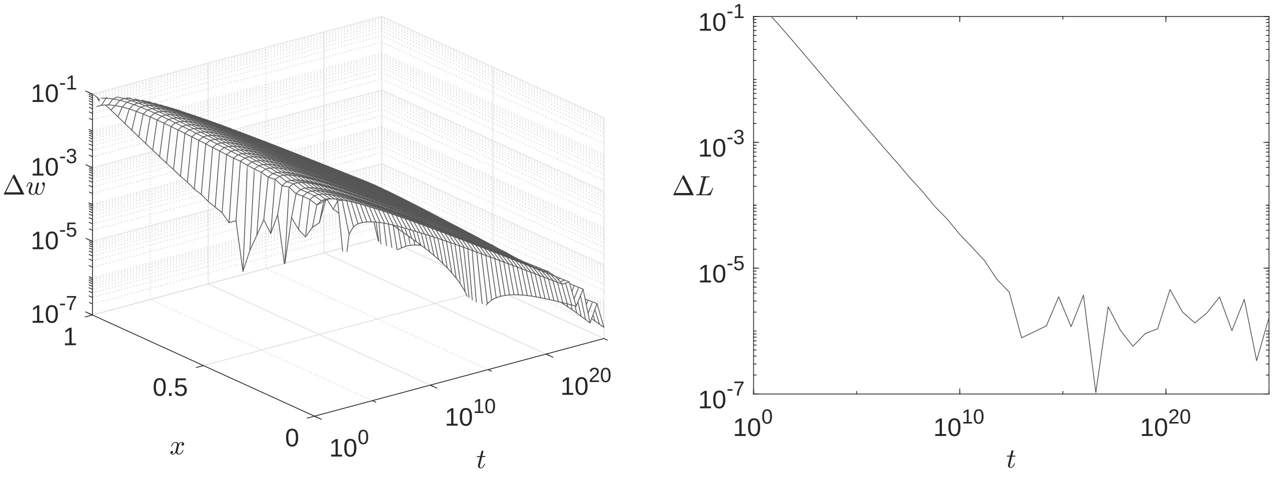


Figure 5.12: Distance between the radial numerical solution and the long time asymptote with required tolerance 10^{-5}.



5.10 Conclusion

In this chapter we have built an effective solver for the HF 1D time-dependent problem.

- In the case of impermeable rocks the solver has been validated comparing the solutions with the self-similar problem previously solved.
- In the case of Carter leak-off the solver has been validated showing its convergence to the long time asymptote.
- The adaptive algorithm performed as expected, keeping the error under the desired tolerance.
- For our purposes the most effective couple of integrators in the Radau IIA Runge Kutta family is that with 3 and 4 stages.
- Thanks to the multi-grid approach only 1 or 2 Newton iterations are required with the finest discretisation.

Chapter 6

Simulations

In this chapter we will run some simulations to investigate the behaviour of HF under some particular boundary conditions. To do so we will set the physical parameters to some real world values, similar to the ones that can be found in [38, 40]:

$$\frac{E}{1-\nu^2} = 3 \cdot 10^{10}\ Pa, \qquad \nu = .25, \qquad K_{IC} = 1.3 \cdot 10^6\ Pa\sqrt{m}, \tag{6.1}$$

$$k_{cl} = 5 \cdot 10^{-5}\ m/\sqrt{s}, \qquad \mu_f = 10^{-3}\ Pa\ s, \qquad Q_* = 6 \cdot 10^{-3}\ m^3/s, \tag{6.2}$$

$$h = 8\ m\ \text{(PKN)}, \qquad h = 30\ m\ \text{(KGD)}, \qquad t_{end} = 10^4 s. \tag{6.3}$$

In the normalised variables these values corresponds approximately to:

$$\qquad\qquad \text{PKN} \qquad\qquad \text{KGD} \qquad\qquad \text{radial} \tag{6.4}$$

$$\tilde{K}_{IC} = \qquad\qquad\qquad\qquad 10^{-4} \qquad\qquad 10^{-4} \tag{6.5}$$

$$\tilde{k}_{cl} = \qquad 2 \cdot 10^{-10} \qquad\qquad 10^{-10} \qquad\qquad 10^{-10} \tag{6.6}$$

$$\tilde{q}_* = \qquad 1.5 \cdot 10^{-15} \qquad\qquad 10^{-16} \qquad\qquad 10^{-15} \tag{6.7}$$

$$\tilde{t}_{end} = \qquad 2.5 \cdot 10^{15} \qquad\qquad 10^{16} \qquad\qquad 10^{16}. \tag{6.8}$$

As for the initial conditions, we will start the simulations from a very small crack, of virtually zero length and zero opening.
We will simulate the following situations:

- pumping rate periodically oscillating in time
- leak-off coefficient periodically oscillating in space
- toughness periodically oscillating in space.

The aim of these simulations, other than to prove the effectiveness of the solver, is to test what is the effect of non-constant physical parameters on fracture propagation. We want to see if there are relevant differences compared to the case with constant parameters or if instead the effect can be neglected. Despite in the real world it is unlikely to have such regular oscillations, studying this simpler case can give an insight on what may actually happen in a situation in which the oscillations can be random.
In the next sections we will make a comparison between the solutions obtained with oscillating parameters and the ones obtained with constant parameters. For this purpose we will consider

the relative distance of crack length, crack propagation speed and crack width at the mouth, the middle and the tip of the fracture

$$\delta l = \left| \frac{l^{osc}(t) - l^{con}(t)}{l^{con}(t)} \right|, \qquad \delta v = \left| \frac{\frac{dl}{dt}^{osc}(t) - \frac{dl}{dt}^{con}(t)}{\frac{dl}{dt}^{con}(t)} \right| \tag{6.9}$$

$$\delta w_{mou} = \left| \frac{w^{osc}(t,0) - w^{con}(t,0)}{w^{con}(t,0)} \right|, \quad \delta w_{mid} = \left| \frac{w^{osc}(t,\frac{l}{2}) - w^{con}(t,\frac{l}{2})}{w^{con}(t,\frac{l}{2})} \right|, \quad \delta w_{tip} = \left| \frac{w^{osc}(t,l) - w^{con}(t,l)}{w^{con}(t,l)} \right|. \tag{6.10}$$

Here f^{osc} stands for a variable taken from the solution of the problem with an oscillating parameter and f^{con} for a variable taken from the solution of the problem with constant parameters.

6.1 Oscillating pumping rate

We consider here the case with oscillating pumping rate and constant toughness (for KGD and radial), with or without leak-off. The oscillating pumping rate is defined by

$$Q_*^{osc}(t) = Q_*(1 + a\sin(2\pi bt)), \tag{6.11}$$

where

$$a = \begin{cases} 1 & \text{if PKN} \\ 1 & \text{if KGD and radial with no leak-off} \\ .5 & \text{ if KGD and radial with carter leak-off} \end{cases} \tag{6.12}$$

$$b = 1 \qquad \text{(period of 1 second).} \tag{6.13}$$

In this way, on the long run, the total amount of fluid pumped is the same as when the pumping rate is constantly Q_*. We chose to consider extreme oscillations with amplitude Q_* or $0.5Q_*$ so that the effects on the fracture are evident as much as possible. In case of KGD and radial models with Carter leak-off we have reduced the amplitude of the oscillations to $0.5Q_*$ to avoid crack stopping, that is not treated by our solver.

PKN

We consider here PKN model with oscillating pumping and no leak-off.
– In figure 6.1 we plot the first few seconds of the simulation, in this way it is possible to see clearly how the oscillating pumping affects the shape and length of the fracture. On the left we plot the evolution in time of the crack opening, on the right we plot crack length and propagation speed for both the oscillating and constant pumping cases.
– In figure 6.2 we plot the relative difference between the oscillating and constant pumping solutions on a longer time interval, in this way it is possible to see the trend as time increases. On the left we plot the crack opening at the mouth, middle and tip of the crack, on the right we plot the crack length and propagation speed.
From the plots it is possible to see that the crack opening oscillates more in the area close to the crack mouth, while the effect is less evident moving near the crack tip. We can also see that the oscillations in crack opening, length and speed decay rapidly in time becoming soon negligible.

Figure 6.1: PKN model, oscillating pumping (amplitude Q_*, period $1s$), no leak-off.
Crack opening (left), crack length and speed (constant pumping case in red) (right).

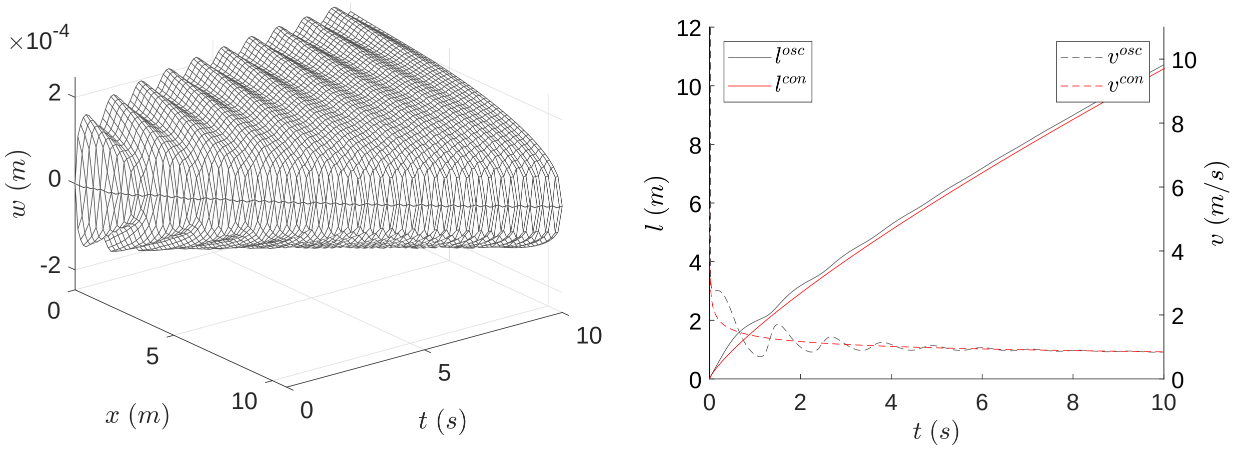
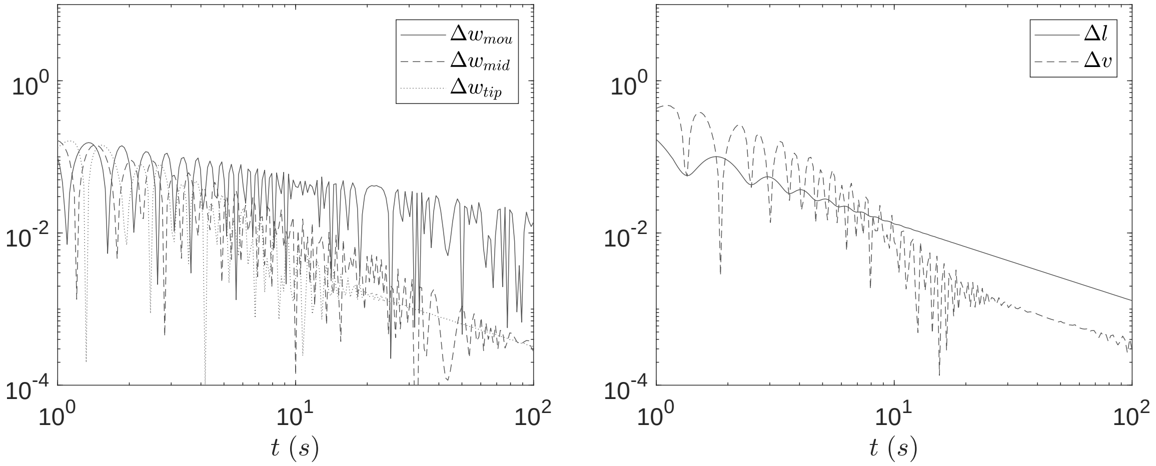


Figure 6.2: PKN model, oscillating pumping (amplitude Q_*, period $1s$), no leak-off.
Comparison with constant pumping case, crack opening (left), crack length and speed (right).



Now we proceed in a similar way in the case with Carter leak-off.
– In figure 6.3 we plot crack opening, length and speed.
– In figure 6.4 we plot the relative differences of crack opening, length and speed.
This time, compared to the case with no leak-off, we can see that the crack opening oscillations are more noticeable also near the crack tip and that the crack speed oscillations are bigger. However, in this case too, the oscillations decay rapidly in time becoming soon negligible.

Figure 6.3: PKN model, oscillating pumping (amplitude Q_*, period $1s$), Carter leak-off. Crack opening (left), crack length and speed (constant pumping case in red) (right).

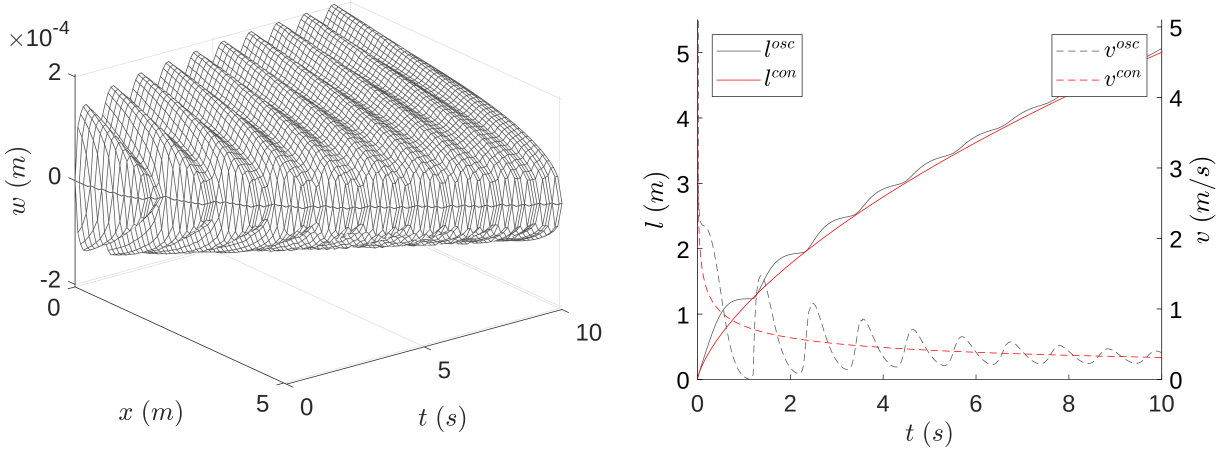
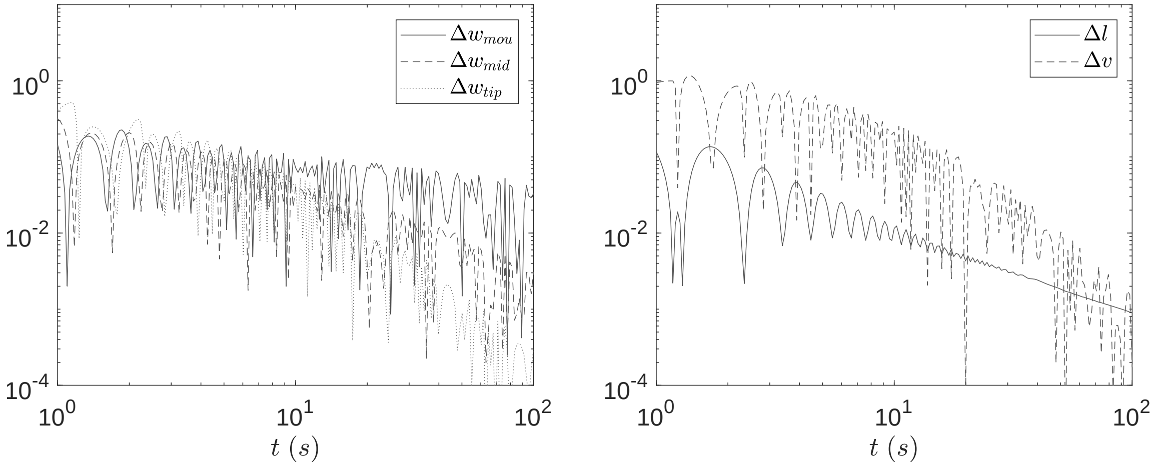


Figure 6.4: PKN model, oscillating pumping (amplitude Q_*, period $1s$), Carter leak-off. Comparison with constant pumping case, crack opening (left), crack length and speed (right).



KGD

We consider here KGD model with oscillating pumping and we proceed as done previously for PKN. In the case with Carter leak-off we have reduced the amplitude of the oscillations to avoid crack stopping.
– In figures 6.5 and 6.6 we plot the case with no leak-off.
– In figures 6.7 and 6.8 we plot the case with Carter leak-off.
Compared to the simulations run with PKN model, this time the crack opening oscillations are more evident also near the crack tip and the crack speed oscillations are stronger. This is related to the fact that in KGD model the elasticity operator is non-local and tends to propagate the effects of the oscillating pumping further away from the crack mouth. However, as seen previously for PKN, also this time the oscillations decay rapidly in time becoming soon negligible.

Figure 6.5: KGD model, oscillating pumping (amplitude Q_*, period $1s$), no leak-off. Crack opening (left), crack length and speed (constant pumping case in red) (right).

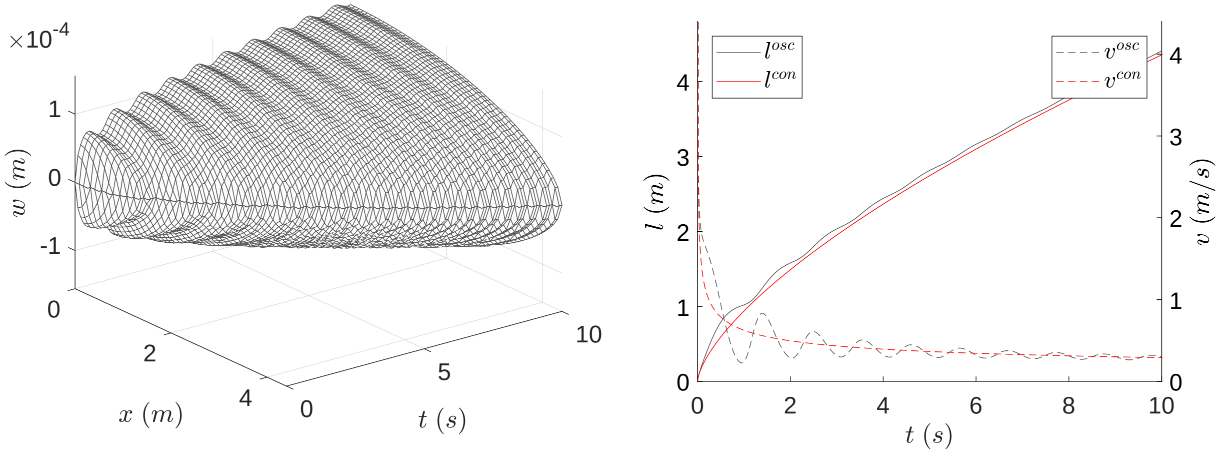


Figure 6.6: KGD model, oscillating pumping (amplitude Q_*, period $1s$), no leak-off. Comparison with constant pumping case, crack opening (left), crack length and speed (right).

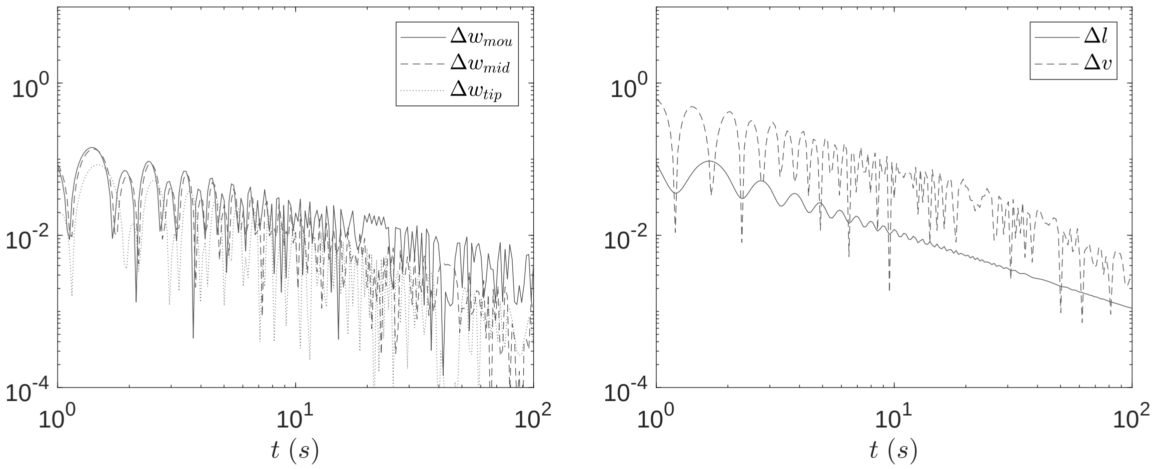


Figure 6.7: KGD model, oscillating pumping (amplitude $0.5Q_*$, period $1s$), Carter leak-off. Crack opening (left), crack length and speed (constant pumping case in red) (right).

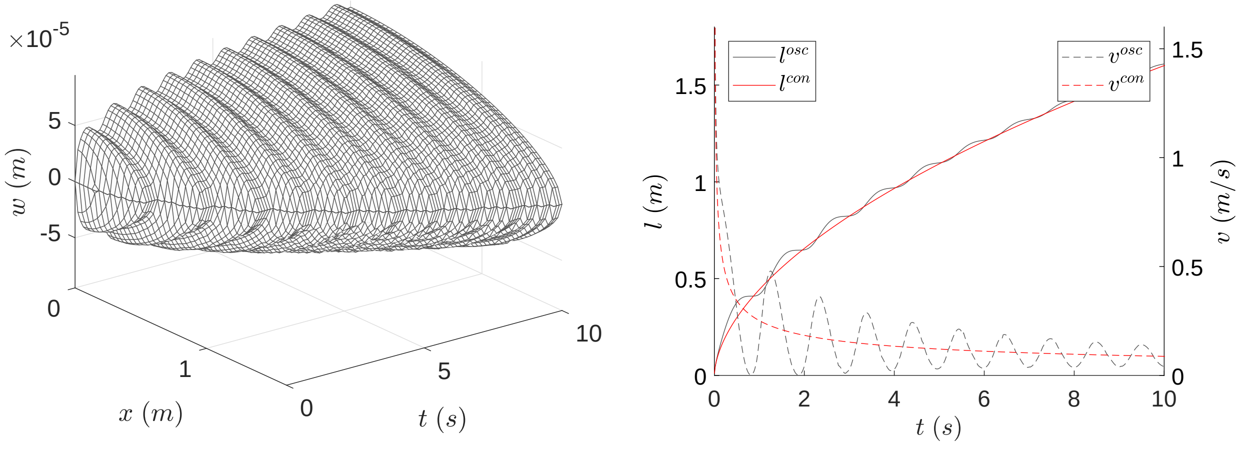
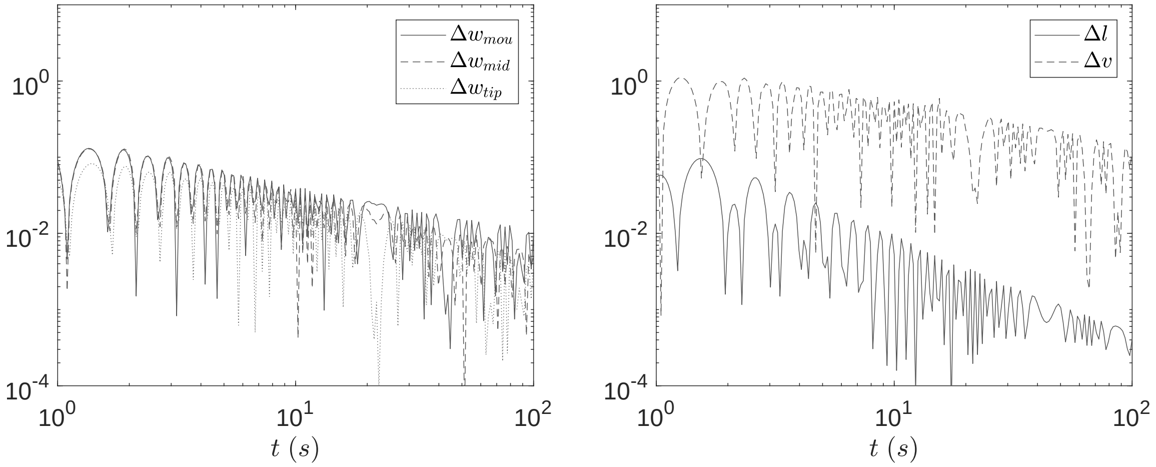


Figure 6.8: KGD model, oscillating pumping (amplitude $0.5Q_*$, period $1s$), Carter leak-off. Comparison with constant pumping case, crack opening (left), crack length and speed (right).



Radial

We consider here radial model with oscillating pumping and we proceed as done previously for PKN and KGD. Also for radial, in the case with Carter leak-off we have reduced the amplitude of the oscillations to avoid crack stopping.
– In figures 6.9 and 6.10 we plot the case with no leak-off.
– In figures 6.11 and 6.12 we plot the case with Carter leak-off.
The results in the case of radial model are qualitatively similar to the ones obtained in the case of KGD. This means that also for radial the oscillations decay rapidly in time becoming soon negligible.

Figure 6.9: Radial model, oscillating pumping (amplitude Q_*, period $1s$), no leak-off. Crack opening (left), crack length and speed (constant pumping case in red) (right).

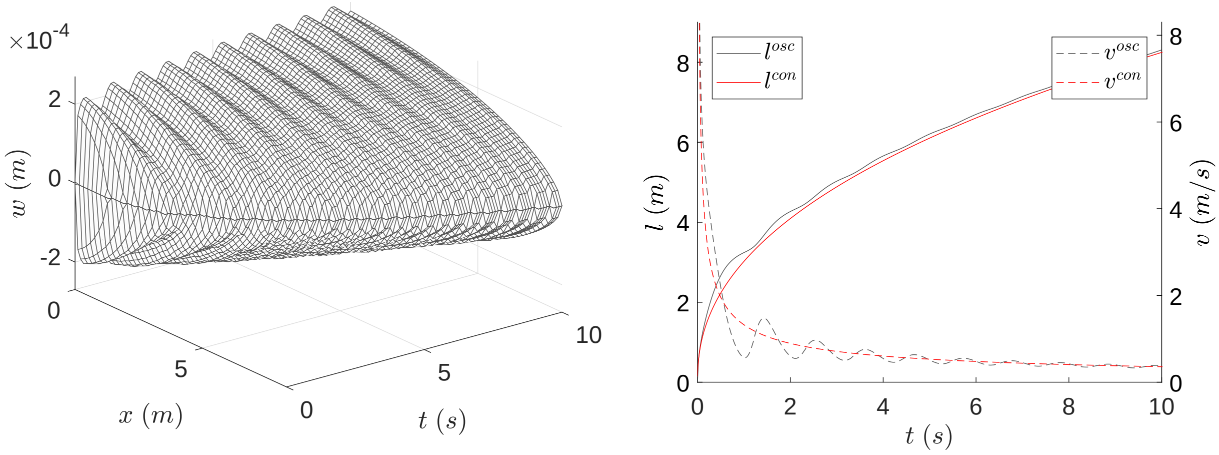


Figure 6.10: Radial model, oscillating pumping (amplitude Q_*, period $1s$), no leak-off. Comparison with constant pumping case, crack opening (left), crack length and speed (right).

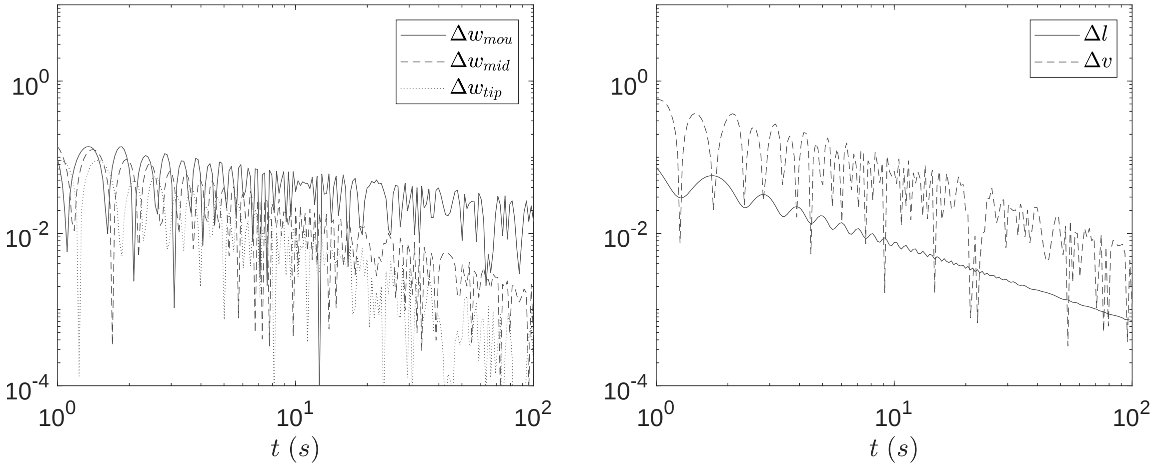


Figure 6.11: Radial model, oscillating pumping (amplitude $0.5Q_*$, period $1s$), Carter leak-off. Crack opening (left), crack length and speed (constant pumping case in red) (right).

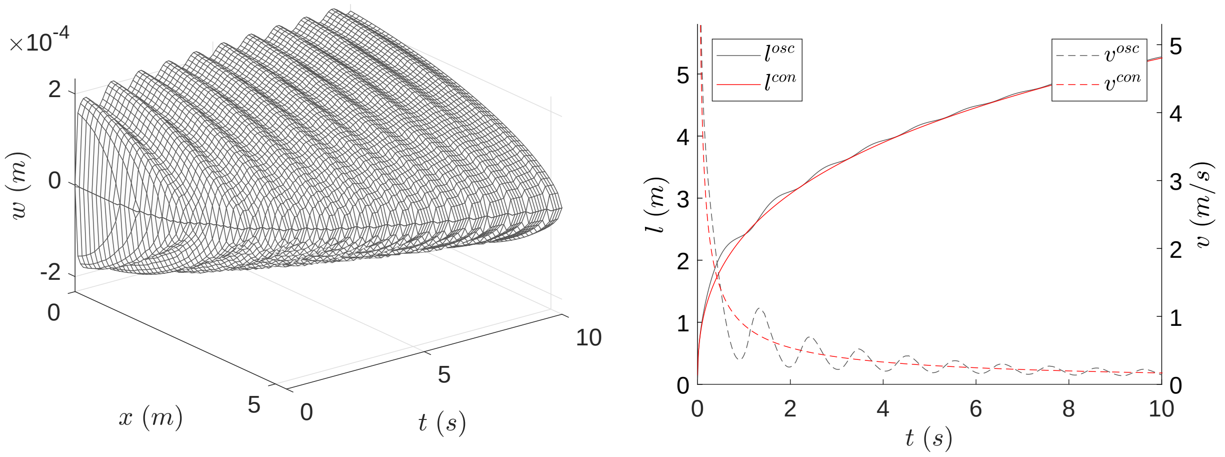
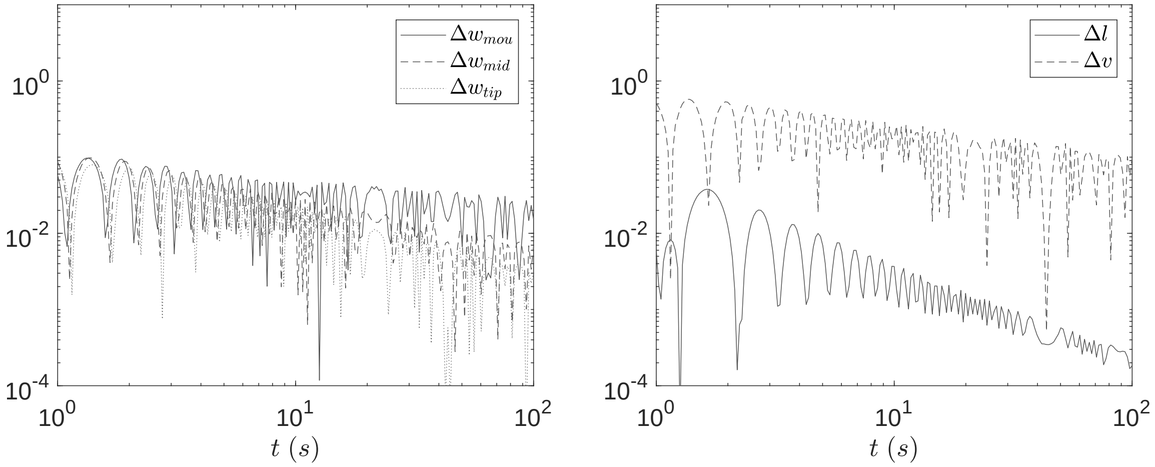


Figure 6.12: Radial model, oscillating pumping (amplitude $0.5Q_*$, period $1s$), Carter leak-off. Comparison with constant pumping case, crack opening (left), crack length and speed (right).



6.2 Oscillating leak-off coefficient

We consider here the case with oscillating leak-off coefficient, constant pumping rate and constant toughness (for KGD and radial). The oscillating leak-off coefficient is defined by

$$k_{cl}^{osc}(x) = k_{cl}(1 + a\sin(2\pi bx)), \tag{6.14}$$

where with x we refer to the original, not normalised coordinate, and

$$a = -1 \tag{6.15}$$

$$b = 1 \qquad \text{(period of 1 meter)}. \tag{6.16}$$

In this way, on a long interval, the average value of k_{cl}^{osc} is exactly k_{cl}. Again we chose to consider extreme oscillations with amplitude k_{cl} so that the effects on the fracture are evident as much as possible.

PKN

We consider here PKN model with oscillating leak-off coefficient and we proceed similarly to what we did previously in the case with oscillating pumping rate.
– In figure 6.13 we plot crack opening length and speed.
– In figure 6.14 we plot the relative differences of crack opening, length and speed.
From the plots it is possible to see that the crack opening oscillations are very small even for small values of time and then decay in time. Differently to the oscillating pumping case, this time the oscillations are more evident near the tip and less pronounced at the mouth. The crack length oscillations are more evident for small values of time, but still decay rapidly becoming soon negligible. The crack propagation speed instead has sharp variations that remain big also as time increases.

Figure 6.13: PKN model, oscillating leak-off (amplitude $-k_{cl}$, period $1m$).
Crack opening (left), crack length and speed (constant leak-off case in red) (right).

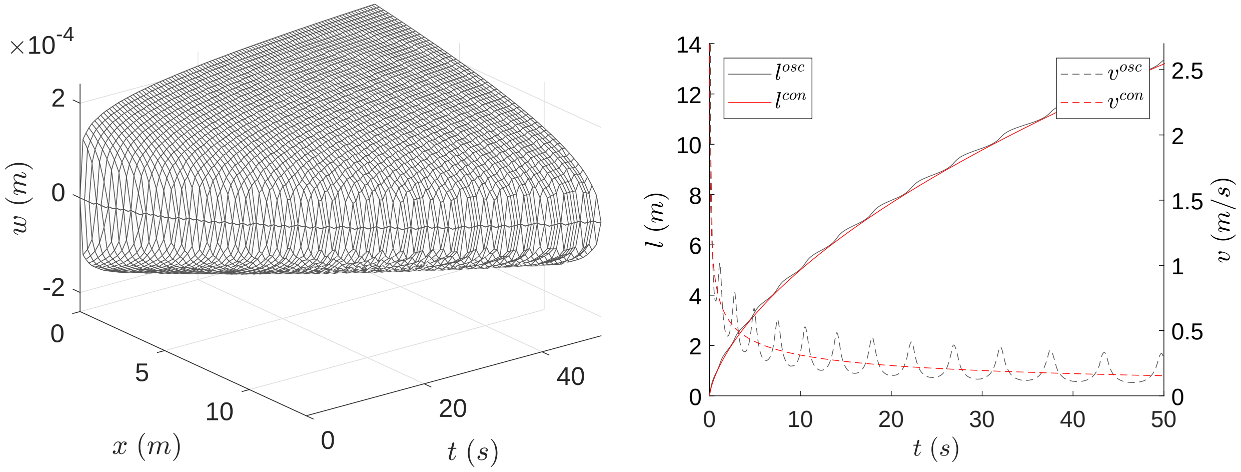
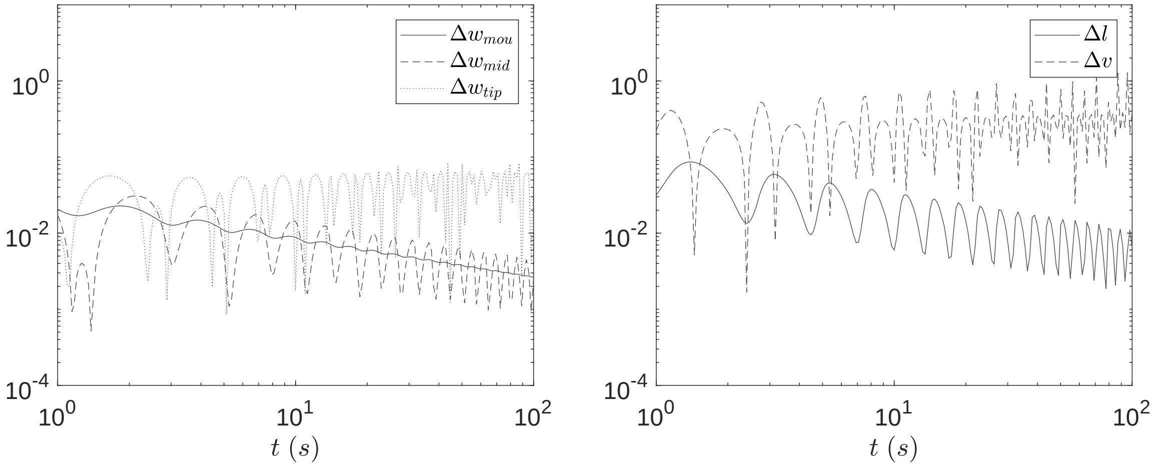


Figure 6.14: PKN model, oscillating leak-off (amplitude $-k_{cl}$, period $1m$).
Comparison with constant leak-off case, crack opening (left), crack length and speed (right).



KGD

We consider here KGD model with oscillating leak-off coefficient and we proceed as done previously for PKN.
– In figures 6.15 and 6.16 we plot the results of the simulation.
Compared to the PKN model, the crack opening oscillations are now more evident also near the crack mouth but still decay in time. The crack length oscillations are also stronger but decay in time as well. The crack propagation speed has sharper variations and again they remain big when time increases.

Figure 6.15: KGD model, oscillating leak-off (amplitude $-k_{cl}$, period $1m$).
Crack opening (left), crack length and speed (constant leak-off case in red) (right).

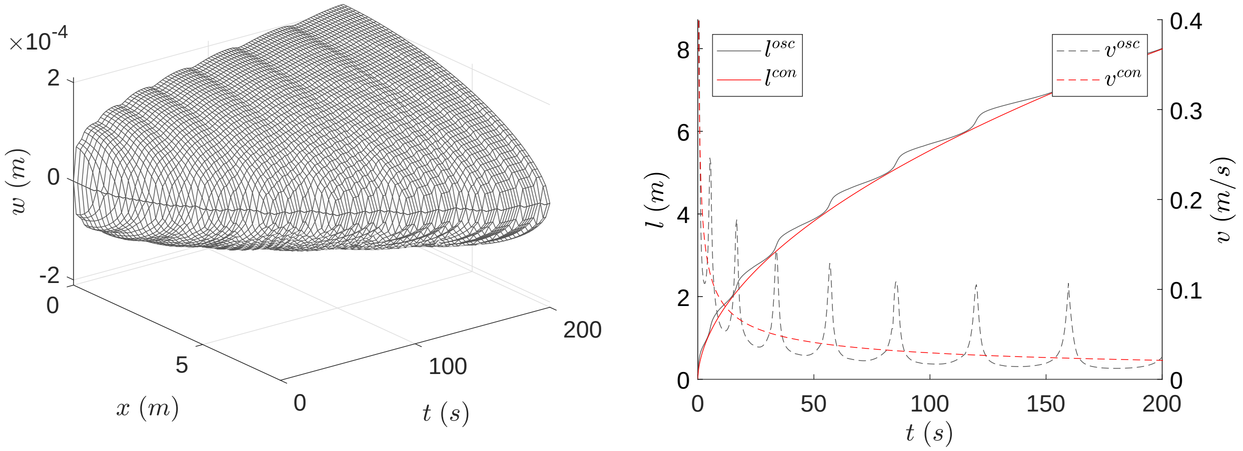
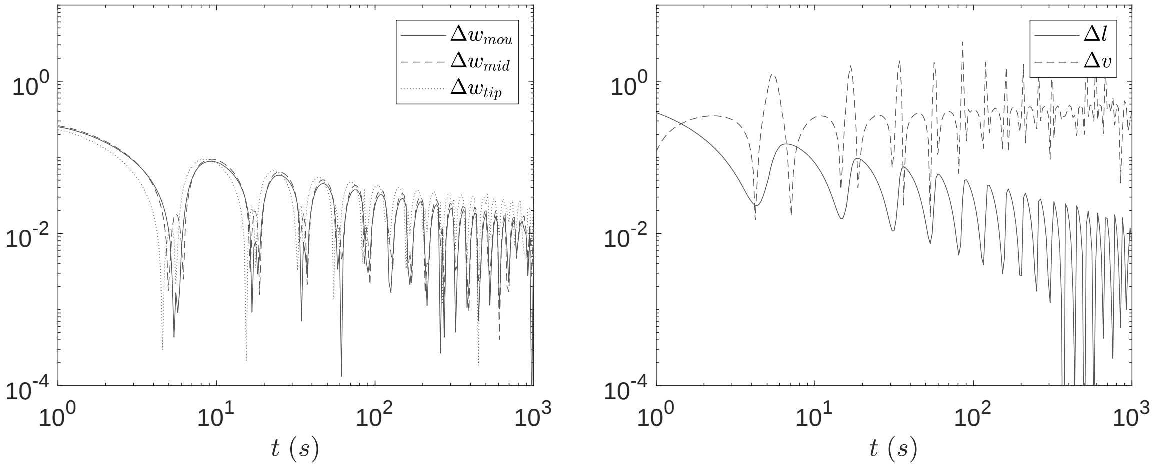


Figure 6.16: KGD model, oscillating leak-off (amplitude $-k_{cl}$, period $1m$).
Comparison with constant leak-off case, crack opening (left), crack length and speed (right).



Radial

We consider here radial model with oscillating leak-off coefficient and we proceed as done previously for PKN and KGD.
– In figures 6.17 and 6.18 we plot the results of the simulation.
This time the crack opening and crack length oscillations are small from the beginning but decay slowly in time. The crack propagation speed has again sharp variations and they do not decay when time increases.

Figure 6.17: Radial model, oscillating leak-off (amplitude $-k_{cl}$, period $1m$).
Crack opening (left), crack length and speed (constant leak-off case in red) (right).

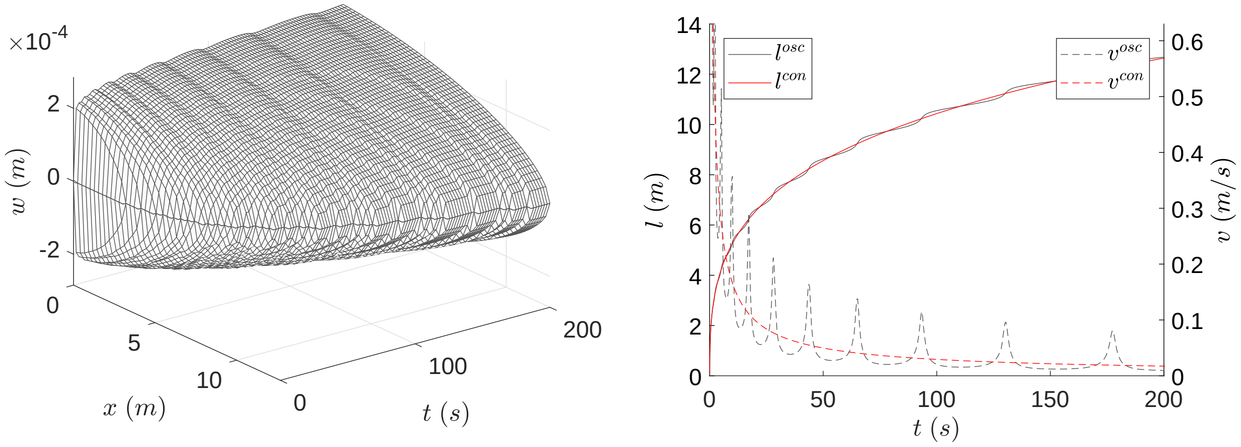
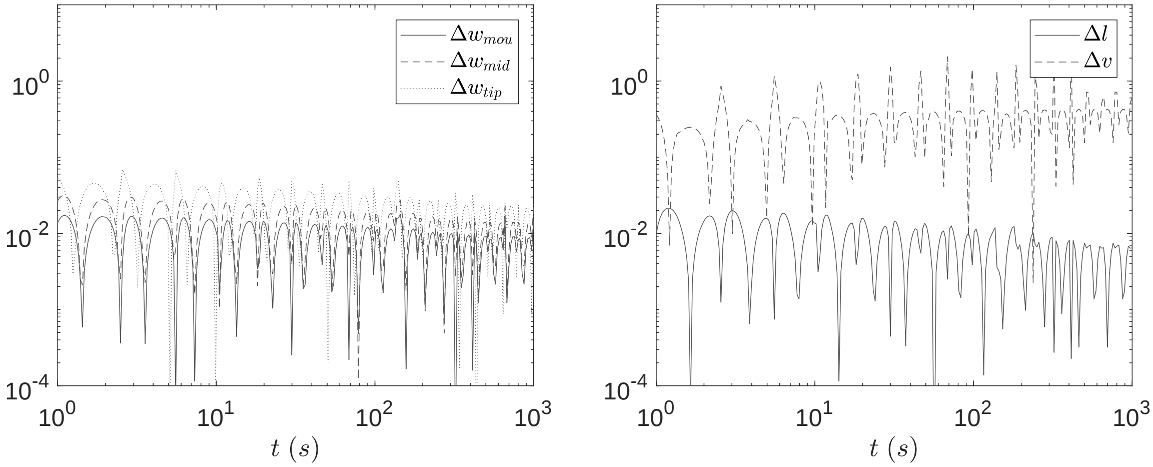


Figure 6.18: Radial model, oscillating leak-off (amplitude $-k_{cl}$, period $1m$).
Comparison with constant leak-off case, crack opening (left), crack length and speed (right).



6.3 Oscillating toughness

We consider here the case with oscillating toughness, constant pumping rate, with or without leak-off. The oscillating toughness is defined by

$$K_{IC}^{osc}(x) = K_{IC}(1 + a\sin(2\pi bx)), \tag{6.17}$$

where with x we refer to the original, not normalised coordinate, and

$$0.1 \leq a \leq 1 \tag{6.18}$$

$$0.1 \leq b \leq 1 \qquad \text{(period of 1 to 10 meters)}. \tag{6.19}$$

We will do the simulations with different values of amplitude and period to see how it affects the results. With K_{IC}^{osc} defined in this way, on a long interval, the average value of K_{IC}^{osc} is exactly K_{IC}.

KGD

We consider here KGD model with oscillating toughness and no leak-off and we proceed similarly to what we did previously in the other cases.
– In figure 6.19 we plot crack opening, length and speed.
– In figures 6.20, 6.21, 6.22 we plot the relative differences of crack opening, length and speed.
From the plots it is possible to see that the crack opening oscillations can be relevant and that they are persistent in time. The same is true also for the crack length oscillations. The crack propagation speed has sharp variations that are persistent in time as well.
In addition, comparing figures 6.20 and 6.21, we note that changing the period of the oscillations does not seem to influence the long time behaviour of the solution. Finally we see that if the amplitude is about $0.1K_{IC}$ or smaller, then the crack opening and length oscillations become negligible. However the crack propagation speed has still sharp variations that persist when time gets larger.

Figure 6.19: KGD model, oscillating toughness (amplitude K_{IC}, period $1m$), no leak-off. Crack opening (left), crack length and speed (constant toughness case in red) (right).

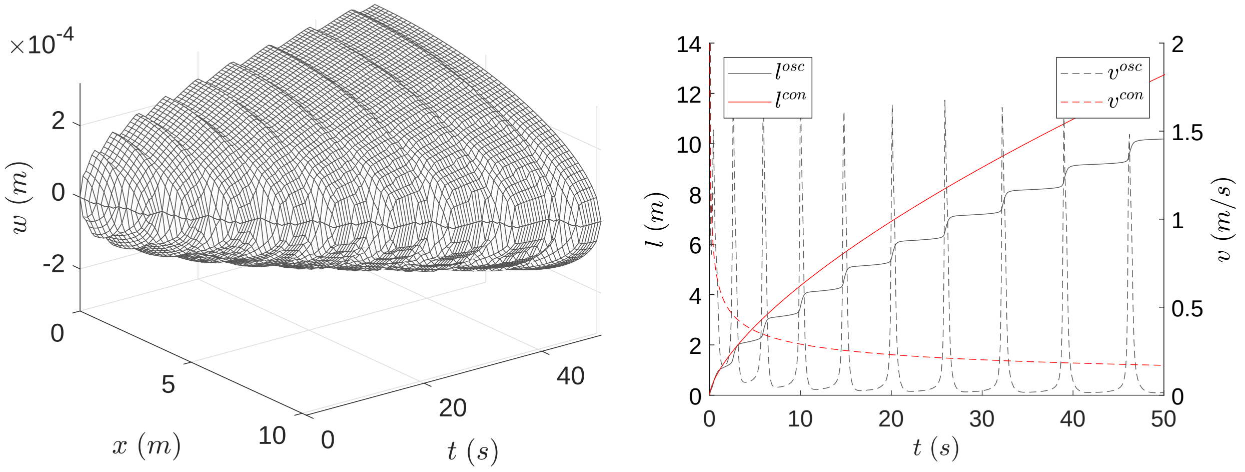


Figure 6.20: KGD model, oscillating toughness (amplitude K_{IC}, period $10m$), no leak-off. Comparison with constant toughness case, crack opening (left), crack length and speed (right).

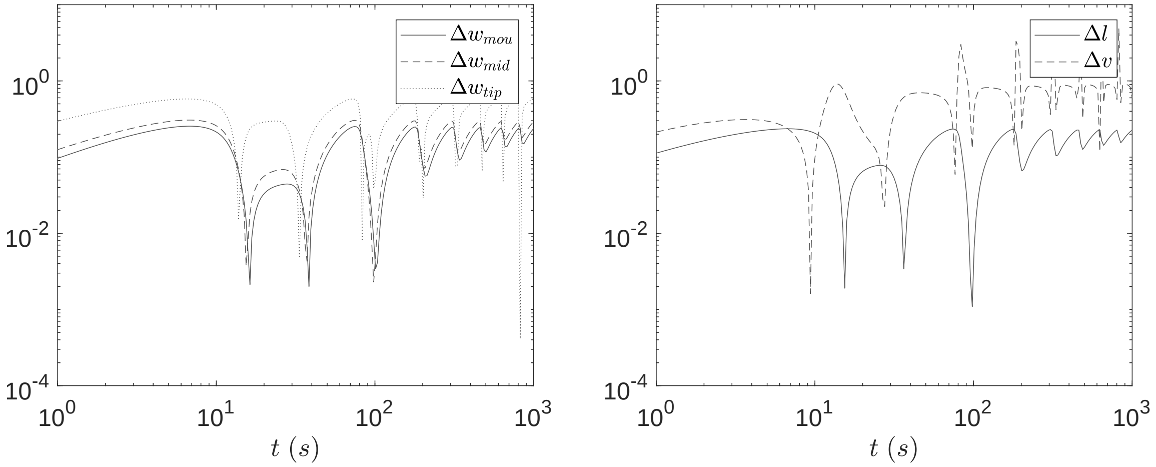


Figure 6.21: KGD model, oscillating toughness (amplitude K_{IC}, period $1m$), no leak-off. Comparison with constant toughness case, crack opening (left), crack length and speed (right).

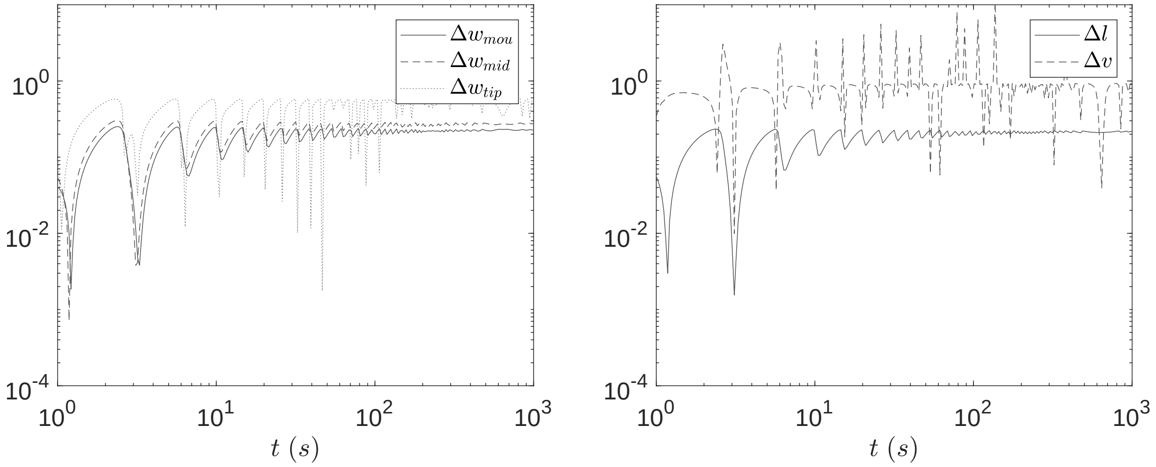
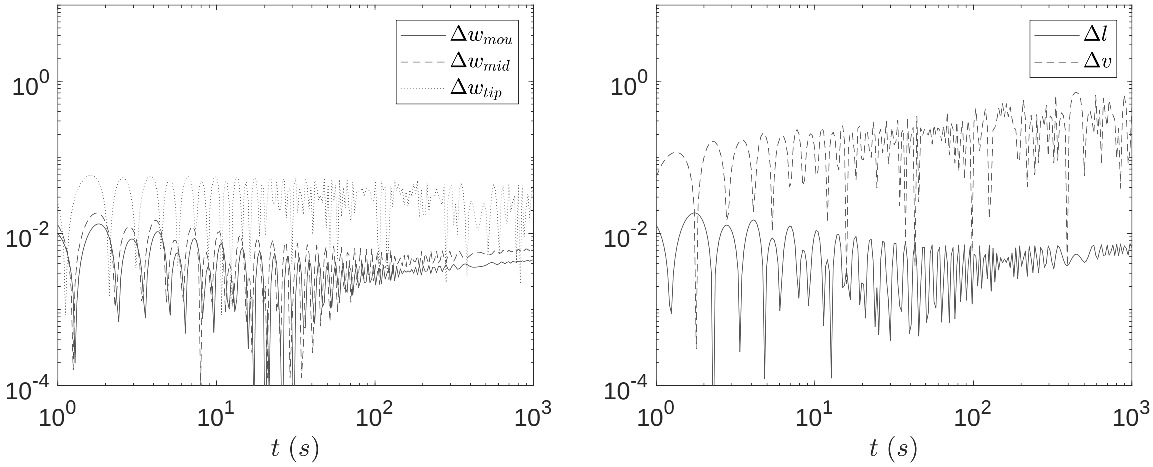


Figure 6.22: KGD model, oscillating toughness (amplitude $0.1K_{IC}$, period $1m$), no leak-off. Comparison with constant toughness case, crack opening (left), crack length and speed (right).



We consider now the case with Carter leak-off, where we reduce the amplitude of the oscillations to avoid crack stopping.
– In figure 6.23 we plot crack opening, length and speed.
– In figures 6.24, 6.25, 6.26 we plot the relative differences of crack opening, length and speed.
We can see how the presence of the leak-off reduces the size of the oscillations compared to the no leak-off case. Indeed we can see that with an amplitude of $0.5K_{IC}$ the oscillations are already small. However, like in the no leak-off case, the oscillations do not decay in time and changing the period of the oscillations does not seem to influence the long time behaviour of the solution. Finally, like in the no leak-off case, the crack propagation speed has also sharp variations that persist in time.

Figure 6.23: KGD model, oscillating toughness (amplitude K_{IC}, period $1m$), Carter leak-off. Crack opening (left), crack length and speed (constant toughness case in red) (right).

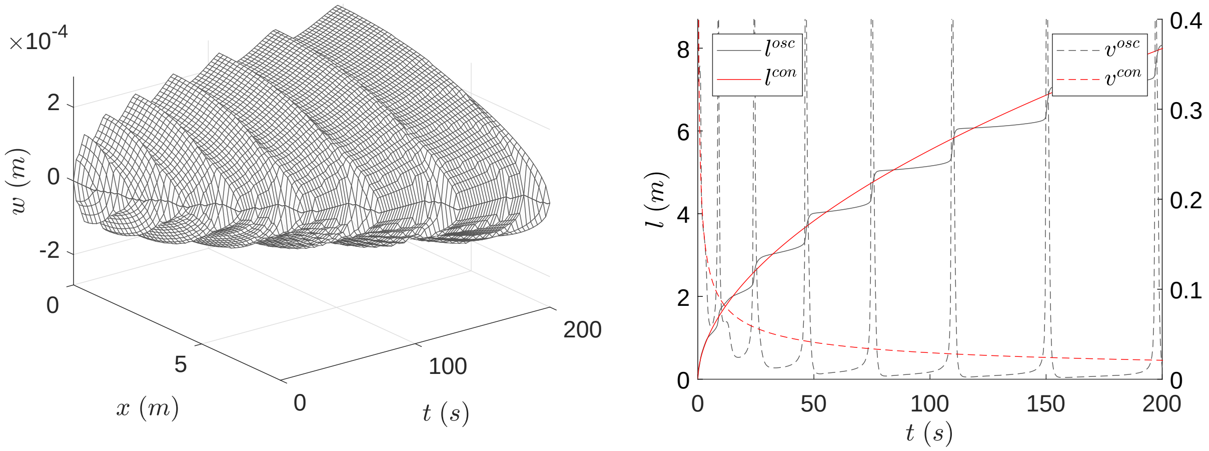


Figure 6.24: KGD model, oscillating toughness (amplitude $0.5K_{IC}$, period $10m$), Carter leak-off. Comparison with constant toughness case, crack opening (left), crack length and speed (right).

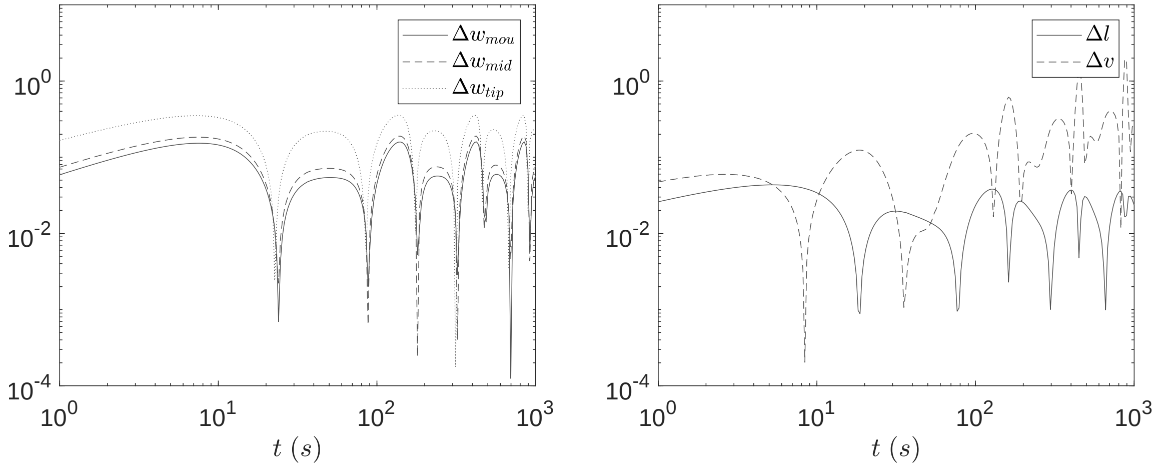


Figure 6.25: KGD model, oscillating toughness (amplitude $0.5K_{IC}$, period $1m$), Carter leak-off. Comparison with constant toughness case, crack opening (left), crack length and speed (right).

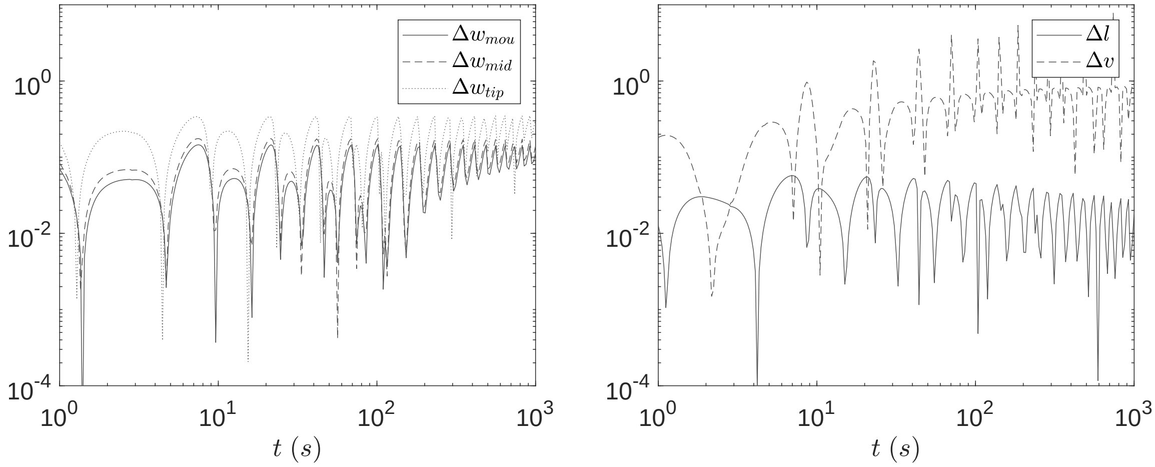
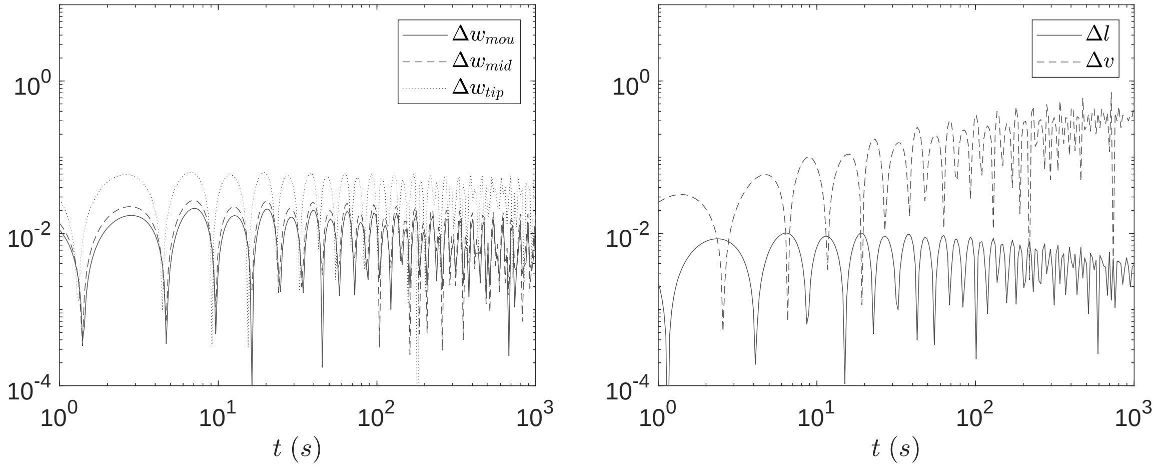


Figure 6.26: KGD model, oscillating toughness (amplitude $0.1K_{IC}$, period $1m$), Carter leak-off. Comparison with constant toughness case, crack opening (left), crack length and speed (right).



Radial

We consider here radial model with oscillating toughness and we proceed as done previously for KGD.

– In figures 6.27, 6.28, 6.29, 6.30 we plot the case with no leak-off.

The results in the case of radial model are qualitatively similar to the ones in the case of KGD. Therefore, also for radial, the crack opening and length oscillations can be relevant and are persistent in time. The crack propagation speed has sharp variations that are also persistent in time. Changing the period of the oscillations does not influence the long time behaviour of the solution and, if the amplitude is about $0.1K_{IC}$ or smaller, then the crack opening and length oscillations are negligible.

– In figures 6.31, 6.32, 6.33, 6.34 we plot the case with Carter leak-off.

Again the presence of the leak-off reduces the size of the oscillations compared to the no leak-off case. If the amplitude is about $0.5K_{IC}$ or smaller, then the crack opening and length oscillations are already small.

Figure 6.27: Radial model, oscillating toughness (amplitude K_{IC}, period $1m$), no leak-off. Crack opening (left), crack length and speed (constant toughness case in red) (right).

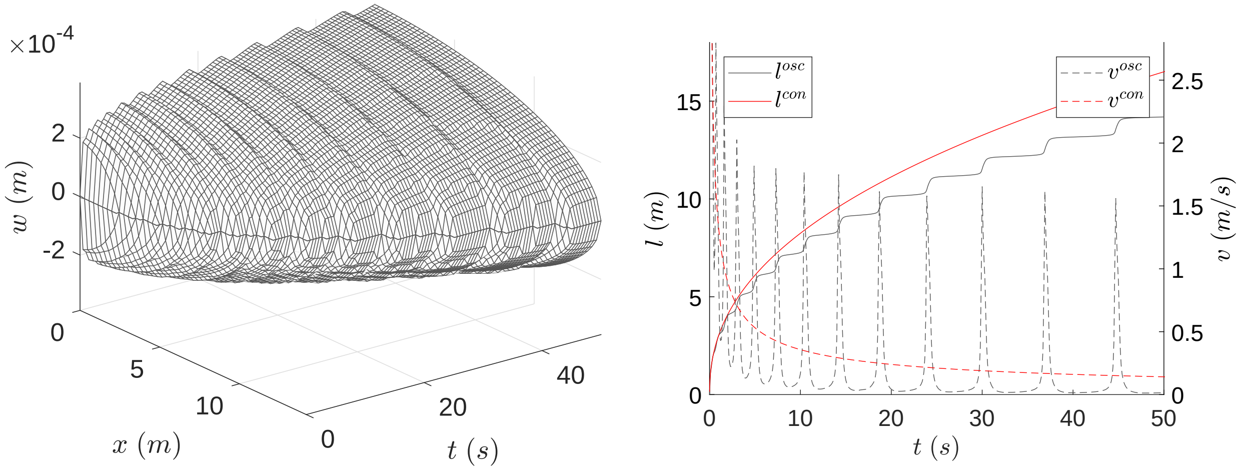


Figure 6.28: Radial model, oscillating toughness (amplitude K_{IC}, period $10m$), no leak-off. Comparison with constant toughness case, crack opening (left), crack length and speed (right).

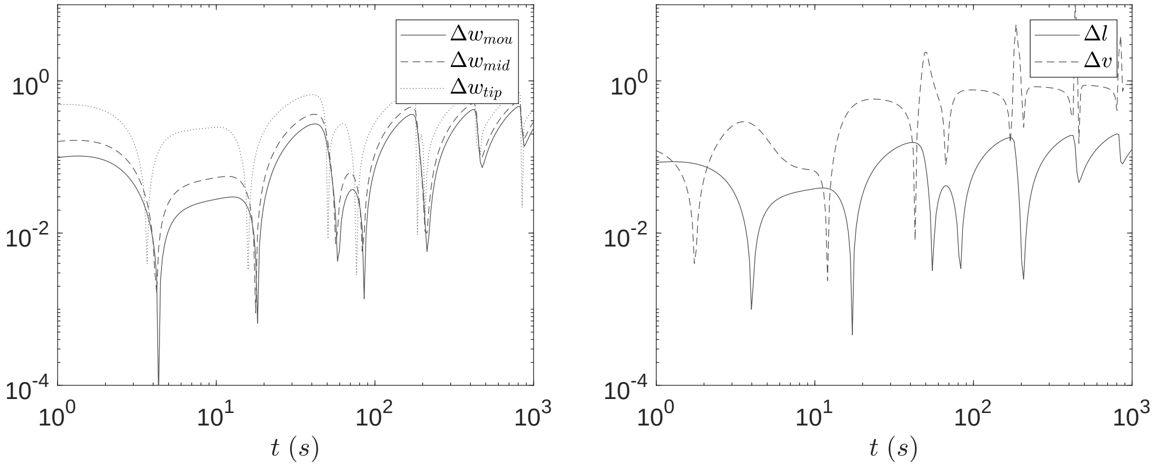


Figure 6.29: Radial model, oscillating toughness (amplitude K_{IC}, period $1m$), no leak-off. Comparison with constant toughness case, crack opening (left), crack length and speed (right).

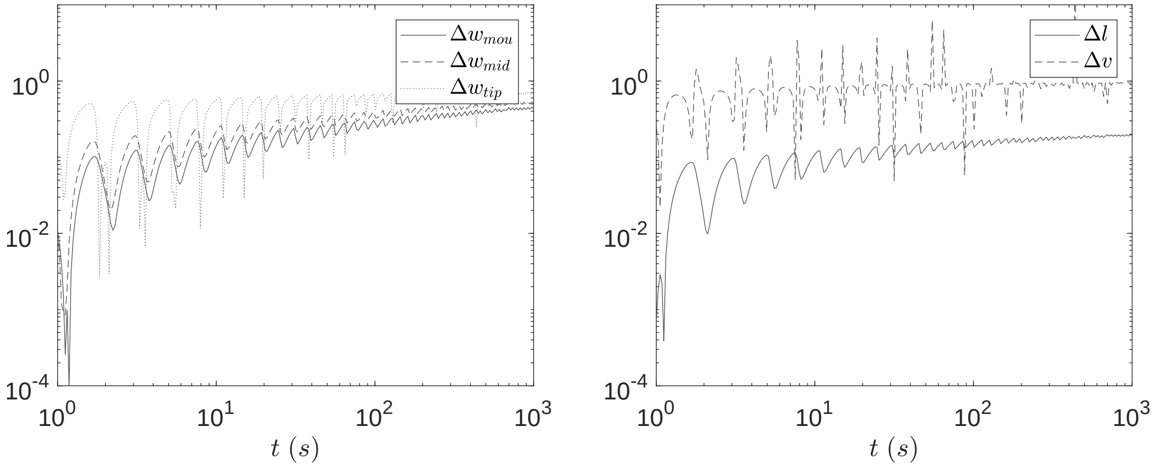


Figure 6.30: Radial model, oscillating toughness (amplitude $0.1K_{IC}$, period $1m$), no leak-off. Comparison with constant toughness case, crack opening (left), crack length and speed (right).

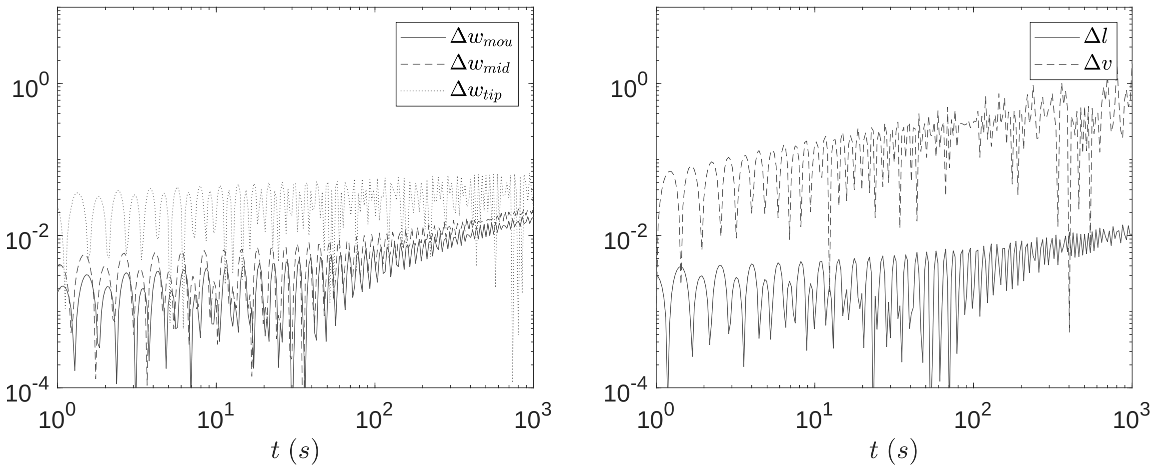


Figure 6.31: Radial model, oscillating toughness (amplitude K_{IC}, period $1m$), Carter leak-off. Crack opening (left), crack length and speed (constant toughness case in red) (right).

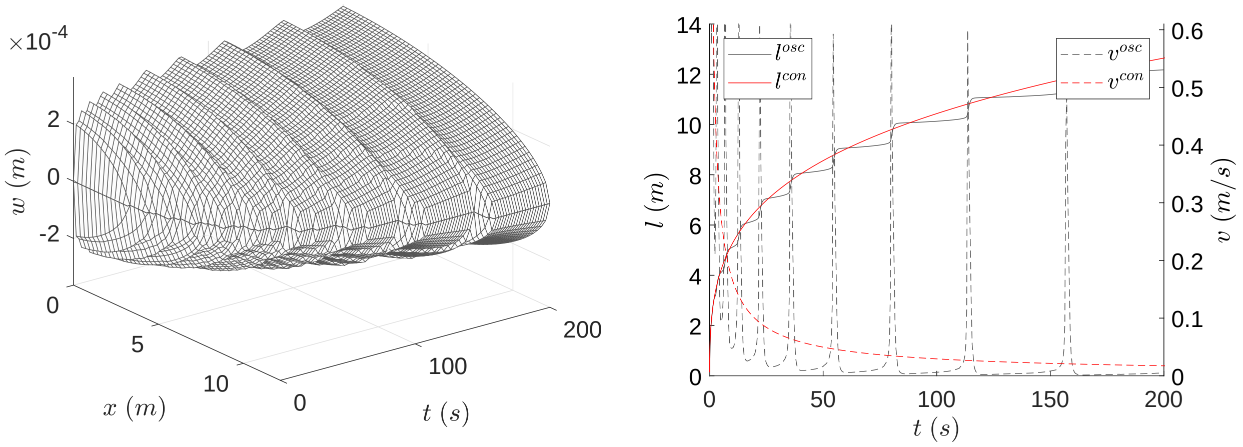


Figure 6.32: Radial model, oscillating toughness (amplitude $0.5K_{IC}$, period $5m$), Carter leak-off. Comparison with constant toughness case, crack opening (left), crack length and speed (right).

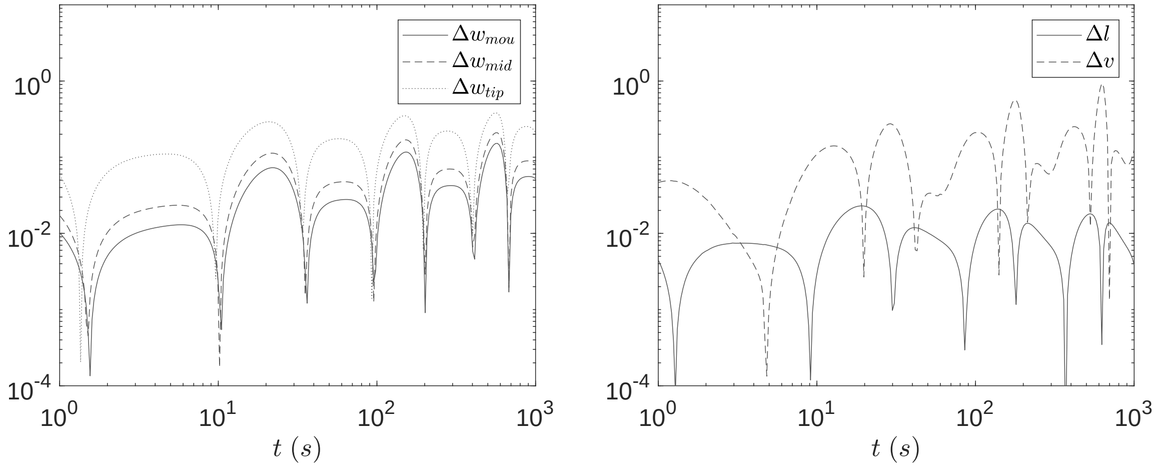


Figure 6.33: Radial model, oscillating toughness (amplitude $0.5K_{IC}$, period $1m$), Carter leak-off. Comparison with constant toughness case, crack opening (left), crack length and speed (right).

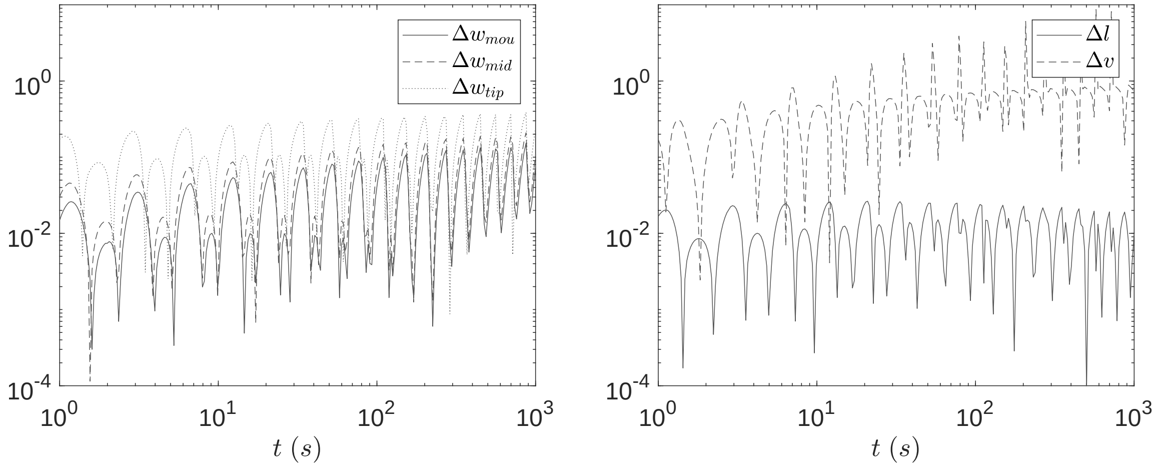
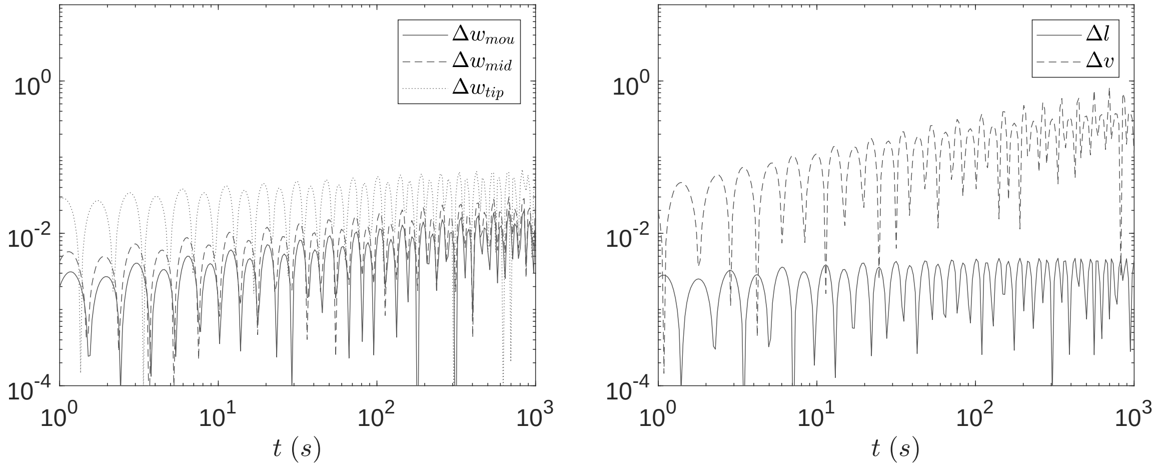


Figure 6.34: Radial model, oscillating toughness (amplitude $0.1K_{IC}$, period $1m$), Carter leak-off. Comparison with constant toughness case, crack opening (left), crack length and speed (right).



Remarks

If the amplitude of the oscillations in toughness is large enough, then its effect can be relevant and noticeable also for large values of time. Indeed we can see that the distance between the solutions of the oscillating case and of the constant case does not change as time passes.
In practice this means that the solution with oscillating toughness corresponds to a solution with a constant effective toughness, where this effective toughness is larger than the average of the original oscillating toughness.
The part of the solution that is most affected by the oscillations is the area close to the crack tip even though the oscillations are strong close to the crack mouth as well. The crack propagation speed has sharp variations in correspondence to the variations of toughness: when the crack reaches a tougher area it slows down and inflates and then when it passes to a weaker area it propagates rapidly and deflates.
The long time effect does not depend on the period of the oscillations, while instead it depends greatly on their amplitude and on the average value of the toughness itself. Indeed, if we look at the limiting cases, 0 average toughness implies 0 amplitude (toughness must be positive) and 0 amplitude obviously produces no change. Therefore the maximum effect can be expected in cases with big oscillations and large average toughness.
The presence of Carter leak-off decreases the effect of the oscillation compared to the case with impermeable rocks.

6.4 Conclusion

In this chapter we have run some simulations with oscillating parameters and we have seen the effects on crack propagation.

- Oscillating pumping rate.
  - The effect decays very rapidly in time becoming rapidly negligible.

- Oscillating leak-off coefficient.
  - The effect is stronger than in the case with oscillating pumping rate.
  - Despite this the oscillations of crack opening and length become rapidly negligible as well.
  - However crack propagation speed has sharp variations also for large values of time.

- Oscillating toughness.
  - The effect can be strong, in particular in case of big oscillations and large average toughness.
  - The effect does not decay in time and corresponds to a higher "effective toughness".
  - The long term effect does not depend on the period of the oscillations.

Chapter 7

Shear stress and energy release rate

The impact of shear stress on HF has been recently questioned in the case of a Newtonian fluid for KGD [81] and radial [55] models. In these papers a modified elasticity equation and a modified fracture propagation condition are proposed, we will implement them in our solver and make a comparison with the classic versions of KGD and radial models.

7.1 Modified models

KGD

We start from the modified KGD model, see [81] for more details.

Elasticity

The elasticity equation is modified to keep the shear stress into account and it takes the form

$$p(t,x) - \sigma_c = \int_{-l(t)}^{l(t)} \left(\frac{1-2\nu}{2\pi(1-\nu)} \tau(t,s) - \frac{E}{4\pi(1-\nu^2)} \frac{\partial w(t,s)}{\partial s} \right) \frac{ds}{s-x}, \tag{7.1}$$

where τ represents the shear stress. We note that if we set $\tau = 0$ we recover the classic KGD formulation seen previously. The shear stress can be expressed in terms of the crack opening and the derivative of the pressure as

$$\tau(t,x) = -\frac{w}{2} \frac{\partial p}{\partial x}, \tag{7.2}$$

where the sign is chosen accordingly to the formulation of the modified elasticity equation. Taking symmetry into account the equation can be rewritten as

$$p(t,x) - \sigma_c = \int_{0}^{l(t)} \left(\frac{1-2\nu}{\pi(1-\nu)} \tau(t,s) - \frac{E}{2\pi(1-\nu^2)} \frac{\partial w(t,s)}{\partial s} \right) \mathcal{I}(x,s) ds, \tag{7.3}$$

where $\mathcal{I}(x,s)$ is like in (2.84). It is possible to invert the integral operator obtaining

$$\int_{x}^{l(t)} \left(\frac{1-2\nu}{\pi(1-\nu)} \tau(t,s) - \frac{E}{2\pi(1-\nu^2)} \frac{\partial w(t,s)}{\partial s} \right) ds = \frac{4}{\pi^2} \int_{0}^{l(t)} (p(t,s) - \sigma_c) \mathcal{J}(x,s) ds, \tag{7.4}$$

where $\mathcal{J}(x,s)$ is like in (2.86). So we can express w as a function of τ and p

$$w(t,x) = k_s \int_{l(t)}^{x} \tau(t,s)ds + k_e \int_0^{l(t)} (p(t,s) - \sigma_c)\mathcal{J}(x,s)ds, \tag{7.5}$$

where the constant k_e is like in (2.88) and

$$k_s = \frac{2(1+\nu)(1-2\nu)}{E}. \tag{7.6}$$

Finally w can be written as

$$w(t,x) = k_s \int_{l(t)}^{x} \tau(t,s)ds + k_e \int_0^{l(t)} \frac{\partial p(t,s)}{\partial s}\mathcal{K}(x,s)ds + k_e\sqrt{l^2(t) - x^2}\int_0^{l(t)} \frac{p(t,s)-\sigma_c}{\sqrt{l^2(t)-s^2}}ds, \tag{7.7}$$

where $\mathcal{K}(x,s)$ is like in (2.90).

Fracture propagation

In [81], an asymptotic analysis of the modified KGD model has been done resulting in

$$w(t,x) = w_{1,0}(t)(l(t)-x)^{\frac{1}{2}} + \text{h.o.t.} \quad x \to l(t) \tag{7.8}$$

$$\tau(t,x) = \tau_{1,0}(t)(l(t)-x)^{-\frac{1}{2}} + \text{h.o.t.} \quad x \to l(t). \tag{7.9}$$

As a consequence, from the study of the stresses near the crack tip the authors found that the leading asymptotic coefficient of w must satisfy

$$w_{1,0}(t) = k_e\sqrt{\frac{\pi}{2}}K_I + \frac{4(1+\nu)}{E}\tau_{1,0}(t). \tag{7.10}$$

In addition they derived also a modified fracture propagation condition to keep the shear stress into account

$$K_{IC}^2 = K_I^2 + 4(1-\nu)K_F K_I = K_I^2 + 2\sqrt{2\pi}\tau_{1,0}(t)K_I. \tag{7.11}$$

Now we want to express these two conditions like in the classic KGD model. To do so we solve (7.11) for K_I and we get

$$K_I = -\sqrt{2\pi}\tau_{1,0}(t) \pm \sqrt{2\pi\tau_{1,0}^2(t) + K_{IC}^2}, \tag{7.12}$$

where we take the solution with the plus sign, because it generalises the classic case. If we substitute K_I in w_0 we get

$$w_{1,0}(t) = k_e\sqrt{\frac{\pi}{2}}\sqrt{K_{IC}^2 + 2\pi\tau_{1,0}^2(t)} - 2k_s\tau_{1,0}(t), \tag{7.13}$$

but from asymptotic analysis of (7.7) we also have that

$$w_{1,0}(t) = k_e\sqrt{2l(t)}\int_0^{l(t)} \frac{p(t,s)-\sigma_c}{\sqrt{l^2(t)-s^2}}ds - 2k_s\tau_{1,0}(t). \tag{7.14}$$

Therefore equating these last two formulas we get the modified condition on the pressure in the form

$$\sqrt{K_{IC}^2 + 2\pi\tau_{1,0}^2(t)} = 2\sqrt{\frac{l(t)}{\pi}}\int_0^{l(t)} \frac{p(t,s)-\sigma_c}{\sqrt{l^2(t)-s^2}}ds. \tag{7.15}$$

We note that if we set $\tau_{1,0} = 0$ we recover the classic formulation of KGD.

Normalisation

We normalise the model in the same way we have done in the classic case, but in addition this time we have the shear stress

$$\tilde{\tau}(\tilde{t},\tilde{x}) = k_e \tau(t,x), \quad \tilde{\tau}_{1,0}(\tilde{t}) = \frac{k_e}{\sqrt{l(t)}}\tau_{1,0}(t), \quad \tilde{k}_s = \frac{k_s}{k_e}. \tag{7.16}$$

We notice that for standard materials $\nu \in [0, \frac{1}{2}]$ therefore we have that $\tilde{k}_s = \frac{\pi}{4}\frac{1-2\nu}{1-\nu} \in [0, \frac{\pi}{4}]$. We can now rewrite the modified equations in the normalised form, while all the others remain the same as in the classic case.

– **elasticity equation**

$$\tilde{w}(\tilde{t},\tilde{x}) = \tilde{k}_s L(\tilde{t})\int_1^{\tilde{x}} \tilde{\tau}(\tilde{t},\tilde{s})ds + L(\tilde{t})\int_0^1 \frac{\partial \tilde{p}(\tilde{t},\tilde{s})}{\partial \tilde{s}}\tilde{\mathcal{K}}(\tilde{x},\tilde{s})d\tilde{s} + \sqrt{\tilde{K}_{IC}^2 + \frac{\pi^2}{2}L(\tilde{t})\tilde{\tau}_{1,0}^2(\tilde{t})}\sqrt{L(\tilde{t})}\sqrt{1-\tilde{x}^2}, \tag{7.17}$$

where $\tilde{\mathcal{K}}(\tilde{x},\tilde{s})$ is like in (2.104).

– **fracture propagation**

$$\sqrt{\tilde{K}_{IC}^2 + \frac{\pi^2}{2}L(\tilde{t})\tilde{\tau}_{1,0}^2(\tilde{t})} = \sqrt{L(\tilde{t})}\int_0^1 \frac{\tilde{p}(\tilde{t},\tilde{s})}{\sqrt{1-\tilde{s}^2}}d\tilde{s}. \tag{7.18}$$

– **shear stress**

$$\tilde{\tau}(\tilde{t},\tilde{x}) = -\frac{\tilde{w}(\tilde{t},\tilde{x})}{2L(\tilde{t})}\frac{\partial \tilde{p}}{\partial \tilde{x}}(\tilde{t},\tilde{x}) = \frac{1}{2}\frac{\tilde{q}(\tilde{t},\tilde{x})}{\tilde{w}^2(\tilde{t},\tilde{x})}. \tag{7.19}$$

From here on we will only consider the normalised modified KGD problem and we will omit the $\sim$ symbol for simplicity.

Asymptotics

As remarked in [81], the asymptotic behaviour at the crack tip of the modified model does not depend on toughness any longer and it corresponds to what happens in the case of the classic KGD model with $K_{IC} > 0$ and $n = 1$.

However it can be seen that if the pumping rate does not change dramatically in time, then, when time goes to infinity, the effect of shear stress becomes negligible. This means that the long time asymptote is the same as for the classic KGD. Therefore if $K_{IC} = 0$, then the long time asymptote of the modified model is the same of the one of the classic model and as a consequence at the crack tip it has a different behaviour from the rest of the solution.

We add here the asymptotic behaviour of the shear stress.

Crack mouth, when $(x \to 0^+)$ we have:

$$\tau(t,x) = \tau_{0,0}(t) + \tau_{0,1}(t)x + \text{h.o.t.}. \tag{7.20}$$

Crack tip, when $(x \to 1^-)$ we have:

$$\tau(t,x) = \tau_{1,0}(t)(1-x)^{-\frac{1}{2}} + \tau_{1,1}(t) + \text{h.o.t.}. \tag{7.21}$$

Remarks

In the modified KGD model it is possible to find a self-similar solution like in the classic case, but this time only in the exponential form. It is useful for testing purposes and it has been studied in [81].
We will also consider a partially modified version of the model, setting $\tilde{k}_s$ to 0 but keeping the modified fracture propagation condition. The asymptotic behaviour of the solution remains the same as in the case of the fully modified model and again it is possible to find a self-similar solution only in exponential form.

Radial

We discuss here the modified radial model, see [55] for more details, the procedure is basically the same as for KGD case.

Elasticity

Similarly to KGD, the elasticity equation is modified to keep the shear stress into account and it takes the form

$$p(t,r) - \sigma_c = \int_0^{l(t)} \left(\frac{1-2\nu}{\pi(1-\nu)} \tau(t,s) - \frac{E}{2\pi(1-\nu^2)} \frac{\partial w(t,s)}{\partial s} \right) \mathcal{I}(r,s) ds, \tag{7.22}$$

where τ represents the shear stress and $\mathcal{I}(r,s)$ is like in (2.139). We observe that if we set $\tau = 0$ we recover the classic formulation of radial seen previously. The shear stress can be expressed in terms of the crack opening and the derivative of the pressure as

$$\tau(t,r) = -\frac{w}{2} \frac{\partial p}{\partial r}, \tag{7.23}$$

where the sign is chosen accordingly to the formulation of the modified elasticity equation. It is possible to invert the integral operator and express w as a function of τ and p

$$w(t,r) = k_s \int_{l(t)}^{r} \tau(t,s) ds + k_e \int_0^{l(t)} (p(t,s) - \sigma_c) \mathcal{J}(r,s) ds, \tag{7.24}$$

where $\mathcal{J}(r,s)$ is like in (2.141), the constant k_e is like in (2.143) and

$$k_s = \frac{2(1+\nu)(1-2\nu)}{E}. \tag{7.25}$$

Finally w can be written as

$$w(t,r) = k_s \int_{l(t)}^{r} \tau(t,s) ds + k_e \int_0^{l(t)} \frac{\partial p(t,s)}{\partial s} \mathcal{K}(r,s) ds + k_e \sqrt{l^2(t) - r^2} \int_0^{l(t)} \frac{p(t,s) - \sigma_c}{\sqrt{l^2(t) - s^2}} \frac{s}{l(t)} ds, \tag{7.26}$$

where $\mathcal{K}(r,s)$ is like in (2.145).

Fracture propagation

Similarly to KGD, the asymptotic behaviour of the modified radial model is

$$w(t,r) = w_{1,0}(t)(l(t)-r)^{\frac{1}{2}} + \text{h.o.t.} \quad x \to l(t) \tag{7.27}$$

$$\tau(t,r) = \tau_{1,0}(t)(l(t)-r)^{-\frac{1}{2}} + \text{h.o.t.} \quad x \to l(t), \tag{7.28}$$

the leading asymptotic coefficient of w must satisfy

$$w_{1,0}(t) = k_e\sqrt{\frac{\pi}{2}}K_I + \frac{4(1+\nu)}{E}\tau_{1,0}(t) \tag{7.29}$$

and we have a modified fracture propagation condition

$$K_{IC}^2 = K_I^2 + 4(1-\nu)K_F K_I = K_I^2 + 2\sqrt{2\pi}\tau_{1,0}(t)K_I. \tag{7.30}$$

We observe that if we set $\tau_{1,0} = 0$ in the last two equations we recover the results of the classic formulation of radial. Proceeding like in KGD we get the modified condition on the pressure in the form

$$\sqrt{K_{IC}^2 + 2\pi\tau_{1,0}^2(t)} = 2\sqrt{\frac{l(t)}{\pi}}\int_0^{l(t)} \frac{p(t,s)-\sigma_c}{\sqrt{l^2(t)-s^2}}\frac{s}{l(t)}ds. \tag{7.31}$$

Normalisation

We normalise the model in the same way we have done in the classic case, but in addition this time we have the shear stress

$$\tilde{\tau}(\tilde{t},\tilde{r}) = k_e\tau(t,r), \quad \tilde{\tau}_{1,0}(\tilde{t}) = \frac{k_e}{\sqrt{l(t)}}\tau_{1,0}(t), \quad \tilde{k}_s = \frac{k_s}{k_e}. \tag{7.32}$$

We observe that for standard materials $\nu \in [0, \frac{1}{2}]$ therefore we have that $\tilde{k}_s = \frac{\pi}{4}\frac{1-2\nu}{1-\nu} \in [0, \frac{\pi}{4}]$. We can now rewrite the modified equations in the normalised form, while all the others remain the same as in the classic case.

– **elasticity equation**

$$\tilde{w}(\tilde{t},\tilde{r}) = \tilde{k}_s L(\tilde{t})\int_1^{\tilde{r}} \tilde{\tau}(\tilde{t},\tilde{s})ds + L(\tilde{t})\int_0^1 \frac{\partial\tilde{p}(\tilde{t},\tilde{s})}{\partial\tilde{s}}\tilde{\mathcal{K}}(\tilde{r},\tilde{s})d\tilde{s} + \sqrt{\tilde{K}_{IC}^2 + \frac{\pi^2}{2}L(\tilde{t})\tilde{\tau}_{1,0}^2(\tilde{t})}\sqrt{L(\tilde{t})}\sqrt{1-\tilde{r}^2}, \tag{7.33}$$

where $\tilde{\mathcal{K}}(\tilde{r},\tilde{s})$ is like in (2.158).

– **fracture propagation**

$$\sqrt{\tilde{K}_{IC}^2 + \frac{\pi^2}{2}L(\tilde{t})\tilde{\tau}_{1,0}^2(\tilde{t})} = \sqrt{L(\tilde{t})}\int_0^1 \frac{\tilde{p}(\tilde{t},\tilde{s})\tilde{s}}{\sqrt{1-\tilde{s}^2}}d\tilde{s}. \tag{7.34}$$

– **shear stress**

$$\tilde{\tau}(\tilde{t},\tilde{r}) = -\frac{\tilde{w}(\tilde{t},\tilde{r})}{2L(\tilde{t})}\frac{\partial\tilde{p}}{\partial\tilde{r}}(\tilde{t},\tilde{r}) = \frac{1}{2}\frac{\tilde{q}(\tilde{t},\tilde{r})}{\tilde{w}^2(\tilde{t},\tilde{r})}. \tag{7.35}$$

From here on we will only consider the normalised modified radial problem and we will omit the $\sim$ symbol for simplicity.

Asymptotics

The asymptotic behaviour of the modified radial model at the crack tip is the same as the one of the modified KGD, therefore what said previously for the modified KGD holds also for the modified radial. The asymptotic behaviour at the crack mouth instead this time is different.
Crack mouth, when $(r \to 0^+)$ we have:

$$\tau(t,r) = \tau_{0,0}(t) r^{-1} + \tau_{0,1}(t) + \text{h.o.t.}. \tag{7.36}$$

We note that in this way the asymptotic behaviour of τ is not compatible with the asymptotic behaviour of w coming from the classic radial model. For this reason they should be both corrected to match together, resulting in a logarithmic singularity of the crack opening at the crack mouth. In order to avoid this problem and considering that the effect of shear stress is not important close to the crack mouth, instead than using the definition in (7.35), we will define the shear stress as

$$\tau(t,r) = -\frac{w(t,r)}{2L(t)} \frac{\partial p}{\partial r}(t,r) r = \frac{1}{2} \frac{q(t,r)}{w^2(t,r)} r. \tag{7.37}$$

In this way the behaviour of the shear stress at the crack tip is not modified, while at the crack mouth it becomes

$$\tau(t,r) = \tau_{0,0}(t) + \tau_{0,1}(t) r + \text{h.o.t.}. \tag{7.38}$$

Therefore now the asymptotic behaviour of τ matches with the asymptotic behaviour of w coming from the classic radial model.

Remarks

Similarly to KGD, in the modified radial model it is possible to find a self-similar solution only in exponential form.
We will also consider a partially modified version of the model, setting $\tilde{k}_s$ to 0 but keeping the modified fracture propagation condition. The asymptotic behaviour of the solution remains the same as in the case of the fully modified model and again it is possible to find a self-similar solution only in exponential form.

7.2 Numerical tests

We test here the modified KGD and radial models comparing them to the classic versions. We consider both the fully modified models (shear stress and fracture propagation condition) and the partially modified models (only fracture propagation condition), both with and without leak-off. We run the simulations with constant inflow and with constant toughness, using the following parameters (the difference from the previous chapter is only the time interval):

$$\frac{E}{1-\nu^2} = 3 \cdot 10^{10}\ Pa, \qquad \nu = .25, \qquad K_{IC} = 1.3 \cdot 10^6\ Pa\sqrt{m}, \tag{7.39}$$

$$k_{cl} = 5 \cdot 10^{-5}\ m/\sqrt{s}, \qquad \mu_f = 10^{-3}\ Pa\ s, \qquad Q_* = 6 \cdot 10^{-3}\ m^3/s, \tag{7.40}$$

$$h = 30\ m\ (\text{KGD}), \qquad t_{start} = 10^{-6} s, \qquad t_{end} = 10^4 s. \tag{7.41}$$

As a consequence in the normalised variables we get approximately

$$\tilde{K}_{IC} = 10^{-4}, \qquad \tilde{k}_s = .5, \qquad k_{cl} = 10^{-10}, \tag{7.42}$$

$$\tilde{q}_* = 10^{-16}\ \text{(KGD)}, \quad 10^{-15}\ ,\text{(radial)}, \qquad \tilde{t}_{start} = 10^6, \qquad \tilde{t}_{end} = 10^{16}. \tag{7.43}$$

As for the initial conditions, we will start the simulations from a very small crack, of virtually zero length and zero opening.
For the comparison we will measure the relative difference between the solution generated using the classic model and the one coming from the modified versions

$$\delta l = \left| \frac{l^{mod}(t) - l^{cla}(t)}{l^{cla}(t)} \right|, \qquad \delta v = \left| \frac{\frac{dl}{dt}^{mod}(t) - \frac{dl}{dt}^{cla}(t)}{\frac{dl}{dt}^{cla}(t)} \right| \tag{7.44}$$

$$\delta w_{mou} = \left| \frac{w^{mod}(t,0) - w^{cla}(t,0)}{w^{cla}(t,0)} \right|, \quad \delta w_{mid} = \left| \frac{w^{mod}(t,\frac{l}{2}) - w^{cla}(t,\frac{l}{2})}{w^{cla}(t,\frac{l}{2})} \right|, \quad \delta w_{tip} = \left| \frac{w^{mod}(t,l) - w^{cla}(t,l)}{w^{cla}(t,l)} \right|. \tag{7.45}$$

Here f^{mod} stands for a variable taken from the solution of the modified model and f^{cla} for a variable taken from the solution of the classic model. The tests have been done asking an error smaller than 10^{-6}.

KGD

We consider here the fully and partially modified KGD models. We plot the relative difference between the two modified solutions and the solution coming from the classic model.
– In figure 7.1 we consider the fully modified model with no leak-off.
– In figure 7.2 we consider the fully modified model with Carter leak-off.
– In figure 7.3 we consider the partially modified model with no leak-off.
– In figure 7.4 we consider the partially modified model with Carter leak-off.
In the plots on the left we have the relative difference of the crack opening, in those on the right we have the relative difference of crack length and crack propagation speed.
From the numerical simulations we can see that, for the values of the physical parameters that we have used, the effects of the shear stress and of the modified fracture propagation condition are very small and visible only for very small values of time. In practice they are outside HF practical purposes and of the range of validity of the model. The difference is more noticeable in the case of the fully modified model and in the case of no leak-off. The most influenced part of the solution is the area of w near the crack tip, while crack length and propagation speed are less affected.

Figure 7.1: Fully modified KGD model against classic with no leak-off (tolerance 10^{-6}). Comparison of the crack opening (left), crack length and speed (right).

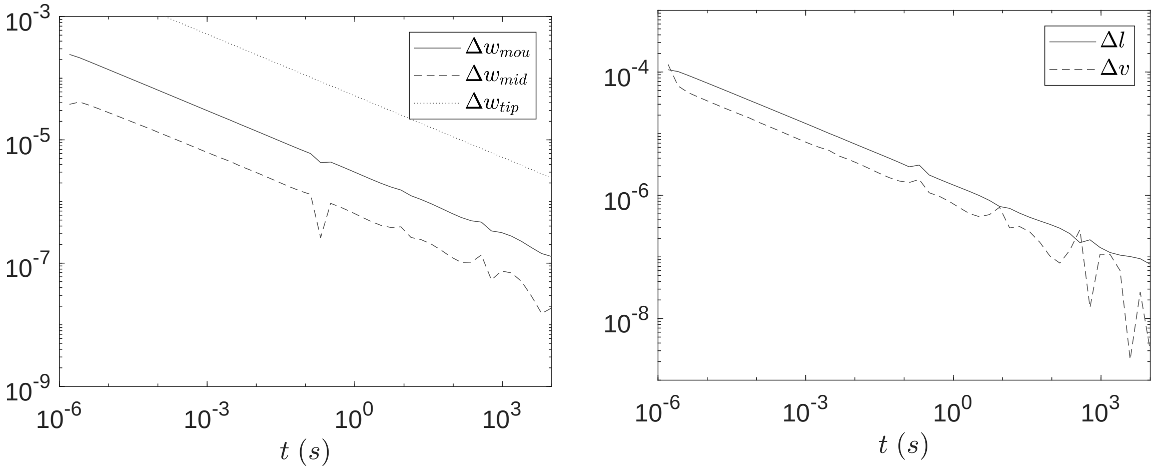


Figure 7.2: Fully modified KGD model against classic with Carter leak-off (tolerance 10^{-6}). Comparison of the crack opening (left), crack length and speed (right).

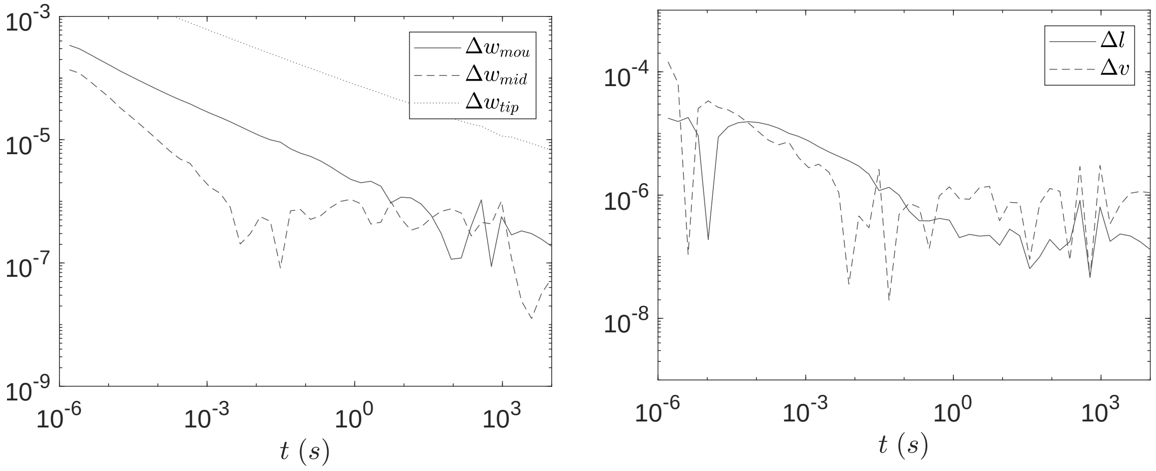


Figure 7.3: Partially modified KGD model against classic with no leak-off (tolerance 10^{-6}). Comparison of the crack opening (left), crack length and speed (right).

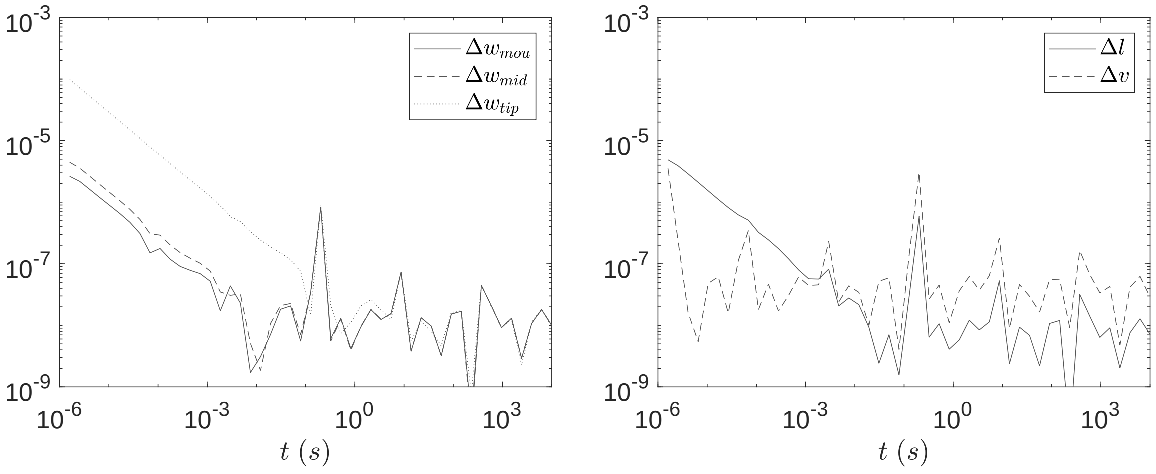
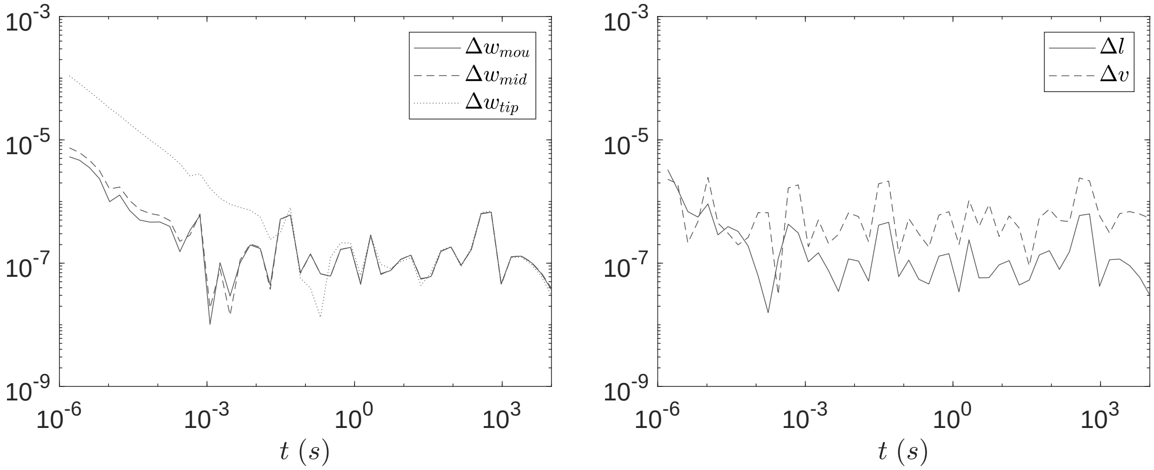


Figure 7.4: Partially modified KGD model against classic with Carter leak-off (tolerance 10^{-6}). Comparison of the crack opening (left), crack length and speed (right).



Radial

We consider here the fully and partially modified radial models and proceed like previously in the KGD case.
– In figure 7.5 we consider the fully modified model with no leak-off.
– In figure 7.6 we consider the fully modified model with Carter leak-off.
– In figure 7.7 we consider the partially modified model with no leak-off.
– In figure 7.8 we consider the partially modified model with Carter leak-off.
The simulations for the radial model show similar results to the KGD case. Also for the radial model, for the values of the physical parameters that we have used, the effects of shear stress and of the modified fracture propagation condition are very small and visible only for very small values of time.

Figure 7.5: Fully modified radial model against classic with no leak-off (tolerance 10^{-6}). Comparison of the crack opening (left), crack length and speed (right).

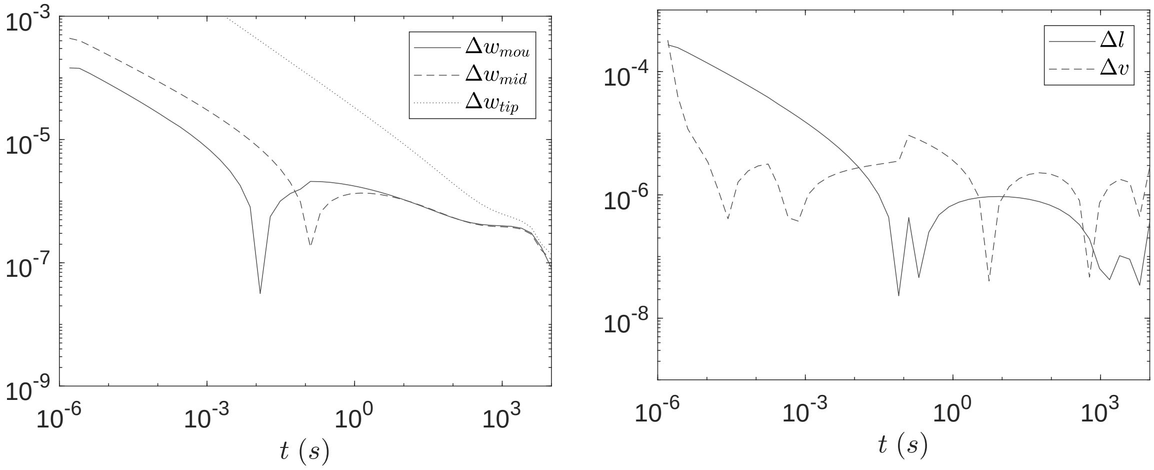


Figure 7.6: Fully modified radial model against classic with Carter leak-off (tolerance 10^{-6}). Comparison of the crack opening (left), crack length and speed (right).

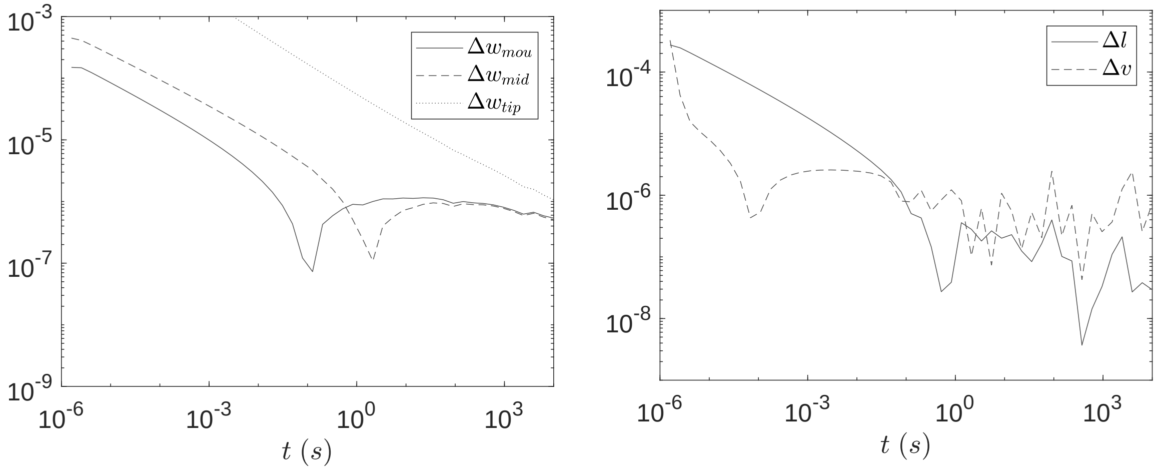


Figure 7.7: Partially modified radial model against classic with no leak-off (tolerance 10^{-6}). Comparison of the crack opening (left), crack length and speed (right).

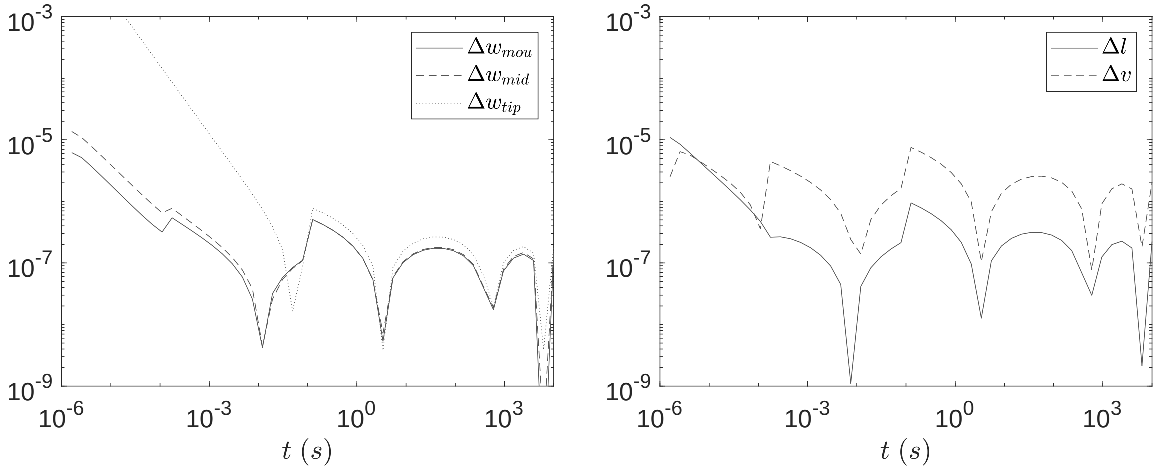
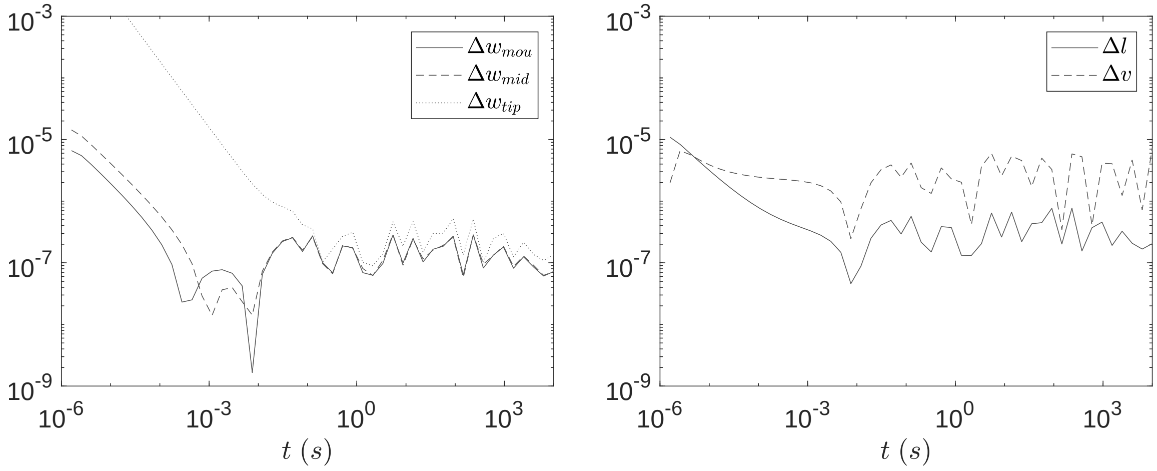


Figure 7.8: Partially modified radial model against classic with Carter leak-off (tolerance 10^{-6}). Comparison of the crack opening (left), crack length and speed (right).



7.3 Conclusion

In this chapter we have run some simulations with the modified KGD and radial models that consider the effects of fluid induced shear stress at the fracture walls.

- For the values of the problem that we have used in the tests, the effect of shear stress is very small. Indeed the effect is barely visible for small values of time and then decays rapidly.
- A higher pumping rate and a more viscous fluid would make the effect more relevant, further tests can be done to find a threshold.
- An advantage of the modified model is that the asymptotic behaviour of the solution is simpler because it does not change when $K_{IC} = 0$.
- However when $K_{IC} = 0$ and pumping rate does not change dramatically in time, the long time asymptote has still the asymptotic behaviour of the classic viscosity solution. How far in time the transition in the asymptotics starts, depends again on the input parameters.

Chapter 8

Future work

8.1 Models

Fluid flow

We have treated the case where the fluid is modelled as power-law. More in general, for a generalised Newtonian fluid, Poiseuille equation cannot be expressed in a closed form. It is however possible to write it as

$$\frac{\partial p}{\partial x} = -f\left(\frac{\bar{q}}{\bar{w}^2}\right)\frac{\bar{q}}{\bar{w}^3}, \tag{8.1}$$

where f is a function that depends on the fluid model and can be approximated numerically.
In particular, in the case of a Carreau fluid, the behaviour at zero and infinite shear rate is like that of a Newtonian fluid. Therefore if $\mu_{c,0}$ and $\mu_{c,\infty}$ are finite and greater than 0, then also f is always finite and greater than 0. This means that if we use the Carreau fluid model in HF, the resulting solution would have the same asymptotic behaviour of the case with a Newtonian fluid. Therefore once one has approximated the function f, the Carreau fluid model can be easily implemented in our solver with only small modifications.

Leak-off

The same thing just said for fluid flow can be applied to leak off. For a generalised Newtonian fluid, a Carter-like equation cannot be expressed in a closed form. It is however possible to write it as

$$\bar{q}_l = \frac{g(t - t_0(x))}{\sqrt{t - t_0(x)}}, \tag{8.2}$$

where g is a function that depends on the fluid model and can be approximated numerically.
For the same reasons stated above, if we use the Carreau fluid model for leak-off and $\mu_{c,0}$ and $\mu_{c,\infty}$ are finite and greater than 0, then also g is always finite and greater than 0. As a consequence, with a Carreau fluid, the solution of the problem would have the same asymptotic behaviour of the case with a Newtonian fluid. Therefore once one has approximated the function g, the Carreau fluid model can be easily implemented in our solver with only small modifications.
It must be noted however that there is still some debate in the scientific community on how non-Newtonian flow through porous media should be modelled [8]. As a consequence there is not an agreed unique way to compute mathematically the function g. For this reason an alternative approach could be instead approximating this function directly from physical experiments.

8.2 Numerical methods

Approximation

We have seen in chapter 3 how to increase the regularity of a function at the boundary to get better results in its approximation. Yet it is possible to go further and, with a suitable change of variable, make it become C^∞ and possibly get geometric convergence. To build such a transformation it is sufficient to consider a function such that its derivative has compact support, as we will see more in detail below.

Asymptotics on one side

For f with irregular behaviour on the right as in (3.117), we can define g as

$$g(y) = \frac{f(s(y))}{w(y)}, \quad \text{where} \tag{8.3}$$

$$s(y) = (1+e^2)\tanh\left(\frac{2}{1-y}\right) - e^2 \tag{8.4}$$

$$w(y) = \exp\left(-\frac{4\alpha_0}{1-y}\right), \tag{8.5}$$

therefore $f(x) = g(s^{-1}(x))w(s^{-1}(x))$ and the asymptotic behaviour of g is

$$g(y) = a_0 + O\left(\exp\left(-\frac{4\min(\alpha_1-\alpha_0, 1)}{(1-y)}\right)\right) \quad y \to 1. \tag{8.6}$$

At point 1 all the derivatives of g are sent to 0 and g is C^∞.

Asymptotics on both sides

For f with irregular behaviour at both endpoints as in (3.122,3.123), we can define g as

$$g(y) = \frac{f(s(y))}{w(y)}, \quad \text{where} \tag{8.7}$$

$$s(y) = \tanh\left(\frac{2}{1-y} - \frac{2}{1+y}\right) \tag{8.8}$$

$$w(y) = \exp\left(-\frac{4\alpha_0}{1-y}\right)\exp\left(-\frac{4\beta_0}{1+y}\right), \tag{8.9}$$

therefore $f(x) = g(s^{-1}(x))w(s^{-1}(x))$ and the asymptotic behaviour of g is

$$g(y) = \hat{a}_0 + \hat{a}_1(1-y) + \hat{a}_2(1-y)^2 + \dots \quad y \to 1 \tag{8.10}$$

$$g(y) = \hat{b}_0 + \hat{b}_1(1+y) + \hat{b}_2(1+y)^2 + \dots \quad y \to -1. \tag{8.11}$$

This means that at points -1 and 1 g is now C^∞.

Approximation

We can now effectively approximate g with a polynomial, obtaining

$$f(x) \approx p_n(s^{-1}(x))w(s^{-1}(x)), \tag{8.12}$$

a polynomial p_n composed with a smooth change of variables, multiplied by a weight function. The polynomial p_n can be computed interpolating the function $g(y)$ at the Chebyshev nodes y_i, that in the original variable become $x_i = s(y_i)$.

Remarks

The use of the hyperbolic tangent can be found for example in [72, 73], to perform definite integrals with singularities moving them to an infinite domain. In those papers other similar transformations are presented as well. In our case we chose the simplest transformation that goes from $[-1, 1]$ to $[-1, 1]$ such that its derivative has compact support in $[-1, 1]$.

Indefinite integration

The hyperbolic tangent change of variable can be used also for the integration to get a C^∞ function and possibly geometric convergence.

Asymptotics on both sides

For simplicity we will consider only the case with bad asymptotic behaviour on both sides, the case of one side can be derived similarly. For f as in (3.122,3.123), we can define g as

$$g(y) = \frac{f(s(y))s'(y)}{w(y)}, \quad \text{where} \tag{8.13}$$

$$s(y) = \tanh\left(\frac{2}{1-y} - \frac{2}{1+y}\right) \tag{8.14}$$

$$s'(y) = \frac{4(1+y^2)}{(1-y^2)^2}\,\mathrm{sech}\left(\frac{2}{1-y} - \frac{2}{1+y}\right)^2 \tag{8.15}$$

$$w(y) = \frac{\exp\left(\frac{-4(\alpha_0+1)}{1-y}\right)}{(1-y)^2}\frac{\exp\left(\frac{-4(\beta_0+1)}{1+y}\right)}{(1+y)^2}, \tag{8.16}$$

therefore $\int f(x)dx = \int g(y)w(y)dy$ and the asymptotic behaviour of g is

$$g(y) = \hat{a}_0 + \hat{a}_1(1-y) + \hat{a}_2(1-y)^2 + \dots \quad y \to 1 \tag{8.17}$$

$$g(y) = \hat{b}_0 + \hat{b}_1(1+y) + \hat{b}_2(1+y)^2 + \dots \quad y \to -1. \tag{8.18}$$

This means that at points -1 and 1 g is now C^∞.

Approximation

We can now effectively approximate g with a polynomial, obtaining

$$\int f(x)dx \approx \int p_n(y)w(y)dy, \tag{8.19}$$

the integral of a polynomial multiplied for a weight function. As in the previous section the polynomial p_n can be computed interpolating the function $g(y)$ at the Chebyshev nodes y_i, that in the original variable become $x_i = s(y_i)$.

Integral operator

In this case we can split the interval in two in x_0 and then apply the change of variables defined above to the two semi intervals. We can define the function g

$$g(y) = f(s(y)), \quad \text{where} \tag{8.20}$$

$$s(y) = \begin{cases} \left(\tanh\left(\frac{1-y_0}{y-y_0} - \frac{1-y_0}{1-y}\right) + 1\right)\frac{y_0-1}{2} + 1, & y > y_0 = x_0 \\ \left(\tanh\left(\frac{1+y_0}{y_0-y} - \frac{1+y_0}{1+y}\right) + 1\right)\frac{1+y_0}{2} - 1, & y < y_0 = x_0, \end{cases} \tag{8.21}$$

therefore the asymptotic behaviour of g is

$$g(y) = \exp\left(-2\alpha_0\frac{1-y_0}{1-y}\right)(\hat{a}_0 + \hat{a}_1(1-y) + \hat{a}_2(1-y)^2 + \ldots) \quad y \to 1 \tag{8.22}$$

$$g(y) = \hat{c}_0 + \exp\left(-2\frac{1-y_0}{y-y_0}\right)(\hat{c}_{r0}(y-y_0)^{-1} + \hat{c}_{r1} + \hat{c}_{r2}(y-y_0) + \ldots) \quad y \to y_0^+ \tag{8.23}$$

$$g(y) = \hat{c}_0 + \exp\left(-2\frac{1+y_0}{y_0-y}\right)(\hat{c}_{l0}(y_0-y)^{-1} + \hat{c}_{l1} + \hat{c}_{l2}(y_0-y) + \ldots) \quad y \to y_0^- \tag{8.24}$$

$$g(y) = \exp\left(-2\beta_0\frac{1+y_0}{1+y}\right)(\hat{b}_0 + \hat{b}_1(1+y) + \hat{b}_2(1+y)^2 + \ldots) \quad y \to -1. \tag{8.25}$$

Finally we consider the integral

$$\int_{-1}^{1} f(x)dx = \int_{-1}^{1} g(y)s'(y)dy = \int_{-1}^{1} h(y)dy, \tag{8.26}$$

where h

$$h(z) = g(y)s'(y) \tag{8.27}$$

and its asymptotic behaviour is

$$h(y) = \exp\left(-2(\alpha_0+1)\frac{1-y_0}{1-y}\right)(\bar{a}_0(1-y)^{-2} + \bar{a}_1(1-y)^{-1} + \bar{a}_2 + \ldots) \quad y \to 1 \tag{8.28}$$

$$h(y) = \exp\left(-2\frac{1-y_0}{y-y_0}\right)(\bar{c}_{r0}(y-y_0)^{-2} + \bar{c}_{r1}(y-y_0)^{-1} + \bar{c}_{r2} + \ldots) \quad y \to y_0^+ \tag{8.29}$$

$$h(y) = \exp\left(-2\frac{1+y_0}{y_0-y}\right)(\bar{c}_{l0}(y_0-y)^{-2} + \bar{c}_{l1}(y_0-y)^{-1} + \bar{c}_{l2} + \ldots) \quad y \to y_0^- \tag{8.30}$$

$$h(y) = \exp\left(-2(\beta_0+1)\frac{1+y_0}{1+y}\right)(\bar{b}_0(1+y)^{-2} + \bar{b}_1(1+y)^{-1} + \bar{b}_2 + \ldots) \quad y \to -1. \tag{8.31}$$

This means that at points -1, y_0 and 1 h is now C^∞.

Approximation

We can now effectively approximate h with a polynomial, obtaining

$$\int_{-1}^{1} f(x)dx \approx \int_{-1}^{1} p_n(y)dy. \tag{8.32}$$

Again the polynomial p_n can be computed interpolating the function $h(y)$ at the Chebyshev nodes y_i, that in the original variable become $x_i = s(y_i)$.
In alternative we can also consider splitting the integral in two parts, that might give better results because of the asymptotic behaviour of h in y_0. In this case we interpolate h with two different polynomials on the two sub-intervals, obtaining

$$\int_{-1}^{1} f(x)dx \approx \int_{-1}^{y_0} pl_n(y)dy + \int_{y_0}^{1} pr_n(y)dy. \tag{8.33}$$

Integral of a weighted polynomial

The integral of a weighted polynomial in the case of the hyperbolic tangent change of variables can be performed using the same approach adopted with the beta change of variable.

Asymptotics on one side

We want to compute the integral

$$F(x) = \int p_n(x)w(x)dx, \quad \text{where} \quad w(x) = \frac{\exp(-\frac{\alpha}{1-x})}{(1-x)^2}. \tag{8.34}$$

It is possible to express $F(x)$ as

$$F(x) = P_{n-1}(x)\exp\left(-\frac{\alpha}{1-x}\right) + cg(x), \tag{8.35}$$

where

$$g(x) = \int_{1}^{x} (1-x)w(x)dx \tag{8.36}$$

and $P_{n-1}(x)$ is a polynomial of degree $n-1$. Then if we differentiate we get

$$\begin{aligned} F'(x) =& \frac{dP_{n-1}(x)}{dx}\exp\left(-\frac{\alpha}{1-x}\right) - P_{n-1}(x)\exp\left(-\frac{\alpha}{1-x}\right)\frac{\alpha}{(1-x)^2} + cg'(x) && (8.37)\\ =& \left(\frac{dP_{n-1}(x)}{dx}(1-x)^2 - \alpha P_{n-1}(x) + c(1-x)\right)w(x) && (8.38)\\ =& q_n(x)w(x), && (8.39) \end{aligned}$$

where $q_n(x)$ is a polynomial of degree n. Therefore we need to find $P_{n-1}(x)$ and c such that $q_n(x) = p_n(x)$. We start writing $P_{n-1}(x)$ in terms of the Chebyshev basis of first kind and $p_n(x)$ in terms of the Chebyshev basis of second kind

$$P_n(x) = \sum_{i=1}^{n} a_i T_{i-1}(x) \quad \text{and} \quad p_n(x) = \sum_{i=0}^{n} b_i U_i(x). \tag{8.40}$$

Using the properties of the Chebyshev polynomials it can be seen that

$$\frac{d(T_i(x)\exp\left(-\frac{\alpha}{1-x}\right))}{dx} = \left(\frac{i}{4}U_{i+1}(x) - \left(i+\frac{\alpha}{2}\right)U_i(x) + \frac{3}{2}iU_{i-1}(x) - \left(i-\frac{\alpha}{2}\right)U_{i-2}(x) + \frac{i}{4}U_{i-3}(x)\right)w(x) \tag{8.41}$$

$$\frac{d(T_2(x)\exp\left(-\frac{\alpha}{1-x}\right))}{dx} = \left(\frac{1}{2}U_3(x) - \left(2+\frac{\alpha}{2}\right)U_2(x) + 3U_1(x) - \left(2-\frac{\alpha}{2}\right)U_0(x)\right)w(x) \tag{8.42}$$

$$\frac{d(T_1(x)\exp\left(-\frac{\alpha}{1-x}\right))}{dx} = \left(\frac{1}{4}U_2(x) - \left(1+\frac{\alpha}{2}\right)U_1(x) + \frac{5}{4}U_0(x)\right)w(x) \tag{8.43}$$

$$\frac{d(T_0(x)\exp\left(-\frac{\alpha}{1-x}\right))}{dx} = -\alpha U_0(x)w(x) \tag{8.44}$$

and in addition we have that

$$\frac{dg(x)}{dx} = \left(-\frac{1}{2}U_1(x) + U_0(x)\right)w(x). \tag{8.45}$$

Now if we set $a_0 = c$, then we have a relation between a_i and b_i in terms of the linear system

$$\begin{pmatrix} d_{1,0} & d_{2,0} & d_{3,0} & d_{4,0} & d_{5,0} & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & d_{1,n-4} & d_{2,n-4} & d_{3,n-4} & d_{4,n-4} & d_{5,n-4} \\ & & & d_{1,n-3} & d_{2,n-3} & d_{3,n-3} & d_{4,n-3} \\ & & & & d_{1,n-2} & d_{2,n-2} & d_{3,n-2} \\ & & & & & d_{1,n-1} & d_{2,n-1} \\ & & & & & & d_{1,n} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}, \tag{8.46}$$

that has an upper triangular band matrix and can be solved in $O(n)$ operations using back substitution. Finally $g(x)$ can be expressed in the form of the exponential integral function

$$g(x) = \mathrm{Ei}\left(-\frac{\alpha}{1-x}\right), \tag{8.47}$$

that provides fast and accurate computation.

We note that when $\alpha = 0$ the term $a_1 T_0(x)$ is sent to 0 after differentiation, therefore in this case it must be substituted by $a_1 \frac{1}{1-x}$, and we obtain

$$\frac{d\left(\frac{1}{1-x}\right)}{dx} = U_0(x)w(x). \tag{8.48}$$

In addition when $\alpha = 0$ for $g(x)$ we obtain the simpler formula

$$g(x) = -\ln(1-x) \tag{8.49}$$

Asymptotics on both sides

We want to compute the integral

$$F(x) = \int p_n(x)w(x)dx, \quad \text{where} \quad w(x) = \frac{\exp(-\frac{\alpha}{1-x})}{(1-x)^2}\frac{\exp(-\frac{\beta}{1+x})}{(1+x)^2} = \frac{\exp(-\frac{\alpha}{1-x}-\frac{\beta}{1+x})}{(1-x^2)^2}. \tag{8.50}$$

It is possible to express $F(x)$ as

$$F(x) = P_{n-3}(x)\exp\left(-\frac{\alpha}{1-x}-\frac{\beta}{1+x}\right)+c_3g_3(x)+c_2g_2(x)+c_1g_1(x)+c_0g_0(x), \tag{8.51}$$

where

$$\begin{align} g_3(x) &= \int_1^x T_3(x)w(x)dx \tag{8.52}\\ g_2(x) &= \int_1^x T_2(x)w(x)dx \tag{8.53}\\ g_1(x) &= \int_1^x T_1(x)w(x)dx \tag{8.54}\\ g_0(x) &= \int_1^x T_0(x)w(x)dx \tag{8.55}\end{align}$$

and $P_{n-3}(x)$ is a polynomial of degree $n-3$. Then if we differentiate we get

$$\begin{align} F'(x) =& \frac{dP_{n-3}(x)}{dx}\exp\left(-\frac{\alpha}{1-x}-\frac{\beta}{1+x}\right) \tag{8.56}\\ &- P_{n-1}(x)\exp\left(-\frac{\alpha}{1-x}-\frac{\beta}{1+x}\right)\left(\frac{\alpha}{(1-x)^2}-\frac{\beta}{(1+x)^2}\right) \tag{8.57}\\ &+ c_3g_3'(x)+c_2g_2'(x)+c_1g_1'(x)+c_0g_0'(x) \tag{8.58}\\ =& \Bigg(\frac{dP_{n-3}(x)}{dx}(1-x)^2(1+x)^2 - P_{n-3}(x)(\alpha(1+x)^2-\beta(1-x)^2) \tag{8.59}\\ &+ c_3T_3(x)+c_2T_2(x)+c_1T_1(x)+c_0T_0(x)\Bigg)w(x) \tag{8.60}\\ =& q_n(x)w(x), \tag{8.61}\end{align}$$

where $q_n(x)$ is a polynomial of degree n. Therefore we need to find $P_{n-3}(x)$ and c_3, c_2, c_1, c_0 such that $q_n(x) = p_n(x)$. We start writing $P_{n-3}(x)$ and $p_n(x)$ in terms of the Chebyshev basis of first kind

$$P_n(x) = \sum_{i=4}^{n} a_iT_{i-3}(x) \quad \text{and} \quad p_n(x) = \sum_{i=0}^{n} b_iT_i(x), \tag{8.62}$$

where we note that in the sum we do not use the term $a_3T_0(x)$. Using the properties of the Chebyshev polynomials it can be seen that

$$\frac{d(T_i(x)\exp\left(-\frac{\alpha}{1-x}-\frac{\beta}{1+x}\right))}{dx} = \Bigg(\frac{i}{8}T_{i+3}(x) - \frac{\alpha-\beta}{4}T_{i+2}(x) - \left(\alpha+\beta+\frac{3}{8}i\right)T_{i+1}(x) - \frac{3}{2}(\alpha-\beta)T_i(x) \tag{8.63}$$

$$- \left(\alpha+\beta-\frac{3}{8}i\right)T_{i-1}(x) - \frac{\alpha-\beta}{4}T_{i-2}(x) - \frac{i}{8}T_{i-3}(x)\Bigg)w(x) \tag{8.64}$$

$$\frac{d(T_2(x)\exp\left(-\frac{\alpha}{1-x}-\frac{\beta}{1+x}\right))}{dx} = \Bigg(\frac{1}{4}T_5(x) - \frac{\alpha-\beta}{4}T_4(x) - \left(\alpha+\beta+\frac{3}{4}\right)T_3(x) - \frac{3}{2}(\alpha-\beta)T_2(x) \tag{8.65}$$

$$- \left(\alpha+\beta-\frac{1}{2}\right)T_1(x) - \frac{\alpha-\beta}{4}T_0(x)\Bigg)w(x) \tag{8.66}$$

$$\frac{d(T_1(x)\exp\left(-\frac{\alpha}{1-x}-\frac{\beta}{1+x}\right))}{dx} = \Bigg(\frac{1}{8}T_4(x) - \frac{\alpha-\beta}{4}T_3(x) - \left(\alpha+\beta+\frac{1}{2}\right)T_2(x) - \frac{7}{4}(\alpha-\beta)T_1(x) \tag{8.67}$$

$$- \left(\alpha+\beta-\frac{3}{8}\right)T_0(x)\Bigg)w(x) \tag{8.68}$$

$$\frac{d(T_0(x)\exp\left(-\frac{\alpha}{1-x}-\frac{\beta}{1+x}\right))}{dx} = \Bigg(- \frac{\alpha-\beta}{2}T_2(x) - 2\left(\alpha+\beta\right)T_1(x) - \frac{3}{2}(\alpha-\beta)T_0(x)\Bigg)w(x) \tag{8.69}$$

and in addition we have that

$$\frac{dg_3(x)}{dx} = T_3(x)w(x) \tag{8.70}$$

$$\frac{dg_2(x)}{dx} = T_2(x)w(x) \tag{8.71}$$

$$\frac{dg_1(x)}{dx} = T_1(x)w(x) \tag{8.72}$$

$$\frac{dg_0(x)}{dx} = T_0(x)w(x). \tag{8.73}$$

Then we set $a_0 = c_0, a_1 = c_1, a_2 = c_2, a_3 = c_3$, so we have now a relation between a_i and b_i in terms of the linear system

$$\begin{pmatrix}
d_{1,0} & d_{2,0} & d_{3,0} & d_{4,0} & d_{5,0} & d_{6,0} & d_{7,0} & & \\
 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\
 & & d_{1,n-6} & d_{2,n-6} & d_{3,n-6} & d_{4,n-6} & d_{5,n-6} & d_{6,n-6} & d_{7,n-6} \\
 & & & d_{1,n-5} & d_{2,n-5} & d_{3,n-5} & d_{4,n-5} & d_{5,n-5} & d_{6,n-5} \\
 & & & & d_{1,n-4} & d_{2,n-4} & d_{3,n-4} & d_{4,n-4} & d_{5,n-4} \\
 & & & & & d_{1,n-3} & d_{2,n-3} & d_{3,n-3} & d_{4,n-3} \\
 & & & & & & d_{1,n-2} & d_{2,n-2} & d_{3,n-2} \\
 & & & & & & & d_{1,n-1} & d_{2,n-1} \\
 & & & & & & & & d_{1,n}
\end{pmatrix}
\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}, \tag{8.74}$$

that has an upper triangular band matrix and can be solved in $O(n)$ operation using back substitution. Finally, the functions $g_i(x)$ can be computed numerically splitting the interval $[-1, 1]$ in two halves and then using the result of the previous case (asymptotics on one side) to integrate them in $[-1, 0]$ and $[0, 1]$.

Chapter 9

Conclusion

The aim of this work was to build a solver for the three classic 1D models that is able to provide a fast and accurate solution in every reasonable condition. In addition we wanted to build an algorithm that is adaptive in time and space, to be able to keep the error under control at every step while at the same time minimising the amount of computations required.

We started from the main points of:

- normalisation of the problem to pass from a moving to a fixed boundary
- appropriate treatment of the asymptotic behaviour of the solution at the boundary
- modified formulation of the elasticity integral operator.

Particular attention was required to deal with the irregular asymptotic behaviour of the solution at the crack tip and at the crack mouth. Moreover we had to take into account that the asymptotic behaviour of the solution can change when moving through different regimes (toughness or viscosity and storage or leak-off). Therefore the function spaces used to approximate the solution had to be carefully chosen in a way that they could provide good results in all the situations. Additional difficulties also had to be overcome in KGD and radial models because of the evaluation of the elasticity integral operators.

Despite these complications, our aim was to use high order approximation to obtain fast convergence and accurate results with few evaluation points. To do so we proceeded with:

- a smoothing transformation to make the solution more regular at the boundary
- approximation through weighted polynomial interpolation on the Chebyshev nodes
- an additional smoothing transformation to evaluate the elasticity integral operator
- accurate evaluation of the kernels through the use of symmetric elliptic integrals.

Another objective was to build a solver adaptive in space and time to save machine computations, while keeping the error under control. For this purpose, at every time step we had to use a spacial mesh as small as possible and we had to choose the time step itself dynamically depending on the behaviour of the solution. We obtained this using the following techniques:

- multigrid approach to save time and keep the error under control
- implicit Runge Kutta method for time discretisation, that has a high order while remaining stable for stiff problems.

Using all these techniques combined together allowed us to build a solver that met our goals.

- We have built a space-time adaptive algorithm that works for all classic 1D HF models, with or without leak-off. It can also include the effects of fluid induced shear stress.
- The solver works well in all the regimes (viscosity, toughness, storage, leak-off) and during the transition among them.
- We have tested the algorithm against all known semi-analytical solutions to asses its correctness.
- The algorithm has proved to be very effective. It can provide very accurate results with the use of few approximation points even in the hardest cases (small toughness for instance).
- We have analysed the effect of the oscillation of different physical parameters. In particular oscillating toughness can have relevant effects on the solution.
- We have investigated the impact of the shear stress. Further tests with different values of the parameters will allow to understand when the effects can be relevant and to verify also the consequences on the asymptotics.

Appendix A

Additional derivations

A.1 DCT-I and DST-I in terms of FFT

DCT-I through FFT

It is easy to verify that a N-terms DCT-I can be expressed through a $2N-2$-terms FFT as

$$\mathcal{C}(x_0, x_1, \ldots, x_{N-1}) = \Re(Y_0, Y_1, \ldots, Y_{N-1}), \quad \text{where} \tag{A.1}$$
$$(Y_0, Y_1, \ldots, Y_{N-1}, Y_{N-2}^*, Y_{N-3}^*, \ldots, Y_1^*) = \mathcal{F}(x_0/2, x_1, x_2, \ldots, x_{N-2}, x_{N-1}/2, 0, \ldots, 0). \tag{A.2}$$

It is also possible to express a N-terms DCT-I through a $N-1$-terms FFT with a more complicated formula

$$\mathcal{C}(x_0, x_1, \ldots, x_{N-1}) = \Re(Y \odot \exp(-iZ)) \oslash (\cos(Z) + \sin(Z)), \tag{A.3}$$

where

$$X^0 = \begin{cases} (x_0, x_2, \ldots, x_{N-2}, x_{N-2}, x_{N-4}, \ldots, x_2) & \text{if } N \text{ is even} \\ (x_0, x_2, \ldots, x_{N-3}, x_{N-1}, x_{N-3}, \ldots, x_2) & \text{if } N \text{ is odd} \end{cases} \tag{A.4}$$

$$X^1 = \begin{cases} (x_1, x_3, \ldots, x_{N-3}, x_{N-1}, x_{N-3}, \ldots, x_1) & \text{if } N \text{ is even} \\ (x_1, x_3, \ldots, x_{N-2}, x_{N-2}, x_{N-4}, \ldots, x_1) & \text{if } N \text{ is odd} \end{cases} \tag{A.5}$$

$$Y^0 = \left(\mathcal{F}\left(\frac{X^0 + X^1}{2}\right),\ 0\right) \tag{A.6}$$

$$Y^1 = \left(\mathcal{F}\left(i\frac{X^0 - X^1}{2}\right),\ i\sum_{n=0}^{N-1}{}' (-1)^n x_n\right) \tag{A.7}$$

$$Y = Y^0 + Y^1 = \left(\mathcal{F}\left(\frac{X^0 + X^1}{2} + i\frac{X^0 - X^1}{2}\right),\ i\sum_{n=0}^{N-1}{}' (-1)^n x_n\right) \tag{A.8}$$

$$Z = \frac{\pi}{2}\frac{(0, \ldots, N-1)}{(N-1)} \tag{A.9}$$

and $\odot$ and $\oslash$ are the Hadamard product and division (i.e. product and division term by term). To prove this equality it is sufficient to verify that

$$\mathcal{C}(x_0, x_1, \ldots, x_{N-1}) \odot \cos(Z) = \Re(Y^0 \odot \exp(-iZ)) \tag{A.10}$$
$$\mathcal{C}(x_0, x_1, \ldots, x_{N-1}) \odot \sin(Z) = \Re(Y^1 \odot \exp(-iZ)). \tag{A.11}$$

Indeed using product-to-sum trigonometric identities it can be seen that

$$\mathcal{C}(x_0, x_1, \dots, x_{N-1}) \odot \cos(Z) \tag{A.12}$$

$$= \left(\text{DCT-II}\left(\frac{x_0}{2}, x_1, x_2, \dots, x_{N-3}, x_{N-2} + \frac{x_{N-1}}{2}\right) + \text{DCT-II}\left(\frac{x_0}{2} + x_1, x_2, x_3, \dots, x_{N-2}, \frac{x_{N-1}}{2}\right), 0\right)/2 \tag{A.13}$$

$$= \left(\text{DCT-II}\left(\frac{x_0+x_1}{2}, \frac{x_1+x_2}{2}, \dots, \frac{x_{N-2}+x_{N-1}}{2}\right), 0\right) = \Re(Y^0 \odot \exp(-iZ)), \tag{A.14}$$

where DCT-II is the type-II discrete cosine transform and the last equality can be found in [45]. Similarly we have that

$$\mathcal{C}(x_0, x_1, \dots, x_{N-1}) \odot \sin(Z) \tag{A.15}$$

$$= \left(0, \text{DST-II}\left(\frac{x_0}{2}, x_1, x_2, \dots, x_{N-3}, x_{N-2} - \frac{x_{N-1}}{2}\right) - \text{DST-II}\left(-\frac{x_0}{2} + x_1, x_2, x_3, \dots, x_{N-2}, \frac{x_{N-1}}{2}\right)\right)/2 \tag{A.16}$$

$$= \left(0, \text{DST-II}\left(\frac{x_0-x_1}{2}, \frac{x_1-x_2}{2}, \dots, \frac{x_{N-2}-x_{N-1}}{2}\right)\right) = \Re(Y^1 \odot \exp(-iZ)) \tag{A.17}$$

where DST-II is the type-II discrete sine transform and the proof is completed.

DST-I through FFT

It is also easy to verify that a a N-terms DST-I can be expressed through a $2N+2$-terms DFT as

$$\mathcal{S}(x_0, x_1, \dots, x_{N-1}) = -\Im(Y_0, Y_1, \dots, Y_{N-1}), \quad \text{where} \tag{A.18}$$

$$(Y_{-1}, Y_0, \dots, Y_N, Y^*_{N-1}, Y^*_{N-2}, \dots, Y^*_0) = \mathcal{F}(0, x_0, x_1, \dots, x_{N-1}, 0, \dots, 0). \tag{A.19}$$

It is also possible to express a N-terms DST-I through a $N+1$-terms FFT with a more complicates formula

$$(0, \mathcal{S}(x_0, x_1, \dots, x_{N-1})) = \Re(Y \odot \exp(-iZ)) \oslash (\cos(Z) + \sin(Z)), \tag{A.20}$$

where

$$X^0 = \begin{cases} (x_0, x_2, \dots, x_{N-2}, 0, -x_{N-2}, -x_{N-4}, \dots, -x_0) & \text{if } N \text{ is even} \\ (x_0, x_2, \dots, x_{N-1}, -x_{N-1}, -x_{N-3}, \dots, -x_0) & \text{if } N \text{ is odd} \end{cases} \tag{A.21}$$

$$X^1 = \begin{cases} (0, x_1, x_3, \dots, x_{N-1}, -x_{N-1}, -x_{N-3}, \dots, -x_1) & \text{if } N \text{ is even} \\ (0, x_1, x_3, \dots, x_{N-2}, 0, -x_{N-2}, -x_{N-4}, \dots, -x_1) & \text{if } N \text{ is odd} \end{cases} \tag{A.22}$$

$$Y^0 = \mathcal{F}\left(\frac{X^0+X^1}{2}\right) \tag{A.23}$$

$$Y^1 = \mathcal{F}\left(i\frac{X^0-X^1}{2}\right) \tag{A.24}$$

$$Y = Y^0 + Y^1 = \mathcal{F}\left(\frac{X^0+X^1}{2} + i\frac{X^0-X^1}{2}\right) \tag{A.25}$$

$$Z = \frac{\pi}{2}\frac{(0, \dots, N)}{(N+1)}. \tag{A.26}$$

The proof is based on the same idea as the case of the DCT-I seen above.

A.2 Kernel alternative formulations

KGD

The equivalences between 3.204 and 3.205 and between 3.217 and 3.218 follow immediately from [21, Eq. 4.37.24]

$$\operatorname{artanh}(x) = \frac{1}{2}\ln\left(\frac{1+x}{1-x}\right). \tag{A.27}$$

The equivalence between 3.205 and 3.206 instead follows directly from the definition of $\operatorname{artanh}(x)$ [21, Eq. 4.37.3]

$$\operatorname{artanh} x = \int_0^x \frac{1}{1-t^2}dt. \tag{A.28}$$

Therefore when $s < x$ we have that

$$\operatorname{artanh}\left(\sqrt{\frac{1-x^2}{1-s^2}}\right) = \int_0^{\sqrt{\frac{1-x^2}{1-s^2}}} \frac{1}{1-t^2}dt = F\left(\arcsin\left(\sqrt{\frac{1-x^2}{1-s^2}}\right)\middle|1\right) \tag{A.29}$$

and in the same way in the case when $s > x$. Finally the equivalence between 3.218 and 3.219 can be seen in a similar fashion, indeed when $s < x$ we have that

$$x\operatorname{artanh}\left(\frac{s}{x}\sqrt{\frac{1-x^2}{1-s^2}}\right) - s\operatorname{artanh}\left(\sqrt{\frac{1-x^2}{1-s^2}}\right) \tag{A.30}$$

$$= x\int_0^{\frac{s}{x}\sqrt{\frac{1-x^2}{1-s^2}}} \frac{dt}{1-t^2} - s\int_0^{\sqrt{\frac{1-x^2}{1-s^2}}} \frac{dt}{1-t^2} = x\int_0^{\sqrt{\frac{1-x^2}{1-s^2}}} \frac{s}{x}\frac{dt}{1-\frac{s^2}{x^2}t^2}dt - s\int_0^{\sqrt{\frac{1-x^2}{1-s^2}}} \frac{dt}{1-t^2} \tag{A.31}$$

$$= s\frac{s^2-x^2}{x^2}\int_0^{\sqrt{\frac{1-x^2}{1-s^2}}} \frac{t^2dt}{(1-\frac{s^2}{x^2}t^2)(1-t^2)} = s\frac{s^2-x^2}{x^2}J\left(\arcsin\left(\sqrt{\frac{1-x^2}{1-s^2}}\right), \frac{s^2}{x^2}\middle|1\right) \tag{A.32}$$

and in the same way in the case when $s > x$.

Radial

The equivalence in 3.223 can be derived as follows

$$\left[E\left(\arcsin(s)\middle|\frac{r^2}{s^2}\right) - E\left(\arcsin\left(\frac{s}{r}\right)\middle|\frac{r^2}{s^2}\right)\right] = \int_{\frac{s}{r}}^{s} \frac{\sqrt{1-\frac{r^2}{s^2}t^2}}{\sqrt{1-t^2}}dt \tag{A.33}$$

$$= \frac{s}{r}\int_1^r \frac{\sqrt{1-t^2}}{\sqrt{1-\frac{s^2}{r^2}t^2}}dt = \left[B\left(\arcsin(r)\middle|\frac{s^2}{r^2}\right) - B\left(\frac{s^2}{r^2}\right)\right]\frac{s}{r}. \tag{A.34}$$

With some more manipulation also the equivalence between 3.223 and 3.224 can be found. In the case $s < r$ we have

$$\left[B\left(\arcsin\left(r\right)\middle|\frac{s^2}{r^2}\right) - B\left(\frac{s^2}{r^2}\right)\right] = \int_1^r \frac{\sqrt{1-t^2}}{\sqrt{1-\frac{s^2}{r^2}t^2}}dt = \tag{A.35}$$

$$-\int_0^{\sqrt{1-r^2}} \frac{t^2}{\sqrt{1-\frac{s^2}{r^2}(1-t^2)}\sqrt{1-t^2}}dt = -\frac{r}{\sqrt{r^2-s^2}}\int_0^{\sqrt{1-r^2}} \frac{t^2}{\sqrt{1-\frac{s^2}{s^2-r^2}t^2}\sqrt{1-t^2}}dt = \tag{A.36}$$

$$-\frac{r}{\sqrt{r^2-s^2}}D\left(\arcsin(\sqrt{1-r^2})\middle|\frac{s^2}{s^2-r^2}\right) \tag{A.37}$$

and in the same way in the case when $s > x$.

Bibliography

[1] Acheson D.J., Elementary Fluid Dynamics. Oxford Applied Mathematics and Computing Science Series, Oxford University Press (1990).

[2] Adachi J., Siebrits E., Peirce A., Desroches J., Computer simulation of hydraulic fractures. International Journal of Rock Mechanics and Mining Sciences 44, 739-757 (2007).

[3] Ahmed N., Natarajan T., Rao K.R., Discrete Cosine Transform. IEEE Transactions on Computers, C-23, 90-93 (1974).

[4] Alexander R., Diagonally implicit Runge-Kutta methods for stiff OD Es. SIAM Journal on Numerical Analysis, 14, 1006-1021 (1977).

[5] Anderson T.L., Fracture Mechanics: Fundamentals and Applications. CRC Press (1995).

[6] Anderson C. and Dahleh M.D., Rapid computation of the discrete Fourier transform. SIAM Journal on Scientific Computing, 17, 913-919 (1996).

[7] Arora K.R., Soil Mechanics and Foundation Engineering. Standard Publishers (1989).

[8] Bowers C.A., Miller C.T. Non-Newtonian Fluid Flow in Porous Media, in review (2020).

[9] Butcher J.C., On the implementation of implicit Runge-Kutta methods. BIT 16, 237-240 (1976).

[10] Butcher J.C., A transformed implicit Runge-Kutta method. Journal of the Association for Computing Machinery, 26, 731-738 (1979).

[11] Butcher J.C., Numerical Methods for Ordinary Differential Equations, Second Edition. Wiley (2008).

[12] Carlson B.C., Computing elliptic integrals by duplication. Numerische Mathematik 33, 1-16 (1979).

[13] Carlson B.C., Numerical computation of real or complex elliptic integrals. Numerical Algorithms 10, 13-26 (1995).

[14] Carlson B.C., Three Improvements in Reduction and Computation of Elliptic Integrals. Journal of Research of the National Institute of Standards and Technology 107, 413-418 (2002).

[15] Chen W.H., Smith C.H., Fralick S.C., A Fast Computational Algorithm for the Discrete Cosine Transform. IEEE Transactions on Communications 25, 1004-1009 (1977).

[16] Clifton R.J, Abou-Sayed A.S., A variational approach to the prediction of the three-dimensional geometry of hydraulic fractures. Proceedings of the SPE/DOE low permeability symposium, Denver, May 27-29 (1981).

[17] Cooley, J.W., Turkey J.W., An algorithm for the machine calculation of complex Fourier series. Mathematics of Computation 19, 297-301 (1965).

[18] Dahlquist G.G., A special stability problem for linear multistep methods. BIT 3, 27-43 (1963).

[19] Damjanac B., Detournay C., Cundall P.A., Varun. Three-dimensional numerical model of hydraulic fracturing in fractured rock masses. In: Effective and sustainable hydraulic fracturing, 819-30 Bunger A, Jeffrey RG, McLennan J, editors (2013).

[20] Detournay E., Garagash D., The near-tip region of a fluid-driven fracture propagating in a permeable elastic solid. Journal of Fluid Mechanics, 494, 1-32 (2003).

[21] NIST Digital Library of Mathematical Functions. http://dlmf.nist.gov/, Release 1.0.26 of 2020-03-15. Olver F.W.J., Olde Daalhuis A.B., Lozier D.W., Schneider B.I., Boisvert R.F., Clark C.W., Miller B.R., Saunders B.V., Cohl H.S., McClain M.A., eds.

[22] Dontsov E.V., Peirce A.P., A multiscale implicit level set algorithm (ILSA) to model hydraulic fracture propagation incorporating combined viscous, toughness, and leak-off asymptotics. Computer Methods in Applied Mechanics and Engineering 313, 53-84 (2017).

[23] Fukushima, T. Fast computation of incomplete elliptic integral of first kind by half argument transformation. Numerische Mathematik 116, 687-719 (2010).

[24] Fukushima, T. Precise and fast computation of a general incomplete elliptic integral of second kind by half and double argument transformations. Journal of Computational and Applied Mathematics 235, 4140-4148 (2011)

[25] Fukushima, T. Precise and fast computation of a general incomplete elliptic integral of third kind by half and double argument transformations. Journal of Computational and Applied Mathematics 236, 1961-1975 (2012)

[26] Garagash D. and Detournay E., The Tip Region of a Fluid-Driven Fracture in an Elastic Medium. ASME. Journal of Applied Mechanics 67, 183-192 (2000).

[27] Geertsma J., de Klerk F., A rapid method of predicting width and extent of hydraulically induced fractures. Journal of Petroleum Technology 21, 1571-81 (1969).

[28] Gordeliy E., Abbas S., Peirce A., Modeling nonplanar hydraulic fracture propagation using the XFEM: An implicit level-set algorithm and fracture tip asymptotics. International Journal of Solids and Structures 159, 135-155 (2019).

[29] Gordeliy E., Peirce A., Coupling schemes for modeling hydraulic fracture propagation using the XFEM. Computer Methods in Applied Mechanics and Engineering 253, 305-322 (2013).

[30] Gordeliy E., Peirce A., Implicit level set schemes for modeling hydraulic fractures using the XFEM. Computer Methods in Applied Mechanics and Engineering 266, 125-143 (2013).

[31] Gordeliy E., Peirce A., Enrichment strategies and convergence properties of the XFEM for hydraulic fracture problems. Computer Methods in Applied Mechanics and Engineering 283, 474-502 (2015).

[32] Greengard L.; Lee J.Y., Accelerating the Nonuniform Fast Fourier Transform. SIAM Review 46, 443-45 (2004)

[33] Green A.E, Sneddon I.N., The distribution of stress in the neighbourhood of a flat elliptical crack in an elastic solid. Mathematical Proceedings of the Cambridge Philosophical Society 46, 159-63 (1950).

[34] Griffith A.A., The phenomena of rupture and flow in solids. Philosophical Transactions of the Royal Society of London A-221, 163-198 (1921).

[35] Hairer E. and Wanner G., Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems. Springer (1996).

[36] Irgens F., Rheology and Non-Newtonian Fluids. Springer (2014).

[37] Irwin G., Analysis of stresses and strains near the end of a crack traversing a plate. Journal of Applied Mechanics 24, 361-364 (1957).

[38] Kanin E.A., Dontsov E.V., Garagash D.I., Osiptsov A.A., A radial hydraulic fracture with pressure-dependent leak-off. Journal of the Mechanics and Physics of Solids, 104062 (2020).

[39] Khristianovic S.A., Zheltov Y.P., Formation of vertical fractures by means of highly viscous liquid. In: Proceedings of the fourth world petroleum congress, Rome, 579-86 (1955).

[40] Kusmierczyk P., Numerical Simulation of Hydraulic Fractures. PhD thesis, Aberystwyth University (2015).

[41] Lecampion B., Bunger A., Zhang X., Numerical methods for hydraulic fracture propagation: A review of recent trends. Journal of Natural Gas Science and Engineering 49, 66-83 (2018).

[42] Linkov A.M., On efficient simulation of hydraulic fracturing in terms of particle velocity. International Journal of Engineering Science 52, 77-88 (2012).

[43] Linkov A.M., Analytical solution of hydraulic fracture problem for a non-Newtonian fluid. Journal of Mining Science 49, 8-18 (2013).

[44] Linkov A.M., On decaying influence of initial conditions in the problem of hydraulic fracturing. Doklady Physics 61, 351-353 (2016).

[45] Makhoul J., A fast cosine transform in one and two dimensions. IEEE Transactions on Acoustics, Speech, and Signal Processing 28, 27-34 (1980).

[46] Mishuris G., Wrobel M., Linkov A., On modeling hydraulic fracture in proper variables: Stiffness, accuracy, sensitivity. International Journal of Engineering Science 61, 10-23 (2012).

[47] Mitchell S., Kuske R., Peirce A., An asymptotic framework for finite hydraulic fractures including leak-off. SIAM Journal on Applied Mathematics 67, 364-86 (2007).

[48] Mitchell S., Kuske R., Peirce A., An asymptotic framework for the analysis of hydraulic fractures: the impermeable case. ASME Journal of Applied Mechanics 74, 365-72 (2007).

[49] Monegato G., Scuderi L., Numerical integration of functions with boundary singularities. Journal of Computational and Applied Mathematics 112, 201-214 (1999).

[50] Monegato G., Sloan I.H., Numerical Solution of the Generalized Airfoil Equation for an Airfoil with a Flap. SIAM Journal on Numerical Analysis 34, 2288-2305 (1997).

[51] Morozov A., Popkov D., Duplyakov V., Mutalova R., Osiptsov A., Vainshtein A., Burnaev E., Shel E., Paderin G., Machine Learning on Field Data for Hydraulic Fracturing Design Optimization: Digital Database and Production Forecast Model. European Association of Geoscientists and Engineers, Conference Proceedings, First EAGE Digitalization Conference and Exhibition, Nov 2020, 1-5 (2020).

[52] Muskhelishvili N.I., Kveselava D.A., Singular integral equations with a Cauchy type kernel on open contours. (Russian) Trans. Tbil. Math. Inst. Acad. Sci. Georgian SSR 2, 141-172 (1942).

[53] Nordgren R.P., Propagation of a vertical hydraulic fracture. Society of Petroleum Engineers Journal 12, 306-14 (1972).

[54] Osiptsov A.A., Fluid Mechanics of Hydraulic Fracturing: a Review. Journal of Petroleum Science and Engineering 156, 513-535 (2017).

[55] Peck D., Axisymmetric problems involving fractures with moving boundaries. PhD thesis, Aberystwyth University (2017).

[56] Peck D., Wrobel M., Perkowska M., Mishuris G., Fluid velocity based simulation of hydraulic fracture: a penny shaped modelвЂ”part I: the numerical algorithm. Meccanica 53, 3615-3635 (2018).

[57] Peirce A., Implicit level set algorithms for modelling hydraulic fracture propagation. Philosophical Transactions of the Royal Society A 374, 20150423 (2016).

[58] Peirce A., Detournay E., An implicit level set method for modeling hydraulically driven fractures. Computer Methods in Applied Mechanics and Engineering 197, 2858-2885 (2008).

[59] Perkins T.K., Kern L.R. Widths of hydraulic fractures. Journal of Petroleum Technology 13, 937-49 (1961).

[60] Perkowska M., Mathematical and numerical modelling of hydraulic fractures for non-Newtonian fluids. PhD thesis, Aberystwyth University (2016).

[61] Perkowska M., Wrobel M., Mishuris G., Universal hydrofracturing algorithm for shear-thinning fluids: Particle velocity based simulation. Computers and Geotechnics 71, 310-337 (2016).

[62] Piessens R., Branders M., Numerical solution of integral equations of mathematical physics, using Chebyshev polynomials. Journal of Computational Physics 21, 178-196 (1976).

[63] Rice J., Mathematical analysis in the mechanics of fracture. In: Fracture, an Advanced Treatise, vol.2, Liebowitz H, (ed). Academic Press, New York; 191-311 (1968).

[64] Ruiz-AntolГn D., Townsend A., A Nonuniform Fast Fourier Transform Based on Low Rank Approximation. SIAM Journal on Scientific Computing 40, A529-A547 (2018).

[65] Siebrits E., Peirce A.P. An efficient multiвЂђlayer planar 3D fracture growth algorithm using a fixed mesh approach. International journal for numerical methods in engineering 53, 691-717 (2001).

[66] Shahsavari S., McKinley G.H., Mobility of power-law and Carreau fluids through fibrous media. Physical Review E 92, 063012 (2015).

[67] Simonson E.R., Abou-Sayed A.S., Clifton R.J., Containment of massive hydraulic fractures. Society of Petroleum Engineers Journal 18, 27-32 (1978).

[68] Slaughter W.S., The linearized theory of elasticity, Birkhauser (2002).

[69] Sneddon I.N., Elliot H.A., The opening of a Griffith crack under internal pressure. Quarterly of Applied Mathematics 4, 262-7 (1946).

[70] Sneddon I.N. The distribution of stress in the neighbourhood of a crack in an elastic solid. Proceedings of the Royal Society A 187, 229-260 (1946).

[71] Sneddon I. N., Lowengrub M., Crack Problems in the Classical Theory of Elasticity. The SIAM series in applied mathematics, Wiley (1969).

[72] Takahasi H., Mori M., Quadrature formulas obtained by variable transformation. Numerische Mathematik 21, 206-219 (1973).

[73] Takahasi, H., Mori, M. Double Exponential Formulas for Numerical Integration. Publications of the Research Institute for Mathematical Sciences 9, 721-741 (1974).

[74] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P., Numerical Recipes: The Art of Scientific Computing (3rd ed.). New York: Cambridge University Press (2007).

[75] Trefethen L.N., Approximation theory, approximation practice. Society for Industrial and Applied Mathematics (2012).

[76] van der Houwen P.J., Sommeijer B.P., Iterated Runge-Kutta Methods on Parallel Computers. SIAM Journal on Scientific and Statistical Computing 12, 1000-1028 (1991).

[77] Yamamoto K., Shimamoto T., Sukemura S., Multiple fracture propagation model for a three-dimensional hydraulic fracturing simulator. International Journal of Geomechanics 4, 46-57 (2004).

[78] Yew C.H., Weng X., Mechanics of Hydraulic Fracturing - 2nd Edition. Gulf Professional Publishing (2014).

[79] Wangen M., Finite element modeling of hydraulic fracturing in 3D. Computational Geosciences 17, 647-59 (2013).

[80] Wrobel M., Mishuris G., Hydraulic fracture revisited: Particle velocity based simulation. International Journal of Engineering Science 94, 23-58 (2015).

[81] Wrobel M., Mishuris G., Piccolroaz A., Energy release rate in hydraulic fracture: can we neglect an impact of the hydraulically induced shear stress? International Journal of Engineering Science 111, 28-51 (2017).

[82] Zia H., Lecampion B., Explicit versus implicit front advancing schemes for the simulation of hydraulic fracture growth. Numerical and Analytical Methods in Geomechanics 43, 1300-1315 (2019).

[83] Zia H., Lecampion B., PyFrac: A planar 3D hydraulic fracture simulator. Computer Physics Communications 255, 107368 (2020).