1 Extra computational challenges

1.1 Leak off computation strategies

To compute $\tau(x)$ it is necessary to find the inverse length function over the entire interval $x \in (0,1)$. The initial condition gives a non-zero positive value l_* (**??**), however the information about the history of the process, including inverse length function, is not given in the original formulation. It could be understood that the crack was previously opened and filled with fluid for unknown time period, or have just been opened and well defined $l^{-1}(x)$ is available. The physical justification for Carter law is that a cake filter layer is being formed on exposed crack surfaces. Hence one could expect an impermeable layer to already exist on the initial opening, or try to use some reference values for early $\tau(x)$ from some known solution. This two approaches will be considered:

- If the initial fracture was opened for a long time, the initial opening is fully saturated:

$$\tau_a(x) = \begin{cases} -\infty & \text{if } 0 \le l(t)x \le l_*, \\ \tau_{num}(x) & \text{if } l_* < l(t)x \le l(t); \end{cases} \tag{1}$$

- If the initial fracture developed similarly to the zero leak-off self-similar solution [**?**] (for small time it is reasonable that $\int_0^1 q_l(x)dx \to 0$):

$$\tau_b(x) = \begin{cases} (x\xi_* x_n)^{5/4} - t_0; & \text{if } 0 \le l(t)x \le l_*, \\ \tau_{num}(x) & \text{if } l_* < l(t)x \le l(t); \end{cases} \tag{2}$$

(*ewentualnie scenariusz nr 3 pusta szczelina z wybuchu*)

where $\tau_{num}(x)$ refers to output of numerical exposure time calculation scheme (1.2). The effects on $q_l$value of both of these approaches are presented in Figure 1.1. The comparison of these two cannot be however completed without understanding the effect of such on fracture solution. On Figure 2 it is shown how a fracture with sufficiently large leak off can recede at initial times instead propagating as expected. Further more second strategy can be extended to two variants

$\tau_b^{(1)}$- only recent fracture history is taken into account, reopened segments are treated as if no cake layer existed

$\tau_b^{(2)}$- fracture history is traced form the beginning, reopened segments are taken into account, thous $\tau_b^{(2)}(x)$ might return multiple opening times.

1.2 Exposure time computation

For the both variants of Carter leak-off **??** it is necessary to perform additional computation of exposure time $t - \tau(x)$ for each grid point x_i. This is however be a separate challenge on its own, especially if it is to be performed within the derivative function supplied to an arbitrary ODE solver (as it is in our

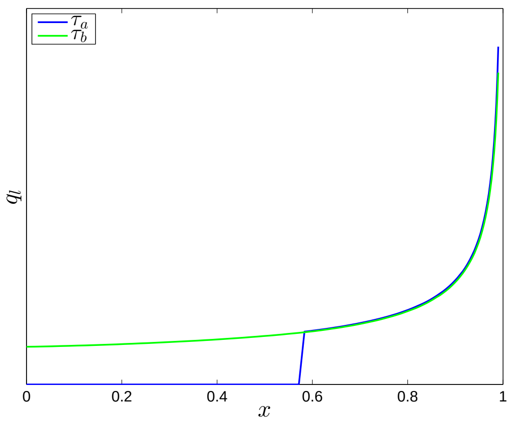

Figure 1: Comparison of leak off variations τ_a and $\tau_b$on a fracure with $l_* \approx 0.6$

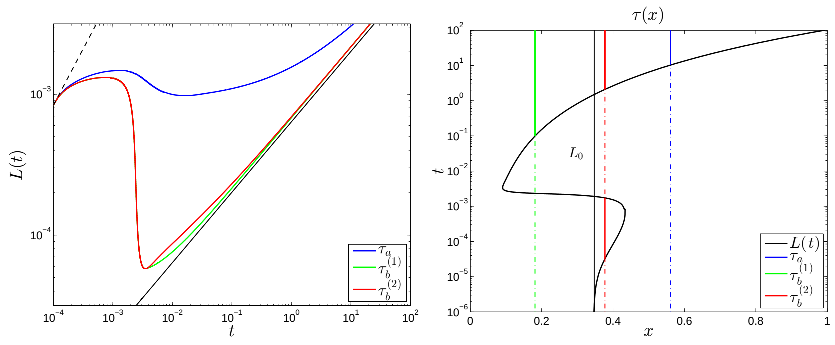

Figure 2: Comparison of effects various interpretations of initial fracture history and $\tau(x)$ calculation.

approach). Furthermore if a point on was fracture previously opened and closed several times the exposure time is rather:

$$\tau_{num}(x) = \tau_1(x) + \tau_2(x) - \tau_3(x) + ... + \tau_{2N_\tau}(x) - \tau_{2N_\tau+1}(x) \tag{3}$$

Where odd terms are opening times and even times are closure times, and N_τ is the number of times a fracture was opened.

As the solution is integrated the function [ref] is called, and values of $L(t_i)$, t_i, and $V_0(t_i)$ are as calculated at each call are stored in an sorted, by t_i, dynamic array. Values are filtered before insertion so that each value of L_i and t_i is greater than the previous one, with the exception of closing fracture events (4). Then when $t - \tau(x)$ is to be found, the formed array is iterated over to form the trajectory of the crack tip (black line in Figure 2. As this iteration is made, for each grid point on x-axis the number of times the crack tip passed through that point is accounted for , as shown by intersections of vertical lines with black line in Figure 2. Cubic polynomial is interpolated on two points $(L(t_i), t_i)$, $(L(t_{i+1}), t_{i+1})$, such so $L(t_i) < xL(t_i) < L(t_{i+1})$ or $L(t_{i+1}) < xL(t_i) < L(t_i)$ for closing segments, with derivative conditions from $V_0(t_i)$ and $V_0(t_{i+1})$ to approximate this intersections. Finally these times are combined to obtain total exposure time for each grid point.

Algorithm 1 Scheme for numerically aproximating $\tau(x)$ in a propagating or closing fracture

```
store each $L(t_i)$, $t_i$ and $V_0(t_i)$ sorted by $t_i$;
start with first grid point $x_k = x_0$;
for each stored $t_i$
-if $L(t_{i+1}) > L(t_i)$
–do
—find $\tau(x_k)$ by interpolating $L(t_i)$, $t_i$,$V_0(t_i)$ and $L(t_{i+1})$, $t_{i+1}$,$V_0(t_{i+1})$;
—store this $\tau(x_k)$ as opening time for $x_k$;
—move to next grid point $k := k + 1$;
–while $x_k$>$L(t_i)$
-end
-if $L(t_{i+1}) < L(t_i)$
–do
—find $\tau(x_k)$ by interpolating $L(t_i)$, $t_i$,$V_0(t_i)$ and $L(t_{i+1})$, $t_{i+1}$,$V_0(t_{i+1})$;
—store this $\tau(x_k)$ as closing time for $x_k$;
—move to previous grid point $k := k - 1$;
–while $x_k$<$L(t_i)$
-end
end
for each grid point $x$
-find $\tau_{num}(x)$ from 3 using previously stored $\tau$ values;
end
```

1.3 Modified closing fracture algorithm

The original problem formulation was designed strictly for propagating or already opened fracture. The underlying assumptions were that the crack propagation speed is positive ($V(1) > 0$) and the crack width is greater than zero ($w(t, x) > 0$), except for the crack tip where $w(t, 1) = 0$. These could however become invalid under specific conditions. Fracture remains open as long as fluid flow inside the fracture is greater than the leak-off, then proposed a simple observation that: *if the fracture width is monotonically decreasing from mouth to tip, and the value of leak-off monotonically increases up to the crack tip, then the first possible non-tip point x where $w(t, x) = 0$ is $1-\varepsilon$ for an infinitely small ε.* (pisac do tego dowod ? dla p carter trzeba porownac $w = ()^{1/3} + ()^{5/6}$ do $\frac{w}{\sqrt{1-x}}$)

Noting that MATLAB ODE15s solver provides an option to include events, lets supply a simple event function:

$$f_{event}(t) = w_N. \tag{4}$$

Then when the value of the crack width at the last grid point reaches $w_N = 0$, ODE15s is going to stop computation. Now it is possible to restart the computation after a small modification of the solution at the time the event occurred:

$$l_* = (1-\varepsilon)l_{end}, \qquad w_*(x) = w_{end}(x(1-\varepsilon)), \tag{5}$$

where l_{end} and w_{end} refer to the values at the last time step. The discretised value w_{end} needs to be extrapolated, this is done using spline interpolation and asymptotic terms w_0 and w_1 for "good enough" accuracy. The fluid balance is satisfied (see subsection 1.4). This operation essentially chops off closed fracture segment off the length εl, thus it is a discontinuous step, however the change is relatively small and insignificant. Final solution is obtained by combining all the outputs together. If fracture is to decrease significantly, this operation needs to be repeated multiple times, which might lead to high computational cost, as each backward step is of order ε. For this reason to compute possibly closing fractures it would be unwise to use $\varepsilon < 10^{-2}$ for closing fracture, but it is possible to use small ε when fracture is propagating, and switch to a larger value if a change of propagation regime is detected. Note that this approach should not be treated as a proper solution to closing fracture problem, it is a straightforward simplified extension to our propagating fracture model. Behavior of closing fracture is presented in Figure 3.

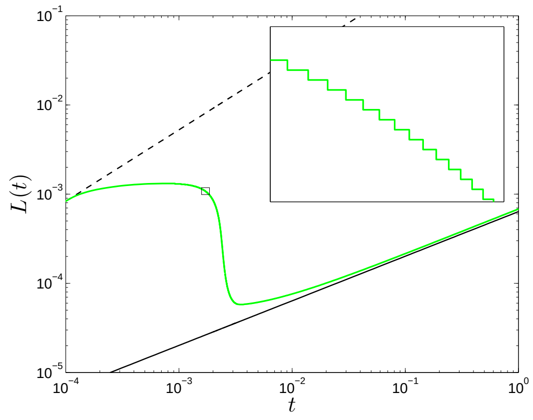

Figure 3: Close view on closing fracture length. Fracture is allowed to close by ε of its length, thus staircase pattern is formed.

1.4 Fluid balance check

Fluid balance equation is used to verify if solver output is valid:

$$\int_0^t q_0(t)dt = \int_0^1 w_*(x)dx + \int_0^t \int_0^1 w(t,x)dxdt + \int_0^t \int_0^1 q_l(t,x)dxdt. \tag{6}$$

A good way to integrate leak off function$\int_0^t \int_0^1 q_l(t,x)dxdt$ is to add an extra ode to ODE15s, lets call it:

$$\mathcal{C} = \int_0^1 q_l(t,x)dx \tag{7}$$

So then the derivative function can be extended by attaching $\mathcal{D}$:

$$\frac{dy}{dt} = \{\mathcal{A}, \mathcal{B}, \mathcal{C}\} \tag{8}$$