Report for the semester thesis "Development of a Monte Carlo algorithm for optimal control problems"

Stefano Weidmann
Institute of Fluid Dynamics
ETH Zurich
Zurich, Switzerland
stefanow@student.ethz.ch

Abstract—The Monte-Carlo adjoint algorithm described in [**?**] has been applied to a variant of the *quarter five spot configuration* found in petroleum engineering literature [**?**]. Results have been unsatisfactory due to lack of an appropriate preconditioner.

I. INTRODUCTION

This report aims to apply the Monte-Carlo adjoint algorithm described in [**?**] to a variant of problem known as *quarter five spot configuration* in petroleum engineering literature [**?**]. The set up is described in section II. Numerical experiments and a discussion follow in section IV. Possible ways how to proceed are given in section V. Derivatives needed for the implementation of the Monte-Carlo adjoint algorithm are given in the appendix in subsection A. The source code can be obtained by visiting the repository at https://gitlab.ethz.ch/stefanow/mcadjoint.

The main advantage of the Monte-Carlo adjoint algorithm in comparison to the better known *checkpointing* scheme proposed by Griewank and Walter [**?**] lies in its forward-in-time nature. Instead of having to store or recompute solutions, the adjoint equation can be solved in lockstep with the forward one. In principle, it is an application of the classic Monte-Carlo linear solver developed in [**?**] applied to the adjoint linear system of equations.

II. PROBLEM DESCRIPTION

We model a cross section of an oil field as a two dimensional square $\Omega := [0,1]^2$. In the field, there are two phases: water and oil. At the lower left corner $\boldsymbol{x}_{\text{drill}} := (0,0)$, we know the pressure $p_{\text{drill}}(t)$ and the water inflow $Q_{\text{w, drill}}(t)$. Opposite of that, at $\boldsymbol{x}_{\text{well}} := (1,1)$ a well is located. There we can measure the pressure $p_{\text{well}}(t)$ as well as the total volumetric outflow

$$Q_{\text{tot}}(t) := Q_{\text{o}}(t) + Q_{\text{w}}(t) = Q_{\text{w, drill}}(t) \tag{1}$$

per unit depth. There is no flow through the boundary of the domain except at the drill location $\boldsymbol{x}_{\text{drill}}$ and the well location $\boldsymbol{x}_{\text{well}}$.

The flow rates for both phases are described by Darcy's law

$$\boldsymbol{v}_{\text{w}} = -\frac{k k_{\text{rel, w}}}{\mu_{\text{w}}} \operatorname{grad}(p), \tag{2}$$

for water and

$$\boldsymbol{v}_{\text{o}} = -\frac{k k_{\text{rel, o}}}{\mu_{\text{o}}} \operatorname{grad}(p) \tag{3}$$

for oil. Here, $p(\boldsymbol{x}, t)$ is the pressure, μ_{o}, μ_{w} are dynamic viscosities for oil and water, $k(\boldsymbol{x}, t)$ is the permeability. $k_{\text{rel, o}}(S)$, $k_{\text{rel, w}}(S)$ are relative permeabilities and are assumed to depend quadratically on the saturation of water $S_{\text{w}} \in [0, 1]$ and the saturation of oil $S_{\text{o}} \in [0, 1]$:

$$k_{\text{rel, o}} = S_{\text{o}}^2 \tag{4}$$

$$k_{\text{rel, w}} = S_{\text{w}}^2. \tag{5}$$

The saturations are linked by the constitutive relation

$$S_{\text{o}} + S_{\text{w}} = 1. \tag{6}$$

$\boldsymbol{v}_{\text{o}}(\boldsymbol{x}, t)$, $\boldsymbol{v}_{\text{w}}(\boldsymbol{x}, t)$ finally are the volumetric flow rates per unit area (Darcy velocities).

The saturation S is not assumed constant but instead is transported according to the equation

$$\phi \frac{\partial}{\partial t} S_{\text{w}} + \operatorname{div}(\boldsymbol{v}_{\text{w}}) = q_{\text{w}}. \tag{7}$$

The term

$$q_{\text{w}} := Q_{\text{w}} \delta(\boldsymbol{x} - \boldsymbol{x}_{\text{well}}) \tag{8}$$

describes a line sink of water located at the well. Similarly, we use

$$q_{\text{o}} := Q_{\text{o}} \delta(\boldsymbol{x} - \boldsymbol{x}_{\text{well}}) \tag{9}$$

$$q_{\text{tot}} := q_{\text{o}} + q_{\text{w}}. \tag{10}$$

ϕ is the porosity of the rock which is assumed to be constant over the domain.

Conservation of the total mass then reads

$$\operatorname{div}(\boldsymbol{v}_{\text{tot}}) = q_{\text{tot}}, \tag{11}$$

where

$$\boldsymbol{v}_{\text{tot}} := \boldsymbol{v}_{\text{o}} + \boldsymbol{v}_{\text{w}} \tag{12}$$

is the total Darcy velocity.

We then introduce the mobilities

$$\lambda_{\text{o}} := \frac{k k_{\text{rel, o}}}{\mu_{\text{o}}} \tag{13}$$

$$\lambda_{\text{w}} := \frac{k k_{\text{rel, w}}}{\mu_{\text{w}}} \tag{14}$$

$$\lambda_{\text{tot}} := \lambda_{\text{o}} + \lambda_{\text{w}}. \tag{15}$$

Substituting in the λ from (13) into Darcy's law (3), (2) and adding both sides of the results we get the total Darcy's law

$$\boldsymbol{v}_{\text{tot}} = -\lambda_{\text{tot}} \operatorname{grad}(p). \tag{16}$$

Plugging this (16) into the conservation of mass (11) leads to the pressure equation

$$\boxed{\operatorname{div}(\lambda_{\text{tot}} \operatorname{grad}(p)) = -q_{\text{tot}}.} \tag{17}$$

Comparing the total Darcy's law (16) and the Darcy's law for water (2), we see that

$$\boldsymbol{v}_{\text{w}} = \frac{\lambda_{\text{w}}}{\lambda_{\text{tot}}} \boldsymbol{v}_{\text{tot}}. \tag{18}$$

We then plug in the model for the relative permeabilities in terms of the saturations (4), to get the final for of saturation transport equation

$$\boxed{\frac{\partial}{\partial t} S_{\text{w}} + \operatorname{div}\left(f(S_{\text{w}}) \boldsymbol{v}_{\text{tot}}\right) = \frac{q_{\text{w}}}{\phi},} \tag{19}$$

where f is the flux function

$$\boxed{f(S_{\text{w}}) := \frac{S_{\text{w}}^2 / \phi}{S_{\text{w}}^2 + (1 - S_{\text{w}})^2 \mu_{\text{w}} / \mu_{\text{o}}}.} \tag{20}$$

A. Boundary and initial conditions

We assume the initial saturation of water to be given, which is $S_{\text{w}}(\boldsymbol{x}, 0)$. For the pressure equation we use homogeneous Neumann boundary conditions (no flow), i.e.

$$\operatorname{grad}(p) = p_0 \begin{pmatrix} \delta(x - x_w) \\ \delta(y - y_w), \end{pmatrix}, \tag{21}$$

where we choose p_0 such that the compatibility condition

$$\int_{\Omega} q_{\text{tot}} \mathrm{d}A \overset{!}{=} \int_{\partial \Omega} \lambda_{\text{tot}} \operatorname{grad}(p)^{\mathsf{T}} \boldsymbol{n} \mathrm{d}l \tag{22}$$

is satisfied.

B. What to optimize?

To test the Monte-Carlo adjoint method, we want to match the pressure at the drill, i.e.

$$c(T) := \int_0^T \left(p_{\text{drill}}(t) - \tilde{p}_{\text{drill}}(t) \right)^2 \mathrm{d}t, \tag{23}$$

where the variables with a tilde denote computed quantities and T is a final time.

C. What do we control?

We control the log-permeabilities $\ln(k)$, as these are hard to measure. Controlling the logarithms $\ln(k)$ instead of just the permeabilities k has the added advantage of being able to cover a wider range of permeabilities and implicitly making sure that the permeabilities are always positive regardless of the choice of optimizer.

III. DISCRETIZATION

We discretize the square domain Ω with $N \times N$ square finite volumes, thus getting a mesh width of $h := 1/N$.

An overview of the discretization technique is given in the following procedure:

1) Solve the pressure equation (17) as detailed in subsection III-A
2) Compute the total Darcy velocity as in (16), using the same approximation of the gradient as in the first step
3) Solve the saturation equation (19) as in subsection III-B
4) Update the relative permeabilities according to (4) and repeat

A. Discretizing the pressure Poisson equation

Averaging the pressure Poisson equation (17) over such a finite volume K, and using the divergence theorem leads to

$$\frac{1}{h^2} \int_{\partial K} \lambda_{\text{tot}} \operatorname{grad}(p)^{\mathsf{T}} \boldsymbol{n} \mathrm{d}l = -\frac{1}{h^2} \int_K q_{\text{tot}} \mathrm{d}A. \tag{24}$$

Using the four boundaries North, East, South and West of the finite volume K, we approximate (24) as

$$\frac{1}{h} \left((\lambda_{\text{tot}} \frac{\delta}{\delta x} p)|_{\text{E}} - (\lambda_{\text{tot}} \frac{\delta}{\delta x} p)|_{\text{W}} \right.$$
$$\left. + (\lambda_{\text{tot}} \frac{\delta}{\delta y} p)|_{\text{N}} - \frac{\delta}{\delta y} p)|_{\text{S}} \right)$$
$$= -\frac{Q_{\text{tot}}}{h^2} \cdot \begin{cases} 1, & \text{if } K \text{ is the finite volume nearest to the well} \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

$\frac{\delta}{\delta \cdot}$ denote the standard finite difference quotients, i.e.

$$\frac{\delta}{\delta x} p|_{\text{E}} = \frac{1}{h} (p_R - p_K) \tag{26}$$

$$\frac{\delta}{\delta x} p|_{\text{W}} = \frac{1}{h} (p_K - p_D) \tag{27}$$

$$\frac{\delta}{\delta y} p|_{\text{N}} = \frac{1}{h} (p_U - p_K) \tag{28}$$

$$\frac{\delta}{\delta y} p|_{\text{S}} = \frac{1}{h} (p_K - p_D). \tag{29}$$

Here, U stands for the upper neighbor of K, D for the lower (down), R for the right and L for the left. p_K is the pressure at the center of the volume, which is taken to be the same as the volume averaged $\bar{p}$, as our scheme is just first order.

The total mobilities λ_{tot} at the boundaries are approximated by the harmonic mean of the total mobilities inside the adjacent finite volumes as

$$\lambda_{\text{tot}}|_{\text{E}} \approx \text{hm}(\lambda_{\text{tot}}|_K, \lambda_{\text{tot}}|_R) \tag{30}$$

$$\lambda_{\text{tot}}|_{\text{W}} \approx \text{hm}(\lambda_{\text{tot}}|_K, \lambda_{\text{tot}}|_L) \tag{31}$$

$$\lambda_{\text{tot}}|_{\text{N}} \approx \text{hm}(\lambda_{\text{tot}}|_K, \lambda_{\text{tot}}|_U) \tag{32}$$

$$\lambda_{\text{tot}}|_{\text{S}} \approx \text{hm}(\lambda_{\text{tot}}|_K, \lambda_{\text{tot}}|_D), \tag{33}$$

where

$$\text{hm}(a,b) = 2ab/(a+b). \tag{34}$$

We take the harmonic average instead of the arithmetic average for the following reasons as is detailed also in [?]. The total pressure drop over two adjacent cells is given as

$$\Delta p_{\text{tot}} = \Delta p_1 + \Delta p_2, \tag{35}$$

plugging in a simplified version of Darcy's law which reads

$$\Delta p = \frac{-v\mu h}{k}, \tag{36}$$

and canceling terms we have that

$$\frac{1}{k_{\text{avg}}} = \frac{1}{2}\left(\frac{1}{k_1} + \frac{1}{k_2}\right), \tag{37}$$

which is a harmonic average for k and so also for λ.

The discretized pressure Poisson equation reads

$$
\begin{aligned}
&T_E(p_K - p_R) + T_W(p_K - p_L) \\
&\quad + T_N(p_K - p_U) + T_S(p_K - p_D) \\
&= Q_{\text{tot}} \cdot \begin{cases} 1, & \text{if } K \text{ is the finite volume nearest to the well} \\ 0, & \text{otherwise} \end{cases},
\end{aligned}
\tag{38}
$$

where

$$T_N = \text{hm}(\lambda_{\text{tot}}|_K, \lambda_{\text{tot}}|_U), \tag{39}$$

and so on.

B. Discretizing the saturation equation

The finite volume reformulation of the saturation equation (19) leads to the following:

$$
\begin{aligned}
&\frac{\partial}{\partial t}\frac{1}{h^2}\int_K S_{\text{w}}\,dA \\
&\quad + \frac{1}{h}\Big((f(S_{\text{w}})v_{\text{tot},x})|_E - (f(S_{\text{w}})v_{\text{tot},x})|_W \\
&\qquad + (f(S_{\text{w}})v_{\text{tot},y})|_N - (f(S_{\text{w}})v_{\text{tot},y})|_S\Big) \\
&= \frac{Q_{\text{w}}}{h^2\phi}\cdot\begin{cases} 1, & \text{if } K \text{ is the finite volume nearest to the well} \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
\tag{40}
$$

We identify the volume average

$$\frac{1}{h^2}\int_K S_{\text{w}}\,dA =: S_K, \tag{41}$$

with the saturation at the center of the volume S_K. This is legit, as our scheme is just first order.

For the total Darcy velocity $\boldsymbol{v}_{\text{tot}}$ we use the same discretization of the pressure gradients as in the pressure equation, (26). This makes it available at the boundaries (N, S, E, W), as required by (40).

For the flux function f at the boundaries, we use an upwind discretization. The standard formulation can be simplified by noting that

$$\frac{\mathrm{d}}{\mathrm{d}S_{\text{w}}}f > 0 \tag{42}$$

and so instead of the advection velocity

$$\frac{\partial}{\partial S_{\text{w}}}(f \cdot \boldsymbol{v}_{\text{tot}}) \tag{43}$$

we can use the Darcy velocity $\boldsymbol{v}_{\text{tot}}$. We remember that this requires fixing the total Darcy velocity $\boldsymbol{v}_{\text{tot}}$ to be independent of the saturation S_{w}

For timestepping of the saturation equation (40), we use explicit Euler, as this simplifies the Jacobian used in the Monte-Carlo adjoint.

C. Discretizing the cost function

The cost function (23) is discretized as a sum of squares over the timesteps, i.e.

$$F := \sum_{i=1}^{n}(p_{\text{drill}}(i\Delta t) - p^{(i)}_{\text{drill cell}})^2, \tag{44}$$

where $p_{\text{drill cell}}$ is the computed pressure at the drill cell, p_{drill} is the provided pressure at the drill in the i-th timelevel, and Δt is the timestep.

D. Formal definition of the residuals for the Monte-Carlo adjoint solver

First, we define the pressure residuals

$$
\Pi^{(i)}_K := \begin{cases} p^{(i)}_K - p_{\text{well}}(i\Delta t), & \text{if } K \text{ is the cell} \\ & \text{nearest to the well} \\ T^{(i-1)}_E(p^{(i)}_K - p^{(i)}_R) & \\ +T^{(i-1)}_W(p^{(i)}_K - p^{(i)}_L) & \\ +T_N(p^{(i)}_K - p^{(i)}_U) & \\ +T^{(i-1)}_S(p^{(i)}_K - p^{(i)}_D) - Q^{(i-1)}_{\text{tot, K}}, & \text{else} \end{cases}
\tag{45}
$$

where

$$Q^{(i)}_{\text{tot, K}} := Q_{\text{tot}}(i\Delta t)\cdot\begin{cases} 1, & \text{if } K \text{ is the cell nearest to the drill} \\ 0, & \text{otherwise} \end{cases} \tag{46}$$

The saturation residuals are given by

$$
\begin{aligned}
\Sigma^{(i)}_K := &\ S^{(i)}_{\text{w}}|_K - S^{(i-1)}_{\text{w}}|_K \\
&+ \frac{\Delta t}{h}\Big((f(S^{(i-1)}_{\text{w}})v^{(i)}_{\text{tot},x})|_E - (f(S^{(i-1)}_{\text{w}})v^{(i)}_{\text{tot},x})|_W \\
&+ (f(S^{(i-1)}_{\text{w}})v^{(i)}_{\text{tot},y})|_N - (f(S^{(i-1)}_{\text{w}})v^{(i-1)}_{\text{tot},y})|_S\Big) - \Delta t Q^{(i)}_{\text{tot, K}}.
\end{aligned}
\tag{47}
$$

Here a superscript of (i) denotes the values in the i-th timelevel and a subscript of K the values in the cell K.

E. The adjoint system of equations

The full adjoint system of equations then has the following form:

$$\underbrace{\begin{pmatrix} \boldsymbol{J}_{11}^{\mathsf{T}} & \boldsymbol{J}_{21}^{\mathsf{T}} & & & \\ & \boldsymbol{J}_{22}^{\mathsf{T}} & \boldsymbol{J}_{32}^{\mathsf{T}} & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{J}_{(n-1)(n-1)}^{\mathsf{T}} & \boldsymbol{J}_{n(n-1)}^{\mathsf{T}} \\ & & & & \boldsymbol{J}_{nn} \end{pmatrix}}_{:=\boldsymbol{A}} \boldsymbol{\psi} = \underbrace{\begin{pmatrix} \boldsymbol{b}^{(1)} \\ \boldsymbol{b}^{(2)} \\ \vdots \\ \boldsymbol{b}^{(n-1)} \\ \boldsymbol{b}^{(n)} \end{pmatrix}}_{:=\boldsymbol{b}}. \tag{48}$$

We defined the jacobian block matrices

$$\boldsymbol{J}_{ii} := \begin{pmatrix} \frac{\partial}{\partial \boldsymbol{p}^{(i)}} \boldsymbol{\Pi}^{(i)} & \boldsymbol{0} \\ \frac{\partial}{\partial \boldsymbol{p}^{(i)}} \boldsymbol{\Sigma}^{(i)} & \frac{\partial}{\partial \boldsymbol{S}_{\mathrm{w}}^{(i)}} \boldsymbol{\Sigma}^{(i)} \end{pmatrix} \tag{49}$$

$$\boldsymbol{J}_{i(i-1)} := \begin{pmatrix} \boldsymbol{0} & \frac{\partial}{\partial \boldsymbol{S}_{\mathrm{w}}^{(i-1)}} \boldsymbol{\Pi}^{(i)} \\ \boldsymbol{0} & \frac{\partial}{\partial \boldsymbol{S}_{\mathrm{w}}^{(i-1)}} \boldsymbol{\Sigma}^{(i)} \end{pmatrix}, \tag{50}$$

where the expressions for the jacobians of the residuals at timelevel (i), $\boldsymbol{\Pi}^{(i)}, \boldsymbol{\Sigma}^{(i)}$, can be found in the appendix in subsection A.

Also,

$$\boldsymbol{b}^{(i)} := \left(\left(\frac{\partial}{\partial \boldsymbol{p}^{(i)}} F \right)^{\mathsf{T}} \\ \boldsymbol{0} \right), \tag{51}$$

where F is the discretized cost function and

$$\frac{\partial}{\partial p_K^{(i)}} F = \begin{cases} 2(p_K^{(i)} - p_{\mathrm{drill}}(i\Delta t)) & \text{if } K \text{ is next to the drill} \\ 0 & \text{otherwise.} \end{cases} \tag{52}$$

IV. NUMERICAL EXPERIMENTS

A. Preconditioning

The Monte-Carlo adjoint solver relies on a Neumann series representation of the adjoint system matrix A. For this series to converge, the spectral radius (maximum magnitude of eigenvalues) of the iteration matrix

$$\boldsymbol{I} - \boldsymbol{P}\boldsymbol{A}, \tag{53}$$

has to be less than one. $\boldsymbol{P}$ stands for a freely choosable invertible matrix, a preconditioner. This condition has been achieved by setting

$$\boldsymbol{P} := \boldsymbol{D} \operatorname{diag}\left(\left(\frac{\partial}{\partial \boldsymbol{p}^{(1)}} \boldsymbol{\Pi}^{(1)} \right)^{-\mathsf{T}}, \dots, \left(\frac{\partial}{\partial \boldsymbol{p}^{(n)}} \boldsymbol{\Pi}^{(n)} \right)^{-\mathsf{T}} \right). \tag{54}$$

The diag denotes a (here block-) diagonal matrix where the blocks on the diagonal are given in order by its arguments. By virtue of the discretization,

$$\left(\frac{\partial}{\partial \boldsymbol{p}^{(i)}} \boldsymbol{\Pi}^{(i)} \right)^{-\mathsf{T}} \tag{55}$$

is the same system matrix as the one used to solve the pressure Poisson matrix and thus a factorization is already available. However, the resulting matrices

$$\left(\frac{\partial}{\partial \boldsymbol{p}^{(1)}} \boldsymbol{\Pi}^{(1)} \right)^{-\mathsf{T}} \left(\frac{\partial}{\partial \boldsymbol{p}^{(i)}} \boldsymbol{\Sigma}^{(i)} \right)^{\mathsf{T}} \tag{56}$$

are dense at least at later timesteps, where the water saturations are higher. This requires a certain additional amount of storage which limits the usefulness of the preconditioner.

The $\boldsymbol{D}$ matrices is diagonal too and serves to reduce the 1-norms of the row vectors of the adjoint matrix $\boldsymbol{A}$. This is necessary because of a central result in the original paper [?]: If any of the row sum norms

$$\|\boldsymbol{A}_{i,:}\|_1 = \sum_j |A_{ij}|, \tag{57}$$

is greater than one, the variance of the random walks can grow exponentially. Thus we choose

$$D_{ll} := \frac{1}{\max\{1, \|\boldsymbol{A}_{l,:}\|_1\}}. \tag{58}$$

This however doesn't fulfill the required condition of

$$\|\boldsymbol{I} - \boldsymbol{P}\boldsymbol{A}\|_\infty < 1, \tag{59}$$

where $\|\cdot\|_\infty$ is the maximum row sum norm. Instead, we have

$$\|(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{A})_{l,:}\|_1 = |1 - \frac{1}{D_{ll}}| + \sum_{j \neq l} \frac{|(PA)_{(il)}|}{D_{ll}} \leq 2(|1 - \frac{1}{D_{ll}}|) < 2, \tag{60}$$

which is not the required value of 1. However, the Neumann series surely converges, because $\boldsymbol{I} - \boldsymbol{P}\boldsymbol{A}$ is upper triangular with positive entries on the diagonal, which are all smaller than one. This means that all eigenvalues lie inside the unit circle and thus the spectral radius is smaller than one.

B. Regularization

To avoid huge jumps in the permeabilities k, a regularization penalty has been added to the cost function. This term has the form

$$\alpha \sum_{\text{cell } K} (k_{\mathrm{ref}} - k|_K)^2. \tag{61}$$

α is a tunable positive parameter and k_{ref} is a reference permeability. This reference permeability is the value it should have so that the prescribed and computed pressures at the drill match, assuming that the permeabilities were homogeneous across the whole domain Ω and if there was no water in the field, just oil. However, this regularization turned out not to be necessary in the tested cases.

C. Smoothing the gradient

Another technique to improve optimization is by smoothing the noisy gradient which Monte-Carlo yields, as is done in the original paper [?]. This can be done by noticing the symmetry of the problem. The diagonal from the drill cell to the well cell serves as mirror line. When we start the optimizer with an initial permeability field respecting this symmetry, all
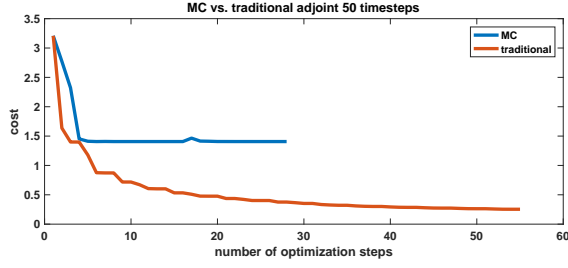
Fig. 1. Sample run of the Monte-Carlo based optimizer, in orange (lower line) the optimization using the exact adjoint, in blue (upper line) the one approximating the gradient with Monte-Carlo.

subsequently calculated gradients $\left(\frac{\mathrm{d}}{\mathrm{d}\ln(\boldsymbol{k})}F\right)^{\mathsf{T}}$ should also be symmetric.

This can be exploited if one of the two symmetric gradient entries is way off the mean gradient entry, for instance because the corresponding random walks suffered from exponentially increasing variance. Then we can just replace it with its symmetric counterpart or a combination of too. However, this technique wasn't found to be useful because either both components were found to be off in experiments or none.

Also exponential smoothing of the form

$$\frac{\mathrm{d}}{\mathrm{d}\ln(k)|_K}F := \beta \cdot \frac{\mathrm{d}}{\mathrm{d}\ln(k)|_K}F + (1-\beta)\mu \qquad (62)$$

has been tried. Here, $\beta \in [0,1]$ is a smoothing parameter and

$$\mu = \frac{1}{N^2 - 1}\sum_{K' \neq K}\frac{\mathrm{d}}{\mathrm{d}\ln(k)|'_K}F \qquad (63)$$

is the mean gradient entry over the other cells. This proved to be more effective than symmetrizing the gradient but not sufficient for a better optimization.

D. Sample optimization run

A sample optimization run with just 50 timesteps and a 3×3 and 10^6 random walks per time step and mesh cell reveals that the Monte-Carlo adjoint optimizer is not up to the task, as can be seen in figure 1. The optimizer used was a gradient descent with a Moré-Thuente line search [?], As birth and transition probabilities we used the almost optimal values described by [?], which amounts to setting them to be proportional to the magnitude of corresponding matrix entries in a given row. No regularization has been used in this case. Other parameters as for instance the dynamic viscosities μ are given directly in the source code available at https://gitlab.ethz.ch/stefanow/mcadjoint. The Monte-Carlo based optimization stopped early because of a small gradient. Another noteworthy characteristic is that the optimization problem seems to be hard: The difference between Monte-Carlo and exact optimal costs is just one. In conclusion, the preconditioner given by (54) is not satisfactory.

V. POSSIBLE SOLUTIONS

A possible solution of the preconditioning dilemma could lie in using *sequential Monte Carlo* as described in [?],

which relies on repeated application of the *control variates* technique [?] known for reducing the variance in Monte-Carlo simulations. This variant requires the eigenvalues of

$$|\boldsymbol{I} - \boldsymbol{P}\boldsymbol{A}|, \qquad (64)$$

to lie inside the unit circle where $|\cdot|$ denotes a componentwise magnitude. This is fulfilled for our choice of preconditioner $\boldsymbol{P}$. Moreover, sequential Monte-Carlo guarantees the variance of the random walks to be exponentially decreasing with the number of iterations, as long as the number of random walks per iteration is high enough.

A recent alternative seems to be [?], where the transition matrix of the random walks can vary over time. This approach requires the same prerequisites as sequential Monte-Carlo, but needs to have the full iteration matrix $\boldsymbol{I} - \boldsymbol{P}\boldsymbol{A}$ to build the transition hypermatrix. This is not feasible for the adjoint system of equations and would need a workaround.

APPENDIX

A. Computing the derivatives for the Monte-Carlo adjoint solver

The nonzero derivatives for the diagonal blocks are given by

$$\frac{\partial}{\partial p_K^{(i)}}\Pi_K^{(i)} = T_E^{(i-1)} + T_W^{(i-1)} + T_N^{(i-1)} + T_S^{(i-1)} \qquad (65)$$

$$\frac{\partial}{\partial p_R^{(i)}}\Pi_K^{(i)} = -T_E^{(i-1)} \qquad (66)$$

$$\frac{\partial}{\partial p_L^{(i)}}\Pi_K^{(i)} = -T_W^{(i-1)} \qquad (67)$$

$$\frac{\partial}{\partial p_N^{(i)}}\Pi_K^{(i)} = -T_U^{(i-1)} \qquad (68)$$

$$\frac{\partial}{\partial p_S^{(i)}}\Pi_K^{(i)} = -T_D^{(i-1)} \qquad (69)$$

for cells which are not nearest to the well

and by

$$\frac{\partial}{\partial S_K^{(i)}} \Sigma_K^{(i)} = 1 \tag{70}$$

$$\frac{\partial}{\partial S_{wK}^{(i-1)}} \Sigma_K^{(i)} = -1 + f'(S_{wK}^{(i-1)}) \frac{\Delta t}{h} \cdot \Big(\mathbb{1}(v_{\text{tot, x}}|_E > 0) v_{\text{tot, x}}|_E \tag{71}$$

$$- \mathbb{1}(v_{\text{tot, x}}|_W < 0) v_{\text{tot, x}}|_W + \mathbb{1}(v_{\text{tot, y}}|_N > 0) v_{\text{tot, y}}|_N$$

$$- \mathbb{1}(v_{\text{tot, y}}|_S < 0) v_{\text{tot, y}}|_S \Big)$$

$$+ \frac{\Delta t}{h} \cdot \Big(- f(S_{wK}^{(i-1)})|_E \frac{\partial}{\partial b} \text{hm}($$

$$\lambda_{\text{tot, R}}, \lambda_{\text{tot, K}}) \frac{\partial}{\partial S_w} \lambda|_K \frac{\delta}{\delta x} p|_E \pm \dots \Big)$$

$$\frac{\partial}{\partial S_{wR}^{(i-1)}} \Sigma_K^{(i)} = f'(S_{wR}^{(i-1)}) \frac{\Delta t}{h} \cdot \mathbb{1}(v_{\text{tot, x}}|_E < 0) v_{\text{tot, x}}|_E$$

$$- \frac{\Delta t}{h} \frac{\partial}{\partial b} \text{hm}(\lambda_{\text{tot, K}}, \lambda_{\text{tot, R}}) \frac{\partial}{\partial S_w} \lambda|_R \frac{\delta}{\delta x} p|_E \tag{72}$$

where

$$f'(S_w) = \frac{2\mu_o \mu_w (1 - S_w) S_w}{\phi \cdot (\mu_w (1 - S_w)^2 + \mu_o S_w^2)^2} \tag{73}$$

$$\mathbb{1}(b) = \begin{cases} 1, & \text{if } b \text{ true} \\ 0, & \text{otherwise.} \end{cases} \tag{74}$$

For the off diagonal blocks, we have

$$\frac{\partial}{\partial p_K^{(i)}} \Sigma_K^{(i)} = + \frac{\Delta t}{h^2} \Big(T_E^{(i-1)} f(S_w^{(i-1)})|_E \tag{75}$$

$$+ T_W^{(i-1)} f(S_w^{(i-1)})|_W$$

$$+ T_N^{(i-1)} f(S_w^{(i-1)})|_N$$

$$+ T_S^{(i-1)} f(S_w^{(i-1)})|_S \Big)$$

$$\frac{\partial}{\partial p_R^{(i)}} \Sigma_K^{(i)} = - \frac{\Delta t}{h^2} T_E^{(i-1)} f(S_w^{(i-1)})|_E \tag{76}$$

$$\frac{\partial}{\partial p_L^{(i)}} \Sigma_K^{(i)} = - \frac{\Delta t}{h^2} T_W^{(i-1)} f(S_w^{(i-1)})|_W \tag{77}$$

$$\tag{78}$$

$$\frac{\partial}{\partial p_U^{(i)}} \Sigma_K^{(i)} = - \frac{\Delta t}{h^2} T_N^{(i-1)} f(S_w^{(i-1)})|_N \tag{79}$$

$$\frac{\partial}{\partial p_D^{(i)}} \Sigma_K^{(i)} = - \frac{\Delta t}{h^2} T_S^{(i-1)} f(S_w^{(i-1)})|_S, \tag{80}$$

for cells not nearest to the well and

$$\frac{\partial}{\partial S_{wK}^{(i-1)}} \Pi_K^{(i)} = 2 \frac{\partial}{\partial S_w} \lambda_{\text{tot}}^{(i-1)}|_K \tag{81}$$

$$\sum_{n \in \{U,D,L,R\}} \frac{(p_K^{(i-1)} - p_n^{(i-1)}) \cdot (\lambda_{\text{tot, n}}^{(i-1)})^2}{(\lambda_{\text{tot, K}}^{(i-1)} + \lambda_{\text{tot, n}}^{(i-1)})^2}$$

$$\frac{\partial}{\partial S_{wR}^{(i-1)}} \Pi_K^{(i)} = 2 \frac{\partial}{\partial S_w} \lambda_{\text{tot}}^{(i-1)}|_R \cdot \frac{(p_K^{(i-1)} - p_R^{(i-1)}) \cdot (\lambda_{\text{tot, K}}^{(i-1)})^2}{(\lambda_{\text{tot, K}}^{(i-1)} + \lambda_{\text{tot, R}}^{(i-1)})^2} \tag{82}$$

$$\frac{\partial}{\partial S_{wL}^{(i-1)}} \Pi_K^{(i)} = 2 \frac{\partial}{\partial S_w} \lambda_{\text{tot}}^{(i-1)}|_L \cdot \frac{(p_K^{(i-1)} - p_L^{(i-1)}) \cdot (\lambda_{\text{tot, K}}^{(i-1)})^2}{(\lambda_{\text{tot, K}}^{(i-1)} + \lambda_{\text{tot, L}}^{(i-1)})^2} \tag{83}$$

$$\frac{\partial}{\partial S_{wU}^{(i-1)}} \Pi_K^{(i)} = 2 \frac{\partial}{\partial S_w} \lambda_{\text{tot}}^{(i-1)}|_U \cdot \frac{(p_K^{(i-1)} - p_U^{(i-1)}) \cdot (\lambda_{\text{tot, K}}^{(i-1)})^2}{(\lambda_{\text{tot, K}}^{(i-1)} + \lambda_{\text{tot, U}}^{(i-1)})^2} \tag{84}$$

$$\frac{\partial}{\partial S_{wD}^{(i-1)}} \Pi_K^{(i)} = 2 \frac{\partial}{\partial S_w} \lambda_{\text{tot}}^{(i-1)}|_D \cdot \frac{(p_K^{(i-1)} - p_D^{(i-1)}) \cdot (\lambda_{\text{tot, K}}^{(i-1)})^2}{(\lambda_{\text{tot, K}}^{(i-1)} + \lambda_{\text{tot, D}}^{(i-1)})^2} \tag{85}$$

where

$$\frac{\partial}{\partial S_w} \lambda_{\text{tot}} = 2k \left(\frac{S_w - 1}{\mu_o} + \frac{S_w}{\mu_w} \right). \tag{86}$$

For solving the adjoint equations, we also need the derivatives of the residuals with respect to the parameters, in our case $\ln(k)$.

For the pressure residuals and if K is not the cell at the well, those read

$$\frac{\partial}{\partial \ln(k)_K} \Pi_K^{(i)} = \lambda^{(i-1)}|_K \sum_{n \in \{L,R,U,D\}} (p_K^{(i)} - p_n^{(i)}) \frac{\partial}{\partial b} \text{hm}(\lambda^{(i-1)}|_n, \tag{87}$$

$$\lambda_{\text{tot}}^{(i-1)}|_K)$$

$$\frac{\partial}{\partial \ln(k)_L} \Pi_K^{(i)} = (p_K^{(i)} - p_L^{(i)}) \lambda_{\text{tot}}^{(i-1)}|_L \frac{\partial}{\partial b} \text{hm}(\lambda_{\text{tot}}^{(i-1)}|_K, \lambda_{\text{tot}}^{(i-1)}|_L) \tag{88}$$

$$\frac{\partial}{\partial \ln(k)_R} \Pi_K^{(i)} = (p_K^{(i)} - p_R^{(i)}) \lambda_{\text{tot}}^{(i-1)}|_R \frac{\partial}{\partial b} \text{hm}(\lambda_{\text{tot}}^{(i-1)}|_K, \lambda_{\text{tot}}^{(i-1)}|_R) \tag{89}$$

$$\frac{\partial}{\partial \ln(k)_U} \Pi_K^{(i)} = (p_K^{(i)} - p_U^{(i)}) \lambda_{\text{tot}}^{(i-1)}|_U \frac{\partial}{\partial b} \text{hm}(\lambda_{\text{tot}}^{(i-1)}|_K, \lambda_{\text{tot}}^{(i-1)}|_U) \tag{90}$$

$$\frac{\partial}{\partial \ln(k)_D} \Pi_K^{(i)} = (p_K^{(i)} - p_D^{(i)}) \lambda_{\text{tot}}^{(i-1)}|_D \frac{\partial}{\partial b} \text{hm}(\lambda_{\text{tot}}^{(i-1)}|_K, \lambda_{\text{tot}}^{(i-1)}|_D) \tag{91}$$

where

$$\frac{\partial}{\partial b} \text{hm}(a, b) := \frac{2a^2}{(a + b)^2}. \tag{92}$$

On the other hand, the saturation residuals Σ_K lead to the following derivatives

$$\frac{\partial}{\partial\ln(k)_K}\Sigma_K = \lambda_{\mathrm{tot}}^{(i-1)}|_K \frac{\Delta t}{h}\Bigg(\tag{93}$$

$$(f(S_{\mathrm{w}}^{(i-1)})v_{\mathrm{tot},x}^{(i-1)})|_E \frac{\partial}{\partial b}\mathrm{hm}(\lambda_{\mathrm{tot}}^{(i-1)}|_E, \lambda_{\mathrm{tot}}^{(i-1)}|_K)$$

$$\cdot(-\frac{\delta}{\delta x}p|_E)$$

$$-(f(S_{\mathrm{w}}^{(i-1)})v_{\mathrm{tot},x}^{(i-1)})|_W \frac{\partial}{\partial b}\mathrm{hm}(\lambda_{\mathrm{tot}}^{(i-1)}|_W, \lambda_{\mathrm{tot}}^{(i-1)}|_K)$$

$$\cdot(-\frac{\delta}{\delta x}p|_W)$$

$$+(f(S_{\mathrm{w}}^{(i-1)})v_{\mathrm{tot},y}^{(i-1)})|_N \frac{\partial}{\partial b}\mathrm{hm}(\lambda_{\mathrm{tot}}^{(i-1)}|_N, \lambda_{\mathrm{tot}}^{(i-1)}|_K)$$

$$\cdot(-\frac{\delta}{\delta y}p|_N)$$

$$-(f(S_{\mathrm{w}}^{(i-1)})v_{\mathrm{tot},y}^{(i-1)})|_S \frac{\partial}{\partial b}\mathrm{hm}(\lambda_{\mathrm{tot}}^{(i-1)}|_E, \lambda_{\mathrm{tot}}^{(i-1)}|_S)$$

$$\cdot(-\frac{\delta}{\delta y}p|_S)$$

$$\Bigg)$$

$$\frac{\partial}{\partial\ln(K)_R}\Sigma_K = \lambda_{\mathrm{tot}}^{(i-1)}|_R \frac{\Delta t}{h}f(S_{\mathrm{w}}^{(i-1)})v_{\mathrm{tot},x}^{(i-1)})|_E$$

$$\cdot\frac{\partial}{\partial b}\mathrm{hm}(\lambda_{\mathrm{tot}}^{(i-1)}|_K, \lambda_{\mathrm{tot}}^{(i-1)}|_E)(-\frac{\delta}{\delta x}p|_E) \tag{94}$$

$$\frac{\partial}{\partial\ln(K)_L}\Sigma_K = -\lambda_{\mathrm{tot}}^{(i-1)}|_L \frac{\Delta t}{h}f(S_{\mathrm{w}}^{(i-1)})v_{\mathrm{tot},x}^{(i-1)})|_W$$

$$\cdot\frac{\partial}{\partial b}\mathrm{hm}(\lambda_{\mathrm{tot}}^{(i-1)}|_K, \lambda_{\mathrm{tot}}^{(i-1)}|_L)\cdot(-\frac{\delta}{\delta x}p|_W) \tag{95}$$

$$\frac{\partial}{\partial\ln(K)_U}\Sigma_K = \lambda_{\mathrm{tot}}^{(i-1)}|_U \frac{\Delta t}{h}f(S_{\mathrm{w}}^{(i-1)})v_{\mathrm{tot},x}^{(i-1)})|_N$$

$$\cdot\frac{\partial}{\partial b}\mathrm{hm}(\lambda_{\mathrm{tot}}^{(i-1)}|_K, \lambda_{\mathrm{tot}}^{(i-1)}|_U)\cdot(-\frac{\delta}{\delta y}p|_N) \tag{96}$$

$$\frac{\partial}{\partial\ln(K)_D}\Sigma_K = -\lambda_{\mathrm{tot}}^{(i-1)}|_D \frac{\Delta t}{h}f(S_{\mathrm{w}}^{(i-1)})v_{\mathrm{tot},x}^{(i-1)})|_S$$

$$\cdot\frac{\partial}{\partial b}\mathrm{hm}(\lambda_{\mathrm{tot}}^{(i-1)}|_K, \lambda_{\mathrm{tot}}^{(i-1)}|_D)(-\frac{\delta}{\delta y}p|_S) \tag{97}$$

REFERENCES

[1] Q. Wang, D. Gleich, A. Saberi, N. Etemadi, and P. Moin, "A monte carlo method for solving unsteady adjoint equations," *Journal of Computational Physics*, vol. 227, no. 12, pp. 6184 – 6205, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999108001393

[2] C.-Y. Chen and E. Meiburg, "Miscible porous media displacements in the quarter five-spot configuration. part 1. the homogeneous case," *Journal of Fluid Mechanics*, vol. 371, p. 233–268, 1998.

[3] A. Griewank and A. Walther, "Algorithm 799: Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation," *ACM Trans. Math. Softw.*, vol. 26, no. 1, pp. 19–45, Mar. 2000. [Online]. Available: http://doi.acm.org/10.1145/347837.347846

[4] G. E. Forsythe and R. A. Leibler, "Matrix inversion by a monte carlo method," *Mathematics of Computation*, vol. 4, no. 31, pp. 127–127, sep 1950. [Online]. Available: https://doi.org/10.1090/s0025-5718-1950-0038138-x

[5] P. Inc., *Harmonic-Average Permeability*, 2018 (accessed May 30, 2018). [Online]. Available: http://perminc.com/resources/fundamentals-of-fluid-flow-in-porous-media/chapter-2-the-porous-medium/permeability/harmonic-average-permeability/

[6] J. J. Moré and D. J. Thuente, "Line search algorithms with guaranteed sufficient decrease," *ACM Trans. Math. Softw.*, vol. 20, pp. 286–307, 1994.

[7] H. J. H., *mcma*, 2008, vol. 14, ch. Sequential Monte Carlo for linear systems - a practical summary, p. 1, 1.

[8] D. P. Kroese, T. Taimre, and Z. Botev, "Handbook of monte carlo methods," 09 2011.