Universität Münster
Fachbereich Mathematik und Informatik

Masterarbeit Mathematik

Machine learning based surrogate modeling to accelerate parabolic PDE constrained optimization

vorgelegt am: 9. Juli 2024
von: Andy Kevin Wert
Matrikelnummer: 461478
Erstgutachter: Prof. Dr. Mario Ohlberger
Zweitgutachter: Dr. Stephan Rave

Contents

1 Introduction **3**

2 Parabolic optimal control problems **4**
2.1 Introduction to the problem 4
2.2 Finite element discretization 5
2.2.1 Discretization in space 5
2.2.2 Discretization in time 6
2.2.3 Crank-Nicolson scheme 7
2.2.4 Calculation of the objective function value 7
2.3 Optimization of the control variable 8

3 Ensemble-based optimization algorithm **9**

4 Adaptive-ML-EnOpt algorithm **12**
4.1 Deep neural networks 12
4.2 Modifying the EnOpt algorithm by using a neural network-based surrogate 12

5 Numerical experiments **13**

Bibliography **14**

1 Introduction

[1]

2 Parabolic optimal control problems

2.1 Introduction to the problem

Our optimization problem is based on the problem presented in [2]. We consider a state variable u and a control variable q, defined on $(0,T)\times\Omega$ with $T\in\mathbb{R}$ and $\Omega\subset\mathbb{R}^n$. Our goal is to minimize the function

$$J(q,u)=\frac{1}{2}\int_0^T\int_\Omega (u(t,x)-\hat{u}(t,x))^2\,\mathrm{d}x\,\mathrm{d}t+\frac{\alpha}{2}\int_0^T\int_\Omega q(t,x)^2\,\mathrm{d}x\,\mathrm{d}t \tag{2.1a}$$

subject to the constraints

$$\begin{aligned}\partial_t u-\Delta u&=f+q \quad \text{in } (0,T)\times\Omega,\\ u(0)&=u_0 \qquad\qquad \text{in } \Omega,\end{aligned} \tag{2.1b}$$

with homogeneous Dirichlet boundary conditions on $(0,T)\times\partial\Omega$.

Let $V=H_0^1(\Omega)$, $H=L^2(\Omega)$ and $I=(0,T)$. We define our state space as

$$X:=\{v\mid v\in L^2(I,V) \text{ and } \partial_t v\in L^2(I,V^*)\}$$

and the control space as

$$Q=L^2(I,L^2(\Omega)).$$

The notion of the inner products and norms on $L^2(\Omega)$ and $L^2(I,L^2(\Omega))$ is introduced as

$$\begin{aligned}(v,w)&:=(v,w)_{L^2(\Omega)}, & (v,w)_I&:=(v,w)_{L^2(I,L^2(\Omega))}\\ \|v\|&:=\|v\|_{L^2(\Omega)}, & \|v\|_I&:=\|v\|_{L^2(I,L^2(\Omega))}.\end{aligned}$$

Using this inner product, the weak form of the state equations (2.1b) for $q,f\in Q$ and $u_0\in V$ is given as

$$\begin{aligned}(\partial_t u,\phi)+(\nabla u,\nabla\phi)&=(f+q,\phi) \quad \forall\phi\in X,\\ u(0)&=u_0 \qquad\qquad \text{in } \Omega.\end{aligned} \tag{2.2}$$

With the weak state equations (2.2), we define the weak formulation of the optimal control problem (2.1) as

$$\text{Minimize } J(q,u):=\frac{1}{2}\|u-\hat{u}\|_I^2+\frac{\alpha}{2}\|q\|_I^2 \text{ subject to (2.2) and } (q,u)\in Q\times X. \tag{2.3}$$

Now, we cite two results of the problems (2.2) and (2.3).

Proposition 2.1 ([2])**.** *For fixed $q,f\in Q$, and $u_0\in V$ there exists a unique solution $u\in X$ of problem* (2.2)*. Moreover, the solution exhibits the improved regularity*

$$u\in L^2(I,H^2(\Omega)\cap V)\cap H^1(I,L^2(\Omega))\hookrightarrow C(\bar{I},V).$$

It holds the stability estimate

$$\|\partial_t u\|_I+\|\nabla^2 u\|_I\le C\{\|f+q\|_I+\|\nabla u_0\|\}.$$

Proposition 2.2 ([2]). *For given $f, \hat{u} \in L^2(I,H)$, $u_0 \in V$, and $\alpha > 0$, the optimal control Problem (2.3) admits a unique solution $(\bar{q}, \bar{u}) \in Q \times X$. The optimal control $\bar{q}$ posesses the regularity*

$$\bar{q} \in L^2(I, H^2(\Omega)) \cap H^1(I, L^2(\Omega)).$$

Due to the existence and uniqueness results from Proposition 2.1, we define $u(q)$ as the unique solution of (2.2) with respect to some $q \in Q$. This enables us to define a reduced cost functional $j : Q \to \mathbb{R}$ that is only dependent on the control q as

$$j(q) := J(q, u(q)).$$

From now on, the optimal control problem that we examine is:

$$\text{Minimize } j(q) \text{ subject to } q \in Q. \tag{2.4}$$

2.2 Finite element discretization

In order to solve the optimization problem (2.4) numerically, the discretization of our model is now discussed. We begin with the presentation of the discretization in space with a n-D continuous Galerkin method. Then, we look at the discretization in time, which is done with a 1D continuous Galerkin method.

2.2.1 Discretization in space

The discretization in space is shown on a 2-dimensional rectangular space $\Omega \subset \mathbb{R}^2$ with linear finite elements. We assume to have a vertex set $\mathcal{V} = (x_1, \ldots, x_N) \in (\mathbb{R}^2)^N$ with a convex hull that is equal to $\bar{\Omega}$ and $x_i \neq x_j$ for all $i \neq j$ in $\{1, \ldots, N\}$. Let $\hat{T} = \{(x,y) \in [0,1]^2 \mid y \leq 1-x\}$ be the reference triangle. Then,

$$\theta_l(\xi) = x_{l_1} + D\theta_l \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} \text{ with } D\theta_l = \begin{pmatrix} x_{l_2} - x_{l_1} & x_{l_3} - x_{l_1} \end{pmatrix}$$

is a transformation from the reference triangle $\hat{T}$ to some other triangle T_l with the corners $x_{l_1}, x_{l_2}, x_{l_3}$.

We define now a mesh $\mathcal{T} = \{T_l\}$ which consists of triangles $T_l = \theta_l(\hat{T})$, where $T_l \cap T_m$ for $T_l, T_m \in \mathcal{T}$ is either a common side/corner or empty and where $\bar{\Omega} = \cup_{T_l \in \mathcal{T}} T_l$. We also assume that every vertex in $\mathcal{V}$ is a corner of at least one triangle of $\mathcal{T}$.

Let $\mathcal{P}_1(\hat{T}, \mathbb{R})$ be the space of polynomials up to order 1 in $\hat{T}$. Then, $\{\psi_1, \psi_2, \psi_3\}$ with $\psi_1(\xi) = 1 - \xi_1 - \xi_2, \psi_2(\xi) = \xi_1, \psi_3(\xi) = \xi_2$ defines a basis of $\mathcal{P}_1(\hat{T}, \mathbb{R})$. Using this basis, we set

$$V_s = \text{span}\{\phi_i, i = 0, \ldots, N\} \cap V$$

as the finite element space of our state variables and

$$V_c = \text{span}\{\phi_i, i = 0, \ldots, N\}$$

as the finite element space of our control variables with

$$\phi_i|_{T_l} = \begin{cases} 0 & \text{if } x_i \notin T_l \\ \psi_1 \circ \theta_l^{-1} & \text{if } \theta_l\left(\begin{pmatrix}0\\0\end{pmatrix}\right) = x_i \\ \psi_2 \circ \theta_l^{-1} & \text{if } \theta_l\left(\begin{pmatrix}1\\0\end{pmatrix}\right) = x_i \\ \psi_3 \circ \theta_l^{-1} & \text{if } \theta_l\left(\begin{pmatrix}0\\1\end{pmatrix}\right) = x_i \end{cases}$$

for all $T_l \in \mathcal{T}$ and $i = 1, \dots, N$. By construction, every $u \in V_c$ (and therefore also every $u \in V_s$) is uniquely defined by

$$u = \sum_{i=1}^{N} U_i \phi_i$$

with $U_i = u(x_i)$.

Now, we want to calculate $\int_\Omega u \cdot v \,\mathrm{d}x$ and $\int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}x$ for all $u, v \in V_c$. In order to do that, we set the mass matrix $M_n = \left(\int_\Omega \phi_i \cdot \phi_j \,\mathrm{d}x\right)_{i,j=1,\dots,N}$ and the stiffness matrix $L_n = \left(\int_\Omega \nabla\phi_i \cdot \nabla\phi_j \,\mathrm{d}x\right)_{i,j=1,\dots,N}$. Let

$$U = \begin{pmatrix} U_1 \\ \vdots \\ U_n \end{pmatrix} \text{ and } V = \begin{pmatrix} V_1 \\ \vdots \\ V_n \end{pmatrix}.$$

Then we have

$$\int_\Omega u \cdot v \,\mathrm{d}x = U^T M_n V \text{ and } \int_\Omega \nabla v \cdot \nabla u \,\mathrm{d}x = U^T L_n V.$$

2.2.2 Discretization in time

At first, we partition the time interval $\bar{I} = [0, T]$ as

$$\bar{I} = \{0\} \cup I_1 \cup I_2 \cup \cdots \cup I_M,$$

with subintervals $I_m = (t_{m-1}, t_m]$, where $t_m = m\frac{T}{M}$ for $m = 0, \dots, M$ and $M \in \mathbb{N}$. We want that the discretizations of our functions are continuous in $\bar{I}$ and piecewise polynomial of order 1 in all subintervals I_m, so our discretization space is

$$X_{k,s} := \{v \in C(\bar{I}, H) \mid v|_{I_m} \in \mathcal{P}_1(I_m, V_s), m = 1, 2, \dots, M\}$$

for our state variables and

$$X_{k,c} := \{v \in C(\bar{I}, H) \mid v|_{I_m} \in \mathcal{P}_1(I_m, V_c), m = 1, 2, \dots, M\} \supset X_{k,s}$$

for our control variables, where $\mathcal{P}_1(I_m, V)$ denotes the space of polynomials up to order 1 defined on I_m with values in V.

By using the Lagrange basis of $\mathcal{P}_1(I_m, \mathbb{R})$, we can write every function $v \in X_{k,c}$ as

$$v(t, \cdot) = \left(m - t\frac{M}{T}\right) v_{t_{m-1}}(\cdot) + \left(t\frac{M}{T} - m + 1\right) v_{t_m}(\cdot) \text{ for } t \in I_m,$$

where $v_{t_m}(\cdot) = v(t_m, \cdot)$.

2.2.3 Crank-Nicolson scheme

Now, we solve the weak state equations (2.2) for the state $U \in X_{k,s}$ and $f, q \in X_{k,c}$ numerically. For $m = 0$, we set:

$$U_0 = \begin{pmatrix} U_{0,1} \\ \vdots \\ U_{0,n} \end{pmatrix}$$

For $m = 1, \dots, M$, we get with the Crank-Nicolson scheme that for all $\phi \in V_s$:

$$\begin{aligned}(U_m, \phi) + \frac{T}{2M}(\nabla U_m, \nabla \phi) &= (U_{m-1}, \phi) - \frac{T}{2M}(\nabla U_{m-1}, \nabla \phi) \\ &\quad + \frac{T}{2M}(f_{m-1} + q_{m-1}, \phi) + \frac{T}{2M}(f_m + q_m, \phi).\end{aligned}$$

To solve that, we define the matrix $\tilde{M}_n \in \mathbb{R}^{N \times N}$ as

$$\left(\tilde{M}_n\right)_{i,j} = \begin{cases} 0 & \text{if } x_j \text{ in } \partial\Omega \\ (M_n)_{i,j} & \text{else,} \end{cases}$$

so that $(V_n, W_n) = V_n^T \tilde{M}_n W_n$ for $V_n \in V_c$ and $W_n \in V_s$. Now we solve

$$\begin{aligned}\tilde{M}_n^T U_m + \frac{T}{2M} L_n^T U_m &= \tilde{M}_n^T U_{m-1} - \frac{T}{2M} L_n^T U_{m-1} \\ &\quad + \frac{T}{2M}\tilde{M}_n^T (f_{m-1} + q_{m-1}) + \frac{T}{2M}\tilde{M}_n^T (f_m + q_m)\end{aligned}$$

$$\begin{aligned}\implies U_m = \left(\tilde{M}_n^T + \frac{T}{2M} L_n^T\right)^{-1} &\left(\tilde{M}_n^T U_{m-1} - \frac{T}{2M} L_n^T U_{m-1}\right. \\ &\left. + \frac{T}{2M}\tilde{M}_n^T (f_{m-1} + q_{m-1}) + \frac{T}{2M}\tilde{M}_n^T (f_m + q_m)\right) \quad (2.5)\end{aligned}$$

2.2.4 Calculation of the objective function value

For fixed $\hat{u}, f \in X_{k,c}$, we define $u = u(q)$ for all $q \in X_{k,c}$ so that it satisfies (2.5). We calculate $j(q)$ now in the following way

$$\begin{aligned}j(q) = \quad & \tfrac{1}{2}\textstyle\sum_{m=1}^{M} \int_{t_{m-1}}^{t_m} \quad \left(\left(m - t\frac{M}{T}\right)\left(u_{t_{m-1}} - \hat{u}_{t_{m-1}}\right) + \left(t\frac{M}{T} - m + 1\right)\left(u_{t_m} - \hat{u}_{t_m}\right), \right. \\ & \left. \left(m - t\frac{M}{T}\right)\left(u_{t_{m-1}} - \hat{u}_{t_{m-1}}\right) + \left(t\frac{M}{T} - m + 1\right)\left(u_{t_m} - \hat{u}_{t_m}\right) \right) \mathrm{d}t \\ & + \tfrac{\alpha}{2}\textstyle\sum_{m=1}^{M} \int_{t_{m-1}}^{t_m} \quad \left(\left(m - t\frac{M}{T}\right)\left(q_{t_{m-1}}\right) + \left(t\frac{M}{T} - m + 1\right)\left(q_{t_m}\right), \right. \\ & \left. \left(m - t\frac{M}{T}\right)\left(q_{t_{m-1}}\right) + \left(t\frac{M}{T} - m + 1\right)\left(q_{t_m}\right) \right) \mathrm{d}t.\end{aligned}$$

Integration by substitution yields

$$\begin{aligned} j(q) = \quad & \tfrac{T}{6M}\textstyle\sum_{m=1}^{M} \quad \left(u_{t_{m-1}} - \hat{u}_{t_{m-1}}, u_{t_{m-1}} - \hat{u}_{t_{m-1}}\right) + \left(u_{t_{m-1}} - \hat{u}_{t_{m-1}}, u_{t_m} - \hat{u}_{t_m}\right) \\ & \qquad + \left(u_{t_m} - \hat{u}_{t_m}, u_{t_m} - \hat{u}_{t_m}\right) \\ & + \tfrac{\alpha T}{6M}\textstyle\sum_{m=1}^{M} \quad \displaystyle\int_{t_{m-1}}^{t_m} \left(q_{t_{m-1}}, q_{t_{m-1}}\right) + \left(q_{t_{m-1}}, q_{t_m}\right) + \left(q_{t_m}, q_{t_m}\right) \\ = \quad & \tfrac{T}{6M}\textstyle\sum_{m=1}^{M} \quad \left(u_{t_{m-1}} - \hat{u}_{t_{m-1}}\right) M_n \left(u_{t_{m-1}} - \hat{u}_{t_{m-1}}\right) \\ & \qquad + \left(u_{t_{m-1}} - \hat{u}_{t_{m-1}}\right) M_n \left(u_{t_m} - \hat{u}_{t_m}\right) \\ & \qquad + \left(u_{t_m} - \hat{u}_{t_m}\right) M_n \left(u_{t_m} - \hat{u}_{t_m}\right) \\ & + \tfrac{\alpha T}{6M}\textstyle\sum_{m=1}^{M} \quad q_{t_{m-1}} M_n q_{t_{m-1}} + q_{t_{m-1}} M_n q_{t_m} + q_{t_m} M_n q_{t_m} \end{aligned}$$

2.3 Optimization of the control variable

To optimize the control variable, we write every $q \in X_{k,c}$, using a fixed basis $\Phi = \{\phi_1, \ldots, \phi_{N_b}\}$ with $\phi_1, \ldots, \phi_{N_b} \in V_c$ and scalars $q_1^0, q_1^1, \ldots, q_1^M, \ldots, q_{N_b}^0, q_{N_b}^1 \ldots, q_{N_b}^M \in \mathbb{R}$, as

$$q(t,x) = \sum_{i=1}^{N_b} \alpha_i(t)\phi_i(x) \tag{2.6}$$

with

$$\alpha_i(t) = \begin{cases} q_i^{m-1}\left(m - t\frac{M}{T}\right) + q_i^m\left(t\frac{M}{T} - m + 1\right) & \text{if } t \in I_m \text{ with } m = 1, \ldots, M \\ q_i^0 & \text{if } t = 0 \end{cases}$$

Each control variable, that is written in this form, can be represented as a vector

$$\mathbf{q} = \left[q_1^0, q_1^1, \ldots, q_1^M, \ldots, q_{N_b}^0, q_{N_b}^1 \ldots, q_{N_b}^M\right]^T \in \mathcal{D} := \mathbb{R}^{N_q},$$

with $N_q = (M+1) \cdot N_b$. Therefore, we write

$$j(\mathbf{q}) := j(q)$$

for each q defined like in (2.6). In the next chapters we present algorithms that minimize $j(\mathbf{q})$ with respect to its control vector $\mathbf{q}$.

3 Ensemble-based optimization algorithm

The adaptive ensemble-based algorithm (EnOpt) is usually used to maximize the net present value of an oil recovery method with respect to a control vector. We begin by describing this algorithm for a general function $F : \mathbb{R}^{N_q} \to \mathbb{R}$ to iteratively solve the optimization problem

$$\underset{\mathbf{q}\in\mathcal{D}}{\text{maximize}}\, F(\mathbf{q}).$$

We start with an initialization $\mathbf{q}_0$ and update it iteratively using a preconditioned gradient ascent method given by

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \beta_k \mathbf{d}_k,$$
$$\mathbf{d}_k \approx \frac{\mathbf{C}^k_{\mathbf{q}_k}\mathbf{G}_k}{\|\mathbf{C}^k_{\mathbf{q}_k}\mathbf{G}_k\|_\infty},$$

where $k = 0, 1, 2, ...$ denotes the optimization iteration. β_k with $0 < \beta_k \leq 1$ is computed by using a line search. Furthermore, $\mathbf{C}^k_{\mathbf{q}_k}$ denotes the user-defined covariance matrix of the control variables at the k-th iteration and $\mathbf{G}_k$ is the approximate gradient of F with respect to the control variables.

We define the initial covariance matrix $\mathbf{C}^0_{\mathbf{q}_0}$ so that the covariance between controls of different basis functions ϕ_i, ϕ_j is zero and

$$\mathrm{Cov}(q^i_j, q^{i+h}_j) = \sigma^2_j \rho^h \left(\frac{1}{1-\rho^2}\right), \text{ for all } h \in \{0, ..., M-i\},$$

where $\sigma^2_j > 0$ is the variance for the basis function ϕ_j and $\rho \in (-1, 1)$ the correlation coefficient.

That means that for $\mathbf{C}_j := \left(\mathrm{Cov}(q^i_j, q^k_j)\right)_{i,k}$ with $j = 1, \ldots, N_b$, we set

$$\mathbf{C}^0_{\mathbf{q}_0} = \begin{pmatrix} \mathbf{C}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{C}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{C}_{N_b} \end{pmatrix}.$$

To compute the step direction $\mathbf{d}_k$ at iteration step k, we sample $\mathbf{q}_{k,m} \in \mathcal{D}$ for $m = 1, ..., N$, with $N \in \mathbb{N}$, from a multivariate Gaussian distribution with mean $\mathbf{q}_k$ and covariance $\mathbf{C}^k_{\mathbf{q}_k}$ and define

$$\mathbf{C}^k_{\mathbf{q}_k,F} := \frac{1}{N-1}\sum_{m=1}^{N}(\mathbf{q}_{k,m} - \mathbf{q}_k)(F(\mathbf{q}_{k,m}) - F(\mathbf{q}_k)).$$

Now, we set $\mathbf{d}_k = \frac{\mathbf{C}^k_{\mathbf{q}_k,F}}{\|\mathbf{C}^k_{\mathbf{q}_k,F}\|_\infty}$. This is valid since $\mathbf{C}^k_{\mathbf{q}_k,F}$ is an estimation of $\mathbf{C}^k_{\mathbf{q}_k}\mathbf{G}_k$, which can by shown like in [3]. Here, we begin with the Taylor expansion around $\mathbf{q}_k$ and get

$$\begin{aligned} & F(\mathbf{q}) = F(\mathbf{q}_k) + (\mathbf{q} - \mathbf{q}_k)^T \nabla F(\mathbf{q}_k) + O(\|\mathbf{q} - \mathbf{q}_k\|^2) \\ \Longrightarrow \quad & F(\mathbf{q}) - F(\mathbf{q}_k) = (\mathbf{q} - \mathbf{q}_k)^T \mathbf{G}_k + O(\|\mathbf{q} - \mathbf{q}_k\|^2). \end{aligned}$$

Multiplying both sides by $(\mathbf{q} - \mathbf{q}_k)$ and setting $\mathbf{q} = \mathbf{q}_{k,m}$ yields

$$\begin{aligned} & (\mathbf{q}_{k,m} - \mathbf{q}_k)(F(\mathbf{q}_{k,m}) - F(\mathbf{q}_k)) \\ = \; & (\mathbf{q}_{k,m} - \mathbf{q}_k)(\mathbf{q}_{k,m} - \mathbf{q}_k)^T \mathbf{G}_k + O(\|\mathbf{q}_{k,m} - \mathbf{q}_k\|^3), \end{aligned}$$

where $O(\|\mathbf{q}_{k,m} - \mathbf{q}_k\|^3)$ are the remaining terms containing order ≥ 3 of $(\mathbf{q}_{k,m} - \mathbf{q}_k)$. Neglecting $O(\|\mathbf{q}_{k,m} - \mathbf{q}_k\|^3)$ gives by summation over all samples and multiplication of both sides with $\frac{1}{N-1}$:

$$\begin{aligned} & \frac{1}{N-1} \sum_{m=1}^{N} (\mathbf{q}_{k,m} - \mathbf{q}_k)(F(\mathbf{q}_{k,m}) - F(\mathbf{q}_k)) \\ \approx \; & \left(\frac{1}{N-1} \sum_{m=1}^{N} (\mathbf{q}_{k,m} - \mathbf{q}_k)(\mathbf{q}_{k,m} - \mathbf{q}_k)^T \right) \mathbf{G}_k \\ \Longrightarrow \; & \mathbf{C}^k_{\mathbf{q}_k,F} \approx \mathbf{C}^k_{\mathbf{q}_k} \mathbf{G}_k, \end{aligned}$$

since $\frac{1}{N-1} \sum_{m=1}^{N} (\mathbf{q}_{k,m} - \mathbf{q}_k)(\mathbf{q}_{k,m} - \mathbf{q}_k)^T$ is itself an approximation of $\mathbf{C}^k_{\mathbf{q}_k}$.

Using the samples $\{\mathbf{q}_{k,m}\}_{m=1}^N$ and the covariance matrix $\mathbf{C}^k_{\mathbf{q}_k}$ from the last iteration, we can update $\mathbf{C}^k_{\mathbf{q}_k}$ by setting

$$\mathbf{C}^{k+1}_{\mathbf{q}_{k+1}} = \mathbf{C}^k_{\mathbf{q}_k} + \tilde{\beta}_k \tilde{\mathbf{d}}_k, \text{ with}$$

$$\tilde{\mathbf{d}}_k = N^{-1} \sum_{m=1}^{N} (F(\mathbf{q}_{k,m}) - F(\mathbf{q}_k))((\mathbf{q}_{k,m} - \mathbf{q}_k)(\mathbf{q}_{k,m} - \mathbf{q}_k)^T - \mathbf{C}^k_{\mathbf{q}_k}),$$

where $\tilde{\beta}_k$ is a step size.

Now, we iterate until $F(\mathbf{q}_k) \leq F(\mathbf{q}_{k-1}) + \varepsilon$, with $\varepsilon > 0$. This gives us the following algorithms:

Algorithm 1 EnOpt algorithm

1: **procedure** EnOpt($F, \mathbf{q}_0, N, \varepsilon, k^*, \beta, \tilde{\beta}, r, \nu^*, \sigma^2, \rho, \text{pr} = \text{id}$)
2: $F_0 \leftarrow F(\mathbf{q}_0)$
3: $\mathbf{q}_1, T_1, \mathbf{C}^0_{\mathbf{q}_0}, F_1 \leftarrow \text{OptStep}(F, \mathbf{q}_0, N, 0, \{\}, 0, F_0, \beta, \tilde{\beta}, r, \varepsilon, \nu^*, \sigma^2, \rho, \text{pr})$
4: $k \leftarrow 1$
5: **while** $F_k > F_{k-1} + \varepsilon$ and $k < k^*$ **do**
6: $\mathbf{q}_{k+1}, T_{k+1}, \mathbf{C}^k_{\mathbf{q}_k}, F_{k+1} \leftarrow \text{OptStep}(F, \mathbf{q}_k, N, k, T_k, \mathbf{C}^{k-1}_{\mathbf{q}_{k-1}}, F_k, \beta, \tilde{\beta}, r, \varepsilon, \nu^*, \sigma^2, \rho, \text{pr})$
7: $k \leftarrow k + 1$
8: **return** $\mathbf{q}^* \leftarrow \mathbf{q}_k$

The EnOpt algorithm takes the objective function $F : \mathbb{R}^{N_q} \to \mathbb{R}$, our initial iterate $\mathbf{q}_0 \in \mathbb{R}^{N_q}$, the sample size $N \in \mathbb{N}$, the tolerance $\varepsilon > 0$, the maximum number of iterations $k^* \in \mathbb{N}$, the initial step size $\beta > 0$ for the compuation of the next iterate, the step size $\tilde{\beta}$ for the iteration of the covariance matrix, the step size contraction $r \in (0, 1)$, the maximum number of step size trials $\nu^* \in \mathbb{N}$, the variance $\sigma^2 \in \mathbb{R}^{N_b}$ with positive elements, the correlation coefficient $\rho \in (-1, 1)$ and a projection pr, where the default is the identity function $\text{id} : x \to x$.

Algorithm 2 OptStep algorithm

procedure OptStep(F,$\mathbf{q}_k, N, k, T_k, \mathbf{C}^{k-1}_{\mathbf{q}_{k-1}}, F_k, \beta, \tilde{\beta}, r, \varepsilon, \nu^*, \sigma^2, \rho,$ pr)
  if $k = 0$ **then**
    Compute the initial covariance matrix $\mathbf{C}^0_{\mathbf{q}_0}$ like defined with $\mathbf{q}_0, \sigma^2, \rho$
  else
    Compute the covariance matrix $\mathbf{C}^k_{\mathbf{q}_k}$ like defined with $\mathbf{q}_k, F_k, T_k, \mathbf{C}^{k-1}_{\mathbf{q}_{k-1}}, \tilde{\beta}, N$
  Sample N control vectors $\{\mathbf{q}_{k,j}\}_{j=1}^N$ from a distribution $\mathcal{N}(\mathbf{q}_k, \mathbf{C}^k_{\mathbf{q}_k})$
  $T_{k+1} \leftarrow \{\mathbf{q}_{k,j}, F(\mathbf{q}_{k,j})\}_{j=1}^N$
  Compute the vector $\mathbf{C}^k_{\mathbf{q}_k,F}$ with $\mathbf{q}_k, F_k$ and the stored values of T_{k+1}
  Compute the search direction $\mathbf{d}_k = \mathbf{C}^k_{\mathbf{q}_k,F} / \|\mathbf{C}^k_{\mathbf{q}_k,F}\|_\infty$
  $\mathbf{q}_{k+1} \leftarrow$ LineSearch$(F, \mathbf{q}_k, \mathbf{d}_k, \beta, r, \varepsilon, \nu^*,$ pr)
  $F_{k+1} \leftarrow F(\mathbf{q}_{k+1})$
  return $\mathbf{q}_{k+1}, T_{k+1}, \mathbf{C}^k_{\mathbf{q}_k}, F_{k+1}$

Algorithm 3 Line search

procedure LineSearch$(F, \mathbf{q}_k, \mathbf{d}_k, \beta, r, \varepsilon, \nu^*,$ pr)
  $\beta_k \leftarrow \beta$
  $\mathbf{q}_{k+1} \leftarrow \text{pr}(\mathbf{q}_k + \beta_k \mathbf{d}_k)$
  $\nu \leftarrow 0$
  while $F(\mathbf{q}_{k+1}) - F(\mathbf{q}_k) \leq \varepsilon$ and $\nu < \nu^*$ **do**
    $\beta_k \leftarrow r\beta_k$
    $\mathbf{q}_{k+1} \leftarrow \text{pr}(\mathbf{q}_k + \beta_k \mathbf{d}_k)$
    $\nu \leftarrow \nu + 1$
  return $\mathbf{q}_{k+1}$

Now we use this algorithm to optimize our objective function j. Since this is a maximization procedure and j should be minimized, we apply $-j$ to the EnOpt algorithm which gives us:

Algorithm 4 FOM-EnOpt algorithm

procedure FOM-EnOpt$(\mathbf{q}_0, N, \varepsilon, k^*, \beta, \tilde{\beta}, r, \nu^*, \sigma^2, \rho)$
  return EnOpt$(-j, \mathbf{q}_0, N, \varepsilon, k^*, \beta, \tilde{\beta}, r, \nu^*, \sigma^2, \rho)$

4 Adaptive-ML-EnOpt algorithm

4.1 Deep neural networks

4.2 Modifying the EnOpt algorithm by using a neural network-based surrogate

5 Numerical experiments

Bibliography

[1] T. Keil, H. Kleikamp, R. J. Lorentzen, M. B. Oguntola, and M. Ohlberger, "Adaptive machine learning-based surrogate modeling to accelerate PDE-constrained optimization in enhanced oil recovery," *Advances in Computational Mathematics*, vol. 48, no. 6, p. 73, Nov. 2022.

[2] D. Meidner and B. Vexler, "A priori error estimates for space-time finite element discretization of parabolic optimal control problems part i: Problems without control constraints," *SIAM Journal on Control and Optimization*, vol. 47, no. 3, pp. 1150–1177, 2008. DOI: 10.1137/070694016. eprint: https://doi.org/10.1137/070694016. [Online]. Available: https://doi.org/10.1137/070694016.

[3] M. B. Oguntola and R. J. Lorentzen, "Ensemble-based constrained optimization using an exterior penalty method," *Journal of Petroleum Science and Engineering*, vol. 207, p. 109 165, 2021, ISSN: 0920-4105. DOI: https://doi.org/10.1016/j.petrol.2021.109165. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920410521008184.