AME60714: Advanced Numerical Methods
Homework 3: Due Monday, October 26, 2020

Instructions: Complete four problems of your choice.

Problem 1: (20 points) Consider the following fully discrete, steady nonlinear PDE-constrained optimization problem

$$\begin{aligned} \underset{\boldsymbol{u},\boldsymbol{\mu}}{\text{minimize}} \quad & f(\boldsymbol{u};\boldsymbol{\mu}) \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{u};\boldsymbol{\mu}) = \boldsymbol{0} \\ & \boldsymbol{c}(\boldsymbol{u};\boldsymbol{\mu}) \geqslant \boldsymbol{0} \end{aligned} \tag{1}$$

where $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ is the state variable, $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ is the parameter vector, $\boldsymbol{r} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}^{N_{\boldsymbol{u}}}$ is the discretized PDE, and $\boldsymbol{c} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}^{N_{\boldsymbol{c}}}$ are side constraints.

a) Re-write (1) as an equivalent optimization problem over only $\boldsymbol{\mu}$ by eliminating the PDE constraint.

b) Derive expressions for the gradients (w.r.t. $\boldsymbol{\mu}$) of functionals in the reduced optimization problem using the sensitivity and adjoint approaches. Discuss which approach is more computationally efficient based on $N_{\boldsymbol{c}}$ and $N_{\boldsymbol{\mu}}$.

c) How do these expressions simplify (and the corresponding observations change) if

$$f(\boldsymbol{u};\boldsymbol{\mu}) = \frac{1}{2}(\boldsymbol{u}-\bar{\boldsymbol{u}})^T(\boldsymbol{u}-\bar{\boldsymbol{u}}) + \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\mu}, \quad \boldsymbol{r}(\boldsymbol{u};\boldsymbol{\mu}) = \boldsymbol{A}\boldsymbol{u} + \boldsymbol{\mu}, \quad \boldsymbol{c}(\boldsymbol{u};\boldsymbol{\mu}) = \boldsymbol{C}\boldsymbol{u} + \boldsymbol{D}\boldsymbol{\mu},$$

where $\boldsymbol{A}$, $\boldsymbol{C}$, $\boldsymbol{D}$ are fixed matrices of the appropriate size, $\bar{\boldsymbol{u}}$ is a reference solution, and $\boldsymbol{\mu}$ is the same size as $\boldsymbol{u}$. This situation corresponds to a linear-quadratic control: a linear PDE with fixed coefficients and parametrized source term, linear inequality constraints, and a quadratic objective in $\boldsymbol{u}$ and $\boldsymbol{\mu}$. Explain why these problems are quite simple to solve numerically if $\boldsymbol{A}$, $\boldsymbol{C}$, $\boldsymbol{D}$, and $\bar{\boldsymbol{u}}$ are available.

d) Suppose we use an optimization solver that *only* requires evaluations of the Lagrangian and its gradient (w.r.t. $\boldsymbol{\mu}$). Propose an approach to compute $\frac{d}{d\boldsymbol{\mu}}\mathcal{L}$ that only requires the solution of *one* linear system of equation, regardless of $N_{\boldsymbol{c}}$ and $N_{\boldsymbol{\mu}}$, given $\boldsymbol{u}(\boldsymbol{\mu})$ (the PDE solution at $\boldsymbol{\mu}$) and an estimate of the Lagrange multipliers $\boldsymbol{\lambda}$.

Problem 2: (40 points) In this problem you will determine the optimal shape of the incoming branch of a aorto-coronaric bypass (Figure 1). We will model the (steady) blood flow using the incompressible Navier-Stokes equations

$$-\nu\Delta v + \nabla v \cdot v + \nabla p = 0, \quad \nabla \cdot v = 0 \quad \text{in} \quad \Omega \subset \mathbb{R}^2,$$

where $v = (v_1, v_2)$ is the velocity of the fluid, p is the pressure, and $\nu = 2 \times 10^{-2}$ is the kinematic viscosity. The boundary ($\partial\Omega$) is split into no-slip walls (Γ_w), an inlet (Γ_in), and outlet (Γ_out): $\partial\Omega = \overline{\Gamma_\text{w} \cup \Gamma_\text{in} \cup \Gamma_\text{out}}$ with boundary conditions

$$v = 0 \quad \text{on} \quad \Gamma_\text{w}, \qquad v = g \quad \text{on} \quad \Gamma_\text{in}, \qquad \sigma(v,p)\cdot n = 0 \quad \text{on} \quad \Gamma_\text{out},$$

where $\sigma(v,p) = \nu\nabla v - pI$ is the stress tensor, $n : \partial\Omega \rightarrow \mathbb{R}^2$ is the outward unit normal to $\partial\Omega$, and $g : \partial\Omega \rightarrow \mathbb{R}^2$ is a prescribed parabolic velocity profile $g(x) = (v_\text{in}(x_2 - y_\text{l})(y_\text{u} - x_2)/D^2, 0)$, where $D = y_\text{u} - y_\text{l}$, $y_\text{l} = 0$, $y_\text{u} = 0.3$, and $v_\text{in} = 10$. The objective is to minimize the vorticity of the flow in the down-field zone Ω_wd

$$J(v,\boldsymbol{\mu}) = J_\omega(v,\boldsymbol{\mu}) + \frac{\alpha}{2}\boldsymbol{\mu}^T\boldsymbol{\mu}, \qquad J_\omega(v,\boldsymbol{\mu}) = \int_{\Omega_\text{wd}(\boldsymbol{\mu})} |\omega|^2 \, d\Omega,$$

where $\omega = \nabla \times v$ is the vorticity and $\boldsymbol{\mu}$ are parameters controlling the shape of the domain. This problem is adapted from *Rozza, Gianluigi. "On Optimization, Control and Shape Design of an Arterial Bypass." International Journal for Numerical Methods in Fluids, vol. 47, no. 10–11, 2005, pp. 1411–19. doi:10.1002/fld.888.*

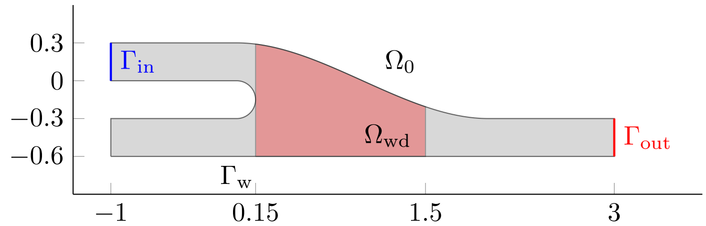

Figure 1: Schematic of nominal configuration of aorto-coronaric bypass.

a) Define the parametrized domain as $\Omega(\boldsymbol{\mu}) = \mathcal{G}(\Omega_0, \boldsymbol{\mu})$, where Ω_0 is the nominal domain (Figure 1) and $\mathcal{G}$ is a bijection. In a computational setting, $\mathcal{G}$ is used to map the nodal coordinates of a mesh of Ω_0 to define a mesh of Ω, i.e., let $X \in \Omega$ be the coordinate of a mesh node, then $x = \mathcal{G}(X, \boldsymbol{\mu})$ is the coordinate of the corresponding node in $\Omega(\boldsymbol{\mu})$. In this work, we will use radial basis functions (RBFs) to parametrize the domain

$$\mathcal{G}_i(X, w) = X_i + \sum_{j=1}^{N_c} w_{ij} \phi(\|X - \hat{X}_j\|), \qquad \phi(r) = \begin{cases} \exp\left(-\frac{1}{1-(r/R)^2}\right) & \text{if } r < R \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

for $i = 1, 2$, where $\phi : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ is the RBF kernel with support R (smooth bump in this case; there are many others), $w \in \mathbb{R}^{2 \times N_c}$ are the RBF weights, and $\{\hat{X}_j\}_{j=1}^{N_c}$ are the RBF centers. We will fix all of the x-weights to zero, e.g., $w_{1j} = 0$ for $j = 1, \ldots, N_c$, and take the y-weights as our parameters: $\boldsymbol{\mu} = (w_{21}, \ldots, w_{2N_c})$. Implement this domain parametrization and its sensitivity to the weights $\frac{\partial \mathcal{G}}{\partial w}$ using the following modular design and the starter code provided.

```
function phi0 = eval_rbf_bumpfcn(eta, R)
%EVAL_RBF_BUMPFCN Evaluate RBF bump function kernel with radius R and
%parameter eta.
```

```
function phi = compute_dist_eval_rbf_kern(X, Xcntrl, R)
%COMPUTE_DIST_EVAL_RBF_KERN Compute distance between evaluation points (X,
%Array (ndim, neval)) (usually nodes of mesh) and control nodes (Xcntrl,
%Array (ndim, ncntrl)), and evaluate smooth bump RBF kernal with support R.
```

```
function [X, dXdW] = compute_meshmot_rbf(W, X0, phi)
%COMPUTE_MESHMOT_RBF Compute mesh motion (X, Array (ndim*nv,)) and
%sensitivity (dXdW, Array (ndim*nv, ndim*ncntrl)) from RBF weights (W,
%Array (ndim*ncntrl)) and kernal (phi, Array (nv, ncntrl)), and original
%(undeformed) mesh nodes (X0, Array (ndim*nv,)).
```

Test your implementation (and gain intuition for RBFs) using a single RBF by plotting $\Omega(\boldsymbol{\mu})$ for a number of configurations.

b) The finite element discretization of the PDE is abstracted as

$$\boldsymbol{r}_u(\boldsymbol{u}_u; \boldsymbol{u}_c, \boldsymbol{x}) = \boldsymbol{0},$$

where $\boldsymbol{x}$ are the coordinates of the (deformed) mesh nodes, $\boldsymbol{u}_u$ and $\boldsymbol{u}_c$ are the unconstrained and constrained finite element degrees of freedom, respectively. After discretization, the discretion of the objective function J takes the form: $f(\boldsymbol{u}_u; \boldsymbol{u}_c, \boldsymbol{x})$. This is setup for you in the `FEdu` (see starter code). Make sure you can run it at the nominal configuration (Figure 1) (where $\boldsymbol{x} = \boldsymbol{X}$, the nodes of the mesh of Ω_0), evaluate f, and visualize the velocity field.

c) Through an abuse of notation, we can write the nodes of the deformed mesh $\boldsymbol{x}$ as a function of the parameters $\boldsymbol{\mu}$: $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\mu})$ using the mapping in (2). Verify that the sensitivity of the unconstrained degrees of freedom w.r.t. the parameters and total derivative of f reads

$$\frac{\partial \boldsymbol{u}_u}{\partial \boldsymbol{\mu}} = -\left[\frac{\partial \boldsymbol{r}_u}{\partial \boldsymbol{u}_u}\right]^{-1}\left(\frac{\partial \boldsymbol{r}_u}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}}\right), \qquad \frac{d}{d\boldsymbol{\mu}} f(\boldsymbol{u}_u(\boldsymbol{\mu}); \boldsymbol{u}_c, \boldsymbol{x}(\boldsymbol{\mu})) = \frac{\partial f}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}} + \frac{\partial f}{\partial \boldsymbol{u}_u}\frac{\partial \boldsymbol{u}_u}{\partial \boldsymbol{\mu}},$$

provided $\frac{\partial \boldsymbol{u}_c}{\partial \boldsymbol{\mu}} = \boldsymbol{0}$ (which is true for this problem since all constrained degrees of freedom correspond to stick walls where $v = 0$). These expressions are implemented in the starter code. Use a single RBF centered at $(1, 0)$ with radius $R = 0.6$ and plot the sensitivity of the velocity magnitude with respect to the RBF weights at $w_{21} = 0$.

d) Solve the fully discrete PDE-constrained optimization problem using $\alpha = 500$ (gradients computed using sensitivity method described above) and the single RBF configuration from the previous part. Use MATLAB's `fminunc` to solve unconstrained optimization problem. Be sure to use the user-supplied gradient option (`options = optimoptions('fminunc', 'SpecifyObjectiveGradient', true);`). Initialize all RBF weights to zero. Plot the velocity magnitude in the undeformed and optimized domains. Experiment with the sensitivity of the shape optimization with respect to the RBF configuration and parameter α, and comment on your observations.

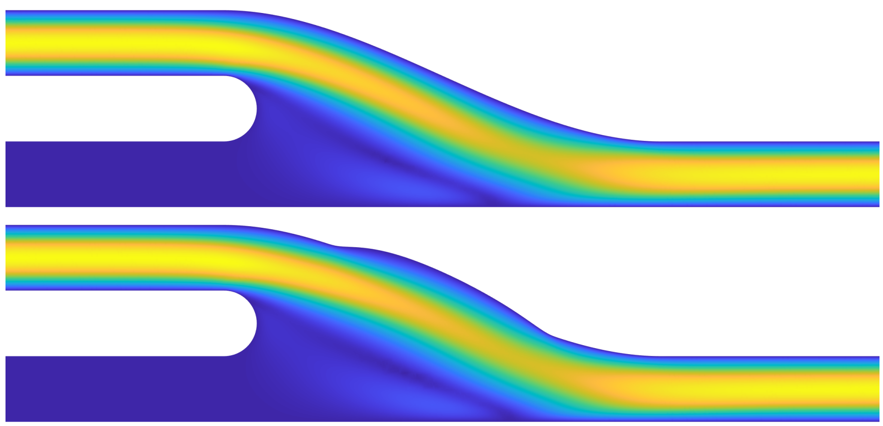

Figure 2: Example of flow through nominal (*top*) and optimized (*bottom*) bypass.

Problem 3: (50 points) In this problem you will determine the source of a contaminant that has dispersed through a known two-dimensional velocity field based on an observation of the contaminant taken some time T after the initial spill (Figure 3). The evolution of the concentration of the contaminant $q(x, t)$ is modeled by the convection-diffusion equation

$$q_{,t} + \nabla \cdot (\beta q) - \nu \Delta q = 0 \quad \text{in} \quad \Omega \times (0, T],$$

where $\Omega = [-1, 1] \times [-1, 1]$ is the spatial domain, $\beta(x) = (\sin(\pi x_1)\cos(\pi x_2), -\cos(\pi x_1)\sin(\pi x_2))$ is the velocity field (Taylor-Green vortex), $\nu = 10^{-3}$ is the diffusion coefficient, and $T = 4$ is the observation time. The boundary ($\partial\Omega$) is split into Neumann (Γ_N) and Dirichlet (Γ_D) (Figure 3): $\partial\Omega = \overline{\Gamma_N \cup \Gamma_D}$ with boundary conditions

$$\nabla q \cdot n = 0 \quad \text{on} \quad \Gamma_N, \qquad q = 0 \quad \text{on} \quad \Gamma_D,$$

where $n : \partial\Omega \to \mathbb{R}^2$ is the outward unit normal to $\partial\Omega$. Let $\hat{q}(x)$ denote the observation at time T and the initial condition be $q(x, 0) = \mathring{q}(x)$, where $\mathring{q}(x)$ is an unknown function we seek that leads to the state q that best explains the observation $\hat{q}$ at time T. Thus, we will take $\mathring{q}$ as optimization variables and minimize the discrepancy between $q(x, T)$ and $\hat{q}$

$$J(q) = \frac{1}{2}\int_\Omega |q(x, T) - \hat{q}(x)|^2 \, d\Omega + \frac{\alpha}{2}\int_\Omega |\mathring{q}(x) - \bar{q}|^2 \, d\Omega$$

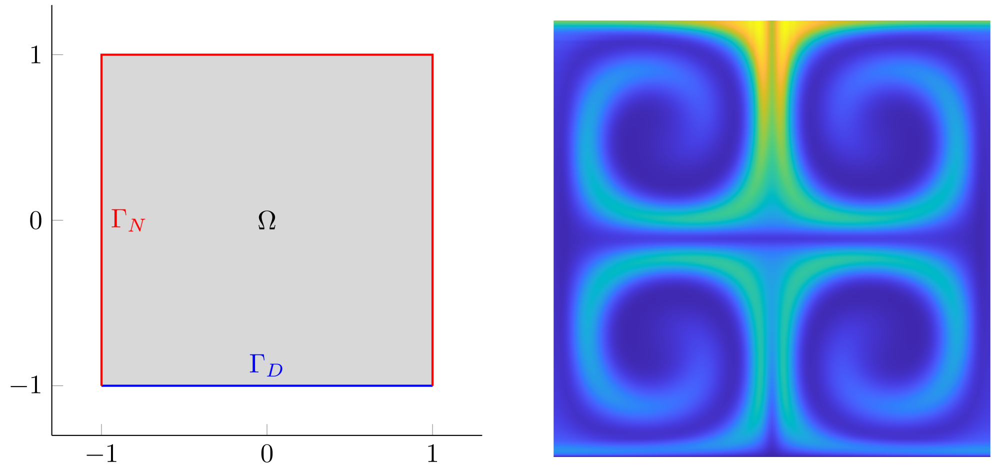

Figure 3: Convection-diffusion source inversion: geometry (*left*) and observation $\hat{q}$ at time $t = T$ (*right*).

subject to the constraint that q satisfies the above PDE with initial condition $\mathring{q}$, where $\bar{q}$ is a reference state that incorporates any prior knowledge of the spill location/magnitude and $\alpha > 0$ is a regularization parameter. This problem is adapted from *Kärcher, Mark, et al. "Reduced Basis Approximation and a Posteriori Error Bounds for 4D-Var Data Assimilation." Optimization and Engineering, vol. 19, no. 3, 2018, pp. 663–95. doi:10.1007/s11081-018-9389-2.* This is a simplified data assimilation setting; usually, it is not possible to observe the entire state; rather, some function of the state is measured at a finite number of sensors and time instances.

Semi-discretization of the above PDE leads to a system of ODEs

$$\boldsymbol{M}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = 0, \quad \boldsymbol{q}(0) = \mathring{\boldsymbol{q}},$$

where $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{K}$ is the discretization of the convection and diffusion terms, $\boldsymbol{q}(t)$ is the state of the discretized PDE, and $\mathring{\boldsymbol{q}}$ is the discretization of the initial condition $\mathring{q}$. Semi-discretization of the optimization functional leads to the finite-dimensional approximation of J (written in terms of nodal values of solution, integral replaced with quadrature)

$$J_h(\boldsymbol{q}) = \frac{1}{2}(\boldsymbol{q}(T) - \hat{\boldsymbol{q}})^T \boldsymbol{M}(\boldsymbol{q}(T) - \hat{\boldsymbol{q}}) + \frac{\alpha}{2}(\mathring{\boldsymbol{q}} - \bar{\boldsymbol{q}})^T \boldsymbol{M}(\mathring{\boldsymbol{q}} - \bar{\boldsymbol{q}}),$$

where $\hat{\boldsymbol{q}}$ is the discretization of the observation state $\hat{q}$ and $\bar{\boldsymbol{q}}$ is the discretization of the reference state $\bar{q}$. This semi-discretization is setup for you in `FEdu` (see starter code).

Partition the temporal domain into N_t timesteps and apply backward Euler to yield the fully discrete formulation

$$\boldsymbol{R}_n(\boldsymbol{q}_n; \boldsymbol{q}_{n-1}) := \boldsymbol{M}\frac{\boldsymbol{q}_n - \boldsymbol{q}_{n-1}}{\Delta t} + \boldsymbol{K}\boldsymbol{q}_n = \boldsymbol{0}$$

for $n = 1, \ldots, N_t$, where Δt is the timestep size, $\boldsymbol{q}_n \approx \boldsymbol{q}(n\Delta t)$ is the fully discrete state, and $\boldsymbol{q}_0 = \mathring{\boldsymbol{q}}$. The fully discrete objective function takes the form

$$f(\boldsymbol{q}_{N_t}) = \frac{1}{2}(\boldsymbol{q}_{N_t} - \hat{\boldsymbol{q}})^T \boldsymbol{M}(\boldsymbol{q}_{N_t} - \hat{\boldsymbol{q}}) + \frac{\alpha}{2}(\mathring{\boldsymbol{q}} - \bar{\boldsymbol{q}})^T \boldsymbol{M}(\mathring{\boldsymbol{q}} - \bar{\boldsymbol{q}}).$$

From the fully discrete objective function and PDE, the fully discrete PDE-constrained optimization problem is

$$\begin{aligned} \underset{\mathring{\boldsymbol{q}}, \boldsymbol{q}_0, \ldots, \boldsymbol{q}_{N_t}}{\text{minimize}} \quad & f(\boldsymbol{q}_{N_t}) \\ \text{subject to} \quad & \boldsymbol{q}_0 = \mathring{\boldsymbol{q}} \\ & \boldsymbol{R}_n(\boldsymbol{q}_n; \boldsymbol{q}_{n-1}) = 0, \quad n = 1, \ldots, N_t. \end{aligned}$$

a) Implement a function that advances the solution $\boldsymbol{q}_{n-1}$ to $\boldsymbol{q}_n$ using backward Euler

```
function u_np1 = advance_ode_lin_bdf1(u_n, dt, M, K)
%ADVANCE_ODE_LIN_BDF1 Advance a linear system of ODEs defined by
%
%                   M * \dot{u} + K * u = 0
%
% from time n to n+1 (step size = dt) using Backward Euler (BDF1).
%
%Input arguments
%----------------
%  u_n : Array (m,) : Solution vector at step n
%
%  dt : number : Time step
%
%  M : Array (m, m) : Mass matrix of ODE system
%
%  K : Array (m, m) : Stiffness matrix of ODE system
%
%Output arguments
%----------------
%  u_np1 : Array (m,) : Solution vector at step n+1
```

and use to approximate the convection-diffusion equation with initial condition

$$\mathring{q}(x) = \exp\left(-\frac{(x_1-0.2)^2+(x_2+0.5)^2}{0.2}\right).$$

Plot the solution at the final time $T = 4$.

b) Derive the fully discrete sensitivity equations (taking $\mathring{\boldsymbol{q}}$ as the parameter vector) and an expression for the total derivative of $F(\mathring{\boldsymbol{q}}) = f(\boldsymbol{q}_{N_t}(\mathring{\boldsymbol{q}}))$. Explain why this is not practical approach to solve the PDE-constrained optimization as the number of degrees of freedom in the spatial discretization increases.

c) Derive the optimality system for the fully discrete PDE-constrained optimization problem.

d) Identify the adjoint equations from the optimality system and an expression for the total derivative of $F(\mathring{\boldsymbol{q}})$ *independent of the sensitivities*. Describe how your backward Euler code can be used without modification to advance the adjoint solution from $\boldsymbol{\lambda}_{n+1}$ to $\boldsymbol{\lambda}_n$.

e) Implement a function that solves the primal and adjoint equations and returns $F(\mathring{\boldsymbol{q}})$ and its gradient. Ensure your gradients are correct by running a finite difference test (use only a few time steps and a coarse spatial discretization for speed).

```
function [F, dF] = solve_prim_dual_eval_funcl_contaminv(Q0, M, K, dt, nstep, Uhat, ...
    alpha, U0ref)
%SOLVE_PRIM_DUAL_EVAL_FUNCL_CONTAMINV Solve primal and adjoint equations
%(fully discrete) for contaminant problem, evaluate objective functional
%and its total derivative.
```

f) Generate a synthetic observation by defining a "true" initial condition

$$\mathring{q}_\star(x) = 5\,\exp\left(-\frac{x_1^2+x_2^2}{0.1}\right)$$

and using your BDF code to compute the corresponding PDE solution at time T, i.e., $\hat{q}(x) = q_\star(x, T)$, where $q_\star$ is the solution of the convection-diffusion equation with initial condition $\mathring{q}_\star$.

g) Solve the fully discrete PDE-constrained optimization problem using using MATLAB's `fminunc` to solve unconstrained optimization problem. Be sure to use the user-supplied gradient option (`options = ... optimoptions('fminunc', 'SpecifyObjectiveGradient', true);`). Use $\alpha = 10^{-3}$ and the reference solution

$$\bar{q}(x) = \exp\left(-\frac{(x_1 - 0.2)^2 + (x_2 + 0.5)^2}{0.2}\right).$$

Also use $\bar{q}$ as your initial guess for the optimization problem. Did you recover the expected origin of the spill and its magnitude? Plot the initial condition you recovered, the true initial condition $\mathring{q}_\star$, the observation $\hat{\boldsymbol{q}}$, and $\boldsymbol{q}(T)$ based on the optimized initial condition. Experiment with the sensitivity of the inversion with respect to the parameter α and the final time T, and comment on your observations.

h) If we knew the true initial concentration distribution is well-approximated as a Gaussian function, how could we use this to improve our optimization setup?

Problem 4: (20 points) Read the following paper and discuss what properties the numerical flux and boundary functions must satisfy for DG to be adjoint consistent for systems of nonlinear conservation laws (such as the Euler equations). *Hartmann, Ralf. "Adjoint Consistency Analysis of Discontinuous Galerkin Discretizations." SIAM Journal on Numerical Analysis, vol. 45, no. 6, Jan. 2007, pp. 2671–96. doi:10.1137/060665117.*

Problem 5: (30 points) Consider a parametrized one-dimensional periodic system of nonlinear conservation laws

$$q_{,t} + f(q)_{,x} = 0, \quad q(x,0;\mu) = g(x;\mu), \quad q(a,t;\mu) = q(b,t;\mu) \tag{3}$$

for all $x \in (a,b)$ and $t \in (0,T]$, where $q(x,t;\mu) \in \mathbb{R}^m$ is the conservative state, $g(x;\mu)$ is the parametrized initial condition, and $f(q) \in \mathbb{R}^m$ is the (nonlinear) flux function.

a) Derive the continuous sensitivity equations.

b) Discuss the challenges associated with using a FV/DG spatial discretization and RK4 temporal discretization to discretize and solve the continuous sensitivity equations. Which of these challenges would be alleviated using a semi-discrete or fully discrete approach to sensitivities? What happens for the special case of linear advection ($f(q) = cq$)?

c) Use your code from Homework 2 to solve the primal equations to compute $q(x,t;\mu)$ and solve the sensitivity equations linearized about $q(x,t;\mu)$ for linear advection ($f(q) = cq$) with periodic boundary conditions and parametrized initial condition $g(x;\mu) = e^{-x^2/\mu}$. Take $c = 1$, $b = -a = 1$, $T = 0.6$, and $\mu = 0.1$. Plot the primal and sensitivity solutions at the initial $t = 0$ and final time $t = T$. Justify the sensitivity solution is reasonable based on your knowledge of linear advection.

Problem 6: (20 points) Consider the following PDE-constrained optimization problem

$$\begin{aligned} \underset{q,g}{\text{minimize}} \quad & J(q) \\ \text{subject to} \quad & q_{,t} + f(q)_{,x} = 0 \\ & q(x,0) = g(x) \\ & q(a,t) = q(b,t) \end{aligned}$$

where the constraints must hold for all $x \in \Omega := (a,b)$ and $t \in (0,T]$, $q(x,t) \in \mathbb{R}^m$ is the conservative state, $g(x)$ is the initial condition, $f(q) \in \mathbb{R}^m$ is the (nonlinear) flux function, the objection functional is

$$J(q) = \int_0^T \int_\Omega h(q)\, dx,$$

and h is some operator, e.g., $h(q) = |q - \hat{q}|^2$ where $\hat{q}$ is some desired state.

a) Derive the optimality conditions.

b) Identify the adjoint equations from the optimality system and write as a conservation law (clearly identify: flux function, source term, boundary conditions, and initial condition).

c) Discuss the challenges associated with using a FV/DG spatial discretization and RK4 temporal discretization to discretize and solve the continuous adjoint equations. Which of these challenges would be alleviated using a semi-discrete or fully discrete approach to adjoints?