MANE-6960
Vignesh Ramakrishnan
Due: Wednesday, 11 December 2024

Assignment 4

1 Abstract

This work presents a computational study of the transient solution of the one-dimensional Euler equations, utilizing the implicit Crank-Nicholson method for time integration of the primal problem. The focus lies on the development and validation of an adjoint-based optimization framework to control the time-integrated pressure towards a specified target value. The optimization problem is formulated using a Lagrangian approach, which incorporates the objective functional and the governing equations as constraints.

The adjoint variables were computed by solving the adjoint equations through time integration in reverse (backward) mode. This facilitated the computation of gradients of the discrete objective functional with respect to state variables, denoted as $\frac{\partial J_h}{\partial q_h^{(n)}}$. The implementation of these gradients was rigorously verified against the complex step method, ensuring precision and correctness. Furthermore, the calculated adjoints were employed to evaluate the total derivatives of the objective functional with respect to control parameters, $\frac{DJ_h}{Ds^{(n)}}$. These adjoint-based gradients were subsequently verified against forward difference approximations, demonstrating their accuracy and robustness.

The study highlights the interplay between numerical optimization, adjoint methods, and robust validation techniques within the context of computational fluid dynamics (CFD). The methodology and results underscore the effectiveness of adjoint-based sensitivity analysis in solving complex optimization problems governed by partial differential equations.

2 Introduction and problem setup

In this assignment we consider the 1D Euler equations given by the PDE,

$$R(q) \equiv \frac{\partial q}{\partial t} + \frac{\partial}{\partial x}\left[F(q)\right] - G(x,t) = 0, \ \forall x \in [0,1], \ t \in [0,2], \tag{1}$$

where the flux and the source terms are,

$$F(q) = \left[\rho u, \left(\rho u^2 + p\right), u\left(e+p\right)\right]^T, \text{ and } G(x,t) = \left[0, s(t)\exp\left(-\frac{(x-x_s)^2}{\sigma_s^2}\right), 0\right]^T. \tag{2}$$

We apply characteristic boundary conditions at the inflow, $x = 0$, such that ~~the~~ $q(0,t) = (1,1,1)^T$. We will assume that the flow is supersonic for all the sources; thus, no boundary condition is necessary at the outflow, $x = 1$.

The magnitude of the source is controlled by $s(t)$ and the discretized $s_h(t)$ are the design variables for this problem that controls and parametrizes the flow governed by Equation (1). The initial condition for the flow is a momentum perturbation on a uniform supersonic flow form left to right given by,

$$q(x, t=0) = \left(1, 1+\tilde{\rho u}(x), 1\right)^T, \tag{3}$$

where,

$$\tilde{\rho u}(x) = -0.01\exp\left(-\frac{(x-x_{\text{IC}})^2}{\sigma_{\text{IC}}^2}\right). \tag{4}$$

The center of the initial momentum perturbation is $x_{\text{IC}} = 0.25$, with a “width” of $\sigma_{\text{IC}} = 0.05$.

2.1 Functional

The functional considered is a "pressure sensor", which consists of the space-time integral of a weighted (squared) pressure difference:

$$J(q) = \int_0^2 \int_0^1 \kappa(x) \frac{1}{2} \left(p(x,t) - p_{\text{targ}}\right)^2 dx dt, \tag{5}$$

where $\kappa(x)$ is a weighted kernel defined by,

$$\kappa(x) = 10^{10} \exp\left(-\frac{(x - x_\kappa)^2}{\sigma_\kappa^2}\right), \tag{6}$$

with $x_\kappa = 0.85$ and $\sigma_\kappa = 0.05$. The target pressure is $p_{\text{targ}} = 0.2$.

2.2 Discretization

The 1D Euler equations are discretized using a discontinuous spectral element method. The spatial summation by parts integration on node i of element k the discrete residual takes the form

$$R^x_{k,i}\left(q_h, s_h\right) = -\sum_{j=1}^{n_{\text{sbp}}} Q_{j,i} F_{k,j} + \delta_{i n_{\text{sbp}}} \hat{F}_{k,n_{\text{sbp}}} - \delta_{i1} \hat{F}_{k,1} - G_{k,i}, \tag{7}$$

where $F_{k,j} \equiv F\left(q_{k,j}\right)$ and $G_{k,j} \equiv G\left(x_{k,j}, s_{k,j}\right)$. Combining this with the ~~time discretization and using an~~ Implicit Crank-Nicholson time marching scheme, the discretization becomes,

$$\hat{R}^{(n+1)}\left(q_h^{(n)}, q_h^{(n+1)}, s^{(n)}, s^{(n+1)}\right) = \frac{1}{\Delta t} M\left(q_h^{(n+1)} - q_h^{(n)}\right) + \frac{1}{2}\left[R_h^x\left(q_h^{(n+1)}, s^{(n+1)}\right) + R_h^x\left(q_h^{(n)}, s^{(n)}\right)\right] = 0. \tag{8}$$

The method `unsteadySolveCN!` advances the solution from time $t = 0$ to $t = T$ using the second-order Crank-Nicholson method described above. This method uses `stepCN!` to perform one step of Crank-Nicholson, where Newton's method is used to solve the nonlinear system described in Equation(8) and find $q_h^{(n+1)}$.

3 Discrete adjoint formulation

Our goal is to optimize the discrete design variables $s^{(1)}, s^{(2)}, \ldots, s^{(N)}$ to minimize the discrepancy between the computed pressure and the target pressure. To achieve this, we employ the discrete adjoint formulation to efficiently compute the total gradients of the objective functional with respect to the design variables, denoted as $\frac{DJ_h}{Ds^{(n)}}$. The optimization problem is formulated by constructing the Lagrangian of the system, which incorporates the governing equations at every time-step as residuals, given by:

$$\begin{aligned} L_h\left(q_h^{(1)}, q_h^{(2)}, \ldots, q_h^{(N)}, s^{(1)}, s^{(2)}, \ldots, s^{(N)}\right) &= J_h\left(q_h^{(1)}, q_h^{(2)}, \ldots, q_h^{(N)}\right) \\ &\quad + \psi_h^{(1)^T} M\left(q_h^{(1)} - q_{\text{IC}}\right) \\ &\quad + \sum_{n=1}^{N-1} \psi_h^{(n+1)^T}\left\{M\left(q_h^{(n+1)} - q_h^{(n)}\right) + \frac{\Delta t}{2}\left[R_h^x\left(q_h^{(n+1)}, s^{(n+1)}\right) + R_h^x\left(q_h^{(n)}, s^{(n)}\right)\right]\right\}. \end{aligned} \tag{9}$$

3.1 Finding adjoints

The primal problem is recovered by setting the stationary points $\partial L_h / \partial \psi_h^{(n)^T} = 0,\ n = 1, 2, \ldots, N$ and the states $q_h^{(n)}$ can be found. We solve for the adjoints by setting the stationary points $\partial L_h / \partial q_h^{(n)} = 0$.

$$\frac{\partial L_h}{\partial q_h^{(1)}} = \frac{\partial J_h}{\partial q_h^{(1)}} + \psi_h^{(1)^T} M + \psi_h^{(2)^T} \left(-M + \frac{\Delta t}{2} \frac{\partial R_h^x}{\partial q_h^{(1)}} \right) = 0, \tag{10a}$$

$$\frac{\partial L_h}{\partial q_h^{(n)}} = \frac{\partial J_h}{\partial q_h^{(n)}} + \psi_h^{(n)^T} \left(M + \frac{\Delta t}{2} \frac{\partial R_h^x}{\partial q_h^{(n)}} \right) + \psi_h^{(n+1)^T} \left(-M + \frac{\Delta t}{2} \frac{\partial R_h^x}{\partial q_h^{(n)}} \right) = 0,\ n = 2, 3, \ldots, N-1, \tag{10b}$$

$$\frac{\partial L_h}{\partial q_h^{(N)}} = \frac{\partial J_h}{\partial q_h^{(N)}} + \psi_h^{(N)^T} \left(M + \frac{\Delta t}{2} \frac{\partial R_h^x}{\partial q_h^{(N)}} \right) = 0. \tag{10c}$$

We calculate $\frac{\partial R_h^x}{\partial q_h}$ using the `calcStateJacobian!` method which uses color coding and complex step to find the state jacobian. Subsequently, we use the method `solveAdjoint` to solve the time-marching problem in reverse mode and find $\psi_h^{(n)},\ n = N, N-1, \ldots, 1$.
We have implemented analytical derivatives to compute $\frac{\partial J_h}{\partial q_h^{(n)}}$ in `calcTimeIntegratedObjectivedJdqn!`. Julia script `test_dJdqn.jl` verifies the accuracy of the RMS error from Equation (11) between the analytical derivatives and the complex-step approximates as shown in Fig 1d. During this verification, the complex step method showed a convergence trend towards the analytical gradients as the step size decreased. Initially, for larger perturbations, the results exhibited behavior resembling second-order convergence, similar to central difference schemes. However, as the step size became smaller, the complex step method distinguished itself by avoiding truncation and round-off errors inherent to finite difference approximations. This allowed the computed gradients to align closely with the analytical values, achieving machine precision-level accuracy and demonstrating the robustness of the implementation.
We plotted the adjoint variables $\psi_\rho, \psi_{\rho u}, \psi_e$ at times $t = 0.5, 0.25, 0.0$ in Fig 1a-1c. We can notice the adjoints moving backward in time from the right side of the domain at $t = 0.5$ towards the left at $t = 0$.

$$e_{rms} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(\left\| \left. \frac{\partial J_h}{\partial q_h^{(n)}} \right|_{ad} - \left. \frac{\partial J_h}{\partial q_h^{(n)}} \right|_{cs} \right\| \right)^2} \tag{11}$$

3.2 Finding total derivatives

With the states $q_h^{(n)}$ found by solving the primal problem and the adjoints $\psi_h^{(n)}$ found from solving the reverse mode problem described in Equations (10a)-(10c), we can find the total derivatives $DJ_h / Ds^{(n)}$ using,

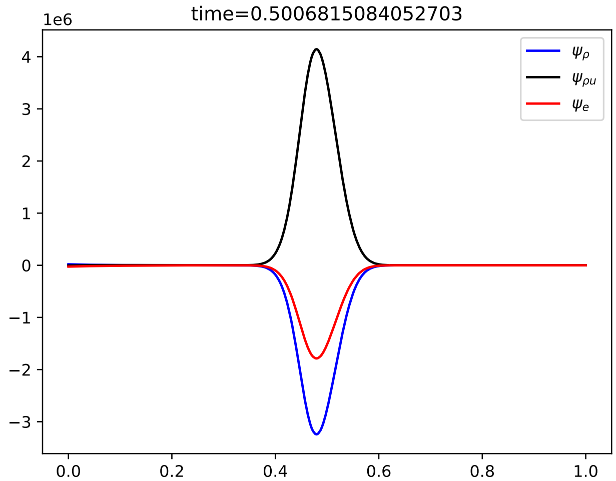
$$\frac{DJ_h}{Ds^{(1)}} = \frac{\partial J_h}{\partial s^{(1)}} + \psi_h^{(2)^T} \left\{ \frac{\Delta t}{2} \frac{\partial R_h^x}{\partial s^{(1)}} \right\},\ n = 1 \tag{12a}$$

$$\frac{DJ_h}{Ds^{(n)}} = \frac{\partial J_h}{\partial s^n} + \left(\psi^{(n)^T} + \psi^{(n+1)^T} \right) \left\{ \frac{\Delta t}{2} \frac{\partial R_h^x}{\partial s^{(n)}} \right\},\ n = 2, 3, \ldots, N-1 \tag{12b}$$

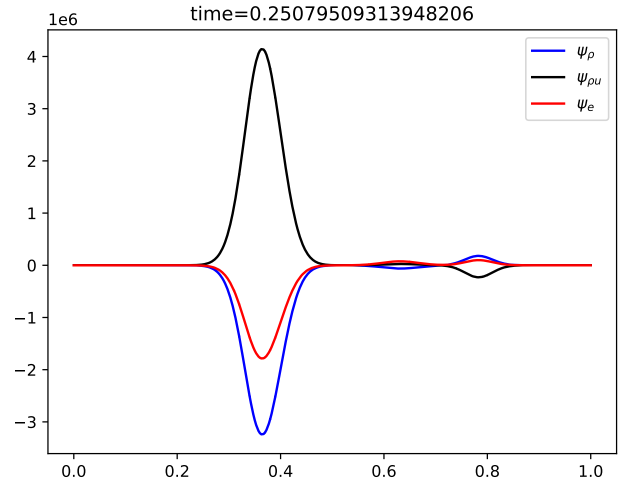
$$\frac{DJ_h}{Ds^{(N)}} = \frac{\partial J_h}{\partial s^{(N)}} + \psi^{(N)^T} \left\{ \frac{\Delta t}{2} \frac{\partial R_h^x}{\partial s^{(N)}} \right\},\ n = N \tag{12c}$$

where,

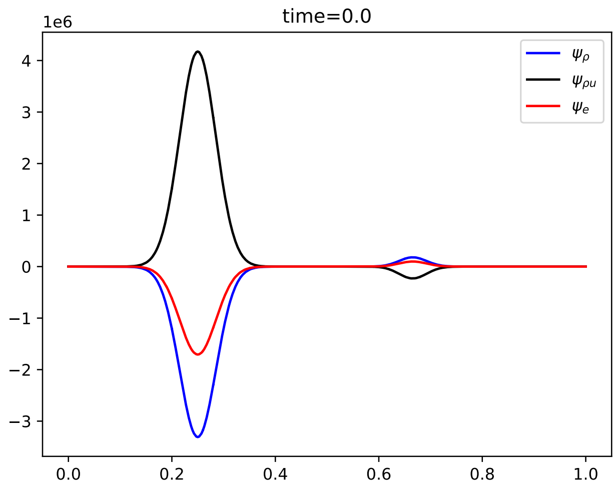
$$\frac{\partial R_{k,i}^x}{\partial s^{(n)}} = -\frac{\partial G_{k,i}}{\partial s^{(n)}} = -\left(0, \exp\left(-\frac{(x_{k,i} - x_s)^2}{\sigma_s^2} \right), 0 \right)^T. \tag{13}$$



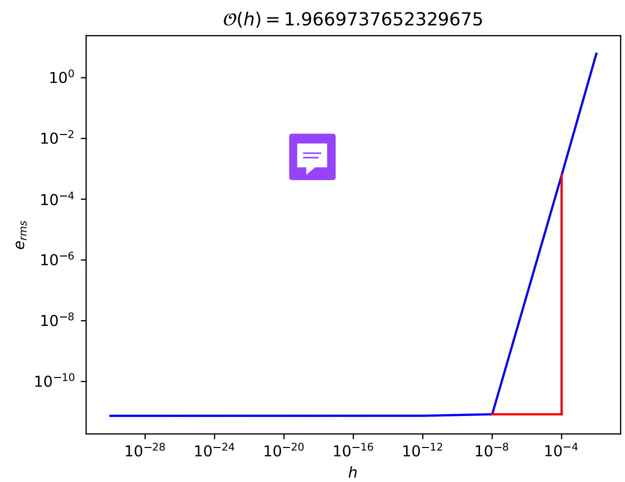
(a) Adjoint variables at $t = 0.5$



(b) Adjoint variables at $t = 0.25$



(c) Adjoint variables at $t = 0$



(d) Verification of $\frac{\partial J_h}{\partial q_h^{(n)}}$ using complex-step method

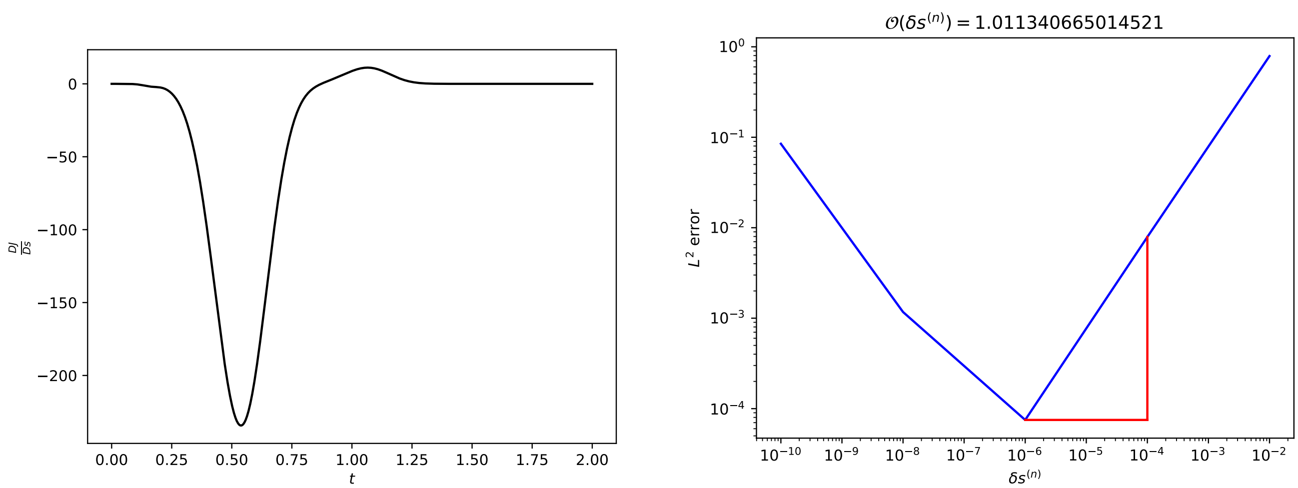
Figure 1: Adjoint variables computed from the Implicit Crank-Nicholson time-stepping algorithm being used to time march the primal problem.

$\partial J_h/\partial s^{(n)} = 0, \forall\ n \in [1, N]$ as the functional is not directly dependent on the source term. We find the jacobian $\frac{\partial R_h^x}{\partial s^{(n)}}$ in `calcWeakResidualJacobiandRds!`. We verify this implementation by comparing it against a directional derivative. Equation (14) provides a forward difference approximate to the total derivatives we need to perform gradient-based optimization.

$$\frac{DJ_h}{Ds^{(n)}} \approx \lim_{\delta s^{(n)} \to 0} \frac{J\left(q_h(t); s^{(n)} + \delta s^{(n)}\right) - J\left(q_h(t); s^{(n)}\right)}{\delta s^{(n)}} \tag{14}$$

We can verify the first order accuracy from Fig 2b, the finite difference approximates $\left(\left.\frac{DJ_h}{Ds_h}\right|_{\text{fd}}\right)$ by performing the error convergence analysis in the L^2 norm using the adjoint based total derivatives $\left(\left.\frac{DJ_h}{Ds_h}\right|_{\text{ad}}\right)$ as ground truth.

$$e_{h,\delta s^{(n)}} = \sqrt{\sum_{n=1}^{N} \left| \left.\frac{DJ_h}{Ds^{(n)}}\right|_{\text{ad}} - \left.\frac{DJ_h}{Ds^{(n)}}\right|_{\text{fd},\delta s^{(n)}} \right|^2} \tag{15}$$



(a) Total gradients $\frac{DJ_h}{Ds^{(n)}}$ as a function of time t (b) Verification of $\frac{DJ_h}{Ds^{(n)}}$ using forward-difference method

Figure 2: A plot of gradient over time using the time-varying adjoints

4 Acknowledgements

I gratefully acknowledge the assistance provided by ChatGPT in formulating and refining sections of this work.