Adjoint Lattice Boltzmann for Optimal Control

Grzegorz Gruszczyński

Łukasz Łaniewski-Wołłk, prof. Jacek Szumbarski

Faculty of Power and Aeronautical Engineering
Warsaw University of Technology

April 3, 2017

Table of contents

Motivation

Goal

Framework for solving optimal control problems of a discrete dynamical systems in the CFD area.
study case: optimal mixing of fluid

Motivation

Goal

Framework for solving optimal control problems of a discrete dynamical systems in the CFD area.
study case: optimal mixing of fluid

Method

- Lattice Boltzmann on GPU
- Adjoint
- Automatic Differentiation

Problem statement

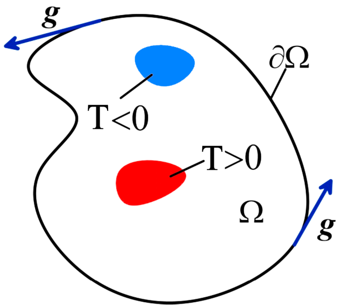
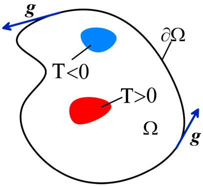


Figure: Flow Domain

Fluid Motion

Incompressible Navier Stokes and continuity equation:

$$\begin{cases} \partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \nu \Delta \boldsymbol{u} \\ \nabla \cdot \boldsymbol{u} = 0 \end{cases}$$



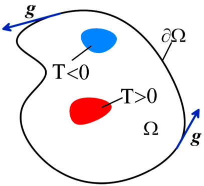
Boundary and Initial Conditions for the fluid:

$$\boldsymbol{u}\Big|_{\partial\Omega} = \boldsymbol{g} \quad ; \quad \boldsymbol{g} \cdot \boldsymbol{n}\Big|_{\partial\Omega} = 0 \quad ; \quad \boldsymbol{u}\Big|_{t=0} = 0$$

Passive Scalar

Advection - diffusion equation:

$$\partial_t T + \boldsymbol{u} \cdot \nabla T = \lambda \nabla^2 T$$



Boundary and Initial Conditions for the Passive Scalar:

$$\frac{\partial}{\partial \boldsymbol{n}} T \bigg|_{\partial\Omega} = 0 \quad ; \quad T \bigg|_{t=0} = T_0$$

Objective function

Find an extremum of the functional:

$$J[\boldsymbol{u}, T] = \underbrace{\int_\Omega [T(t_{end}) - \overline{T}]^2 d\boldsymbol{x}}_{I_1 \text{ - mix quality}} + \varepsilon \underbrace{\left[\int_0^{t_{end}} \left(\int_{\partial\Omega} 2\nu \boldsymbol{u} \cdot \boldsymbol{D_u} \boldsymbol{n} dS \right) dt \right]^2}_{I_2 \text{ - work needed to impose motion}}$$

where:

$$\boldsymbol{D_u} = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla^T \boldsymbol{u}) \quad - \quad \text{deformation rate tensor}$$

$$\overline{T} = \frac{1}{|\Omega|} \int_\Omega T d\boldsymbol{x} \quad - \quad \text{average value of the passive scalar}$$

$$\varepsilon \quad - \quad \text{weight coefficient}$$

The Lattice Boltzmann equation

$$\underbrace{f_i(\boldsymbol{x} + \boldsymbol{e_i}\Delta\boldsymbol{x}, t + \Delta t) - f_i(\boldsymbol{x}, t)}_{\textit{Streaming}} = \underbrace{-\frac{1}{\tau}(f_i - f_i^{eq})}_{\textit{Collision}}$$

- τ - relaxation parameter, $\tau = 3\nu\frac{(\Delta x)^2}{\Delta t} + \frac{1}{2}$ where ν is the kinematic viscosity
- f_i - discrete probability distribution function

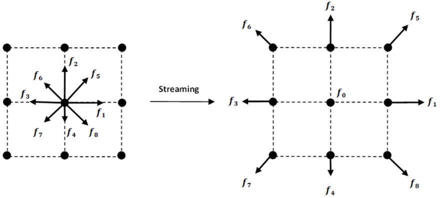


Figure: D2Q9: Streaming

Algorithm: Fluid

1 Initialize f_i^{in}

Algorithm: Fluid

1 Initialize f_i^{in}

2 Compute ρ, $\boldsymbol{u}(\boldsymbol{x}, t)$

$$\rho = \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t) \qquad \textit{and} \qquad \boldsymbol{u}(\boldsymbol{x}, t) = \frac{1}{\rho} \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t)\boldsymbol{e}_i$$

Algorithm: Fluid

1 Initialize f_i^{in}

2 Compute ρ, $\boldsymbol{u}(\boldsymbol{x}, t)$

$$\rho = \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t) \qquad \textit{and} \qquad \boldsymbol{u}(\boldsymbol{x}, t) = \frac{1}{\rho} \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t)\boldsymbol{e}_i$$

3 Compute $f_i^{eq}(\boldsymbol{x}, t)$

$$f_i^{eq}(\boldsymbol{x}, t) = w_i \rho(\boldsymbol{x}, t) \left[1 + 3\frac{\boldsymbol{e}_i \boldsymbol{u}}{e^2} + \frac{9}{2}\frac{(\boldsymbol{e}_i \boldsymbol{u})^2}{e^4} - \frac{3}{2}\frac{\boldsymbol{u}^2}{e^2}\right]$$

Algorithm: Fluid

1 Initialize f_i^{in}

2 Compute ρ, $\boldsymbol{u}(\boldsymbol{x}, t)$

$$\rho = \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t) \qquad \textit{and} \qquad \boldsymbol{u}(\boldsymbol{x}, t) = \frac{1}{\rho} \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t)\boldsymbol{e}_i$$

3 Compute $f_i^{eq}(\boldsymbol{x}, t)$

$$f_i^{eq}(\boldsymbol{x}, t) = w_i \rho(\boldsymbol{x}, t) \left[1 + 3\frac{\boldsymbol{e}_i \boldsymbol{u}}{e^2} + \frac{9}{2}\frac{(\boldsymbol{e}_i \boldsymbol{u})^2}{e^4} - \frac{3}{2}\frac{\boldsymbol{u}^2}{e^2} \right]$$

4 BGK Collision

$$f_i^{out}(\boldsymbol{x}, t) = f_i^{in}(\boldsymbol{x}, t) - \frac{1}{\tau_f}\left(f_i^{in}(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t) \right)$$

Algorithm: Fluid

1 Initialize f_i^{in}

2 Compute ρ, $\boldsymbol{u}(\boldsymbol{x}, t)$

$$\rho = \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t) \qquad \textit{and} \qquad \boldsymbol{u}(\boldsymbol{x}, t) = \frac{1}{\rho} \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t)\boldsymbol{e}_i$$

3 Compute $f_i^{eq}(\boldsymbol{x}, t)$

$$f_i^{eq}(\boldsymbol{x}, t) = w_i \rho(\boldsymbol{x}, t) \left[1 + 3\frac{\boldsymbol{e}_i \boldsymbol{u}}{e^2} + \frac{9}{2}\frac{(\boldsymbol{e}_i \boldsymbol{u})^2}{e^4} - \frac{3}{2}\frac{\boldsymbol{u}^2}{e^2}\right]$$

4 BGK Collision

$$f_i^{out}(\boldsymbol{x}, t) = f_i^{in}(\boldsymbol{x}, t) - \frac{1}{\tau_f}\left(f_i^{in}(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t)\right)$$

5 Streaming

$$f_i^{in}(\boldsymbol{x} + \boldsymbol{e}_i, t + 1) = f_i^{out}(\boldsymbol{x}, t)$$

Algorithm: Passive Scalar

Advection - Diffusion of the T is solved on a separate D2Q5 lattice

1 Initialize $\theta_i^{in}(\boldsymbol{x}, t)$

2 Compute $T(\boldsymbol{x}, t)$

$$T(\boldsymbol{x}, t) = \sum_{i=0}^{4} \theta_i^{in}(\boldsymbol{x}, t)$$

3 Compute $\theta_i^{eq}(\boldsymbol{x}, t)$

$$\theta_i^{eq}(\boldsymbol{x}, t) = T(\boldsymbol{x}, t) w_i \left[1 + 3 \frac{\boldsymbol{e}_i \boldsymbol{u}}{e^2} \right]$$

4 BGK Collision

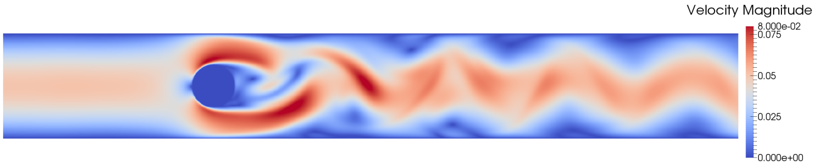
$$\theta_i^{out}(\boldsymbol{x}, t) = \theta_i^{in}(\boldsymbol{x}, t) - \frac{1}{\tau_T} \left(\theta_i^{in}(\boldsymbol{x}, t) - \theta_i^{eq}(\boldsymbol{x}, t) \right)$$

5 Streaming

$$\theta_i^{in}(\boldsymbol{x} + \boldsymbol{e}_i, t + 1) = \theta_i^{out}(\boldsymbol{x}, t)$$

Validation

- flow - frequency of shedding of von Karman vortices
- advection
- diffusion
- passive scalar conservation
- wall shear force - Couette flow



Why adjoint?

Primal equation

$$Au = b$$

Find the value of a functional $h \cdot u$

$$h^T u =$$

Why adjoint?

Primal equation

$$Au = b$$

Dual equation

$$A^T v = h$$

Find the value of a functional $h \cdot u$

$$h^T u = (A^T v)^T u = v^T \underbrace{Au}_{b} = v^T b$$

Discrete dynamical system

$$\begin{bmatrix} u_{n+1} \\ g_{n+1} \end{bmatrix} = H(u_n, \alpha_n)$$

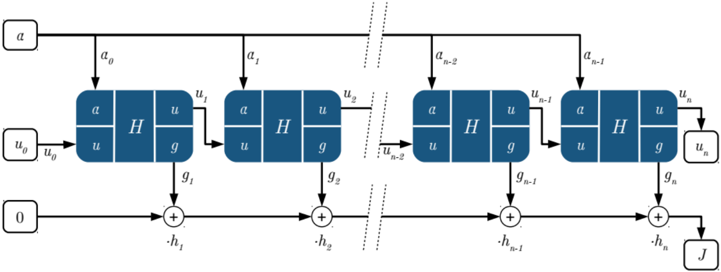


Figure: Discrete iterative process (from Łaniewski-Wołłk [1])

Discrete dynamical system

$$\begin{bmatrix} u_{n+1} \\ g_{n+1} \end{bmatrix} = H(u_n, \alpha_n)$$

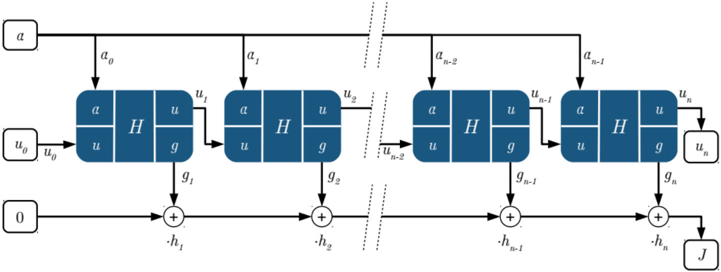


Figure: Discrete iterative process (from Łaniewski-Wołłk [1])

Find the derivative of the objective $J = \mathbf{h} \cdot \mathbf{g}$ with respect to a formal differentiation parameter s:

$$\frac{d}{ds}J = \frac{d}{ds}(\mathbf{h} \cdot \mathbf{g}) = \sum_{n=1}^{N} h_n \cdot \frac{\partial g_n}{\partial s}$$

Discrete dynamical system

$$\begin{bmatrix} u_{n+1} \\ g_{n+1} \end{bmatrix} = H(u_n, \alpha_n) \xrightarrow{\text{adjoint}} \begin{bmatrix} v_{n-1} \\ \beta_{n-1} \end{bmatrix} = [dH]^T \begin{bmatrix} v_n \\ h_n \end{bmatrix}$$

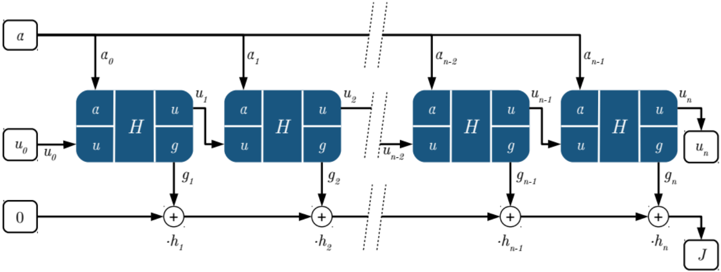


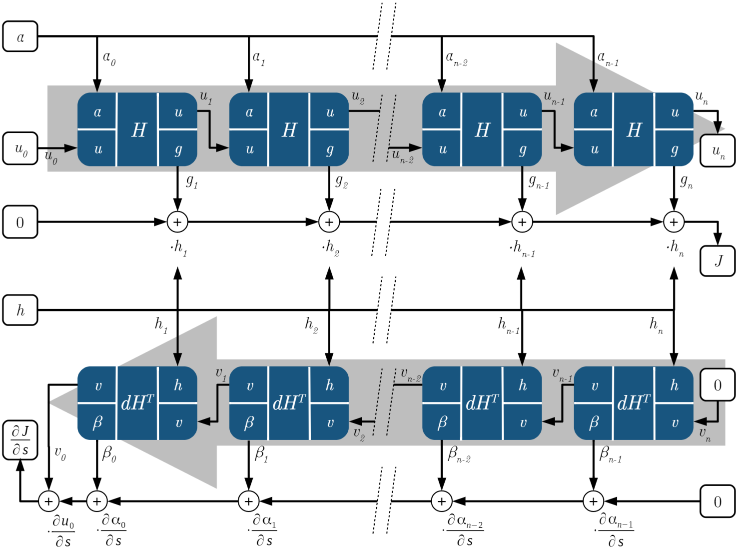
Figure: Discrete iterative process (from Łaniewski-Wołłk [1])

Find the derivative of the objective $J = \mathbf{h} \cdot \mathbf{g}$ with respect to a formal differentiation parameter s:

$$\frac{d}{ds} J = \frac{d}{ds}(\mathbf{h} \cdot \mathbf{g}) = \sum_{n=1}^{N} h_n \cdot \frac{\partial g_n}{\partial s} \xrightarrow{\text{adjoint}} v_0 \cdot \frac{\partial u_0}{\partial s} + \sum_{n=0}^{N-1} \beta_n \cdot \frac{\partial \alpha_n}{\partial s}$$

The primal → adjoint formulation:

$$\begin{bmatrix} u_{n+1} \\ g_{n+1} \end{bmatrix} = H(u_n, \alpha_n) \xrightarrow{\text{adjoint}} \begin{bmatrix} v_{n-1} \\ \beta_{n-1} \end{bmatrix} = [dH]^T \begin{bmatrix} v_n \\ h_n \end{bmatrix}$$



Automatic differentiation

To compute the derivatives the automatic differentiation is used.
Automatic differentiation is not:

- Symbolic differentiation
- Numerical differentiation (the method of finite differences)

Automatic differentiation

To compute the derivatives the automatic differentiation is used.
Automatic differentiation is not:

- Symbolic differentiation
- Numerical differentiation (the method of finite differences)

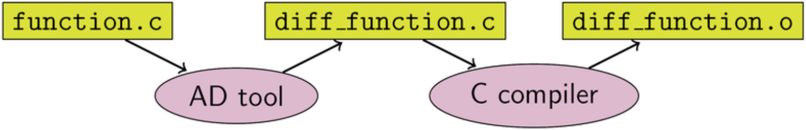


Figure: Source Code Transformation

Study case - lid driven cavity

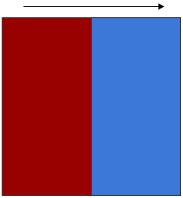


Figure: Initial conditions for the passive scalar distribution, fluid is at rest

- Reference control function: $U_{lid} = 0.1\, sin(t)$ and $t \in [0, 2\pi]$
- Lattice size (with walls): 128x128
- Lattice fluid viscosity : $\nu = 0.1$
- Lattice fluid thermal diffusivity: $\lambda = 0.005$
- passive scalar intensity: half domain $\pm 1 \Rightarrow$ average $\overline{T} = 0$

Study case - results

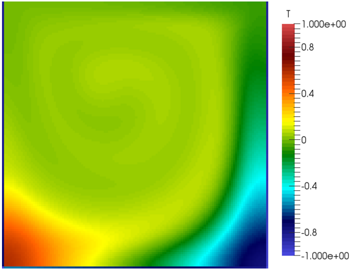


Figure: Initial control

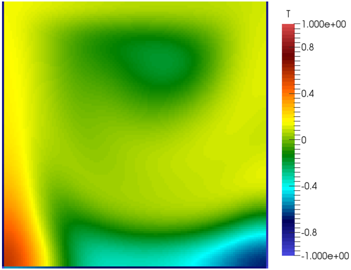
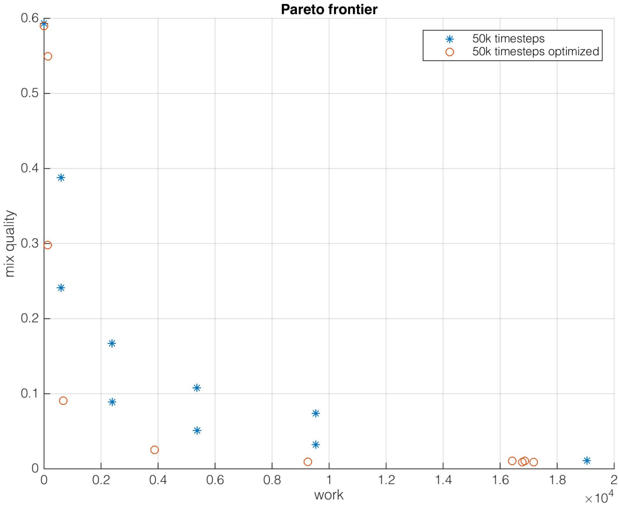


Figure: Optimized control

	mix quality	**work**	ϵ - **work weight**	**J**
initial control	0.032	9536	1E-5	0.127
optimized control	0.025	3873	1E-5	0.064

Pareto Frontier



Conclusions - What we achieved:

Feature:

Benefits:

Conclusions - What we achieved:

Feature:	**Benefits:**
framework for optimization of discrete dynamical system	applicable to complex CFD problems like optimal control or topology optimization

Conclusions - What we achieved:

Feature:	**Benefits:**
framework for optimization of discrete dynamical system	applicable to complex CFD problems like optimal control or topology optimization
Lattice Boltzmann Method	almost ideal scalability on HPC

Conclusions - What we achieved:

Feature:	**Benefits:**
framework for optimization of discrete dynamical system	applicable to complex CFD problems like optimal control or topology optimization
Lattice Boltzmann Method	almost ideal scalability on HPC
adjoint	reduce numerical cost of finding the derivative of the objective depending on many control variables

Conclusions - What we achieved:

Feature:	Benefits:
framework for optimization of discrete dynamical system	applicable to complex CFD problems like optimal control or topology optimization
Lattice Boltzmann Method	almost ideal scalability on HPC
adjoint	reduce numerical cost of finding the derivative of the objective depending on many control variables
automatic differentiation	exact derivatives, code maintainability

Questions?

?

References I

- Ł. Łaniewski-Wołłk. "Adjoint Lattice Boltzmann for Topology Optimalization and Optimal Control on multi-GPU Architecture". PhD thesis. Warsaw Univerisity of Technology, Faculty of Power and Aeronautical Engineering, 2016 (in preparation).
- Ł. Łaniewski-Wołłk, M. Dzikowski, et al. *TCLB*. C-CFD Group at Warsaw University of Technology. 2012. URL: https://github.com/CFD-GO/TCLB.