Robust Reduced Order Models Of A Wake Controlled By Jets

Jessie Weller   Edoardo Lombardi   Angelo Iollo

Institut de Mathématiques de Bordeaux

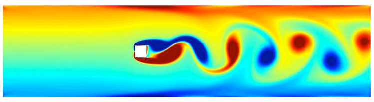
INRIA project MC2

Acknowledgments :
We thank S. Camarri and M.-V. Salvetti from University of Pisa
for help with adding the jets to the Navier-Stokes code

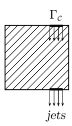
7th Ercoftac Sig33 - Flubio Workshop
on Open issues in transistion and flow control
october 16-18 2008

Setup

- Confined square cylinder + incompressible Navier-Stokes at Re=150



- Simulation of a blowing/suction control using actuators placed on the cylinder :



$$\forall x \in \Gamma_c, \quad v(\mathbf{x}, t) = c(t) \text{ (control law)}$$

Choice of actuation

Two types of actuation were used :

- **Precomputed control law**
  A function $c(t)$ defined on $[0, T_{max}]$ was chosen before starting the computation and given as input to the Navier-Stokes equations. Outside this time interval $c(t) = 0$.

- **Feedback control law**
  Function $c(t)$ is defined as a proportional feedback law :

$$c(t) = \sum_{j=1}^{N_s} K_j v(\mathbf{x}_j, t)$$

  where x_j is a point in the cylinder wake, K_j a gain coefficient and N_s the number of sensors.

Reduced order modelling using POD

- Reduced-order solution :

$$\mathbf{u}_R(\mathbf{x}, t) = \sum_{r=1}^{N_r} a_r(t)\phi^r(\mathbf{x})$$

  avec $N_r << N_{grid}$.

- Base functions ϕ^r are obtained by Proper Orthogonal Decomposition.

- Reduced order model : POD-Galerkin model

$$\left(\partial_t \mathbf{u}_R + \mathbf{u}_R \cdot \nabla \mathbf{u}_R + \nabla p - \frac{1}{Re}\Delta \mathbf{u}_R, \phi^r\right) = 0$$

  is a system of ordinary differential equations.

Proper orthogonal decomposition

- Simulation of Navier-Stokes over $[0, T_{max}]$

$$\Rightarrow \{\mathbf{u}^i(\mathbf{x}) = \mathbf{u}(\mathbf{x}, t^i)\}_{i=1..N_t} = \textbf{snapshots}$$

- Define $\mathcal{L} = span\{\mathbf{u}^1, \cdots, \mathbf{u}^{N_t}\}$
  Find low-dimensional subspace of $\mathcal{L}$ that gives *the best approximation* of $\mathcal{L}$:

  Find orthogonal functions $\{\phi^r\}_{r=1...N_r}$ with $N_r << N_t$ and the coefficients $\hat{a}_k^i$ such that :

$$\sum_{i=1}^{N_t} \left\| \mathbf{u}^i - \sum_{r=1}^{N_r} \hat{a}_k^i \phi^r \right\|_{L^2(\Omega)}^2 \quad \text{is minimal}$$

POD-ROM of flow past a bluff body with actuators

- Actuators turned on once the flow is fully developped ($t = 0$).
- Navier-Stokes simulations with control laws $c^i(t)$, $i = 1, N_c$
- Each simulation is over a period $[0, T_{max}]$.

  $\Rightarrow$ solutions at N_t time instants : $\mathbf{u}^i(\mathbf{x}, t^k), k = 1..N_t, \; i = 1..N_c$

- Definition of snapshots for building a POD basis :

$$\mathbf{w}^{k,i}(\mathbf{x}) = \mathbf{u}^i(\mathbf{x}, t^k) - \mathbf{u}_0(\mathbf{x}) - c^i(t^k)\mathbf{u}_c(\mathbf{x})$$

  where functions $\mathbf{u}_0$ and $\mathbf{u}_c$ are chosen such that the snapshots are equal to zero at inflow, outflow, *and jet* boundaries.
- Build one POD basis $\Phi^k(c^1, \cdots, c^{N_c})$

POD-ROM of flow past a bluff body

We wish the reduced order model to be :

- **Accurate** when integrated with database control law(s)
- **Robust** to changes in the control laws applied
- a tool for **Optimization** : to be used to estimate descent directions

Building an accurate ROM

- In Navier-Stokes $\mathbf{u}(\mathbf{x}, t)$ is replaced by $\mathbf{u}_0 + c(t)\mathbf{u_c} + \sum_{k=1}^{N_r} a_k(t)\Phi_k(\mathbf{x})$.
- Projection onto the POD modes leads to a system of ODEs :

$$\left\{ \begin{array}{rcl} \dot{a}_r(t) & = & f_r(\mathbf{a}(t), c(t), \widehat{\mathbf{X}}) \\ a_r(0) & = & a_r^0 \end{array} \right. \qquad 1 \leq r \leq N_r$$

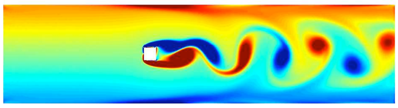
$$\text{where : } f_r(\mathbf{a}(t), c(t), \widehat{X}) = \widehat{A}_r + \widehat{C}_{kr} a_k(t) - \widehat{B}_{ksr} a_k(t) a_s(t) \\ - \widehat{E}_r \dot{c}(t) - \widehat{F}_r c^2(t) + [\widehat{G}_r - \widehat{H}_{kr} a_k(t)] c(t)$$

System matrices $\widehat{A}$, $\widehat{B}$, $\widehat{C}$, $\widehat{E}$, $\widehat{F}$, $\widehat{G}$ and $\widehat{H}$ depend only on $\mathbf{u}_0$, $\mathbf{u}_c$ and the modes Φ_r and their derivatives.

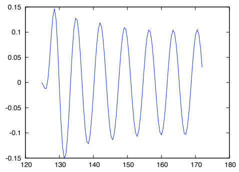
Model Accuracy on an example with $N_c = 1$

A control Law obtained by feedback

- 10 sensors placed in the cylinder wake, $\mathbf{K} = [0.2\,0.2\,0.2\,0.2\,0\,0\,0\,0\,0\,0]$



- Control Law obtained



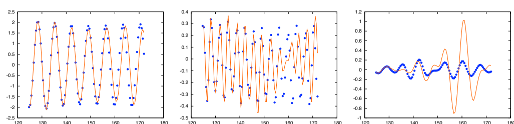
Model Accuracy (1)

- A small number of modes are needed to capture the energy in a snapshot :

$$\text{If } \hat{a}_k(t) = \left(\mathbf{u}(\cdot, t), \Phi_k^{\mathcal{C}}\right)_2$$

$$\text{Then } \mathbf{u}_0(\mathbf{x}) + c(t)\mathbf{u_c}(\mathbf{x}) + \sum_{k=1}^{N_r} \hat{a}_k(t)\Phi_k^{\mathcal{C}}(\mathbf{x}) \approx \mathbf{u}(\mathbf{x}, t) \text{ for } N_r \text{ small}$$

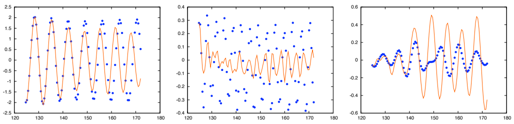
- **but** there are differences between, 'a' (solution of the ROM) and '$\hat{a}$'



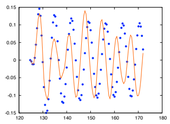
a_k vs. $\hat{a}_k$ for $k = 1, k = 5$ and $k = 10$

Model Accuracy (2)

These differences are emphasized is the case of feedback control laws :



a_k vs. $\hat{a}_k$ for $k = 1$,$k = 5$ and $k = 10$



Control Law : N-S output vs. ROM output

Calibration

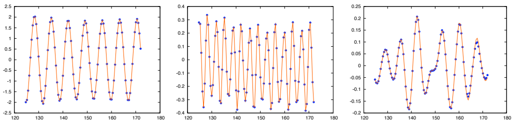
$\longrightarrow$ Adjust certain system matrices so as to minimize the difference between $\hat{a}_k$ and a_k
where $\hat{a}_k(t^n)$ is the the projection of the nth snapshot on the kth mode.

$\longrightarrow$ Small linear system, independant of N_c and N_t :

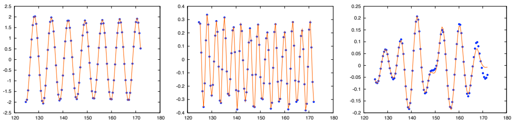
$$\min_X \int_0^T \sum_{k=1}^{N_r} \sum_{i=1}^{N_c} \left(\dot{\hat{a}}^i_k(t) - f_k(\hat{\mathbf{a}}^{\mathbf{i}}(t), c^i(t), X)\right)^2 dt + \alpha \|X - \widehat{X}\|^2$$

where $\hat{a}^i_k(t^n) = \left(\mathbf{u}^i(\cdot, t^n), \Phi_k\right)$
with α a small regularization parameter .

Effect of calibration (1)

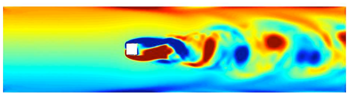


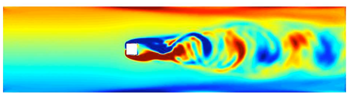
a_k vs. $\hat{a}_k$ for $k = 1, k = 5$ and $k = 10$



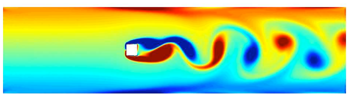
a_k vs. $\hat{a}_k$ for $k = 1, k = 5$ and $k = 10$

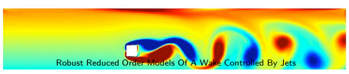
Effect of calibration (2)





Vorticity reconstruction before calibration, without and with feedback.





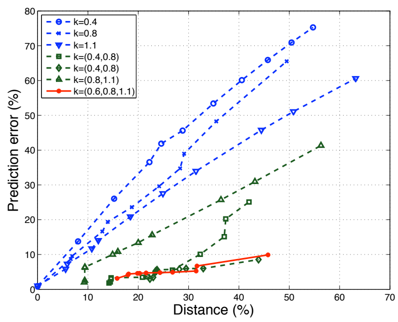
Multi-control Models

- **Three controls** $\Rightarrow$ Base solutions $\mathbf{u}^1$, $\mathbf{u}^2$, $\mathbf{u}^3$
- **Seven Databases = Seven POD bases = Seven Models**

$$\mathcal{U}_1 = \{\mathbf{u}^1\}, \mathcal{U}_2 = \{\mathbf{u}^2\}, \mathcal{U}_3 = \{\mathbf{u}^3\}$$
$$\mathcal{U}_4 = \{\mathbf{u}^1, \mathbf{u}^2\}, \mathcal{U}_5 = \{\mathbf{u}^1, \mathbf{u}^3\}, \mathcal{U}_6 = \{\mathbf{u}^2, \mathbf{u}^3\}$$
$$\mathcal{U}_7 = \{\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3\}$$

- Project base solutions onto POD bases : $\rightarrow a(\text{base } k, \text{pod } i)$

Numerical Results for model robustness



Using the Model for control (1)

Optimization problem

- Formulation :
  Find control law $c(t)$ to apply on Γ_c such that :

$$c = \text{argmin}_c \mathcal{F}(\mathbf{u}) = \text{argmin}_c \int_0^T \mathcal{J}(\mathbf{u}(\cdot, t))\, dt$$

  is minimal where $\mathbf{u}$ is solution of the incompressible N-S equations.

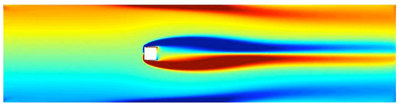
- Degrees of freedom :
  - if c is precomputed, c is projected onto a B-spline subspace of dimension N_b and optimization is performed over the B-spline control points.
  - if c is obtained by proportional feedback, optimization is performed over the gain coefficients.

$\rightarrow$ Optimization performed of a set of N_{opt} control coefficients.

Using the Model for control (2)

Choice of functional

Seek c that solves : $\min_c \|\mathbf{u} - \bar{\mathbf{u}}\|$
where $\bar{\mathbf{u}}$ is the steady unstable solution.



Steady unstable solution at Re=150

Using the Model for control (3)

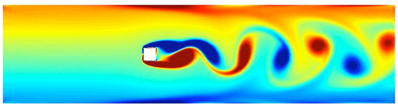
ROM/Optimization algorithm

- step 0 : choose c^0,run N-S with c^0, build POD-ROM model.
  Set $m = 0$
- step 1 (ROM) : Use the model to calculate gradient of the functional with respect to the control coefficients of c^m.
  Use gradient conjugate method to reduce functional, stop when
  - a minimum is reached
  - the change in control coefficients is *too large*
- step 2 (FULL NS) : Inject new control into N-S and reevaluate functional
  - if functional has evolved as predicted :
    Form database with new simulation and previous one
    m ← m+1
  - if not : Form a database including the previous solution(s) and the new one
- step 3 : Build POD-ROM model with defined database.
  Go to step 1.

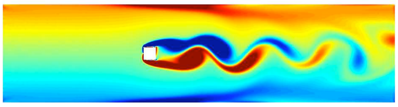
Using the Model for control (4)

Numerical results at Re=150

- time period ≈ 6 shedding cycles
- precomputed control



Initial vorticity at time T_{max}

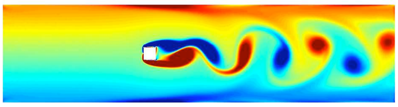


Vorticity at time T_{max} after ROM/Optimization

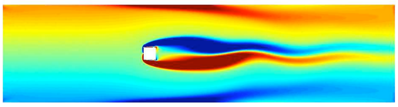
Using the Model for control (5)

Numerical results at Re=150

- time period ≈ 6 shedding cycles
- proportional feedback control with 10 sensors



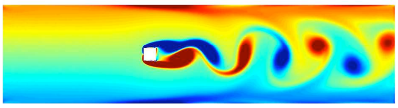
Initial vorticity at time T_{max}



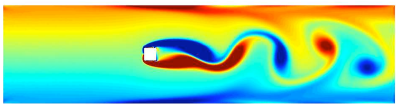
Vorticity at time T_{max} after ROM/Optimization

Using the Model for control (6)

And after T_max ?



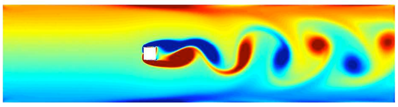
Initial vorticity at time T_{max}



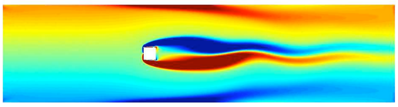
Vorticity at time $2T_{max}$ after ROM/Optimization on $[0, T_{max}]$.

Using the Model for control (7)

Optimization fter T_max ?



Initial vorticity at time T_{max}



Vorticity at time T_{max} after ROM/Optimization on $[0, T_{max}]$.

Conclusions and Future Work

- Accurate reduced-order model of the actuated flow, that is robust to parameter variations
- The model has been used to control the flow past a square cylinder, at Re=150
- Use the model *online*.
- Use More efficient controls : we are working on a proportional-integral control
- Add the pressure to the model