Numerical shape optimization using CFD

Praveen. C
praveen@math.tifrbng.res.in

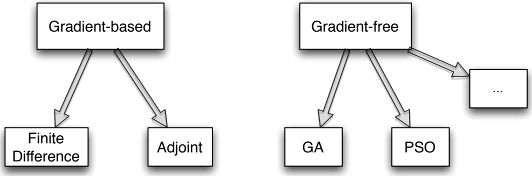
Tata Institute of Fundamental Research
Center for Applicable Mathematics
Bangalore 560065
http://math.tifrbng.res.in

12'th AeSI CFD Symposium
IISc, Bangalore
11–12 August, 2010

Outline

1. Approaches to optimization
2. Elements of shape optimization
   - Shape parameterization
   - Grid generation/deformation
   - CFD solution/Adjoint solution
   - Optimizer
3. Free-form deformation (FFD)
4. Particle swarm method
5. Surrogate models
6. Examples

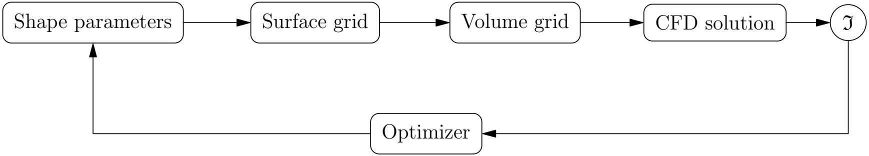
Approaches to optimization



- Local optimum
- FD accuracy problem
- Adjoint solver required
- Issues with adjoint consistency

- Global optimum possible
- "Easy" to implement for engg. problems
- Slow convergence: surrogate models and parallelization

Elements of shape optimization



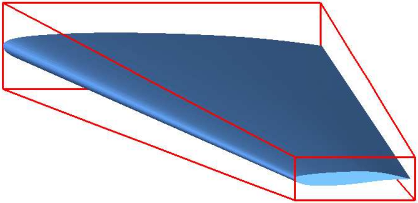
Shape parameterization

Shape parameterization approaches

- Aerodynamic DVs:
  - LE radius, max camber, taper ratio
- PARSEC, Kulfan parameterization, etc.
- BSplines/NURBS
- Need to re-generate surface/volume grid whenever shape is changed
- Or, use a free-form approach like RBF-based grid deformation

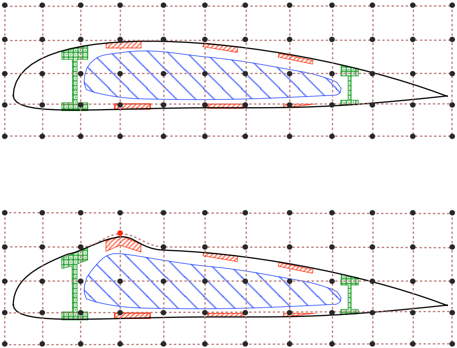
Free Form Deformation

- Originated in computer graphics field
- Embed the object inside a box and deform the box
- Independent of the representation of the object
- Deform CFD grid also, independent of grid type



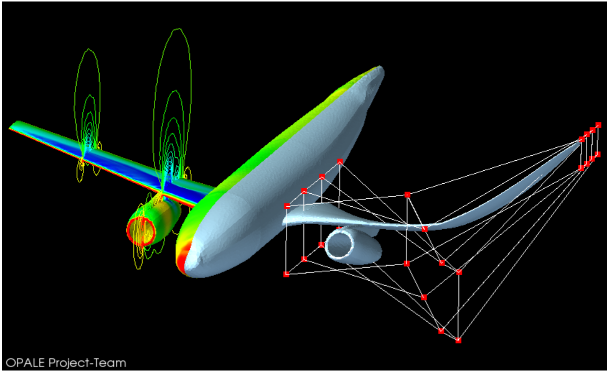
Free Form Deformation

Consistent parameterization	
Airplane shape DVs	
Compact set of DVs	
Smooth geometry	
Local control	
Analytical sensitivity	
Grid deformation	
Setup time	
Existing grids	
CAD connection	



(Samareh)

Free Form Deformation: Example



(R. Duvigneau, INRIA)

Free Form Deformation

- $X^0(P)$ = coordinate of point P wrt reference shape
- Movement of point P under the deformation

$$X(P) = X^0(P) + \sum_{i=0}^{n_i} \sum_{j=0}^{n_j} \sum_{k=0}^{n_k} Y_{ijk} B_i^{n_i}(\xi_p) B_j^{n_j}(\eta_p) B_k^{n_k}(\zeta_p)$$

- Bernstein polynomials

$$B_m^n(t) = C_m^n t^m (1-t)^{n-m}, \quad t \in [0,1], \quad m = 0, 1, \ldots, n$$

- Design variables

$$\{Y_{ijk}\}, \quad 0 \le i \le n_i, \quad 0 \le j \le n_j, \quad 0 \le k \le n_k$$

- Cannot change wing planform
- Wing twist can be added as additional variables

Optimizer

Particle swarm optimization

- Kennedy and Eberhart (1995)
- Modeled on behaviour of animal swarms: ants, bees, birds
- Cooperative behaviour of large number of individuals through simple rules
- Emergence of swarm intelligence

Optimization problem

$$\min_{x \in D} J(x), \quad D \subset \mathbb{R}^d$$

Particle swarm optimization

- Kennedy and Eberhart (1995)
- Modeled on behaviour of animal swarms: ants, bees, birds
- Cooperative behaviour of large number of individuals through simple rules
- Emergence of swarm intelligence

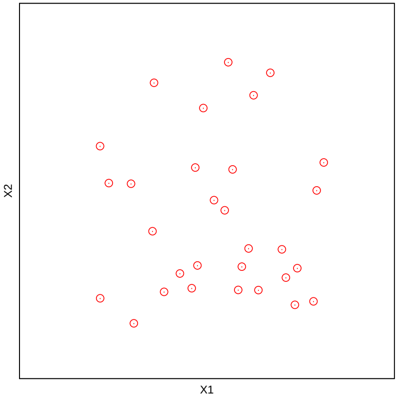
Optimization problem

$$\min_{x \in D} J(x), \quad D \subset \mathbb{R}^d$$

Particle swarm optimization

Particles distributed in design space

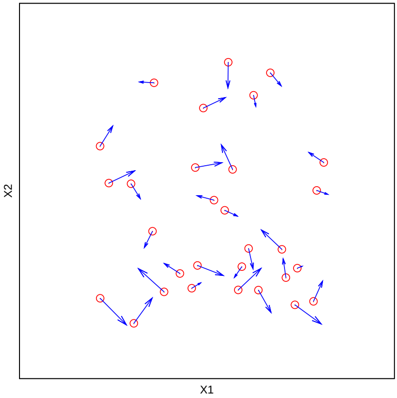
$$x_i \in D, \quad i = 1, ..., N_p$$



Particle swarm optimization

Each particle has a velocity

$$v_i \in \mathbb{R}^d, \quad i = 1, ..., N_p$$



Particle swarm optimization

- Particles have memory (t = iteration number)

$$\text{Local memory} \; : p_i^t = \underset{0 \le s \le t}{\text{argmin}} \, J(x_i^s)$$

$$\text{Global memory} \; : p^t = \underset{i}{\text{argmin}} \, J(p_i^t)$$

- Velocity update

$$v_i^{t+1} = \omega v_i^t + \underbrace{c_1 r_1^t \otimes (p_i^t - x_i^t)}_{Local} + \underbrace{c_2 r_2^t \otimes (p^t - x_i^t)}_{Global}$$

- Position update

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

PSO: embarassingly parallel

$$
\begin{array}{cccc}
x_1^n & x_2^n & \ldots & x_{N_p}^n \\
\downarrow & \downarrow & & \downarrow \\
\mathcal{J}(x_1^n) & \mathcal{J}(x_2^n) & \ldots & \mathcal{J}(x_{N_p}^n) \\
\downarrow & \downarrow & & \downarrow \\
\hline \\
v_1^{n+1} & v_2^{n+1} & \ldots & v_{N_p}^{n+1} \\
\downarrow & \downarrow & & \downarrow \\
x_1^{n+1} & x_2^{n+1} & \ldots & x_{N_p}^{n+1}
\end{array}
$$

Parallel evaluation of cost functions using MPI

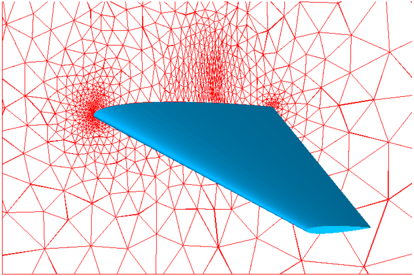
Test case: Wing shape optimization

- Minimize drag under lift constraint

$$\min \frac{C_d}{C_{d_0}} \quad \text{s.t.} \quad \frac{C_l}{C_{l_0}} \geq 0.999$$

- FFD parameterization, $n = 20$ design variables
- Particle swarm optimization: 120 particles

$M_\infty = 0.83,\ \alpha = 2^o$



(Piaggio Aero. Ind.)
Grid: 31124 nodes

Cost function

$$\mathcal{J} = \frac{C_d}{C_{d_0}} + 10^4 \max\left(0, 0.999 - \frac{C_l}{C_{l_0}}\right)$$

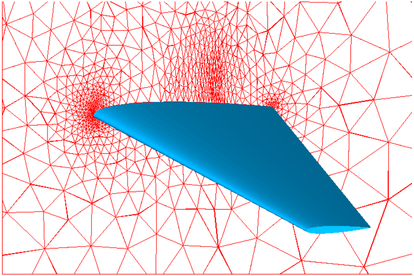
Test case: Wing shape optimization

- Minimize drag under lift constraint

$$\min \frac{C_d}{C_{d_0}} \quad \text{s.t.} \quad \frac{C_l}{C_{l_0}} \geq 0.999$$

- FFD parameterization, $n = 20$ design variables
- Particle swarm optimization: 120 particles

$M_\infty = 0.83$, $\alpha = 2^o$



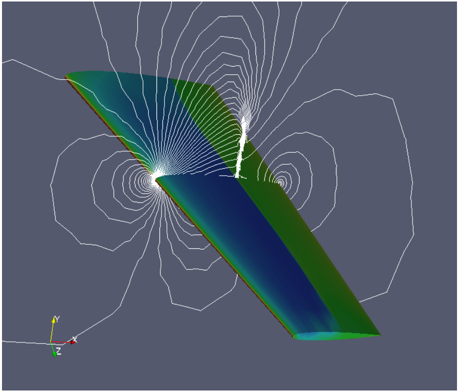
(Piaggio Aero. Ind.)
Grid: 31124 nodes

Cost function

$$\mathcal{J} = \frac{C_d}{C_{d_0}} + 10^4 \max\left(0, 0.999 - \frac{C_l}{C_{l_0}}\right)$$

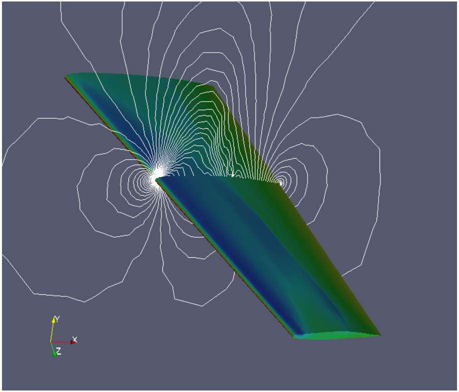
Wing optimization

Initial shape



Wing optimization

Optimized shape



PSO computational cost

- Slow convergence: O(100)-O(1000) iterations
- Require large swarm size: O(100) particles
- CFD is expensive: few minutes to hours
- Example: Transonic wing optimization (coarse CFD grid)

  (10 min/CFD) (120 CFD/pso iter) (200 pso iter) = 4000 hours

Surrogate Models

Metamodels

- Expensive PDE-based model

Shape parameters $\rightarrow$ Surface grid $\rightarrow$ Volume grid $\rightarrow$ CFD solution $\rightarrow$ J, C

- Replace costly model with cheap model:
  metamodel or surrogate model

Shape parameters $\rightarrow$ Surrogate model $\rightarrow$ $\tilde{J}, \tilde{C}$

- Approximation of cost function and constraint function(s)
  - Response surfaces (polynomial model)
  - Neural networks
  - Radial basis functions
  - Kriging/Gaussian Random Process models

Kriging I

Unknown function $f : \mathbb{R}^d \to \mathbb{R}$

Given the data as $F_N = \{f_1, f_2, \ldots, f_N\} \subset \mathbb{R}$ sampled at $X_N = \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^d$, infer the function value at a new point x_{N+1}.

Treat result of a computer simulation as a fictional gaussian process

F_N is assumed to be one sample of a multivariate Gaussian process with joint probability density

$$p(F_N) = \frac{\exp\left(-\frac{1}{2} F_N^\top C_N^{-1} F_N\right)}{\sqrt{(2\pi)^N \det(C_N)}} \tag{1}$$

where C_N is the $N \times N$ covariance matrix.

Kriging II

When adding a new point x_{N+1}, the resulting vector of function values F_{N+1} is assumed to be a realization of the $(N+1)$-variable Gaussian process with joint probability density

$$p(F_{N+1}) = \frac{\exp\left(-\frac{1}{2}F_{N+1}^\top C_{N+1}^{-1} F_{N+1}\right)}{\sqrt{(2\pi)^{N+1}\det(C_{N+1})}} \tag{2}$$

Using Baye's rule we can write the probability density for the unknown function value f_{N+1}, given the data (X_N, F_N) as

$$p(f_{N+1}|F_N) = \frac{p(F_{N+1})}{p(F_N)} = \frac{1}{Z}\exp\left[-\frac{(f_{N+1} - \hat{f}_{N+1})^2}{2\sigma^2_{f_{N+1}}}\right]$$

where

$$\underbrace{\hat{f}_{N+1} = k^\top C_N^{-1} F_N}_{\text{Inference}}, \qquad \underbrace{\sigma^2_{f_{N+1}} = \kappa - k^\top C_N^{-1} k}_{\text{Error indicator}} \tag{3}$$

Kriging III

Covariance matrix: Given in terms of a correlation function, $C_N = [C_{mn}]$,

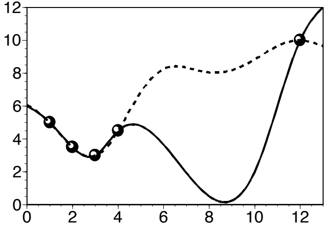
$$C_{mn} = \mathrm{corr}(f_m, f_n) = c(x_m, x_n)$$

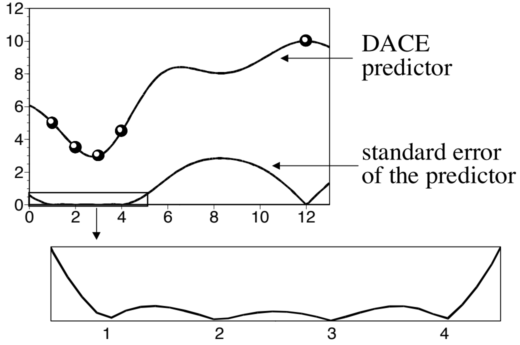
$$c(x,y) = \theta_1 \exp\left[-\frac{1}{2}\sum_{i=1}^{d}\frac{(x_i - y_i)^2}{{r_i}^2}\right] + \theta_2$$

Parameters $\Theta = (\theta_1, \theta_2, r_1, r_2, \ldots, r_d)$ determined to maximize the likelihood of known data

$$\max_{\Theta} \ \log(p(F_N))$$

Kriging: Illustration





Exploration verses exploitation

- Need a reasonably accurate surrogate model to represent global behaviour.
- But final goal is to find minimum of J, not to construct the most accurate model.
- In large dimensional spaces, it is quite impossible to construct a uniformly accurate model – *Curse of dimensionality*.
- Exploration refers to sampling all regions of design space.
  - Sample in regions where σ is large
- Exploitation refers to doing greater sampling around promising regions.
  - Sample in regions where $\tilde{J}$ is small
- First iteration: sample uniformly in design space using LHS to create a database of design variables and objective function values

Merit functions

- Merit function based on a statistical lower bound

$$\min_x J_\kappa(x) := \tilde{J}(x) - \kappa\sigma(x)$$

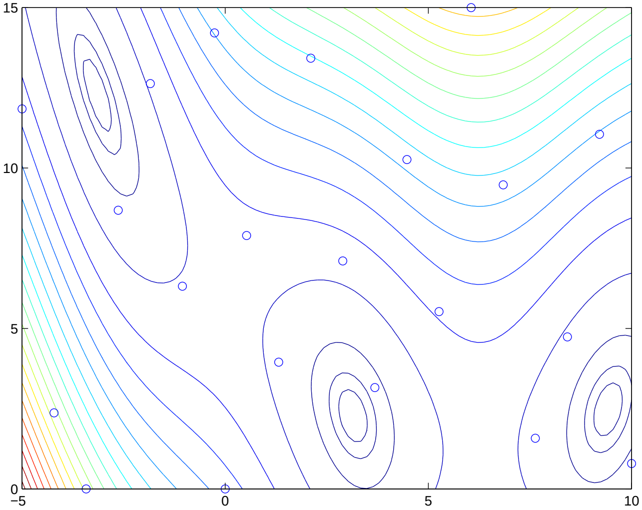
$$\begin{aligned} \kappa &= 0 & \Longrightarrow & \quad \text{exploitation} \\ \kappa &= \text{large} & \Longrightarrow & \quad \text{exploration} \end{aligned}$$

- Choose a set of $\kappa = 0, 1, 2, 3$
- Minimize four merit functions

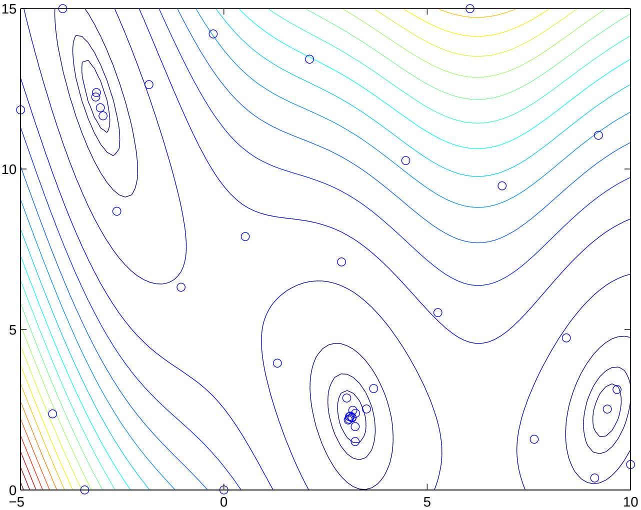
$$\begin{aligned} \min_x J_0(x) &\Longrightarrow x_0 \\ \min_x J_1(x) &\Longrightarrow x_1 \\ \min_x J_2(x) &\Longrightarrow x_2 \\ \min_x J_3(x) &\Longrightarrow x_3 \end{aligned}$$

- Evaluate x_0, x_1, x_2, x_3 on exact model (CFD)
- Add $J(x_0), J(x_1), J(x_2), J(x_3)$ to database
- Update metamodel $\hat{J}$

Minimization of 2-D Branin function: Initial database

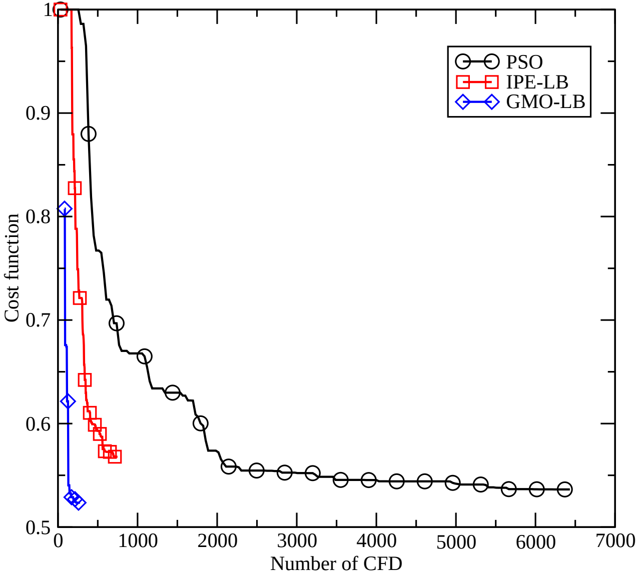


Minimization of 2-D Branin function: after 20 iter

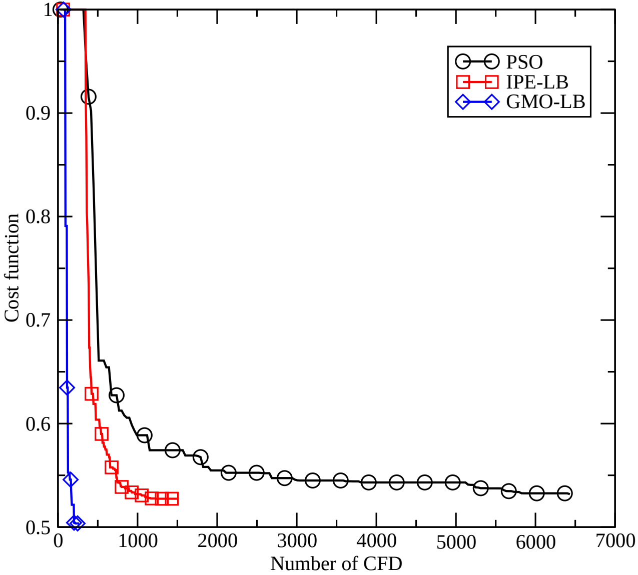


Transonic wing optimization

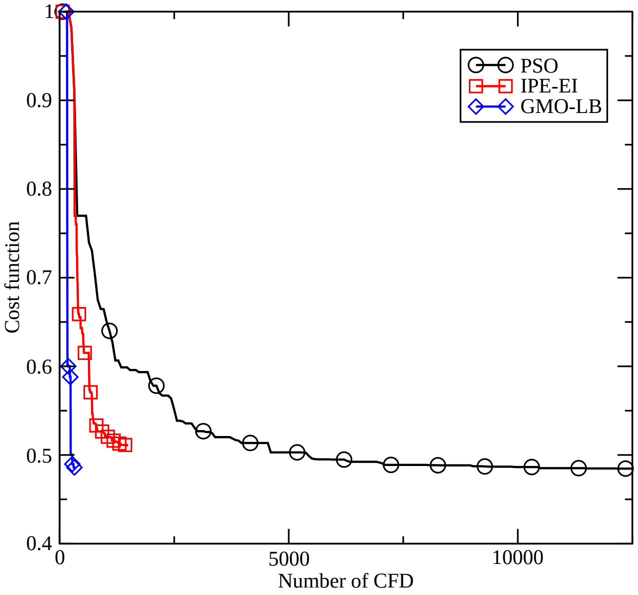
Transonic wing optimization: 8 design variables



Transonic wing optimization: 16 design variables



Transonic wing optimization: 32 design variables



Transonic wing optimization

No of DV	No. of CFD	Drag reduction
8, PSO	11136	0.526
8, PSO+Surrogate	181	0.523
16, PSO	9920	0.525
16, PSO+Surrogate	218	0.503
32, PSO	12736	0.483
32, PSO+Surrogate	305	0.485

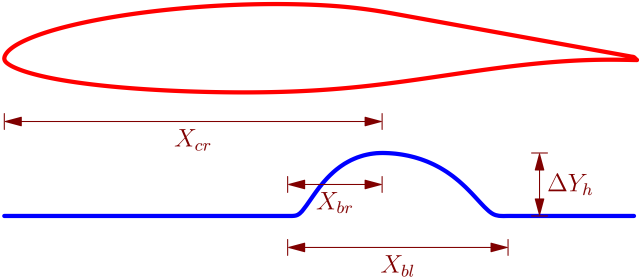
Transonic, turbulent airfoil optimization

TA5 test case

- Optimize RAE5243 airfoil to reduce drag under lift constraint

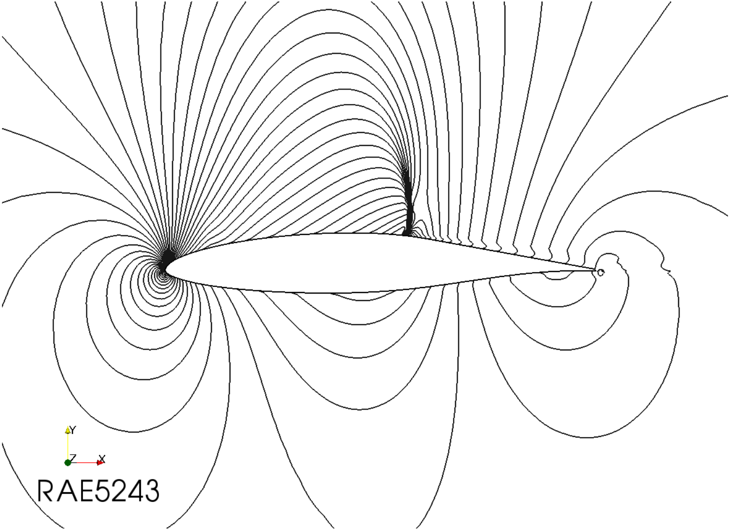
Mach	Re	C_l	Flow condition
0.68	19 million	0.82	Fully turbulent

- Modify shape of upper airfoil surface by adding a bump



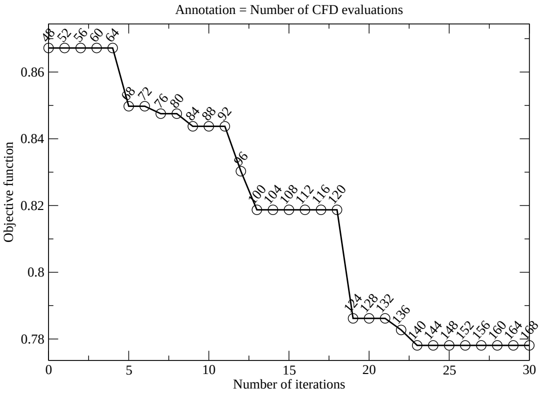
Reference solution: Pressure

$\alpha = 2.5$ deg.



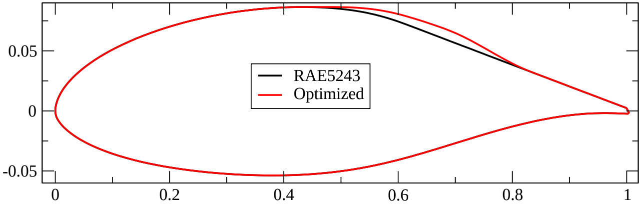
Optimization test

- 5 design variables
- Initial database of 48 using LHS
- 4 merit functions based on statistical lower bound with $\kappa = 0, 1, 2, 3$
- Gaussian process models
- Merit functions minimized using PSO

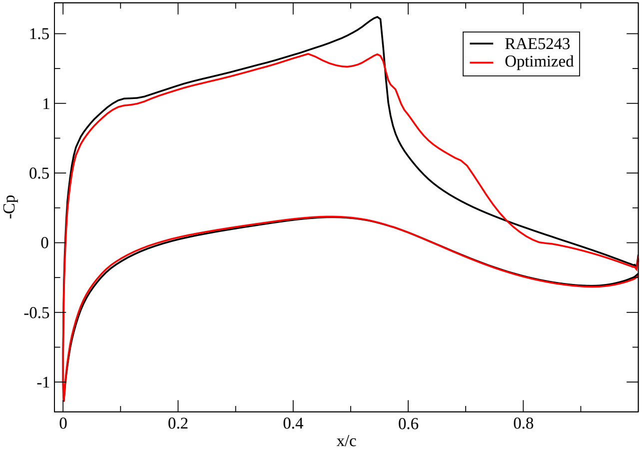


Shape parameters

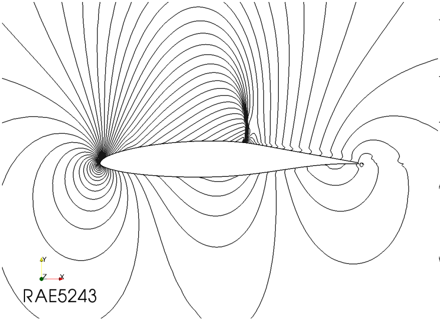
Case	X_{cr}	X_{bl}	X_{br}	$\Delta Y_h \times 10^{-3}$
Present	0.688	0.399	0.257	8.578
Qin et al.	0.597	0.313	0.206	5.900

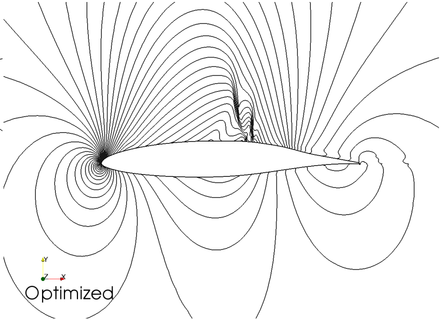


Case	C_d	ΔC_d	C_l	AOA
Present	0.01266	-22.2%	0.8204	2.19
Qin et al.	0.01326	-18.2%	0.82	-



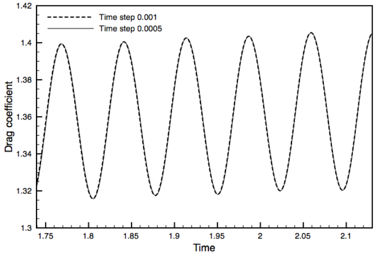
Pressure contours

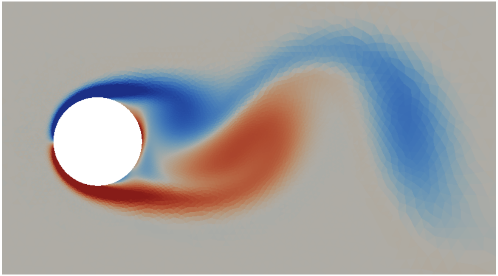




Optimal control parameters for cylinder flow

- Flow past 2-D cylinder at $Re = 200$
- Periodic vortex shedding, oscillatory forces



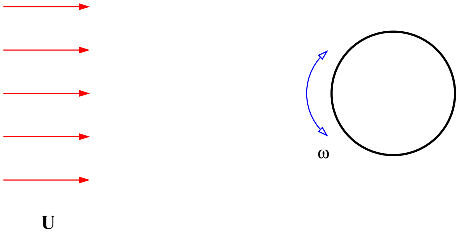


Ref.	St	Cd
Bergmann *et al.* (2005)	0.195	1.382
Braza *et al.* (1986)	0.200	1.400
Henderson (1997)	0.197	1.341
Homescu *et al.* (2002)	-	1.440
current study	0.198	1.370

Oscillating cylinder

Oscillating cylinder: Apply oscillating velocity boundary condition to cylinder wall

$$\omega(t) = A \sin(2\pi N t)$$



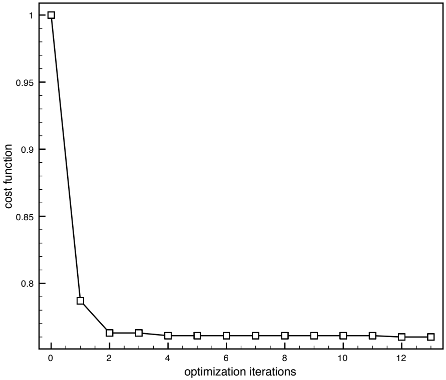
$$\text{Find } (A, N) \text{ to minimise } \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} C_D(t; A, N) dt$$

Non-dimensional variables and bounds:

$$A^* = \frac{AD}{U_\infty} \in [0, 5], \quad N^* = \frac{ND}{U_\infty} \in [0, 1]$$

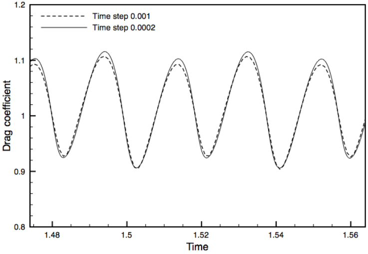
Optimization

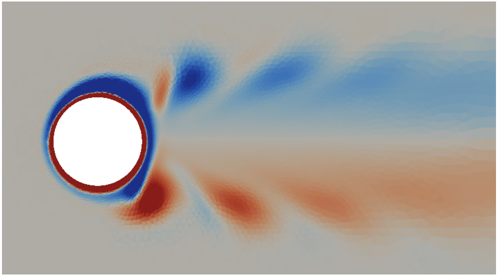
Initial sample of 16 using LHS



Good convergence in 3 iterations, 24 CFD solutions

Controlled case





Ref.	Method	$A^\star$	$N^\star$	ΔCd
Bergmann *et al.*(2004)	POD	2.2	0.53	25%
Bergmann *et al.*(2004)	POD-ROM	4.25	0.74	30%
He *et al.*(2002)	NS 2D	3.00	0.75	30%
current study	NS 3D	3.20	0.80	25%

Optimization problem

Optimization

$$\min_{x \in \mathbb{R}^d} \ \mathcal{J}(x, a_o)$$

$$\mathcal{C}(x, a_o) \le 0$$

Robust optimization

$$\min_{x \in \mathbb{R}^d} \begin{cases} \mu_J(x) = \int_{\Omega(A)} \mathcal{J}(x,a)\, \rho_A(a)\, da \\ \sigma_J^2(x) = \int_{\Omega(A)} [\mathcal{J}(x,a) - \mu_J]^2\, \rho_A(a)\, da \end{cases}$$

$$\mathrm{Prob}[\mathcal{C}(x, A) \le 0] \ge p$$

Optimization problem

Optimization

$$\min_{x \in \mathbb{R}^d} \; \mathcal{J}(x, a_o)$$

$$\mathcal{C}(x, a_o) \le 0$$

Robust optimization

$$\min_{x \in \mathbb{R}^d} \begin{cases} \mu_J(x) = \displaystyle\int_{\Omega(A)} \mathcal{J}(x, a)\, \rho_A(a)\, da \\ \sigma_J^2(x) = \displaystyle\int_{\Omega(A)} [\mathcal{J}(x, a) - \mu_J]^2\, \rho_A(a)\, da \end{cases}$$

$$\mathrm{Prob}[\mathcal{C}(x, A) \le 0] \ge p$$

Summary

- Numerical optimization using CFD
- Gradient-free + Metamodels + Free-form + CFD + MPI
- Major obstacle: CAD $\longrightarrow$ Grid
- Other shape parameterizations + Radial basis function deformation = free-form approach
- Develop numerical tools
- In future:
  - 3-D RANS-based optimization
  - Multi-point/Multi-objective optimization
  - Optimization under uncertainties/robust optimization