# A discrete shape optimization problem coming from yacht design.

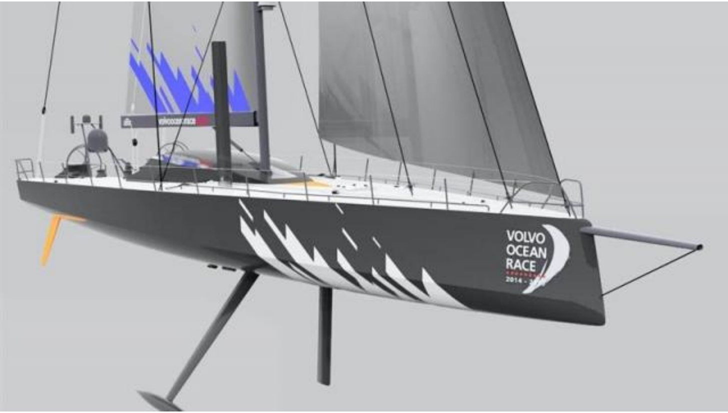
Rosa Donat, Sergio López-Ureña

Universitat de València.

23rd August 2017. Benasque.

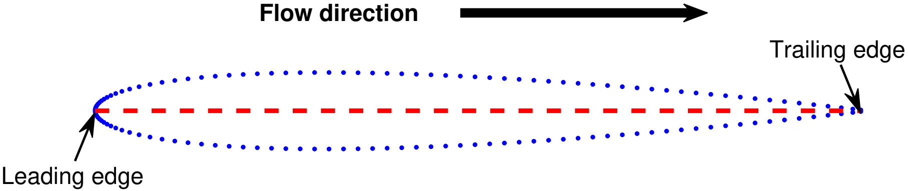
In yacht design we have several components which are designed based on sections. For example:

- Keel
- Rudder
- Daggerboard

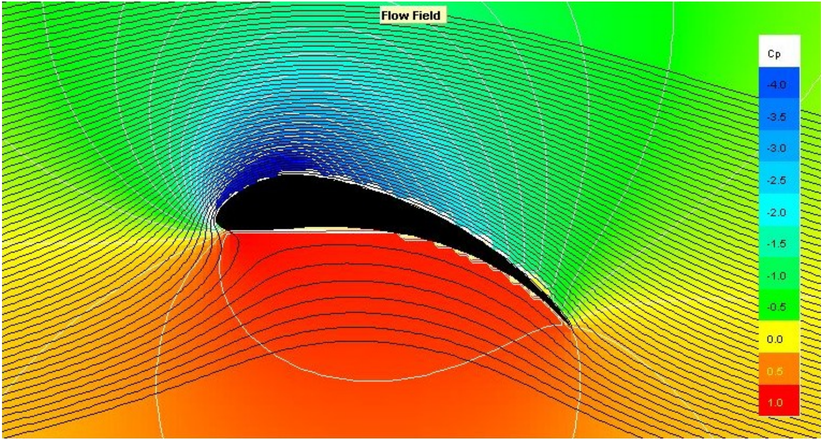


A 2D section: $\alpha(t) = (x(t), y(t)),\ t \in [0, 1]$,

$(x(0), y(0)) = (x(1), y(1))$ is the trailing edge.



- In general, the problem consists in optimizing the shape of some physical components while preserving some structural features.
- For keel sections, we are interested in minimizing the drag while preserving the lift.



# Discrete Framework

$$\alpha(t) \quad \equiv \quad (\alpha(t_i))_{i=1}^N \quad \to \quad D(\alpha) \text{ (Drag Coefficient)}$$

$$\varepsilon = (\varepsilon_i)_{i=1}^N, \quad \to \quad \alpha^\varepsilon := (\alpha(t_i) + \varepsilon_i)_{i=1}^N$$

**Minimization Problem:** $\boxed{\text{Find } \varepsilon_* \in \mathbb{R}^N : D(\alpha^{\varepsilon_*}) = \min_{\varepsilon \in \mathbb{R}^N} D(\alpha^\varepsilon)}$

Solution by iterative search algorithms. Initial guess: $\varepsilon_0 = 0 \equiv \alpha^{\varepsilon_0} = \alpha$

The process is likely to:

- depend on the initial guess being (sufficiently) close to a minimum.
- be (very) slow for $N$ large.

The cost may be reduced by using a multiscale strategy

# An academic example

Given the grid $(t_i)_{i=0}^{2^L} = (i2^{-L})_{i=0}^{2^L}$, compute the minimum of the functional

$$F(\alpha) := \|(\alpha_i - \cos(2\pi x_i))_i\|_2^2.$$

Minimization strategies:

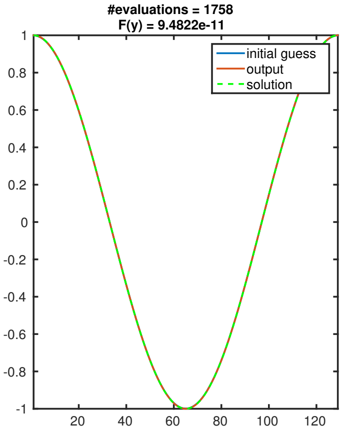
- Using the MATLAB `fminsearch` function directly.
- Using the MATLAB `fminsearch` function combined with a multi-scale strategy.

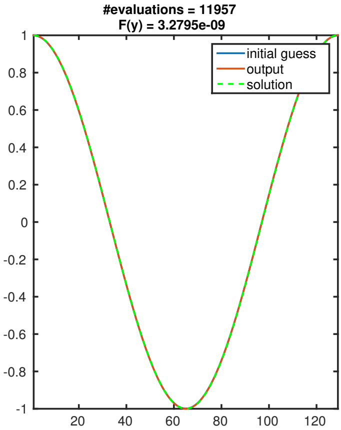
Initial guess $y_i := \lambda \cos(2\pi x_i)$, $\qquad L = 7, N = 2^L = 128.$

Stopping Criteria: difference between two consecutive iterates $< 10^{-4}$ + Cost function $< 10^{-8}$

Maximum # of evaluations: $10^5$.

**Cost $\equiv$ Number of function evaluations.**

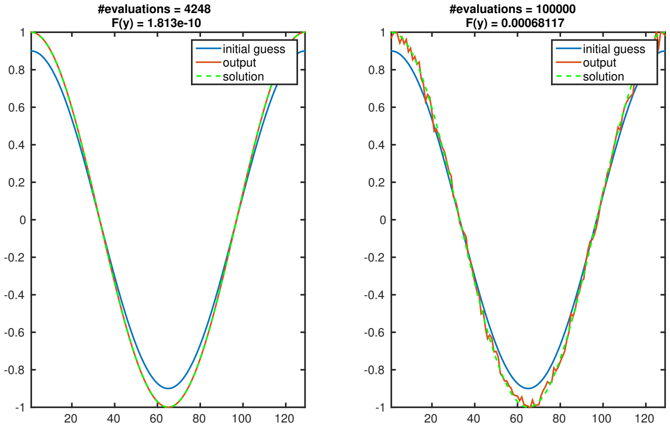



# funct. eval.

$\lambda = 0.999$

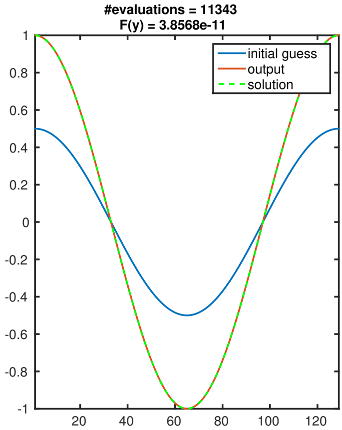
| MR | direct |
|---|---|
| 1758 | 11957 |

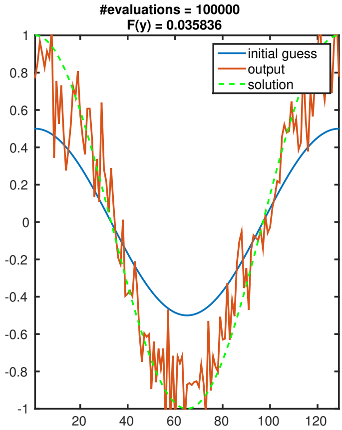



$\lambda = 0.9$

\# funct. eval.

| MR | direct |
|---|---|
| 4248 | $10^5$ |




$\lambda = 0.5$

# funct. eval.

| MR | direct |
|---|---|
| 11343 | $10^5$ |

# Discrete Multiresolution Framework

A multiresolution (MR) decomposition of a discrete data set is an equivalent representation that encodes the information as a coarse realization of the given data set plus a sequence of detail coefficients of ascending resolution.

$$\begin{array}{ccccccccc} \alpha^L & \rightarrow & \alpha^{L-1} & \rightarrow & \alpha^{L-2} & \rightarrow & \ldots \rightarrow \alpha^0 \\ & \searrow & d^{L-1} & \searrow & d^{L-2} & \searrow & \ldots \searrow d^0 \end{array}$$

$$\alpha^L \equiv M\alpha^L = (\alpha^0, d^0, d^2, \ldots, d^{L-1})$$

- levels of resolution: Hierarchy of nested computational meshes
- detail coefficients: difference in information between consecutive levels

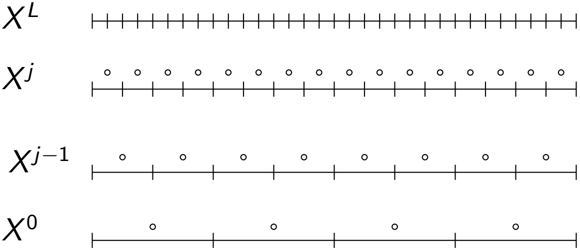
Frameworks for MR:

- Wavelets (Daubechies, Mayer, Mallat etc..)
- Lifting (Sweldens)
- Harten

# Harten's Framework for MR: Decimation and Prediction

$$\begin{array}{ccccccccc} \alpha^L & \rightarrow & \alpha^{L-1} & \rightarrow & \alpha^{L-2} & \rightarrow & \ldots \rightarrow \alpha^0 \\ & \searrow & d^{L-1} & \searrow & d^{L-2} & \searrow & \ldots \searrow d^0 \end{array}$$

$\alpha^j \leftrightarrow X^j$. Nested grid structure associated to MR ladder

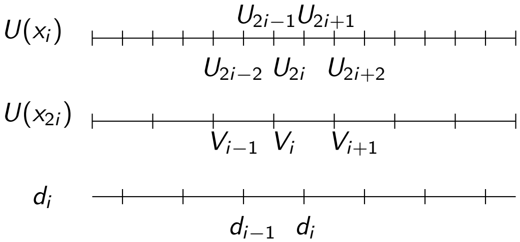



Decimation (from fine to coarse) $\alpha^{j-1} = D_j^{j-1} \alpha^j$

Prediction (from coarse to fine): $\tilde{\alpha}^j = P_{j-1}^j \alpha^{j-1}$

**Consistency:** $D_j^{j-1} P_{j-1}^j = I \quad I - P_{j-1}^j D_j^{j-1} \rightarrow d^j$

# An Example: Harten's Interpolatory MR framework

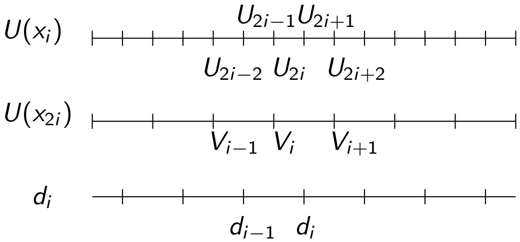



Decimation $\equiv$ Restriction to even values

Prediction $\equiv$ Via an interpolatory reconstruction $\mathcal{I}(x, \cdot)$

$$\left\{ \begin{array}{ccc} V_i & = & U_{2i} \\ d_i & = & U_{2i+1} - \mathcal{I}(x_{2i+1}, V) \end{array} \right\} \leftrightarrow \left\{ \begin{array}{ccc} U_{2i} & = & V_i \\ U_{2i+1} & = & \mathcal{I}(x_{2i+1}, V) + d_i \end{array} \right\}$$

# An Example: Harten's Interpolatory MR framework




**Prediction**: $\tilde{U}_i = \mathcal{I}(x_i, V)$.

$\mathcal{I}(x, \cdot)$ Data-independent $\rightarrow$ Linear transform.

**Consistency with Decimation by restricion**: $\mathcal{I}(x, \cdot)$ Interpolatory

$$\tilde{U}_{2i} = \mathcal{I}(x_{2i}, V) = V_i = U_{2i}$$

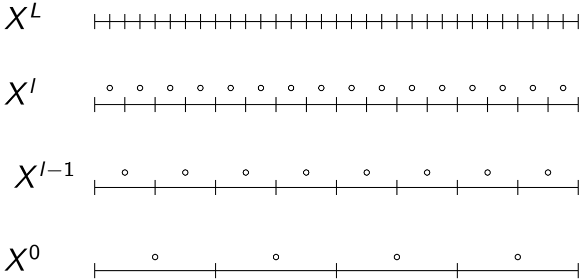
Prediction $\equiv$ Subdivision Refinement scheme $S$

# MR transformation:

Start at finest level $X^L$ and repeat process for $l = L, \dots, 1$.

$$u^L \;\Leftrightarrow\; \{u^{L-1}, d^{L-1}\} \quad \cdots \Leftrightarrow \cdots \; \{u^0; d^0; d^1; \cdots d^{L-1}\} = Mu^L$$

$$\begin{array}{ccccccc} u^L & \rightarrow & u^{L-1} & \rightarrow & u^{L-2} & \rightarrow & \dots \rightarrow u^0 \\ & \searrow & d^{L-1} & \searrow & d^{L-2} & \searrow & \dots \searrow d^0 \end{array}$$



# A Two-Scale 'parameter-reduction' approach

Initial curve: $U^0 = (U_i)_{i=1}^N$, $U^\varepsilon := (U_i + \varepsilon_i)_{i=1}^N$, $\varepsilon \in \mathbb{R}^N$,

**Minimization problem:** Find $\boxed{\varepsilon_* = \text{argmin}_{\varepsilon \in \mathbb{R}^N} D(U^\varepsilon)}$

$$MU = (V, d) \quad \overset{M,\text{MR Transform}}{\leftarrow - - \rightarrow} \quad U = M^{-1}(V, d) \quad V, d \in \mathbb{R}^{N/2}$$

**Solve two minimization problems**:

- Consider perturbations $V + \epsilon^1$, $\varepsilon^1 \in \mathbb{R}^{N/2}$.

  $M$ linear $\rightarrow$ $M^{-1}(V + \varepsilon^1, d) = U + M^{-1}(\varepsilon^1, \vec{0}) = U + S\varepsilon^1 = U^{S\varepsilon^1}$

  Find $\boxed{\varepsilon_*^1 = \text{argmin}_{\varepsilon^1 \in \mathbb{R}^{N/2}} D(U^{S\varepsilon^1})}$ inital guess $\varepsilon_0^1 = 0$, $U^0 = U$.

- Find $\boxed{\varepsilon_* = \text{argmin}_{\varepsilon \in \mathbb{R}^N} D(U^\varepsilon)}$ initial guess $\begin{cases} \varepsilon_0 = & S\varepsilon_*^1 \\ \bar{U}^0 = & U + S\varepsilon_*^1 \end{cases}$

# A Multi-scale 'parameter-reduction' approach

Initial data: $\alpha^L = (\alpha_i)_{i=1}^{N_L}$, $N_L = 2^L N_0$, $\quad (N_j = 2^j N_0,\ j = 1, \dots, L)$

$$
\begin{array}{ccccccccc}
\color{blue}{\alpha^L} & \rightarrow & \alpha^{L-1} & \rightarrow & \alpha^{L-2} & \rightarrow & \dots \rightarrow & & \color{red}{\alpha^0} \\
 & \searrow & \color{red}{d^{L-1}} & \searrow & \color{red}{d^{L-2}} & \searrow & \dots & \searrow & \color{red}{d^0} \\
 & & \mathbb{R}^{N_{L-1}} & & \mathbb{R}^{N_{L-2}} & & \dots & & \mathbb{R}^{N_0}
\end{array}
$$

- Find $\varepsilon_*^0 = \text{argmin}_{\varepsilon^0 \in \mathbb{R}^{N_0}} D(\alpha^L + S^L \varepsilon^0)$, Init. guess: $\epsilon_0^0 = \vec{0}$, $(\alpha^L)$
  Best 0th-level approximation $\alpha^{L,0} := \alpha^L + S^L \varepsilon_*^0$
- Find $\epsilon_*^1 = \text{argmin}_{\varepsilon^1 \in \mathbb{R}^{N_1}} D(\alpha^L + S^{L-1} \varepsilon^1)$, Init. guess: $\varepsilon_0^1 = S\varepsilon_*^0, (\alpha^{L,0})$
  Best 1st-level approximation $\alpha^{L,1} := \alpha^L + S^{L-1} \varepsilon_*^1$
- ....
- Find $\varepsilon_*^L = \text{argmin}_{\varepsilon^L \in \mathbb{R}^{N_L}} D(\alpha^L + \varepsilon^L)$, Init. guess: $\varepsilon_0^L = S\varepsilon_*^{L-1}(\alpha^{L,L-1})$
  Best $L$th-level approximation $\alpha^{L,L} := \alpha^L + \varepsilon_*^L$

# Academic example

Given the grid $(t_i)_{i=0}^{N_L} = (i2^{-L}h_0)_{i=0}^{N_L}$, compute the minimum of the functional

$$D(\alpha) := \|(\alpha_i - \cos(2\pi t_i))_i\|_2^2.$$

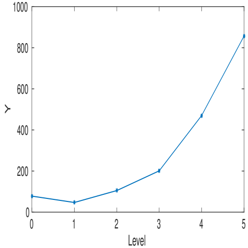
Initial curve: $\alpha = (\lambda \cos(2\pi t_i))_{i=1}^{N_L}$

Minimization strategies:

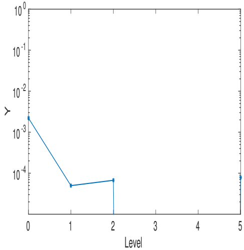
- Using the MATLAB `fminsearch` function directly.
  Starting from $\varepsilon = 0$, Find $\boxed{\varepsilon_* = \text{ argmin }_{\varepsilon \in \mathbb{R}^{N_L}} F(\alpha + \varepsilon)}$
- Using the MATLAB `fminsearch` function combined with the multi-scale strategy that uses a Prediction scheme based on B-splines

# Academic Example: $\lambda = 0.999$, MR-F-eval 1758

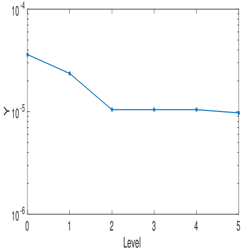
\# of F eval.



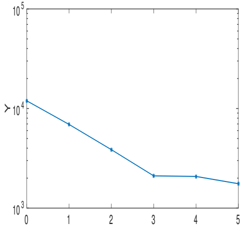

$||\varepsilon_0^j - \varepsilon_*^j||_\infty$



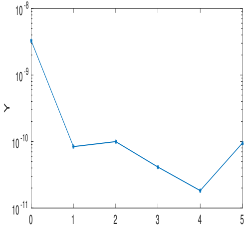

$||\alpha^{L,j} - \text{target}||_2$




\# of F-eval vs. L

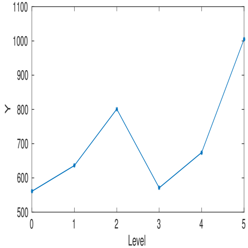



$D(\alpha)$ vs. L

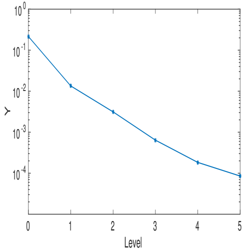


# Academic Example: $\lambda = 0.9$, MR-F-eval 4248

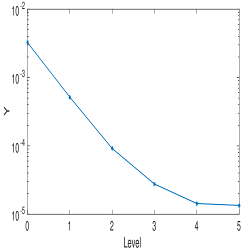
\# of F eval.



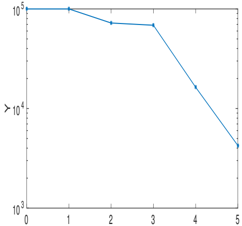

$||\varepsilon_0^j - \varepsilon_*^j||_\infty$



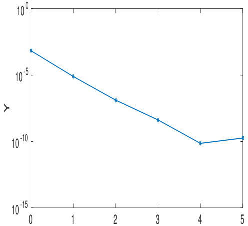

$||\alpha^{L,j} - \text{target}||$




\# of F-eval vs. L

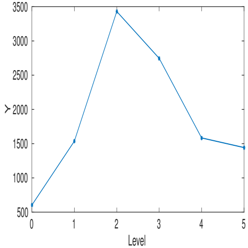



$D(\alpha)$ vs. L

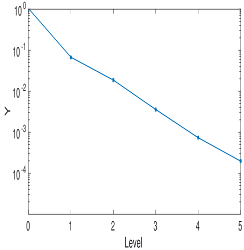


# Academic Example: $\lambda = 0.5$, MR-F-eval 11343

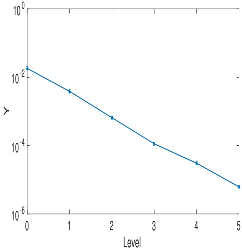
$\#$ of F eval.



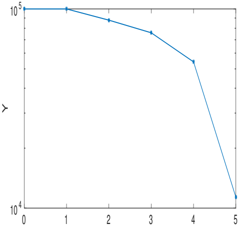

$||\varepsilon_0^j - \varepsilon_*^j||_\infty$



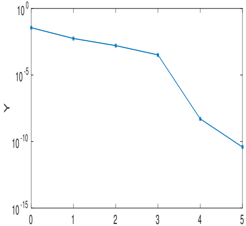

$||\alpha^{L,j} - \text{target}||$



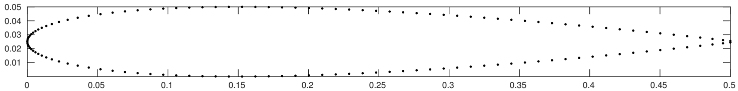

$\#$ of F-eval vs. L




$D(\alpha)$ vs. L



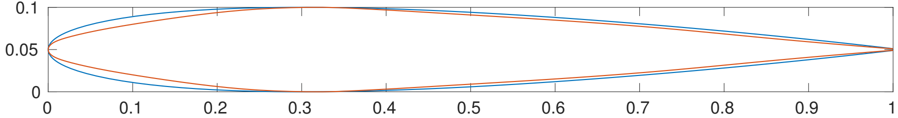
Closed Curve: NACA-profile, $\alpha = (x, y)$, $N = 128$ points



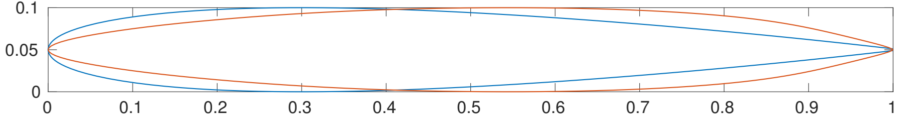

Required: Minimize the Drag Coefficient $D(\alpha)$ (computed with *Xfoil*) maintaining the *chord length and thickness*.
Have used: MR-algorithm with $L = 7$, $N_0 = 2^2$ on both components.

Reynolds number $Re = 10^6$. Initial Drag: 9.21e-3

**fminsearch** Final Drag: 7.74e-3




**patternsearch** Final Drag: 4.47e-3



# A discrete shape optimization problem coming from yacht design.

Rosa Donat, Sergio López-Ureña

Universitat de València.

23rd August 2017. Benasque.