Free-form deformation for multi-level 3D parallel optimization in aerodynamics

Michele Andreoli, Ales Janka, J-A Desideri, T. Nguyen

INRIA

INRIA Rhône-Alpes

Projet OPALE

International Conference Parallel CFD 2004
Las Palmas de Gran Canaria

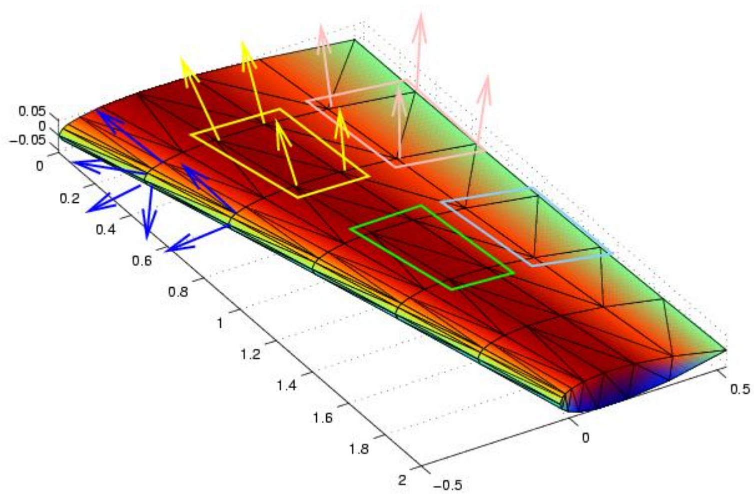
Objectives and possibilities

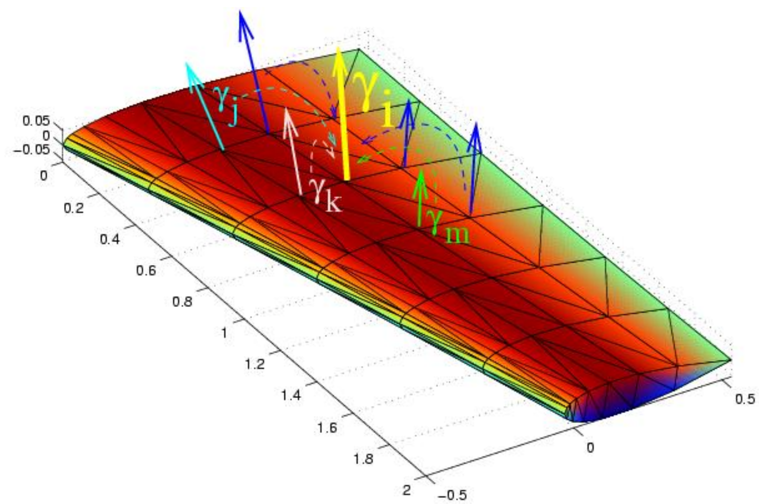
- Parameterization of (complex) shapes in 3D (airfoils or complete aircrafts) for shape optimization purposes
- Multilevel-approach: progressive enriching of the search space
- Reduction of shape parameters for genetic algorithms
- Inherent regularity properties of shape deformations
- Adaptability (future developments)

Possibilities in 2D/3D:

- CAD-free parametrization (based on finite-element mesh)
- CAD-like parametrization (modelization of surface by splines)
- Free-form deformation

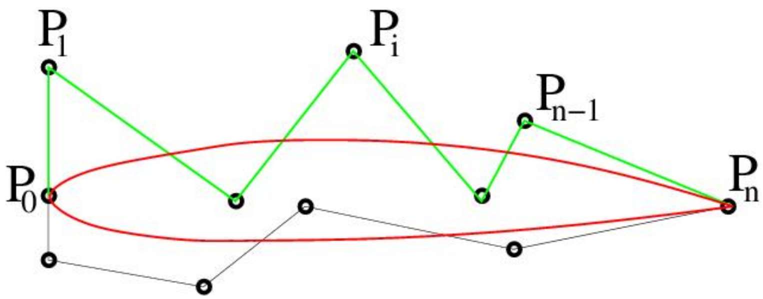
CAD-free parametrizations

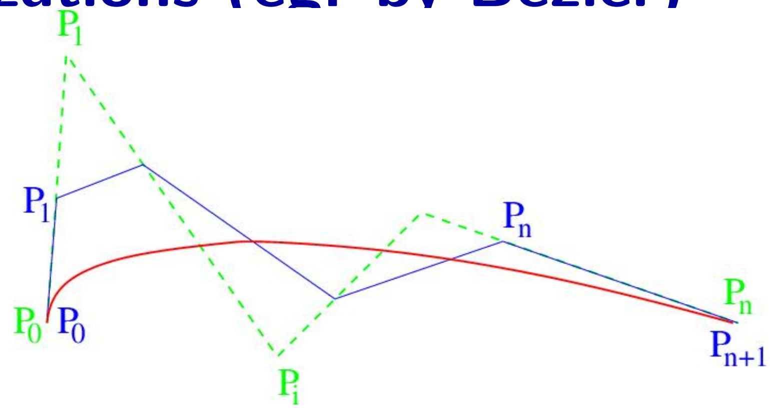




- Based only on 3D mesh of the skin
- Hierarchy of parametrizations
- Two important issues: smoothness of the shape deformations, local support of the basis (do we need it?)
- Versatility for complex 3D objects, but some problems with definition of normals to a discrete surface
- On finest level, too much parameters (as many as mesh nodes on the skin)

CAD parametrizations (eg. by Bezier)

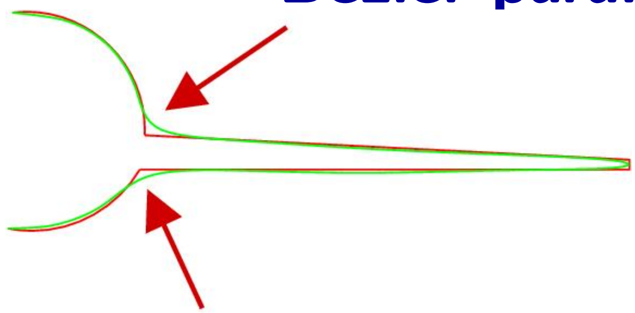


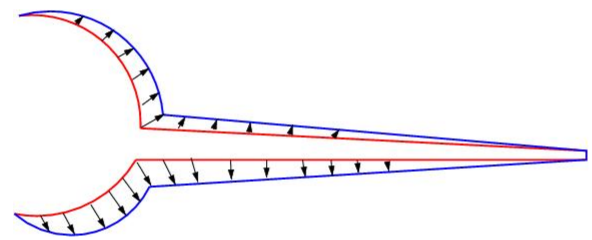


- Bezier curve: $\vec{x}(t) = \sum_{k=0}^{n} B_n^k(t)\vec{p}_k$
- Bezier patch: $\vec{x}(s,t) = \sum_{i=0}^{n_i} \sum_{k=0}^{n_k} B_{n_i}^i(s) B_{n_k}^k(t)\vec{p}_{ik}$
- Nice properties:
  - Assures smoothness
  - Elevation of degree (both curves and patches):

$$\vec{x}(t) = \sum_{k=0}^{n+1} B_{n+1}^k(t)\vec{P}_k \quad , \quad \vec{P}_k = \frac{k}{n+1}\vec{p}_{k-1} + (1 - \frac{k}{n+1})\vec{p}_k$$

Bezier parametrization: inconveniences

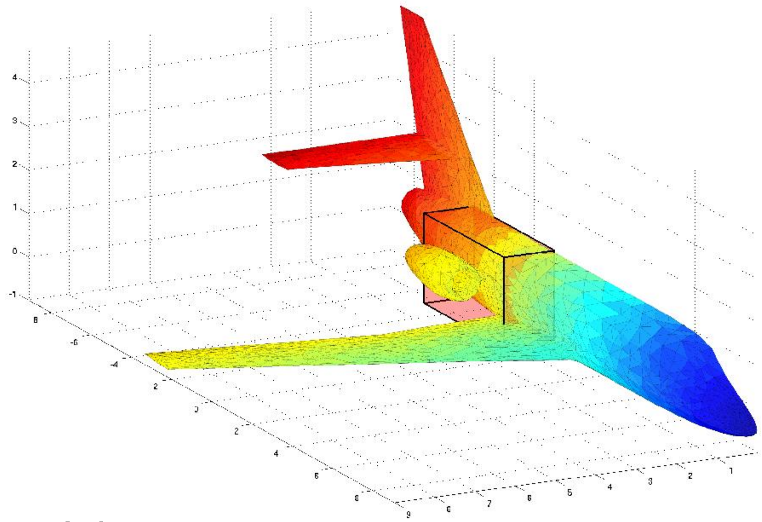




- Standard Bezier describes only smooth objects
- For non-smooth objects needs either
  - very high order of one Bezier curves (with danger of oscillations), or
  - two curves/patches joined by some condition on smoothness C^0, C^1: complicated to handle, looses degree-elevation property
- Do we need to describe the optimized shape? Or do we need to describe just its deformation?
- Bezier "delta" formulation $\vec{x}(t) = \vec{x}^{\text{init}} + \sum_{k=0}^{n} B_n^k(t)\vec{\delta}p_k$
  - Coordinates of mesh node i: $\vec{x}_i = \vec{x}_i^{\text{init}} + \sum_{k=0}^{n} B_n^k(t_i)\vec{\delta}p_k$
  - Control points $\vec{p}_k$ lose its meaning of "position"
  - The parametrization tasks resumes to assignment of t_i for $\vec{x}_i$

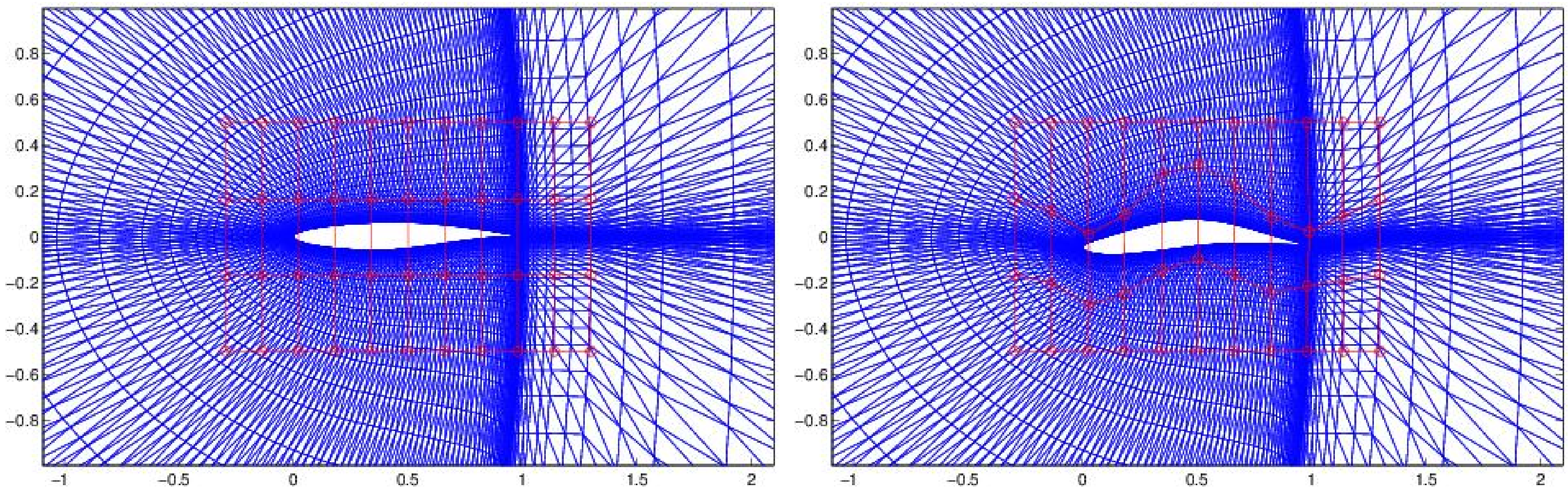
Free-form deformation

- Generalization of Bezier patches to 3D surfaces is difficult (intersecting/merging patches)
- How about parametrizing the whole space? Free-form deformation: Sederberg and Parry 1985 (computer graphics), Samareh (CFD).



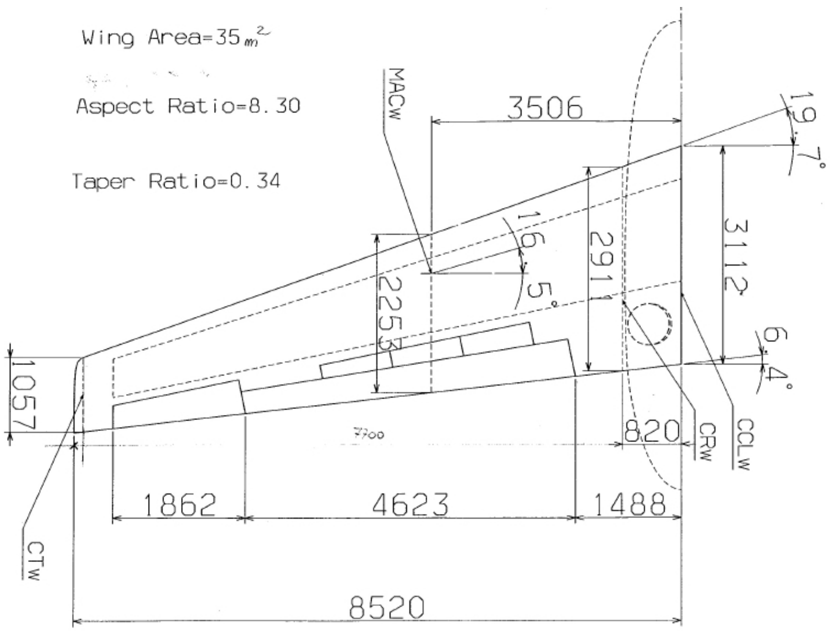
- Choose a box with the optimized form (part of the form) inside
- Parameterize the deformation of the volume inside the box by some 3D parameterization technique (tensorial B-splines, tensorial Bezier, . . .).
- The deformations of the parameterized shape is the trace of the 3D deformation inside the box on the mesh skin.
- Advantages:
  - Avoids the complexity of 3D object
  - Deforms space, ie. can handle deformation of computational mesh in a very cheap way

Free-form deformation and Bezier parameterization



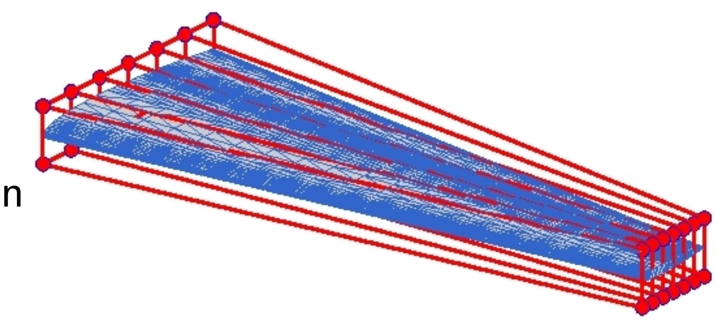
- in 2D: $\vec{x}_m = \vec{x}_m^{\text{init}} + \sum_{i=0}^{n_i} \sum_{k=0}^{n_k} B_{n_i}^{i}(s_m) B_{n_k}^{k}(t_m) \vec{\delta} p_{ik}$
- in 3D: $\vec{x}_m = \vec{x}_m^{\text{init}} + \sum_{i=0}^{n_i} \sum_{j=0}^{n_j} \sum_{k=0}^{n_k} B_{n_i}^{i}(r_m) B_{n_j}^{j}(s_m) B_{n_k}^{k}(t_m) \vec{\delta} p_{ijk}$
- Advantages:
  - Refinement of research space by degree elevation
  - Differentiability of the parametrization formula

Transonic 3D test case



- section profile: NACA0012
- transonic Euler model
  - Angle of incidence 2^o
  - Free stream Mach number 0.83
- Lift-drag minimization
  - minimize drag
  - aerodynamic constraint: keep lift up to 0.1%
  - geometric constraint: leading and trailing edge fixed, "asymptotical thickness" at leading and trailing edge preserved

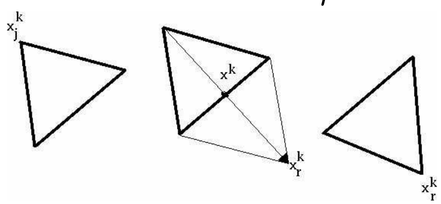
- cost: $J = \dfrac{\mathrm{Cd}}{\mathrm{Cd}_0} + 10^4 \cdot \max(0, 0.999 - \dfrac{\mathrm{Cl}}{\mathrm{Cl}_0})$
- corner Bezier control points are fixed, others can move in thickness-wise direction



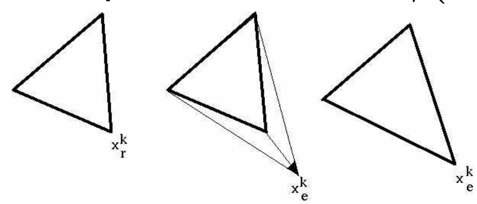
Simplex algorithm (Nelder-Mead)

- Initialize $n+1$ vertices of simplex ($n\ldots$ number of parameters)
- Calculate the centroid $\bar{x}^k$ of the other points
- Identify the worst vertex x_j^k in k-th iteration and replace it with better vertex x_r^k:

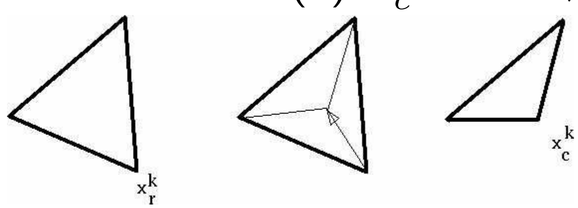
- Reflection: $x_r^k = \bar{x}^k + \alpha(\bar{x}^k - x_j^k)$



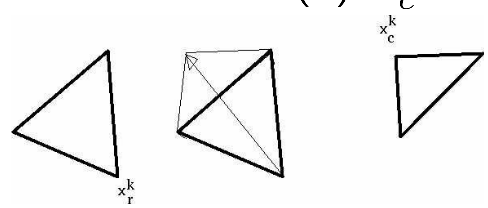
- Expansion: $\bar{x}^k + \gamma(x_r^k - \bar{x}^k)$



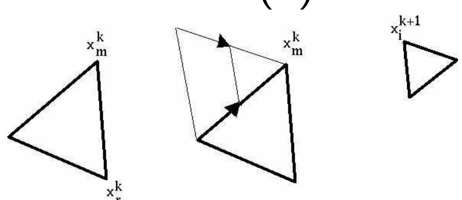
- Contraction (1) $x_c^k = \bar{x}^k + \beta(x_r^k - \bar{x}^k)$



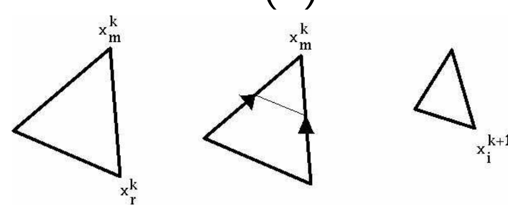
- Contraction (2) $x_c^k = \bar{x}^k + \beta(x_j^k - \bar{x}^k)$



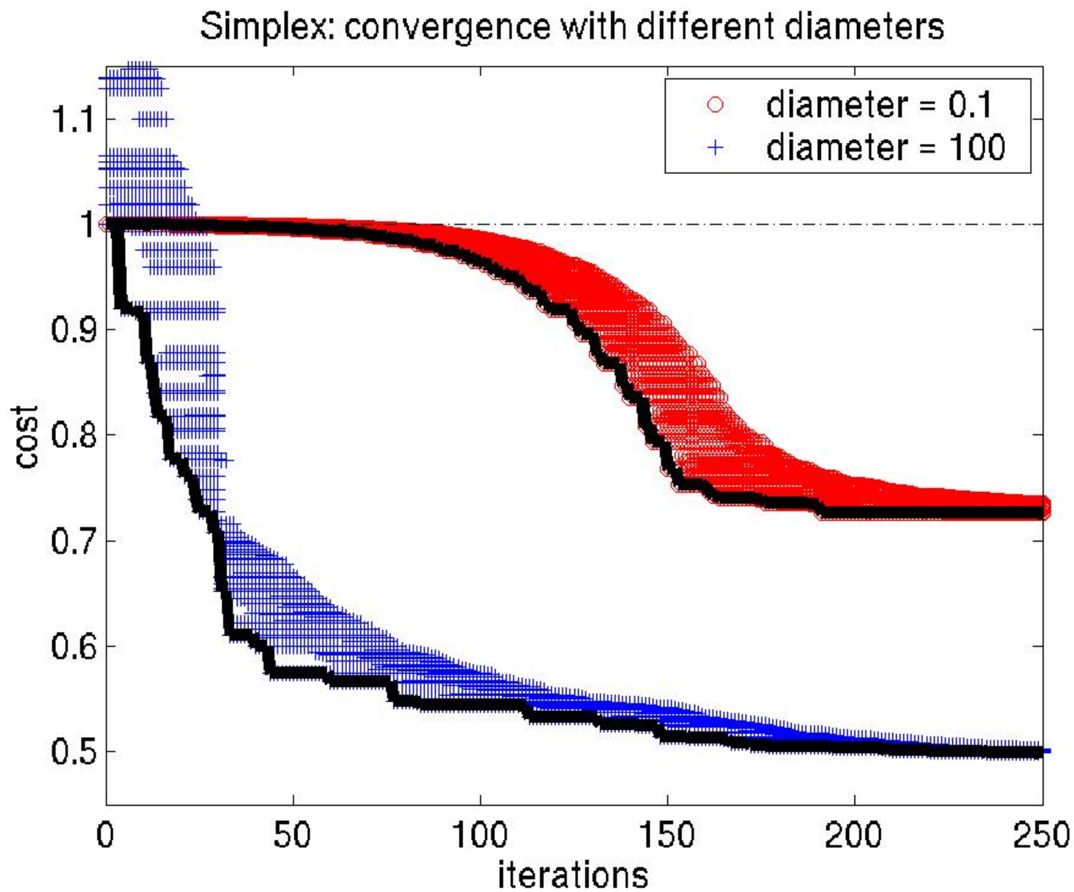
- Reduction (1): around the best x_m^k



- Reduction (2): around the best x_m^k



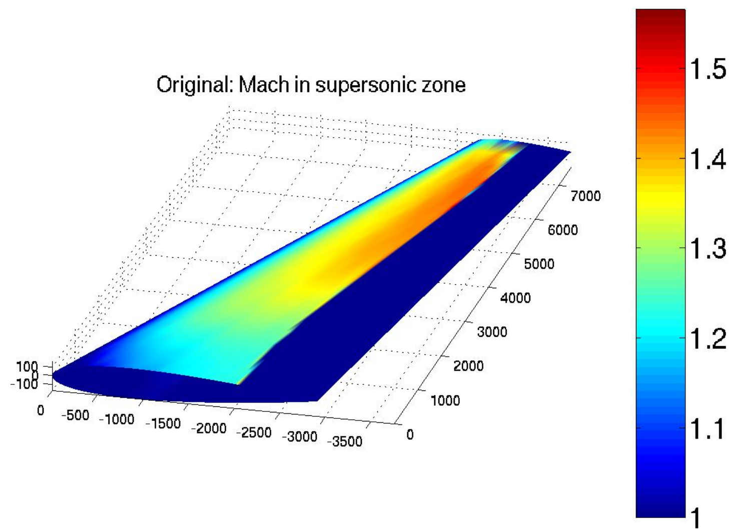
Simplex algorithm: diameter



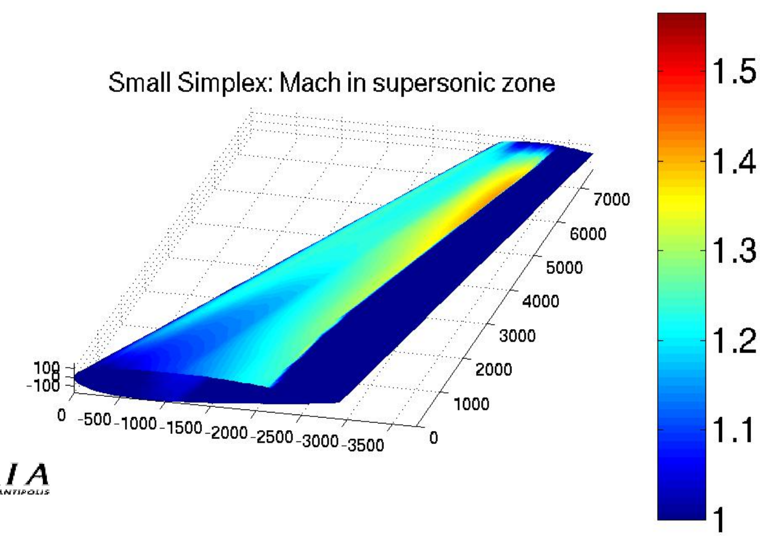
2 simplex optimizations with different simplex diameters:

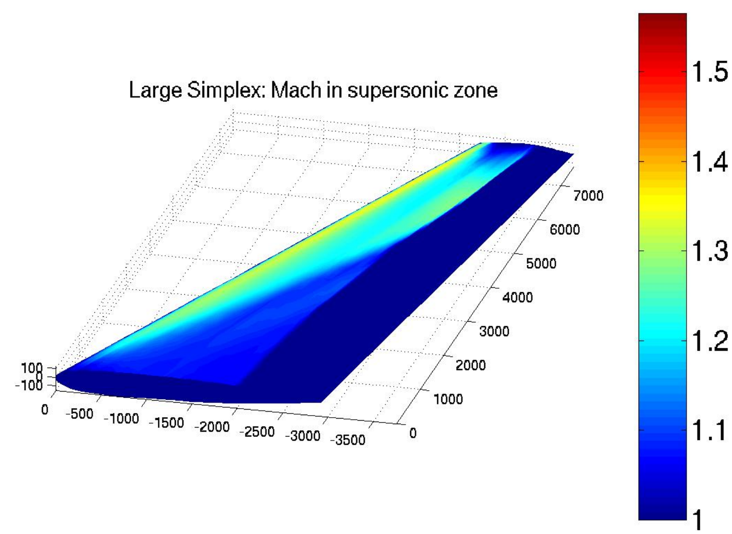
- small diameter (0.1) for 250 iterations
- large diameter (100) for 250 iterations

	small	large
cost	0.726	0.492
gain	27.4%	50.8%

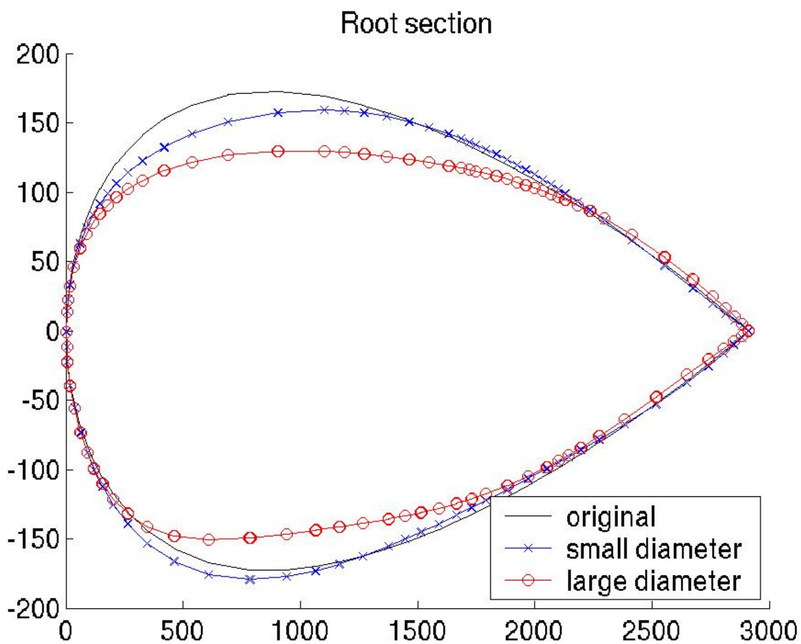


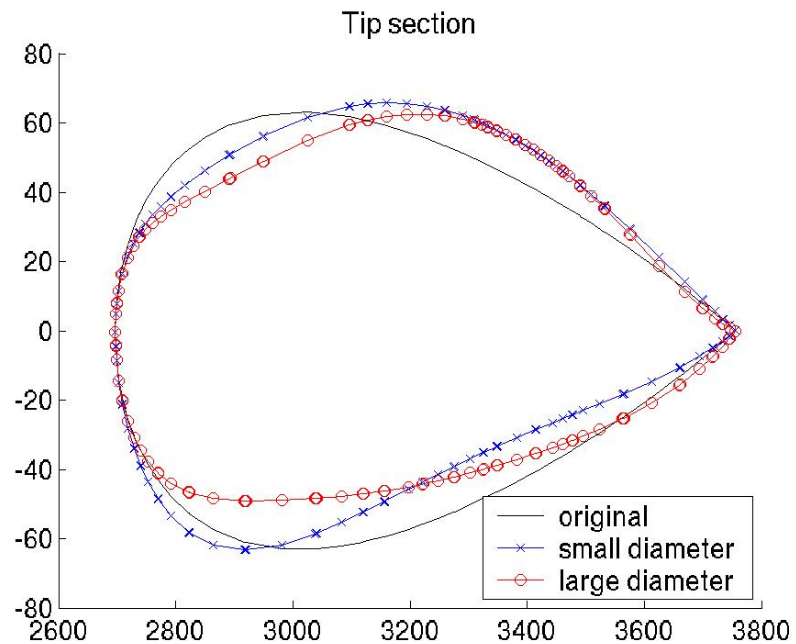
	C_L	C_D	Gain
orig	0.319200	0.026353	
small	0.318938	0.019139	27.4%
large	0.318885	0.012980	50.8%



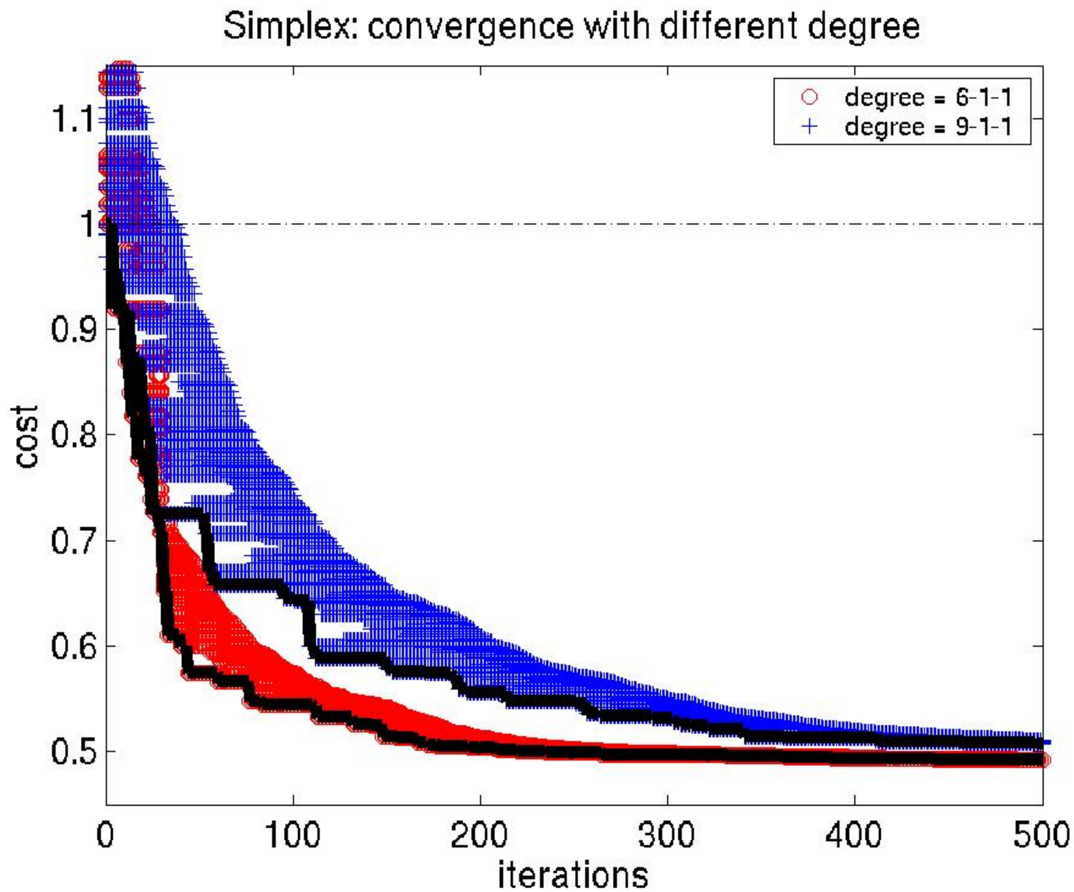


Simplex diameter: root and tip sections





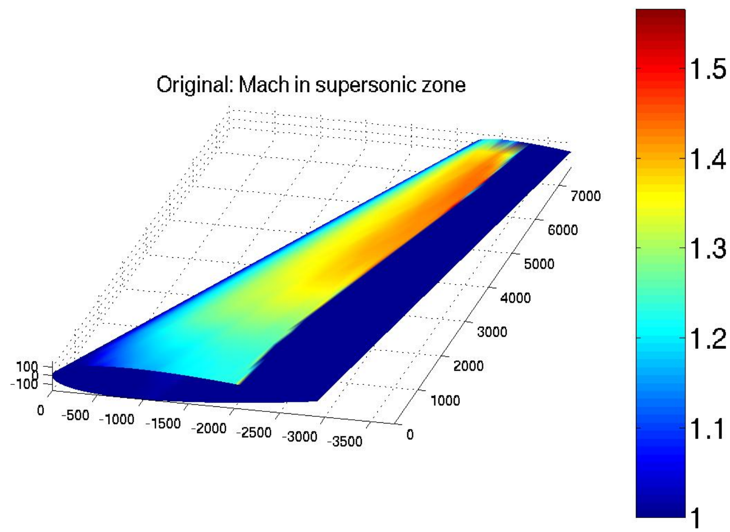
Simplex algorithm: different degree



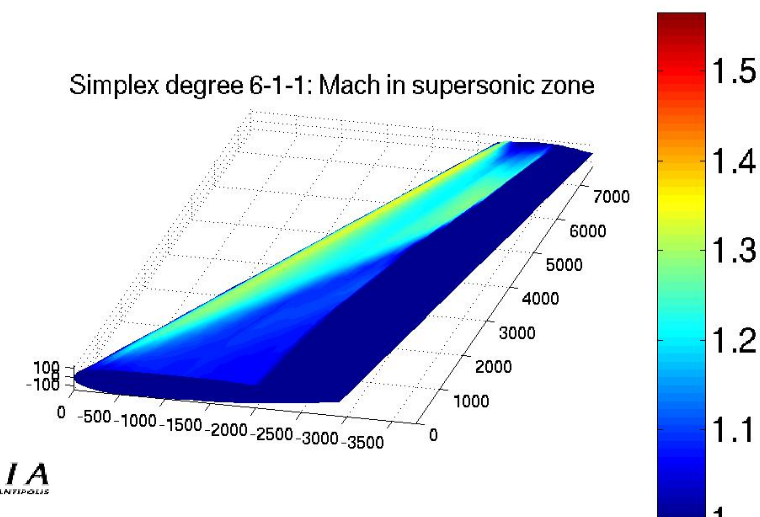
2 simplex optimizations with Bézier parameterization in chord-wise direction of different degree:

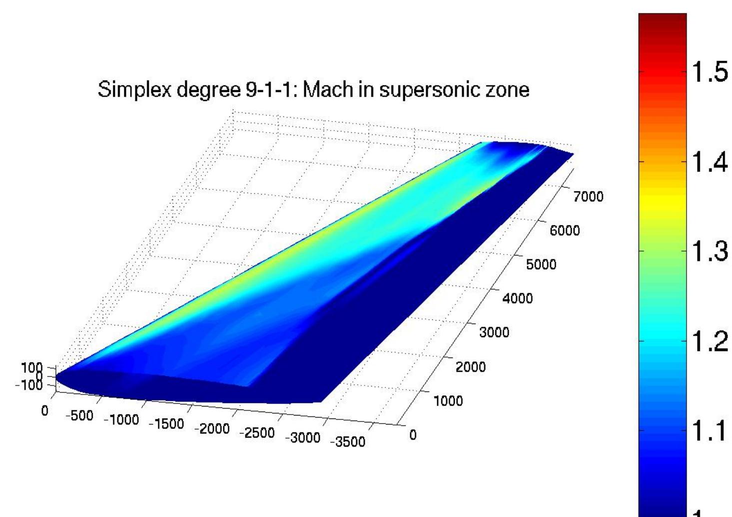
- degree 6-1-1 for 500 iterations
- degree 9-1-1 for 500 iterations

	6-1-1	9-1-1
cost	0.492	0.496
gain	50.8%	50.4%

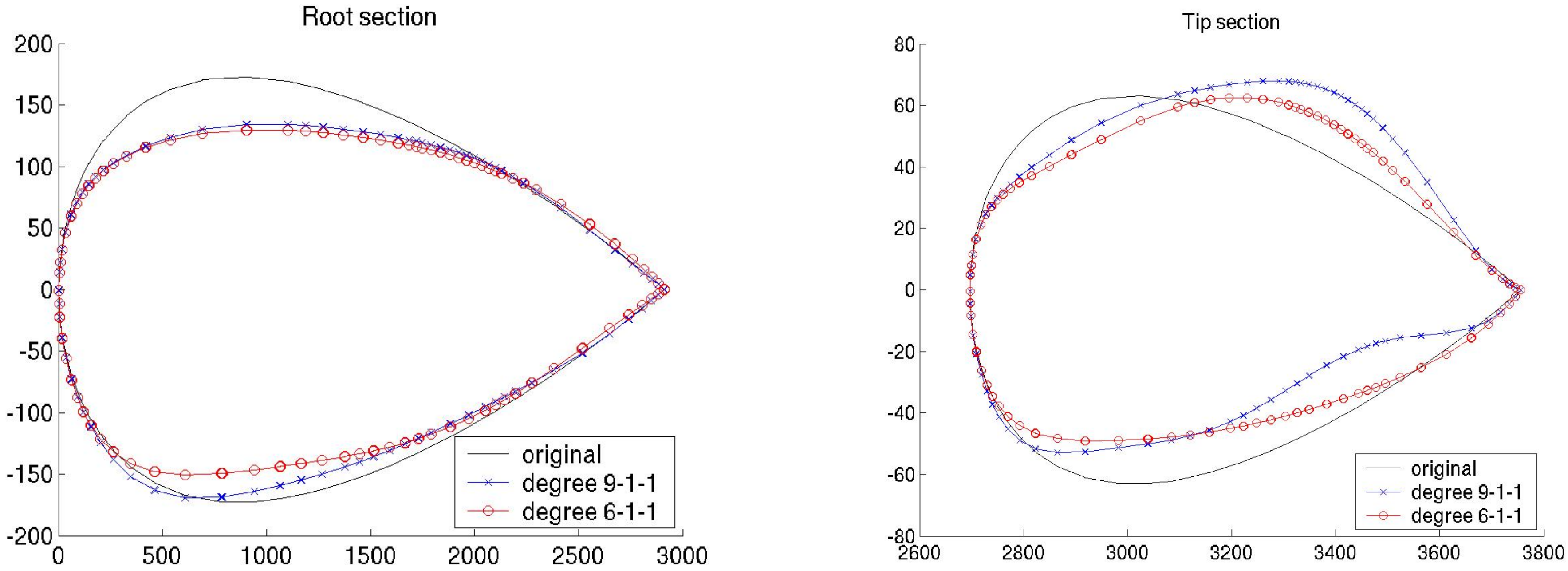


	C_L	C_D	Gain
orig	0.319200	0.026353	
6-1-1	0.318885	0.012980	50.8%
9-1-1	0.319079	0.013075	50.4%

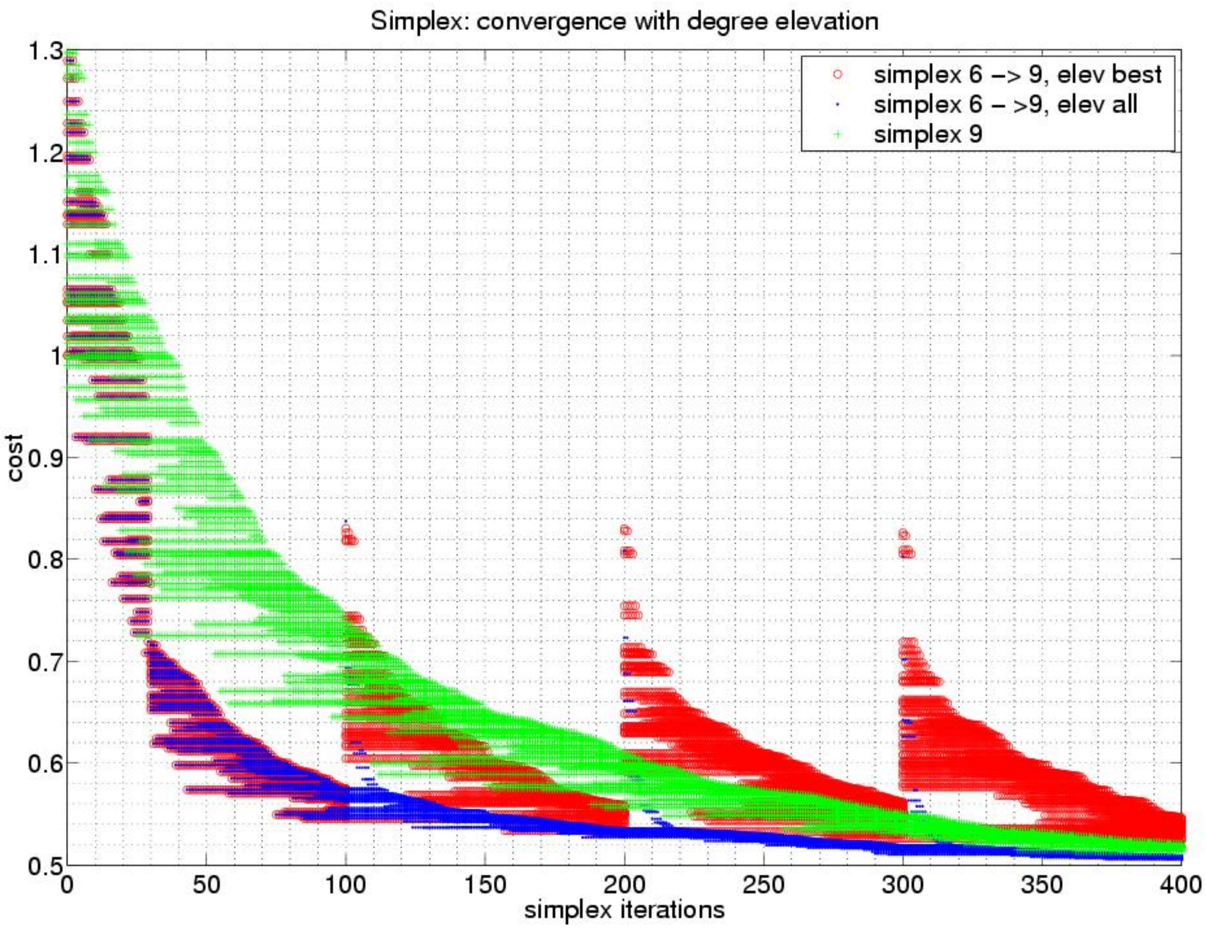




Simplex different degree: root and tip sections

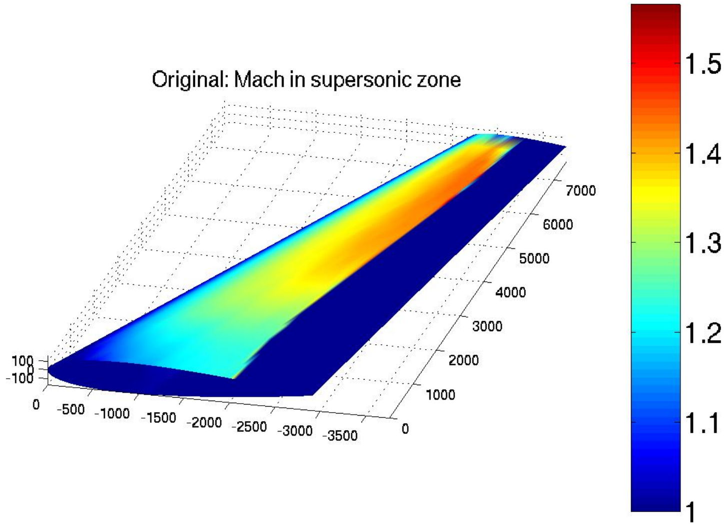


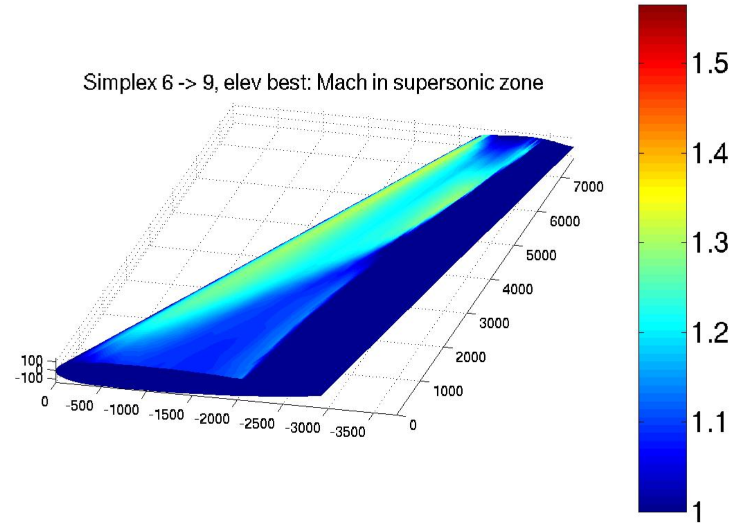
Simplex algorithm - degree elevation, large diameter

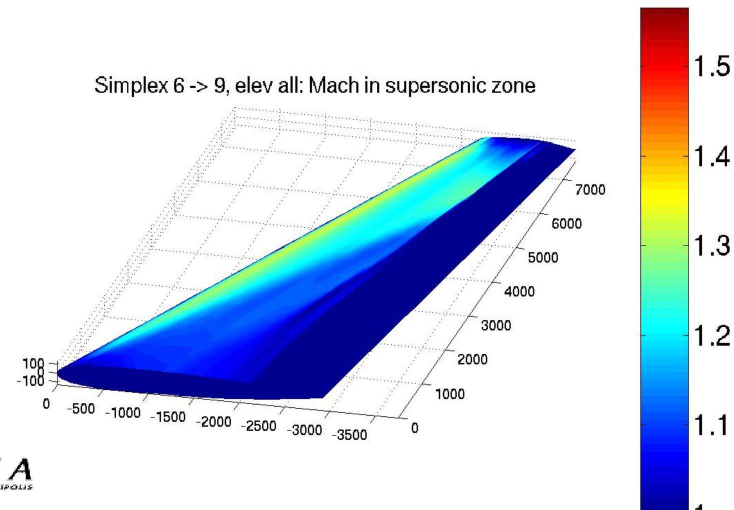


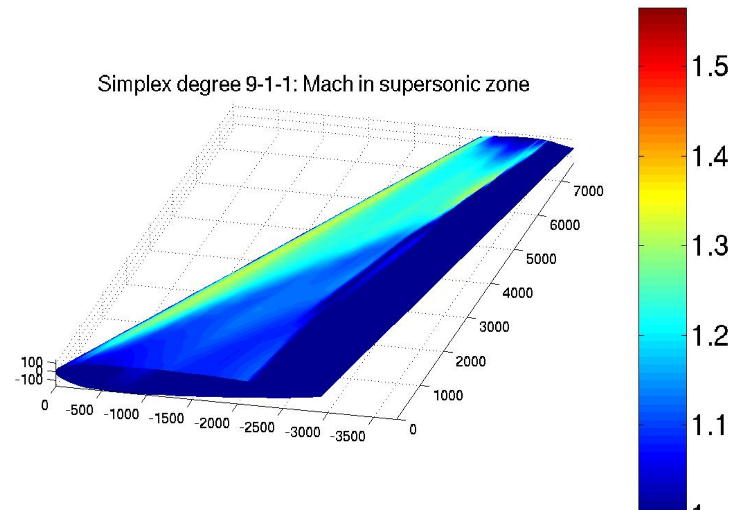
3 simplex optimizations with Bezier parametrization in chord-wise direction of different degrees:

- degree 6 elevated by one degree each 100 iterations of simplex method, only best result is being elevated, simplex is re-initialized by perturbation of best result,
- degree 6 elevated by one degree each 100 iterations of simplex method, the whole simplex is being elevated, or
- degree 9 only during 400 simplex iterations.

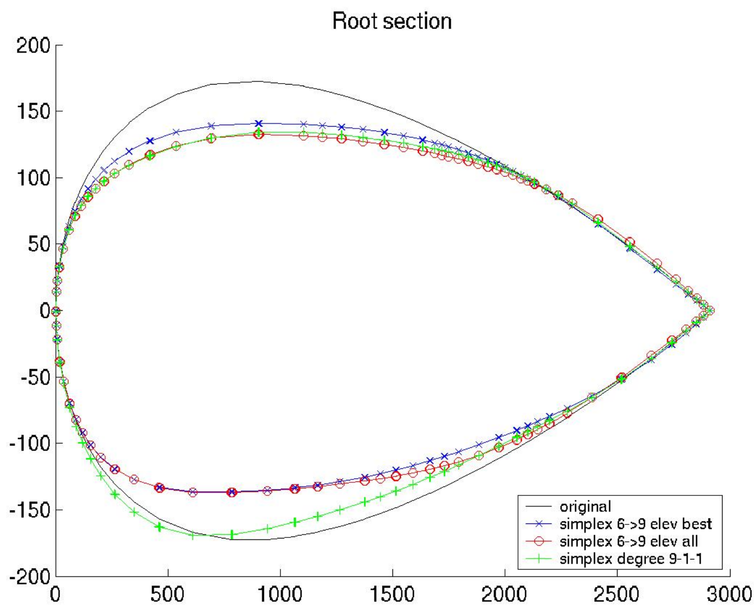

Original: Mach in supersonic zone

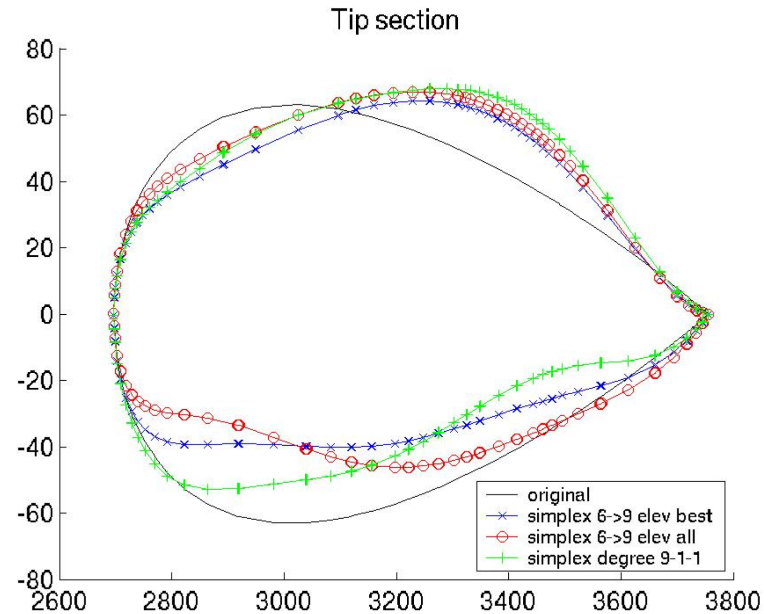

Simplex 6 -> 9, elev best: Mach in supersonic zone


Simplex 6 -> 9, elev all: Mach in supersonic zone


Simplex degree 9-1-1: Mach in supersonic zone

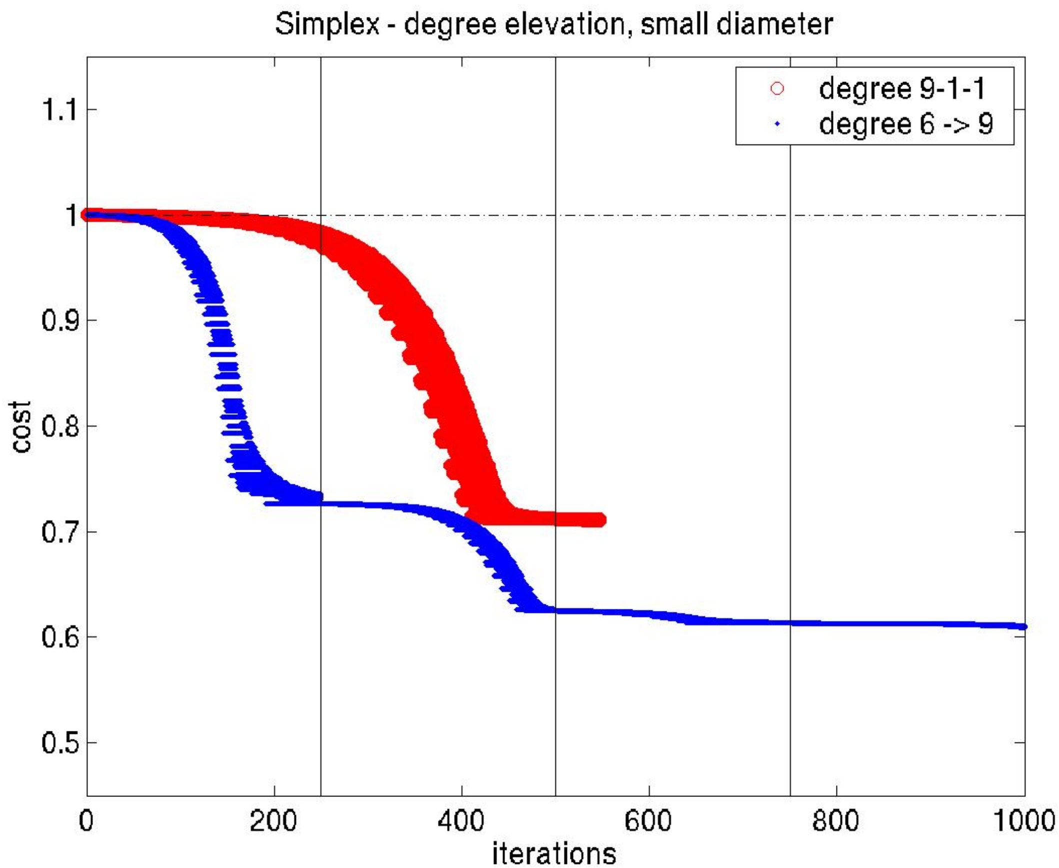
Simplex degree elevation: root and tip sections





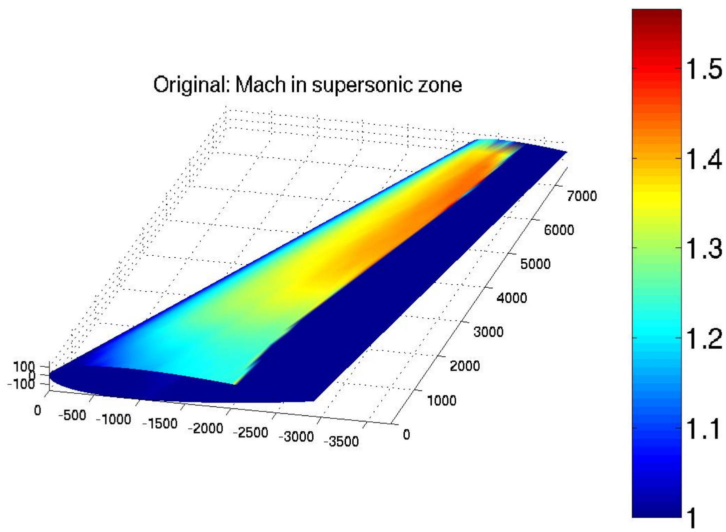
	C_L	C_D	Gain
orig	0.319200	0.026353	
6 $\longrightarrow$ 9 elev best	0.318885	0.012597	52.2%
6 $\longrightarrow$ 9 elev all	0.319477	0.012515	52.5%
degree 9-1-1	0.319079	0.013075	50.4%

Simplex algorithm - degree elevation, small diameter

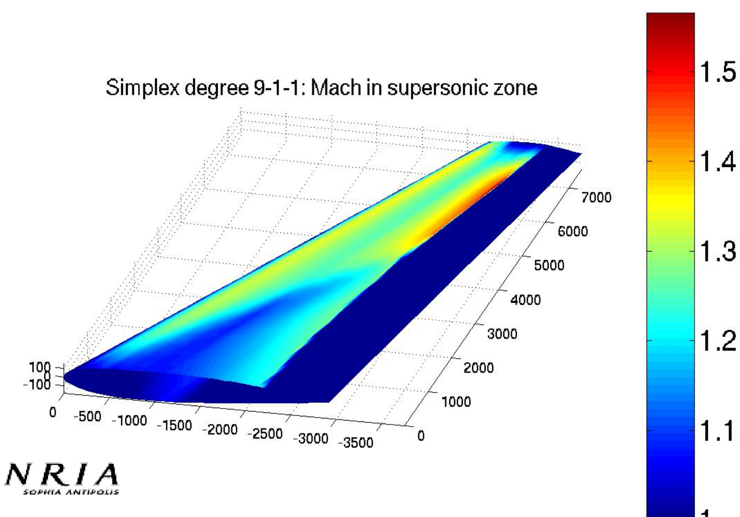


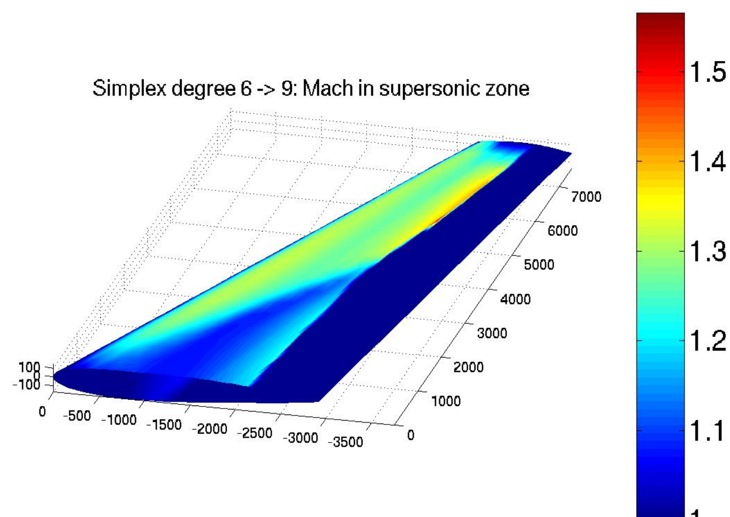
2 simplex optimizations with Bezier parametrization in chord-wise direction of different degrees:

- degree 6 elevated by one degree each 250 iterations of simplex method, only best result is being elevated, simplex is re-initialized by perturbation of best result, or
- degree 9 only during 550 simplex iterations.



	C_L	C_D	Gain
orig	0.319200	0.026353	
9-1-1	0.326127	0.018709	29%
6 $\longrightarrow$ 9	0.319166	0.016054	39%

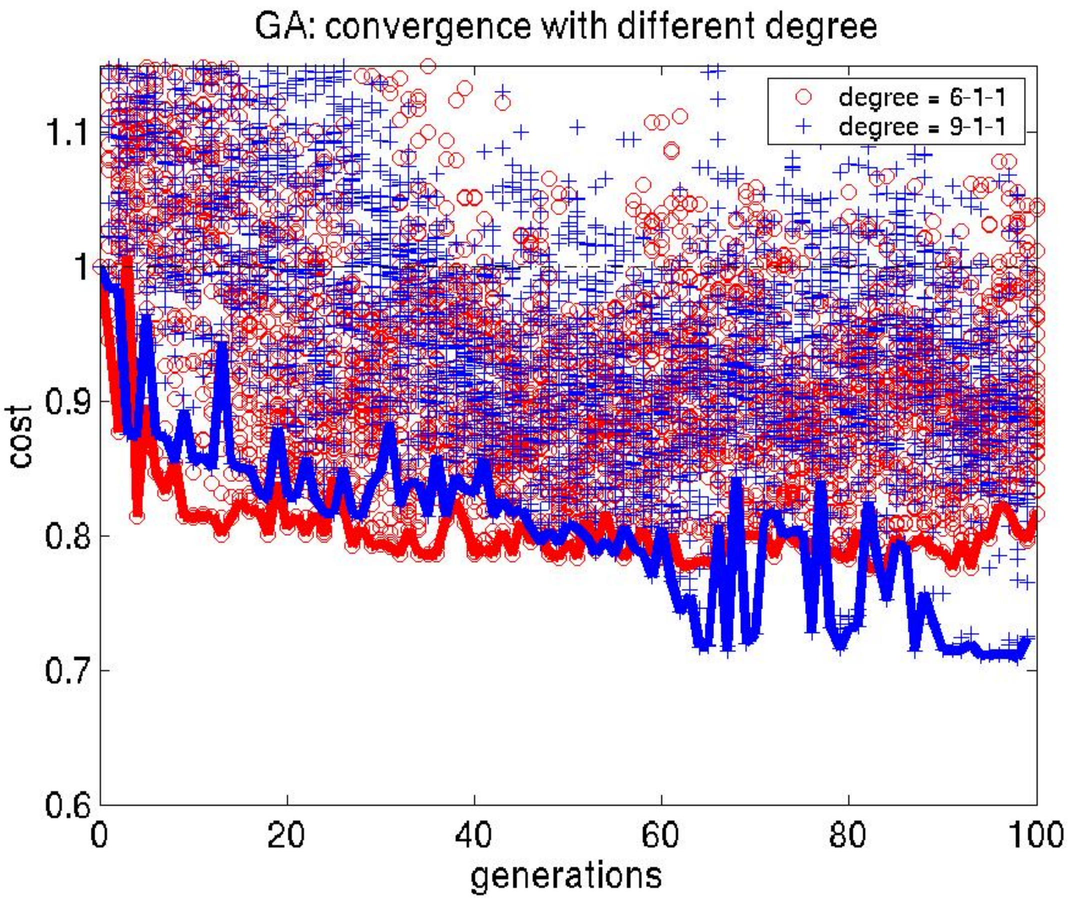




Genetic algorithm

- Coding of parameters: binary:
  - Search interval: $\pm 10\%$ of wing thickness
  - Precision: 0.005% of wing thickness
- Selection: roulette-wheel
- Crossover: two-point binary crossover with probability 85%
- Mutation: binary with probability 0.5%
- Population size: 40
- Number of parameters (3D): 20 for Bezier degree 6, 32 for Bezier degree 9
- Length of chromozome: 200 for Bezier degree 6, 320 for Bezier degree 9

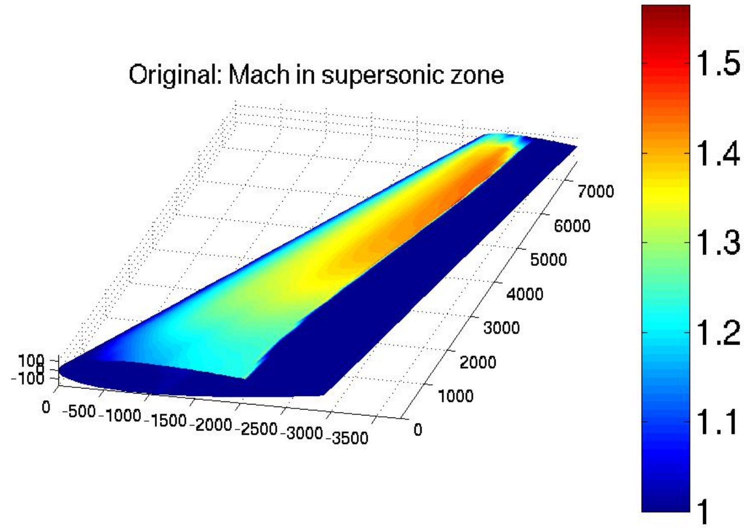
Genetic algorithm: different degree



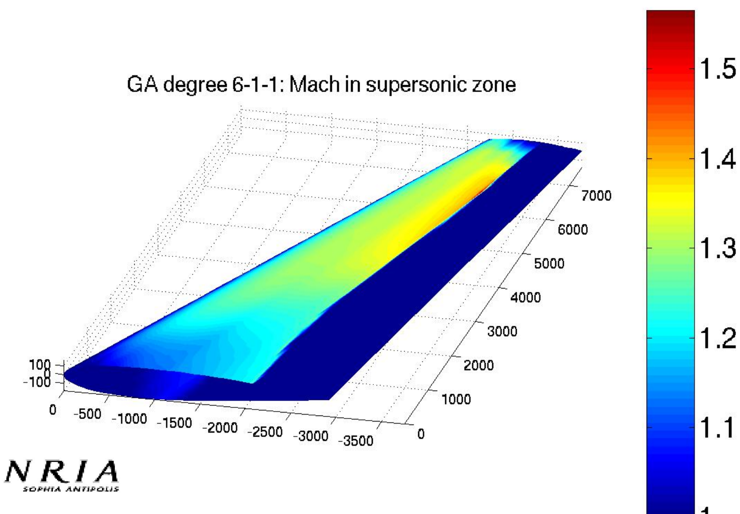
2 GA optimizations with Bezier parametrization in chord-wise direction of different degrees:

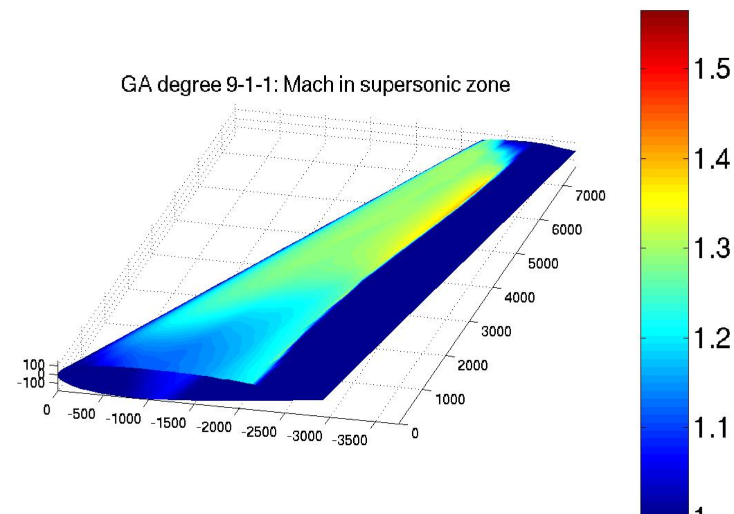
- degree 6 only during 100 generations, or
- degree 9 only during 110 generations.

	6-1-1	9-1-1
cost	0.775	0.709
gain	22.5%	29.1%

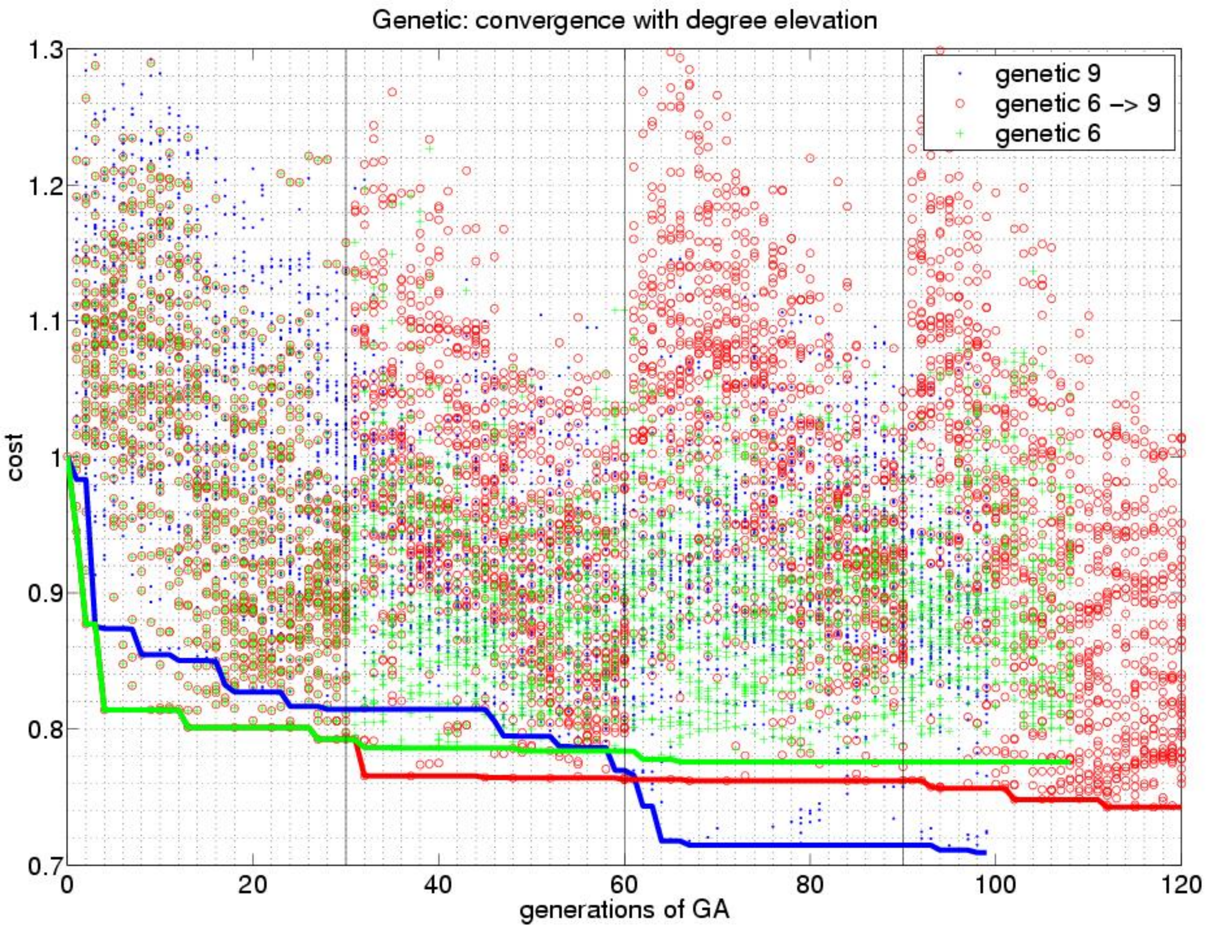


	C_L	C_D	Gain
orig	0.319200	0.026353	
6-1-1	0.319028	0.020443	22.5%
9-1-1	0.326346	0.018683	29.1%



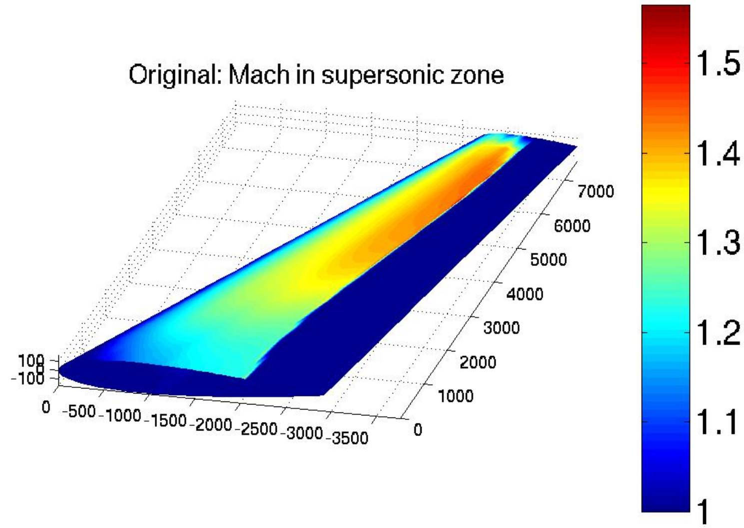


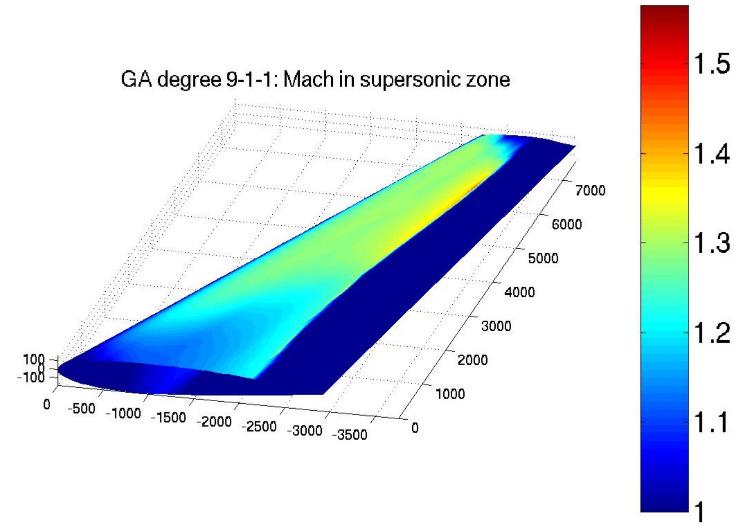
Genetic algorithm: degree elevation

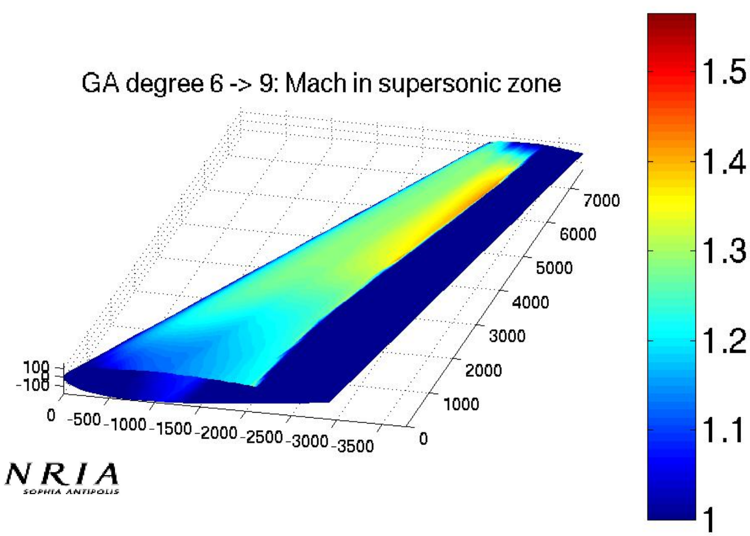


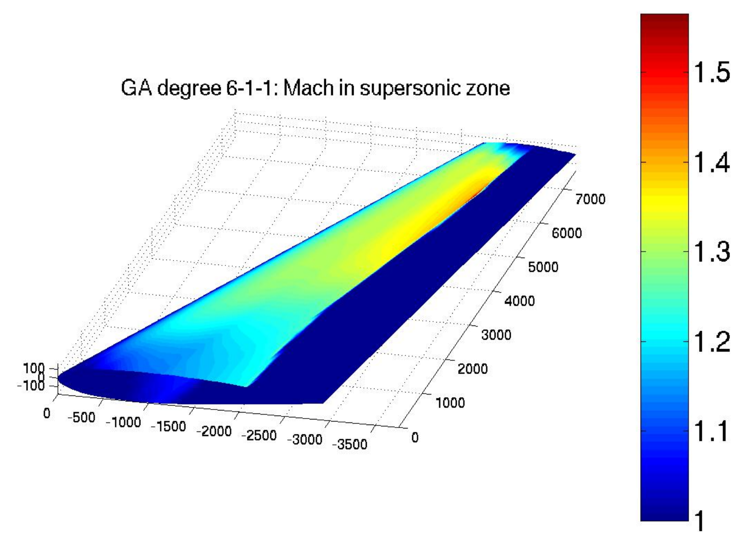
3 GA optimizations with Bezier parametrization in chord-wise direction of different degrees:

- degree 9 only during 100 generations, or
- degree 6 elevated by one degree after each 30 generations of GA, only best individual is being elevated, population is re-initialized by perturbation of best individual, or
- degree 6 only during 110 generations.

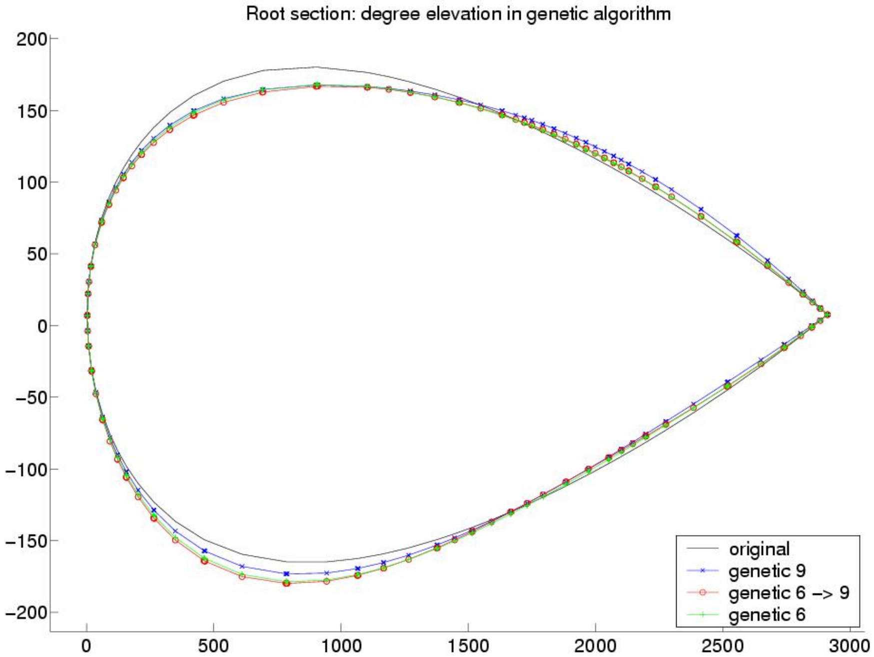

Original: Mach in supersonic zone
1.5
1.4
1.3
1.2
1.1
1

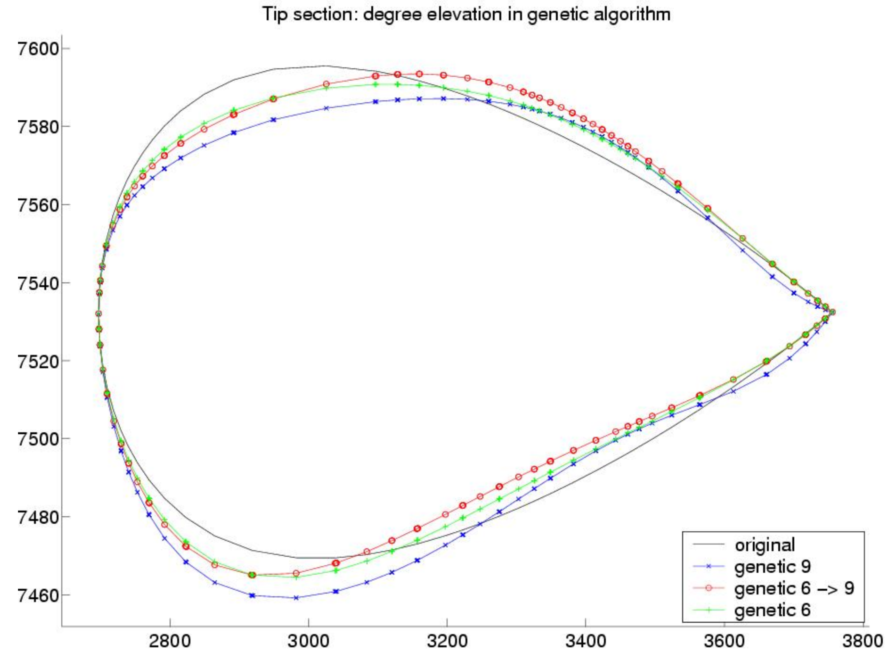

GA degree 9-1-1: Mach in supersonic zone
1.5
1.4
1.3
1.2
1.1
1


GA degree 6 -> 9: Mach in supersonic zone
1.5
1.4
1.3
1.2
1.1
1


GA degree 6-1-1: Mach in supersonic zone
1.5
1.4
1.3
1.2
1.1
1

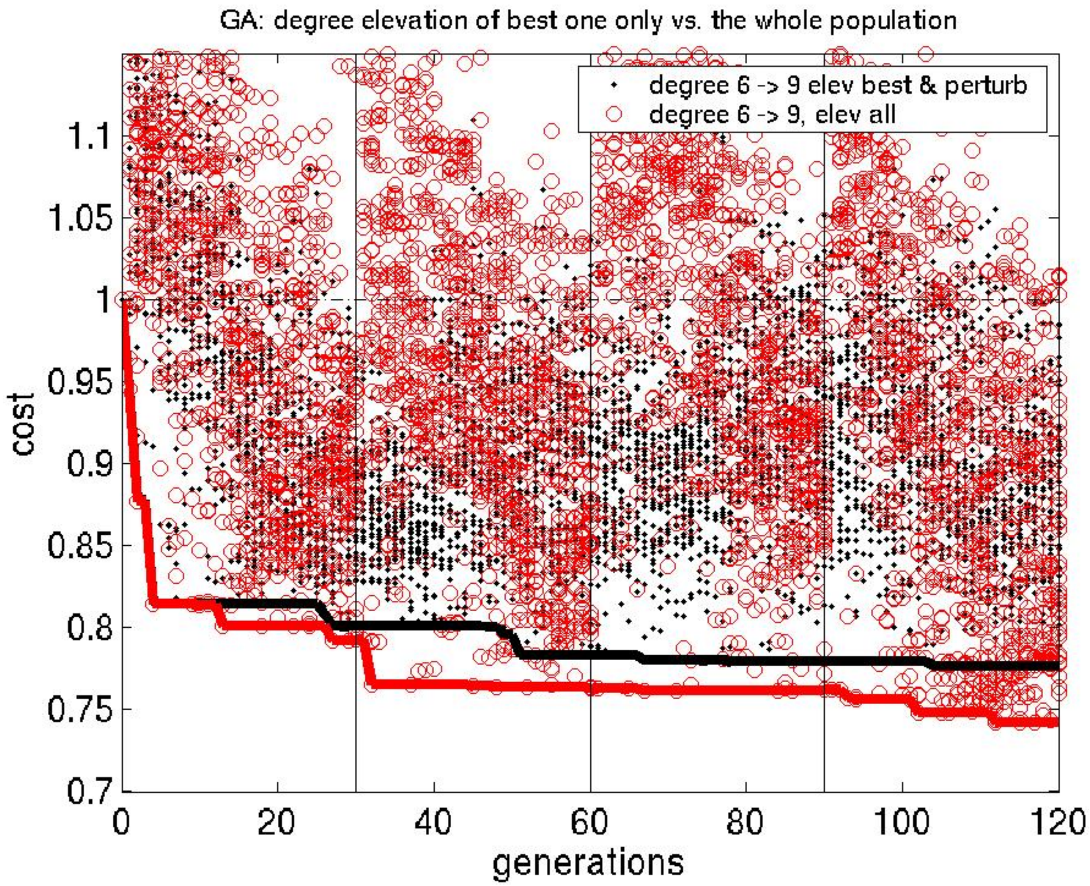
Genetic algorithm: root and tip sections





	C_L	C_D	Gain
orig	0.319200	0.026353	
degree 9-1-1	0.326346	0.018683	29.1%
$6 \longrightarrow 9$	0.320001	0.019564	26.5%
degree 6-1-1	0.319028	0.020443	22.5%

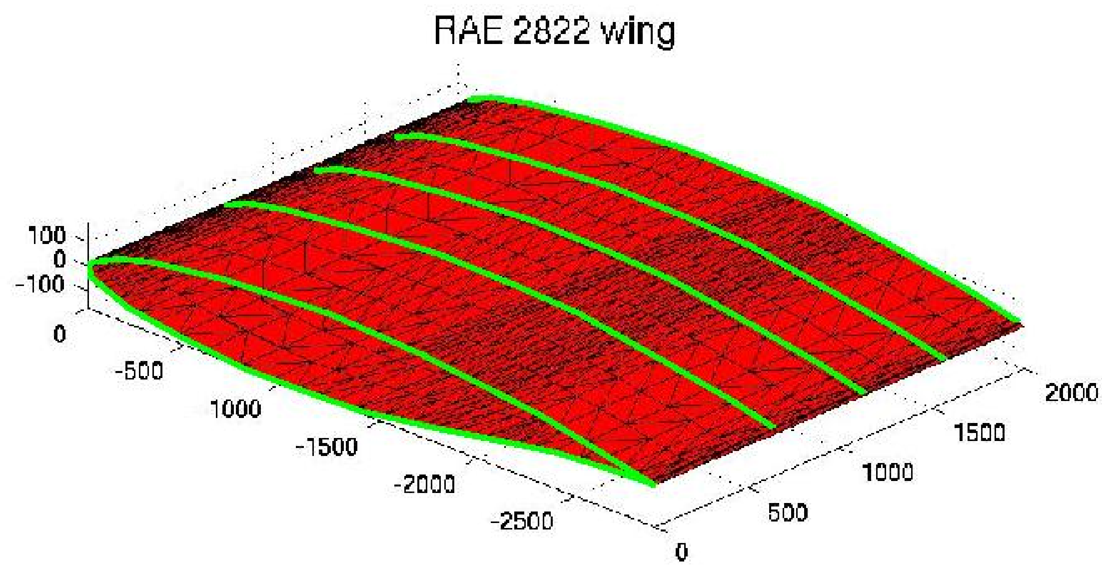
Genetic algorithm: elevate all or just the best?



2 GA optimizations with Bezier degree elevation, the same parametrization, difference in inheriting information at elevation process:

- after the elevation, only best individual is kept, rest of the generation is re-set by random perturbation
- after the elevation, all individuals are kept (no loss of information)

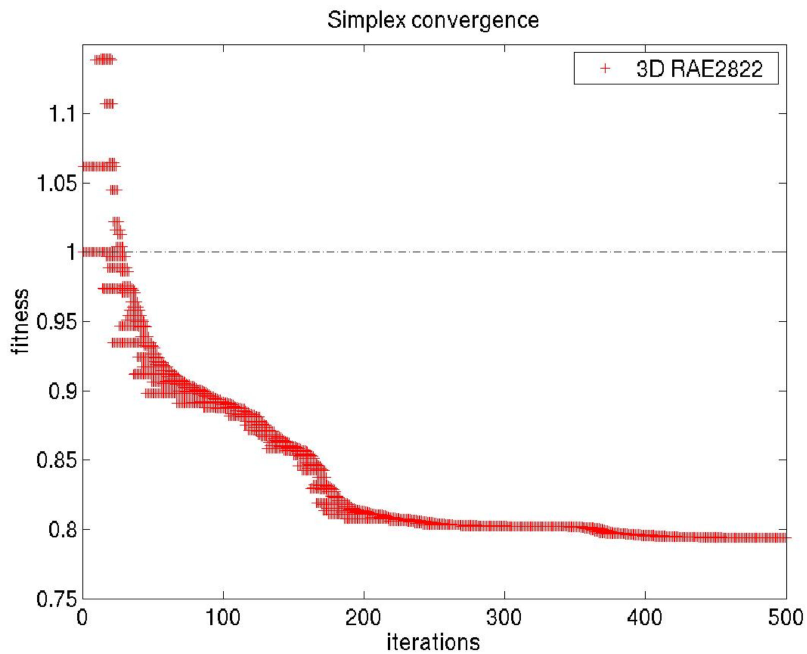
One particular 3D test-case



Simplex optimization using 3D RAE2822 wing:

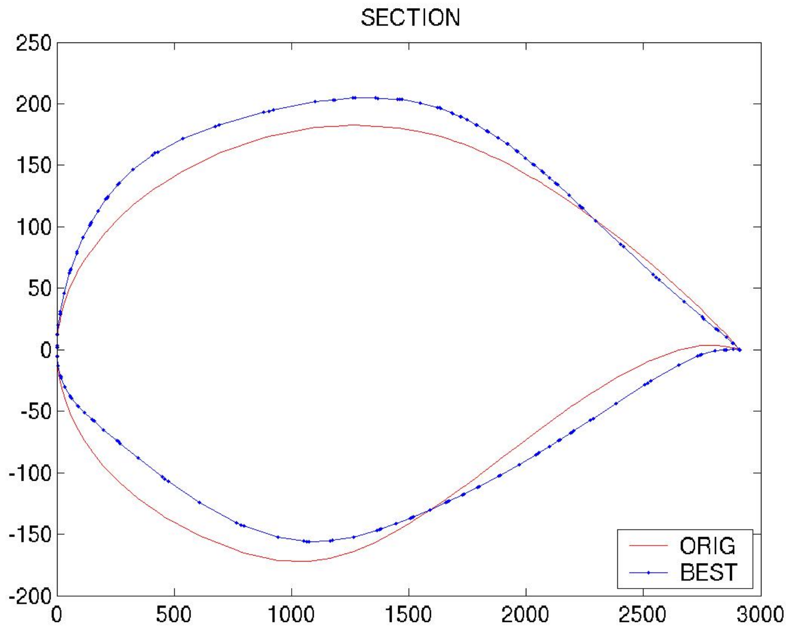
- 2D RAE2822 airfoil extruded in span-wise direction
- 2 symmetry planes at tip and root sections
- degree 6-1-0, diameter = 100, 500 iterations
- Mach = 0.73
- angle of incidence = 2°

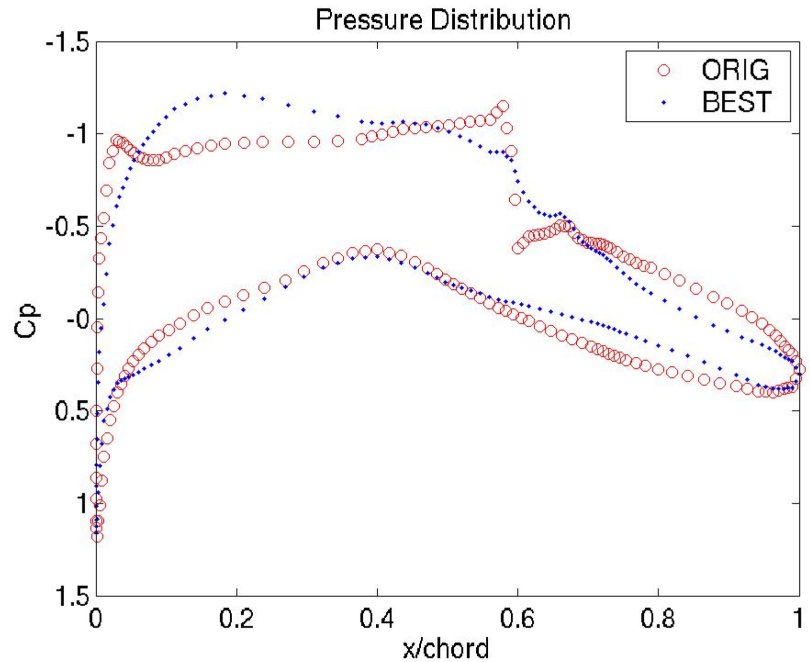
3D RAE2822 results



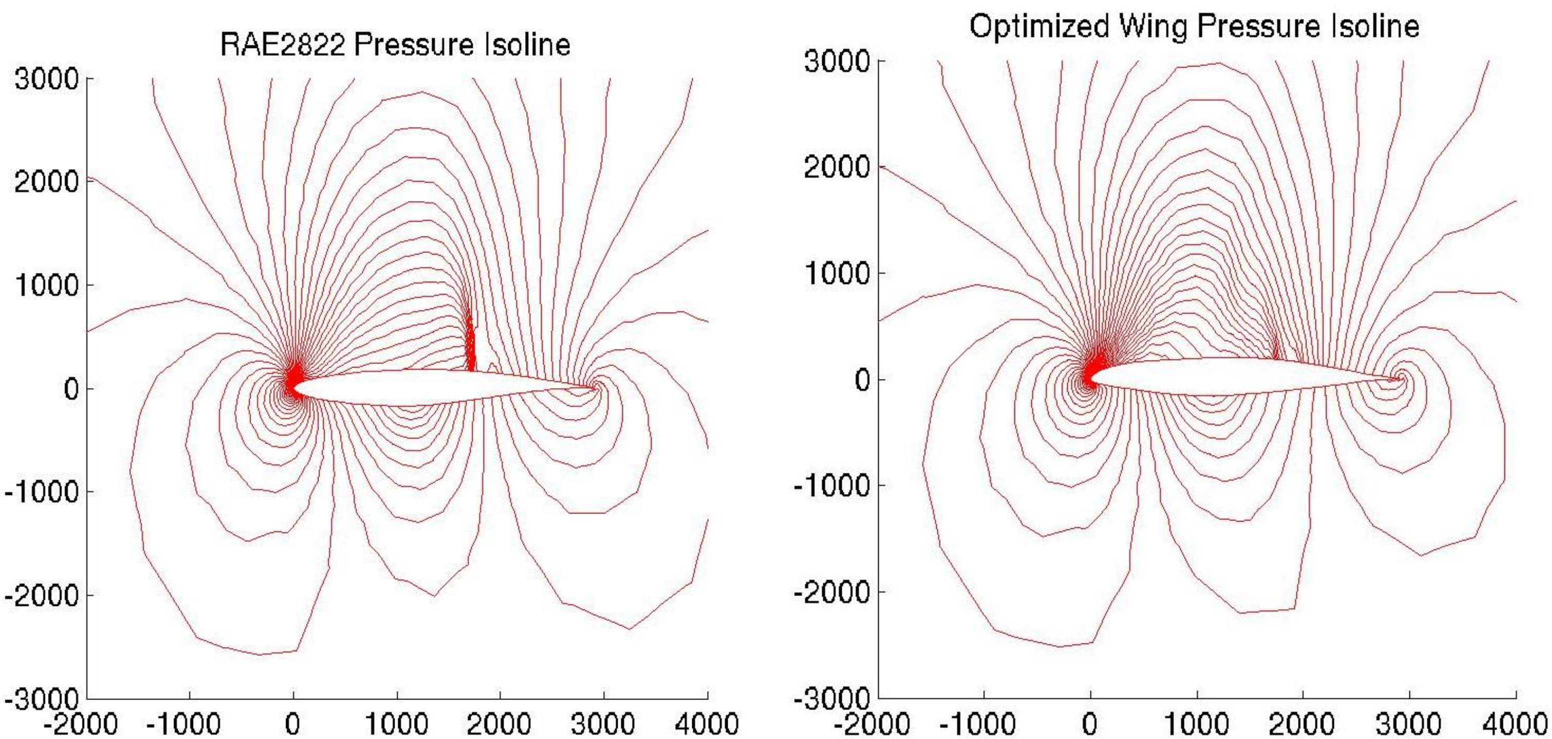
	C_L	C_D	Gain
RAE2822	0.698773	0.013167	
best	0.699080	0.010454	20.7%

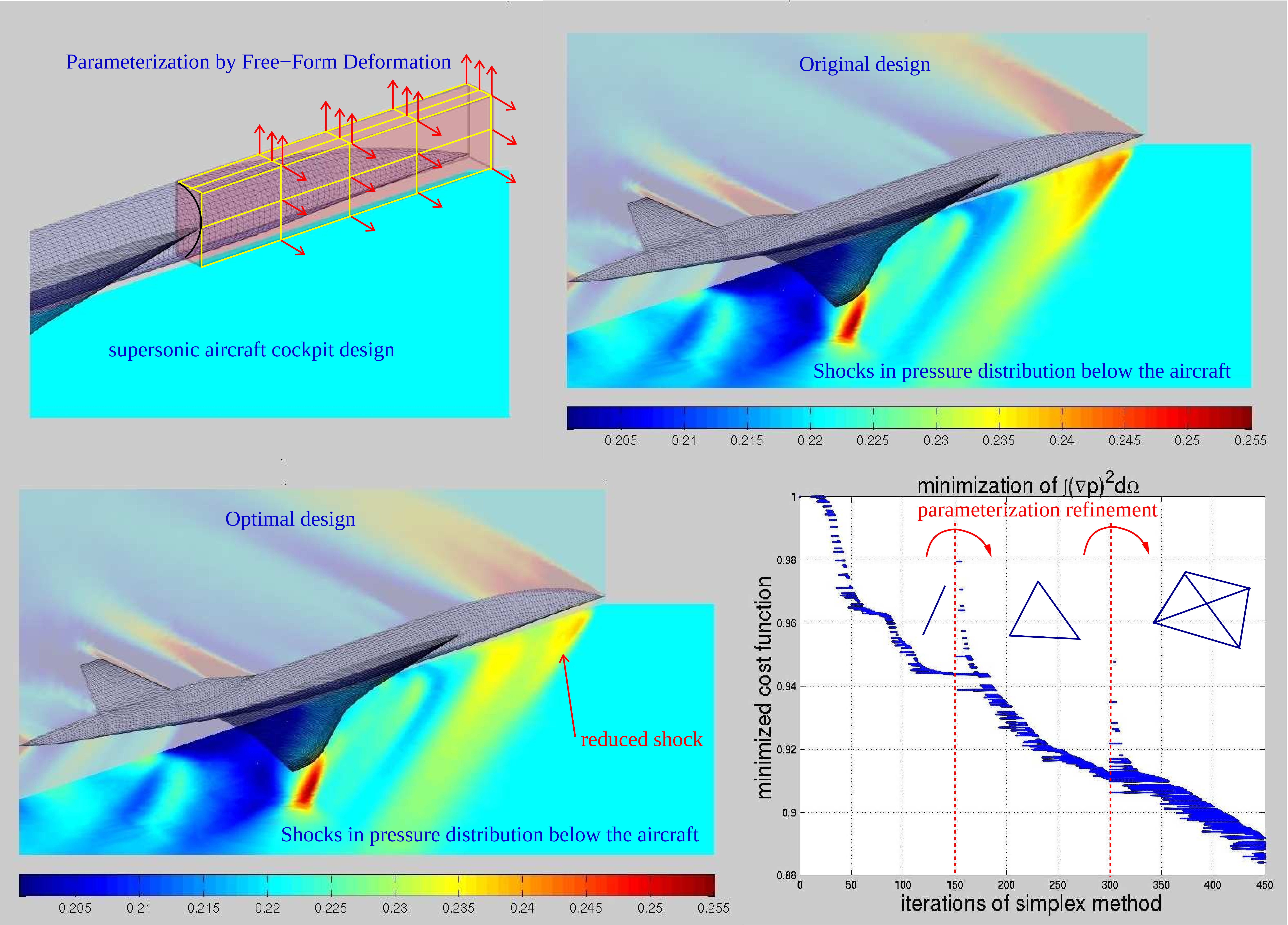
3D RAE2822 results



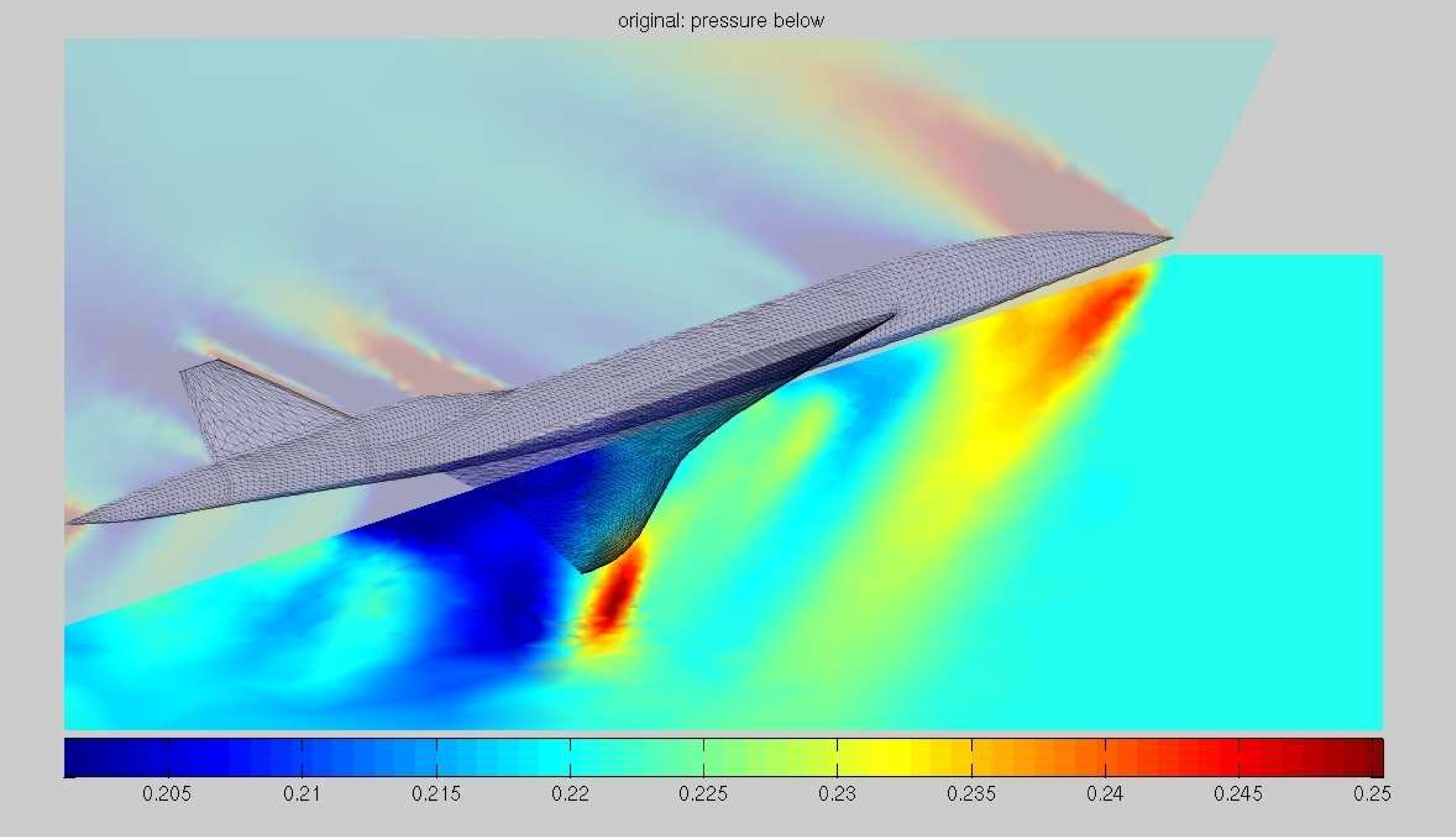


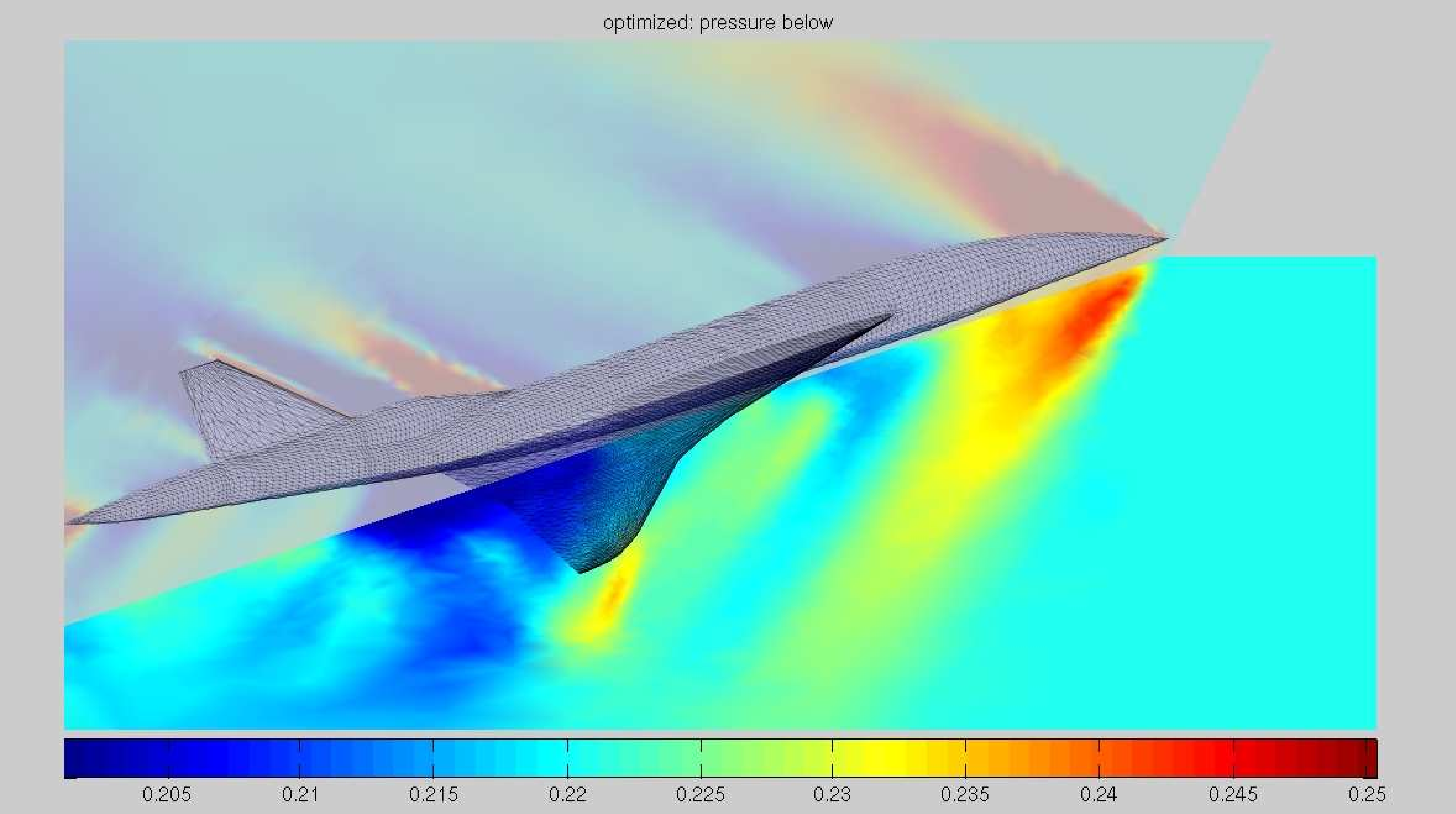
Pressure Isoline

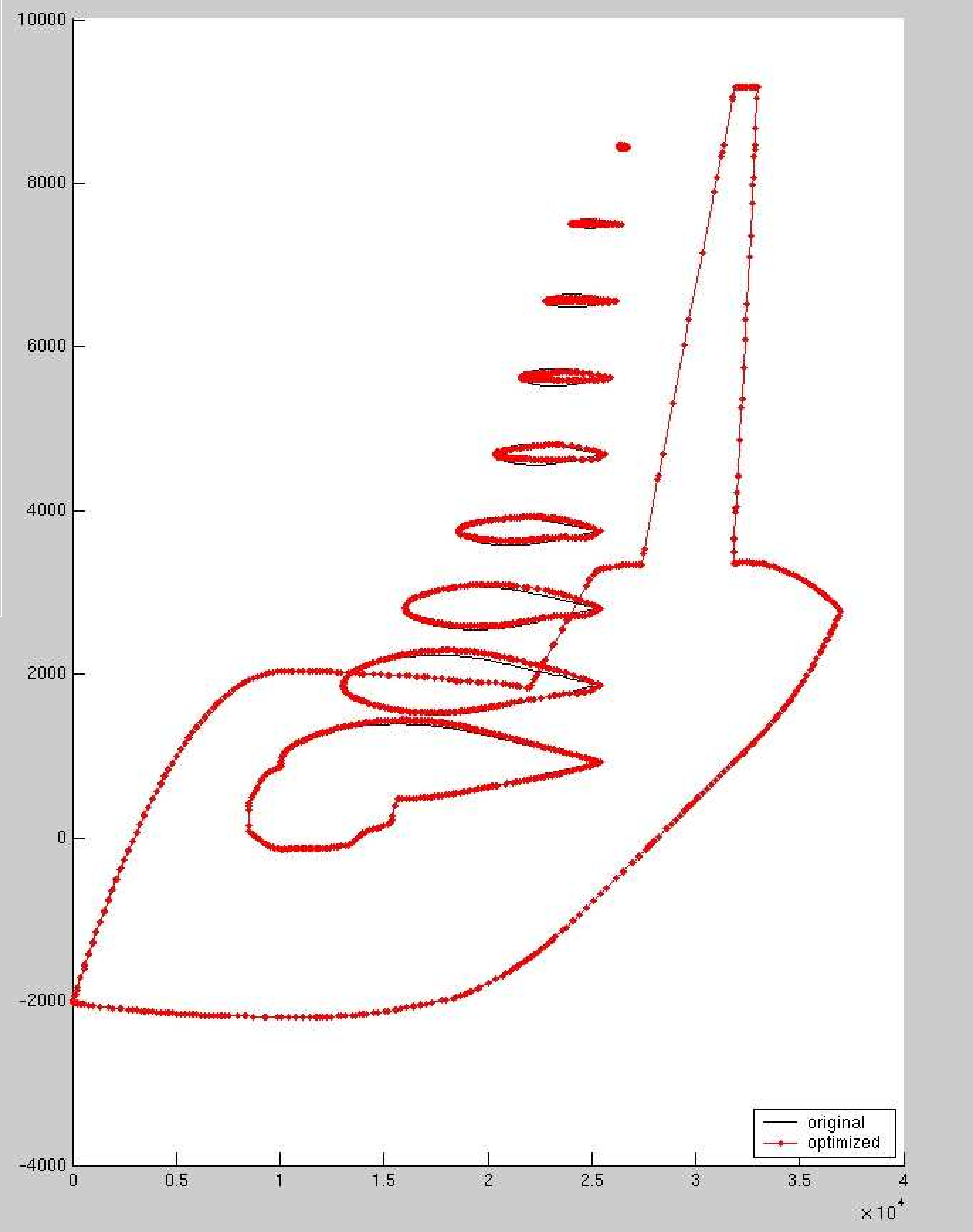




Parameterization by Free−Form Deformation
supersonic aircraft cockpit design
Original design
Shocks in pressure distribution below the aircraft
0.205 0.21 0.215 0.22 0.225 0.23 0.235 0.24 0.245 0.25 0.255
Optimal design
reduced shock
Shocks in pressure distribution below the aircraft
0.205 0.21 0.215 0.22 0.225 0.23 0.235 0.24 0.245 0.25 0.255
minimization of ∫(∇p)^{2}dΩ
parameterization refinement
minimized cost function
1 0.98 0.96 0.94 0.92 0.9 0.88
0 50 100 150 200 250 300 350 400 450
iterations of simplex method


original: pressure below
0.205
0.21
0.215
0.22
0.225
0.23
0.235
0.24
0.245
0.25


optimized: pressure below
0.205
0.21
0.215
0.22
0.225
0.23
0.235
0.24
0.245
0.25


10000
8000
6000
4000
2000
0
-2000
-4000
0
0.5
1
1.5
2
2.5
3
3.5
4
$\times 10^4$
original
optimized

Conclusions and perspectives

Conclusions:

- Hierarchical optimization algorithm based on multi-level parametrization (via degree elevation) is very promising (efficient with the simplex method)
- Multi-level genetic algorithm: information transfer from level to level still an open question: loss of genetic information vs. loss of search versatility
- Tensorial Bezier parametrization in conjunction with the free-form deformation technique provides a very versatile framework for 3D shape description and potentially also for automatic update of the computational 3D mesh

Perspectives and ongoing work:

- Adaptivity of the parametrization
- Cheap mesh update through free-form deformation