Computers and Structures 89 (2011) 1224–1231

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Shape optimization for drag reduction in linked bodies using evolution strategies

Mattia Gazzola [a], Oleg V. Vasilyev [b], Petros Koumoutsakos [a,*]

[a] Institute of Computational Science, ETH Zurich, CH-8092 Zurich, Switzerland
[b] Department of Mechanical Engineering, University of Colorado at Boulder, United States

ARTICLE INFO

Article history:
Received 15 June 2010
Accepted 2 September 2010
Available online 6 October 2010

Keywords:
Shape optimization
Flow optimization
Brinkman penalization
Covariance matrix adaptation–evolutionary strategy
Remeshed vortex particle method

ABSTRACT

We present results from the shape optimization of linked bodies for drag reduction in simulations of incompressible flow at moderate Reynolds numbers. The optimization relies on the covariance matrix adaptation evolution strategy (CMA-ES) and the flow simulations use vortex methods with the Brinkman penalization to enforce boundary conditions in complex bodies. We exploit the inherent parallelism of CMA-ES, by implementing a multi-host framework which allows for the distribution of the expensive cost function evaluations across parallel architectures, without being limited to one computing facility. This study repeats *in silico* for the first time Ingo Rechenberg's pioneering wind tunnel experiments for drag reduction that led to the inception of evolution strategies. The simulations confirm that the results of these experimental studies indicate a flat plate is not the optimal solution for drag reduction in linked bodies. We present the vorticity field of the flow and identify the governing mechanisms for this drag reduction by the slightly corrugated linked plate configuration.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Shape optimization for applications related to drag reduction is an increasingly important area of research in fluid mechanics due to the ever increasing concerns on aerodynamic energy efficiency. As shape optimization has progressed over the years, significant advancements in its capabilities have been developed. Major contributions starting with the analysis of energy loss in Stokes flow over various shapes [1–3] paved the way for research to come. One of the major challenges in shape optimization problems is that the majority of flows of engineering interest are unsteady, e.g., flutter, buffeting, dynamic stalls, and, to improve design performance, unsteady simulations are necessary thus, drastically increasing the computational cost. Further challenges include the highly inhomogeneous and intermittent nature of industrial flows [4], thus requiring the use of sophisticated computational and optimization approaches for dynamically adaptive simulations of high Reynolds number turbulent flows.

One of the key issues in developing effective approaches for shape optimization is the ability to model arbitrary flow geometries that can be easily modified in an automated fashion as part of the optimization process. This automation may imply the automated discretisation of the computational domain (without human interference) for accurate and computationally efficient flow simulations. Body fitted meshes usually result in more accurate flow simulations but in an automated design cycle one may question the value of the additional computational time required for the generation of these meshes. Furthermore, the reliance on human interference in the process of mesh generation, makes such meshes rather impractical for large scale optimization, especially when the flow geometry and, consequently, the mesh can be drastically different between different flow realizations. Reducing computational costs associated with the grid generation and more importantly full automation is one of the main task in developing robust approaches in flow optimizations. In this work we explore the use of Brinkman penalization [5,6] for shape optimization of fluid flows. The definition of the flow geometry using Brinkman volume penalization is simplified and automated: solid obstacles are modeled as porous media with porosity and permeability approaching zero. The main advantage of Brinkman penalization, compared to other techniques that allow for simulations past complex geometries using regular grids such as immersed boundary methods, is that the error between the solutions of penalized and non-penalized equations can be estimated rigorously in terms of the penalization parameter [7]. Furthermore, it can be shown that the solution of the penalized equations converges to the exact solution in the limit as the penalization parameter tends to zero [5,8].

The efficiency of the flow solvers are very important to optimization strategies that require large numbers of iterations such as genetic algorithms and evolution strategies. Here, we employ the method of evolution strategy [9] that was originally pioneered by Rechenberg and motivated by shape optimization for frag reduction in fluid mechanics. Rechenberg performed wind tunnel experiments in linked flat plates and changed their relative orientation

* Corresponding author.
E-mail addresses: mattia.gazzola@inf.ethz.ch (M. Gazzola), Oleg.Vasilyev@Colorado.edu (O.V. Vasilyev), petros@ethz.ch (P. Koumoutsakos).

0045-7949/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compstruc.2010.09.001

so as to obtain minimal drag. In his work optimizing the orientation of one plate at a time as well as the use of approximate gradients always led to curved geometries far from the perceived optimum of aligned linked bodies that amount to a flat plate. He then devised the concept of evolution strategies by performing random mutations to the angles of the linked bodies using a Galton box (Galtonbrett). Following this pioneering work, in the last 40 years there have been a number of advances in the development of effective ES and in particular with an emphasis on fluid mechanics problems [10–15,4]. Recent efforts [13–15,4] that aim to provide a mathematically consistent framework for ES have led to the development of the covariance matrix adaptation-evolution strategy (CMA-ES). The CMA-ES has been proven to be robust and effective in handling noise, multi-modality, and discontinuities in the cost function. In this work CMA-ES was extended to provide a multi-host framework able to distribute the computational load due to cost function evaluations across several parallel architectures. Such multi-host framework enhances CMA-ES performances in terms of available computational power as it allows the user to gather resources without being limited to only one computing facility.

In this work we combine remeshed vortex methods [16–19] along with Brinkmann penalization [20] for simulations of two dimensional flows past linked bodies, thus providing an *in silico* validation of the original experiments presented by Rechenberg. The simulations enable the quantification of the vortical structures that in turn enable the identification of the drag reduction mechanisms.

The paper is structured as follows: In Section 2 we introduce the shape optimization problem. Section 3 presents the methodologies employed to tackle such problem. Section 4 discusses the results and outlines future work.

2. Evolutionary strategy for shape optimization

In 1973, Ingo Rechenber, devised the first evolution strategy in order to optimize for drag reduction a set of linked flat plates [9]. His experimental setup (Fig. 1d) consisted of six flat plates connected by hinges at their extremities. The relative angle between the plates could be manually regulated in order to produce different structure configurations. He performed two experiments involving different global arrangements of the linked bodies with the axis of the flow, that hereafter are referred as Experiments 1 and 2. In this work we repeat these two experiments in simulations of 2D flows past linked plates.

In both experiments the parameters that determine the final structure are the five angles shown in Fig. 1a. These five angles are the parameters to be optimized. In this work, the five dimensional search space is bound in order to limit the number of infeasible configurations. The bounds are illustrated in Table 1a for Experiment 1 and in Table 1b for Experiment 2.

In Experiment 1, one extremity of the structure was constrained leaving to the corresponding plate only one degree of freedom, namely the angle of rotation. The second extremity was let free to rotate, and displace horizontally, while the vertical position

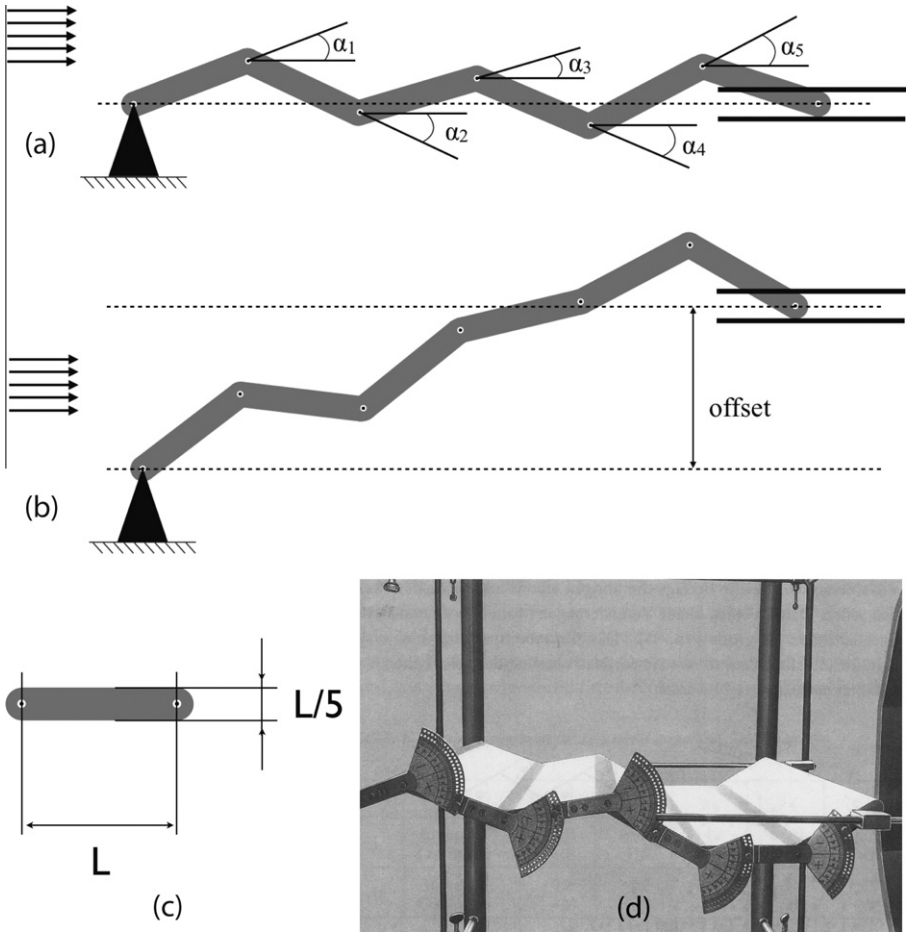

Fig. 1. Geometrical configurations of (a) Experiment 1, (b) Experiment 2 and (c) single plate as implemented in this work. (d) Original experimental setup by Rechenberg [9].

Table 1
Optimization setup. For the five angles to be optimized, bounds, initial mean (start search point) and standard deviation of CMA-ES multivariate distribution are indicated. Angles are expressed in degrees.

Experiment 1 $\lambda = 200$					Experiment 2 $\lambda = 200$				
	Min	Max	Start	Std		Min	Max	Start	Std
α_1	−45	45	20	45	α_1	−10	40	15	25
α_2	−45	45	20	45	α_2	−10	40	15	25
α_3	−45	45	20	45	α_3	−10	40	15	25
α_4	−45	45	20	45	α_4	−10	40	15	25
α_5	−45	45	20	45	α_5	−10	40	15	25

was constrained to be aligned with the first extremity (Fig. 1a). In our computational experiments each cost function evaluation consists of a simulation of flow past the structure at Reynolds number $Re = 1000$ for an incompressible, viscous fluid. During every simulation the generated drag C_D is tracked over time in order to compute the following cost function f to be minimized:

$$f = \frac{1}{T_2 - T_1}\int_{T_1}^{T_2} C_D\, dT, \quad T = \frac{2U_\infty}{L}t, \tag{1}$$

where C_D is the drag coefficient, T the dimensionless time, t the physical time and U_∞ the unperturbed velocity of the flow. For Experiment 1 we set $T_1 = 80$ and $T_2 = 100$. We chose this time interval for the drag averaging to avoid the initial transient and to account for oscillations in the drag evolution.

In Experiment 2, an offset was imposed in the vertical direction of the second extremity of the structure (Fig. 1b). The magnitude of the vertical displacement was set to offset $= 6L\sin(14°)$, where L represents the length of a single plate (Fig. 1c). The cost function used is the same as Eq. (1), with $T_1 = 100$ and $T_2 = 200$. The wider time interval was chosen to cope with the larger drag oscillations observed in this configuration.

3. Methodology

3.1. Remeshed vortex particle method with Brinkman penalization

We consider a two-dimensional incompressible flow governed by the Navier–Stokes equations in their vorticity–velocity formulation:

$$\frac{\partial\omega}{\partial t} + (\mathbf{u}\cdot\nabla)\omega = \nu\Delta\omega, \tag{2}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{3}$$

where $\omega = \nabla\times\mathbf{u}$ is vorticity and ν is the kinematic viscosity. In vortex methods the vorticity field is discretized by a linear superposition of particles. They carry a strength $\Gamma_p = \int_V \omega\, d\mathbf{x}$, where V is the particle volume. Particles are displaced according to the velocity field $\mathbf{u}$ and their strength is recomputed to account for the diffusion term. In order to avoid Lagrangian distortion, a remeshing approach is used, i.e. particle strengths are interpolated through a high order kernel onto an underlying regular grid at the end of each step [16,21,22]. In this work we used the third order accurate interpolation kernel M'_4 [18]. The presence of the particles on a regular grid, after remeshing, allows the use of fast discretization of the differential operators and fast grid based Poisson solvers.

$$\nabla^2\mathbf{u} = -\nabla\times\boldsymbol{\omega}. \tag{4}$$

We used a second order approximation of the diffusion operator and a Fourier Poisson solver extended for unbounded domains [23]. The presented remeshed vortex method is coupled with the Brinkman penalization [24,25], as it was introduced for vortex particle methods [20,26], in order to account for the presence of an obstacle O immersed in the fluid. Such model approximates the no-slip boundary condition at the boundary of the obstacles ∂O by adding to the Navier–Stokes equations a penalization term:

$$\frac{\partial\omega_\eta}{\partial t} + (\mathbf{u}_\eta\cdot\nabla)\omega_\eta = \nu\Delta\omega_\eta - \frac{1}{\eta}\chi_0\nabla\times\mathbf{u}_\eta, \tag{5}$$

$$\nabla\cdot\mathbf{u}_\eta = 0, \tag{6}$$

where $\eta \ll 1$ is the penalization factor (in this work $\eta = 10^{-4}$) and χ_O $(0 \leqslant \chi_O \leqslant 1)$ is the mollified characteristic function that localizes the obstacle O in the flow field:

$$\chi_O = \begin{cases} 0 & d < -\epsilon, \\ \frac{1}{2}\left[1 + \frac{d}{\epsilon} + \frac{1}{\pi}\sin\left(\pi\frac{d}{\epsilon}\right)\right] & |d| \leqslant \epsilon, \\ 1 & d > \epsilon, \end{cases}$$

where d is the signed distance function and ϵ is the mollification length. In this work we set $\epsilon = 2\sqrt{2}h$, h being the grid spacing. It is important to note that as $\eta \to 0$, it was proved theoretically by Carbou and Fabrie [8] that the error of the solution of the penalized equations (Eqs. 5 and 6) is bounded by $\|\mathbf{u} - \mathbf{u}_\eta\| \leqslant C\eta^{1/2}\|\mathbf{u}\|$. Furthermore, η is an arbitrary parameter, independent of the spatial or temporal discretization, and thus the boundary conditions can be enforced to any desired accuracy by choosing η appropriately, provided that the penalized equations are properly resolved. This property distinguishes the Brinkman method from other penalization schemes and allows the error to be controlled precisely.

The main advantage of the Brinkman penalization is that Eqs. (5) and (6) are valid on the whole domain, providing a simple way to deal with complex geometries. Furthermore the force $\mathbf{F}$ acting on the obstacle is recovered by integrating the penalization term in Eq. (5). Finally, the implemented approach comprises the good stability properties of vortex methods for convection [18].

3.2. Covariance matrix adaptation-evolutionary strategy

The shape optimization problem addressed in this work is nonlinear, likely multimodal and inherently noisy. Furthermore the cost function derivatives in terms of the optimization parameters are not readily available and their estimation would be computationally expensive and possibly inaccurate. In order to circumvent these difficulties we obtained the cost function using the flow solver as a black box and tackled the optimization problem via an evolutionary algorithm, instead of using a gradient based method. We employed the covariance matrix adaptation-evolutionary strategy (CMA-ES) [13,27,14]. CMA-ES samples a multivariate normal distribution, which is progressively learned from past samples and adapted based on the information gathered in the course of the optimization. This algorithm, in its rank-μ and weighted recombination form, has been proven to be robust and effective in handling noise, non-linearity and multi-modality in the cost function [15].

The formulation of CMA-ES is given in Algorithm 1. At the beginning of each generation, a population of λ individuals $\mathbf{y}_k$ is sampled (Eqs.)(7)–(9) from a multivariate normal distribution $\mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$, characterized by mean $\mathbf{m}$, overall standard deviation σ and covariance matrix $\mathbf{C}$. Such individuals are then evaluated through the cost function f (Eq. 10) and ranked from the best to the worst, according to the corresponding cost function value. Given the best μ individuals, $\mathbf{m}$, σ and $\mathbf{C}$ are updated in a derandomized fashion to increase the probability of sampling good individuals in the next generation Eqs. (11)–(15).

Algorithm 1. CMA-ES in its rank-μ and weighted recombination form. Notation: $\langle\mathbf{a}\rangle_w = \sum_{i=1}^{\mu} w_i\mathbf{a}_{1:\lambda}$ indicates the weighted recombination average, where w_i are its coefficients and the index $i{:}\lambda$ denotes the ith best individuals, $\mu_{\text{eff}} = 1/\sum_{i=1}^{\mu} w_i^2$ is the "variance

effective selection mass", $E(\cdot)$ indicates the expected value, c_c, c_{cov}, c_σ and d_σ are empirical coefficients. Initially $\mathbf{C} = \mathbf{I}$, $\mathbf{p}_c = \mathbf{0}$ and $\mathbf{p}_\sigma = \mathbf{0}$. For further details refer to [13,27,14,15].

for $k = 1, \ldots, \lambda$

$$\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{7}$$

$$\mathbf{x}_k = \sqrt{\mathbf{C}}\mathbf{n}_k \tag{8}$$

$$\mathbf{y}_k = \mathbf{m} + \sigma \mathbf{x}_k \tag{9}$$

$$f_k = f(\mathbf{y}_k) \tag{10}$$

end

$$\mathbf{m} = \mathbf{m} + \sigma \langle \mathbf{x} \rangle_w \tag{11}$$

$$\mathbf{p}_c = (1 - c_c)\mathbf{p}_c + \sqrt{c_c(2 - c_c)}\sqrt{\mu_{\text{eff}}}\langle \mathbf{x} \rangle_w \tag{12}$$

$$\mathbf{C} = (1 - c_{cov})\mathbf{C} + \frac{c_{cov}}{\mu_{\text{eff}}}\mathbf{p}_c\mathbf{p}_c^T + c_{cov}\left(1 - \frac{1}{\mu_{\text{eff}}}\right)\langle \mathbf{x}\mathbf{x}^T \rangle_w \tag{13}$$

$$\mathbf{p}_\sigma = (1 - c_\sigma)\mathbf{p}_\sigma + \sqrt{c_\sigma(2 - c_\sigma)}\sqrt{\mu_{\text{eff}}}\langle \mathbf{n} \rangle_w \tag{14}$$

$$\sigma = \sigma \exp\left[d_\sigma\left(\frac{\|\mathbf{p}_\sigma\|}{\mathrm{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1\right)\right] \tag{15}$$

In CMA-ES the choice of the population size (λ) is crucial. A small population ($\lambda \sim 10$) reduces the number of total evaluations to converge to a solution, but increases the number of required generations. On the other hand a large population ($\lambda \sim 100$), reduces the generations and augments the number of evaluations [14]. A high value of λ also increases the robustness of the algorithm [14,15]. In terms of time to solution, a sequential cost function evaluation (4–8 h each) would make the use of large λ impractical. Given our framework, which allows parallel evaluation and submission to supercomputing facilities, we chose to set $\lambda = 200$ to ensure robustness and fast convergence in terms of generations (<10, for a time to solution of the whole optimization of approximately ~100 h).

The search space is bounded as indicated in Table 1. Bounds are enforced during the sampling through a rejection algorithm, and we deal with invalid configurations (due to the end point constraint, see Fig. 1a and b) assigning a high default cost function value.

3.3. Coupling of flow solver and optimizer

The CMA-ES algorithm is not computationally costly and can be run on a local workstation. On the other hand, our cost function

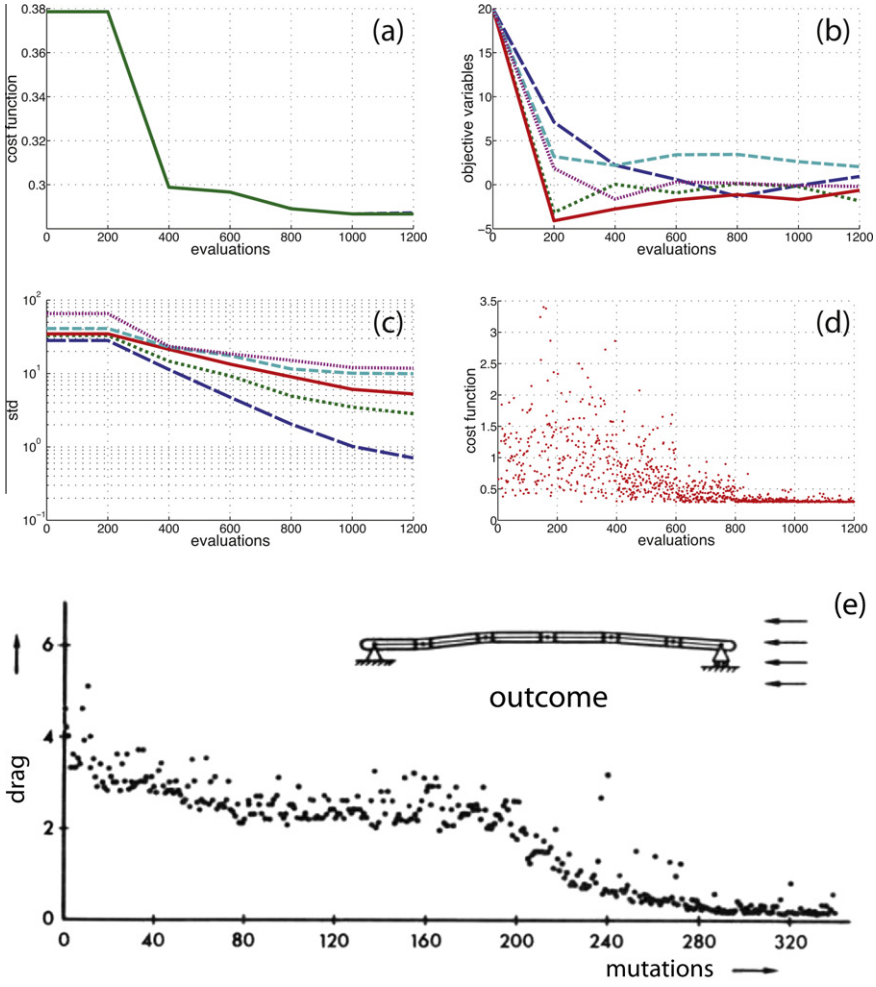

Fig. 2. Optimization history of Experiment 1 using vortex methods with Brinkman penalization: (a) cost function f, (b) parameter values (mean $\mathbf{m}$ of the multivariate distribution, blue = α_1, green = α_2, red = α_3, cyan = α_4, violet = α_5) and (c) standard deviation of the principal axes versus number of cost function evaluations. (d) and (e) Scatter plots showing the cost function value of each parameter set during the optimization process, respectively, in our simulation and in the original setup by Rechenberg [9]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

evaluations are computationally expensive and require access to parallel architectures. We take advantage of the fact that cost function evaluations within a CMA-ES generation can be computed independently. We developed a multi-host CMA-ES framework that, through an automated scripting language, can distribute cost function evaluations across many hosts (CPU/multicore/GPU clusters). Each evaluation is submitted to one such host as a job. The framework waits until all evaluations are performed and collects the necessary results in order to proceed in the optimization procedure. This multi-host implementation greatly enhances CMA-ES performance in terms of available computational power as it allows the user to gather resources, without being limited to only one computing facility. The CMA-ES framework is completely independent from the problem-specific application, which is treated as a black box. The communication between optimizer and solver is carried out through the exchange of two files: one from CMA-ES to the application, containing the parameter set to be evaluated, and one from the application to CMA-ES, containing the corresponding computed cost function value. This way generality and flexibility are preserved and the interfacing between optimizer and application requires minimal code rewriting.

4. Results

4.1. CMA-ES and vortex methods with penalization: Experiment 1

The evolution of the optimization process for Experiment 1 using vortex methods with Brinkman penalization is shown in Fig. 2. As can be seen CMA-ES required six generations, for a total of 1200 cost function evaluations. Each evaluation run on 16 cores for approximately ~4 h. This entails a total of 80,000 CPU hours. Initially, the evolutionary optimization process produces irregular and highly asymmetric linked bodies arrangements. For example, the worst candidate in the first generation (Fig. 3) presents a large frontal area to the flow, leading to the shedding of powerful vortices that increase the measured drag. In the course of the optimization such configurations are discarded and the best solution found (Table 2a, Fig. 4) leads to a drag reduction of ~26%, with respect to the best candidate in the first generation.

Interestingly, in the present studies, and consistent with the findings in Rechenberg's work, the optimal configuration (Figs. 4 and 2e) consists of a corrugated form and are not flat, as intuition would suggest. To make sure that CMA-ES did not get stuck in a local minima, without converging to the assumed optimum of a flat plate, we simulated the flat plate configuration (Fig. 5). It turned out that the best candidate performs slightly better than the flat plate (Fig. 6a), with a drag ~3.5% lower ($f_{\text{flat}} = 0.2966$). This drag reduction is attributed to the secondary vorticity produced at the first and second bending of the corrugated plate (Fig. 4). The secondary vorticity leads, along with the primary vorticity, to a stable asymmetric dipole like structure that has lower drag than that induced by the primary vortex alone on both sides of the plate. Furthermore this secondary vorticity weakens the primary vortex and, in turn, delays the instability and roll-up of the vortices that are shed in the wake on each side of the plate. The vorticity magnitude and spread in the wake is then slightly reduced with respect to the flat plate (all figures are plotted using the same color map and, as can be seen, Fig. 4 presents less intense vortices than Fig. 5), and there is accordingly a drag reduction for the corrugated configuration with respect to the flat plate.

The deviation of our solution from the one found in Rechenberg's Experiment 1 (Fig. 2e) could be related to the different experimental conditions (Rechenberg's experiments were carried out using compressible flow at unspecified Reynolds and Mach numbers) or to the fact that the solution identified by CMA-ES

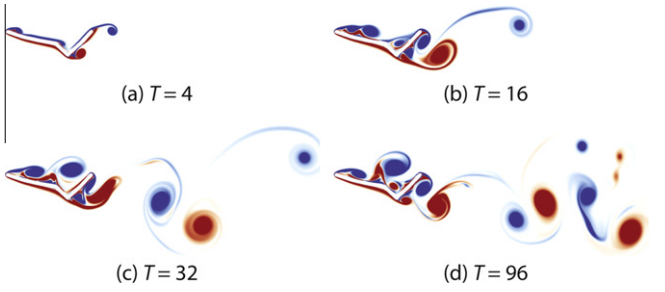

Fig. 3. Experiment 1 using vortex methods with Brinkman penalization: vorticity at different dimensionless times (T) for the worst candidate in the first generation of the optimization process.

Table 2
Optima: best candidates found for Experiments 1 and 2 using vortex methods with Brinkman penalization. Angles are expressed in degrees.

Vortex methods	Exp1	Exp2
α_1	−10.82	6.24
α_2	13.23	−7.82
α_3	3.97	4.79
α_4	0.33	36.50
α_5	−5.77	36.99
f	0.2867	1.23321

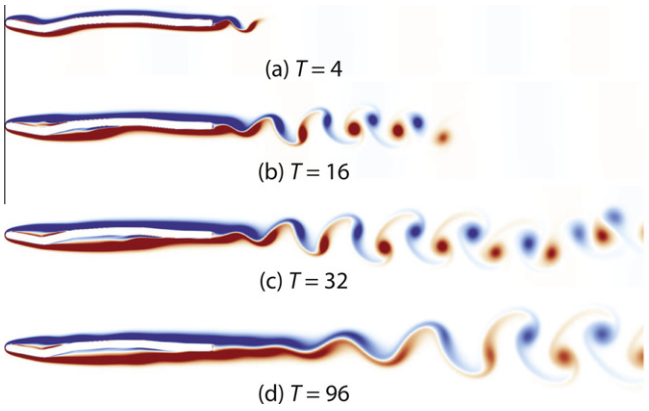

Fig. 4. Experiment 1 using vortex methods with Brinkman penalization: vorticity at different dimensionless times (T) for the best candidate ever of the optimization process.

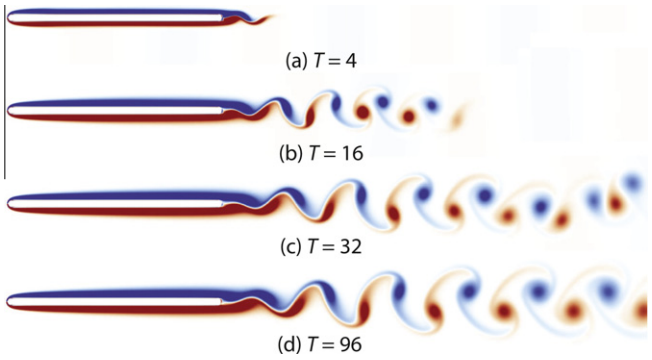

Fig. 5. Experiment 1 using vortex methods with Brinkman penalization: vorticity at different dimensionless times (T) for flat plate configuration.

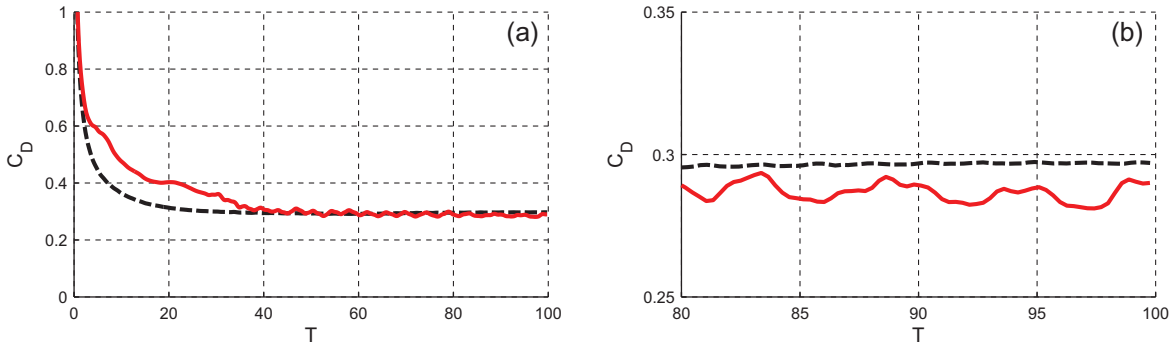

Fig. 6. Experiment 1 using vortex methods with Brinkman penalization: drag (C_D) curve against dimensionless times (T) for flat configuration (black) and best candidate ever of the optimization process (red). (a) Drag evolution over the entire simulated time. (b) Enlargement over the period taken into account in the cost function definition. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(or by Rechenberg's evolutionary strategy) may not correspond to the global minimum (stochastic optimization does not ensure that the global minimum is found). Finally we note that the present results involve two-dimensional simulations that may not represent all the governing flow physics of the three-dimensional flow configuration in the experimental setup.

4.2. CMA-ES and vortex methods with penalization: Experiment 2

The evolution of the optimization process for Experiment 2 using vortex methods with Brinkman penalization is shown in Fig. 7. CMA-ES performed seven generations, for a total of 1400 cost function evaluations. Each evaluation run on 16 cores for

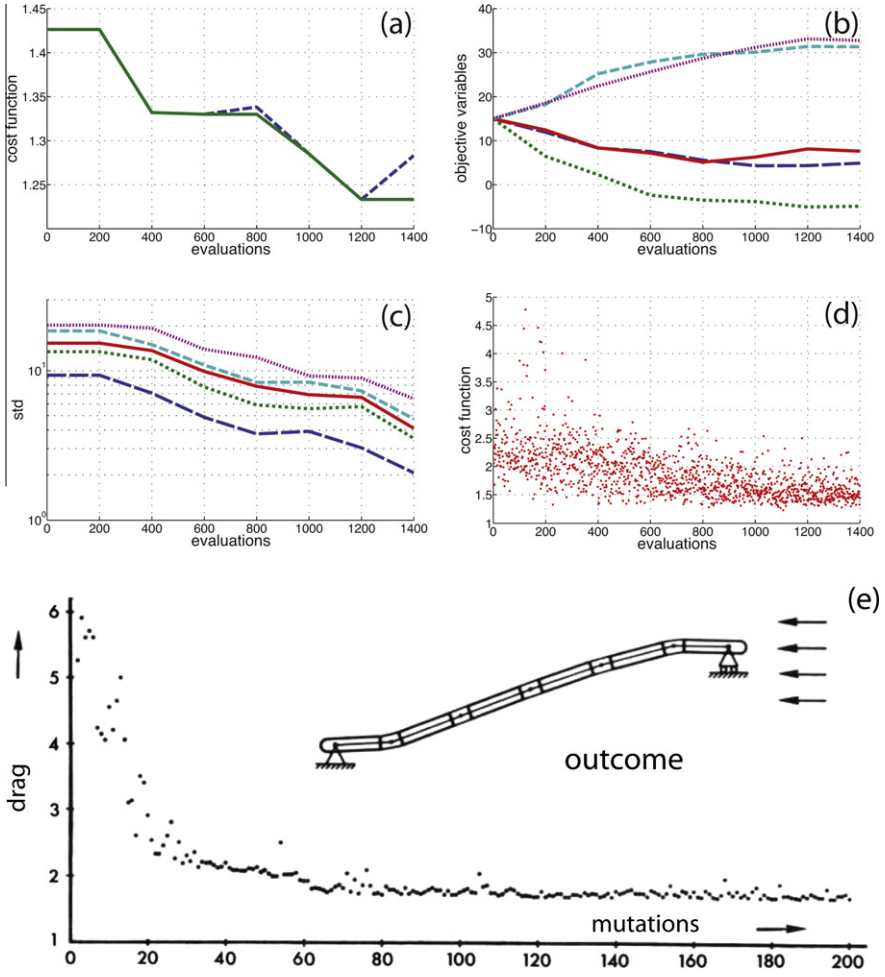

Fig. 7. Optimization history of Experiment 2 using vortex methods with Brinkman penalization: (a) cost function f, (b) parameter values (mean $\mathbf{m}$ of the multivariate distribution, blue = α_1, green = α_2, red = α_3, cyan = α_4, violet = α_5) and (c) standard deviation of the principal axes versus number of cost function evaluations. (d) and (e) Scatter plots showing the cost function value of each parameter set during the optimization process, respectively, in our simulation and in the original setup by Ingo Rechenberg [9]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

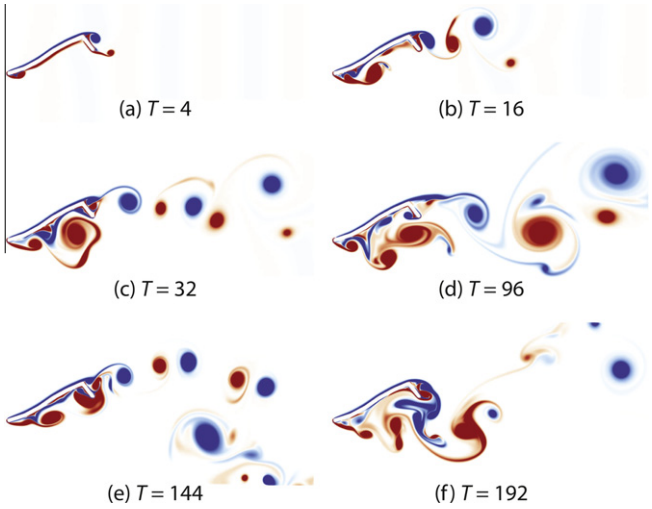

Fig. 8. Experiment 2 using vortex methods with Brinkman penalization: vorticity at different dimensionless times (T) for the worst candidate in the first generation of the optimization process.

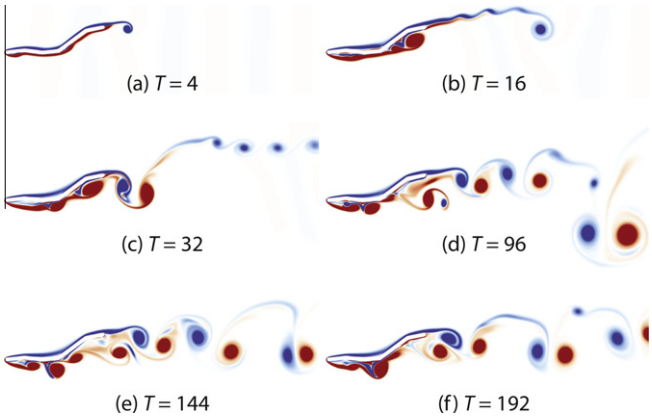

Fig. 9. Experiment 2 using vortex methods with Brinkman penalization: vorticity at different dimensionless times (T) for the best candidate ever in the first generation of the optimization process.

approximately ~8 h. This entails a total of 160,000 CPU hours. The best solution found (Table 2a,) Fig. 9) leads to a drag reduction of ~14%, with respect to the best candidate in the first generation. As in Experiment 1, the initial candidates generated are highly inefficient, presenting large frontal areas to the flow, leading to the generation of powerful vortices (Fig. 8).

The best case found presents an angle of attack close to zero for the first and last plates of the structure, as in Rechenberg's work (Fig. 7e). On the other hand, the inclination of the remaining plates is different: our solution favors a flat orientation of the second and third plate, with high angle of incidence for the fourth and fifth plate, while in Rechenberg's work the middle plates present all the same angle. The drag measured in our simulations is lower than the one reported by Rechenberg (Figs. 7d, 7e, 10). As in Experiment one, such discrepancies could be related to the differences between the experimental setup and our two-dimensional simulations and/or to the fact that the solution identified by CMA-ES (or by Rechenberg's evolutionary strategy) may not correspond to the global minimum. Nevertheless we wish to emphasize that the simulations presented herein are consistent with the results reported by the experimental studies of Rechenberg.

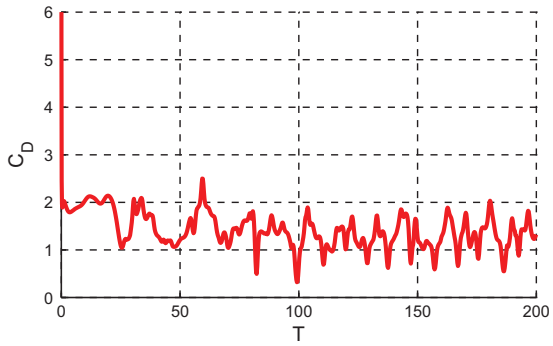

Fig. 10. Experiment 2 using vortex methods with Brinkman penalization: drag (C_D) curve against dimensionless times (T) for the best candidate ever of the optimization process.

5. Conclusions

In this work we developed a shape optimization process for drag reduction that entails a hybrid vortex particle method with Brinkman penalization, and covariance matrix adaptation evolution strategy. The process is demonstrated in shape optimization of linked flat plates for drag reduction. The present simulations are the first to repeat *in silico* the pioneering wind tunnel experiments in drag reduction using Evolution Strategies by Ingo Rechenberg. Without imposing any *a priori* knowledge, on the arrangement of the linked bodies, we show that a flat configuration (flat plate) is not a shape with a global drag minimum. This rather counterintuitive result is in agreement with experimental studies [9] and can be explained by the generation of a stable secondary vorticity over the linked bodies due to the slight corrugation in the optimal configuration. The proposed framework provides a flexible, robust tool for shape optimization problems, combining accuracy and computational efficiency. The framework is not limited to flow problems as the cost function is computed in a black box configuration and independent of the optimization process. Ongoing work, focuses on enhancing the efficiency of the flow solver by using multi-resolution vortex methods [28,29] that can adapt the size of the computational elements dynamically to resolve the flow structures.

References

[1] Pironneau O. On optimum design in fluid mechanics. J Fluid Mech 1974;64(1):97–110.
[2] Pironneau O. On optimum profiles in stokes flow. J Fluid Mech 1972;59(1):117–28.
[3] Simon J. Domain variation for drag in Stokes flow. In: Xunjing L, editor. Proceedings of the lecture notes in control and information science, vol. 1; 1990.
[4] Hansen N, Niederberger ASP, Guzzella L, Koumoutsakos P. Method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. IEEE Trans Evolut Comput 2009;13(180–197).
[5] Khadra K, Angot P, Parneix S, Caltagirone JP. Fictitious domain approach for numerical modelling of Navier–Stokes equations. Int J Numer Methods Fluids 2000;34:651–84.
[6] Liu Q, Vasilyev OV. A Brinkman penalization method for compressible flows in complex geometries. J Comput Phys 2007;227(2):946–66.
[7] Angot P, Bruneau C-H, Fabrie P. A penalization method to take into account obstacles in viscous flows. Numer Math 1999;81:497–520.
[8] Carbou G, Fabrie P. Boundary layer for a penalization method for viscous incompressible flow. Adv Differ Equat 2003;8:1453–80.
[9] Rechenberg I. Evolutionsstrategie '94. frommann-holzboog; 1994.
[10] Kern S, Koumoutsakos P. Simulations of optimized anguilliform swimming. J Exp Biol 2006;209(24):4841–57.
[11] Bueche D, Schraudolph N, Koumoutsakos P. Accelerating evolutionary algorithms with gaussian process fitness function models. IEEE Trans Syst Man Cybernet Part C 2005;32(2):183–94 (Special Issue on Knowledge Extraction and Incorporation in Evolutionary Computation).
[12] Bueche D, Stoll P, Dornberger R, Koumoutsakos P. Multi-objective evolutionary algorithm for the optimization of noisy combustion problems. IEEE Trans Syst Man Cybernet Part C 2002;32(4):460–73.

[13] Hansen N, Ostermeier A. Completely derandomized self-adaptation in evolution strategies. Evolut Comput 2001;9(2):159–95.
[14] Hansen N, Muller SD, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolut Comput 2003;11(1):1–18.
[15] Hansen N, Kern S. Evaluating the CMA evolution strategy on multimodal test functions. Parallel Problem Solving from Nature – PPSN VIII 2004;3242:282–91.
[16] Koumoutsakos P. Inviscid axisymmetrization of an elliptical vortex. J Comput Phys 1997;138(2):821–57.
[17] Hieber SE, Walther JH, Koumoutsakos P. Remeshed smoothed particle hydrodynamics simulation of the mechanical behavior of human organs. Technol Health Care 2004;12(4):305–14.
[18] Cottet G-H, Koumoutsakos P. Vortex Methods, Theory and Practice. Cambridge University Press; 2000.
[19] Cottet G-H. Artificial viscosity models for vortex and particle methods. J Comput Phys 1996;127(2):299–308.
[20] Coquerelle M, Cottet GH. A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. J Comput Phys 2008;227(21):9121–37.
[21] Koumoutsakos P, Leonard A. High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. J Fluid Mech 1995;296:1–38.
[22] Koumoutsakos P. Multiscale flow simulations using particles. Ann Rev Fluid Mech 2005;37(457–487).
[23] Chatelain Philippe, Koumoutsakos Petros. A fourier-based elliptic solver for vortical flows with periodic and unbounded directions. J Comput Phys 2010;229(7):2425–31.
[24] Brinkman H. A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles. Appl Sci Res Sect A – Mech Heat Chem Eng Math DS 1947:27–34.
[25] Angot P. Analysis of singular perturbations on the brinkman problem for fictitious domain models of viscous flows. Math Methods Appl Sci 1999;22:1395–412.
[26] Rossinelli Diego, Bergdorf Michael, Cottet Georges-Henri, Koumoutsakos Petros. GPU accelerated simulations of bluff body flows using vortex particle methods. J Comput Phys 2010;229(9):3316–33.
[27] Muller SD, Hansen N, Koumoutsakos P. Increasing the serial and the parallel performance of the CMA-evolution strategy with large populations. Parallel Problem Solving from Nature – PPSN VII 2002:422–31.
[28] Bergdorf M, Cottet GH, Koumoutsakos P. Multilevel adaptive particle methods for convection–diffusion equations. Multiscale Model Simulat 2005;4(1):328–57.
[29] Bergdorf M, Koumoutsakos P. A lagrangian particle-wavelet method. SIAM J Multiscale Model Simulat 2006;5(3):980–95.