Application of machine learning algorithms to flow modeling and optimization

By S. Müller [1], M. Milano [1] AND P. Koumoutsakos

1. Motivation and objectives

We develop flow modeling and optimization techniques using biologically inspired algorithms such as neural networks and evolution strategies. The applications presented herein encompass a variety of problems such as cylinder drag minimization, neural net modeling of the near wall structures, enhanced jet mixing, and parameter optimization in turbine blade film cooling. The unifying concept is the utilization of automated processes for the solution of these problems, devised from machine learning algorithms.

The results presented herein encompass a wide variety of problems such as drag minimization, neural net modeling of the near wall structures, enhanced jet mixing, and parameter optimization in turbine blade film cooling.

Flow control has been a fundamental concept in fluid mechanics research in this century. In the early 1900's research was focused on the development of experimental procedures that would elucidate the governing flow phenomena in order to devise efficient control devices. A number of empirical methods were proposed such as rotating cylindrical sails for ship propulsion and the placement of wires around wing profiles (a precursor of riblets) for drag minimization. In the second half of the century, developments such as the discovery of coherent structures in the wake of bluff body flows and the understanding of the processes of flow separation led to a number of devices (e.g vortex generators, splitter plates, mass transpiration, etc.) for the efficient manipulation of flow structures in experiments and realistic engineering configurations. In the 80's and 90's the advent of Direct Numerical Simulations (DNS) provided us with a thorough understanding of fundamental processes such as the mechanisms of skin friction drag in turbulent flows (Kim *et al.*, 1987). DNS of turbulent flows have been used as the testing grounds for a number of control algorithms such as the opposition control scheme (Choi *et al.*, 1984), feedback control using models derived via POD (Lumley *et al.*, 1998) or neural networks (Lee *et al.*, 1997), the feedback control of vorticity generation (Koumoutsakos, 1997 and 1999), and optimal (Bewley, 1999) and suboptimal control (Lee *et al.*, 1998) strategies. These simulations have provided us with valuable insight into the behavior of controlled flows. Moreover, these algorithms have demonstrated that the effective control of turbulence in engineering applications requires strategically placed, micro/nano devices that would be capable of sensing and actuating frequencies in the order of a few MHz.

1 Institute for Fluid Dynamics, Swiss Federal Institute of Technology, ETH Zentrum, CH-8092 Zürich, Switzerland.

Recent progress in manufacturing has provided us with an unprecedented array of such potent devices such as MEMS (Ho *et al.*, 1996) capable of sensing and effecting the detailed structure of a turbulent flow. However, the proper integration of control devices and control algorithms in realistic applications remains a challenging problem. While a wealth of different possible geometry modifications and/or open loop actions have been proposed, little effort has focused in trying to devise a concise optimization algorithm to adjust the diverse control parameters. The development of low order models that can be used in conjunction with control theories is a challenging problem that, when solved, could lead to drastically improved designs.

We envision neural network approaches as an effective way of developing such models and incorporating them in feedback control algorithms. In section 2 we present some preliminary results from the application of neural networks as a method to construct low order models, describing the near wall dynamics in turbulent flows. Neural networks are viewed as a general procedure of model formation encompassing schemes such as the Proper Orthogonal Decomposition.

Another key issue in the effort to reduce time to market of new engineering designs is the optimization of the design parameters in an efficient manner. The design cycle usually involves multi-objective and multi-disciplinary optimization problems, requiring the iterative solution of empirical formulas, the appropriate integration of numerical simulations, and the incorporation of physical understanding of the various aspects of the problem. At the same time, the optimization cycle of the physical problem must take into consideration financial and manufacturing constraints. In flow related problems, this optimization cycle has benefited from advances in optimization theory which usually aim at tackling the most costly aspects of the optimization problem such as the solution of the Navier-Stokes equations. Powerful techniques such as the adjoint procedure have been implemented successfully in the design cycle of aircrafts (Reuther *et al.*, 1999).

However, such optimization strategies are usually based on the efficient calculation of gradients of functions relating the quantity to be optimized to the parameters of the problem. Such gradients are not always readily available as often the optimization cycle would involve empirical formulas and cost functions that are difficult to express analytically in terms of the optimization problem. Moreover, gradient based algorithms are usually converging to local extrema. Therefore, the result strongly depends on the initial selection of parameters.

Evolution strategies (Rechenberg, 1973) are optimization techniques that avoid the problems associated with the use of gradients as they require only the calculation of the cost function at each point in the parameter space. They operate based on natural principles of evolution such as mutation, recombination, and selection. These operations are adapted so that the algorithm automatically develops and attempts to optimize a *model landscape* relating the cost function to its parameters. Compared with gradient based techniques, their convergence rate is usually much lower, thus requiring large numbers of iterations that could be unrealistic for some problems of engineering interest. On the other hand, they are highly parallel algorithms that efficiently exploit today's powerful parallel computer architectures and they are more likely than gradient based algorithms to identify a global optimum. This latter aspect makes them attractive in many engineering applications where the fitness landscape cannot be assumed unimodal. In the next section we report results from the application of evolutionary algorithms to a number of diverse areas such as drag minimization in cylinder flows, jet mixing optimization, and turbine blade film cooling.

2. Accomplishments

2.1 Neural networks as near wall flow models

Feedback control algorithms can be devised using a low order model representation of the flow field. In this context we examine the reconstruction of the near wall structures using wall only information with the aid of linear and nonlinear neural nets.

The POD as a linear neural network

A *model reduction* can be accomplished by projecting the model equations, i.e. the Navier-Stokes equations, on a properly selected lower dimensional phase subspace. A reasonable choice for a "proper" selection criterion for the base of this manifold is the maximization of the energy content of the projection. This can be done by applying the Karhunen-Loeve decomposition to a data set that is representative of the dynamics of the system that we wish to approximate. This operation is called *Proper Orthogonal Decomposition* (POD) (Berkooz *et al.*, 1993), or *Linear Principal Components Analysis* (PCA).

The linear POD is an approximation of the flow vector v by a finite expansion of orthonormal functions ϕ_n such that: $v = V + \sum_{i=1}^{n} a_n(t)\phi_n(x)$. where V is the time averaged flow, ϕ_n is the set of the first n eigenvectors of the covariance matrix $C = E[(v_i - V)(v_j - V)]$; when this representation for v is substituted in the Navier Stokes equations, the original PDE model is transformed in an ODE model, composed by n equations.

The POD can be expressed as a multi-layer feed-forward neural network. Such a network is defined by the number of layers, the specification of the output function for the neurons in each layer, and the weight matrices for each layer. Baldi and Hornik Baldi *et al.*, 1989) have shown that training a linear neural network structure to perform an identity mapping on a set of vectors is equivalent to obtaining the POD of this set of vectors. A neural network performing the linear POD can be specified as a 2 layer linear network:

$$x = W_1 v; \quad \hat{v} = W_2 x$$

where $\hat{v}$ is the reconstructed field, v is the original flow field, having N components, x is the reduced order representation of the field, having n components, and W_1 and W_2 are the network weight matrices, of sizes $N \cdot n$ and $n \cdot N$ respectively. Non-linearity can be introduced by a simple extension to this basic network:

$$x = W_2 tanh(W_1 v); \quad \hat{v} = W_4 tanh(W_3 x)$$

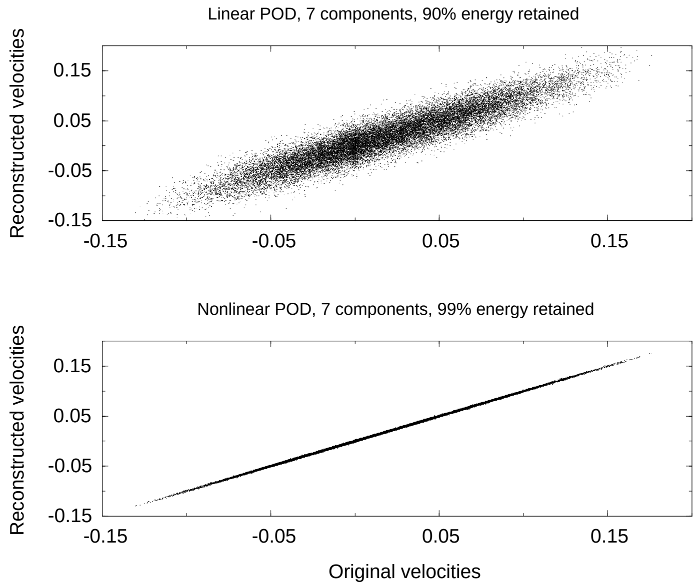

Fig. 1. Reconstruction of the velocity field in Burger's equation, using POD (top) and Neural Networks (bottom).

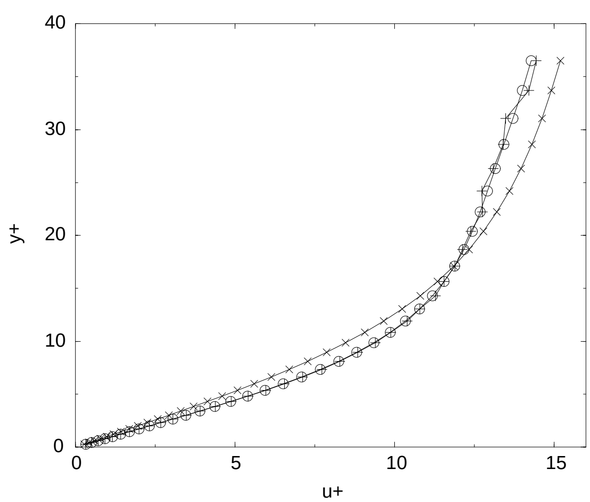

Fig. 2. Streamwise and spanwise averaged profile. o : original, x : reconstruction using linear POD, + : reconstruction using NN

This corresponds to a neural network model with 4 layers: the first one, with an $m \cdot N$ weight matrix W_1, nonlinear; the second one, with an $n \cdot m$ weight matrix W_2, linear; the third one, also nonlinear, with an $m \cdot n$ weight matrix W_3, and the last one, linear with an $N \cdot m$ weight matrix W_4. However, the resulting system of ODEs is more involved as compared to the one resulting from the application of the linear POD.

A simple comparison of POD and nonlinear NN is provided by the reconstruction of the velocity field in the stochastically forced Burger's equation — a classical 1D model for turbulent flow (Chambers *et al.*, 1998). The linear POD was used to obtain a set of 256 linear eigenfunctions using 10000 snapshots extracted from a simulation. Using the first 7 eigenfunctions it is possible to reconstruct the original flow field, keeping the 90 percent of the energy. A nonlinear neural network was trained on the same data set to perform the identity mapping: this network is composed by 256 inputs and 4 layers having respectively 64 nonlinear neurons, 7 linear neurons, 64 nonlinear neurons, and 256 linear neurons. For validation purposes, a data set of 1000 snapshots, not used in the training phase, was used. In Fig. 1 it is possible to appreciate the reconstruction performances of both the approaches; the proposed nonlinear NN clearly outperforms the linear POD.

We have also used neural networks to reconstruct the near wall region of a turbulent flow. By making a Taylor series expansion of the velocity field for small distances above the wall, one can reconstruct a second order model of the velocity field using wall only information such as the shear stresses and the pressure at the wall. For the streamwise velocity field we have that:

$$u(\mathbf{x}, t) = \omega_z^w y + \frac{Re}{2} \frac{\partial P^w}{\partial x} y^2 + \mathcal{O}(y^3)$$

where $\omega_z^w = S^w$ are the shear stresses, P^w is the wall pressure. This model has been found to be accurate only up to $y^+ = 10$ for a wide range or Reynolds numbers. It can be improved by using a neural network to approximate the higher order terms as functions of wall quantities:

$$u(\mathbf{x}, t) = \omega_z^w y + \frac{Re}{2} \frac{\partial P^w}{\partial x} y^2 + \mathbf{M}(P^w, S^w)$$

where $\mathbf{M}(P^w, S^w)$ denotes a linear (i.e. a POD) or nonlinear neural network model using as input information the wall pressure and shear stresses.

This model has been applied to a turbulent channel flow with $Re_\tau = 250$, based on the channel half height. In Fig. 2 we present a snapshot of the streamwise and spanwise averaged u^+ profile, comparing a POD reconstruction and a nonlinear model reconstruction, using 2 neurons in the inner layer, 1280 inputs containing the shear stresses and wall pressures measured on the bottom of a minimal flow unit, and 49920 output neurons carrying the estimated velocities. The snapshot in Fig. 2 results from a simulation using samples that the neural network did not use in the training phase, thus showing the good generalization of this model.

2.2 Evolutionary optimization algorithms

Neural networks are machine learning algorithms that attempt to generate low order models from existing data in an automatic fashion. Other constituents of machine learning algorithms are evolution strategies and genetic algorithms. Evolution strategies (ES) and Genetic Algorithms (GA) operate on a population with a number of individuals, each of them represented by a real or binary vector. For an optimization problem with n parameters, each vector comprises n elements. Three operators are defined to modify the population members: *I. Recombination/crossover*, that generates new trial solution points (offsprings), using some elements drawn from the population; *II. Mutation*, that randomly changes some of the offsprings components; *III. Selection*, that chooses the population elements that will be used by the crossover. For each population element a *fitness function* is defined, measuring in a quantitative way how close a given solution is to the desired goal. Based on their fitness, the old population members are compared with the newly generated ones, and the solutions with the better fitness constitute the new population members. In this way, iterating the selection-crossover-mutation process, the population evolves towards the optimal solution.

We have developed self-organizing genetic algorithms particularly suitable for finding clusters of good solutions (Milano and Koumoutsakos, 1999) — a desirable scheme when seeking non-sharp, non-single point optima. A variable mutation operator, depending on the local fitness value and on the global success history of the population, allows the population to avoid local optima. The algorithm operates in a hierarchical fashion by identifying well correlated clusters of population, leading to optimization schemes employing few strategically placed actuators.

In our research efforts we have been concerned with the convergence rate of evolution strategies. A crucial parameter is the adaptation of the step size of the evolution strategy as, in effect, this reflects the properties of the environment to the parameter population. A powerful control scheme for step size adaptation is the covariance matrix adaptation (CMA) (Hansen and Ostermeier, 1996). With this method, the step sizes are adapted using prior information. The adaptation of the mutation distribution with the CMA is independent of the coordinate system and — in combination with the adaptation of the global step size — yields a high convergence rate.

A further speed-up is achieved by combining the CMA-ES with an intermediate recombination that averages the variable vector elements of some of the parents. This combined method is called $(\mu/\mu_I, \lambda)$-CMA-ES where μ/μ_I denotes the recombination of μ_I out of μ parents.

In the $(\mu/\mu, \lambda)$-CMA-ES, the parameter vectors $\mathbf{x}_k^{(k+1)}, k = 1, ..., \lambda$ in generation $g+1$ are computed by

$$\mathbf{x}_k^{(g+1)} = \langle \mathbf{x} \rangle_\mu^{(g)} + \delta^{(g)} \mathbf{B}^{(g)} \mathbf{D}^{(g)} \mathbf{z}_k,$$

where the center of mass of the selected individuals is given by

$$\langle \mathbf{x} \rangle_\mu^{(g)} = \frac{1}{\mu} \Sigma_{j=1}^{\mu} \mathbf{x}_j^{(g)}.$$

$\mathbf{B}$ and $\mathbf{D}$ are computed from the covariance matrix $\mathbf{C}$ such that the eigenvectors of $\mathbf{C}$ become the columns of $\mathbf{B}$ and such that the square roots of the eigenvalues of $\mathbf{C}$ become the diagonal elements of the diagonal matrix $\mathbf{D}$, mathematically expressed by

$$\mathbf{C}^{(g)} = \mathbf{B}^{(g)} \mathbf{D}^{(g)} (\mathbf{B}^{(g)} \mathbf{D}^{(g)})^{\mathrm{T}}.$$

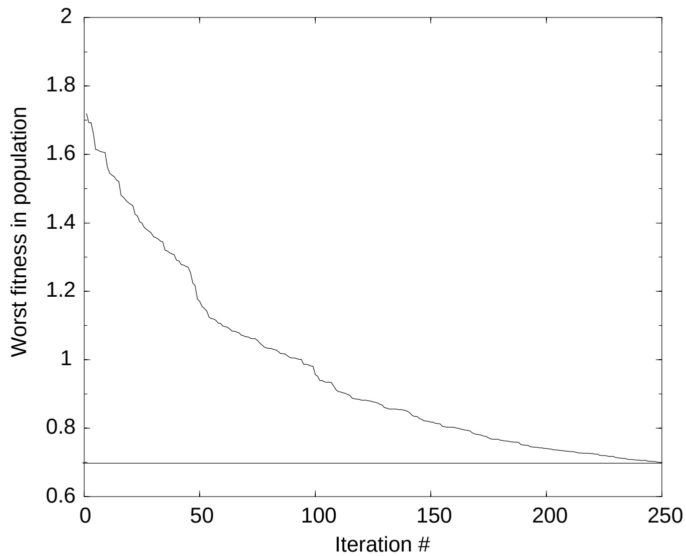

Fig. 3. Worst fitness/drag in the population as a function of the optimization iterations. The horizontal line is the desired target drag coefficient.

The covariance matrix is based on the evolution path, **s**, by the following scheme:

$$\mathbf{s}^{(g+1)} = (1-c)\cdot \mathbf{s}^{(g)} + c_u \cdot \frac{\sqrt{\mu}}{\delta^{(g)}}\left(\langle \mathbf{x}\rangle_\mu^{(g+1)} - \langle \mathbf{x}\rangle_\mu^{(g)}\right)$$

$$\mathbf{C}^{(g+1)} = (1-c_{cov})\cdot \mathbf{C}^{(g)} + c_{cov}\cdot \mathbf{s}^{(g+1)}(\mathbf{s}^{(g+1)})^{\mathrm{T}},$$

where $1/c$ represents the accumulation time for the evolution path and where $1/c_{cov}$ represents the averaging time for the covariance matrix. The accumulation time parameter, c, can be written in normalized form as $c_u = \sqrt{c(2-c)}$.

2.3 Optimization results

These evolutionary optimization algorithms have been applied to a number of optimization problems with sufficient success. The self-organizing genetic algorithm was applied in order to optimize the parameters of a prototypical cylindrical configuration (Milano and Koumoutsakos, 1999). In this configuration the surface of the cylinder is subdivided into 16 equal segments that are allowed to move independently tangentially to the surface of the body. Using a hierarchical clustering approach, the evolutionary algorithm was able to identify automatically the critical points of the flow (near the uncontrolled flow separation points) while resulting in about 50% drag reduction for two-dimensional flow at Re=250. In Fig. 3 we show the convergence of the algorithm as a function of the number of iterations.

The CMA-ES strategy was applied to the parameter optimization of scalar mixing in DNS of jet flow at Re=1500 (Hilgers *et al.*, 1999, Koumoutsakos *et al.*, 1999). This is a challenging problem for evolution strategies as each iteration requires lengthy computations. Using various cost functions it was possible to identify previously unknown effective parameters that induce various types of behavior to the jet (Figs. 4, 5, 6).

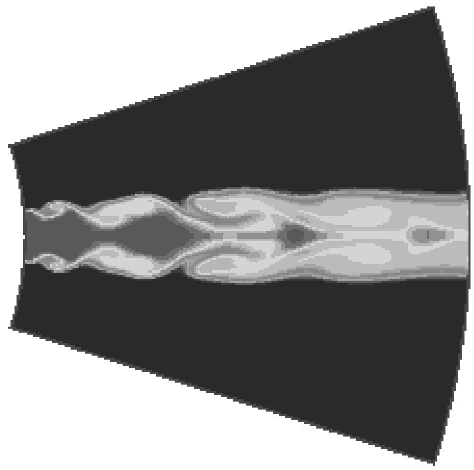

Fig. 4. Contours of scalar concentration in jet flow at Re = 1500. No control.

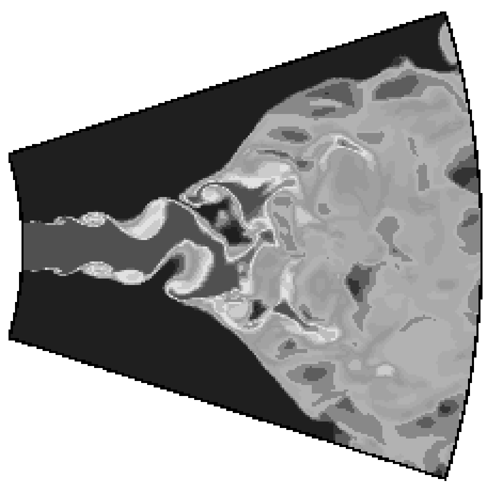

Fig. 5. Controlled flow - bifurcating jet.

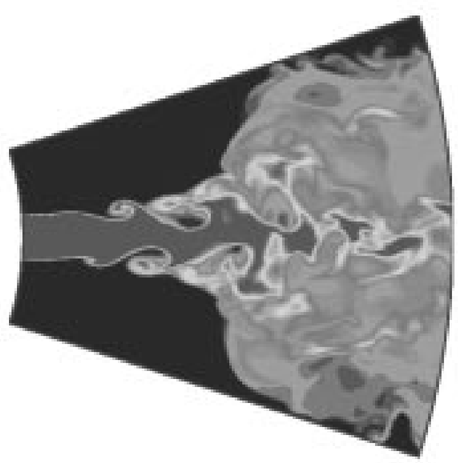

Fig. 6. Controlled flow - flapping jet.

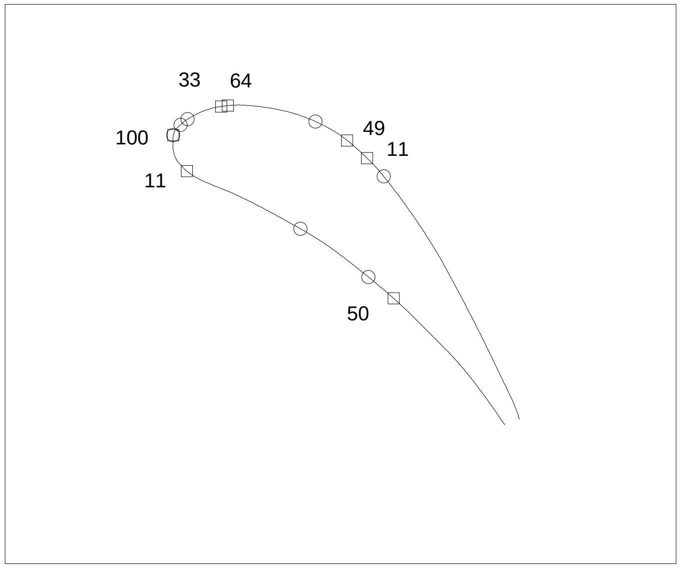

Fig. 7. Initial position of rows: ○ . Final position of rows (plus number of holes per row): □ .

Furthermore, the CMA-ES algorithm was implemented in the realistic design cycle of a gas turbine blade film cooling (Müller and Koumoutsakos, 1999). This optimization cycle involved the optimization of parameters used in empirical algebraic formulas along with a heat conduction simulation program for the film cooling problem. The optimization goals were to reduce the coolant mass flow and at the same time achieve a homogeneous surface temperature while observing certain constraints in the range of the temperature distribution. Starting from a number of initial configurations, the optimization algorithm was always able to produce a number of improved designs, improving initial engineering estimates up to 25%. It should be emphasized that, besides classical designs requiring large numbers of cooling rows in the leading edge of the blade, the optimization algorithm was also able to identify novel designs involving certain ratios and placement of a number of rows on the two sides of the blade (Fig. 7).

3. Future plans

Our near future research is directed to the development of low order models using neural networks and their integration in feedback control algorithms. Evolutionary algorithms are being further developed aiming at the production of a multidisciplinary optimization tool.

Applications of interest involve the optimization of control parameters in jet flows and three-dimensional bluff-body flows as computed using LES formulations.

REFERENCES

Baldi, P. & Hornik, K. 1989 Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks.* **2,** 53-58.

Berkooz, G., Holmes, P. & Lumley, J. L. 1993 The proper orthogonal decomposition in the analysis of turbulent flows. *Ann. Rev. Fluid Mech.* **25,** 539-575.

Bewley, T. R. 1999 Optimal and robust control and estimation of transition and turbulence. *Ph.D. dissertation,* Stanford University.

Chambers, D. H., Adrian R. J., Moin, P. & Stewart, S. 1988 Karhunen-Loeve expansion of Burgers' model of turbulence. *Phys Fluids.* **31,** 2573-2582.

Choi, H., Moin, P. & Kim, J. 1994 Active turbulence control for drag reduction in wall bounded flows. *J. Fluid Mech.* **262,** 75-110.

Hansen, N. & Ostermeier, A. 1996 Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. *Proceedings of the IEEE Intl. Conf. on Evolutionary Computation (ICEC'96).* 312-317.

Hilgers, A. & Müller, S.D., Boersma, B.J., Koumoutsakos, P.D. 1999 Optimization of Turbulent Jet Mixing using Evolution Strategies. *Phys. Fluids.* In preparation.

Ho, C. & Tai, Y. 1996 Review: MEMS and its applications for flow control. *J. of Fluids Engr.* **118,** 437-447.

Kim, J., Moin, P. & Moser, R. 1987 Turbulence statistics in fully developed channel flow at low Reynolds number. *J. Fluid Mech.* **177,** 133-166.

Koumoutsakos, P. 1999 Vorticity flux control for a turbulent channel flow. *Phys. Fluids.* **11**(2)**,** 248-250.

Koumoutsakos, P., Müller, S., Hilgers, A., Freund, J. & Parekh, D. 1999 Evolution strategies for jet mixing optimization. *IEEE Trans. on Evolutionary Comp.* Submitted.

Lee, C., Kim, J., Babcock, D. & Goodman, R. 1997 Application of neural networks to turbulence control for drag reduction. *Phys Fluids.* **9,** 1740-1747.

Lee, C., Kim, J. & Choi, H. 1998 Suboptimal control of turbulent channel flow for drag reduction. *J. Fluid Mech.* **358,** 245-258.

Lumley, J. & Blossey, P. 1998 Control of turbulence. *Ann. Rev. Fluid Mech.* **30,** 311-327.

Milano, M. & Koumoutsakos, P. 1999 A Self-organizing genetic algorithm for cylinder drag minimization. *Phys Fluids.* Submitted.

Müller, S. & Koumoutsakos, P. 1999 Evolution strategies for film cooling optimization. *AIAA J.* Submitted.

Rechenberg, I. 1973 *Evolutions strategie: Optimierung technischer systeme nach prinzipien der biologischen evolution.* Fromann-Holzboog, Stuttgart.

Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. L. & Saunders, D. 1999 Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 1. *J. Aircraft.* **39**(1)**,** 51-60.