

Low-Cost Optimization for Fast-Microfluidic-Mixer Shape Design

Benjamin Ivorra, Bijan Mohammadi
Mathematics and Modelling Institute
Montpellier University, 34095 Montpellier, France

Abstract - We give an unified formulation for deterministic and stochastic global optimization algorithms via initial and boundary value problems for different first and second order dynamic systems. The ensemble is applied to the design of a fast-micro-mixer. The aim is to optimize a given mixer shape in order to reduce its mixing time. The algorithms are compared from their complexity and optimization efficiency. In order to reduce computational time approximate state and sensitivity evaluations are introduced during optimization.

Keywords: Shape optimization, Global optimization, Dynamical systems, Boundary value problem, Microfluidic mixers.

Acknowledgement: The work was done/partially done while the author was visiting the Institute for Mathematical Sciences, National University of Singapore in 2004. The visit was supported by the Institute.

1 Introduction

Microfluidic channel systems used in bio-analytical applications are fabricated using technologies derived from microelectronics industry including lithography, wet etching and bonding of substrates. Industrial applications of these techniques concern DNA sequencing, new drug molecules trials, pollution detection in water or food and protein folding [1].

Focusing on this last domain, important structural events occur on a microsecond time scale [2]. To study their kinetics, folding reactions must be initiated at even shorter timescales. This for instance using photochemical initiation [3] and changes in temperature [4], pressure [5, 6] or chemical potential, as in salt or chemical denaturant concentration changes [7, 8, 9]. All these techniques provide the perturbation of protein conformational equilibrium necessary to initiate folding. In comparison to temperature- and pressure-jump relaxation techniques, folding experiments based on changes in chemical potential, via rapid mixing of protein solutions into and out of chaotrope solvents, are more versatile. The technique is applicable to a wide range of proteins as most unfold reversibly in the presence of chemical denaturants such as urea and guanidine hydrochloride (GdCl) [7].

Until recently, the main limitation of mixer-based experiments was their inability to access very short timescales. Mixing of chemical species is ultimately limited by the time required for molecular diffusion across a finite length scale, and diffusion time scales as the square of diffusion length. Brody et al. [10] first proposed rapid mixers based on hydrodynamic focusing as a way to address the issue of reducing diffusion lengths under laminar flow conditions while minimizing sample consumption. Hydrodynamic focusing has been used to measure protein and RNA folding [11], with mixing times of a few hundreds of microseconds.

This paper discusses specific shape optimization for a new microfluidic mixer based on a continuous flow principle by Knight et al. [12] which leverages hydrodynamic focusing on the micron scale to reduce diffusion lengths [1]. This mixer uses about eight orders of magnitude less labelled protein sample mass flow than a previously reported ultra-fast protein folding mixer [13], with flow rates of 3 nl/s and protein concentrations of tens of nanomolar.

This problem is multi-model in the sense that several PDEs are involved in the definition of the state variables and the cost function. In particular, a transport-diffusion equation is used to model a passive scalar in a flow field simulated by the incompressible Navier-Stokes equations. Simulation of these equations being computation intensive, we would like to use low complexity approach in sensitivity or intermediate state calculations together with various global optimization tools. Previous application of control theory to the design of microfluidic devices have been reported [14, 15, 16, 17].

Section 2 presents the three global optimization algorithms with associated mathematical background. In Section 3 low-cost sensitivity approaches are given. Section 4 introduces a short presentation of the considered problem physics and its mathematical modelling. Finally Section 5 shows and compares optimization results.

2 Global optimization methods

We present three minimization methods: A typical genetic algorithm [18, 19], a new global optimization method based on the solution of boundary value problems, and an hybrid algorithm using ingredients from the two previous approaches.

2.1 Genetic algorithm

Consider the minimization of a real functional $J(x)$, $x \in \Omega_{ad}$, x is the optimization parameter and belongs to an admissible space Ω_{ad} of dimension n_{dim} . Genetic algorithms approximate the global minimum (or maximum) of J , called the fitness function, through a stochastic process based on an analogy with the Darwinian evolution of species [19]:

A first family $X^1 = \{(x)_i^1, i = 1, \dots, N_p\}$ of N_p possible solutions of the optimization problem, called 'individuals', is randomly generated in the search space Ω_{ad} . Starting from this population X^1 we build recursively N_{gen} new populations $X^i, i = 1, \dots, N_{gen}$ through three stochastic steps:

We write X^i using the following (N_p, n_{dim}) -matrix form:

$$X^i = \begin{bmatrix} x_1^i(1) & \dots & x_1^i(n_{dim}) \\ \vdots & \ddots & \vdots \\ x_{N_p}^i(1) & \dots & x_{N_p}^i(n_{dim}) \end{bmatrix} \quad (1)$$

At each iteration the following three steps are performed.

- **Selection:** each 'individual', representing a potential solution of the problem, is ranked with respect to its 'fitness' function. In this process, better individuals have higher chances to be chosen. They are called 'parents'.

Introducing \mathcal{S} a binary (N_p, N_p) -matrix with, for each line i , a value 1 on the j th row when the j th individual has been selected and 0 elsewhere

$$X^{n+1/3} = \mathcal{S}X^n \quad (2)$$

- **Crossover:** this process leads to a data exchange between two 'parents' and apparition of new individuals called 'child'.

- We introduce \mathcal{C} a real-valued (N_p, N_p) -matrix where for each couple of consecutive lines $(2i - 1, 2i)$ ($1 \leq i \leq \frac{N_p}{2}$), the coefficients of the l -th and k -th row are given by a 2×2 matrix of the form

$$\begin{bmatrix} \lambda_1 & 1 - \lambda_1 \\ \lambda_2 & 1 - \lambda_2 \end{bmatrix} \quad (3)$$

In this expression, $\lambda_1 = \lambda_2 = 1$ if no crossover is applied on the selected 'parents' l and k and are randomly chosen with a probability p_c in $[0, 1]$ in the other case.

This step can be summarized as:

$$X^{n+2/3} = \mathcal{C}X^{n+1/3} \quad (4)$$

- **Mutation:** This process leads to new parameter values. For each 'child', we determine with a fixed probability p_m if their parameters should mute.

Introducing following families:

$$\mathcal{M}_+ = \{\mathcal{M}_{+i} \ i = 1, \dots, N_p\} \text{ and } \mathcal{M}_- = \{\mathcal{M}_{-j} \ j = 1, \dots, n_{dim}\} \text{ and } \{(\varepsilon_{i,j}) \ i = 1, \dots, N_p \ j = 1, \dots, n_{dim} / \varepsilon_{i,j} \in \mathbb{R}\}$$

with \mathcal{M}_{+i} a binary (N_p, N_p) -matrix which keep unchanged the i^{th} line of the matrix $X^{n+2/3}$ and set to zero the other lines. \mathcal{M}_{+j} a (n_{dim}, n_{dim}) -matrix which keep unchanged the j^{th} column of matrix $X^{n+2/3}$ and set to zero the other columns. $\varepsilon_{i,j}$ is equal to 1 if no mutation is applied and to (mutated value / (i, j) coefficient) otherwise.

In the same way, this step can be summarized as:

$$X^{n+1} = \sum_{i=1}^{N_p} \sum_{j=1}^{n_{dim}} \varepsilon_{i,j} \mathcal{M}_{+i} X^{n+2/3} \mathcal{M}_{-j} \quad (5)$$

Therefore, genetics algorithms can be seen as discrete dynamic systems:

$$X^{n+1} = \sum_{i=1}^{N_p} \sum_{j=1}^{n_{dim}} \varepsilon_{i,j} \mathcal{M}_{+i}(\mathcal{C}S X^n) \mathcal{M}_{-j} \quad (6)$$

this can be formally rewritten as:

$$X^{n+1} - X^n = \Lambda_2^n X^n \Lambda_3^n - X^n$$

which is a particular discretization of a set of nonlinear 1st order ODEs [20]:

$$\dot{X} = \Lambda_2(X) X \Lambda_3(X) - X \quad (7)$$

where the construction of Λ_i has been described above.

With these three basic evolution processes, it is generally observed that the best obtained individual is getting closer after each generation to the optimal solution of the problem [19, 18].

Engineers like GAs because these algorithms do not require sensitivity computation, perform global and multi-objective optimization and are easy to parallelize. Their drawbacks remain their weak mathematical background, their computational complexity and their lack of accuracy. The semi-deterministic algorithm algorithm (SDA) below aims to address some of these issues.

2.2 Semi-deterministic multi-level optimization

Most deterministic minimization algorithms, which perform the minimization of a function $J : \Omega_{ad} \rightarrow \mathbb{R}$, can be seen as discretizations of the following dynamical system [21, 22, 23] where x denotes the vector of control parameters belonging to an admissible space Ω_{ad} . ζ is a fictitious parameter. M is a local metric transformation and d a direction in Ω_{ad} .

$$\begin{cases} M(\zeta)x_\zeta = -d(x(\zeta)) \\ x(\zeta = 0) = x_0 \end{cases} \quad (8)$$

For example if $d = \nabla J$, the gradient of the functional, and $M = Id$, the identity operator, we recover the classical steepest descent method while with $d = \nabla J$ and $M(\zeta) = \nabla^2 J(x(\zeta))$ the Hessian of J , we recover the Newton method. Quasi-Newton methods can also be recovered using approximate Hessian definition [24].

Theoretical background of the method requires the following assumptions [20]:

H1: $J \in C^2(\Omega_{ad}, \mathbb{R})$.

H2: the infimum J_m is known. This is often the case in industrial applications.

H3: the problem is admissible: the infimum is reached inside the admissible domain: $\exists x_m \in \Omega_{ad}, s.t. J(x_m) = J_m$.

H4: J is coercive (i.e. $J(x) \rightarrow \infty$ when $|x| \rightarrow \infty$).

We consider that system (8) has a solution if for a given $x_0 \in \Omega_{ad}$, we can find a finite Z_{x_0} such that $J(x(Z_{x_0})) = J_m$:

$$\begin{cases} M(\zeta)x_\zeta = -d(x(\zeta)) \\ x(0) = x_0 \\ J(x(Z_{x_0})) = J_m \end{cases} \quad (9)$$

This is an over-determined boundary value problem which can be solved using classical techniques for BVPs (e.g. shooting, finite differences,...). Because we are interested in constrained global optimization we prefer to express the condition at Z_{x_0} on the functional instead of its gradient. Indeed, in our context a first order optimality condition is usually not satisfied at the infimum.

This over-determination is an explanation of why we should not solve global optimization problems with methods which are particular discretizations of first order differential systems. We could use variants of classical methods after adding second order derivatives [21]:

$$\begin{cases} \eta x_{\zeta\zeta} + M(\zeta)x_\zeta = -d(x(\zeta)), \\ x(0) = x_0, \quad \dot{x}(0) = \dot{x}_0, \\ J(x(Z_{x_0})) = J_m \end{cases} \quad (10)$$

To avoid introducing too much perturbation in the method, we consider, in practice, $|\eta| \ll 1$.

The over determination can be removed, for instance, by considering $x_0 = v$ for (8) (resp. $\dot{x}(0) = v$ for (10)) as a new variable to be found by the minimization of $h(v) = J(x_v(Z_v)) - J_m$, where $x_v(Z_v)$ is the solution of (8) (resp. (10)) found at $\zeta = Z_v$ starting from v .

The algorithm $A_1(v_1, v_2)$ reads:

- $(v_1, v_2) \in \Omega_{ad} \times \Omega_{ad}$ given
- Find $v \in \operatorname{argmin}_{w \in \mathcal{O}(v_2)} h(w)$ where $\mathcal{O}(v_2) = \{t\overrightarrow{v_1 v_2}, t \in \mathbb{R}\} \cap \Omega_{ad}$
- return v

The line search minimization might fail. For instance, a secant method degenerates on plateau and critical points. In this case, we add an external level to the algorithm A_1 , keeping v_1 unchanged, and looking for v_2 by minimizing a new functional h^2 defined by $h^2(v') = h(A_1(v_1, v'))$.

This leads to the following two-level algorithm $A_2(v_1, v_2^2)$:

- $(v_1, v_2^2) \in \Omega_{ad} \times \Omega_{ad}$ given
- Find $v' \in \operatorname{argmin}_{w \in \mathcal{O}(v_2^2)} h^2(w)$ where $\mathcal{O}(v_2^2) = \{\overrightarrow{t v_1 v_2^2}, t \in \mathbb{R}\} \cap \Omega_{ad}$
- return v'

The choice of initial conditions in this algorithm contains the non-deterministic feature of the algorithm. The construction can be pursued building

recursively $h^i(v_2^i) = \min_{v_2^i \in \Omega_{ad}} h^{i-1}(A_{i-1}(v_1, v_2^i))$, with $h^1(v) = h(v)$ where i denotes the external level. Mathematical background for this approach and validation on academic test cases and solution of nonlinear PDEs are available [20, 23, 25, 26, 27, 28, 22, 29].

In practice, this algorithm succeeds if the trajectory passes close enough to the infimum (i.e. in $B_\varepsilon(x_m)$ where ε defines the accuracy in the capture of the infimum). This means that we should consider for h a functional of the form

$$h(v) = \int_{T_1}^T (J(x_v(\tau)) - J_m)^2 d\tau, \quad \text{for } 0 < T_1 < \tau < T$$

where $x_v(\tau)$ is the trajectory generated by (8) and $T_1 = T/2$ for instance. Also, in the algorithm above, $x_w(Z_w)$ is replaced by the best solution found over $[0, Z_w]$.

In cases where J_m is unknown, we set $J_m = -\infty$ and look for the best solution for a given complexity and computational effort. This is the approach adopted here where we predefine the effort we would like to make in each level of the algorithm.

2.3 Hybridization

It is interesting to notice that once GA is seen as a dynamical system (7) for the population, it can be used as core optimization method in SDA. We call this approach HGSA (hybrid genetic/semi-deterministic algorithm). The aim here is to find a compromise between the robustness of GAs and low-complexity features of SDA.

In practice, as final convergence is difficult with GA based algorithms, one should always complete GA iterations by a descent method for better accuracy. This is useful especially when the functional is flat around the infimum.

2.4 Academic test case

SDA, HGSA and GA algorithms have been compared on the Rastrigin function given by:

$$J(x) = \sum_{i=1}^2 (x_i^2 - \cos(18x_i))$$

with $x \in [-5, 5]^I$ and $I = 10, 100, 1000$. The infimum is 0 reached at the origin.

The two-level SDA algorithm $A_2(v_1, v_2)$ is used with (v_1, v_2) chosen randomly in $\mathbb{R}^I \times \mathbb{R}^I$ and the dynamical system corresponding to the steepest descent method as core optimization algorithm [23].

HGSA and GA are applied with the following values for the three associated stochastic processes (see section 2.1):

- The population size has been set to $N_p = 180$ (resp. 20) for GA (resp. HGSA).
- The selection is a roulette wheel type proportional to the rank of the individual in the population.

	SDA	GA	HGSA
I=10	66	6000	2600
N=100	70	O(1e5)	O(1e4)
N=1000	80	O(1e7)	O(1e5)

Table 1: Rastrigin function. Total number of functional evaluations needed to reach the infimum with an accuracy of 10^{-6} .

- The crossover is barycentric in each coordinate with a probability of $p_c = 0.45$.
- The mutation process is non-uniform with probability of $p_m = 0.15$.
- A one-elitism principle, that consists in keeping the current best individual in the next generation, has also been imposed.

All these parameters are fixed and used in all computations of this paper. For this test case, the Generation number N_{gen} not fixed.

Results are presented on Table 1. The SDA algorithm is faster and more efficient than GA and HGSA on this case. However the HGSA give a good compromise to the GA. This has also been observed on other analytical cases available in the literature [25, 20].

3 Low-cost sensitivity

Consider a general simulation loop, leading from shape parametrization x to the cost functional J :

$$J(x) : x \rightarrow q(x) \rightarrow U(q(x)) \rightarrow J(x, q(x), U(q(x))) \quad (11)$$

where q is the shape geometry and U is the state equation solution.

The Jacobian of J is given by:

$$\frac{dJ}{dx} = \frac{\partial J}{\partial x} + \frac{\partial J}{\partial q} \frac{\partial q}{\partial x} + \frac{\partial J}{\partial U} \frac{\partial U}{\partial q} \frac{\partial q}{\partial x} \quad (12)$$

The last term $\frac{\partial J}{\partial U} \frac{\partial U}{\partial q} \frac{\partial q}{\partial x}$ is the more expensive to compute as it requires the linearization of the state equations.

One way to reduce computational effort of sensitivity evaluations is to use reduced complexity models which provide an inexpensive approximation of the last term in [12]. For instance, consider the following reduced model for the definition of $\tilde{U}(x) \sim U(q(x))$. Suppose \tilde{U} is a wall function to be used instead of the full flow equation on the wall and giving wall values knowing local internal flow description. The incomplete gradient of J with respect to x can be improved evaluating the former term in [12] linearizing the simple model. Note that \tilde{U} is never used in the definition of the state U , but only in an approximation of $\partial \tilde{U} / \partial x$. It is also important to notice that the reduced model needs to be valid only over the support of the control parameters. More precisely, we linearize the following approximate simulation loop

$$x \rightarrow q(x) \rightarrow \tilde{U}(x) \left(\frac{U(q(x))}{\tilde{U}(x)} \right). \quad (13)$$

freezing $U(q(x))/\tilde{U}(x)$ which gives

$$\frac{dJ}{dx} \approx \frac{\partial J(U)}{\partial x} + \frac{\partial J(U)}{\partial q} \frac{\partial q}{\partial x} + \frac{\partial J(U)}{\partial U} \frac{\partial \tilde{U}}{\partial x} \frac{U(q(x))}{\tilde{U}(x)}. \quad (14)$$

A simple example shows the importance of the scaling introduced in [13]. Consider $U(x) = \log(1+x)$ scalar for simplicity and $J(x) = U^2(x)$ with $dJ(x)/dx = 2U(x)U'(x) = 2\log(1+x)/(1+x) \sim 2\log(1+x)(1-x+x^2\dots)$ and consider $\tilde{U}(x) = x$ as the reduced complexity model, valid around $x = 0$. To see the impact of the scaling factor we compare $J'(x) \sim 2U(x)\tilde{U}'(x) = 2\log(1+x)$ with $J'(x) \sim 2U(x)\tilde{U}'(x)(U(x)/\tilde{U}(x)) = 2\log(1+x)(\log(1+x)/x) \sim 2\log(1+x)(1-x/2+x^2/3\dots)$.

Another way to define low-complexity models is to use a different level of discretization for U with the same state equation. We can look for state sensitivity on coarse meshes while the state is evaluated on much finer discretizations:

$$\frac{dJ}{dx} = \frac{\partial J}{\partial x}(U_f, q_f) + \frac{\partial J}{\partial q} \frac{\partial q}{\partial x}(U_f, q_f) + \frac{\partial J}{\partial U} \frac{\partial U}{\partial q} \frac{\partial q}{\partial x}(I_f^c U_f, q_c)$$

where f and c subscripts denote fine and coarse meshes, I_f^c is an interpolation operator between the fine and coarse meshes. By fine mesh we mean a mesh enough fine for the solution to become independent from the mesh. This means that the linearization is performed on a coarse mesh, however around an accurate state variable computed on a fine mesh. In that case, obviously if the coarse mesh tends to the fine one, the approximate gradient tends to the gradient on the fine mesh. In addition, to the two levels of refinements used for state and sensitivity calculations, state evaluations for gradient calculation can be only made partially. Hence, only partial convergence of solver method in the solution of the state equations is required starting from $I_f^c U_f$. This corresponds to the fact that the semi-deterministic algorithm above only needs a descent direction d such that $d \cdot \nabla J > \varepsilon > 0$ (see Figure 4) [20, 25, 23].

This last approach is easy to couple with any commercial code.

4 Fast-micro-mixer modelling

The SDA, HGSA and GA algorithms above are used to optimize the shape of a given fast-micro-mixer in order to reduce its mixing time.

4.1 Shape design

The mixer shape considered is a typical three-inlet/single-outlet channel architecture proposed by Knight [12] (see Figure 1). Due to the fact that our model is symmetric we only study half of the mixer [1] (see Figure 2-*Left*). Our model is a 2D approximation of the physical system [30]. Experiments show a 5 percent deviation from a 3D modelling which is satisfactory for a 2D model to be used as low-complexity model in optimization [1]. Our aim is to optimize the

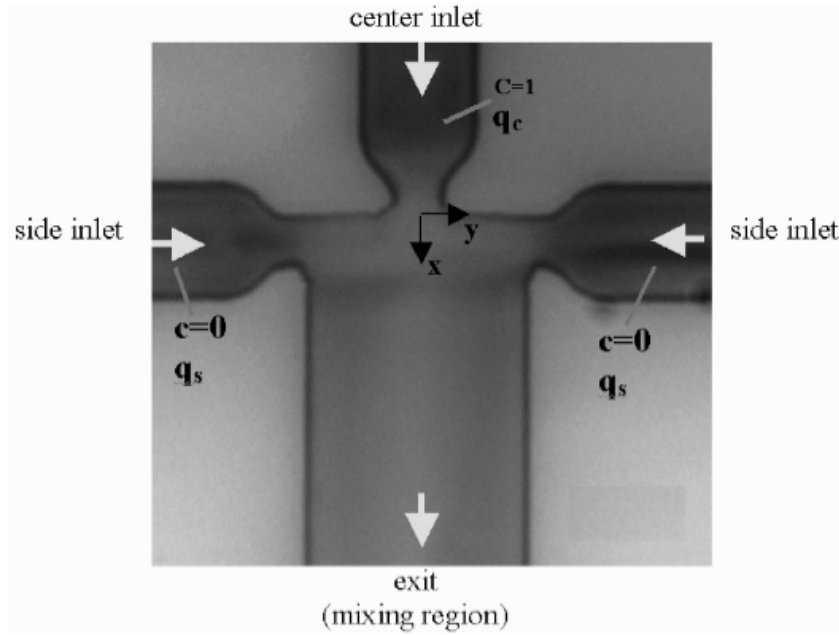


Figure 1: Typical fast-micro-mixer geometry. q_s and q_c are respectively the side/center injection velocities. c is the denaturant concentration.

corner shapes. We parameterize the corner regions by cubic splines (see Figure 2-Right). The total number of parameters is 8, 4 for each corner. In addition, we account for the following constraints:

- The considered fast-Micro-Mixer is $22\mu m$ long and $10\mu m$ large.
- The lithography step in fabrication limits the minimum feature size to a minimum of 1 to 2 μm .
- The width of the side channel nozzles is set to 3 μm and the width of the center channel nozzles to 2 μm to mitigate clogging issues.
- The depth of the channels is set to $\sim 10\mu m$ to optimize the fluorescence signal with a confocal system. In addition, it is difficult to etch deeper on fused silica [1].
- The physical properties of buffers and guanidine hydrochloride denaturant used here for protein folding studies have known parameters such as density, viscosity, and diffusivity.
- Finally, the maximum side flow rate is $U_s = 10^{-4}ml/s$. Hence, a typical flow Reynolds number based on sides channel thickness and flow inlet is $Re_w = \frac{U_s w_s}{\eta} \sim 15$.

Thus, the corresponding search space of the optimization problem is $\Omega = [x_i^{min}, x_i^{max}]_{i=1}^8$ where x_i^{min} (resp. x_i^{max}), the minimum (resp. maximum) value of the i^{th} parameter, are fixed by the previous constraints.

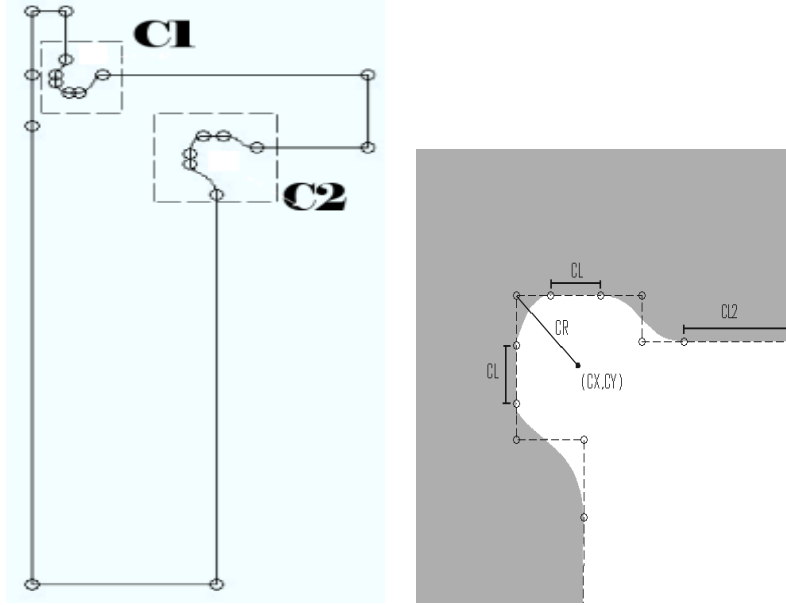


Figure 2: **Left:** Half-Shape parameterization. The corners are denoted by C_1 and C_2 . **Right:** Typical parameterization of a corner. Here C_2 . We consider 4 parameters: C_x, C_r, C_l, C_{l2} . C_y is fixed.

4.2 State equations

The mixer flow was analyzed using numerical solutions of the full Navier-Stokes fluid flow equations and a Convective diffusion equation describing concentration fields c of the guanidine hydrochloride denaturant. Only steady configurations have been considered as we are not interested in the behavior of the device during its transient set up.

These flow simulations were used to explore the guanidine hydrochloride performance of a variety of mixer designs with systematically varied flow and geometric parameters. The model is applied to mixer shape designs described in 4.1. We approximate flow at the vertical midplane with two-dimensional flow simulations:

$$\begin{cases} -\nabla \cdot (\eta(\nabla u + (\nabla u)^\top)) + \rho(u \cdot \nabla)u + \nabla p = 0 \\ \nabla \cdot u = 0 \\ \nabla \cdot (-D\nabla c + cu) = 0 \end{cases} \quad (15)$$

where (u, p) is the flow velocity vector and pressure field, $\rho = 1,013 \text{kg/m}^3$ is the density, $\eta = 110^{-3} \text{Kg/ms}$ the dynamic viscosity and $D = 2^{-9} \text{m}^2/\text{s}$ is the diffusion coefficient.

Finally, the following boundary conditions are assumed: $u = 0$ on shape border, $u = 3.210^{-4} \text{m/s}$ on side inlets, $u = 3.210^{-6} \text{m/s}$ on center inlet, $u \cdot t = 0$ on the exit, $u \cdot n = 0$ on the center symmetry line. (t, n) is the local orthonormal reference frame along the boundary. c is prescribed at inlet and normal zero gradient is assumed for all other boundaries. $c = 0$ at side inlet and $c = 1$ at

center inlet.

In order to achieve a numerical solution, the Incompressible Navier-Stokes equation nonlinear solver solves the equations iteratively. It uses Lagrange P2-P1 elements to stabilize the pressure and to realize the Ladyzhenskaya, Babouska and Brezzi (LBB) stability condition. Thus 2nd-order Lagrange elements model the velocity components while linear elements model the pressure. The element settings in this application mode always provides one order higher Lagrange elements for the velocity components than for the pressure. The convective diffusion equation is solved using a streamline-upwind/Petrov-Galerkin (SUPG) method in order to stabilize the advection [31]. These both stabilization techniques prevent numerical oscillations and other instabilities in solving problems with advection-dominated flows and when using equal-order interpolation functions for velocity and pressure. A Direct Damped Newton method is then used to solve the linear systems leaking from 15- 15 [32].

4.3 Cost Function

The cost function to minimize is the mixing time of the considered Lagrangian fluid particle travelling along the centerline into our fast-micro-mixer with a shape associated to $x_{shape} \in \Omega$. In this paper, we define mixing time as the time required to change the concentration of a typical protein particle from 90% to 30% of the initial value c_0 . Then the cost function is given by:

$$J(x_{shape}) = \int_{c_{90}^{x_{shape}}}^{c_{30}^{x_{shape}}} \frac{dy}{u^{x_{shape}}(y).t} \quad (16)$$

Where $c_{90}^{x_{shape}}$ and $c_{30}^{x_{shape}}$ denote respectively the points along the symmetry line where the concentration is at 90% and 30% of c_0 .

This modelling has been validated by a posteriori prototyping [1]. We are interested by an ensemble of state equations enough rich for the optimization problem to be valid but also as simple as possible to control the computational complexity. In particular, we need to keep the cost of state evaluations low for genetic algorithms and also for sensitivity analysis.

5 Results and discussion

In GA (resp. HGSA) the maximum generation number is set to 30 (resp 10). SDA starts from an initial shape made with a smoothed 90 degrees corners parameterized with splines to keep the admissible regularity. GA and HGSA start from a random initial population. In three optimizations, the mixing time has been decreased from $8\mu s$ to $1.15\mu s$ (see Figure 3-*Right*). Initial and optimized shapes are presented in Figure 3-*Left*. For GA (resp. HGSA) the total number of functional evaluations is 5400 (resp. 2500). For SDA the evaluation number is 3500 with more than 90 % of the evaluations on a coarse mesh with incomplete state evaluations (Figure 5). The cost of an incomplete evaluation of the gradient is around two evaluations of the functional on a fine mesh. Convergencehistories are given in Figure 6. As we can see on this Figure, SDA has visited several attraction basins before exploring the best element basin. Each evaluation on the fine (resp. coarse) level requires about one minute (resp.

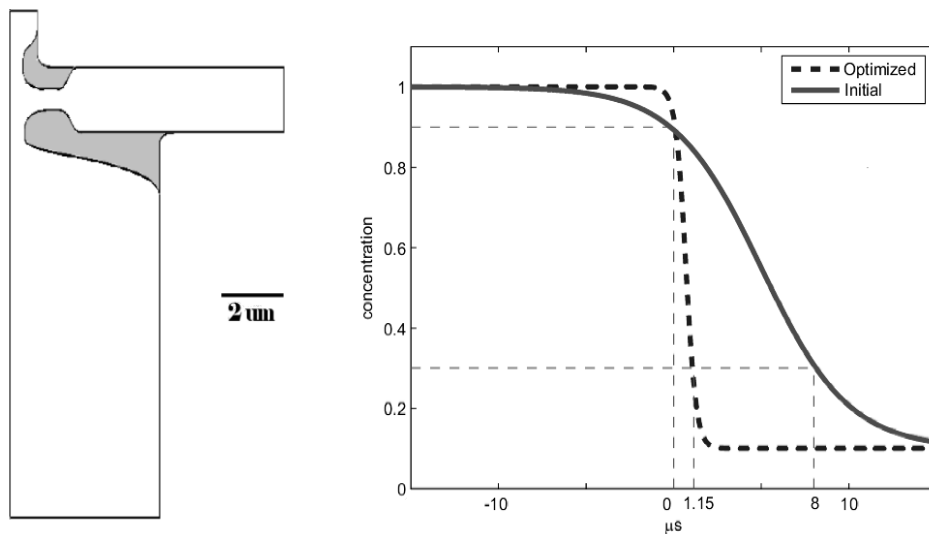


Figure 3: **Left:** SDA and GA optimized shape superposed over Initial shape. Parts in grey have been removed by the algorithms. **Right:** Concentration evolution for the initial and Optimized shapes.

20s) on a 3Ghz PC computer. Hence, GA requires about 4 days, HGSA 2 days and SDA less than 4 hours to reach the same minimum. In addition, with SDA the infimum is reached sooner in the optimization (see Figure 6).

6 conclusion

An unified formulation for deterministic and stochastic global optimization based on the solution of initial and boundary value problems for dynamics systems has been presented. The solution of this boundary value problem leads to a recursive semi-deterministic minimization algorithm where non-deterministic aspects is reduced as much as possible to limit the complexity of the method. To keep the computational complexity low and make the problem easy to solve with industrial softwares approximate gradient evaluation has been used on coarse meshes.

Both algorithms over-perform in term of computational complexity genetic algorithms when applied to the academic configurations and for the design of a fast-micro-mixer. The obtained geometries are currently under prototyping at Stanford microfluidic laboratory.

References

- [1] E. Hertzog, X. Michalet, M. Jager, X. Kong, J. Santiago, J. Weiss, and O. Bakajin. Femtomole mixer for microsecond kinetic studies of protein folding. *Proceedings of the National Academy of Sciences of the USA*, page Accepted, 2005.

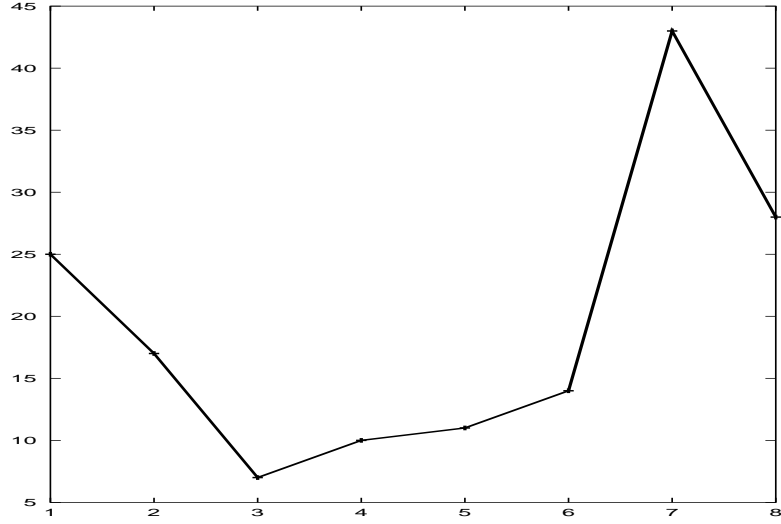


Figure 4: Comparison of gradients computed on the fine vs. coarse meshes ($\text{sign}(J_x^{Fine} J_x^{Coarse}) \frac{|J_x^{Fine} - J_x^{Coarse}|}{|J_x^{Fine}|}$) We notice that the sign is always correct.

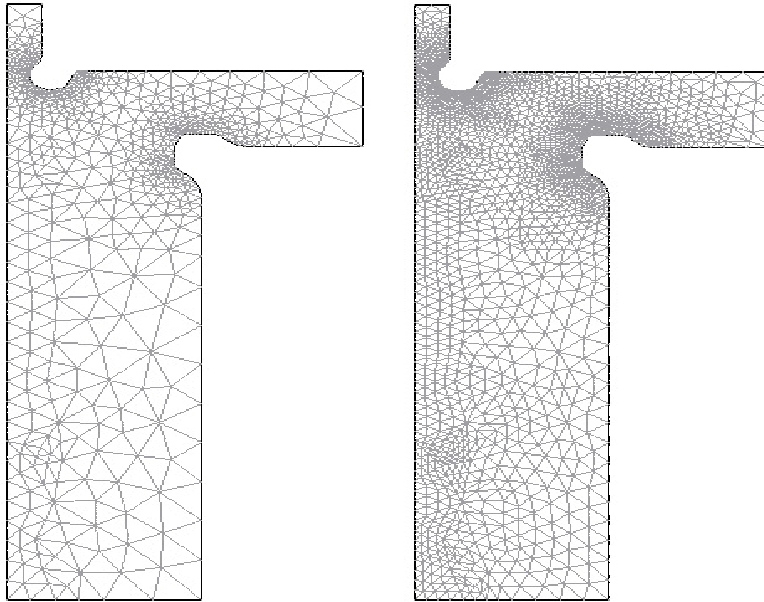


Figure 5: **Left:** For gradient calculation the state is partially evaluated on a coarse mesh. **Right:** On each new shape the state is calculated on a fine mesh.

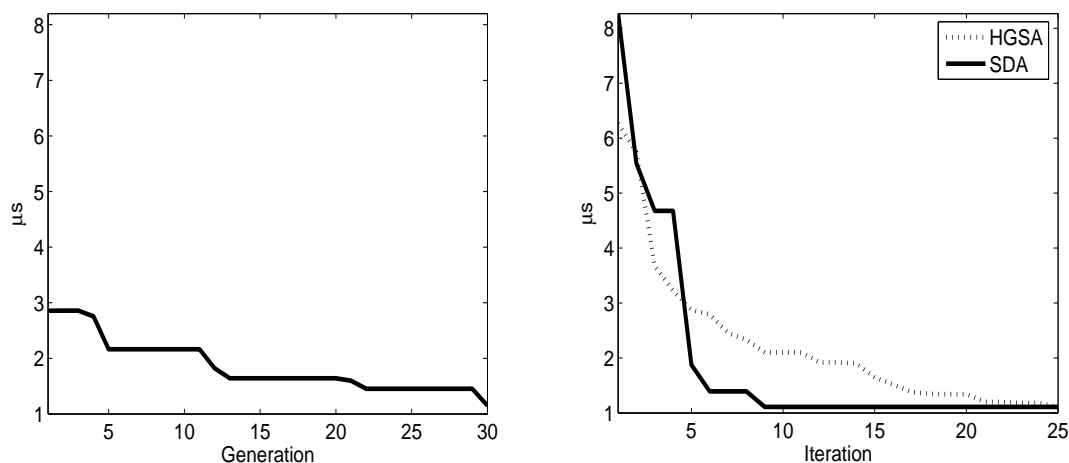


Figure 6: **Left:** Best element convergence history vs. generation iterations for GA. **Right:** Best element convergence history vs. iterations for SDA and HGSA.

- [2] Roder H. *Proceedings of the National Academy of Sciences of the United States of America*, 101:1793–1794, 2004.
- [3] C. M. Jones, E. R. Henry, Y. Hu, C.-K. Chan, S. D. Luck, A. Bhuyan, H. Roder, J. Hofrichter, and W. A. Eaton. *Proceedings of the National Academy of Sciences of the USA*, pages 11860–11864, 1993.
- [4] S. J. Hagen and W. A. Eaton. *Journal of Molecular Biology*, 301:1037–1045.
- [5] K. M. Pryse, T. G. Bruckman, B. W. Maxfield, and E. L. Elson. *Biochemistry*, 31:5127–5136, 1992.
- [6] M. Jacob, G. Holtermann, D. Perl, J. Reinstein, T. Schindler, M. A. Geeves, and F. X. Schmid. *Biochemistry*, 38:2882–2891, 1999.
- [7] C.K. Chan, Y. Hu, S. Takahashi, D. L. Rousseau, and W. A. Eaton. *Proceedings of the National Academy of Sciences of the United States of America*, 94:1779–1784, 1997.
- [8] Pollack L., M. W. Tate, N. C. Darnton, J. B. Knight, S. M. Gruner, W. A. Eaton, and R. H. Austin. *Proceedings of the National Academy of Sciences of the USA*, 96:10115–10117, 1999.
- [9] S.-H. Park, M. C. R. Shastry, and H. Roder. *Nature, Structural Biology*, 6:943–947, 1999.
- [10] J. P. Brody, P. Yager, R.E. Goldstein, and R.H. Austin. *Biophysical Journal*, 71:3430–3441, 1996.
- [11] R. Russell, S. Ian, M. W. T. Millett, W. Lisa, Kwok, Bradley, Nakatani, M. Sol, S. G. J. Gruner, V. P. S. Mochrie, D. H. Doniach, and Pollack Lois.

- Proceedings of the National Academy of Sciences of the United States of America*, 99:4266–4271, 2002.
- [12] J. B. Knight, A. Vishwanath, J. P. Brody, and R. H. Austin. *Physical Review Letters*, pages 3863–3866, 1998.
- [13] M. C. R. Shastry, S. D. Luck, and H. Roder. *Biophysical Journal*, 74:2714–2721, 1998.
- [14] B. Mohammadi, J. Santiago, and J. Molho. Incomplete sensitivities in the design of minimal dispersion fluidic channels. *Computers and Fluids*, 192, 2003.
- [15] B. Mohammadi, R. Bharadwaj, and J. Santiago. Design and optimization of on-chip capillary electrophoresis. *Electrophoresis Journal*, 23(16), 2002.
- [16] B. Mohammadi, J. Molho, A. Herr, J. Santiago, T. Kenny, R. Brennen, and Gordon G. Optimization of turn geometries for on-chip electrophoresis. *Analytical Chemistry*, 73(6), 2001.
- [17] B. Mohammadi, E. Leclerc, and P. Sagaut. Optimal shape design of coastal structures. *Computers and Fluids*, 12(5), 2003.
- [18] L. Dumas, V. Herbert, and F. Muyl. Hybrid method for aerodynamic shape optimization in automotive industry. *Computers and Fluids*, 33(5):849–858, 2004.
- [19] D. Goldberg. Genetic algorithms in search, optimization and machine learning. *Addison Wesley*, 1989.
- [20] B. Ivorra. Semi-deterministic global optimization. *PhD. University of Montpellier 2*, 2006.
- [21] H. Attouch and R. Cominetti. A dynamical approach to convex minimization coupling approximation with the steepest descent method. *J. Differential Equations*, 128(2):519–540, 1996.
- [22] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Oxford University Press, 2001.
- [23] B. Mohammadi and J-H. Saiaac. *Pratique de la simulation numrique*. Dunod, 2002.
- [24] G.N. Vanderplaats. Numerical optimization techniques for engineering design. *Mc Graw-Hill*, 1990.
- [25] B. Ivorra, B. Mohammadi, and P. Redont. Low-complexity global optimization by solution of bvp. *Optimal Control and Applied Mathematics*, submitted, 2005.
- [26] D. Isebe, B. Mohammadi, P. Azerad, B. Ivorra, and F. Bouchette. Optimal shape design of coastal structures. *Proceeding of Siam on Mathematical & Computational Issues in the Geosciences*, 2005.

- [27] L. Debiane, B. Ivorra, B. Mohammadi, F. Nicoud, A. Ern, T. Poinso, and H. Pitsch. Temperature and pollution control in flames. In *Proceeding of the Summer Program*, pages 367–375, Center for Turbulence Research, NASA AMES/Stanford University, USA, 2004.
- [28] B. Ivorra, B. Mohammadi, L. Dumas, Y. Moreau, G. Pille, and O. Durand. Apodisation de fibres à réseaux de bragg pour la synthèse de codes cdma spectral. *IEEE Lasers and Electro-Optics Society French Chapter, Conference COSTO04*, 2004.
- [29] B. Mohammadi and O. Pironneau. Shape optimization in fluid mechanics. *Annual Review of Fluid Mechanics.*, 36:255–279, 2004.
- [30] N. Darnton, O. Bakajin, R. Huang, B. North, J. Tegenfeldt, E. Cox, J. Sturn, and R. H. Austin. Condensed matter. *Journal of Physics*, 13:4891–4902, 2001.
- [31] T. Hughes and A. Brooks. A multi-dimensional upwind scheme with no crosswind diffusion, in t. hughes, ed., finite element methods for convection dominated flows. *ASME, New York*, 34:19–35, 1979.
- [32] P. Deuffhard. A modified newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting. *Numer. Math.*, 32:289–315, 1974.