Derivative Free Optimization Methods for Optimizing Stirrer Configurations

Ö. Uğur [a], B. Karasözen [b,*], M. Schäfer [c], K. Yapıcı [d]

[a] *Institute of Applied Mathematics, Middle East Technical University, 06531 Ankara, Turkey.*

[b] *Department of Mathematics and Institute of Applied Mathematics, Middle East Technical University, 06531 Ankara, Turkey*

[c] *Department of Numerical Methods in Mechanical Engineering, Darmstadt University of Technology, Petersenstr. 30, D-64287 Darmstadt, Germany*

[d] *Department of Chemical Engineering, Middle East Technical University, 06531 Ankara, Turkey*

Abstract

In this paper a numerical approach for the optimization of stirrer configurations is presented. The methodology is based on a flow solver, and a mathematical optimization tool, which are integrated into an automated procedure. The flow solver is based on the discretization of the incompressible Navier-Stokes equations by means of a fully conservative finite-volume method for block-structured, boundary-fitted grids, for allowing a flexible discretization of complex stirrer geometries. Two derivative free optimization algorithms, the DFO and CONDOR are considered, they are implementations of trust region based derivative-free methods using multivariate polynomial interpolation. Both are designed to minimize smooth functions whose evaluations are considered to be expensive and whose derivatives are not available or not desirable to approximate. An exemplary application for a standard stirrer configuration illustrates the functionality and the properties of the proposed methods. It also gives a comparison of the two optimization algorithms.

Key words: Energy, Numerical Optimization, Derivative-Free Optimization, Computational Fluid Dynamics (CFD), Parallel Computing

* Corresponding author.
Email address: bulent@metu.edu.tr (B. Karasözen).

1 Introduction

Stirring and mixing devices are utilized in chemical, pharmaceutical, and biotechnological industries for different purposes: mixing of two substances with each other, generation of a spatially homogeneous concentration in a reactor, preventing sedimentation, disintegrating bacteria etc. In the past, the design of stirring devices for such purposes has been made mostly by a scale-up of small scale experiments. Because the similarity parameters used in this process can never take into account all details of the geometry and the flow, the final large scale design often did not achieve the desired goals. On the other hand, minimization of the amount of energy needed for the creation of certain mixing conditions, the material costs for the stirrer, as well as the lifetime and the breakdown security are important economic issues for the stirring process which depend on the various geometric parameters of the stirrer and the vessel as well as on the rotation rate and the fluid properties. Therefore, numerical simulations are needed for the computation of the fluid flow inside the stirrer for various Reynolds numbers. In Section 2, we shortly describe the CFD solver FASTEST-3D applied to Rushton turbine.

There is a large number of stirrers used in the industry for different mixing tasks. Experimental investigations for selecting or designing a "good" stirrer for a specific process with respect to given criteria are usually very costly and time consuming. Numerical simulation techniques provide great flexibility in dealing with variation of geometric parameters. Therefore, the combination of the flow solver together with mathematical optimization methods can be a very useful tool for the selection or design of stirrers with higher product quality, reduced costs, and lower energy consumption.

In many large scale computations, when flow simulation packages are used, either the calculation of the objective function is very expensive or the gradient of the objective function is not available. In such circumstances, for the minimization of functions of several variables, derivative-free optimization methods can be used instead of gradient based algorithms. In Section 3, we describe two such derivative-free optimization methods, DFO and CONDOR.

The DFO algorithm in combination with FASTEST-3D has been applied to Rushton turbine in [14]. In Section 4, computational results obtained by DFO and CONDOR are presented to show how the derivative-free methods can be helpful in industrial applications. Moreover, within the section the results of both algorithms we applied are compared for two different Reynolds numbers.

The paper concludes with some future perspectives for improving both optimization methods by using the structure of the objective function calculated by the flow solver.

2 Flow Solver

The equations governing fluid dynamics within the stirrer are described in [14]. The numerical procedure adopted in FASTEST-3D [1] to solve the incompressible Navier-Stokes equations is composed of a fully conservative second order finite-volume discretization in time and space (central differences, deferred correction for the convective fluxes) for non-orthogonal, boundary-fitted, block-structured grids, allowing a flexible discretization of even very complex stirrer geometries. The nonlinear algebraic equations are solved at each time-step implicitly by a multigrid method with a pressure-correction smoother. For the parallelization, a block-structured grid partitioning method with automatic load balancing and strongly implicit block coupling is used [9]. The grid movement of the stirrer against the vessel is handled by a clicking mesh approach. The solver has already been applied to a variety of problems in stirrer technology and has proved that it can compute complex problems on parallel computers with high numerical and parallel efficiency [2,16].

The grid generation tool involves an algebraic method based on transfinite interpolation for the generation of multi-block boundary-fitted grids, which facilitates the accurate representation of the complex geometries associated with stirrer configurations. In order to allow an easy design variation, the grid generation is parametrized with respect to the characteristic geometric quantities for the different stirrer types. Thus, the input parameters are radii of hub, shaft, disk, vessel or the numbers and dimensions of blades, baffles and so on. After specifying the number of control volumes for different stirrer sections the grid is created automatically by respecting basic criteria with respect to grid quality, i.e., skewness, aspect ratio, and expansion factor [10], as far as possible. Following this concept the geometrical input parameters can be used directly in an easy way as design parameters for the optimization purpose.

3 Derivative Free Optimization

Applications of numerical optimization techniques for fluid flow problems can be found mostly in the field of aerodynamics. An overview of the subject is given by Mohammadi and Pironneau [12]. For general fluid flow applications the resulting nonlinear optimization problems are typically solved by employing descent algorithms based on gradient information. However, the computation of the gradient of the objective function with respect to the design variables is usually very costly and contaminated by noise. Usually the scalar objective function, $f : x \mapsto f(x),\ x \in \mathbb{R}^n$, is a result of very expensive computer simulations and yet the derivatives may not be available at all.

For the computer simulation of fluid inside the stirrer by the CFD package FASTEST-3D was used in [14]. By simulating the complex discrete Navier-Stokes system, the local gradient of the objective function f with respect to the design variables x, are not provided by the flow solver and are not directly available. In such case we may easily apply derivative-free optimization methods.

There are mainly two classes of derivative-free optimization methods. The first class is based on direct search methods and doesn't require smoothness of the objective function f. A recent and comprehensive survey of this class is given in [11]. The second class of methods is based on interpolation (or approximation) and requires smoothness. These algorithms construct a basis consisting of suitable polynomials spanning a certain space of functions, and then, build a model, using a linear combination of basis polynomials, that interpolates or approximates the known values of f at the point x. In this paper, we consider two derivative-free optimization methods from this class: the DFO (Derivative Free Optimization) method was developed by Conn and co-workers [6–8] and the CONDOR (Constrained, Non-linear, Direct, parallel Optimization using trust Region method for high-computing load function) developed by Berghen [3] which is based on the UOBYQA of Powell [13]. Although both methods were developed mainly for unconstrained optimization problems and for easy (box) constraints, there are extensions for constrained ones. The algorithms are based on a derivative-free trust region method approximating the objective function by a quadratic polynomial, which is then minimized by a sequential quadratic programming (SQP) method.

DFO was successfully applied as a black-box optimization routine in optimizing energy systems [17] and for the helicopter rotor blade design [15], where some practical aspects of DFO were described. Numerical tests in both papers show that DFO works faster and more accurate than the derivative based methods such as the Quasi-Newton. CONDOR was also compared with DFO in some respects for several test problems [4] and its applicability for some CFD simulations has been shown in [3].

Surely, both methods have their own advantages and drawbacks. Below we summarize the main steps as well as similarities and differences between the two algorithms, DFO and CONDOR:

- Both methods choose a basis for the multivariate interpolation. DFO uses Newton polynomials while CONDOR uses Lagrange polynomials as the basis for the space of quadratic polynomials. The objective function f of n independent variables x is then approximated by a quadratic model using $p = (n+1)(n+2)/2$ interpolation points or at least a linear model (in DFO) using $p = n + 1$ interpolation points over the well-poised interpolation set which have to satisfy some geometric conditions that will be specified in the

sequel. At the kth step the quadratic model within the trust-region $\mathcal{B}_k$ may be given as

$$m(x_k + s) = f(x_k) + \langle g(x_k)\,,\, s\rangle + \frac{1}{2}\,\langle s\,,\, H(x_k)s\rangle$$

for some $g \in \mathbb{R}^n$ and some symmetric $n\times n$ matrix H where $\langle \cdot\,,\,\cdot\rangle$ denotes the inner product. The vector g and the matrix H do not necessarily correspond to the first and second derivatives of the objective function f, since they are either unavailable or not desirable to approximate.

However, both algorithms differ in constructing the model m, the polynomial interpolation: If $\{\phi_i\}_{i=1}^{p}\,,\ p \le (n+1)(n+2)/2$, is a basis in the space of quadratic polynomials then the interpolation condition

$$m(y_j) = \sum_{i=1}^{p} \alpha_i \phi_i(y_j) = f(y_j), \qquad j = 1, \ldots, p$$

emphasizes that the coefficient matrix $\Phi(\mathcal{Y}) = [\phi_i(y_j)]$ at the interpolation points $y_j \in \mathcal{Y}$, is closely affected by the chosen basis polynomials as well as the interpolation set $\mathcal{Y}$. For instance, the choice of Newton polynomials (in DFO) causes the matrix to be lower triangular with a special block diagonal structure, while the use of Lagrange polynomials (in CONDOR) results in an identity matrix. Conversely, the reduced forms of $\Phi(\mathcal{Y})$, which can be obtained by a procedure similar to the Gram-Schmidt orthogonalization process, provide the bases of fundamental polynomials. The total number of $(n+1)(n+2)/2$ function evaluations in order to build the full quadratic model is involved with high complexity in linear algebra due to repeated minimization. Therefore, the use of DFO and CONDOR in practice is limited to problems with a very moderate number of variables. For problems with more than 20 variables both methods are computationally very intensive.

- In each of the algorithms the constructed model is minimized over a trusted region (an n-dimensional ellipsoid around the current iterate). The minimization of the quadratic model is done by applying a standard optimization procedure, for instance, a sequential quadratic programming (SQP) method; and in our case, an interface to IPOPT [18], which is also provided by the DFO package, is used. After computing the optimal point, the achieved reduction in the objective function is compared to the reduction obtained by the model by the ratio

$$r_k = \frac{f(x_k) - f(\widehat{x}_k)}{m(x_k) - m(\widehat{x}_k)},$$

where $\widehat{x}_k$ is such that

$$m(\widehat{x}_k) = \min_{x\in\mathcal{B}_k} m(x).$$

- The next step is the updating of interpolation points which is the most critical part. If the reduction of the objective function compared to the reduction r_k of the model is good enough, this point will be included in the interpolation set, otherwise a new point is added to the interpolation set which will improve the approximation model. Interpolation points far from the current iterate or points that are not sufficient to describe a well-posed geometry are removed from the interpolation set.

  Furthermore, when the computation of the trust-region radius δ is complete checking of the validity (within ϵ) of the model around the point x_k in CONDOR is based on whether any of the following conditions [3]

$$\|x_j - x_k\| \leq 2\delta \quad \text{for all } j \tag{3.1}$$

$$\frac{1}{6}M \|x_j - x_k\|^3 \max_d \{|P_j(x_j + d)| : \|d\| \leq \delta\} < \epsilon \quad \text{for all } j \tag{3.2}$$

  holds for the Lagrange interpolating basis $\{P_j\}$, where $\left|\frac{d^3}{d\alpha^3} f(x + \alpha v)\right| \leq M$ such that $\|v\| = 1$ and $\alpha \in \mathbb{R}$. Condition (3.1) prevents the algorithm from sampling the model at $(n+1)(n+2)/2$ new points. However, the checking of the validity of the model in DFO is mainly based on condition (3.2), which is not very often satisfied by the trust-region radius. Hence, in some cases more function evaluations are needed in order to rebuild the interpolation polynomial. Usually one wants to use most possible interpolation points at which objective function values are known and which are referred as sample points. Not all new points are taken in the interpolation set, they have to satisfy some geometric requirements, so called *well-poisedness* [6,15,17], because of the non-uniqueness of quadratic interpolation polynomial. The set $\mathcal{Y} = \{y_1, \cdots, y_p\}$ is poised if the determinant of $\Phi(\mathcal{Y}) = [\phi_i(y_j)]$ is nonzero. It may occur that this matrix ill-conditioned for some iterates. This indicates that the interpolation problem is not good for this set of points $\mathcal{Y}$. This can be improved by adding or removing some points.
- Another essential difference between the DFO and the CONDOR algorithm is that the former uses the smallest Frobenius norm of the Hessian matrix to minimize the local model, which may cause numerical instability, whereas CONDOR uses the Euclidean norm which is more robust.
- After the evaluation of the true objective function value and depending on the quality of the achieved reduction and the predicted one, that is, depending on the ratio,

$$\overline{r}_k = \frac{f(x_k) - f(\overline{x}_k)}{m(x_k) - m(\overline{x}_k)},$$

  where $\overline{x}_k$ is such that

$$f(\overline{x}_k) = \min f(\mathcal{Y} \setminus \{x_k\}),$$

  either the new point $\overline{x}_k$ is accepted as the new iterate x_{k+1} and the trust

region radius is increased, or it is rejected and the trust region radius is decreased.

3.1 The Automated Procedure

The components described above are combined within an integrated optimization tool by means of a control script in a modular form which makes it straightforward to modify or replace individual components of the procedure. This automated procedure is illustrated schematically in Figure 1.

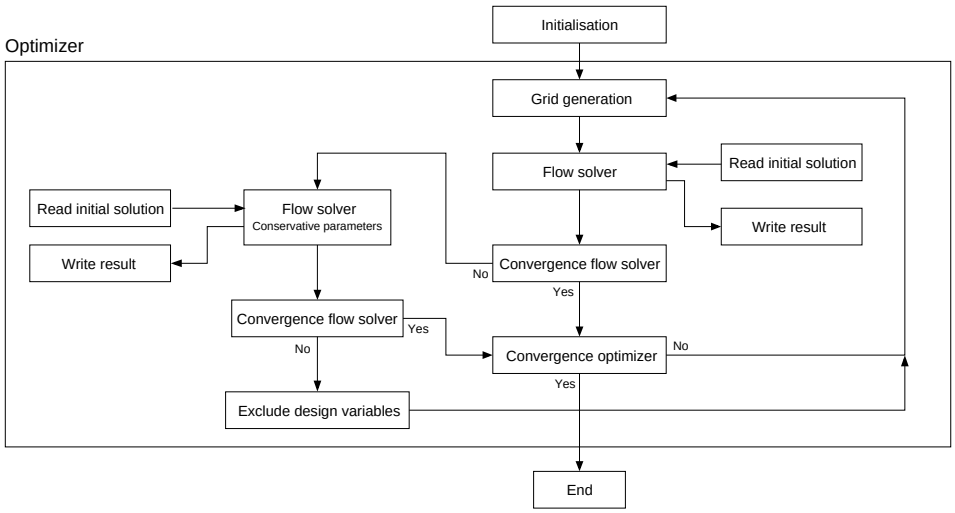

Fig. 1. Flow chart of control script for automated optimization.

After initializations, the procedure involves the following major modular steps [14]:

(1) Optimizer: At each iteration the optimizer computes a new set of design variables unless it converges.
(2) Grid variation: When new design variables are available, the grid generation tool creates the new geometry and the corresponding numerical grid.
(3) Flow simulation: The flow solver computes the flow field and the corresponding objective function value for the new geometry.
(4) Flow solver convergence test: If the flow solver converges, then the optimizer produces new geometrical design variables unless the objective value is found to be minimum. Otherwise, FASTEST-3D runs with more conservative parameters to achieve convergence in its inner calculations. Even if this is not possible, both optimization algorithms may reject the use of those design variables.
(5) Optimizer convergence test: The optimizer decides, by a given criterion, whether the current value of the objective function is accepted as opti-

mum. If yes, the procedure is finished, if not, with the new design variables the optimization procedure is continued to the next iterate.

It is of great importance to note that the flow-chart in Figure 1 represents a systematic way in the optimization of industrial applications, instead of experimenting with several control variables. The presented automated procedure can easily be applied, for instance, in shape-optimizations problems in industry.

4 Computational Results

We consider a Rushton turbine as a representative test case for optimizing a practical stirrer configuration. The geometric parameters, which are considered to define the standard configuration, are given in Table 1.

Parameter	Value
Tank diameter	$T = 0.15\,\mathrm{m}$
Impeller diameter	$D = T/3 = 0.05\,\mathrm{m}$
Bottom clearance	$C = H/2 = 0.075\,\mathrm{m}$
Height of the liquid	$H = T = 0.15\,\mathrm{m}$
Length of the baffles	$W = 3D/10 = 0.015\,\mathrm{m}$
Length of the blade	$\ell = D/4 = 0.0125\,\mathrm{m}$
Height of the blade	$w = D/5 = 0.01\,\mathrm{m}$
Disc thickness	$\mathrm{x} = D/5 = 0.00175\,\mathrm{m}$
Diameter of the disk	$d = 3D/4 = 0.0375\,\mathrm{m}$

Table 1
Geometrical parameters of standard stirrer configuration.

The working Newtonian fluid is a glucose solution with density $\rho = 1330\,\mathrm{kg/m^3}$ and viscosity $\mu = 0.105\,\mathrm{Pas}$.

The numerical grid consists of 22 blocks with a total number of 238 996 control volumes. 17 blocks are defined as rotating with the stirrer while the remaining 5 blocks are defined as stationary with the vessel. A sketch of the considered configuration and the corresponding surface grid of the stirrer are shown in Figure 2. For computational purposes, but not to be too restrictive, the grid on one half of the tank is used due to symmetry. One time step integration needs approximately 20 seconds of computing time on an eight processor Redstone cluster machine. This results in about 8 hours of computing time to reach a

steady state flow in the sense of a frozen rotor computation, a criterion that we adopt for all cases.

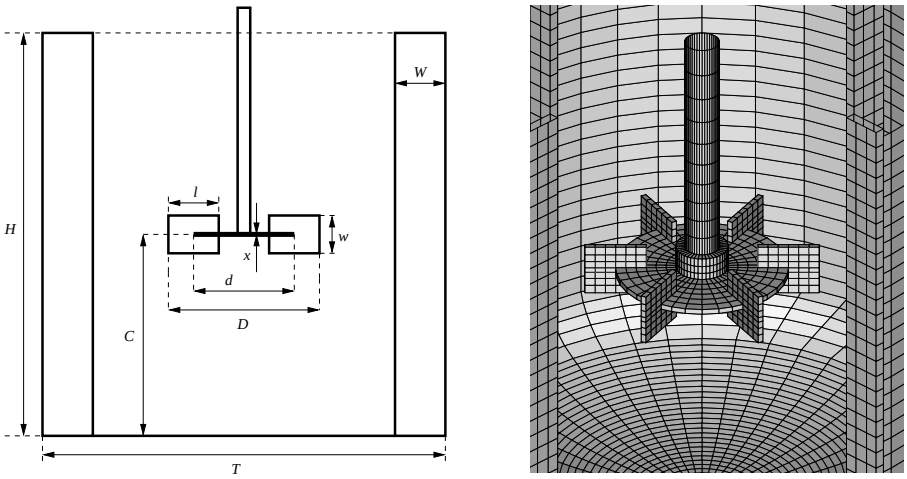

Fig. 2. Sketch of geometrical parameters of stirrer configuration (left) and its corresponding surface grid (right).

The dimensionless Newton number, Ne, which relates the resistance force to the inertia force,

$$Ne = \frac{P}{\rho N^3 D^5}\,, \qquad P = -\int\limits_S \left(p u_j + \tau_{ij} u_i\right) n_j \,\mathrm{d}S\,,$$

is considered as the characteristic reference quantity to be minimized. Here N denotes the rotational speed of the impeller and P is the power computed from the flow quantities over the surface S of the impeller (u_i is the component of the velocity vector corresponding to the Cartesian coordinate x_i, p is the pressure, τ_{ij} is the viscous part of the stress tensor for incompressible Newtonian fluids, and n_j is the component of the outer unit normal vector).

The design variables are the disk thickness x, the bottom clearance C, and the baffle length W, for which the inequality constraints $0.001 \leq \mathrm{x} \leq 0.005$, $0.02 \leq C \leq 0.075$, and $0.005 \leq W \leq 0.03$ are prescribed. All other parameters are kept fixed according to the standard configuration.

Because the gradient information is not available, for both algorithms the minimum trust region radius, δ_{min}, is used as a stopping criterion. However, their default stopping criteria were selected in both algorithms for the subminimization problem. We have investigated two cases of the Reynolds number, $Re = \rho N D^2/\mu$: 100 and 1000. In DFO, for $Re = 100$, $\delta_{min} = 0.0001$ and for $Re = 1000$, $\delta_{min} = 0.001$ were chosen. For CONDOR, we have used $\delta_{min} = 0.0001$ for both Reynolds numbers.

As a first example we consider the minimization of the Newton number for a Reynolds number of $Re = 1000$. Figure 3 shows the Newton number versus the number of cycles for the two optimization algorithms. For both algorithms the Newton number attains the same optimum. CONDOR terminates earlier then the DFO although the latter oscillates around the optimum. Figure 4 depicts the corresponding changes of the design variables during the function evaluations. The three design parameters assign almost the same optimal values. We remark that the two optimization tools differ in building the starting model: DFO starts to build the model objective function approximation (unique or not) from the very beginning. That is, its starting objective polynomial approximation is not fully quadratic. On the other hand, CONDOR starts with a fully quadratic model for the objective function. Despite its long initialization it turns out that CONDOR needs less function evaluations than DFO to reach an optimum point after completing the initial quadratic approximation. DFO, as the time passes, oscillates around the minimum although it approaches the minimum very sharply at the beginning. CONDOR waits for a full quadratic polynomial to build its model, and then gradually approaches the minimum, using a complete interpolating bases. The insignificant overshoots/undershoots (in CONDOR) in the design variables are due to the stopping criteria when the constrained minimization is considered [3, Chapters 9 and 10]. In any case, however, both algorithms reach the same optimized Newton number which is significantly (about 37%) lower than the one for the standard tank configuration stated in Table 1.

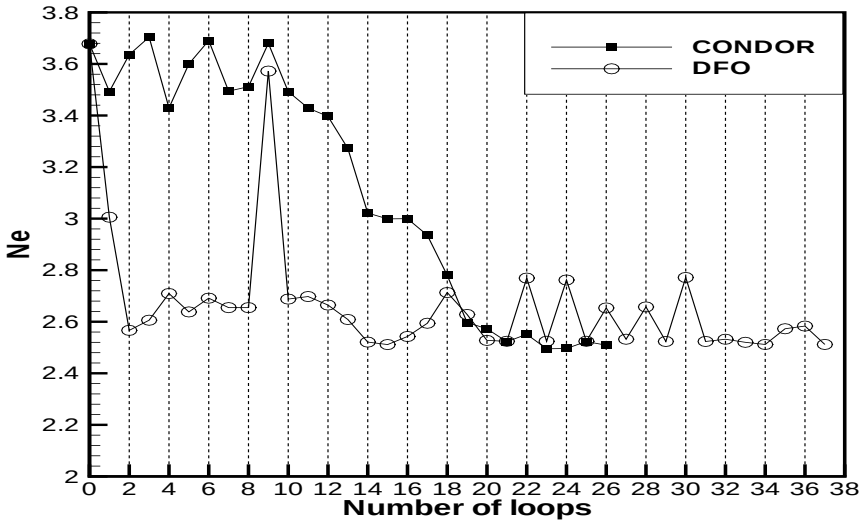

Fig. 3. Newton number versus number of the loops.

As a second example we consider the minimization of the design parameters for the Reynolds number $Re = 100$. Figure 5 depicts the Newton numbers versus number of the function evaluations and shows the local minima of

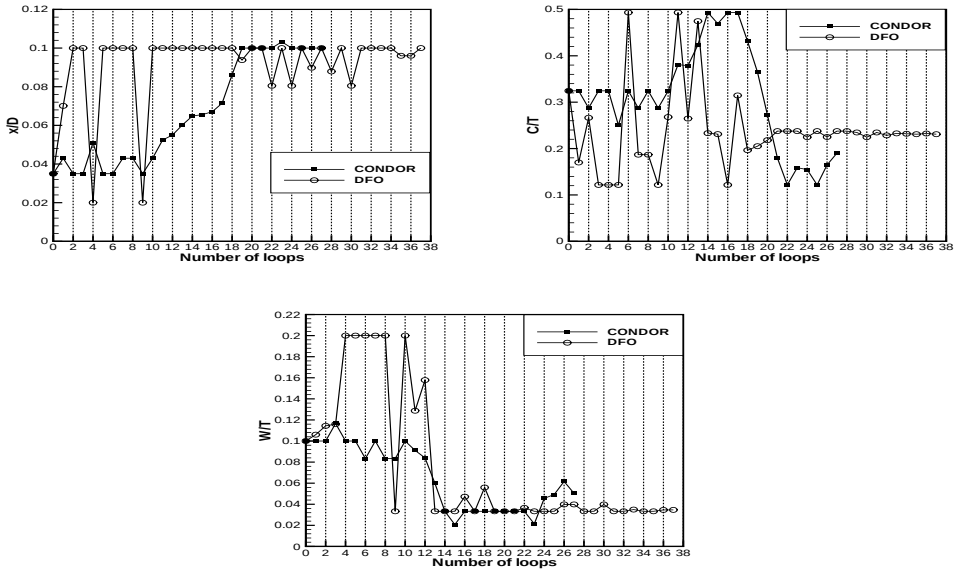

Fig. 4. Design variables (disc thickness, bottom clearance, baffle length) versus number of function evaluations

the objective function. The corresponding design parameters are presented in Figure 6, where we can see the optimum for the clearance C has been achieved differently in each algorithm. In this case the CONDOR and DFO terminate at different Newton numbers, which indicate that they have attained different local minima. The same behavior can be observed for the design variables; disc thickness and baffle length have almost the same optimal values, whereas the optimal bottom clearance resulting from the DFO and CONDOR algorithms are different. This is due to the starting initial trust-region radius and the initial model built within the trust-region. We have used as the initial trust region radius for DFO $\delta_{initial} = 1$ and for CONDOR $\delta_{initial} = 0.1$. We remind the reader that CONDOR uses a quadratic polynomial while DFO builds a first-order polynomial initially to start the optimization procedure.

Two different versions of the DFO algorithm were used; DFO 1.2, for $Re = 1000$ and DFO 2.0 for $Re = 100$. The oscillatory behavior around the optimal value of the DFO 1.2 in Figures 3 and 4 for $Re = 1000$ was eliminated in DFO 2.0 (see Figures 5 and 6) for $Re = 100$.

Another observation that can be obtained from the graphs is that the number of function evaluations in DFO is less than the number of function evaluations in CONDOR. Although the optimum obtained in DFO is "better" than the one obtained in CONDOR, we are not in a position to conclude that either of the algorithm is "better" than the other. But, we should keep in mind that the initialization step of CONDOR algorithm is costly in order to complete the interpolation basis and build a quadratic initial model. This might be a disadvantage for large scale optimization problems. However, it seems, in this

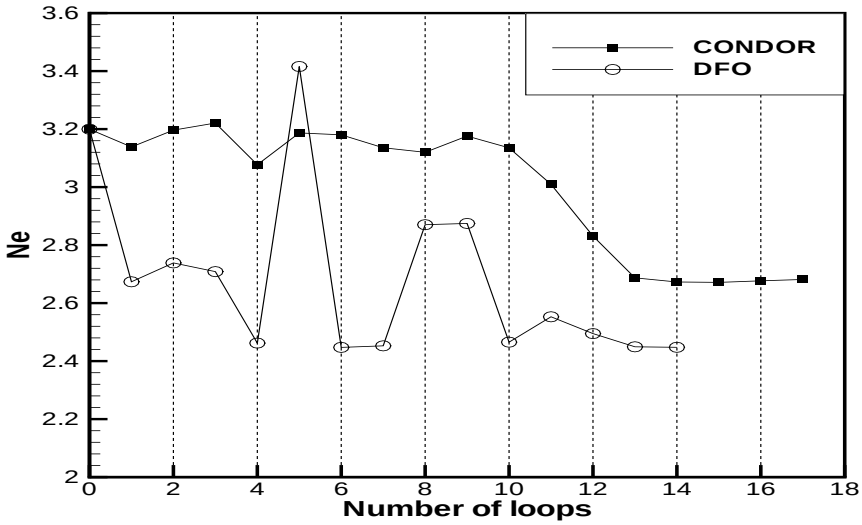

Fig. 5. Newton number versus number of the loops.

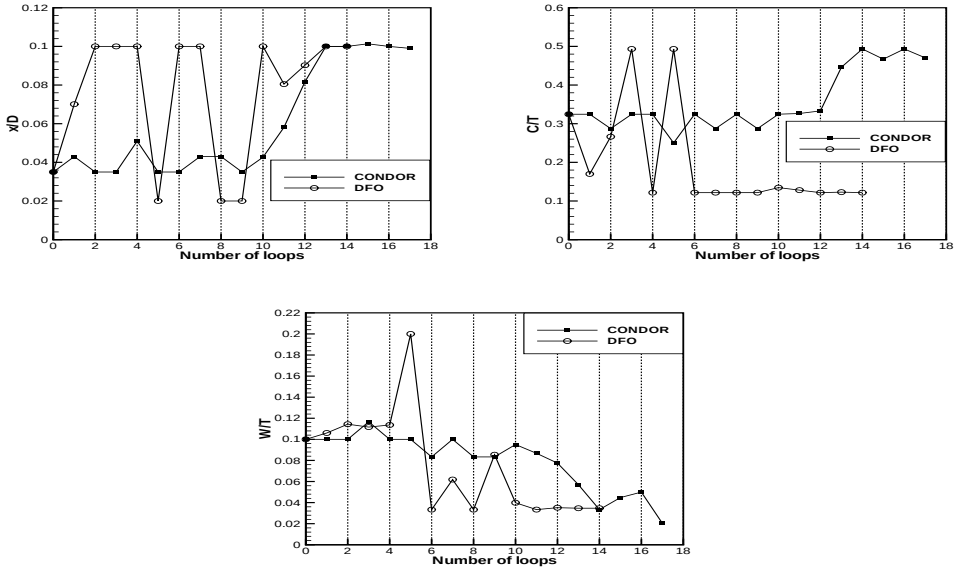

Fig. 6. Design variables (disc thickness, bottom clearance, baffle length) versus number of function evaluations

example, it is better to start the optimization by building a cheap model and gradually to complete the interpolation basis and to use quadratic polynomials whenever possible along the way: this can always be applied when the dimension is large in order to start the optimization algorithm immediately, hoping that linear approximations sufficiently mimics the objective function.

5 Conclusion

In this study we have presented a numerical approach for optimizing practical stirrer configurations. The automated integrated procedure consists of a combination of a parametrized grid generator, a parallel flow solver, and a derivative-free optimization procedure. For the latter, two different methods (DFO and CONDOR) have been investigated with respect to their characteristic convergence properties.

The numerical experiments have shown the principle applicability of the considered approach. For the Rushton turbine, considered as examples in this paper, it has been possible to achieve a significant reduction of the Newton number with relatively low computational effort.

Of course, in a mixing process the power consumption is important but not the sole quantity in obtaining an optimum stirrer configuration. In particular, a satisfactory mixing should be achieved. Due to the generality and modularity of the considered approach, other objectives and/or other design variables can be handled straightforwardly in a similar way.

It should be remarked that; since no rigorous convergence properties for globally optimal solutions are available, the optimal solutions obtained with both optimization tools must be considered as local ones. This aspect can be investigated by variations of the starting value and/or the trust region radius.

As we have seen, if the number of design variables increases, due to the large number of function evaluations, the computational cost of the algorithms becomes very high. For structured problems this cost can be reduced by using the separability of the objective function. The concept of the partially separability was exploited for unconstrained optimization problems by Ph. Toint in a series of papers (see for example [5]) and implemented in the package PSDFO. The FASTEST-3D flow solver, with finite volume discretization, may give us the opportunity to compute the desired function values within blocks and their neighbors and allows us to use the concept of partially separability. A future application would be to apply PSDFO using the structure of the discretized Navier-Stokes equations by the finite volume method within domain decomposition.

6 Acknowledgements

The financial support of the work by the Volkswagen-Stiftung is gratefully acknowledged. The authors would like to thank the anonymous referees for

their helpful comments and suggestions.

References

[1] FASTEST (Flow Analysis for Solving Transport Equations with Simulated Turbulence), Department of Numerical Methods in Mechanical Engineering, Technische Universität Darmstadt, http://www.fnb.maschinenbau.tu-darmstadt.de/en/software/fastest/, 2004

[2] C. Bartels, M. Breuer, K.Wechsel, F. Durst, Computational Fluid dynamics applications on parallel-vector computers:computation of stirred vessel flows. Computer & Fluids, 31 (2000) 69-97.

[3] F. V. Berghen, CONDOR: a constrained, non-linear, derivative-free parallel optimizer for continuous, high computing load, noisy objective functions. PhD thesis, Université Libre de Bruxelles, Belgium, 2004.

[4] F. V. Berghen, H. Bersini, CONDOR, a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. Journal of Computational and Applied Mathematics, 181 (2005) 157-175.

[5] B. Colson, P. L. Toint, Optimizing partially separable functions without derivatives. Optimization Methods and Software, 20 (2005) 493-508.

[6] A. R. Conn, K. Scheinberg, P.L. Toint, On the convergence of derivative-free methods for unconstrained optimization, In A. Iserles and M. Buhmann, editors, Approximation Theory and Optimization: Tribute to M.J.D. Powell, Cambridge University Press, 1997, pp 83-108.

[7] A. R. Conn, K. Scheinberg, P.L. Toint, Recent progress in unconstrained nonlinear optimization without derivatives. Mathematical Programming, 79 (1997), 397-414.

[8] A. R. Conn, P.L. Toint, An algorithm using quadratic interpolation for unconstrained derivative free optimization. In G. Di Pillo and F. Gianessi, editors, Nonlinear Optimization and Applications, New York,Plenum Publishing, 1996, pp 27-47.

[9] F. Durst, M. Schäfer, A parallel blockstructured multigrid method for the prediction of incompressible flows, Int. J. for Num. Meth. in Fluids, 22 (1996), 549-565

[10] J. Ferziger, M. Perić, Computational Methods for Fluid Dynamics. Springer, Berlin, 1996.

[11] T.G. Kolda, R.M, Lewis, V. Torzcon, Optimization by direct search: new perspectives of some classical and modern methods, SIAM Review, 45 (2003) 385-482.

[12] B. Mohammadi, O. Pironneau, Applied Shape Optimization for Fluids. Oxford University Press, 2001.

[13] M. J. D. Powell, A direct search optimization method that models the objective and constraint functions by linear interpolation. In S. Gomez and J.P. Hennart, editors, Advances in optimization and numerical analysis. Kluwer Academic, 1994.

[14] M. Schäfer, B. Karasözen, Y. Uludağ, K. Yapıcı, A. Ö. Uğur, Numerical method for optimizing stirrer configurations. Computers & Chemical Engineering, 30 (2005) 183-190.

[15] K. Scheinberg. Derivative free optimization method. Technical report, IBM T.J. Watson Research Center, Newyork, Tech. Rep. Report 98/11, 2000.

[16] R. Sieber, M. Schäfer, K. Wechsler, F. Durst, Numerical prediction of time-dependent flow in a hyperbolic stirrer. In R. Friedrich and P. Bontoux, editors, Computation and Visualization of Three-Dimensional Vortical and Turbulent Flows, volume 64 of Notes on Numerical Fluid Mechanics, Braunschweig, Vieweg, 1998, pp. 318-322

[17] P. van der Lee, T. Terlaky, T. Woudstra. A new approach to optimizing energy systems. Computer Methods in Applied Mechanics and Engineering, 190 (2001) 5297-5310.

[18] A. Waechter, An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering. PhD thesis, Department of Chemical Engineering, Carnegie Mellon University, 2002