

Solution of the incompressible Navier-Stokes equations via real-valued evolutionary algorithms

R. I. Bourisli¹ & D. A. Kaminski²

¹*Department of Mechanical Engineering, Kuwait University, Kuwait*

²*Department of Mechanical, Aerospace and Nuclear Engineering, Rensselaer Polytechnic Institute, USA*

Abstract

The concept of evolutionary algorithms (EAs) is used to solve the 2-D incompressible Navier-Stokes equations. EAs operate on the principle of natural selection, where candidate solutions compete for survival and are given a chance to survive in accordance with their fitness. In an earlier paper the method was described in detail, with particular emphasis on the various evolutionary operators. In this paper, examples are given on applying the evolutionary solver to practical engineering problems such as viscous flow in channels with multiple contractions and expansions. One of the fundamental qualities of this type of solver is its relative indifference to places of high gradients in the flow field. This, in turn, helps circumvent many of the problems related to the stiffness of the system of equations. We believe the method has great value in tackling fluid flow problems where conventional methods fail to achieve timely convergence.

Keywords: evolutionary algorithms, Navier-Stokes, divergence, mutation.

1 Introduction

Many problems in computational fluid dynamics suffer from occasional divergence or slow convergence, depending on the discretization, types of boundary conditions and scale of flow phenomena. This is especially evident when continuum discretization schemes are combined with gradient-based iterative solution techniques. A good example of such convergence problem is the solution of the Navier-Stokes equations. Typical methods such as finite element and finite volume



can face such difficulties. For example, the methods can progress the solution in an acceptable manner at an acceptable rate until, for example, certain pressure modes of the solution are reached where the solution diverges suddenly, often over a very small number of iterations. There is a need for a non-standard solution technique that can take over the solution process upon incipient divergence or very slow convergence. Evolutionary algorithms are very good candidates to be such rescue techniques.

Evolutionary algorithms are stochastic search and optimization techniques that are based on the principle of natural selection [2]. They have been used extensively and successfully in many optimization problems, especially when the search domain is large and nonconvex. Here, they are used to heuristically optimize the solution to the Navier-Stokes equations by evolving a population of potential solutions and allowing natural selection to promote highly fit members of the population until an acceptable level of fitness is reached.

The use of evolutionary-type optimization algorithms in CFD is not new. However, most applications were focused on optimization of shapes for pressure drop requirements and aerodynamic performance of airfoil and like objects [3–7]. But recently, EAs as fluid flow meta-solvers have seen a promising initiation. In [1], Bourisli and Kaminski introduced a new strategy for adapting an evolutionary algorithm to act as a *go to* solver to be activated when common methods fail to achieve convergence. The method was successfully applied to a sudden expansion problem involving thousands of nodes. Subsequent research in the area followed with more applications [8,9]. In this paper, the method is applied to the full Navier-Stokes equations. The EA solver is designed to be used as a *go to* solver once the basic gradient-based solvers, which can certainly be faster, come close to failure.

2 The evolutionary algorithm

An evolutionary algorithm comprises of four basic operators to mimic the biological evolution process. Similar to biological reproduction steps in haploid organisms, a crossover operator cuts and recombines the series of arrays at a single or more points, not necessarily in the middle as in humans. Randomness is introduced via a mutation operator that changes the value of one allele randomly. To simulate nature, the algorithm requires an objective function that can differentiate between chromosomes based on their fitness. Finally, an appropriate selection scheme is used to select parents for future generations in the evolution process.

From our experience, it was clear that the simple, one-size-fits-all evolutionary algorithm is hardly efficient in solving any but the basic combinatorial problems that can be accurately cast in pseudo-binary form. In order to have an efficient search mechanism, knowledge about the problem and the expected nature of the solution must be incorporated into *specifically designed* evolutionary operators. Some authors prefer to call the resulting algorithms *evolution programs* because of the specific intended use [10]. Since fluid flow is a new area in evolutionary computations, a number of old, altered and newly conceived operators were used. These include: arithmetical and uniform crossover, uniform, nonuniform,



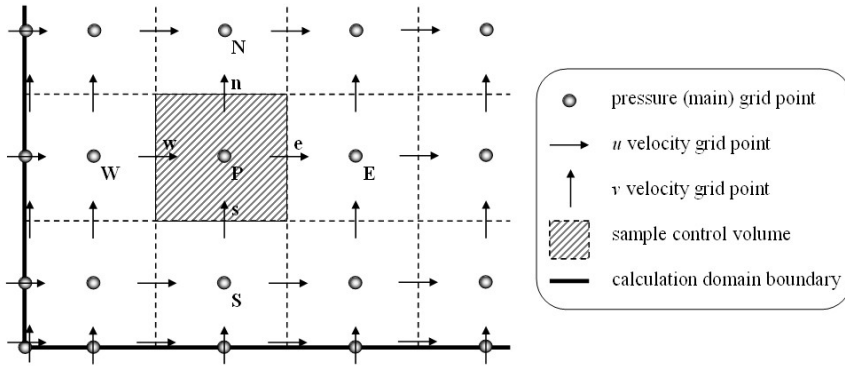


Figure 1: Sample staggered finite volume mesh.

fitness-guided, random-average, and block mutation, population shuffling, gradient-based smoothing, in addition to elitism. The various old and new components are described in detail in [8]. Here, we only discuss the objective function needed to measure relative fitness of the chromosomes.

The objective function used here is based on a staggered finite volume discretization of the flow domain, shown in Figure 1. The flow obeys the steady, incompressible Navier-Stokes equations of motion,

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (\text{continuity}) \quad (1)$$

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right] \quad (x\text{-momentum}) \quad (2)$$

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right] \quad (y\text{-momentum}) \quad (3)$$

The equations are linearized with respect to the convective terms in the momentum equations, resulting in coupled algebraic equations. With Poisson's pressure equation substituted for the continuity equation, the three equations are

$$a_P p_P = a_S p_S + a_E p_E + a_N p_N + a_W p_W + b \quad (4)$$

$$a_p u_p = a_s u_s + a_e u_e + a_n u_n + a_w u_w + A_p (p_W - p_P) \quad (5)$$

$$a_p v_p = a_s v_s + a_e v_e + a_n v_n + a_w v_w + A_p (p_S - p_P) \quad (6)$$

where the various coefficients are functions of geometry and properties of the fluid. The unknown pressures are defined on non-staggered control volumes such as the one shown in Figure 2. The detailed calculation procedures of the different coefficients are described in detail in [8].

Normally in CFD modeling, these equations are solved iteratively using an appropriate method such as TDMA while using the resulting pressure field to

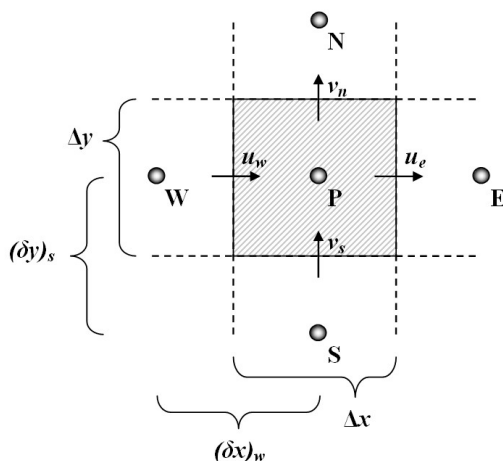


Figure 2: Sample staggered finite volume grid for calculating pressure residuals.

update velocities. The evolutionary algorithm, however, directly uses the residuals of these algebraic equations as an objective measure of the fitness of each chromosome. The resulting form of the objective function is a linear combination of the three residuals for the pressure, x -velocity, and y -velocity equations,

$$r^p = \left| -a_P p_P + a_S p_S + a_E p_E + a_N p_N + a_W p_W + b \right| \quad (7a)$$

$$r^u = \left| -a_p u_p + \sum a_{nb} u_{nb} + A_p (p_W - p_P) \right| \quad (7b)$$

$$r^v = \left| -a_p v_p + \sum a_{nb} v_{nb} + A_p (p_S - p_P) \right| \quad (7c)$$

The actual fitness of a chromosome is defined as the exponential of the maximum residual, r_m , among all interior nodes. This bounds the fitness to be in the interval $[0,1]$ and gives universal assessment of fitness values among different EA runs. In other words,

$$f = e^{-r_m} \quad \text{where,} \quad r_m = \max_{\substack{1 \leq i \leq N_x \\ 1 \leq j \leq N_y}} r_{i,j} \quad (8)$$

A population is taken to be an initially random set of potential solution to the fluid flow problem. These can be any 2D or 3D structures. For 2D problems a two-dimensional array of random real numbers is sufficient. To save effort on the evolutionary algorithm, a proper starting point is given as a gradient method output after a few iterations just enough to set the fluid in motion and provide a starting scale for the semi-random initialization of the velocities and pressure.

Table 1: EA parameters for the triple contraction/expansion simulation.

nodes	<i>Popsize</i>	P_c^a	P_c^u	P_m^r	P_m^u	P_m^a	P_m^b	shuffle
6771	40	0.2	0.4	0.01	0.01	0.001	0.001	20% every 4th

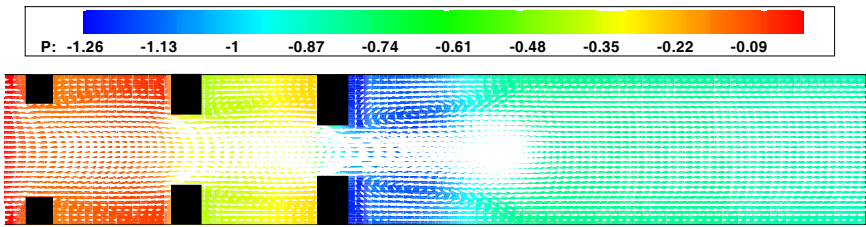


Figure 3: Velocity vectors superimposed on pressure distribution in the flow domain. Actual aspect ratio of the channel is 12-to-1, scaled for appearance.

3 Numerical results and discussion

A channel with a 12-to-1 aspect ratio with three consecutive double steps is modeled. The top and bottom steps simulate 2-to-1, 5-to-2, and 3-to-1 contractions followed by inverse expansions. The flow field is discretized using 6,771 control volumes for a total of about 20,313 unknowns. A population of 40 chromosomes was used, which is slightly higher than what was previously advised but is warranted because of the relative complexity of the present flow situation (cf. [1], where a population size of 18 was determined to be optimum for a potential flow problem.) Other parameters of the simulation are listed in Table 1.

At a Reynolds number of about 55 based on channel width, three different features of the flow are present behind the three different expansions. Specifically, behind the first mild expansion, small eddies develop immediately behind the step and smaller eddies develop in the diagonally opposite corners. As a result, the stagnation point is on the top and bottom surfaces. After the second expansion, the eddies elongate to fill the whole length until the next walls, shifting the stagnation points to the opposite walls (the contractions). After the third and last expansion, the stagnation points are in their usual place at the reattachment points.

The evolutionary algorithm was able to converge to the correct solution, as shown in Figure 3, after 127 sweeps-through, constituting 6,292 generations. The locations of the eddies were found exactly, as validated by two standard finite volume solvers.

The ability to arrive at a valid solution stresses the importance of the evolutionary operators. It is known that the EA does not actually solve any set of equations;



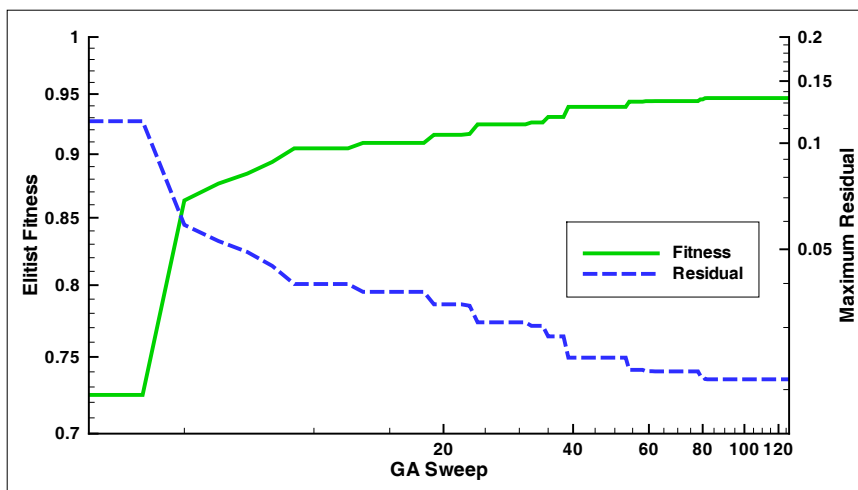


Figure 4: Progress of the fittest individual fitness and the corresponding maximum residual in the domain nodes.

it only looks at their residuals and uses them to assess the fitness of the chromosomes (or potential solutions.) Natural selection then promotes the survival of the more fit chromosomes from one generation to the next. Therefore, all the improvements in the phenotypic (relating to fitness) come from operators work on the genes themselves. This validates past conclusions that operators have to be specially designed with proper knowledge of the search domain.

The nature of the heuristic search has another important quality that can be paramount in a number of other real CFD problems. Namely, the search is indifferent to places of high gradients in the flow field, a source of computational difficulty for most algorithms. The effort needed to solve this problem is closely comparable to that observed for solving a straight channel flow with no obstacles. Many problems involving sharp changes in variables can benefit greatly from such attributes, such as flow with shockwave.

Another test for the EA was its ability to show the expected symmetry in the output solution. At times, the excessive use of one or another operator might cause a drift in the resulting genes. For example, if the block mutation was excessively applied to a subset of the domain where the building blocks were not influential, the chromosome itself would not necessarily realize the damage right away. Suppose that this chromosome has an otherwise very high fitness. Then each time it gets selected for reproduction the area of unrealistically-lowered or -raised blocks will spread to future offspring. The drift noticed in these runs of the algorithm were barely noticeable because of both the design of the operators and the low probabilities of application of most all large-scale operators.

The progress of the evolutionary algorithm is shown in Figure 4. The fitness of the elitist and the corresponding maximum residual in the flow field are plotted against the overall sweep-through number. The convergence of the EA to the acceptable threshold is noticed by the steady rise in fitness and drop in residual. The final solution, when ran through the SIMPLER finite volume algorithm, for example, gives a pressure correction term magnitude of less than 10^{-5} , which is considered convergent by any standard.

It is noted, however, by looking at the figure and from our experience, that the population maximum and average fitness do not improve considerably beyond a certain point in the search. A simple yet not entirely complete explanation is that the random search loses all ability to fine-tune highly fit chromosomes after a given level of fitness is reached. It so happened that the operators were effective enough to carry the population so far such that when convergence was reached the solution was acceptable.

The underlying cause of such loss is the inability of the individual operators to introduce enough randomness to explore the search domain efficiently. This reason has been a fundamental hurdle in the face of past research in this area. The introduction of smart operators such as fitness-guided mutation weighs the amount of randomness supplied to each chromosome in a proportional amount to its fitness. This introduces useful diversity to the population, the backbone of a successful evolution.

4 Conclusions and future work

The current work is another step toward the inclusion of evolutionary techniques in the group of dependable CFD meta-solvers. The problem of viscous flow in a channel with multiple size contractions and expansions was solved using an evolutionary algorithm. The EA optimizer succeeded in arriving at a converged solution to the problem capturing fundamental physical behavior, expected symmetry of solution, and robustness of application.

It should be noted that this problem in particular could be solved using standard gradient-based techniques, albeit with very low relaxation factors. Low relaxation factors were even needed when validation was done using the Fluent segregated solver. This was necessary because it shows that the EA solver is able to negotiate nontrivial CFD problems while we still have the chance to validate its results. The application of the EA solver to nonconvergent problems altogether is discussed in [9].

Regarding the requirement of an operator to scale its randomness up and down depending on the degree of convergence and diversity in the population, the use of a fuzzy logic is an attractive logical next step. The use of a fuzzy controller to perform crossover was explored briefly in [11, 12]. In order to achieve the much needed consistent improvement of evolution over all ranges of diversity and fitness, better control over the operators, mutation in particular, at the genotypic level is still needed.



References

- [1] Bourisli, R. & Kaminski, D.A., Solving fluid flow problems using a real-coded genetic algorithm with uniform refinement. *Advances in Fluid Mechanics*, eds. C. Brebbia, A. Mendes & M. Rahman, WIT Press: Southampton, UK, volume V, pp. 63–72, 2004.
- [2] Holland, J.H., *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [3] Davalos, R.V. & Rubinsky, B., An evolutionary-genetic approach to heat transfer analysis. *Journal of Heat Transfer, Transactions of the ASME*, **118**(3), pp. 528–532, 1996.
- [4] Fabbri, G., A genetic algorithm for fin profile optimization. *International Journal of Heat and Mass Transfer*, **40**(9), pp. 2165–2172, 1997.
- [5] Milano, M. & Koumoutsakos, P., A clustering genetic algorithm for cylinder drag optimization. *Journal of Computational Physics*, **175**, p. 79107, 2002.
- [6] Sasikumar, M. & Balaji, C., Optimization of convective fin systems: A holistic approach. *Heat and Mass Transfer*, **39**(1), pp. 57–68, 2002.
- [7] Duvigneau, R. & Visonneau, M., Hybrid genetic algorithms and artificial neural networks for complex design optimization in CFD. *International Journal for Numerical Methods in Fluids*, **44**, pp. 1257–1278, 2004.
- [8] Bourisli, R.I., *Computationally intelligent CFD: Solving potential, viscous and non-Newtonian fluid flow problems using real-coded genetic algorithms*. Ph.d. thesis, Rensselaer Polytechnic Institute, 110 8th Street, Troy, New York, 2005.
- [9] Bourisli, R.I. & Kaminski, D.A., Evolutionary optimization techniques as versatile solvers for hard-to-converge problems in computational fluid dynamics. *International Journal for Numerical Methods in Fluids*, 2006. In press.
- [10] Michalewicz, Z., *Genetic Algorithms + Data Structure = Evolution Programs*. Springer-Verlag: Berlin, 3rd edition, 1997.
- [11] Herrera, F. & Lozano, M., Fuzzy connectives based crossover operators to model genetic algorithms population diversity. *Fuzzy Sets and Systems*, **92**(1), pp. 21–30, 1997.
- [12] Herrera, F. & Lozano, M., A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *International Journal of Intelligent Systems*, **18**, pp. 309–338, 2003.

