Computational Fluid Dynamics '94

Proceedings of the Second European Computational Fluid Dynamics Conference
5-8 September 1994, Stuttgart, Germany

S. Wagner, E.H. Hirschel, J. Périaux, R. Piva
Editors

ECCOMAS

WILEY
Publishers Since 1807

Other Wiley Editorial Offices

John Wiley & Sons, Inc., 605 Third Avenue,
New York, NY 10158-0012, USA

Jacaranda Wiley Ltd, 33 Park Road, Milton,
Queensland 4064, Australia

John Wiley & Sons (Canada) Ltd, 22 Worcester Road,
Rexdale, Ontario M9W 1L1, Canada

John Wiley & Sons (SEA) Pte Ltd, 37 Jalan Pemimpin #05-04,
Block B, Union Industrial Building, Singapore 2057

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0 471 950637

Produced from camera-ready copy supplied by the authors.
Printed and bound in Great Britain by Bookcraft (Bath) Ltd, Midsomer-Norton, Avon.

Mesh Generation and Adaptive Remeshing by Genetic Algorithms on Transonic Flow Simulation

*G. Winter[1], G. Montero[1], P. Cuesta[1] and M. Galán[1]

Abstract. In this paper, we introduce different applications of genetic algorithms on transonic flow problems. Thus a regularization process of unstructured meshes is proposed. From a starting mesh, a new one is built employing genetic algorithms to minimized a fitness function which is based on geometrical conditions that allow to get better the quality of the mesh, and on error indicators providing information about its density. Several Fitness functions are suggested depending on the proposed objectives to obtain a better mesh, including different geometrical considerations regarding area, perimeter, angles, etc., of the triangles and error measurements based on the density of the fluid, the mach number or both of them. Some comparison criteria must be fixed in order to analyze the quality of the meshes. The control of the nodes is done by binary codes, assuming that they are equivalent to the chromosomes of the elements of a population. From this population, the genetic laws lead to new ones by the selection, crossover and mutation between parent chromosomes. This process is repeated till the approximate solution of the global optimum is found for the fitness function. The parameters of reproduction, crossover, mutation probabilities and size of the population must be analyzed to obtain a robust algorithm. This way, an adaptive finite element is performed moving nodes for a good remeshing.

The procedure may be generalized to three-dimensional unstructured meshes by construction of layers of triangular prism. An extension of the methodology is suggested for optimization problems over profile (design problem) with genetic algorithms.

A transonic compressible flow problem is studied for different mesh strategies, using a version of non linear GMRES.

[1]Centro de Aplicaciones Numéricas en Ingeniería,
Edif. Dept. de Ingenierías, Campus Univ. de Tafira,
Las Palmas de Gran Canaria, Spain.

Computational Fluid Dynamics '94
Published in 1994 by John Wiley & Sons Ltd

1 INTRODUCTION

Nowadays, the velocity and capability of the computers have allowed to try imitating the natural aw of the behaviour of the beings which along millions years, and generation by generation, makes the best adapted survive. The genetic algorithms were born inthe sixties, in the biochemistry field, when some biologists (see [1], [2], [3] and [4]) used digital computers to simulate genetic systems. From that moment, many works ([5], [6], [7] and [8]) began to appear, but J. Holland was who developed the basic theory of the genetic algorithms in 1975 ([9], [10], and [11]).

The main goal of this paper is the study of the mesh adaptation applying genetic algorithms. A high degree of automation of the mesh generation process is required in engineering applications with finite elements.

In the same way, a flexible and efficient method for mesh adaptation that allows to regenerate a nesh attending to several criteria would be an interesting tool for solving transonic flow problems.

In non-convex two dimensional domains, a method for smoothing and adapting meshes, initially built by a conventional mesher, has been developed with genetic algorithms.

2 TRANSONIC FLOW PROBLEM

We consider the potential flow of a compressible inviscid fluid around the airfoil S. We denote by Γ_S the boundary of S and by Ω the flow domain $R^2/\overline{S}$.

The flow is considered steady and irrotational in the region Ω. The state problem is then the full potential equation and it is modelled by,

$$-\nabla\cdot(\rho\nabla\varphi)=0 \;\; in \;\; \Omega\subset R^2 \qquad (1)$$

$$\rho\frac{\partial\varphi}{\partial n}=0 \;\; on \;\; \Gamma_s \qquad (2)$$

$$\nabla\varphi = u_\infty \quad \text{at infinity} \tag{3}$$

being

$$\rho = \rho_0\left(1-\frac{\gamma-1}{\gamma+1}\frac{|\nabla\varphi|^2}{C_*^2}\right)^{\frac{1}{\gamma-1}} \tag{4}$$

The potential φ and the velocity u are related by $u = \nabla\varphi$.

ρ_0 is the fluid density at rest.

γ is the ratio of specific heats ($\gamma = 1.4$ for air).

C_* is the critical velocity.

Relations (1)-(4) have be completed by entropy and Kutta-Joukowsky conditions in order to eliminate nonphisical solution (see [12]-[14]).

We bound the physical space by a large artificial boundary Γ_∞ on which we take,

$$\rho\frac{\partial\varphi}{\partial n} = \rho_\infty u_\infty n_\infty \tag{5}$$

where

$$\rho_\infty = \rho_0\left(1-\frac{\gamma-1}{\gamma+1}\frac{|u_\infty|^2}{C_*^2}\right)^{\frac{1}{\gamma-1}} \tag{6}$$

being n_∞ is the unit outword normal vector at Γ_∞.

The equations (1),(2),(5) define a strongly nonlinear problem of the mixed type, elliptic in the subsonic region and hyperbolic in the supersonic region. A weak solution of (1),(2),(5) is given by,

$$\int_\Omega \rho(\varphi)\nabla\varphi\,\nabla\omega\, d\Omega = \int_{\Gamma_\infty} \rho_\infty u_\infty n_\infty\, d\gamma \tag{7}$$

where ω is a test function. This formulation with resolution by combination of Newton's method and GMRES algorithm has the advantage of being applicable to flows which are supersonic at infinity through nonsymmetric indefinite preconditioning operators reflecting the dominating hiperbolic character of the operator $\nabla\cdot\rho(\varphi)\nabla\varphi$ (see [15].

The state problem is discretized with the finite element method by using triangular linear elements.

The artificial viscosity is included in the artificial density (see [16]),

$$\hat{\rho} = \rho - \mu\frac{\partial\rho}{\partial s}\Delta s \tag{8}$$

which is used in (7) in place of ρ supersonic pocket the density ρ s modified to $\hat{\rho}$. The derivative is the upwind derivative and ΔS is the mesh spacing. The equation (7) is converted into the nonlinear algebraic system of equations. The system is solved by a modified GMRES(k) (see [17]) in each time step of the Newton´s method.

2.1 Error indicators

The localized discretization error can also be estimated by the error indicators η defined on each triangle Ω_e (see [18]):

$$\eta_{\Omega_e} = h_e^{1/2}\max_{E_{\Omega_e}\cap\partial\Omega}\left\| g - \rho\left(|\nabla\varphi_h|^2\right)\frac{\partial\varphi_h}{\partial n}\right\|_E + h_e\max_{E_{\Omega_e}\setminus\partial\Omega}\left\|\left[\rho\left(|\nabla\varphi_h|^2\right)\frac{\partial\varphi_h}{\partial n}\right]_E\right\| \tag{9}$$

$$g = \begin{cases}\rho_\infty u_\infty n_\infty & on\ \partial\Omega \\ 0 & on\ \Gamma_s\end{cases} \tag{10}$$

here the bracket , $\left[\rho\left(|\nabla\varphi_h|^2\right)\frac{\partial\varphi_h}{\partial n}\right]_E$ denotes the jump of $\rho\frac{\partial\varphi}{\partial n}$ accross the edge E. Since $\nabla\varphi_h$ is constant on each triangle Ω_e these jumps are constant on each edge. One may interpret η_{Ω_e} as an approximation of the second derivatives of φ (see [18]) in order to define a second order difference of the value of $\nabla\varphi_h$ on a given triangle Ω_e associated with the center of mass x_e of Ω_e. Then one takes first order differences with neigh boring triangles in order to get a second order difference on Ω_e.

We may rewrite the sum over all triangles into a sum over all nodes of the triangulation. We use the nodal difference error indicator,

$$\eta_j = \sqrt{\sum_{\Omega_e \in \Sigma} \left(\eta_{\Omega_e}\right)^2} \quad (11)$$

which is based at the nodes z_j instead of the triangles. For further details see [19]-[22].

2.2 Applications to the optimal design

Let the profile S=S(α) be defined by a smooth function α,

$$S(\alpha) = \{(x_1, x_2) | 0 \le x_1 \le L, -\alpha(x_1) \le x_2 \le \alpha(x_1)\} \quad (12)$$

for a symmetric airfoil.

Having the solution $\varphi = \varphi(\alpha)$ on the region $\Omega = \Omega(\alpha)$ with the profile $S = S(\alpha)$, we may consider an optimal control problem:

Find $\alpha^* \in U_{ad}$ *such that* $J(\alpha^*) \le J(\alpha)$ *for all* $\alpha \in U_{ad}$ *where* $J(\alpha)$ *is a cost functional and* U_{ad} *the set of admissible controls* α.

Different cost functionals may be (see [23]:

$$J_1(|\nabla\varphi|) = \int_0^L (|\nabla\varphi| - |\nabla\varphi_0|)^2 \, dx_1 \quad (13)$$

$$J_2(|\nabla\varphi|) = \int_0^L \left(\max\left(0, \frac{-\partial|\nabla\varphi|}{\partial x_1}\right)\right)^2 dx_1 \quad (14)$$

The unknown distributed parameter $\alpha(x_1)$ is approximated by the shape parametrization,

$$\alpha_n(x_1) = \alpha_1\sqrt{x_1} + \sum_{i=2}^{n} a_i x_1^{i-1} \quad (15)$$

Many well known test profiles are expressed this way (i.e. NACA 0012).

In general the mapping $a \longrightarrow J(a)$ is noncovex. Therefore, only a local minimum of J may be found traditional methods of optimization. However, using genetic algorithms, the global minimum can be obtained.

3 MESH GENERATION

The proposed algorithm may be divided into five steps:

a. Input data determining the boundary geometry and the characteristics of the mesh.
b. Boundary discretization.
c. Generation of inner nodes in the domain.
d. Triangulation by advancing front technique.
e. Smoothing procedure applying a simple genetic algorithm.

In step a, the boundary geometry of the domain is generated using a modified cubic spline interpolation with condition of infinite derivative at the extremes, that gets a smooth representation of the model. The domain may be divided into several regions of different densities from the begining. This is useful in finite element method, i.e. when multiple materials are involved in the domain.

Steps b and c are developed following criteria of mesh density previously considered. These aspects have been studied by several authors (i.e. see S.H. Lo [24] and [25], Johnston and Sullivan [26]). Here, the boundary discretization has been done inserting nodes in each spline curve as density parameter implies. Step c has been developed taking into account a regular distribution of points in each region for a given degree of discretization. For each region, an array containing the minimum and maximun value of coordinate y for each boundary line is considered in an ordered way. Note that a simple region has an outer boundary line but it may have several inner ones (holes). Between each two points of these, some virtual horizontal lines are drawn depending on the density function. A simple virtual line is divided into several segments formed by its intersections with an outer and inner boundary line. It is easy to know where the line crosses a boundary, observing if it is located between the corresponding minimum and maximum values. When all the intersections are computed and ordered, we study if new points must be introduced between two of them, this is to say, to know if the segment belong to the domain. The discretization is done such as density function implies. This way of allocating inner nodes allows to generate triangles of good quality except for points beside the boundary nodes. In fact, a point is removed if its distance from a boundary node is less than 2/3 times the real distance corresponding at the point.

The advancing front technique of step d implemented here is as follows:

1. Create one array that contains the outer boundary points, one array for each inner boundary discretization, and another one for the inner points in the domain.
2. The first advancing line is the outer boundary.
3. For each two points of the advancing line, we search a point included in any array that allows to create the best triangle.
4. Create such triangle if it satisfies the mesh requirement. If it does not, the search is followed.
5. Introduce the point into the array defining the advancing line and remove it from the array before.
6. Finish the triangulation when there is not any array containing points.

The step e is introduced in the next paragraph.

4 MESH ADAPTATION

In triangulation processes, some triangles that do not perform the requirements about using them as elements in a finite element code, are generated. This is to say, there are degenerate triangles. In this case, several methods of smoothing are applied to get the least of degeneration. Here a new smoothing technique based on the control by artificial genetic algorithms of the nodes which form a triangle, is presented. The simple idea is to find the optimal location of a node inside a polygonal in such a way, that the triangles formed by this node and two of the polygonal are as equilateral as possible. In this sense, different fitness functions have been designed whose minimal is the searched point.

When an approximated solution is known, and thus the error indicators too, this procedure allows to build an adaptive mesh adding a relation between the size of the triangle and its associated error indicator to the fitness function such that the optimal solution of the mesh keeps a nearly constant distribution of error indicators.

4.1 Genetic algorithms

A simple genetic algorithm have three principal steps: reproduction (selection), crossover and mutation. Figure 1 shows a simple genetic algorithm (see [27]).

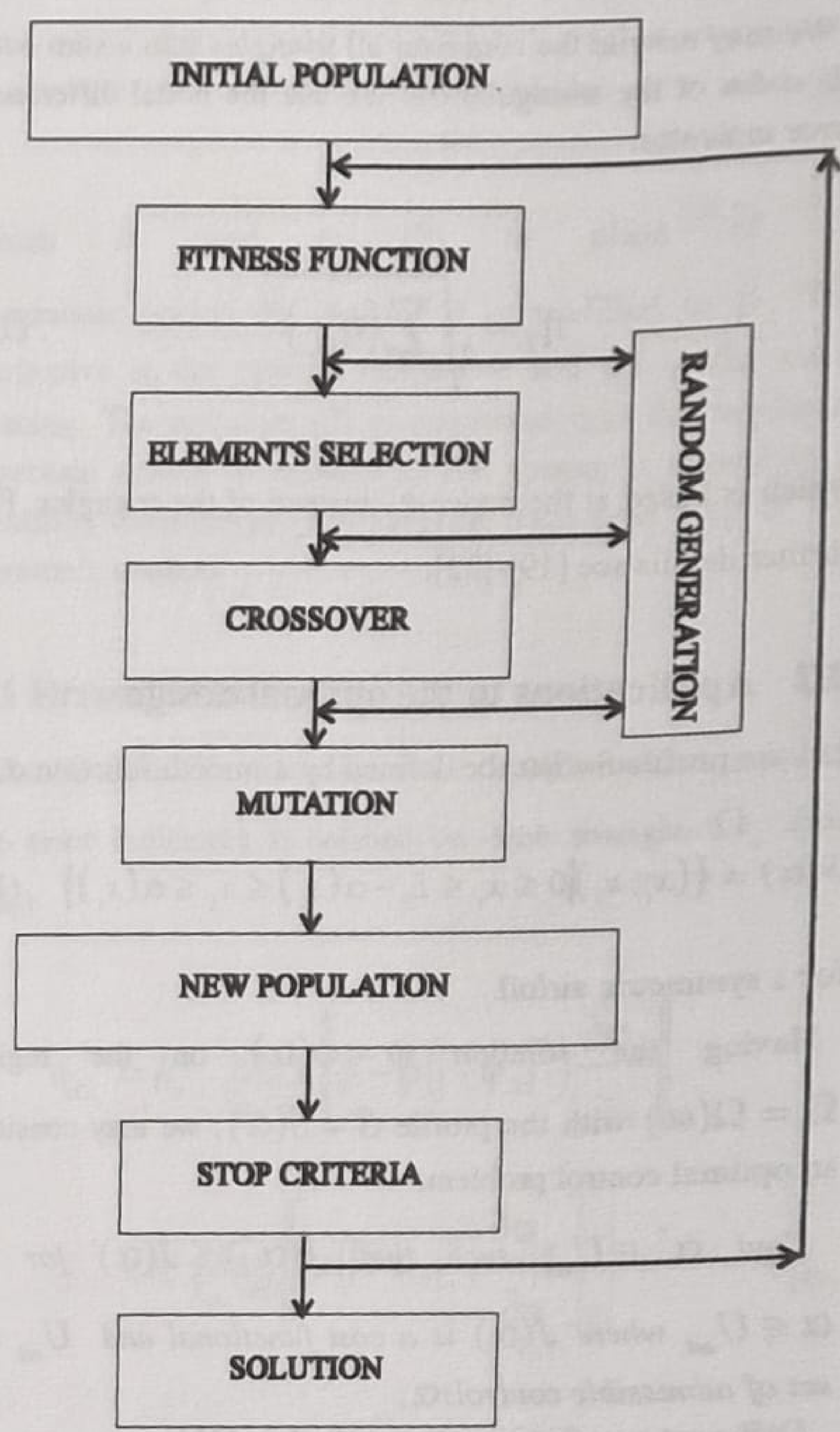

Figure 1. Simple Genetic Algorithm

4.2 Chromosomes

The geometrical representation of a point in two dimensional space requires to fix two numbers to define its location with respect to coordinates axes.

$$Location = \langle x\ y \rangle \tag{16}$$

If both values are written in binary system, a chain of 0 and 1 is obtained. Fifteen digits have been used for each coordinate value.

$$chromosome = \langle 1101100...10110 \rangle \quad (17)$$

In this conditions, each chromosome means one location in the topological space. A population is represented by a set of chromosomes which are identified depending on theirs merit values, that are obtained evaluating a function that have been fixed in order to get the searched objective.

4.3 Selection, crossover and mutation

Once chromosomes are defined, we start from an initial population taken at random. Each chromosome is computed and, agreeing with it, a criteria of selection is stablished. This is a delicated aspect of the genetic algorithms, depending on it that the convergence to the optimal location is got at an admisible time or out of reasonable limits. In this research, a stochastic model was applied based on a congruential method. Supposing that two parents are selected, the crossover operation consists of the following: if we have two chains, both being of n bits, a random number k is chosen in the interval (1,n-1) and two sons are created interchanging the chains between 1 and k inclusive. Let be the parents,

$$parent1 = \langle a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} \rangle \quad (18)$$

$$parent2 = \langle b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10} \rangle \quad (19)$$

where a_i and b_i represent 0 or 1. If we obtain, i.e., k=6 at random, two sons are created from crossover operation,

$$son1 = \langle b_1 b_2 b_3 b_4 b_5 b_6 a_7 a_8 a_9 a_{10} \rangle \quad (20)$$

$$son2 = \langle a_1 a_2 a_3 a_4 a_5 a_6 b_7 b_8 b_9 b_{10} \rangle \quad (21)$$

The last step of the transforming operartion is constituted by the mutation process, which consists of choosing a random number k' belonging to the interval (0,n-1), and if 1 corresponds to location k' we allocated 0, and vice versa. Mutation increases the possibilities of amplifying the space of search to zones that stay hidden in other way. As in crossover operation, there is a parameter that regulates the chance of mutation of the chromosomes.

Although it is not going to be exposed here, the optimization by selection procedure for reproduction, crossover and mutation, may be understood in a more rigorous way, examining from the begining of populations the apparition of robust schemes of bits in the chromosomes that grow generation by generation making the population go to the optimum which is got by every member.

4.4 Random number generation

The effect of the fitness function to select an element for reproduction should be take into account following the expresion:

$$p_i = \frac{f_i}{\sum f_j} \quad (22)$$

where f_i is the value of the fitness function of the element i, and $\sum f_j$ is the sum of fitness function for all the elements.

It provides a density function of probability of the elements which indicates the frecuency of its reproduction.

As soon as this frecuency is obtained, the reproduction takes place by crossing over the elements. A random selection should be made efficiently. After crossover, a random location is computed to raise next generation. In an other hand, mutation is the process where a bit is randomly changed with some probability.

We can conclude that random number generation is very important in selection, crossover and mutation. A good selection of the random number generation algorithm is necessary for rate and convergence of the method. This algorithm should satisfy, first, that the random chain of n elements is sufficiently long (in our problem it is very long), and second, random numbers fatisfy random test. The capacity of the computer is related with the random chain of n numbers and its quality.

In the present paper the Miran Package (see [28]) is used to generate random numbers between high and low ones. It has two subroutines: one initializes random chains and the other generates it. Initialization is made using integer numbers between 1023 and 2^{94}, although this value could be extended if it is necessary. The generation method is known as congruential method and it can be used in any computer.

4.5 Fitness function

For smoothing the mesh, the following objective functions have been chosen (see [29]):

a. First, a function that allows the central angles to be equal.

$$F_a = \sum_{i=1}^{NP} \sin^2\left(\frac{2\pi}{NP} - \alpha_i\right) \qquad (23)$$

being NP the number of vortices of the polygonal containing the point to locate, and α_i the angle corresponding to the inner vortex of the triangle i.

b. Tne second function allows the triangles to be as isosceles as possible.

$$F_b = \sum_{i=1}^{NT} (C_{1i} - C_{2i})^2 \qquad (24)$$

being NT the number of triangles of the polygonal, and C_{ji} the distance between the inner and the j-th point of the i-th triangle.

c. The regeneration of the mesh (adaptation) is given by a fitness function that changes the density in those zones where the error indicators are higher, keeping the initial global number of nodes in the mesh.

$$F_c = \sum_{i=1}^{NP} A_i \varepsilon_i \qquad (25)$$

being NP the of vortices of the polygonal, A_i the area and ε_i the error indicator of i-th triangle.

Taking this fitness function, it is possible that the performed triangles are not up to use in a finite element code due to geometrical degeneration. This leaded us to apply a combination of two fitness functions of different characteristics, including both geometrical and minimal error conditions,

$$F_1 = \theta F_a + (1-\theta) F_c \qquad (26)$$

$$F_2 = \theta F_b + (1-\theta) F_c \qquad (27)$$

being $0 \le \theta \le 1$.

In all the cases, the parameter θ has been chosen to be 1 at first (geometrical regularization) and $0.1 \le \theta \le 0.2$ in the following steps (adaptation).

5 CONCLUSIONS

Being the main goal of this paper to present new possibilities of resolution with genetic algorithms in different problems associated to Transonic Flow Simulation, we have opted here for showing extensively several questions of interest concerning to the capability of the genetic algorithms in this field.

Thus we have proposed a regularization process of unstructured meshes and adaptive remeshing by genetic algorithms on Transonic Flow Simulation.

Also, an optimum shape design problem is suggested with GAs.

Different numerical experiences have been developed in the Center of Numerical Applications in Engineering (CEANI) of Las Palmas G.C. University, some of them with good results, to be published in the next future in several reviews.

REFERENCES

[1] N. A. Barricelli, `Symbiogenetic evolution processes realized by artificial methods´, *Methodos*, **9**, (35-36), 143-182, (1957).

[2] N. A. Barricelli, `Numerical testing of evolution theories´, *Acta Biotheoretica*, **16**, 69-126, (1962).

[3] A. S. Fraser,`Simulation of genetic systems´,*Jour. Theor. Biol.*, **2**, 329-346, (1962).

[4] F. G. Martin and C. C. Cockerham, *High epeed selection study*, 35-45, in O. Kempthorne (Ed.), Biometrical Genetics, Pergamon Press, London, 1960.

[5] J. H. Holland, *Adaptation in natural and artificial*, University of Michigan Press, 1975.

[6] J.H. Holland, *Processing and processors for schemata*, 127-146, in E.L. Jacks (Ed.), Associative Information Processing, American Elsevier, New York, 1971.

[7] J.H. Holland, `Genetic Algorithms and the optimal allocations of trials´, *SIAM Jour. Comput.*,2 (2), 88-105, (1973).

[8] J.H. Holland, *Schemataand intrinsiclally parallel adaptation*, 43-46, in Proc. NSF Workshop of Learning Systems Theory and its Applications, Gainesville, University of Florida, 1973.

[9] G. Levitin and J. Rubinovitz, `Genetic algorithm for linear and ciclic assignment problem´, *Computers Ops. Res..*, **20**, 6, 575-586, (1993).

[10] O. Friedrich, `Anew method for generating inner points of triangulation in two dimension´, *Comp. Meth. Appl. Mech. Eng.*, **104**, 77-86, (1993).

[11] H. Zhang, M. Reggio, J. Y. Trepanier and R. Camarero, `Discrete form of the GCL for moving meshes and its implementation in CFD schemes´, *Computers Fluids*, **22**, 1, 9-23, (1993).

[12] M.O. Bristeau, R. Glowinski, J. Periaux, P. Perrier and G. Poirier,`Transonic Flow simulation by finite element and least square methods´, *Finite elements in Fluids*, **4**, 453-482, (19982).

[13] M.O. Bristeau, R. Glowinski, J. Periaux, O. Pironneau and G. Poirier,`On the numerical solution of nonlinear problems in fluid dinamics by least square and finite elements methods (II). Application to transonic flow simulations´, *Comp. Meth. Appl. Mech. Eng.*,**51**, 363-394, (1985).

[14] M.O. Bristeau, R. Glowinski, J. Periaux, P. Perrier, O. Pironneau and G. Poirier,`Transonic flow and Shock Waves: least-squares and conjugate gradient method´, *Finite Elements Handbook*, **3**, 3229-3243, (1987).

[15] Q.V. Dinh and J. W. He,`A cartesian grid finite element method for potential flows´, *High Performance Computing*, **2**, 295-306, (1991).

[16] M. Hafez and J. Ahmad,`Design of airfoil in transonic flows´, *Twelfth Int. Conf. Num. Meth. Fluid Dyn.*, 389-393, (1990).

[17] M. Galán, G. Montero and G. Winter,`A diret solver for the least square problem arising from GMRES(k)´, to appear in *Com. Num. Meth. Eng.*

[18] U. Gohner, G. Warnecke and W. Wendland,`Error Indicators and adaptive finite elements for transonic flows´, *First European Comp. Fluid Dyn. Conf.*, 667-675, (1992)

[19] U. Gohner, *Adaptive finite element-methoden fur transsonische estromungen*, Thesis Universitat Sttudgart, 1991..

[20] U. Gohner, *Adaptive finite element methodsfor transinic flows*, Manuscript, 1992.

[21] U. Gohner and G. Warnecke, *A second order difference error indicator for adaptive transonic flowcomputations*, Manuscript, 1992.

[22] U. Gohner , G. Warnecke and W. Wendland, *Error indicators for adaptive transonic flow computation*, Manuscript, 1992.

[23] R. A. E. Makinen, ` Optimal shape design for transonic potential flows´, *CIMNE*, 457-466, (1993).

[24] S. H. Lo, ` A new mesh generation scheme for arbitrary domains of the finite element method´, *Int. Jour. Num. Meth. Eng.*, **21**, 1403-1426, (1985).

[25] S. H. Lo, ` Delaunay triangulation of non-convex planar domains´, *Int. Jour. Num. Meth. Eng.*, **28**, 2695-2707, (1989).

[26] B. P. Johnston and J. M. Sullivan Jr., `Fully automatic two dimensional mesh generation using normal off setting´, *Int. Jour. Num. Meth. Eng.*, **33**, 425-442, (1992).

[27] E. Goldberg, *Genetic Algorithms in search, Optimization and Machine Learning*, Addison-Wesley, 19989.

[28] J. R. Simpson, *Random Number Generation for Selected Probability Distributions*, Scientific Officer, Office of Naval Research, 1975.

[29] P. D. Cuesta, M. Galante and G. Montero,` Optimización de mallas generadas poir avance frontal através de un algoritmo genético simple´, *Proc. Cong. Met. Num. Ing. Cinc. Aplic.*, 981-987, (1992).