SMOOTHING OF NON-STRUCTURED GRIDS USING GENETIC ALGORITHM

Abstract.

In this paper an optimization method for triangular non-structured grids using genetic algorithm is presented, in order to find the best position for the grid vertexes. Such position is based in the fitness score, which is evaluated to each internal vertex of the grid. This score is based on the average quality of volumes which share the vertex. The algorithm is executed to a grid created by the Delaunay triangulation and the results show an improvement of the grid in relation to the studied quality parameter. However, the method is computationally more expensive than other smoothing methods.

1. Introduction

The study of numerical accuracy in Computational Fluid Dynamics (CFD) has an important role with the widespreading of numerical simulations. Several methodologies can be applied to solve CFD problems, and one of the most used is the Finite Volume Method (FVM). Such method consists on the integration of the mathematical model for each control volume in which the domain is discretized. This discretization depends on the geometry of domain and the generated grid can be classified as structured or non-structured. The major advantage of non-structured grids places on their adaptability to complex geometries [1].

For non-structured grids, in two-dimensional problems, triangles and four-sided elements are commonly used for domain discretization. One of the most used triangulation technique is the Delaunay triangulation [2,3].

Although there are several implemented tools for the automatic grids generation, in several cases they present low quality levels. Both in tessellation and in grid refinement, it is possible that strongly distorced elements are created. Even when an uniform grid is desired, the tessellation tools can generate too small or too big elements, when compared to the other ones [4].

According to Zhou and Shimada [4], there are serveral schemes of grid suavization, such as the Laplacian smoothing and smoothing based on optimization tools. Each method present different performances with respect to mesh quality and computational efforts. For example, Laplacian smoothing presents a small computational effort, on other hand, it can produce poor quality meshes or even invalid mesh elements. Nevertheless, smoothings based on optimizations are more capable to avoid invalid elements, obtaining better grid quality; however, their computational efforts are higher than the ones of Laplacian smoothing.

* 1. Genetic algorithms

Genetic algorithms (GA’s) are adaptive methods that can be used to solve problems involving searching and optimization. They are based on genetic processes of biological organisms. By several generations, natural populations evolve according to natural selection and the survival of the fittest, enounced by Charles Darwin in “On the origin of species”. Based on this process, the genetic algorithm is able to “evolve” solutions for the real world, since the problems are correctly codified. For example, GA’s are used to draw the structure of bridges in order to achieve the best force/weight ratio, or to obtain the minimal losses in the fabric cut. GA’s are also used in online processes, such as the ones found in chemical factories and the load balancing of multiprocessor systems [5].

GA’s uses a direct analogy to the nature. Individuals of a given population, which represent possible solutions of a specific problem, receive an adaptability level according to their performance as solutions. Such level is known as fitness. Individuals with high fitness values present higher probabilities to reproduce themselves, i.e., to pass their features to the new populations. In the other hand, individuals with low fitness levels present smaller chances to reproduce themselves, and their features will be extinct in future.

In a genetic algorithm, a potential solution to a problem can be represented as a group of parameters. Such parameters (known as genes) are joined forming a chain of values, which are commonly called as chromosomes. Holland [6] was the first to show (and many authors also believe) that the ideal is the use of a binary alphabet to represent this chain of values. For example, if the problem is maximizing a function of three variables, F(*x*, *y*, *z*), each variable can be represented by a 10-bits binary number; in such case, the chromosomes present three genes and 30 binary numbers. However, according Janikow and Michalewicz [7], it is possible to represent such variables using a codification of floating point, and for this representation the algorithm become faster, more consistent and present a higher precision, especially for great domains, in which a binary codification would become too much long.

In genetic terms, a group of parameters represented by a given chromosome is called genotype. The genotype contains all needed information to build an individual – which is called phenotype.

1.2. Optimization and grid smoothing

Most smoothing methods for non-structured grids involve a reposition of individual vertices, which will improve the quality of elements, such as observed in Laplace smoothing [8]. The grid quality is an important factor to improve the accuracy of numerical solutions, as shown by Juretic [1]. Considering the features related to this quality and a triangular non-structured grid, some different quality criteria can be defined. Such criteria are based on the fact that the ideal smoothing must get the ideal triangle element, which is the equilateral one. Another commonly adopted quality factor is described by Falsafioon et al. [9] and is evaluated for each grid element as

 (1)

in which: is the quality factor evaluated for an volume element *i*;  is the surface area of this volume;  are the faces lengths of such volume.

There are several types of smoothing methods as presented by Canann et al. [10]. It is also possible the use of generalized linear programming as discussed by Amenta et al. [11].

1. Methodology

The employed method consists in the generation of a grid by using the Delaunay’s triangulation, which generates high quality grids. Such method is discussed by Shewchuk [3] and Lin et al. [12].

After the grid generation, some triangular elements with poor quality are observed. In order to improve the grid quality, an implemented genetic algorithm is executed for each internal vertex of such grid. The objective is determining the position of such vertex which presents the highest possible quality factor; such factor considers the neighbour triangulations to the vertex. Therefore the genetic algorithm uses two variables related to Cartesian coordinates, one to x-position and other one to y-position. These coordinates can be seen as genes of each individual, while the group of two genes form the genotype and the value of the quality factor is the phenotype.

* 1. Algorithm

The algorithm starts with the generation of initial population, which contains random solutions for vertices positions. Such population present the following parameters:

* Size of population: 300;
* Number of generations: 40;
* Crossover probability: 0.8;
* Mutation probability: 0.01.

These are arbitrary parameters and they are commonly empirically adjusted according to each problem.

After the choice of initial population, the fitness score is evaluated for each population solution, according Eq. 1, and the best one is selected. In the sequence, a structure known as Mating Pool is provided; such structure is filled with the solutions according to their fitness, i.e., if a solution presents high fitness score it will appear more times in the structure while solutions with low fitness score appear less times.

Once the Mating Pool is defined, a new generation is created by the combination of two random selected parents, which generate a child by the combination of their genotypes. Such combination is known as Crossover. Using the genes codification with floating point, a random number of genes of parent1 is combined with the respective genes of parent2; it is made for each involved gene. Such procedure allows the creation of a new generation, which is defined by a new population, which is expected to present, in average, better fitness scores and, because of this it converges to the problem solution. A summary of the methodology using genetic algorithms is provided by Kumar et al. [13] and consists basically by:

* Initialization: Firstly, several solutions are randomly generated in order to perform the initial population. The size of such population depends on the problem features, but typically it consists of hundreds of thousands possible solutions. Traditionally, the generated population is randomly created and covers the whole possible solutions space. Occasionally, a specific region of the space is prioritized, since it has a higher probability to contain the ideal solution.
* Selection: A group of the current population is selected to generate the new generation. The chosen individuals are selected by their fitness scores. Some methods evaluate the fitness score of all generation while other ones select only some random individuals; these last methods, however, can spend much more time. Commonly the fitness function is stochastically chosen and the selection is made even if a solution present low fitness score – in this case, however, this solution has lower chances to be chosen. Such strategy is interesting to keeping high the genetic diversity of the population, preventing the premature appearing of bad solutions. The most studied and employed selection strategies include the roulette whell [14] and the tournament selection [15].
* Reproduction: The next step consists in the generation of new individuals of a population, which will replace the old ones. The reproduction is made using the genotype of select parents, recombining them by crossover and mutation processes. Once the generated “child” has genotypes of both parents, it will share several features with them. New parents are chosen for each child and this process continues until all the population is generated. Although typically only two parents are chosen for each child, some researchers [16] suggest that more than two parents could produce a child with better a genotype.

The evolutionary process of new solutions (individuals) goes on until a defined stop criterion is achieved, which can be:

* A solution is found and it satisfies a minimum criterion;
* The limit number of generations is achieved;
* The limit computational time is achieved;
* The solution with the best fitness score of population keeps constant and new generations cannot improve the result;
* Manual inspection;
* A given combination of previous criteria is observed.

In the current study, one stop criterion is based on the number of generations, which is fixed as 40. In the end of evolutionary process, the individual with best fitness score is achieved and the genotype of such solution is used to update the value of such vertex. The process goes to the next grid vertex, until the update of all vertexes with this algorithm.

1. Results

The optimization algorithm based on the presented method was implemented for two geometries and the results are shown in Figures 1 and 2. In order to better analyze results, for each grid parameters of: highest angle, smallest angle, standard deviation of angles, best quality of volumes, lowest quality of volumes, standard deviation of quality and average quality are presented in Tables 1 to 4.

Figures 1 and 2.

Table 1. Original Grid 1

|  |  |  |
| --- | --- | --- |
| Value | Angle | Quality |
| Average | 60 | 0.830759 |
| Standard deviation | 22.868034 | 0.193268 |
| Minimum | 9.554770 | 0.076371 |
| Maximum | 157.010795 | 0.999923 |

Table 2. Grid 1 with GA

|  |  |  |
| --- | --- | --- |
| Value | Angle | Quality |
| Average | 60 | 0.848804 |
| Standard deviation | 21.527223 | 0.189446 |
| Minimum | 5.504031 | 0.041763 |
| Maximum | 161.853212 | 0.999185 |

Table 3. Original Grid 2

|  |  |  |
| --- | --- | --- |
| Value | Angle | Quality |
| Average | 60 | 0.873503 |
| Standard deviation | 17.606637 | 0.102456 |
| Minimum | 26.888982 | 0.604936 |
| Maximum | 109.030637 | 0.999390 |

Table 4. Grid 1 with GA

|  |  |  |
| --- | --- | --- |
| Value | Angle | Quality |
| Average | 60 | 0.900385 |
| Standard deviation | 15.643499 | 0.908648 |
| Minimum | 24.665694 | 0.611293 |
| Maximum | 106.827269 | 0.998796 |

1. Conclusions

Two triangular non-structured grids were generated by Delaunay triangulation. In the current work an algorithm based on the genetic algorithm was implemented, optimizing the position of internal vertexes of grid based on a quality factor. The results show that, even for a grid generated by the Delaunay triangulation, which is efficient to create triangular grids, the implemented method is able to improve the average quality of volumes of this grid.