M04

Proceedings of the 5th International Conference on Inverse Problems in Engineering: Theory and Practice, Cambridge, UK, 11-15th July 2005

COMPARISON OF A GENETIC ALGORITHM AND A GRADIENT BASED OPTIMISATION TECHNIQUE FOR A THREE-DIMENSIONAL BOUNDARY DETECTION PROBLEM

N.S. MERA[1], L. ELLIOTT[2], D.B. INGHAM[2] and D. LESNIC[2]
[1] *Centre for Computational Fluid Dynamics, University of Leeds, Leeds, LS2 9JT, UK, N.S.Mera@leeds.ac.uk*
[2] *Department of Applied Mathematics, University of Leeds, Leeds, LS2 9JT, UK lionel@amsta.leeds.ac.uk, amt6dbi@amsta.leeds.ac.uk, amt5ld@amsta.leeds.ac.uk*

Abstract - In this paper we consider the determination of the shape of an unkown portion of the boundary of a three-dimensional solution domain from Cauchy data on the remaining portion of the boundary. This problem arises in the study of quantitative non-destructive evaluation of corrosion in materials in which boundary measurements of currents and voltages are used to determine the material loss caused by corrosion. The domain identification problem is considered as a variational problem to minimize a defect functional, which utilises some additional data on certain known parts of the boundary. A sequential quadratic programming (SQP) optimisation algorithm and a real coded genetic algorithm (GA) are used in order to minimise the objective functional. The unknown boundary is parameterized using B-splines. The Laplace equation is discretised using the method of fundamental solutions (MFS). Numerical results are presented and discussed for several test examples.

1. INTRODUCTION

Shape optimisation problems are important in determining the domain of corroding materials and the location of cracks in electrical conductors for the optimal design of aerospace structures and airfoil wings, optimisation of electromagnets, thermal tomography and for many other applications, see [19]. Such problems are highly nonlinear and they are well-known to be ill-posed and require special techniques in order to be accurately solved numerically.

In this paper the problem of boundary detection is reformulated as an optimisation problem and the unknown boundary is sought in a B-splines parametric form. Thus the problem is regularized by using the function specification method, i.e. by assuming a parametric form of the unknown function and reducing the problem to the determination of a reduced number of parameters, i.e. the control points of the B-splines surface.

We note that once the problem has been formulated as an optimisation problem then various optimisation algorithms may be used in order to locate the optimum of the objective function. The efficiency of a particular optimisation method clearly depends on the form of the objective function. In the problem considered in this paper, the objective function has a complex nonlinear and nonmonotonic structure. Genetic algorithms (GAs) have been successfully applied to nonlinear optimisation problems where more traditional methods are often found to fail. Deterministic, gradient based optimisation methods do not search the full parameter space and can tend to converge towards local extrema of the fitness function, which is clearly unsatisfactory for problems where the fitness varies nonmonotonically with the parameters. On the other hand in vicinity of local optima the rates of convergence exhibited by the gradient based algorithms clearly outperform those of evolutionary algorithms.

Therefore it is the purpose of this paper to investigate and to compare the performances and limitations of deterministic and evolutionary algorithms for corrosion damage detection in materials.

2. MATHEMATICAL FORMULATION

In this paper the EIT is used to determine material loss occurring on the inaccessible portion $\gamma \subset \partial\Omega$ of the boundary $\partial\Omega$ of a domain $\Omega \subset \mathbb{R}^3$ by measuring voltages and currents (Cauchy data) on an accessible portion of the boundary. Therefore, we consider the Laplace equation in a three-dimensional damaged finite plate Ω for the potential u, namely,

$$\nabla^2 u = 0, \qquad \text{in} \quad \Omega \tag{1}$$

subject to the boundary conditions

$$u = g, \quad \frac{\partial u}{\partial n} = q \quad \text{on} \quad \Gamma \tag{2}$$

$$\alpha_0 u + \beta_0 \frac{\partial u}{\partial n} = f \quad \text{on} \quad \Sigma \tag{3}$$

$$\alpha_1 u + \beta_1 \frac{\partial u}{\partial n} = h \quad \text{on} \quad \gamma \tag{4}$$

where $f, g, h, q, \alpha_0, \beta_0, \alpha_1$ and β_1 are prescribed functions and n is the outward normal to the boundary $\partial\Omega$. In eqns (2)−(4) the boundaries γ, Γ and Σ are disjoint and $\gamma \cup \Gamma \cup \Sigma = \partial\Omega$. The problem investigated consists of determining the shape of the unknown boundary γ given the boundary data specified by eqns (2)−(4). This problems occurs in several contexts, such as corrosion detection by electrostatic measurements, see [11].

The uniqueness in determining γ follows from the unique analytical continuation property for the Laplace equation, see [5], and other existence and uniqueness results can be found in [3]. An important theoretical issue for this problem is the stability of the solution. Unfortunately, this inverse problem is severely ill-posed [4]. Stability results in the form of logarithmic stability estimates are available, see Alessandrini and Rondi (1998) and Isakov (1993). Alessandrini [1] showed that the most one can hope for is logarithmic continuous dependence of γ on the Cauchy data and furthermore, this stability does not improve with additional boundary measurements. Several computational results for the two-dimensional case can be found in the literature, using, for example, Tikhonov regularization in connection with the L-curve method [12]. Charton *et al.* [6] proposed a variational technique based on parameterising the unknown boundary using the function specification method. For the three dimensional case a sequential quadratic programming method of solution was developed in Mera and Lesnic (2004) while the same problem was investigated using a genetic algorithm in [14]. It is the purpose of this study to compare these two approaches.

3. THE METHOD OF SOLUTION

3.1 Reformulation as an optimisation problem

The boundary detection problem given by eqns (1)-(4) can be reformulated as an optimisation problem. For a given possible solution γ for the unknown boundary, the following direct problem is solved:

$$\nabla^2 u = 0, \qquad \text{in} \quad \Omega \tag{5}$$

$$u = g, \qquad \text{on} \quad \Gamma \tag{6}$$

$$\alpha_0 u + \beta_0 \frac{\partial u}{\partial n} = f \quad \text{on} \quad \Sigma \tag{7}$$

$$\alpha_1 u + \beta_1 \frac{\partial u}{\partial n} = h \quad \text{on} \quad \gamma \tag{8}$$

and the current flux on the known boundary $u'_{calc} = \left.\frac{\partial u}{\partial n}\right|_{\Gamma}$ is evaluated. The solution of the problem may be found by minimising the functional

$$J(\gamma) = \|u'_{calc} - q\|_{L^2(\Gamma)} \tag{9}$$

where q is the measured current flux on the boundary Γ. The boundary γ can be parameterised in different forms and the parameters characterising the shape of the boundary are determined by minimising the functional (9). In this paper we have used B-splines to parameterise the unknown surface γ.

3.2 The parametrisation of the boundary by B-splines surfaces

The unknown boundary γ is sought in the parametric form

$$\gamma(v, w) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,k}(v) N_{j,l}(w) \mathbf{P}_{ij}, \qquad (v, w) \in [0, 1] \times [0, 1] \tag{10}$$

where $N_{i,k}$ and $N_{j,l}$ are the k^{th} and l^{th} degree B-spline basis functions given by

$$N_{i,0}(v) = \chi_{[v_i, v_{i+1}]}; \quad N_{i,k}(v) = \frac{v - v_i}{v_{i+k} - v_i} N_{i,k-1} + \frac{v_{i+k+1} - v}{v_{i+k+1} - v_{i+1}} N_{i+1,k-1} \tag{11}$$

$$N_{j,0}(w)=\chi_{[w_j,w_{j+1}]};\quad N_{j,l}(w)=\frac{w-w_j}{w_{j+l}-w_j}N_{j,l-1}+\frac{w_{j+l+1}-w}{w_{j+l+1}-w_{j+1}}N_{j+1,l-1} \tag{12}$$

defined on the nonperiodic, nonuniform knot vectors $V=(v_i)_{i=0,n+k+1}$ and $W=(w_i)_{i=0,m+l+1}$, see [17]. The points $\mathbf{P}_{ij}$ are the control points of the B-spline surface γ and they determine the shape of this boundary. Thus, the problem of boundary detection is reduced to determining the $(m+1)(n+1)$ control points $\mathbf{P}_{ij}$. The x and y coordinates of the control points are considered to be known while the z coordinate is to be detected. In the problem considered in this paper we assume that the intersection between the boundary γ and Σ is known, i.e. we know the curves that form the boundary of the surface γ but we don't know it's internal shape. Thus, the control points $\mathbf{P}_{ij}$ with $i\in\{0,n\}$ or $j\in\{0,m\}$ are known and the only unknowns are the z coordinates of the remaining control points. Thus, the problem of boundary identification is reduced to determining the components of the matrix

$$\mathbb{Z}\left(z_{P_{ij}}\right)_{i=\overline{1,n-1},j=\overline{1,m-1}}=\begin{bmatrix} z_{P_{11}} & z_{P_{12}} & \dots & z_{P_{1,m-2}} & z_{P_{1,m-1}} \\ \dots & \dots & \dots & \dots & \dots \\ z_{P_{n-1,1}} & z_{P_{n-1,2}} & \dots & z_{P_{n-1,m-2}} & z_{P_{n-1,m-1}} \end{bmatrix} \tag{13}$$

It should be noted that the ill-posed problem of identifying the boundary of the domain is regularized by using the function specification method, i.e. by assuming a pre-defined shape and reducing the problem to the determination of the parameters of the B-spline surface. In the more general case when no parametric shape is assumed for the boundary γ then the problem is ill-posed and regularization terms must be included in the objective functional (9). By using such regularization terms that penalize non-smooth solutions, and choosing the appropriate regularization parameter, a stable numerical solution can be obtained.

3.3 The method of fundamental solutions

A meshless method, namely the method of fundamental solution, [8], is used to solve the direct problem (5)-(8) and to evaluate the objective function (9) for a given boundary γ. The numerical solution is developed by using the fundamental solution of the Laplace equation as a basis function.

In order to introduce the method of fundamental solutions, we consider the entire domain Ω and we consider a set of M nodal points $\underline{x}_i$, $i=\overline{1,M}$ outside Ω. The fundamental solution of the three-dimensional Laplace eqn.(5) is given by:

$$F(\underline{x})=\frac{1}{4\pi\|\underline{x}\|} \tag{14}$$

Then the fundamental solutions centered at the nodal points $\underline{x}_i$, $i=\overline{1,M}$ given by

$$\phi_i(\underline{x})=F(\underline{x}-\underline{x}_i)=\frac{1}{4\pi\|\underline{x}-\underline{x}_i\|} \tag{15}$$

satisfy the partial differential eqn.(5). The method of fundamental solutions is based on the fact that an approximation to the solution of the direct problem (5)-(8) can be expanded in terms of these basis functions, ϕ_i, i.e. the solution is sought in the form:

$$u^*(\underline{x})=\sum_{i=1}^{M}\alpha_i\phi_i(\underline{x}) \tag{16}$$

where α_i are unknown coefficients. For this choice of basis functions the approximated solution u^* already satisfies the Laplace eqn.(5) and the coefficients α_i, $i=\overline{1,M}$, are determined such that u^* satisfies the boundary conditions (6)-(8).

We consider N collocation points $\underline{x}_j$, $j=\overline{1,N}$ on the boundary of the domain Ω. Using the given values of the voltage (or current flux) at these N collocation points, then the resulting system of linear algebraic equations for the unknown coefficients α_i is obtained by collocating (16) at the points $\underline{x}_j$, $j=\overline{1,N}$ and a system of linear algebraic equations,

$$A\underline{\alpha}=\underline{b} \tag{17}$$

is obtained, where the vector $\underline{\alpha}$ contains the unknown coefficients α_i, $i=\overline{1,M}$. It should be noted that in order to uniquely determine the coefficients α_i, $i=\overline{1,M}$ the number of collocation points N must be greater or equal to the number of nodal points M.

However, the system of linear algebraic equations is ill-conditioned and cannot be solved by direct

methods, such as the least squares methods, since such an approach would produce a highly unstable solution. Most standard numerical methods cannot achieve good accuracy in solving the matrix eqn.(17) due to the large value of the condition number of the matrix A which increases dramatically with respect to increasing the total number of collocation points and nodal points. Several regularization methods have been developed for solving these kind of ill-conditioned problems, [9]. In this study we use the standard Tikhonov regularization method, [20]

The Tikhonov regularized solution $\underline{\alpha}_\lambda$ for eqn.(17) is defined as the solution of the following least squares problem:

$$\min_{\underline{\alpha}} \{ \| A\underline{\alpha} - \underline{b} \| + \lambda^2 \| \underline{\alpha} \| \} \tag{18}$$

where $\| \cdot \|$ denotes the usual Euclidean norm and λ is the regularization parameter.

The choice of a suitable value of the the regularization parameter λ is crucial for the accuracy of the final numerical solution and there are several methods for determining an optimal value for this parameter, such as the L-curve method, [8], the discrepancy principle, [19], etc. In this study we use the L-curve method, i.e. we define the curve

$$L = \{ (\ln(\| \underline{\alpha}_\lambda \|), \ln(\| A\underline{\alpha}_\lambda - \underline{b} \|)) , \lambda > 0 \} \tag{19}$$

This curve is known as the L-curve and a suitable regularization parameter λ is one that corresponds to a regularised solution near the "corner", i.e. the point of maximum curvature of the L-curve, [8].

For the problem of boundary detection, the MFS is particularly suitable since the geometry of the system changes for every possible solution tested during the optimisation process. Further, MFS does not require any meshing of the domain or the boundary in order to calculate the boundary data. This reduces the computational effort and eliminates the important perturbations due to changes in the mesh. MFS also is known to be very efficient from the point of view of the computational time required to solve a linear direct problem such as (5)-(8). The MFS can be also used to compute values very near to, or even on the boundaries, and there is no singularity in the solution, as the nodal points are always placed outside the solution domain. The MFS is also suitable for high-dimensional problems and irregular domains.

3.4 Optimisation algorithms

The boundary identification problem (1)-(4) has been solved by using two different algorithms, one deterministic, gradient based optimisation algorithm and one heuristic, evolutionary search algorithm. As a gradient based optimisation algorithm we minimize directly the objective function (9) using a the sequential quadratic programming optimisation algorithm E04UCF from the NAG Libraries [16].

For the evolutionary search we consider a float number encoded genetic algorithm similar to the one proposed in [15] with population size $n_{pop} = 10$, number of offspring $n_{child} = 15$, uniform arithmetic crossover, crossover probability $p_c = 0.65$, tournament selection, tournament size $k = 2$, tournament probability $p_t = 0.8$, non-uniform mutation, mutation probability $p_m = 0.5$, elitism, elitism parameter $n_e = 2$. The float number encoding of the variables enables natural implementation of fine local tuning processes. Indeed the non-uniform mutation operator, [15], is an operator that enables fine local tuning and is defined as follows: if $x =< v_1, v_2, v_3, ..., v_j, ..., v_n >$ is a chromosome, and the element v_j was selected for mutation (domain of v_j is $[\ l_j,\ u_j]$), the result is a vector $< v_1, v_2, v_3, ..., v'_j, ..., v_n >$ where v'_j is given by

$$v'_j = \begin{cases} v_j + \delta(i, u_j - v_j) & \text{if a random digit is 0} \\ v_j - \delta(i, v_j - l_j) & \text{if a random digit is 1} \end{cases} \tag{20}$$

where i is the number of generations performed and the function $\delta(i, y)$ returns a value in the range $[0, y]$ such that the probability of being close to 0 increases as i increases. This property causes this operator to search the space uniformly initially, when i is small, and locally at later stages. We have used the following function

$$\delta(i, y) = \delta(i, I, y, r) = y \cdot (1 - r^{\left(1 - \frac{i}{I}\right)^b}) \tag{21}$$

where r is a random number in the interval $[0, 1]$, I is the maximal generation number and b is a system parameter determining the degree of non-uniformity.

The minimum of the functional J was sought as the maximum of the objective function

$$\overline{f}(\gamma) = \frac{1}{f_{max}^{-1} + J(\gamma)} \tag{22}$$

We note that the maximum of the function $\overline{f}$ is f_{max} which indicates a perfect fit to the data. This constant was included in the form of the fitness function in order to avoid numerical overflow. A value of $f_{max} = 10^2$ was used for this constant in all the results presented in this paper, but various values have been used and similar results were obtained.

4. NUMERICAL RESULTS

In order to test the convergence and the stability of the method proposed, we use as the solution domain the thin plate illustrated in Figure 1. The lengths of the domain are taken to be $L_x = L_y = 1.0$ and $L_z = 0.1$ althought it was found that stable computations can be performed up to $L_z = 0.5$, [14]. The unknown boundary γ is assumed to be maintained at constant voltage, say $u = c$, while a Dirichelet boundary condition is prescribed on Σ and Cauchy boundary conditions are prescribed on Γ. The method

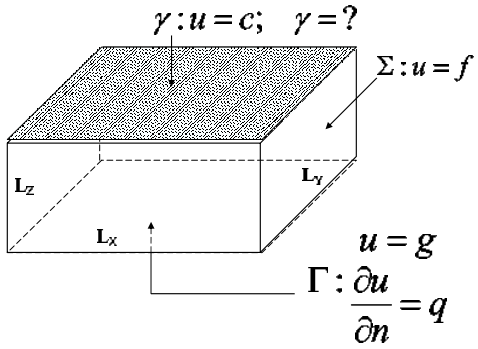


Figure 1: Problem formulation: the domain and the boundary conditions.

of fundamental solution, see section 3.3, is used to solve the direct problem (5)-(8) and to evaluate the objective function (9) for a given boundary γ. In all the results presented in this paper, $N = 150$ collocation points have been uniformly distributed on the boundary $\partial\Omega$ and $M = 150$ nodal point have been distributed on the boundary of a cube of side length $L = 3.0$ that includes in the interior the solution domain Ω. In this section we consider the retrieval of the boundary γ given by various shapes parameterized by B-splines as described in section 3.2. For the numerical test examples considered in this paper we use cubic B-splines, i.e. $k = l = 3$ and we set $m = n = 4$, i.e. the surface γ is given by $(m+1)(n+1) = 25$ control points. The knot vectors are taken to be

$$U = V = (0, 0, 0, 0, \frac{1}{2}, 1, 1, 1, 1) \tag{23}$$

The x and y coordinates of the control points are given by

$$x_{P_{i,j}} = -\frac{L_x}{2} + \frac{i}{n} L_x, \quad i = \overline{0, n}, j = \overline{0, m} \tag{24}$$

$$y_{P_{i,j}} = -\frac{L_y}{2} + \frac{j}{m} L_y, \quad i = \overline{0, n}, j = \overline{0, m} \tag{25}$$

The z coordinates of the control points $\mathbf{P}_{ij}$ with $i \in \{0, n\}$ or $j \in \{0, m\}$ are fixed to $z = 0.5L_z$, since $\gamma \bigcap \Sigma$ is assumed to be known, and the z components of the other control points, i.e. the components of the matrix $\mathbb{Z}$ given by eqn.(13) are to be determined. The boundary data g was perturbed with $s = 1\%$ noise in order to simulate the inherent measurment errors.

The first test example considered is a unimodal, symmetrical test example given by eqn.(10) with

$$\mathbb{Z} = \begin{bmatrix} 2.5L_z & 2.5L_z & 2.5L_z \\ 2.5L_z & 2.5L_z & 2.5L_z \\ 2.5L_z & 2.5L_z & 2.5L_z \end{bmatrix} \tag{26}$$

and the search ranges for the z coordinates of the control points were set to be $z_{min} = 0.1L_z$ and $z_{max} = 3.0L_z$. Throughout this study, the initial guess for SQP optimisation algorithm is provided by assuming all the control points are in the horizontal plane given by

$$z = z_{min} + p(z_{max} - z_{min}) \tag{27}$$

and various values are specified for $p \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$. For the genetic algorithm described in section 3.4 we consider three formulations, different with respect to the methodology used to initialize the GA, i.e. to construct the initial population as follows:

GA1 - the coordinates of the control points are constructed by sampling from a uniform random distribution over the interval $[z_{min}, z_{max}]$, i.e. all the coordinates $z_{P_{i,j}}$ are randomly initialized

GA2 - all the inidividuals in the initial population are constructed by assuming the control points to be in a horizontal plane, The y coordinate of the horizontal plane is randomly sampled from $[z_{min}, z_{max}]$ for every individual.

GA3 - the initial populations contain in equal proportions the five initial guesses specified for the SQP algorithm, given by eqn.(27) for $p = \{0.0, 0.25, 0.5, 0.75, 1.0\}$.

For every result presented in this paper for the GA, the algorithm was run 5 times for various sequences of random numbers and the result presented is the average of the five runs.

In general, when using iterative methods, for solving ill-posed problems then a stopping criterion is required in order to regularize the problem. The iterative processes are not convergent with respect to the number of iterations due to the accumulation of noise effects. The real errors obtained by comparing the numerical solution with the exact solution decrease up to a specific point and then start increasing slowly, as the numerical solution loses its smoothness. Therefore regularizing stopping criteria are used in order to locate the optimum point in iterative methods for ill-posed problems. However, since we are using the function specification method, i.e. the solution is parametrized by B-splines, see section 3.2, the problem is regularized by looking for a solution in a parametric form depending on a reduced number of parameters. Therefore when only the parameters in eqn.(26) are to be retrieved, the iterative process is convergent with respect to the number of generations and simpler stopping criteria can be used, for example, one may stop the iterative process using a Cauchy type criterion, i.e. stop the iterative process when the solution does not change over a large number of iterations/generations.

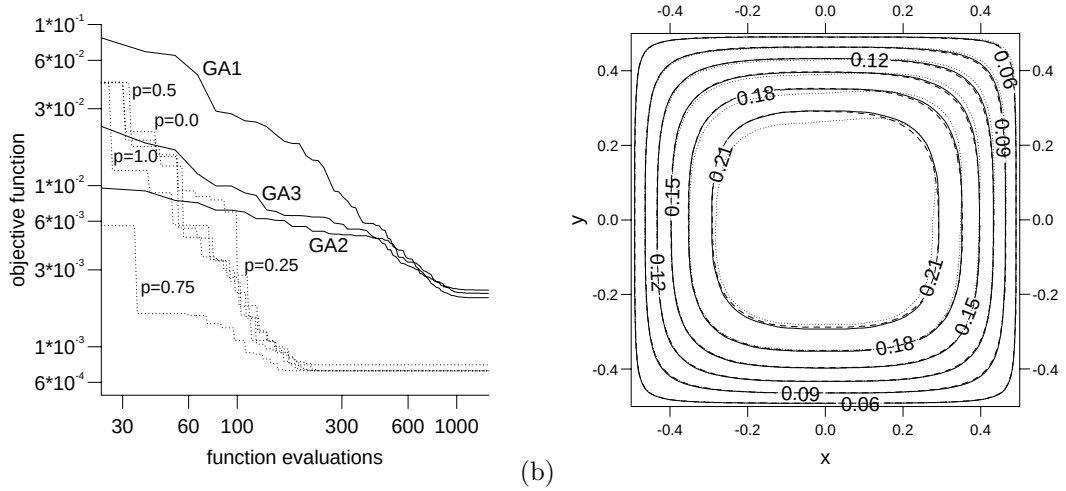
Figure 2(a) presents the objective function evolution for the SQP optimisation process starting with various initial guesses given by eqn.(27) in comparison with the corresponding objective functions obtained in various formulations of the GA. It can be seen that for this simple, uni-modal test example the SQP algoritm is convergent for every initial guess specified and it clearly outperforms the genetic algorithms. A more accurate solution is obtained by the SQP algorithm using only a fraction of the number of fitness function evaluations required in the GA. However the numerical solution generated by the GA is also a good approximation to the real solution of the problem, see Figure 2(b) which presents the isolines of the the numerical solution obtained by the GA and the SQP methods for the boundary γ in comparison with the exact solution. It can be seen that both methods produce a numerical solution that is a very good approximation ot the exact solution. However, the SQP methods obtains a more accurate solution in a smaller number of objective function evaluations.

It is worth noting that in the case of test example (26) the control points of the B-spline surface were all in a horizontal plane, and this information was exploited in the initial guesses specified for the SQP algorithm and the GA2 and GA3 formulations. This explains why during the first stages of the GA, the GA2 and GA3 outperform GA1. However, once convergence has been achieved, then the three GA formulations are relatively equivalent.

A non-symmetrical test example may be obtained by a B-spline surface given by eqn.(10) and the control points

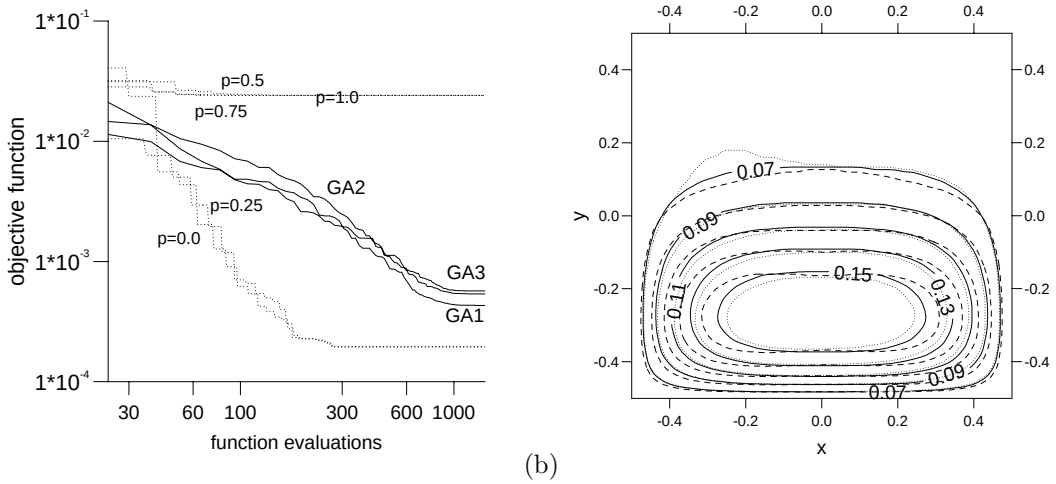
$$\mathbb{Z} = \begin{bmatrix} 2.5L_z & 0.5L_z & 0.5L_z \\ 2.5L_z & 0.5L_z & 0.5L_z \\ 2.5L_z & 0.5L_z & 0.5L_z \end{bmatrix} \tag{28}$$

Figure 3(a) presents the objective function evolution for the SQP optimisation process starting with various initial guesses given by eqn.(27) for the boundary γ given by eqn.(28) in comparison with the corresponding objective functions obtained in various formulations of the GA. In the case of test example (26) the control points of the B-spline surface were all in a horizontal plane, and this information was exploited in the initial guesses specified for the SQP algorithm. Therefore, the SQP algorithm was convergent to the real solution for all initial guesses specifed. However, for the more complex test example (28) the algorithm stagnates in local optima for some initial guesses. Table 1 presents the results obtained for the test example (28) by using the SQP algorithm for various initial guesses and the results obtained by the three GA formulations. The table presents the objective function (9), the corresponding GA



(a) (b)

Figure 2: (a) The evolution of the objective function (9) as obtained by the SQP method ($\cdot\cdot\cdot\cdot\cdot$) with various initial guesses and the various formulations of the GA (——) and (b) the numerical solutions obtained for the isolines of the boundary γ by using the SQP algorithm with $p = 0.5$ (- - -) and the GA1($\cdot\cdot\cdot\cdot\cdot$) formulation of the genetic algorithm in comparison with the exact solution (——) for the test example (26).



(a) (b)

Figure 3: (a) The evolution of the objective function (9) as obtained by the SQP method ($\cdot\cdot\cdot\cdot\cdot$) with various initial guesses and the various formulations of the GA (——) and (b) the numerical solutions obtained for the isolines of the boundary γ by using the SQP algorithm with $p = 0.25$ (- - -) and the GA1($\cdot\cdot\cdot\cdot\cdot$) formulation of the genetic algorithm in comparison with the exact solution (——) for the test example (28).

fitness function (22) and the percentage error calculated as

$$e = 100.0 \frac{\|\gamma - \gamma^*\|_{L^2}}{\|\gamma^*\|_{L^2}} \tag{29}$$

where γ^* is the exact solution for the boundary γ. It can be seen that if the initial guess is far from the real solution the algorithm the SQP algorithm stagnates in local optima while all the GA formulations converge towards the real solutions. Therefore, when applying the SQP numerical procedure in practice, the algorithm should be restarted with several initial guesses in order to obtain the global optimum while the GA was found to always find the exact solution in the first run. It should be noted that during the first GA generations the mutation amplitude given by the eqn.(21) is large and the GA performs a random search distributing the population over the whole search space. By the time the local tunning starts to act, all the GA formulations have already obtaind a level of accuracy equivalent to a good initial

p	$\overline{f}$	J	$\%error$
$p = 0.0$	1.970000E-04	98.071533	4.569
$p = 0.25$	1.950000E-04	98.090234	4.248
$p = 0.5$	2.415400E-02	29.278754	88.962
$p = 0.75$	2.415400E-02	29.278754	88.959
$p = 1.0$	2.415400E-02	29.278755	88.960
GA1	4.344754E-04	95.858485	4.565
GA2	5.433790E-04	94.891760	4.840
GA3	5.733772E-04	94.612097	4.571
exact solution	3.670000E-03	96.461075	0.000

Table 1: The objective function, the corresponding fitness function and the percentage error obtained for the test example (28) by the SQP for various initial guesses given by eqn.(27) and by the genetic algorithms considered.

guess. That is why in the later stages of the evolution all the GA formulations are equivalent and the initial guess does not have a significant impact on the final result. It should be noted that even if the whole population is initialized to only one individual, this GA formulation still behaves similar to GA1, because of the high mutation rates and amplitudes in the first generations.

It can be seen in Table 1 that if the initial guess for the SQP algorithm is not close to the exact solution, then the final result is also far from being accurate, percentage errors of up to 100% are obtained. However, the numerical solution obtained by a SQP algorithm with an accurate initial guess and the GA solutions are good approximations for the exact solution, also for the test example (28) as it can be seen in Figure 3(b).

A more complex boundary γ is considered by utilising the following control points

$$\mathbb{Z} = \begin{bmatrix} 2.5L_z & 0.0 & 2.5L_z \\ 2.5L_z & 0.0 & 2.5L_z \\ 2.5L_z & 0.0 & 2.5L_z \end{bmatrix} \tag{30}$$

Figure 4(a) presents the evolution of the objective function in various SQP and GA formulations for the test example (30). Again it can be seen that the SQP algorithm fails to converge to the global optimum in the case of three out of five initial guesses while the GA converges in all three formulations. However, if the initial guess is a good approximation to the exact solution, the rate of convergence of the SQP is superior to that of the GA.

Finally, we consider one more test example with the control points situated in three different horizontal planes, namely we consider the control points

$$\mathbb{Z} = \begin{bmatrix} 1.4L_z & 0.5L_z & -0.4L_z \\ 1.4L_z & 0.5L_z & -0.4L_z \\ 1.4L_z & 0.5L_z & -0.4L_z \end{bmatrix} \tag{31}$$

and in this case the search domain is considered to be given by $z_{min} = -0.5L_Z$ and $z_{max} = 2.0L_Z$. Figure 4(b) presents the evolution of the objective function in various SQP and GA formulations. Again it can be seen that the SQP has a higher rate of convergence but a lower probability of finding the global optimum.

5. CONCLUSIONS

In this study the problem of boundary detection has been reduced to an optimisation problem, namely to the minimisation of the objective functional (9). Once the problem has been reformulated as a minimisation problem, two optimisation techniques have been employed in order to find the optimal solution. SQP and GA methods have been compared on various test examples. Overall, from the four test examples considered, we may conclude that accurate results are obtained for various shapes of the boundary γ both by the SQP and the GA methods, provided care is taken when specifying the initial guesses for the SQP algorithm.

One major disadvantage of gradient methods is the convergence towards local optima. For example, one cannot expect the objective functional (9) to be convex and twice continuously differentiable, which would ensure the convergence of gradinet based method towards the absolute minimum. Since the functional may have a complex structure, with multiple ridges and valleys, the gradient methods may fail

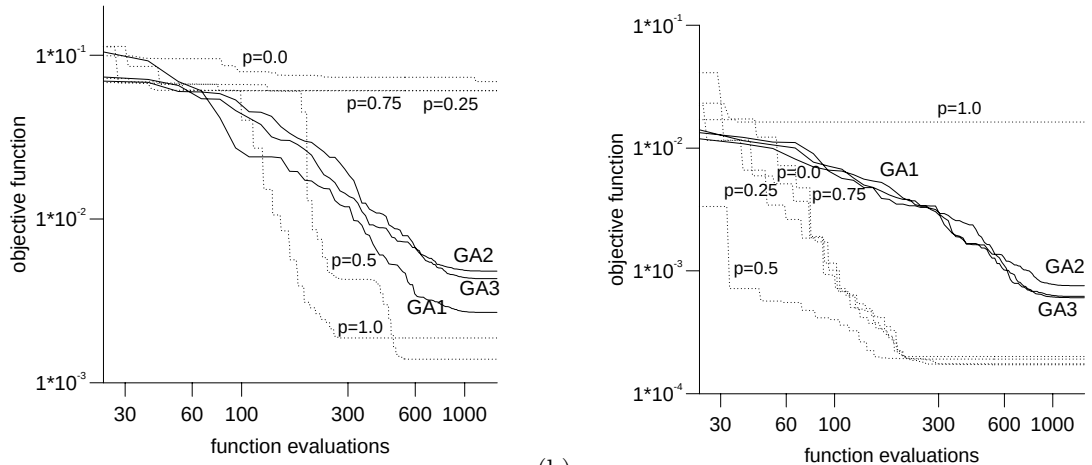


Figure 4: The evolution of the objective function (9) as obtained by the SQP method (·····) with various initial guesses and the various formulations of the GA (——) for (a) the test example (30) and (b) the test example (31).

to locate the global optimum. It should be noted that similar results have been obtained for the two-dimensional case. It was found in [6] that the numerical solutions obtained by the Gradient Projection Method (GPM) or Nelder-Mead Simplex Method (NMNS) depend on the initial guess specified. For some initial guesses, these gradient based methods converge toward local optima which are far from the real solution for the boundary γ.

On the other hand, the genetic algorithm, starting with a random initial population, was always found to converge to the global optimum. In general, it in known that genetic algorithms have the ability to escape local optima and therefore they are particularly suitable for nonlinear functions with complex, highly structured landscape which can appear in shape optimisation problems.

One disadvantage of genetic algorithms, in comparison with gradient methods, is the increased amount of computational time required for the evolutionary search process. However, for the examples considered in this paper it was found that, in general, it was possible to obtain accurate results in less than 100 generations. On a Pentium IV processor at 1700Mz, the CPU time required for performing 100 generations was on average 67 minutes. Since we have solved a three-dimensional problem we may conclude that on using genetic algorithms we can obtain accurate results in a reasonable computational time. It is worth noting that if gradient methods are used then the only way to avoid local convergence is to restart the program from multiple initial guesses and allow the user to decide which solution to use. This process can take longer than the time required for one run of the GA which will locate the global optimum.

Overall, it can be concluded that when solving a complex practical problem, global optimisation algorithms such as an evolutionary algorithms should be used in order to obtained an efficient initial guess, and then the solution may be further refined using gradient based algorithms. Future work will be directed towards developing a hybrid SQP-GA method that combines the advantages of the two methods presented in this study by incorporating gradient based search within the evolutionary scheme of the GA.

REFERENCES

1. G. Alessandrini, Examples of instability in inverse boundary value problems. *Inverse Problems* (1997) **13**, 887-897.

2. G. Alessandrini and L. Rondi, Stable determination of a crack in a planar inhomogeneous conductor. *SIAM J. Math. Anal.* (1998) **30**, 326-340.

3. D.D. Ang and L.K. Vy, Domain identification for harmonic functions. *Act. App. Math.* (1995) **38**, 217-238.

4. N.D. Aparico and M. Pidcock, The boundary inverse problem for the Laplace equation in two dimensions. *Inverse Problems* (1996) **12**, 565-577.

5. E. Berretta and S. Vessella, Stable determination of boundaries from Cauchy data. *SIAM J. Math. Anal.* (1998) **30**, 220-232.

6. M. Charton, H.J. Reinhardt and D.N. Hao, Numerical solution of a shape optimisation problem, Eurotherm seminar 68, *Inverse Problems and Experimental Design in Thermal and Mechanical Engineering*, Poitier, France, 2001, pp 101-105.

7. G. Fairweather and A. Karageorghis, The method of fundamental solutions for elliptic boundary value problems. *Adv. Comput. Math.* (1998) **9**, 69-95.

8. P.C. Hansen, Analysis of discrete ill-posed problems by means of the *L*-curve. *SIAM Review* (1992) **34**, 561-580.

9. P.C. Hansen, *Rank Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, 1998.

10. V. Isakov, New stability results for the soft obstacles in inverse scattering. *Inverse Problems* (1993) **9**, 535-543.

11. P.G. Kaup and F. Santosa , Nondestructive evaluation of corrosion damage using electrostatic boundary measurements. *J. Nondestruct. Eval.* **14**, (1995) 127-136.

12. P.G. Kaup, F. Santosa and M. Vogelius, Method for imaging corrosion damage in thin plates from electrostatic data. *Inverse Problems* (1996), **12** 279-293.

13. N.S. Mera, L. Elliott and D.B. Ingham, Boundary identification for a 3D Laplace equation using a genetic algorithm, *Comput. Meth. App. Mech. Eng.* (2005) in print.

14. N.S. Mera and D. Lesnic, A three-dimensional boundary determination problem in potential corrosion damage, *Comput. Mech.* (2005) in print.

15. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin, 1996.

16. V.A. Morozov, On the solution of functional equations by the method of regularization. *Soviet Math. Dokl.* (1966) **7**, 414-417.

17. NAG Fortran Library Manual, (1999), Mark 18, The Numerical Algorithms Ltd., Oxford.

18. L. Piegl and W. Tiller, *The NURBS book*, Springer, Berlin, 1997.

19. O. Pironneau, *Optimal Shape Design for Elliptic Systems*, Springer-Verlag, New York, 1984.

20. A.N. Tikhonov, A.V. Goncharky, V.V. Stepanov and A.G. Yagola, *Numerical Methods for the Solution of Ill-Posed Problems*, Kluwer Acad. Publ., Dordrecht, 1995.