A Genetic Algorithm for Computing Singular Minimizers of the Calculus of Variations

Pablo V. Negrón-Marrero*
Department of Mathematics
University of Puerto Rico
Humacao, PR 00791-4300

Abstract

We apply a genetic algorithm combined with a direct optimization method to problems of the Calculus of Variations in one dimension that exhibit the Lavrentiev phenomena. The genetic algorithm is used as a predictor of the direct method. In this way we take advantage of the localization property of the genetic algorithm together with the fast local convergence of the direct method. The numerical results show that this combination produces very effective and robust methods for computing singular minimizers.

Key words: Genetic algorithm, calculus of variations, Lavrentiev phenomena.

1 Introduction

The basic problem of the Calculus of Variations in one dimension can be stated as finding a function $u : [a, b] \to \Re$ belonging to a certain class A of functions satisfying certain boundary conditions such that the integral

$$I(u) = \int_a^b f(x, u(x), u'(x))\, dx \tag{1}$$

*pnm@cuhwww.upr.clu.edu

is minimized. The problem of minimizing (1) is called *regular* if the integrand $f(x,u,z)$ is C^2 and $f_{zz} > 0$ for each (x,u). Minimizers of (1) can be *singular* in the sense that they may be discontinuous or have unbounded derivatives at some points. For these problems one could further have:

1. The Lavrentiev phenomena ([11], [6])

$$\inf_{A \cap W^{1,\infty}(a,b)} I(u) > \inf_{A} I(u)$$

2. If (u_j) is a sequence in $W^{1,\infty}(a,b)$ that converges almost everywhere to the minimizer u^* of (1), then $I(u_j) \to \infty$ as $j \to \infty$.

Here and henceforth $L^p(a,b)$ denotes the space of functions whose p-th power is integrable on (a,b) in the sense of Lebesgue, and $W^{1,p}(a,b)$ denotes the Sobolev space of functions u that have a distributional derivative u', both of which belong to $L^p(a,b)$. (See [1]). Examples of singular minimizers are given in [4], [5], [8]. The above two conditions suggest that usual finite element methods to solve (1) may fail both in approximating u^* and $I(u^*)$ in the general case in which the location and type of singularity is unknown. A method based on a decoupling technique was introduced in [3] and [15] that circumvents both of these two problems. In this method the decoupled integral

$$\hat{I}(u,v) = \int_a^b f(x,u(x),v(x))\,dx \tag{2}$$

is introduced where u belongs to $W^{1,1}(a,b)$ plus some boundary conditions and v belongs to $L^1(a,b)$. The problem now is to minimize (2) subject to the condition

$$\int_a^b |u'(x) - v(x)|\,dx \leq \epsilon \tag{3}$$

For an appropriate finite element discretization of (2) and (3) with discretization parameter "h", it is shown in [3] that for a sequence $(h_j, \epsilon_j) \to (0,0)$ both of the problems cited above can be overcome. However this convergence result does not specify how the sequence $\{(h_j,\epsilon_j)\}$ can be chosen. In practice one has to add to these two parameters a penalization parameter corresponding to (3) and a discretization parameter for the integrals in (2) and (3) which makes the method impractical particularly in higher dimensions. A related method that works as well with a modification of the functional (1) is the so-called *element removal method* introduced by [12], [13]. In this method

the regions of large function values or derivatives are omitted in a certain sense from the integrand in (1).

Direct minimization methods for (1) based on quadrature rules can overcome the Lavrentiev phenomenon and energy approximation problems cited above. In [16] a method based on a Richardson extrapolation formula and a generalized mid-point rule is used to compute singular minimizers of certain problems in one and two dimensions. There is no guarantee however that such methods will work for general problems as the discretizations are tailored suited to the problems and convergence depends strongly on the initial guess for the iterations.

In any case, decoupled or direct method, the computations are sensitive to the initial point and can get trapped in local minima of the functional. This is where genetic algorithms can be useful. Genetic algorithms are methods that simulate a process of survival of the fitness for optimizing a given function. Some "fitness" value is assigned to every element in a group or population of possible optima and processes of natural selection and mutation are implemented to generate a new population of hopefully better individuals. The process is repeated and convergence to a solution is guaranteed under certain conditions. (See [17]). The important characteristic of the genetic algorithm is that because of its random nature, the method searches a much larger region of the space than a typical deterministic or direct method. Thus they do not get trapped into local minima very often and consequently can locate global optima fairly well. Also it has been observed that for large optimization problems, the execution time for genetic algorithms grow linearly with the size of the problem compared with the standard minimization methods that grow quadratic in time. In this paper we apply a genetic algorithm combined with a direct method to problems of the form (1) that exhibit the Lavrentiev phenomena. The genetic algorithm is used as a predictor of the direct method. In this way we take advantage of the localization property of the genetic algorithm together with the fast local convergence of the direct method. The numerical results show that this combination produces very effective and robust methods for computing singular minimizers. The bibliography on genetic algorithms is extensive but we refer the reader to [14] and the references there in.

In Section 2 we describe some model problems of the form (1) that exhibit the Lavrentiev phenomena and that we will solve numerically. On Section 3 we describe the discretizations of the model problems and Section 4 is devoted to describing the genetic algorithm and the operators used. Finally

we give in Section 5 some numerical results and conclusions.

2 Model Problems

We present in this section some examples of problems of the form (1) that exhibit the Lavrentiev phenomenon and that we will use to test our numerical schemes. For the proofs of all the results on this section as well as other examples we refer to [2], [4], [5], and [8].

Consider the problem of minimizing the functional

$$I(u) = \int_0^1 (u(x)^3 - x)^2 u'(x)^6 \, dx \tag{4}$$

for functions $u \in W^{1,1}(0,1)$ and satisfying the boundary conditions $u(0) = 0$, $u(1) = 1$. Since the integrand is always non-negative the function $u^*(x) = x^{1/3}$ is the minimizer which has an unbounded derivative at $x = 0$. Furthermore it can be shown that

$$I(u^*) < \inf_{W^{1,r}(0,1)} I(u) \tag{5}$$

for any $r > 3/2$.

For our second example, we consider the radially symmetric deformations of a unit disk in the plane composed of an isotropic, homogenouos, and hyperelastic material. The disk is subjected to a uniform displacement of λ units of its outer boundary. If we let $\rho(s)$ denote the radius in the deformed configuration of the circle with radius s in the undeformed configuration, then the equilibrium configuration of the disk is given as the minimum of the following functional:

$$I(\rho) = \int_0^1 s\Phi\left(\rho'(s), \frac{\rho(s)}{s}\right) ds \tag{6}$$

among functions ρ in

$$A_\lambda = \{\ \rho \in W^{1,1}(0,1) \mid \rho(0) \geq 0\ ,\ \rho(1) = \lambda\ ,\ \rho'(s) > 0 \text{ a.e. } \} \tag{7}$$

Here $\Phi(v_1, v_2)$ represents the stored energy function of an *isotropic, homogeneouos and hyperelastic material* and is given by

$$\Phi(v_1, v_2) = A(v_1^\alpha + v_2^\alpha) + C(v_1 v_2)^\gamma + D(v_1 v_2)^{-\delta} \tag{8}$$

and all the coefficients and variables in the exponents are nonnegative except for α which is usually taken greater or equal to one. The different terms in (8) satisfy the requirement that infinite expansions or compressions of fibers, or surface elements within the material, must be accompanied by an infinite energy. The conditions $\rho(0) \geq 0$ and $\rho'(s) > 0$ in the set A_λ correspond to the requirement that an element of positive area can not be reduced to zero during the deformation. In [2] it is shown that (6) has a minimum $\rho \in A_\lambda$ for any $\lambda > 0$. Furthermore, for $\alpha < 2$ and λ sufficiently large, the minimizer satisfies $\rho(0) > 0$ and (Lavrentiev phenomena)

$$\inf_{A_\lambda} I(\rho) < \inf_{A_\lambda \cap W_s^{1,r}(0,1)} I(\rho) \tag{9}$$

for any $r \geq 2$ where

$$W_s^{1,r}(0,1) = \left\{ \rho \in W^{1,1}(0,1) \mid \int_0^1 s \left[\rho'(s)^r + \left(\frac{\rho(s)}{s} \right)^r \right] ds < \infty \right\} \tag{10}$$

The condition $\rho(0) > 0$ is called *cavitation* and implies that a hole opens at the center of the disk.

3 Discretizations

For the model problem (4) let

$$x_i = ih \quad , \quad 0 \leq i \leq n \quad , \quad h = 1/n \tag{11}$$

and u_i denotes an approximation of $u(x_i)$. We write

$$u_h = (u_0, u_1, \ldots, u_n) \quad , \quad u_0 = 0 \ , \ u_n = 1 \tag{12}$$

Then we discretize (4) as follows:

$$I_h(u_h) = h \sum_{i=0}^{n-1} \frac{1}{2} \left[(u_i^3 - x_i)^2 + (u_{i+1}^3 - x_{i+1})^2 \right] (\delta u_i)^6 \tag{13}$$

where

$$\delta u_i = \frac{u_{i+1} - u_i}{h} \quad , \quad 0 \leq i \leq n-1 \tag{14}$$

Note that I_h is a function of $n-1$ variables that we minimize numerically.

For the model problem (6) we let

$$s_i = ih \quad , \quad 0 \leq i \leq n \quad , \quad h = 1/n \tag{15}$$

and ρ_i denotes an approximation of $\rho(s_i)$. We use the notations

$$s_{i+1/2} = \frac{s_i + s_{i+1}}{2} \quad , \quad \Phi^{i+1/2} = \Phi\left(\delta\rho_i, \frac{\rho_i + \rho_{i+1}}{s_i + s_{i+1}}\right) \tag{16}$$

where $0 \leq i \leq n-1$. With

$$\rho_h = (\rho_0, \rho_1, \ldots, \rho_n) \quad , \quad \rho_n = \lambda \tag{17}$$

we discretize (6) by

$$I_h(\rho_h) = h \sum_{i=0}^{n-1} s_{i+1/2}\Phi^{i+1/2} \tag{18}$$

which now is a function of n variables that we minimize numerically.

4 A Genetic Algorithm

In this section we describe some aspects of the genetic algorithm. For further details on genetic algorithms and related topics see [7], [9], [10], and [14].

Genetic algorithms are methods that simulate a process of survival of the fitness or natural selection for optimizing a given function. A *fitness function* is used to assign a "fitness" value to every element in a group or *population* of possible optima. The processes of natural selection and mutation are implemented using certain operators so as to generate a new population of hopefully better individuals. The process is repeated and convergence to a solution is guaranteed under certain conditions. The important characteristic of the genetic algorithm is that because of its random nature, the method searches a much larger region of the space than a typical deterministic or direct method. Thus they do not get trapped into local minima very often and consequently can locate global optima fairly well. The basic steps of a genetic algorithm can be described as follows:

1. (Initialization) Generate a usually random initial population of N individuals and assign to each a fitness value.

2. (Selection) From the given population select those individuals that are most fitted and that will be represented in the next generation.

3. (Reproduction) A portion of the population reproduces to generate offsprings of the most fitted individuals.

4. (Mutation) A portion of the population undergoes a process of mutation to generate potentially new individuals.

5. (Evaluate) Assign fitness values to the new population and repeat from step (2).

We now describe in some details each of these basic steps but first we mention that we use a floating point implementation of the genetic algorithm in which each individual is represented by a vector of $n-1$ floating point numbers where n is specified in (11) or (14). The use of a floating-point alphabet makes for a more natural representation of the individuals in our population and reduces the size of the search space. We mention however that most of the theoretical analysis for genetic algorithms is for the binary alphabet.

4.1 Initialization

For the initial population we used a randomly generated $N \times (n-1)$ matrix where N represents the population size and $n-1$ the number of variables in (13) or (16). The rows of this matrix are called *individuals* or *chromosomes* and $n-1$ is the *chromosome length.*

4.2 Selection

In the selection process individuals are selected for the next generation according to their fitness. A spinning roulette mechanism is simulated in the following manner. Let fit_i, $1 \leq i \leq N$ be the fitness values for all the population individuals and let $\text{fit}_{tot} = \sum_{j=1}^{N} \text{fit}_j$ be the *total fitness.* Then for a randomly generated number r, the i-th individual is passed to the next generation if

$$\sum_{j=1}^{i-1} \text{fit}_j < r\, \text{fit}_{tot} \leq \sum_{j=1}^{i} \text{fit}_j \tag{19}$$

This process is repeated N times. Note that several copies of a given individual can "survive" to the next generation this of course been more probable for the most fitted individuals.

4.3 Reproduction

After the selection process is completed, a fraction of the selected individuals are chosen for *mating or crossover*. The process is called *arithmetic crossover* and can be described as follows. For a given number p_m between zero and one, the *mating rate*, a fraction p_m of the population is randomly selected for mating. Given two individuals u, v for mating, a new pair $\hat{u}, \hat{v}$ is generated as follows:

$$\hat{u} = r\,u + (1-r)v \quad , \quad \hat{v} = (1-r)u + r\,v \tag{20}$$

where r is a randomly generated number between zero and one.

4.4 Mutation

The mechanism of mutation is used in the genetic algorithm to introduce variability. The selection process described in Subsection (4.2) above has the tendency of stabilizing the population. That is, after several iterations, the selection process by itself tends to produce similar individuals, which may or may not represent global minima. The mutation process then is used to break this early convergence of the method by randomly inserting new individuals into the population. In particular we used what is called *non-uniform mutation*. Given a number μ between zero and one, the *mutation rate*, a fraction μ of the total of $N(n-1)$ "bits" in the population is selected for mutation. If the variable $x \in [a, b]$ is chosen for mutation, then a new value $\hat{x}$ is generated according to:

$$\hat{x} = \begin{cases} x + (b-x)f_g & , \quad r_1 < 0.5 \\ x + (x-a)f_g & , \quad r_1 \geq 0.5 \end{cases} \tag{21}$$

where r_1, r_2 are randomly generated numbers between zero and one, and

$$f_g = 1 - r_2^{\text{ifract}} \quad , \quad \text{ifract} = \left(1 - \frac{\text{iter}_c}{\text{iter}_{max}}\right)^2 \tag{22}$$

and $\text{iter}_c, \text{iter}_{max}$ are the current iteration index and maximum number of iterations respectively. Since f_g is small as iter_c increases, the mutation mechanism disminishes as the iteration counter increases.

The fitness function for our discretized model problems (13) and (16) is given by

$$\hat{f}(u_h) = \frac{1}{1 + I_h(u_h)} \tag{23}$$

so that for $\hat{f}$ we have now a maximization problem. Finally we mention that in the actual implementation we used also an *elitist strategy* in which the best individual of the previous generation is kept. If the best individual of the current generation were worse, then the best member of the previous generation would replace the worst member of the current population.

5 Numerical Results and Conclusions

We describe now the results of applying the genetic algorithm in combination with a direct method to the model problems of Section 2. The direct optimization method we use is based on a Richardson extrapolation and is described in [16]. The actual implementation of the genetic algorithm includes after the initialization step a stage in which a simulated annealing method is applied to the randomly generated initial population thus producing a better initial population for the genetic algorithm. We refer to [14] for details on the simulated annealing method.

The results we present are the best of ten independent runs of the program. The population size was set at 40 with a maximum number of 2000 iterations. The mating rate was set at 0.25 and the mutation rate at 0.01. The chromosome length is 19 or 20, which corresponds to $n = 20$ in (11) or (15) respectively. All computations were carried on a Gateway PC with a 266Mhz Pentium II processor.

For the model problem (4) and its discretization (13) the results are shown in Table 1. The minimum value in this table corresponds to the value of $I_h(u_h)$ in (13). Note that the code successfully computes the global minimum of zero.

	Genetic Algorithm	Direct Method
Execution Time (min)	14.5387	2.5475
Minimum Value	0.095266	1.6284e-016

Table 1: Numerical results for the model problem (4)

The results of Figure 1 show the average fitness (function values here are for (23)) as a function of the iteration for the genetic algorithm. The actual population fitnesses behave in a similar fashion but with a high variability during the first iterations and a small one for the later ones.

In Figure 2 we show the solutions computed by the genetic algorithm, the direct method and the exact solution which is known for this problem. We can see that the direct method computes the exact solution essentially to the machine precision. It is important to observe that the genetic algorithm approximates fairly well the minimizer in the region where the derivative is very large thus capturing the singular behavior of the solution.

The Lavrentiev phenomenon for the discretized problem becomes the case of a function with one local minima with a value strictly greater than the global minimum. On one of the ten runs of our code, the combined method converged to a local optima of (13) which represents an approximation of a smooth solution of (4) with a minimum energy of approximately 0.03489. In Figure 3 we show this solution as the Final Approximate Solution.

For the model problem (6) and its discretization (18) the results are shown in Table 2. The qualitative performance of the genetic algorithm is similar to that for the first test problem as can be seen from Figure 4. The condition $\rho'(s) > 0$ in (7) was not included explicitly in the genetic algorithm thus reducing the overhead time but resulting perhaps in a not so optimal solution. In fact in Figure 5 we can see that the solution computed by the GA in this case is not stricly increasing. The condition of strictly increasing however was included in the direct method as the overhead required was minimal. In Figure 5 we show both the solution computed by the GA and the direct method together with the graph of the deformation corresponding to a uniform expansion of the disk.

	Genetic Algorithm	Direct Method
Execution Time (min)	9.828	2.8698
Minimum Value	2.7502	2.4662

Table 2: Numerical results for the model problem (6)

References

[1] Adams, R. A., Sobolev Spaces, Ed., Academic Press, New York, 1975.

[2] Ball, J. M., Discontinuous Equilibrium Solutions and Cavitation in Nonlinear Elasticity, Phil. Trans. Royal Soc. London **A 306**, 557-611, 1982.

[3] Ball, J. M. and Knowles, G., A Numerical Method for Detecting Singular Minimizers, Numer. Math. 51, 181-197, 1987.

[4] Ball, J. M. and Mizel, V. J., Singular Minimizers for Regular one Dimensional Problems in the Calculus of Variations, Bull. Am. Math. Soc., New Ser. 11, 143-146, 1984.

[5] Ball, J. M. and Mizel, V. J., One Dimensional Variational Problems whose Minimizers do not satisfy the Euler-Lagrange Equation, Arch. Rat. Mech. Anal. 90, 325-388, 1985.

[6] Cesari, L., Optimization - Theory and Applications, (Ed.), Springer-Verlag, New York, 1983.

[7] Davis, L., The Handbook of Genetic Algorithms, Van Nostrand Reingold, New York, 1991.

[8] Davie, A. M., Singular Minimizers in the Calculus of Variations in one Dimension, Arch. Rat. Mech. Anal. 101, 161-177, 1988.

[9] Goldberg, D., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, 1989.

[10] Holland, J., Adaptation in Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, 1975.

[11] Lavrentiev, M., Sur Quelques Problemes du Calcul des Variations, Ann. Math. Pure Appl. 4, 7-28, 1926.

[12] Li, Z., Element Removal Method for Singular Minimizers in Variational Problems Involving Lavrentiev Phenomenon, Proc. Roy. Soc. Lond. A 439, 131- 137, 1992.

[13] Li, Z., Element Removal Method for Singular Minimizers in Problems of Hyperelasticity, Math. Models and Methods in Applied Sciences, Vol. 5, No. 3, 387-399, 1995.

[14] Michalewicz, Z., Genetic Algorithms + Data Structures = Evolution Programs, Third Edition, Springer-Verlag, 1996.

[15] Negrón-Marrero, P. V., A Numerical Method for Detecting Singular Minimizers of Multidimensional Problems in Nonlinear Elasticity, Numer. Math. 58, 135-144, 1990.

[16] Negrón-Marrero, P. V. and Betancourt, O., The Numerical Computation of Singular Minimizers in Two Dimensional Elasticity, Journal of Computational Physics, Vol. 113, No. 2, August 1994.

[17] Suzuki, J., A Markov Chain Analysis on Simple Genetic Algorithms, IEEE Transactions on Systems, Man and Cybernetics, Vol. 25, No. 4, 1995.

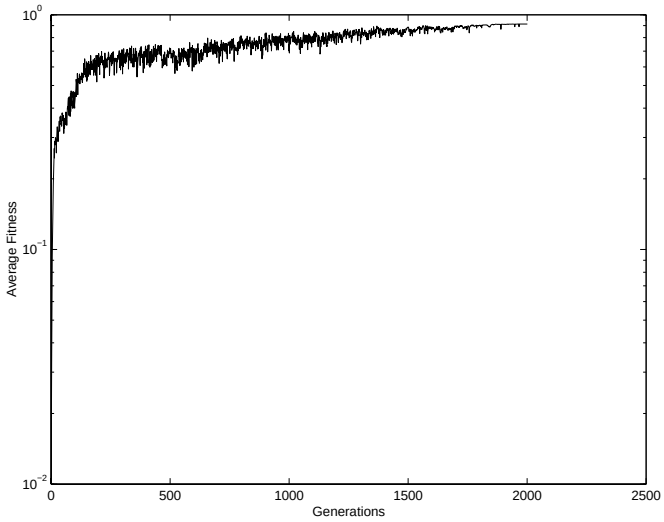

Figure 1: Average fitness as a funtion of the iterations for the GA in problem (4).

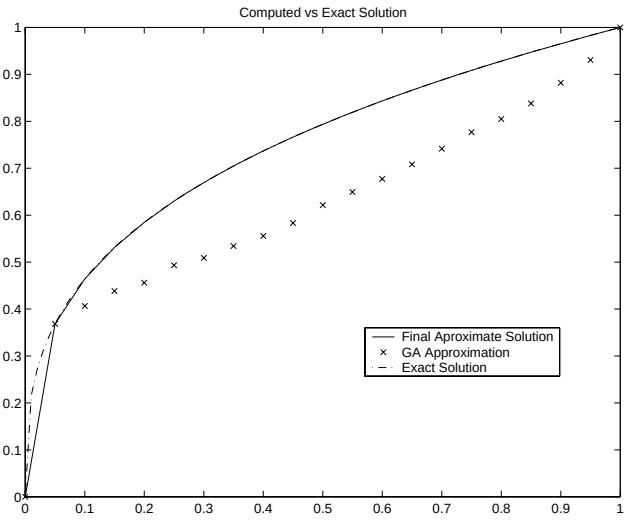

Figure 2: Computed vs exact solution for problem (4).

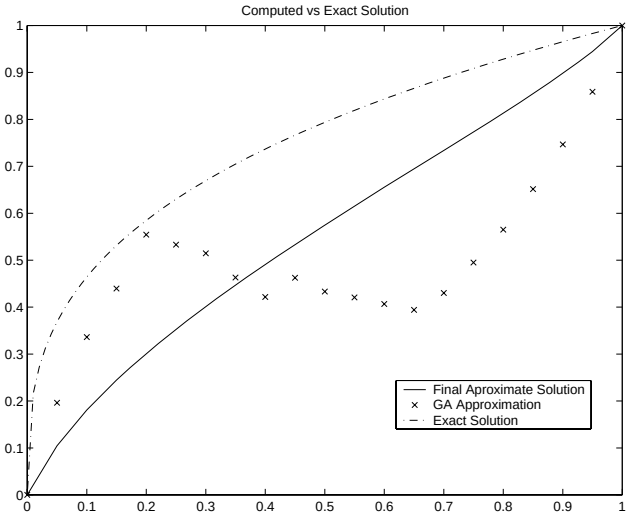

Figure 3: Computed vs exact solution for problem (4).

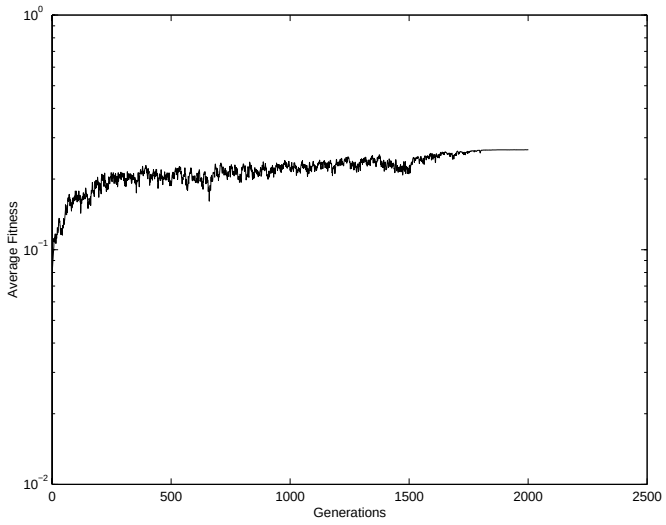

Figure 4: Average fitness as a funtion of the iterations for the GA for problem (6).

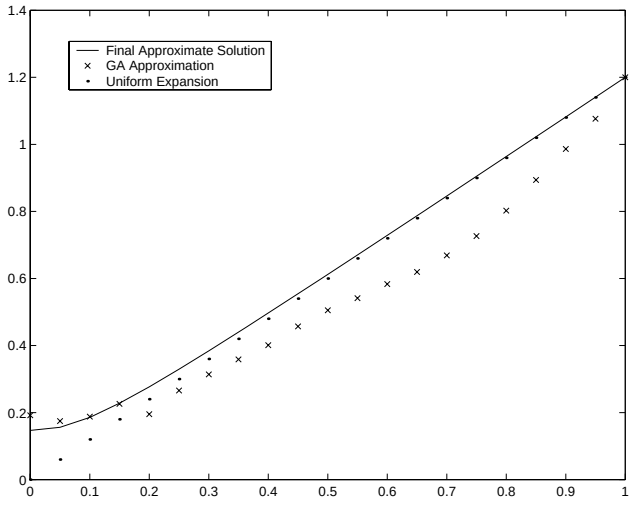

Figure 5: Computed vs exact solution for problem (6).