

FINDING ROOTS OF NONLINEAR SYSTEMS OF EQUATIONS ON A DOMAIN

Alexandru **Horváth**

Department of Mathematics and Computer Science, Petru Maior University, 4300 Tg-Mureş, str. N. Iorga 1, RO

Béla **Finta**

Department of Mathematics and Computer Science, Petru Maior University, 4300 Tg-Mureş, str. N. Iorga 1, RO

Received: December 2003

MSC 2000: 65 H 20; 13 P 10

Keywords: Numerical approximations, Groebner bases.

Abstract: The aim of this paper is to present a method of finding all the roots of a system of nonlinear equations which are located in a given domain. We use here a different approximation method than in the previous paper ([5]).

Let us remember that the usual numerical methods of finding the solutions of a system of nonlinear equations are deficient because of their dependence upon the starting point of the iteration process. We gave in a previous paper ([5]) an effective method of finding all the solutions of a system of nonlinear equations. The only assumptions we supposed were that the system had to have a finite number of roots in the given domain, and the roots had to be regular points of the application defining the system of nonlinear equations.

where

$$\Phi_1(t) = (1 - t)^2(1 + 2t),$$

$$\Phi_2(t) = t^2(3 - 2t),$$

$$\Phi_3(t) = t(1 - t)^2,$$

$$\Phi_4(t) = -t^2(1 - t).$$

It is clear now that each of the functions f_1 and f_2 can be approximated in this way, so we would have instead of the nonlinear system

$$F(x, y) = 0$$

a number of $N \cdot M$ systems of polynomial equations of the form

$$S(f_1)(x_1, x_2) = 0$$

$$S(f_2)(x_1, x_2) = 0,$$

where $x = (x_1, x_2) \in D_{i,j}$ for each sub-domain $D_{i,j}$, $i = 0, 1, \dots, N - 1$ and $j = 0, 1, \dots, M - 1$.

The method we then propose consists of the following steps:

STEP 1. Approximate the differentiable functions f_i as was discussed earlier.

STEP 2. Apply the Groebner bases algorithm to the $N \cdot M$ polynomial systems of equations

$$S_{i,j}(f_1)(x_1, x_2) = 0$$

$$S_{i,j}(f_2)(x_1, x_2) = 0,$$

where for each sub-domain $D_{i,j}$, $i = 0, 1, \dots, N - 1$ and $j = 0, 1, \dots, M - 1$ in order to obtain equivalent "triangular" polynomial systems.

Recall here again some definitions and facts.

Definition. Let $I \subset k[x_1, \dots, x_n]$ be an ideal (not 0).

- (1) the set of leading terms of I we will denote by $LT(I) = \{cx^\alpha : \text{there exists } f \in I, \text{ with } LT(f) = cx^\alpha\}$,
- (2) the ideal of leading terms is the ideal generated by the elements of $LT(I)$, and is denoted by $\langle LT(I) \rangle$.

It is easy to prove that the ideal of leading terms is a monomial ideal and is finitely generated by the leading terms of a finite set of some polynomials of I .

Now we can give the definition of a Groebner base. Let us consider given a monomial order. Let I be an ideal of $k[x_1, \dots, x_n]$.

Definition. A finite subset $G = \{g_1, \dots, g_s\}$ of I is said to be a Groebner basis of I , if ([3])

$$\langle \text{LT}(g_1), \dots, \text{LT}(g_s) \rangle = \langle \text{LT}(I) \rangle.$$

We emphasize the most important property of a Groebner basis: there is an algorithm which compute it. This algorithm starts of course from a finite bases of I , supplied by the Hilbert bases theorem.

For effective computations we used the software package Singular ([4]).

Theorem (see [3], [1]). *If the system has a finite number of solutions, then there exists an ordering such that the Groebner bases of the ideal generated by the polynomials is "triangular".*

STEP 3. Apply a standard numerical method to solve the polynomial equation with one unknown. (In the example which follows we used dedicated software packages as was mentioned earlier.)

STEP 4. Retain the solutions located in D_{ij} for the appropriate system. Generate all the solutions of the "triangular" polynomial system, located in D_{ij} .

STEP 5. Refine the solutions by a standard numerical method using the original system (In the next example we used the Newton-Kantorovich method.)

A detailed analysis of the numerical stability of this method are planned in a future work.

Let us give now a simple example, actually the same example as was given in [5]. We will gain a filing of comparison between the two different approximation method.

Problem. Find all the solutions in $[0, 1] \times [0, 1]$ of the following nonlinear system of equations:

$$\begin{aligned} \cos(x + y) - x - y + \frac{1}{4} &= 0 \\ x^2 + y^2 - x - y + \frac{1}{4} &= 0. \end{aligned}$$

Solution:

STEP 1. Let us choose $N = 3$ and also $M = 3$. Applying the approximation formulas of the Theorem we get a number of $3 \times 3 = 9$ polynomial systems. Here are the first three systems:

For $i = 0$ and $j = 0$ or $D_{0,0} = [0, 0.333] \times [0, 0.333]$ we have the system

$$\begin{aligned}
& 1.564 * 10^{-1}x^3y + 2.757 * 10^{-2}x^3 + 8.394 * 10^{-2}x^2y - \\
& -5.046 * 10^{-1}x^2 + 1.564 * 10^{-1}xy^3 + 8.391 * 10^{-2}xy^2 - \\
& -1.027xy - x + 2.757 * 10^{-2}y^3 - 5.046 * 10^{-1}y^2 - y + 1.250 = 0 \\
& \qquad \qquad \qquad x^2 - x + y^2 - y + 2.500 * 10^{-1} = 0.
\end{aligned}$$

For $i = 0$ and $j = 1$ or $D_{0,1} = [0, 0.333] \times [0.333, 0.666]$ we have the system

$$\begin{aligned}
& 1.301 * 10^{-1}x^3y + 3.643 * 10^{-2}x^3 + 2.414 * 10^{-1}x^2y - \\
& -5.570 * 10^{-1}x^2 + 1.300 * 10^{-1}xy^3 + 1.114 * 10^{-1}xy^2 - \\
& -1.086xy - 9.824 * 10^{-1}x + 7.969 * 10^{-2}y^3 - 5.563 * 10^{-1}y^2 - \\
& \qquad \qquad \qquad -9.829 * 10^{-1}y + 1.248 = 0 \\
& \qquad \qquad \qquad x^2 - x + y^2 - y + 2.500 * 10^{-1} = 0.
\end{aligned}$$

For $i = 0$ and $j = 2$ or $D_{0,2} = [0, 0.333] \times [0.666, 1]$ we have the system

$$\begin{aligned}
& 8.940 * 10^{-2}x^3y + 6.349 * 10^{-2}x^3 + 3.722 * 10^{-1}x^2y - \\
& -6.443 * 10^{-1}x^2 + 8.934 * 10^{-2}xy^3 + 1.934 * 10^{-1}xy^2 - \\
& -1.180xy - 9.439 * 10^{-1}x + 1.230 * 10^{-1}y^3 - 6.422 * 10^{-1}y^2 - \\
& \qquad \qquad \qquad -9.261 * 10^{-1}y + 1.236 = 0 \\
& \qquad \qquad \qquad x^2 - x + y^2 - y + 2.500 * 10^{-1} = 0.
\end{aligned}$$

STEP 2. The Groebner basis method yields to the systems:

$$\begin{aligned}
& y - 9.239 * 10^{-2} = 0 \\
& x + 1.716 * 10^{-2}y^5 - 1.076 * 10^{-1}y^4 + \\
& \quad + 3.785 * 10^{-1}y^3 - 1.969 * 10^{-1}y^2 + \\
& \quad + 9.593 * 10^{-1}y - 8.771 * 10^{-1} = 0. \\
& y^6 - 4.633y^5 + 1.185 * 10y^4 + 1.495y^3 + \\
& \quad + 3.531 * 10y^2 - 3.707 * 10y + 3.043 = 0 \\
& x + 2.048 * 10^{-2}y^5 - 1.070 * 10^{-1}y^4 + \\
& \quad + 4.079 * 10^{-1}y^3 - 3.297 * 10^{-1}y^2 + \\
& \quad + 1.042y - 8.772 * 10^{-1} = 0.
\end{aligned}$$

$$\begin{aligned}
& y^6 - 4.244y^5 + 1.025 * 10y^4 - \\
& -6.669 * 10^{-1}y^3 + 2.922 * 10y^2 - \\
& -2.930 * 10y + 2.291 = 0 \\
x + 2.416 * 10^{-2}y^5 - 1.153 * 10^{-1}y^4 + \\
& +4.322 * 10^{-1}y^3 - 3.864 * 10^{-1}y^2 + \\
& +1.062y - 8.678 * 10^{-1} = 0.
\end{aligned}$$

STEP 3. Solving the polynomial equations in only one unknown we retain only the real solutions. For their values see the next step.

STEP 4. The solutions of these systems are, as follows: For the first system, $(0.797, 0.09239) \notin D_{0,0}$.

For the second system there are two real and four not real roots. The real roots are $(0.786, 0.09)$ and $(0.093, 0.791)$, none of these belong to $D_{0,1}$.

The third system, which has also only two real solutions gives: $(0.78, 0.085) \notin D_{0,2}$ but $(0.093, 0.791) \in D_{0,2}$.

STEP 5. Taking this last solution as the initial value for a Newton approximation method for the original nonlinear system we get: $(0.093, 0.791) \in D_{0,2}$.

This example shows that the last step in this case didn't increased the accuracy of the value obtained in the previous step, suggesting that the spline approximation could be very accurate.

Solving the remaining six systems we get one more solution, which is symmetric to the first one. These are the only two solutions located in the domain D of our search.

Acknowledgment. This work was partially supported by the Institute of Research Programmes of the Hungarian University of Transylvania, where both authors are associate professors.

References

- [1] BECKER, Th. and WEISPFENNING, V.: Gröbner bases: a Computational Approach to Commutative Algebra, Springer-Verlag, New York, 1993.
- [2] CHUI, C. K.: Multivariate splines, SIAM, Philadelphia, PA, 1988.
- [3] COX, D., LITTLE, J. and O'SHEA, D.: Ideals, varieties, and algorithms, An Introduction to Computational Algebraic Geometry and Commutative Algebra, Springer-Verlag, New York, 1997.

- [4] GREUEL, G.-M., PFISTER, G. and SCHÖNEMANN, H.: Singular 2.0. A Computer Algebra System for Polynomial Computations, Centre for Computer Algebra, University of Kaiserslautern, 2001. <http://www.singular.uni-kl.de>.
- [5] HORVÁTH, A. and FINTA, B.: Solving Nonlinear Systems of Equations on a Domain, *Bulletins for Applied and Computer Mathematics*, BAM-1973, **XCIX** (2002), 129–135. Technical University of Budapest.
- [6] MICULA, Gh. and MICULA, S.: Handbook of Splines, Kluwer Academic Publishers, Dordrecht, Boston, London, 1999.
- [7] SCHUMAKER, L. L.: Spline functions: basic theory, Wiley, New York, 1981.
- [8] VASCONCELOS, W. V.: Computational Methods in Commutative Algebra and Algebraic Geometry, Springer-Verlag, Berlin, 1998.