Numerical Optimization in Hybrid Symbolic-numeric Computation

Lihong Zhi
Key Laboratory of Mathematics Mechanization
Academy of Mathematics and Systems Science
Beijing 100080, China
lzhi@mmrc.iss.ac.cn
http://www.mmrc.iss.ac.cn/~lzhi/

ABSTRACT

Approximate symbolic computation problems can be formulated as constrained or unconstrained optimization problems, for example: GCD [3, 8, 12, 13, 23], factorization [5, 10], and polynomial system solving [2, 25, 29]. We exploit the special structure of these optimization problems, and show how to design efficient and stable hybrid symbolic-numeric algorithms based on Gauss-Newton iteration, structured total least squares (STLS), semidefinite programming and other numeric optimization methods.

Categories and Subject Descriptors: I.2.1 [Computing Methodologies]: Symbolic and Algebraic Manipulation —Algorithms; G.1.6 [Mathematics of Computing]: Numerical Analysis—Optimization;

General Terms: algorithms, experimentation

Keywords: symbolic/numeric hybrid methods, numerical optimization.

Numerical Optimization. Optimization is the minimization or maximization of a function subject to constraints on its variables. It can be written as:

$$\min_{x\in\mathbb{R}^n} f(x) \ \text{ subject to } \begin{cases} c_i(x)=0, & i\in\mathcal{E},\\ c_i(x)\geq 0, & i\in\mathcal{I}. \end{cases} \tag{1}$$

Here f and c_i are scalar-valued functions of the variables x, and $\mathcal{E},\mathcal{I}$ are sets of indices. We call f the objective function, while $c_i, i\in\mathcal{E}$ are the equality constraints and $c_i, i\in\mathcal{I}$ are the inequality constraints. Constrained optimization problems can be reformulated as unconstrained optimization problems by eliminating the constraints or replacing the constrains by penalty terms in the objective function. Global solutions are desirable, but they are usually difficult to compute, and can be unattainable sometimes [12]. Local solution is a point at which the objective function is smaller than at all other feasible points in its neighborhood. Numerical optimization algorithms usually converge to local solutions fast if they begin with a good initial guess of the optimal values of the variables. A comprehensive description of powerful numerical techniques for solving optimization problems is given in the book [24] by Nocedal and Wright. In the talk, we describe how to apply some numerical optimization methods to solve problems arise in approximate computer algebra.

Gauss-Newton Iterations. The Gauss-Newton method is used to solve nonlinear least squares problems where the objective function f has the following special form:

$$f(x)=\frac{1}{2}\sum_{j=1}^{m} r_j^2(x), \tag{2}$$

where each r_j is a smooth function from $\mathbb{R}^n$ to $\mathbb{R}$.

The Gauss-Newton iteration is given by

$$x_{k+1}=x_k+\alpha_k p_k, \tag{3}$$

where the positive scalar α_k is the step length, the search direction p_k is computed by solving

$$J_k^T J_k p_k=-J_k^T r(x_k), \tag{4}$$

where J_k is the Jacobian of $r(x_k)=[r_1(x_k),\ldots,r_m(x_k)]^T$.

The Gauss-Newton method has been applied to refine the approximate factorization [10] and the approximate GCD in [3, 33]. For example, the optimization version of the approximate factorization problem is finding a least squares solution to the non-linear system of the form

$$\min_{\mathbf{v}_1,\ldots,\mathbf{v}_r} \|F(\mathbf{v}_1,\ldots,\mathbf{v}_r)-\mathbf{f}\|_2^2,$$

where

$$F(\mathbf{v}_1,\ldots,\mathbf{v}_r)= \ \mathrm{C}^{[\mathrm{tdeg}(v_2\cdots v_r)]}(v_1)\cdots\mathrm{C}^{[\mathrm{tdeg}(v_r)]}(v_{r-1})\,\mathbf{v}_r.$$

Here $\mathbf{v}_i$ and $\mathbf{f}$ denote the coefficient vectors of polynomials v_i and f respectively, and $\mathrm{C}^{[k]}(v)$ denotes the matrix of the linear map multiplication with polynomials of total degree k as described in [10]. The Jacobian of F is a block matrix of the form:

$$[\mathrm{C}^{[\mathrm{tdeg}(v_1)]}(v_2\,v_3\cdots v_r),\ \mathrm{C}^{[\mathrm{tdeg}(v_2)]}(v_1\,v_3\cdots v_r),\ldots \\ \ldots,\mathrm{C}^{[\mathrm{tdeg}(v_r)]}(v_1\,v_2\cdots v_{r-1})].$$

*This research was partially supported by NKBRPC (2004CB318000) and the Chinese National Natural Science Foundation under Grant 10401035.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SNC'07, July 25–27, 2007, London, Ontario, Canada.
Copyright 2007 ACM 978-1-59593-744-5/07/0007 ...$5.00.

The Gauss-Newton iteration converges at a quadratic rate if the nearby local minimum is an exact factorization of f. As shown in [10], the iteration can refine the factorization significantly in very few steps. Björck [26] gives a comprehensive survey on different approaches for nonlinear least squares problems.

Structured Total Least Squares. The STLS problem can be formulated as follows:

$$\begin{gathered} \min_{\Delta A, \Delta b, x} \|\Delta A \ \Delta b\|_F^2 \\ \text{such that } (A+\Delta A)x = b + \Delta b \\ [A+\Delta A,\, b+\Delta b] \quad \text{has the same structure as} \quad [A,\, b] \end{gathered} \tag{5}$$

A brief overview of existing approaches for solving STLS problem is given in [17]. Various matrices appeared in computer algebra such as: Sylvester matrix, Bezout matrix and Ruppert matrix are all structured matrix which can be parameterized by the coefficients of polynomials. The STLS approaches in [17, 21, 27, 28] have been used to solve various problems in approximate polynomial compuations [1, 11, 12]. As an example, we illustrate the STLS approach for the Bezout matrix [30].

The Bezout matrix $B(f_1, \ldots, f_m)$ can be parameterized by a vector ζ which contains the coefficients of polynomials $f_1, \ldots, f_m$. By applying Theorem 3.2 in [4], we can transfer the GCD problem into solving the following minimization problem:

$$\min_{\Delta s \in \mathbb{R}^d} \|\Delta s\|_2 \quad \text{with} \quad \dim(\mathrm{Ker} B(s+\Delta s)) \geq k, \tag{6}$$

where $s = [f_{10}, \ldots, f_{1d_1}, \ldots, f_{m0}, \ldots, f_{md_m}]$, f_{ij} stands for the coefficient of x^j in polynomial f_i, $d_1 \geq d_2 \geq \cdots \geq d_m$ for $d_i = \deg(f_i)$, and $d = m + \sum_{i=1}^m d_i$.

Let $B_k(\zeta) = [D_1(\zeta), b(\zeta), D_2(\zeta)]$ be the first $d_1 - k + 1$ columns of $B(\zeta)$ and let $A(\zeta) = [D_1(\zeta), D_2(\zeta)]$. According to Theorem 3.3 in [4], the minimization problem (6) can be transferred into the following structured nonlinear total least squares problem:

$$\min_{\Delta s \in \mathbb{R}^d} \|\Delta s\|_2^2 \quad \text{with} \quad A(s+\Delta s)\, x = b(s+\Delta s). \tag{7}$$

By introducing the Lagrangian multipliers, and neglecting the second-order terms in Δs, the constrained minimization problem can be transformed into an unconstrained optimization problem [18, 28]:

$$L(\Delta s, x, \lambda) = \frac{1}{2}\Delta s^T \Delta s - \lambda^T (b - Ax - X\Delta s), \tag{8}$$

where $X(\zeta, x)$ is the Jacobian of $r(\zeta, x) = A(\zeta)x - b(\zeta)$ with respect to ζ. Applying the Newton method on the Lagrangian L yields:

$$M \begin{bmatrix} \Delta\tilde{s} \\ \Delta\tilde{x} \\ \Delta\tilde{\lambda} \end{bmatrix} = - \begin{bmatrix} \Delta s + X(s+\Delta s, x)^T \lambda \\ A(s+\Delta s)^T \lambda \\ A(s+\Delta s)x - b(s+\Delta s) \end{bmatrix}, \tag{9}$$

where

$$M = \begin{bmatrix} I_d & \mathbf{0}_{d\times(d_1-k)} & X(s+\Delta s, x)^T \\ \mathbf{0}_{(d_1-k)\times d} & \mathbf{0}_{(d_1-k)\times(d_1-k)} & A(s+\Delta s)^T \\ X(s+\Delta s, x) & A(s+\Delta s) & \mathbf{0}_{(m-1)d_1\times(m-1)d_1} \end{bmatrix}$$

The iterative update $x = x + \Delta\tilde{x}$, $\lambda = \lambda + \Delta\tilde{\lambda}$, $\Delta s = \Delta s + \Delta\tilde{s}$ is stopped when $\|\Delta\tilde{x}\|_2$ and/or $\|\Delta\tilde{s}\|_2$ and/or $\|\Delta\tilde{\lambda}\|_2$ becomes smaller than a given tolerance. The overall computational complexity of the algorithm depends on the number of iterations needed for the first order update. If the starting values are good, then the iteration will converge quickly. Moreover, since the matrix involved in the minimization problem has low displacement rank [9]. It would be possible to apply fast algorithm to solve these minimization problems as in [18, 19].

Semidefinite Programming. Semidefinite program considers the problem of minimizing a linear function of a variable $x \in \mathbb{R}^m$ subject to a matrix inequality:

$$\begin{gathered} \text{minimize} \qquad c^T x \\ \text{subject to} \quad F(x) \succeq 0, \end{gathered} \tag{10}$$

where

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i,$$

$F_0, \ldots, F_m \in \mathbb{R}^{n\times n}$ are symmetric matrices. The linear matrix inequality(LMI) $F(x) \succeq 0$ means that $F(x)$ is positive semidefinite. A semidefinite program is a convex optimization problem since its objective and constraint are convex. There is an associated dual problem:

$$\begin{gathered} \text{maximize} \qquad -\mathrm{Tr} F_0 Z \\ \text{subject to} \quad \mathrm{Tr} F_i Z = c_i, \quad i = 1, \ldots, m, \\ Z \succeq 0. \end{gathered} \tag{11}$$

Here Tr denotes the trace of matrix and the symmetric semidefinite matrix $Z \in \mathbb{R}^{n\times n}$ is subject to m equality constraints. Any feasible solution Z of the dual problem is a low bound of the optimal value of the primal problem (10) since

$$c^T x + \mathrm{Tr} F_0 Z = \sum_{i=1}^m \mathrm{Tr} Z F_i x_i + \mathrm{Tr} F_0 Z = \mathrm{Tr} F(x) Z \geq 0.$$

The inequality above is called weak duality. If the primal problem is strictly feasible, i.e., there exists an x with $F(x) \succ 0$ or the dual problem is strictly feasible, i.e., there exists a Z with $Z = Z^T \succ 0$, and $\mathrm{Tr} F_i Z = c_i$, then the strong duality also holds: the optimal value of the primal and the dual problems coincide.

Semidefinite programs can be solved very efficiently by primal-dual interior-point methods [22]. A worst-case analysis of interior-point methods for semidefinite programming shows that the number of operations required to solve a semidefinite program to a given accuracy grows no faster than $O(m^2 n^{5/2})$. We refer to [20, 32] for a broad overview on semidefinite programming.

Many approximate symbolic computation problems can be formulated as finding global optimum of polynomial or rational functions with or without constraints. For simplicity, let us consider the following unconstrained polynomial minimization over $\mathbb{R}^n$:

$$\begin{gathered} \text{minimize} \qquad f(x) \\ \text{subject to} \quad f(x) = \textstyle\sum_\alpha f_\alpha x^\alpha, \quad \alpha \leq 2m \end{gathered} \tag{12}$$

By introducing a nonnegative measure μ on $\mathbb{R}^n$, we call the quantity $y_\alpha = \int x^\alpha \mu(dx)$ its moment of order α. The minimization problem (12) is closely related to following convex LMI optimization problem:

$$\begin{gathered} \text{minimize} \qquad \mathbf{f}^T \mathbf{y} \\ \text{subject to} \quad M_m(\mathbf{y}) \succeq 0, \end{gathered} \tag{13}$$

where $\mathbf{f}$ is the coefficient vector of the polynomial $f(x)$, $\mathbf{y} = \{y_\alpha\}$ is the vector of moments up to order $2m$, $M_m(\mathbf{y})$ is the moment matrix of dimension $\binom{n+m}{m}$, with rows and columns labelled by

$$1, x_1, x_2, \ldots, x_n, x_1^2, x_1x_2, \ldots, x_n^2, \ldots, x_1^m, \ldots, x_n^m.$$

Suppose f^* is the global minimum of $f(x)$. If the non-negative polynomial $f(x) - f^*$ is a sum of squares(SOS) of polynomials, then f^* is also the global minimum of the problem (13). Moreover, if x^* is a global minimizer of $f(x)$, then the vector

$$\mathbf{y}^* = [x_1^*, \ldots, x_n^*, (x_1^*)^2, (x_1^*)(x_2^*), \ldots, (x_1^*)^2, \ldots, (x_n^*)^{2m}]^T$$

is a minimizer of (13).

The general case, that is, when $f(x) - f^*$ is not a sum of squares, or the minimization is over a semialgebraic set defined by polynomial equations and inequalities, the global minimum can be approximated by solving hierarchies of semidefinite relaxations based on positive semidefinite moment matrices and the dual theory of sums of squares of polynomials [15, 25]. A survey on optimization over polynomials is given by Laurent [16]. SDP techniques have been used to solving polynomial systems [2, 25, 29] and computing real radical ideals [14]. Two abstracts in this conference proceedings also describe how to apply SDP for computing approximate GCD and factorization of polynomials. Although SDP is a powerful tool for finding global optimum, the size of the problems can be solved by SDP is still limited. It is necessary to investigate the sparsity and structure of the problems arising from approximate polynomial computations [31].

1. REFERENCES

[1] Botting, B., Giesbrecht, M., and May, J. Using Riemannian SVD for problems in approximate algebra. In *Proc. 2005 International Workshop on Symbolic-Numeric* (2005), D. Wang and L. Zhi, Eds., pp. 209–219. Distributed at the Workshop in Xi'an, China, July 19–21.

[2] Chesi, G., Garulli, A., Tesi, A., and Vicino, A. An LMI-based approach for characterizing the solution set of polynomial systems. In *Proc. 39th IEEE conference on decision and control* (Sydney, Australia, 2000), pp. 1501–1506.

[3] Chin, P., and Corless, R. M. Optimization strategies for the approximate GCD problem. In Gloor [6], pp. 228–235.

[4] Diaz-Toca, G., and Gonzalez-Vega, L. Barnett's theorems about the greatest common divisor of several univariate polynomials through Bezout-like matrices. *J. Symbolic Comput. 34*, 1 (2002), 59–81.

[5] Gao, S., Kaltofen, E., May, J. P., Yang, Z., and Zhi, L. Approximate factorization of multivariate polynomials via differential equations. In Gutierrez [7], pp. 167–174.

[6] Gloor, O., Ed. *ISSAC 98 Proc. 1998 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 1998), ACM Press.

[7] Gutierrez, J., Ed. *ISSAC 2004 Proc. 2004 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 2004), ACM Press.

[8] Hitz, M. A., and Kaltofen, E. Efficient algorithms for computing the nearest polynomial with constrained roots. In Gloor [6], pp. 236–243.

[9] Kailath, T., and Sayed, A. H. Displacement structure: theory and applications. *SIAM Review 37*, 3 (1995), 297–386.

[10] Kaltofen, E., May, J., Yang, Z., and Zhi, L. Approximate factorization of multivariate polynomials using singular value decomposition. Manuscript, 22 pages. Submitted, Jan. 2006.

[11] Kaltofen, E., and Yang, Z. On exact and approximate interpolation of sparse rational functions. In *Proc. 2007 Internat. Symp. Symbolic Algebraic Comput.* (2007).

[12] Kaltofen, E., Yang, Z., and Zhi, L. Approximate greatest common divisors of several polynomials with linearly constrained coefficients and singular polynomials. In *ISSAC MMVI Proc. 2006 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 2006), J.-G. Dumas, Ed., ACM Press, pp. 169–176. Full version, 21 pages. Submitted, December 2006.

[13] Karmarkar, N., and Lakshman Y. N. Approximate polynomial greatest common divisors and nearest singular polynomials. In *ISSAC 96 Proc. 1996 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 1996), Lakshman Y. N., Ed., ACM Press, pp. 35–42.

[14] Lasserre, J., Laurent, M., and Rostalski, P. Semidefinite characterization and computation of zero-dimensional real radical ideals. Manuscript, 2007. Available at `http://homepages.cwi.nl/~monique/`.

[15] Lasserre, J. B. Global optimization with polynomials and the problem of moments. *SIAM J. on Optimization 11*, 3 (2000), 796–817.

[16] Laurent, M. Moment matrices and optimzation over polynomials. Manuscript, 2005. Available at `http://homepages.cwi.nl/~monique/`.

[17] Lemmerling, P. *Structured total least squares: analysis, algorithms and applications*. Dissertation, Katholieke Universiteit Leuven, Belgium, 1999.

[18] Lemmerling, P., Mastronardi, N., and Huffel, S. V. Fast algorithm for solving the Hankel/Toeplitz structured total least squares problem. *Numerical Algorithms 23* (2000), 371–392.

[19] Li, B., Yang, Z., and Zhi, L. Fast low rank approximation of a Sylvester matrix by structured total least norm. *J. JSSAC (Japan Society for Symbolic and Algebraic Computation) 11*, 3–4 (2005), 165–174.

[20] L.Vandenberghe, and Boyd, S. Semidefinite programming. *SIAM Review 38*, 1 (1996), 49–95.

[21] Moor, B. D. Total least squares for affinely structured matrices and the noisy realization problem. *IEEE Transactions on Signal Processing 42* (1994), 3004–3113.

[22] Nesterov, Y., and Nemirovskii, A. *Interior-Point polynomial algorithms in convex programming*. SIAM, Philidelphia, PA, 1993.

[23] Nie, J., Demmel, J., and Gu, M. Global Minimization of Rational Functions and the Nearest GCDs. *ArXiv Mathematics e-prints* (Jan 2006).

[24] Nocedal, J., and Wright, S. J. *Numerical Optimization*. Springer-Verlag, New York, 1999.

[25] Parrilo, P. A. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, May 2000. Available at `http://www.cds.caltech.edu/~pablo/`.

[26] Å. Björck. *Numerical methods for Least Squares Problems*. SIAM, Philidelphia, PA, 1996.

[27] Rosen, J. B., Park, H., and Glick, J. Total least norm formulation and solution for structured problems. *SIAM J. Matrix Anal. Applic. 17*, 1 (1996), 110–128.

[28] Rosen, J. B., Park, H., and Glick, J. Structured total least norm for nonlinear problems. *SIAM J. Matrix Anal. Applic. 20*, 1 (1999), 14–30.

[29] Sturmfels, B. *Solving systems of polynoial equation*. American Mathematical Society, CBMS regional conferences series, No. 97, Providence, Rhode Island, 2002.

[30] Sun, D., and Zhi, L. Structured low rank approximation of a Bezout matrix. *Mathematics in Computer Science* (2007), to appear.

[31] Waki, H., Kim, S., Kojima, M., and Muramatsu, M. Sums of squares and semidefinite programming relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization 17* (2006), 218–242.

[32] Wolkowicz, H., Saigal, R., and Vandenberghe, L. *Handbook of Semidefinite Programming*. Kluwer Academic, Boston, 2000.

[33] Zeng, Z., and Dayton, B. H. The approximate GCD of inexact polynomials part II: a multivariate algorithm. In Gutierrez [7], pp. 320–327.