TERA'97

Second International Workshop on (Semi)numerical Techniques in Polynomial Equation Solving

Fa.M.A.F., University of Córdoba, Argentina

September, 4–6, 1997

Foreword

TERA, in the meaning of “Turbo Evaluation and Rapid Algorithms”, is the name of a loosely organized project group spread over four countries (Argentina, France, Germany, Spain) and ten universities and research institutions. There are 34 collaborators in this group whose research and development activities are organized around two tasks, a practical and a theoretical one.

The practical task consists in the development of a program library which aims to represent a tool for attacking selected parametric polynomial and algebro–differential equation systems occuring in natural science applications and engineering. The theoretical task however consists in the search for a satisfactory definitive proof (i.e. a proof independent of conjectural open problems in theoretical computer science) that only very exceptional, “well suited”, equation systems allow an efficient treatment. This means the impossibility to construct a general purpose software solving *any* polynomial or algebro-differential equation system in a reasonable amount of time and memory space. Let us mention – by the way– that solving any one of these tasks is also subject of an award, the “Jacques Morgenstern Challenge”, set up by the University of Cantabria with the collaboration of a large number of other scientific organizations, among them the University of Buenos Aires, Argentina.

The aim of the TERA’97 workshop – second in its kind– was twofold : on one hand our purpose was to bring together the argentinean and the european parts of the TERA group in order to promote direct and personal exchange of ideas and information on work progress. On the other hand, we tried to invite several outstanding specialists in some of the main fields of pure and applied mathematics and theoretical and practical computer science which are relevant for the research and development activities within the TERA project. These fields, at first glance unrelated, are subject to a unified view through our project oriented approach. Most of them are represented during the workshop by at least one speaker or participant. We can roughly list these fields as : algebraic, semialgebraic, analytic and diophantine geometry, singularity theory, commutative and differential algebra, theory of invariants, diophantine approximation and transcendence theory from the side of pure mathematics, as numerical analysis, approximation theory, optimization, signal processing, neural networks, crystallography, robotics, biomedicine, from the side of applied mathematics and as complexity theory, computer algebra, seminumerical algorithms, data structures, programming languages and software engineering from the side of theoretical and practical computer science.

The workshop took place in Córdoba, Argentina, during the days September 4–6, 1997 at the Facultad de Matemáticas, Astronomía y Física (Fa.M.A.F.), Universidad Nacional de Córdoba, Argentina. The workshop was attended by 60 specialists coming from Argentina, Spain and Germany and the following research institutions: University of Córdoba, University of Buenos Aires, University of La Plata, University of General Sarmiento (county of Buenos Aires), University of San Andrés (Buenos Aires), University of Cantabria (Spain), Public University of Navarra (Spain), University of Alicante (Spain) and Humboldt University (Berlin, Germany). The friendly atmosphere of the University of Córdoba which kindly offered us its hospitality, made easy the communication between the specialists of such diverse disciplines.

We thank the following scientific institutions for their financial support and encouragement given to the workshop :

- Centro de Investigación y Estudios en Matemática (CIEM)
- Facultad de Matemáticas, Astronomía y Físicas (Fa.M.A.F.), Universidad Nacional de Córdoba, Argentina
- Secretaria de Ciencia y Técnica, Universidad Nacional de Córdoba, Argentina.
- Fundación Antorchas, Argentina
- Universidad de Buenos Aires and Asociación de Universidades "Grupo de Montevideo"
- Fondo para el Mejoramiento de la Calidad Universitaria (FOMEC), Argentina
- Research Project UBACyT EX-001, Universidad de Buenos Aires
  and PIP 4571 CONICET, Argentina
- Universidad de Cantabria and its Department of Mathematics, Santander, Spain
- Research Project DGCyT, PB93–0472–C02–02, Spain
- Universidad Pública de Navarra, Pamplona, Spain
- Universidad de Alcala, Madrid, Spain
- Universitat d'Alacant, Alicante, Spain
- GDR CNRS 1026 MEDICIS
  and Laboratoire GAGE, École Polytechnique, Palaiseau, France

- Humboldt Universität zu Berlin, Germany

C. Turner, R. Wachenchauzer

Program

Thursday 4/9:

10,15: Openning.

10,30: **Conference**: Joos Heintz: *The virtual world as a real trap for the scientist.*

11,30: Susana Puddu, Juan Sabia: *Eliminación de cuantificadores y straight-line programs.*

12,00: Martín Sombra: *Bounds for sparseness of the polynomials in the Nullstellensatz.*

12,30: K. Hägele, J. E. Morais, L. M. Pardo and M. Sombra: *On the intrinsic complexity of the arithmetic Nullstellensatz.*

13,00 to 14,30: Lunch.

14,30: Arcadio Niell: *On the simultaneous approximation of polynomial roots.*

15,00: Pedro Grimalt Ivars: *El método de descomposición (Adomian) y sus aplicaciones a las ciencias de la vida.*

15,30: I. Méndez Díaz: *Algoritmos branch and cut.*

16,00 to 16,30: Coffee break.

16,30: **Conference**: Bernd Bank: *Optimization and real equation solving.*

17,30: F. Menzaque, A. Majda, R. Rosales, E. Tabak, C. Turner: *Interacción no lineal de ondas en el Ecuador.*

18,00: M. Aldaz, J. Heintz, G. Matera, J.L Montaña, L.M. Pardo: *A combinatorial method for the exhibition of polynomials "hard to compute".*

Friday 5/9:

9,00: **Conference:** Teresa Krick: *Consistency of a polynomial equation system: degrees, heights and complexity.*

10,00: G. Matera: *Why geometric reasoning requires space?*

10,30: V. Braberman, K. Hägele, G. Matera, R. Wachenchauzer: *The TERA Development Group as an interface between mathematicians and computer scientists.*

11,00 to 11,30: Coffee break.

11,30: M. Giusti, K. Hägele, G. Lecerf, J. Marchand, B. Salvy: *Computing the dimension of a projective variety.*

12,00: J. Heintz, G. Matera, L.M. Pardo, R. Wachenchauzer: *Numerical stability of versal unfoldings and the intrinsic complexity of elimination.*

12,30 to 14: Lunch.

14,00: E. Segura: *Aproximación de la dinámica inversa de un manipulador robótico mediante un algoritmo estocástico de aprendizaje en redes neuronales.*

14,30: J. M. Santos, A. Cristal: *Un algoritmo de crecimiento para la construcción de redes neuronales artificiales que aproximan funciones reales multivariadas. Dos casos de estudio con orientación a la robótica.*

15,00: L. Franco, S. Cannas: *Estructura de la superficie de energía y aprendizaje en una red neuronal para el problema de suma.*

15,30 to 16,00: Coffee break.

16,00: Fernando Levstein: *Versión discreta del problema biespectral.*

16,30: **Conference**: Alicia Dickenstein: *Residuos de polinomios ralos.*

17,30 a 18,00: Coffee break.

18,00: **Conference**: Roberto Miatello: *Grupos cristalograficos de Hantzsche-Wendt y grafos dirigidos.*

Saturday 6/9:

9,00: **Conference**: Pablo Solernó: *Algoritmos generales en Geometría Semialgebraica.*

10,00: M. Almeida, L. D'Alfonso, P. Solernó: *Sobre los grados de bases de módulos libres sobre anillos de polinomios.*

10,30: N. Krasnogor: *Enfoques algorítmicos para el problema del Protein Folding.*

11,00 to 11,30: Coffee break.

11,30: B. Castaño, J. Heintz, J. Llovet, R. Martínez: *Mixed Representation of Multivariate Polynomials given by Straight-line Programs and the Computation of the GCD.*

12,00: **Conference**: Luis Miguel Pardo: *Diophantine geometry or the approximation of integers by means of integers.*

13,00: Empanadas and wine.

15,00: Discussion session.

Addresses of speakers and their coauthors

MIKEL ALDAZ
Dpto. de Matemática e Informática
Universidad Pública de Navarra
Campus de Rosadía
E-31006 Pamplona, Spain
mikaldaz@upna.es

MARCELA ALMEIDA
Depto. de Matemática, F.C.E. y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pabellón I
1428 Buenos Aires, Argentina
malmeida@dm.uba.ar

BERND BANK
Humbolt–Universität zu Berlin
Unter den Linden 6, D–10099 Berlin
bank@mathematik.hu-berlin.de

VÍCTOR BRABERMAN
Depto. de Computación, F.C.E. y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pab. I
(1428) Buenos Aires, Argentina
vbraber@dc.uba.ar

ESTEBAN DE LA CANAL
L.I.F.I.A.
Departamento de Informática
Universidad Nacional de La Plata
CC 11, 1900 - La Plata, Argentina
steve@sol.info.unlp.edu.ar

SERGIO A. CANNAS
Facultad de Matemática, Astronomía
y Física (Fa.M.A.F.)
Universidad Nacional de Cordoba
Ciudad Universitaria (5000) Cordoba
Argentina
cannas@fis.uncor.edu

BONIFACIO CASTAÑO
Dept. of Mathematics
Alcalá University, Spain
mtcastano@alcala.es

DAVID CASTRO
Depto. Matemáticas, Est. y Comp.
Facultad de Ciencias
Universidad de Cantabria
Av. de los Castros s/n
E-39071 Santander, Spain
castro@matesco.unican.es

ADRIÁN CRISTAL
Departamento de Arquitecturas
del Computador
Univ. Polit. de Catalunya, España
adrian@ac.upc.es

LISI D'ALFONSO
Depto. de Matemática, F.C.E. y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pabellón I
1428 Buenos Aires, Argentina
lisi@dm.uba.ar

Alicia Dickenstein
Depto. de Matemática, F.C.E. y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pabellón I
1428 Buenos Aires, Argentina
`alidick@dm.uba.ar`

Leonardo Franco
Facultad de Matemática, Astronomía
y Física (Fa.M.A.F.)
Universidad Nacional de Cordoba
Ciudad Universitaria (5000) Cordoba
Argentina
`franco@fis.uncor.edu`

Marc Giusti
Laboratoire GAGE
École Polytechnique
F-91128 Palaiseau, France
`giusti@marie.polytechnique.fr`

Pedro Grimalt
Departamento de Análisis Matemático
y Matemática Aplicada
Universidad de Alicante
Apartado de Correos 99
03080 Alicante, Spain

Klemens Hägele
Depto. Matemáticas, Est. y Comp.
Facultad de Ciencias
Universidad de Cantabria
Av. de los Castros s/n
E-39071 Santander, Spain
`haegele@matesco.unican.es`

Joos Heintz
Depto. de Matemáticas, Est. y Comp.
Facultad de Ciencias
Universidad de Cantabria
E-39071 Santander, Spain
`heintz@matesco.unican.es`
and
Depto. de Matemática, F.C.E. y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pabellón I
1428 Buenos Aires, Argentina
`joos@dm.uba.ar`

Natalio Krasnogor
L.I.F.I.A.
Departamento de Informática
Universidad Nacional de La Plata
Calle 50 y 115, 1900 La Plata, Argentina
`natk@sol.info.unlp.edu.ar`

Teresa Krick
Depto. de Matemática, F.C.E. y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pabellón I
1428 Buenos Aires, Argentina
`krick@dm.uba.ar`

Gregoire Lecerf
Laboratoire GAGE, École Polytechnique
F-91128 Palaiseau, France
`lecerf@marie.polytechnique.fr`

Fernando Levstein
Facultad de Matemática, Astronomía
y Física (Fa.M.A.F.)
Universidad Nacional de Cordoba
Ciudad Universitaria (5000) Cordoba
Argentina
`levstein@mate.uncor.edu`

JUAN LLOVET
Dept. of Mathematics
Alcalá University, Spain
illovet@alcala.es

A. MAJDA
Courant Institute
New York, USA
majda@cims.nyu.edu

RAQUEL MARTÍNEZ
Dept. of Computer Science
Polytechnical Univ. Madrid, Spain
raquelm@etsii.upm.es

PABLO E. MARTINEZ LOPEZ
L.I.F.I.A.
Departamento de Informática
Universidad Nacional de La Plata
CC 11, 1900 - La Plata, Argentina
fidel@sol.info.unlp.edu.ar

JOEL MARCHAND
Laboratoire GAGE, École Polytechnique
F-91128 Palaiseau, France
jma@marie.polytechnique.fr

GUILLERMO MATERA
Instituto de Ciencias
Universidad Nac. de Gral. Sarmiento
Roca 850 (1663) San Miguel
Pcia. de Buenos Aires, Argentina
and
Depto. de Matemáticas, F.C.E y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pab. I
(1428) Buenos Aires, Argentina
gmatera@dm.uba.ar

ISABEL MENDEZ DÍAZ
Depto. de Computación, F.C.E y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pab. I
(1428) Buenos Aires, Argentina
imendez@dc.uba.ar

F. MENZAQUE
Facultad de Matemática, Astronomía
y Física (Fa.M.A.F.)
Universidad Nacional de Cordoba
Ciudad Universitaria (5000) Cordoba
Argentina
menzaque@mate.uncor.edu

ROBERTO MIATELLO
Facultad de Matemática, Astronomía
y Física (Fa.M.A.F.)
Universidad Nacional de Cordoba
Ciudad Universitaria (5000) Cordoba
Argentina
miatello@mate.uncor.edu

JOSÉ L. MONTAÑA
Dpto. de Matemática e Informática
Universidad Pública de Navarra
Campus de Rosadía
E-31006 Pamplona, Spain
pepe@upna.es

J.E. MORAIS
Dpto. de Matemática e Informática
Universidad Pública de Navarra
Campus de Rosadía
E-31006 Pamplona, Spain
morais@matesco.unican.es

ARCADIO M. NIELL
Facultad de Matemática, Astronomía
y Física (Fa.M.A.F.)
Universidad Nacional de Cordoba
Ciudad Universitaria (5000) Cordoba
Argentina
niell@mate.uncor.edu

LUIS M. PARDO
Depto. de Matemáticas, Est. y Comp.
Facultad de Ciencias
Universidad de Cantabria
E-39071 Santander, Spain
pardo@matesco.unican.es

DAVID PELTA
L.I.F.I.A.
Departamento de Informática
Universidad Nacional de La Plata
CC 11, 1900 - La Plata, Argentina
davp@ada.info.unlp.edu.ar

SUSANA PUDDU
Depto. de Matemáticas, F.C.E. y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pab. I
(1428) Buenos Aires, Argentina
spuddu@dm.uba.ar

M.J. PUJOL
Departamento de Análisis Matemático
y Matemática Aplicada
Universidad de Alicante
Apartado de Correos 99
03080 Alicante, Spain

MAURICE ROJAS
Massachusetts Institute of Technology
Mathematics Department
77, Massachusetts Avenue, 2–236
Cambridge, MA 02139, USA
rojas@mit.edu.ar
http://www-math.mit.edu/rojas

R. ROSALES
Massachusetts Institute of Technology
Mathematics Department
77, Massachusetts Avenue, 2–236
Cambridge, MA 02139, USA
rrr@math.mit.edu

JUAN SABIA
Depto. de Matemáticas, F.C.E. y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pab. I
(1428) Buenos Aires, Argentina
jsabia@dm.uba.ar

BRUNO SALVY
Projet ALGO
INRIA Rocquencourt
F-78153 Le Chesnay Cedex, France
Bruno.Salvy@inria.fr

JUAN MIGUEL SANTOS
Departamento de Computacion
F.C.E. y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pab. I
(1428) Buenos Aires, Argentina
jmsantos@dc.uba.ar

ENRIQUE CARLOS SEGURA
Depto. de Computacion, F.C.E. y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pab. I
1428 Buenos Aires, Argentina
esegura@dc.uba.ar

PABLO SOLERNÓ
Universidad de San Andrés
Vito Dumas 284
1644 Victoria, Argentina
and
Depto. de Matemáticas, F.C.E. y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pab. I
(1428) Buenos Aires, Argentina
psolerno@dm.uba.ar

MARTÍN SOMBRA
Departamento de Matemática,
Universidad Nacional de La Plata,
Calle 50 y 115, 1900 La Plata, Argentina
sombra@mate.unlp.edu.ar

E. TABAK
Courant Institute
New York, USA
tabak@cims.nyu.edu

CRISTINA TURNER
Facultad de Matemática, Astronomía
y Física (Fa.M.A.F.)
Universidad Nacional de Cordoba
Ciudad Universitaria (5000) Cordoba
Argentina
turner@mate.uncor.edu

ROSITA WACHENCHAUZER
Depto. de Computación, F.C.E. y N.
Universidad de Buenos Aires
Ciudad Universitaria, Pab. I
(1428) Buenos Aires, Argentina
and
Facultad de Matemática, Astronomía
y Física (Fa.M.A.F.)
Universidad Nacional de Cordoba
Ciudad Universitaria (5000) Cordoba
Argentina
rosita@dc.uba.ar

Contents

A combinatorial method for the exhibition of polynomials hard to compute

M. Aldaz, J. Heintz, G. Matera, J.L. Montaña, L.M. Pardo

Abstract: We exhibit a new method for showing lower bounds for the time complexity of polynomial evaluation procedures given by straight–line programs. Time, denoted by L, is measured in terms of nonscalar arithmetic operations. The time complexity function considered here is L^2. As main difference with the previously known methods to study this problem, our general complexity method is purely combinatorial and does not need neither number theory nor powerful tools from algebraic geometry.
Using this method we are able to exhibit new families of polynomials "hard to compute" (this means that the time complexity function L^2 increases linearly in the degree, for some universal constant $c > 0$). We are also able to present in an uniform and easy way, almost all the known specific families of univariate polynomials which are known to be hard to compute. We also show how this method can be used to establish specific new families of linear transformations which are hard to evaluate. Our method can also be applied to classical questions of transcendence in number theory and geometry. A list of (old and new) formal power series is given whose transcendence can be proved easily by our method.

Keywords: Algebraic complexity theory, straight–line program, transcendence proofs.

Subject: 68Q40, 68Q25.

Background and results.

The study of complexity issues for straight–line programs evaluating univariate polynomials is a standard subject in Theoretical Computer Science. One of the most fundamental tasks in this domain is the exhibition of *explicit* families of univariate polynomials which are "hard to evaluate" in some given context. Following [14], [3] and [16] "almost all" univariate polynomials of degree d need for their evaluation at least $\Omega(d)$ additions/subtractions, $\Omega(d)$ scalar multiplications/divisions, and $\Omega(\sqrt{d})$ nonscalar multiplications/divisions. A family $(F_d)_{d\in\mathbb{N}}$ of univariate polynomials F_d satisfying the condition $\deg F_d = d$ is called *hard to compute* in a given complexity model if there exists a constant $c > 0$ such that any straight–line program evaluating the polynomial F_d requires the execution of at least $\Omega(d^c)$ arithmetic operations which count in that model.

In this present contribution we shall restrict ourselves to the nonscalar complexity model. This model is well suited for lower bound considerations and does not represent any limitation for the generality of our statements.

Families of specific polynomials which are hard to compute where first considered by Strassen in [21]. The method used in [21] was later refined by Schnorr in [18] and Stoss in [20]. Heintz and Sieveking introduced in [11] a considerably more adaptive method which allowed the exhibition of quite larger classes of specific polynomials which are hard to compute. However in its beginning the application of this new method was restricted to polynomials with *algebraic* coefficients. In [9] the method of Heintz–Sieveking was adapted to polynomials given by their roots and this adaption was considerably simplified in [2].

Finally the methods of [21] and [11] where unified to a common approach in [1]. This new approach is based on effective elimination and intersection theory and its consequences for diophantine geometry (see e.g. [6], [12] and [17]). This method allows for the first time applications to polynomials having only integer roots.

The results of the present contribution are based on a new, considerable simplified version of the unified approach mentioned before. Geometric considerations are replaced by simple counting arguments which makes our new method more flexible and adaptive. Our new method is inspired in [19] and [2] and relies on a counting technique developed in [21] (see also [18] and [20]).

With our method we are able to prove the following (old and new) complexity results:

(*i*) Polynomials with integer coefficients of the form

$$\sum_{0\leq j\leq d} 2^{j!}X^j$$

satisfy a lower bound $L^2 = \Omega(\frac{d}{\log_2 d})$.

(*ii*) Polynomials with integer roots of the form

$$\prod_{0\leq j\leq d} (X - 2^{2^j})$$

and

$$\prod_{0\leq j\leq d} (X - 2^{j!})$$

satisfy a lower bound $L^2 = \Omega(\frac{d}{\log_2 d})$.

(*iii*) More generally, polynomials with integer coefficients of the form

$$\sum_{0\leq j\leq d} 2^{2^{\lfloor \sqrt[k]{j}\rfloor}}X^j$$

and

$$\prod_{0\leq j\leq d}(X-2^{2^{\lfloor\sqrt[k]{j}\rfloor}})$$

satisfy the lower bound $L^2=\Omega(\frac{\sqrt[k]{d}}{\log_2 d})$ for any natural number $k\geq 1$.

(iv) For f_j being the j–th Fibonacci number the polynomials

$$\sum_{0\leq j\leq d} 2^{f_j}X^j$$

and

$$\prod_{0\leq j\leq d}(X-2^{f_j})$$

satisfy the lower bound $L^2=\Omega(\frac{d}{\log_2 d})$.

In all these examples L denotes the number of nonscalar multiplications/divisions necessary to evaluate the polynomial under consideration.

Our method can also be applied to the following families of polynomials which are known to be hard to compute:

(i) Polynomials with integer coefficients of the form $\sum_{0\leq j\leq d}2^{2^j}X^j$ (see [21]).

(ii) Polynomials with algebraic coefficients of the form $\sum_{0\leq j\leq d}e^{\frac{2\pi i}{j}}X^j$ [11] or of the form $\sum_{0\leq j\leq d}\sqrt{p_j}X^j$, where p_j is the j–th prime number [7].

(iii) Polynomials with algebraic coefficients given by their roots $\prod_{1\leq j\leq d}(X-\sqrt{p_j})$ [9].

Applying our method we are also able to recover the lower complexity bounds of [11] for systems of linear polynomials with algebraic coefficients. Moreover we are able to exhibit systems of linear polynomials with integer coefficients that have a lower complexity bound $\frac{d}{\log_2 d}$.

Our general complexity method can also be applied to classical questions of transcendence in number theory and geometry. Our method implies the following (old and new) transcendence results. The following power series belonging either to $\mathbb{Q}[[X]]$ or to $\mathbb{Q}_{alg}[[X]]$ are transcendental over the function field $\mathbb{C}(X)$:

(i) $\sum_{j\geq 0}\frac{1}{2^{2^j}}X^j$.

(ii) More generally, $\sum_{j\geq 0}\frac{1}{2^{2^{\lfloor\sqrt[k]{j}\rfloor}}}X^j$, for $k\geq 1$.

(iii) $\sum_{j\geq 0}\frac{1}{2^{2^{\lfloor\log_2^k j\rfloor}}}X^j$, for $k\geq 3$.

(iv) $\sum_{j\geq 1}\sqrt{p_j}X^j$, for p_j being the j–th prime number.

(v) $\sum_{j\geq 1} e^{\frac{2\pi i}{j}} X^j$.

Finally, we observe that our method cannot be applied to polynomials with "small" integer roots. Following [13] polynomials of this type are hard to compute under the assumption that factoring is "hard". Results in this direction are of great interest because of their close relationship to the open question whether geometric elimination is an intrinsically hard problem or not (see e.g. [9], [15] and [4] for evidence of this connection).

References

[1] M. ALDAZ, J. HEINTZ, G. MATERA, J.L. MONTAÑA, L.M. PARDO: *Time–space tradeoffs in algebraic complexity theory.* Submitted in Computational Complexity (1996).

[2] W. BAUR: *Simplified lower bounds for polynomials with algebraic coefficients.* To appear in Journal of Complexity (1996).

[3] E.G. BELAGA: *Evaluation of polynomials of one variable with preliminary processing of the coefficients.* In *A.A. Lyapunov*, editor, *Problems in Cybernetics* **5**, Pergamon Press (1961) 1–13.

[4] L. BLUM, F. CUCKER, M. SHUB, S. SMALE: *Real Complexity and Computation.* Manuscript, to appear in Springer Verlag (1997).

[5] P. BÜRGISSER, M. CLAUSEN, A. SHOKROLLAHI: *Algebraic Complexity Theory.* A Series of Comprehensive Studies in Mathematics **315**, Springer–Verlag (1996).

[6] N. FITCHAS, A. GALLIGO, J. MORGENSTERN: *Precise sequential and parallel complexity bounds for the quantifier elimination over algebraically closed fields.* J. Pure Appl. Algebra **67** (1990) 1–14.

[7] J. VON ZUR GATHEN, V. STRASSEN: *Some polynomials that are hard to compute.* Theoretical Computer Science **11(3)** (1981) 159–171.

[8] J. HEINTZ, G. MATERA: Private communication (1994).

[9] J. HEINTZ, J. MORGENSTERN: *On the intrinsic complexity of elimination theory.* Journal of Complexity **9** (1993) 471–498.

[10] J. HEINTZ, C.P. SCHNORR: *Testing polynomials which are easy to compute. Proc. Twelfth Ann. ACM Symp. Theor. Comput.* (1980) 262–268; also in *Logic and Algorithmic. An International Symposium held in Honour of Ernst Specker*, Monographie No. **30** de *l'Enseignement de Mathématiques*, Genève (1982) 237–254.

[11] J. HEINTZ, M. SIEVEKING: *Lower bounds for polynomials with algebraic coefficients.* Theoretical Computer Science **11** (1980) 321–330.

[12] T. Krick, L. M. Pardo: *A Computational Method for Diophantine Approximation.* In *L. González-Vega and T. Recio*, eds., *Algorithms in Algebraic Geometry and Applications. Proceedings of MEGA'94*, Progress in Mathematics **143**, Birkhäuser Verlag (1996) 193–254.

[13] R.J. Lipton: *Straight-line complexity and integer factorization. Proc. First International Symposium, ANTS-I*, eds. *L.M. Adleman and M.–D. Huang*. Lectures notes in Computer Science **877**, Springer–Verlag (1994) 71–79.

[14] T.S. Motzkin: *Evaluation of polynomials.* Bull. Am. Soc. **61** (1955) 163.

[15] L.M. Pardo: *How lower and upper complexity bounds meet in elimination theory. Applied Algebra, Algebraic Algorithms and Error–Correcting Codes, Proc. AAECC–11*, eds. *G. Cohen, M. Giusti and T. Mora*. Lecture Notes in Computer Science **948**, Springer Verlag (1995) 33–69.

[16] M.S. Paterson, L.J. Stockmeyer; *On the number of non–scalar multiplications necessary to evaluate polynomials.* SIAM J. Comput. **2** (1973) 80–96.

[17] S. Puddu, J. Sabia: *An effective algorithm for quantifier elimination over algebraically closed fields using straight line programs.* To appear in Journal of Pure and Applied Algebra (1997).

[18] C.P. Schnorr: *Improved lower bounds on the number of multiplications/divisions which are necessary to evaluate polynomials.* Theoretical Computer Science **7** (1978) 251–261.

[19] V. Shoup, R. Smolensky; *Lower bounds for polynomial evaluation and interpolation problems.* Volume **32** of FOCS (1991) 378–383.

[20] H. J. Stoss: *On the representation of rational functions of bounded complexity.* Theoretical Computer Science **64** (1989) 1–13.

[21] V. Strassen: *Polynomials with rational coefficients which are hard to compute.* SIAM J. Comput. **3** (1974) 128–149.

On the degrees of bases of free modules over a polynomial ring

Marcela Almeida, Lisi D'Alfonso, Pablo Solernó

Research was partially supported by UBACYT and PID-CONICET.

Our article deals with the computation of the solutions of linear equation systems over a polynomial ring. A seminal paper of E. Mayr and A. Meyer [18] shows the limitations of linear algebra methods as a tool in effective commutative algebra. In particular, Mayr-Meyer's monoid leads to an intrinsic hyperexponential growth of the degrees of the syzygies.

In terms of linear equation systems this fact can be restated saying that there exist families of polynomial matrices such that every system of generators of their kernels contains a vector with at least one coordinate of double exponential degree. More precisely, in [6, Corollaire, pag.10] the following result is shown :

Let $\varepsilon > 0$*. Let* n *and* D *be integers such that* $n \geq 10$, $D \geq 3$, $D \geq 2 + \frac{1}{32\varepsilon}$*. There exists a polynomial sequence* $P_1, \ldots, P_n$ *of degree bounded by* D *in* $A := k[x_1, \ldots, x_n]$ *(where* $P_1 := x_1$, $P_2 := x_2$*) such that any system of generators of the* A*-submodule of* A^n *consisting of all the sequences* U_i *with* $\sum_i U_i P_i = 0$*, contains at least one vector whose first coordinate has degree* $\geq N$*, where* $\log_2 \log_2 N > (\frac{1}{8} - \varepsilon)n + \log_2 \log_2 D - \frac{9}{4}$.

However, under certain additional assumptions on the matrix associated to the linear system, more precise estimations can be made. For example, if the matrix is unimodular (i.e. the rows can be extended to a basis of the whole space), a single exponential upper bound for the degree of a basis of its kernel is given in [4, Corollary 3.2].

In the present paper we treat the more general case where the columns of the matrix generate a free A-module (in particular, the Quillen-Suslin Theorem assures that the kernel is also free). In this case it is not too difficult to show a polynomial upper bound for the degree of a system of generators of the kernel (see [2, Corollary 10]). Nevertheless our purpose here is to find bases of low degree for the kernel and the image.

More precisely, let k be an infinite field, $A := k[x_1, ..., x_n]$ be the polynomial ring in the indeterminates $x_1, \ldots, x_n$ and $F \in A^{N \times M}$ be a matrix such that $\operatorname{Im} F$ is A-free.

Denote by D the maximum of the degrees of the entries of F and by s the rank of F. Then we have :

Theorem *There exists a basis* $\{v_1, \ldots, v_M\}$ *of* A^M *such that :*

- $\{v_1, \ldots, v_{M-s}\}$ *is a basis of* $\operatorname{Ker} F$*;*
- *the coordinates of the vectors* v_j *have degrees of order* $((M-s)sD)^{O(n^3)}$ *for* $j = 1, \ldots, M-s$*.*
- $\{F(v_{M-s+1}), \ldots, F(v_M)\}$ *is a basis of* $\operatorname{Im} F$*.*
- *the coordinates of the vectors* v_j *have degrees of order* $((M-s)sD)^{O(n^4)}$ *for* $j = M-s+1, \ldots, M$*.*

Under suitable stronger conditions (for instance, if F corresponds to the matrix of a linear projection) the degree upper bounds of the Theorem can be slightly improved.

The methods we use in order to prove the main theorem are strongly inspired on the works of D.Quillen, A.Suslin, L.Vaserstein and M.Hochster related to the resolution of the so called "Serre's Conjecture" (let us mention here the remarkable books of T.Y.Lam [15] and E.Kunz [13]). We combine this approach with the effective version of Hilbert Nullstellensatz (see [11], [8] and the references given in [3] and [20]) and its consequences in the quantitative study of polynomial unimodular matrices following [4]. Other approaches on effective Quillen-Suslin Theorem may be found in [16] and [17].

The theorem stated before can be applied in the following situation ocurring frequently in effective commutative algebra : let k be an infinite field and $f_1, \ldots, f_{n-r}$ be a regular sequence in $k[x_1, \ldots, x_n]$ of degrees bounded by an integer d. Suppose that the variables $x_1, \ldots, x_n$ are in Noether position with respect to the polynomials f_i (i.e. the natural map $k[x_1, \ldots, x_r] \to k[x_1, \ldots, x_n]/(f_1, \ldots, f_{n-r})$ is an injective and integral morphism).

Write $R := k[x_1, \ldots, x_r]$ and $S := k[x_1, \ldots, x_n]/(f_1, \ldots, f_{n-r})$. It is well known (see for example [7, Corollary 18.17] or [10, Lemma 3.3.1]) that, under these conditions, S is a locally free R-module of finite rank (bounded by d^{n-r} following Bezout's Inequality) and hence free (Quillen-Suslin Theorem).

This situation appears frequently in problems related to effective elimination theory (see [19], [10], [5], [1], [12], [9]). In this context it is natural to ask about quantitative properties of bases of the module S. To our knowledge the only significative result for this problem deals with the homogeneous case, where a basis whose coordinates have single exponential degree is obtained with the aid of elementary properties of Gröbner bases.

In this sense we obtain the following result :

Theorem *Suppose that S is a reduced ring. Then, there exist a basis of S over R consisting of polynomials of degrees of order $d^{O\left((n-r)r^4\right)}$.*

The proof of this theorem combines the previous results of linear algebra over the polynomial ring with consequences of Gorenstein duality theory following [14].

The methods of the proofs of both theorems are explicit and they can be easily transformed into algorithmic procedures; however their complexity bounds are too bad, even for theoretical purposes. Therefore the problem of how to find single exponential algorithms remains open.

References

[1] M. Alonso, E. Becker, M.-F. Roy, T. Wörmann : *Zeros, Multiplicities and Idempotents for Zerodimensional Systems.* Effective Methods in Alg. Geom. (MEGA '94), Progress in Math. **143**, Birkhäuser Verlag (1996).

[2] I. Armendáriz, P. Solernó : *On the computation of the radical of polynomial complete intersection ideals.* Appl. Algebra, Algebraic Algorithms and Error-Correcting Codes (G.Cohen, M.Giusti & T.Mora, eds.), Lect. Notes Comp. Sci. **948**, Springer-Verlag (1995) 106–119.

[3] C. Berenstein, D. Struppa : *Recent improvements in the Complexity of the Effective Nullstellensatz.* Linear Algebra and its Appl. **157** (1991) 203–215.

[4] L. Caniglia, G. Cortiñas, S. Danón, J. Heintz, T. Krick, P. Solernó : *Algorithmic aspects of Suslin's Proof of Serre's Conjecture.* Comput. Complexity **3**, Birkhäusser (1993) 31–55.

[5] J.-P. Cardinal : *Dualité et algorithmes itératifs pour la résolution de systèmes polynomiaux.* Thesis Université de Rennes (1993).

[6] M. Demazure : *Le monoïde Mayr et Meyer.* Notes Informelles de Calcul Formel, Ecole Polytechnique, Palaiseau (1984).

[7] D. Eisenbud : *Commutative Algebra with a view toward Algebraic Geometry.* Graduate Texts in Math. **150**, Springer-Verlag (1994).

[8] N. Fitchas, A. Galligo : *Nullstellensatz effectif et Conjecture de Serre (théorème de Quillen-Suslin) pour le Calcul Formel.* Math. Nachr. **149** (1990) 231–253.

[9] M. Giusti, J. Heintz, J. Morais, J. Morgenstern, L. Pardo : *Straight-line Programs in Geometric Elimination Theory.* J. Pure and Appl. Algebra (1997).

[10] M. Giusti, J. Heintz, J. Sabia : *On the efficiency of effective Nullstellensatz.* Comput. Complexity **3**, Birkhäusser (1993) 56–95.

[11] J. Kollár : *Sharp effective Nullstellensatz.* J. Am. Math. Soc. **1** (1988) 963–975.

[12] T. Krick, L. Pardo : *A computational Method for Diophantine Approximation.* Effective Methods in Alg. Geom. (MEGA '94), Progress in Math. **143**, Birkhäuser Verlag (1996) 193–253.

[13] E. Kunz : *Introduction to Commutative Algebra and Algebraic Geometry.* Birkhäuser (1985).

[14] E. Kunz : *Kähler Differential.* Adv. Lect. in Math. Vieweg Verlag (1986).

[15] T. Lam : *Serre's Conjecture.* Springer LN Math. **635**, Springer-Verlag (1978).

[16] A. Logar, B. Sturmfels : *Algorithms for Quillen-Suslin Theorem.* J. Algebra **145** (1992) 231–239.

[17] H. Lombardi : *La structure d'un module projectif de type fini et le contenu constructif d'un principe local-global.* Preprint (1996).

[18] E. Mayr, A. Meyer : *The complexity of the word problem for commutative semigroups and polynomial ideals.* Adv. in Math. **46** (1982) 305–329.

[19] F. Rossi, W. Spangher : *Some effective methods in the openness of loci for Cohen-Macaulay and Gorenstein properties.* Effective Methods in Alg. Geom. (MEGA '90) (T.Mora & C.Traverso, eds.), Progress in Math. **94**, Birkhäusser (1990) 441–455.

[20] B. Teissier: *Résultats récents d'algèbre commutative effective.* Séminaire Bourbaki 1989-1990, Astérisque vol **189–190** (1991) 107–131.

The TERA development group as an interface between mathematicians and computer scientists

V. Braberman, K. Hägele, G. Matera, R. Wachenchauzer

It is well known that the currently available software for polynomial equation system solving does not live up to the expectations of their potential users when it comes to the computation of realistically sized problems. The typical symptoms are out-of-memory errors or non-terminating programs.

There is strong theoretical evidence that elimination procedures face major complexity problems, for example it has been shown that algebraic elimination problems are typically EXSPACE-complete problems [8]. Algorithms based on Standard or Gröbner bases (using a dense representation of the multivariate polynomials) have necessarily exponential time complexity lower bounds (see for example [3], [2], [7]). Note that this is also the case for geometric elimination procedures in the sense of [4], when the output is required to be given in dense representation. Considerations of a minimum numerical stability necessary for the correct production of solutions with a *general* purpose solver lead to the demonstration of exponential time complexity lower bounds in [6].

A fundamental conclusion to be drawn from these results is twofold: on the one hand, it is impossible to avoid the exponential time complexity of elimination algorithms while insisting on a dense representation, and on the other hand, one has to give up the idea of building general purpose solvers, ignoring thus any geometric or other intrinsic properties of the input systems.

Based on these observations, the theoretical work of the TERA group and other scientists (mainly based on the use of alternative, problem-adapted data structures, called *straight–line programs*) has produced many interesting results during the last few years (see e.g. the TERA web pages[1]). This work resulted in two papers [4] and [1], exhibiting an algorithm for the solution of geometric elimination problems. The time and space complexity is polynomial in the syntactic parameters and a suitably defined geometric degree of the input system.

As a consequence, it becomes now necessary to verify whether these algorithms are suitable for the real world. The TERA Development group[2] was organized to provide a first working prototype for experimental purposes.

This implies working on many different aspects at the same time:

---

[1] `http://hilbert.matesco.unican.es/tera/` `http://www.dm.uba.ar/tera/`

[2] E–mail: `teradev@hilbert.matesco.unican.es`

- The mathematical design of the algorithms described in [4] and [1] has to be reviewed. A detailed description of all the necessary ingredients has to be written.

- The underlying linear algebra has to be redesigned for the specific mathematical structures appearing during the algorithms, taking into account the data-structures used for the representation of the theoretical concept of *straight–line programs*.

- As the idea of using *straight–line programs* corresponds in a natural way to a compression of the appearing objects by a functional encoding of the algorithm necessary for its evaluation, one of the main concerns is the analysis and design of suitable Pebble Games (Evaluation schemes for arithmetic circuits) for these specific algorithms.

- The implementation of the resulting algorithms for the construction of a working prototype requires the development of a programming environment and associated tools for *straight–line programs*. This corresponds to the design and efficient implementation of an abstract data type of *straight–line programs*, be it in `C++` or a lazy functional programming language.

- Finally, but not the least demanding, will be the software-engineering tasks and realization of the practical computer science aspects such as Machine Environment, Multi Processing and Distributed Computing, Coding Standards and Reuse of Standard Software.

Facing these tasks, the TERA development group is currently composed of members of the TERA project as well as other members from the Departments of Mathematics and Computer Science of the School of Science of the University of Buenos Aires, and the School of Mathematics, Astronomy and Physics of the University of Córdoba, Argentina.

References

[1] B. Bank, M. Giusti, J. Heintz, M. Mandel, G. Mbakop: *Polar varieties and efficient real equation solving: the hypersurface case.* Journal of Complexity **13**(1) (1997) 5–27.

[2] T. Dubé: *The structure of polynomial ideals and Gröbner basis.* SIAM Journal of Computing **19** (1990) 750–773.

[3] M. Giusti: *Complexity of standard basis in projective dimension zero.* Lecture Notes in Comp. Science **378**, Springer (1989) 333–335.

[4] M. Giusti, K. Hägele, J. Heintz, J.E. Morais, J.L. Montaña, L.M. Pardo: *Lower Bounds for Diophantine Approximation.* Proceedings of MEGA'96, Journal of Pure and Applied Algebra **117 & 118** (1997) 277–317.

[5] M. Giusti, J. Heintz, J.E. Morais, J. Morgenstern, L.M. Pardo: *Straight–Line Programs In Geometric Elimination Theory.* To appear in J. of Pure and App. Algebra (1997).

[6] J. Heintz, G. Matera, L.M. Pardo, R. Wachenchauzer: *About the intrinsic complexity of parametric elimination procedures.* WAIT'97, Buenos Aires, Argentina (1997).

[7] T. Krick, A. Logar: *Membership problems, Representation problems and the computation of the radical for one–dimensional ideals.* Progress in Mathematics **94**, Birkhäusser (1991) 203–216.

[8] E. Mayr, A. Meyer: *The complexity of the word problem for commutative semigroups and polynomial ideals.* Adv. in Math. **46** (1982).

Mixed Representation of Multivariate Polynomials given by Straight-line Programs and the Computation of the GCD

Bonifacio Castaño, Joos Heintz, Juan Llovet, Raquel Martínez

When a Computer Algebra System shows a poor performance on a "simple" input (e.g. in the case of an apparently "easy" polynomial equation solving problem), one may ask whether the underlying algorithms are efficient enough or whether the implementation was done in an accurate way. However it may happen that both algorithms and implementation are almost optimal and nevertheless our Computer Algebra software is unexpectedly not able to solve our problem.

In this case one may consider the possible influence of two factors:

1. The apparently simple input might hide a semantic structure (e.g. an algebraic variety) which is inherently intricate.

2. The data structure used is not well-suited for our problem.

The influence of both factors may be reflected by an intrinsically long output (see e.g. [2], [8], [10], [7]). If the influence of the first factor prevails, there is almost nothing to do. In this case one has to consider the given problem as intractable. In the sequel, we shall therefore focus our attention on the second factor of influence.

This contribution is devoted to an experimental and application oriented discussion of two fundamental Computer Algebra problems related to polynomial equation solving. These two problems have the property that the second factor of influence predominates.

More precisely, we consider the problem of solving (by means of rational functions) "large" linear equation systems with polynomial entries and a particular, but important case of this problem, namely the computation of the greatest common divisor (GCD) of two given polynomials. Both computational problems become extremely intricate for existing Computer Algebra systems if polynomials in several (let us suppose "many") variables are involved (even in the case that the input polynomials are sparse).

In this paper we present a computer program which computes efficiently the determinant (and all the coefficients of the characteristic polynomial) of a given square matrix whose entries are multivariate polynomials. Moreover this program is able to compute the GCD of two multivariate polynomials.

The new aspect of this program is its consequent application of a data structure which is unusual in Computer Algebra implementations: the straight-line program (SLP) or arithmetic circuit encoding of polynomials, rational functions, integers and rational numbers (see e.g. [1], [4], [9], [3], [5], [6]).

The description of our computer program is complemented by an experimental comparison of its performance with that of a traditional Computer Algebra system. For the sake of transparence we have chosen the MAPLE package as contrast software.

Our program represents a first practical approach to a much more ambitious implementation project, made possible by recent theoretical advance in algorithmics for polynomial equation solving. Our final aim is the construction of a symbolic solver, based on the SLP representation of polynomials and integers, which is able to find the solutions of particular "well suited" polynomial equation systems efficiently.

References

[1] P. Bürgisser, M. Clausen, A. Shokrollahi: *Algebraic Complexity Theory.* A Series of Comprehensive Studies in Mathematics **315**, Springer–Verlag (1996).

[2] J. Davenport, J. Heintz: *Real quantifier elimination is doubly exponential.* J. Symbolic Computation **5** (1988) 29–35. Reprinted in: *Algorithms in Real Algebraic Geometry*, ed. by Denis A. Arnon and Bruno Buchberger, Academic Press (1988) 29–35.

[3] M. Giusti, K. Hägele, J. Heintz, J. E. Morais, J. L. Montaña, L. M. Pardo: *Lower Bounds for Diophantine Approximation. Proceedings of MEGA'96*, Journal of Pure and Applied Algebra **117,118** (1997) 277–317.

[4] M. Giusti, J. Heintz: *La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial.* In D. Eisenbud and L. Robbiano, eds., *Computational Algebraic Geometry and Commutative Algebra, Symposia Matematica* vol. XXXIV, 216–256. Cambridge University Press, 1993.

[5] M. Giusti, J. Heintz, J. E. Morais, J. Morgenstern, L. M. Pardo: *Straight–Line Programs In Geometric Elimination Theory.* To appear in J. of Pure and App. Algebra (1997) 1–46.

[6] K. Hägele, J.L. Montaña: *Polynomial random test for the equivalence problem of integers given by straight–line programs.* Publication University of Cantabria (1996).

[7] J. Heintz, G. Matera, L.M. Pardo, R. Wachenchauzer: *About the intrinsic complexity of parametric elimination procedures.* WAIT'97, Buenos Aires, Argentina (1997).

[8] J. Heintz, J. Morgenstern: *On the intrinsic complexity of elimination theory.* Journal of Complexity **9** (1993) 471–498.

[9] T. Krick, L. M. Pardo: *A Computational Method for Diophantine Approximation.* In L. González-Vega and T. Recio, eds., *Algorithms in Algebraic Geometry and Applications. Proceedings of MEGA'94.* Progress in Mathematics **143**, Birkhäuser Verlag (1996) 193–254.

[10] L. M. Pardo: *How lower and upper complexity bounds meet in elimination theory.* In G. Cohen, H. Giusti, and T. Mora, eds., *Applied Algebra, Algebraic Algorithms and Error Correcting Codes. Proceedings of AAECC-11*, LNCS **948**, Springer (1995).

On the intrinsic complexity of diophantine geometry

D. Castro, K. Hägele, J.E. Morais, L.M. Pardo

Research was partially supported by the following Spanish grants : PB93–0472–C02–02 and PB96–0671–C02–02.

Keywords. Polynomial system solving, elimination, geometric degree, arithmetic networks, straight–line programs, complexity.

This talk focuses on the problem of efficient resolution of multivariate polynomial equation systems. Whereas the meaning of the word *resolution* depends on the mathematical context, the *efficiency* of a method is measured by different theoretical computation models which pretend to give realistic approximations of real-world computers. All these different models have one intrinsic limitation: The codification of the appearing objects is necessarily made over a finite alphabet.

Here, we will discuss the consequences of this observation for a complexity study of the following problem:

Problem 1 (Diophantine Approximation Problem) *Let* $V \subset \mathbb{C}^n$ *be a zero-dimensional diophantine variety not intersecting* $\mathbb{Q}[i]^n$ *given by a regular sequence* $f_1, \ldots, f_n \in \mathbb{Z}[X_1, \ldots, X_n]$ *of multivariate polynomials and* $\varepsilon > 0$ *a bound for the level of approximation. Compute for every solution* $\zeta \in V(f_1, \ldots, f_n)$ *a rational approximation of* ζ *at level* ε *i.e. a point*

$$a := (\frac{a_1}{q}, \ldots, \frac{a_n}{q}) \in \mathbb{Q}[i]^n$$

such that $||a - \zeta|| < \varepsilon$ *holds (here* $||\cdot||$ *denotes the norm associated with the usual hermitian product in* $\mathbb{C}^n$*).*

The question becomes now: How to codify the approximations? Basically, we can distinguish four known different encodings: Bit representation, Continued Fractions, Floating point and program encoding (Straight–line programs).

Let us introduce the following notions and notations: A regular sequence $f_1, \ldots, f_n \in \mathbb{Z}[X_1, \ldots, X_n]$ is said to be smooth if for any $1 \leq i \leq n$ the Jacobian $J(f_1, \ldots, f_i)$ is not a zero-divisor modulo the ideal $(f_1, \ldots, f_i)$. Observe that for a smooth regular sequence the ideals $(f_1, \ldots, f_i)$ are always radical.

Theorem 1 *There exists a universal constant* C *with the following property: given a smooth regular sequence of polynomials* $f_1, \ldots, f_n \in \mathbb{Z}[X_1, \ldots, X_n]$ *of degree at most* *d and of logarithmic height at most h. Furthermore let be given a point $a \in \mathbb{Q}[i]^n$ and a real number $0 < \varepsilon \leq 1$. Suppose that the variety $V := V(f_1, \ldots, f_n)$ verifies the following conditions : $V \cap \mathbb{Q}[i]^n = \emptyset$ and there exists a point $\alpha \in V$ such that $||a - \alpha|| < \varepsilon \leq 1$ holds. Let D denote the degree and H the (affine) logarithmic height of V. Then we have for any denominator q of a the following inequality :*

$$\frac{-\log_2 \varepsilon}{(ndD)^C} - (h + H) \leq \log_2 q.$$

We observe that for $n = 1$ Theorem 1 represents just a restatement of Liouville's classical result with somewhat coarser bounds (compare e.g. [27, Kapitel I, Satz 1]).

The height estimates in [2], [3], [9], [21] and [20], combined with the methods described in [11] produce Liouville bounds that relate the syntactical description of V as given by the input, the approximation ε and the height of the denominator of a. In fact, these estimates yield the inequality :

$$-\log_2 \varepsilon \leq d^{O(n)} h \log_2 q \tag{1}$$

On the other hand: degree and height of the solution set represent reasonable lower time bounds for the numerical analysis approach to polynomial equation solving by rational point approximation. Exponential degrees and exponential height produce exponential lower time bounds for numerical methods of polynomial equation solving (cf. [23]).

A lower bound like inequality (1) above, i.e. a lower bound for $\log_2 q$, represents a lower bound for the output length and therefore a lower time bound for numerical methods of polynomial system solving. The results of [1], [2], [3], [5], [6], [9], [20], [21], [24], [25], [26] imply that an approximation level of $\varepsilon := 2^{-d^{O(n)}}$ is sufficient in order to characterize (and to distinguish adequately) the solutions of the variety. On the other hand the examples of [22, Chapitre 4, Proposition 12] and [18], [19, Exemple 3] show that even in case of semantically and syntactically "easy" systems such an approximation level may be necessary. Therefore it becomes reasonable to fix an approximation level for numerical solving of $\varepsilon := 2^{-d^n}$, where d^n represents the Bézout number of the input system. Under this assumption, inequality (1) becomes meaningless and more precise lower bounds are required.

However an exponential lower time bound for numerical solving is implicitly contained in the following corollary to Theorem 1:

Corollary 2 *Given a smooth regular sequence $f_1, \ldots, f_n \in \mathbb{Z}[X_1, \ldots, X_n]$, and points $\alpha \in V$, $a \in \mathbb{Q}[i]^n$ verifying the conditions of Theorem 1 above, let $\varepsilon > 0$ be a level of approximation such that $-\log_2 \varepsilon = d^n$ (where d^n represents the Bézout* *number of the system). Then we have*

$$\frac{d^n}{(ndD)^C} - (h+H) \leq \log_2 q,$$

where C is a suitable universal constant as in Theorem 1. In particular, for systems of 'small" degree and height, the output length for numerical solving methods is necessarily exponential in the number of variables.

As a consequence of the previous lower bounds we have a trade-off between the number of digits required to encode a rational approximation and the level of approximation. Essentially, this trade-off says that the number of digits grows faster than

$$\Omega(\log_2 \varepsilon^{-1})$$

Encodings with prescribed number of digits (such as floating point encoding) is thus not suitable for this purpose. Moreover, when applying quadratic convergence numerical methods the work spent in writing the bit encoding of the rational approximation compensates the high speed of the method. In other words, the total complexity of the method becomes equivalent to methods of linear convergence. Thus, nothing seems to be gained.

To explain this paradoxical claim, we introduce an alternative encoding for the approximations which originates in the works of S. Smale and M. Shub [30], [31], [32], [28], [29] etc.

The input is a system $f_1, \ldots, f_n \in \mathbb{Z}[X_1, \ldots, X_n]$ of multivariate polynomials satisfying the hypothesis of reduced regular sequence. The degrees and heights of the input polynomials are bounded by $\deg(f_i) \leq d$, $ht(f_i) \leq h$. Let $F := (f_1, \ldots, f_n)$ denote the vector formed by the input polynomials,

$$DF(z) = \left(\frac{\partial f_i}{\partial f_j}(z)\right)_{1 \leq i,j \leq n}$$

The Newton operator $N_F(z)$ is then given as

$$N_F(z) := \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix} - (DF(z))^{-1} \begin{pmatrix} f_1(z) \\ \vdots \\ f_n(z) \end{pmatrix}$$

A point $z \in \mathbb{C}^n$ is then called an *approximate zero* of the system F if there exists $\zeta \in V(f_1, \ldots, f_n)$ such that

$$||N_F^k(z) - \zeta|| \leq \frac{1}{2^{2^k-1}} ||z - \zeta|| \text{ holds.}$$

The output is then represented by the following information: for every solution $\zeta \in V(f_1, \ldots, f_n)$ an approximate zero $z \in \mathbb{Q}\,[i]^n$, and a program encoding of the k Newton iterations, where $k = \log_2 \log_2(||z - \zeta|| \varepsilon^{-1})$.

Following this approach, the authors analyze the situation, giving *Condition numbers* to check for the existence and validity of *approximate zeroes.* Let $F := (f_1, \ldots, f_n) \in (\mathbb{Z}[X_1, \ldots, X_n])^n$, $D^k F(z) \in L_k(\mathbb{C}^n, \mathbb{C}^n)$, $z \in \mathbb{C}^n$ and $DF(z)$ regular. Then we define:

$$\gamma(F, z) = \sup_{k \geq 2} \left\| \frac{(DF(z))^{-1} D^{(k)} F(z)}{k!} \right\|^{\frac{1}{k-1}}$$

$$\beta(F, z) = \|(DF(z))^{-1} F(z)\|$$

$$\alpha(F, z) = \gamma(F, z)\beta(F, z)$$

Using this notations, the authors give the following:

Theorem 3 (Gamma-Theorem) *For* $z, \zeta \in \mathbb{C}^n$, $f_i(\zeta) = 0$, $1 \leq i \leq n$, *if*

$$||z - \zeta|| < \frac{3 - \sqrt{7}}{2} \gamma^{-1}(F, \zeta)$$

then z is an approximate zero *of the system F and the k-fold Newton operator $N_F^k(z)$ converges quadratically to ζ.*

Theorem 4 (Alpha-Theorem) *With the same notations as before, there exists a universal constant α_0 such that $\alpha(F, z) < \alpha_0$ implies that z is an* approximate zero *of the system F.*

Let us remark that both results have a local character. As a consequence of the Robustness Theorem (cf. [4]) that says that the conditions of the Alpha-Theorem imply those of the Gamma-Theorem. Unfortunately, this theory also involves information on the separation bound, i.e. the minimum root distance of the system F:

Theorem 5 (Separation Bound) *Let $F : \mathbb{C}^n \to \mathbb{C}^n$ be a system of polynomial equations in $\mathbb{C}[X_1, \ldots, X_n]$. Let*

$$sep(F) = \min\{\|\zeta - \zeta'\| : F(\zeta) = F(\zeta') = 0, \zeta \neq \zeta'\}$$

denote the separation bound of F. Let

$$\gamma(F) := \max_{\zeta \in V(f_1, \ldots, f_n)} \gamma(F, \zeta)$$

Then it holds that

$$sep(F) \geq \frac{5 - \sqrt{17}}{4\gamma(F)}.$$

Even when using this new encoding (approximate zero plus Newton iteraton program) we face an exponential lower time bound as a consequence of the following example:

Example 1 (Based on [22]) *Consider the following multivariate polynomial equation system:*

$$f_1 := X_1 - a, f_2 := X_2 - X_1^d, \dots, f_n := X_n - X_{n-1}^d, f_{n+1} := X_{n+1}^d - (2X_nX_{n+1} - 1)^2$$

In [22] it is shown that the polynomial $P = X^d - (2aX - 1)^2$ *has two roots at a distance strictly smaller than* $2a^{-(d+2)/2}$ *from* $1/a$*. Therefore the above equation system has two roots at a distance stricly less then*

$$a^{-d^{n-1}(d+2)/2}$$

from the point $(a, a^d, \dots, a^{d^{n-1}}, a^{-d^{n-1}})$*. This system verifies that any* approximate zero *satisfying the Gamma-Theorem (or the Alpha-Theorem) has necessarily a representation of exponential bit length.*

Our intention here is to explain the above examples from a Diophantine Geometry point of view, using the notions developed during the last years of research work on the subject.

The complexity of computing approximations to the solutions of a system is characterized in terms of affine geometric degree and height of diophantine varieties. In [17] the more classical notion of height based on the Chow form or elimination polynomial of a variety (cf. e.g. [24] or [33]) and the alternative height based on a geometric solution of the variety (cf. [12], [13], [21], [10], [15], [23], [14], [11] for more details) are shown to be equivalent up to polynomial terms. For the representation of the appearing structures we rely on well parallelizable straight–line programs, cf. e.g. [11].

Theorem 6 (Castro, 1997, [7]) *Let* $F := (f_1, \dots, f_n) \in \mathbb{Z}[X_1, \dots, X_n]$ *be a reduced regular sequence given by a well parallelizable straight–line program of length* L*,* $D := \deg V(f_1, \dots, f_n)$*,* $h := \max ht(f_i)$*,* $H := ht V(f_1, \dots, f_n)$ *and the degree of the* f_i *bounded by* d*. Then there exists a positive universal constant* c *such that*

$$\frac{1}{2^{\Omega(HD+\log_2 n)}} \leq \gamma(F) \leq 2^{(ndhLDH)^c}.$$

As a consequence, this says that in the diophantine context, there always exist *approximate zeroes* whose bit representation length is polynomial in the affine geometric degree and height of the variety.

We now give a theorem resuming the complexity of computing approximate zeroes efficiently, using a symbolic solution method which reduces the multivariate polynomial equation system to an equivalent univariate parametrized representation, which in turn can be then treated numerically.

Theorem 7 *Let* $F := (f_1, \dots, f_n) \in \mathbb{Z}[X_1, \dots, X_n]$ *be a reduced regular sequence given by a well parallelizable straight–line program of length* L*,* $D := \deg V(f_1, \dots, f_n)$*,*

$h := \max ht(f_i)$, $H := htV(f_1,\ldots,f_n)$ and the degree of the f_i bounded by d. Let $\delta := \max_{1\leq i\leq n-1} \deg V(f_1,\ldots,f_n)$. Then there exists an algorithm computing approximate zeroes *for all the solutions of the system F in time*

$$(ndhL\delta\eta)^{O(1)} + (HD)^{O(1)}.$$

This theorem follows from composing the work of [16], [14], [11], applying [8].

From the above Theorem 7 and the encoding by an approximate zero plus a Newton iteration program as introduced before we derive the following corollary on the complexity of Diophantine Approximation:

Corollary 8 *There exists an algorithm for the following task: Given $F := (f_1,\ldots,f_n) \in \mathbb{Z}[X_1,\ldots,X_n]$ a reduced regular sequence by a well parallelizable straight–line program of length L and $\varepsilon > 0$, compute for every $\zeta \in V(f_1,\ldots,f_n)$ an approximation $a \in \mathbb{Q}\,[i]^n$ such that $||\zeta - z|| < \varepsilon$. If $\max\deg(f_i) \leq d$, $h := \max ht(f_i)$, $H := htV(f_1,\ldots,f_n)$ and $\delta := \overset{\max}{1 \leq i \leq n} \deg V(f_1,\ldots,f_i)$, then the computation time for this algorithm is of order*

$$(nhdL\delta\eta)^{O(1)} + (HD)^{O(1)} + n^{O(1)} L \log_2 \log_2 \varepsilon^{-1}.$$

References

[1] E. Becker, J. P. Cardinal, M. F. Roy, Z. Szafraniec: *Multivariate Bezoutians, Kronecker symbol and Eisenbud-Levine formula. Algorithms in Algebraic Geometry and Applications. Proceedings of MEGA'94.* Progress in Mathematics **142**, Birkhäuser (1996) 79–104.

[2] C. Berenstein, A. Yger: *Effective Bézout identities in $\mathbb{Q}\,[X_1,\ldots,X_n]$.* Acta. Math. **166** (1991) 69–120.

[3] C. Berenstein, A. Yger: *Une Formule de Jacobi et ses conséquences.* Ann. Sci. E.N.S. **24 (4)** (1991) 363–377.

[4] L. Blum, F. Cucker, M. Shub, S. Smale: *Real Complexity and Computation.* Manuscript, to appear in Springer Verlag (1997).

[5] J.-B. Bost, H. Gillet, C. Soulé: *Un analogue arithmétique du théorème de Bézout.* C. R. Acad. Sci. Paris **312** (1991) 845–848.

[6] J.-B. Bost, H. Gillet, C. Soulé: *Heights of projective varieties and positive Green forms.* Manuscript I.H.E.S. (1993).

[7] D. Castro: *Sobre la complejidad de la aproximación diofántica y los fundamentos del análisis numérico.* Master's thesis, Universidad de Cantabria, Santander, Spain (1997).

[8] J. Dixon: *Exact solution of linear equations using p-adic expansions.* Numer. Math. **40** (1982) 137–141.

[9] M. Elkadi: *Bornes pour le degré et les hauteurs dans le problème de division.* Michigan Math. J. **40** (1993) 609–618.

[10] N. Fitchas, M. Giusti, F. Smietanski: *Sur la complexité du théorème des zéros.* In J. Guddat et al, ed., *Approximation and Optimization in the Caribbean II, Proceedings 2nd Int. Conf. on Non-Linear Optimization and Approximation*, Approximation and Optimization **8**, Peter Lange Verlag, Frankfurt am Main (1995) 247–329.

[11] M. Giusti, K. Hägele, J. Heintz, J. E. Morais, J. L. Montaña, L. M. Pardo: *Lower Bounds for Diophantine Approximation. Proceedings of MEGA'96*, Journal of Pure and Applied Algebra **117,118** (1997) 277–317.

[12] M. Giusti, J. Heintz: *Algorithmes - disons rapides - pour la décomposition d' une variété algébrique en composantes irréductibles et équidimensionelles.* In T. Mora and C. Traverso, eds., *Proceedings of MEGA'90*, Progress in Mathematics **94**, Birkhäuser (1991) 169–194.

[13] M. Giusti, J. Heintz: *La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial.* In D. Eisenbud and L. Robbiano, eds., *Computational Algebraic Geometry and Commutative Algebra*, Symposia Matematica vol. XXXIV, Cambridge University Press (1993) 216–256.

[14] M. Giusti, J. Heintz, J. E. Morais, J. Morgenstern, L. M. Pardo: *Straight–Line Programs In Geometric Elimination Theory.* To appear in J. of Pure and App. Algebra (1997) 1–46.

[15] M. Giusti, J. Heintz, J. E. Morais, L. M. Pardo: *When Polynomial Equation Systems can be solved fast ?* In G. Cohen, H. Giusti, and T. Mora, eds., *Applied Algebra, Algebraic Algorithms and Error Correcting Codes, Proceedings AAECC-11*, LNCS **948**, Springer (1995) 205–231.

[16] M. Giusti, J. Heintz, J. E. Morais, L. M. Pardo: *Le rôle des structures de données dans les problèmes d'élimination.* To appear in C. R. Acad. Sci. Paris (1997) 1–7.

[17] K. Hägele, J. E. Morais, L. M. Pardo, M. Sombra: *On the intrinsic complexity of the arithmetic Nullstellensatz.* In *Proceedings of TERA'97, Córdoba, Argentina.* September 1997.

[18] J. Heintz, T. Krick, A. Slissenko, P. Solernó: *Searching for a shortest path surrounding semi–algebraic obstacles in the plane (in russian).* Teorija složnosti vyčislenij **192 (5)** (1991) 163–173.

[19] J. Heintz, T. Krick, A. Slissenko, P. Solernó: *Une borne inferieure pour la construction de chemins polygonaux dans* $\mathbb{R}^n$. C. R. Acad. Sci. Paris **309 (I)** (1992) 825–830.

[20] T. Krick, L. M. Pardo: *Une approche informatique pour l' approximation diophantienne.* C. R. Acad. Sci. Paris **318 (1)** (1994) 407–412.

[21] T. Krick, L. M. Pardo: *A Computational Method for Diophantine Approximation.* In L. González-Vega and T. Recio, eds., *Algorithms in Algebraic Geometry and Applications. Proceedings of MEGA'94*, Progress in Mathematics **143**, Birkhäuser Verlag (1996) 193–254.

[22] M. Mignotte: *Mathématiques pour le Calcul Formel.* Presses Univ. de France (1989).

[23] L. M. Pardo: *How lower and upper complexity bounds meet in elimination theory.* In G. Cohen, H. Giusti, and T. Mora, eds., *Applied Algebra, Algebraic Algorithms and Error Correcting Codes. Proceedings of AAECC-11*, LNCS **948**, Springer (1995).

[24] P. Philippon: *Sur des hauteurs alternatives I.* Math. Ann. **289** (1991) 255–283.

[25] P. Philippon: *Sur des hauteurs alternatives II.* Ann. Inst. Fourier, Grenoble **44 (2)** (1994) 1043–1065.

[26] P. Philippon: *Sur des hauteurs alternatives III.* J. Math. Pures Appl. **74** (1995) 345–365.

[27] T. Schneider: *Einführung in die transzendenten Zahlen.* Springer (1957).

[28] M. Shub: *Some Remarks on Bezout's Theorem and Complexity Theory.* Technical Report RC 15993 (71044), IBM T.J. Watson Research Center (1990).

[29] M. Shub, S. Smale: *Complexity of Bézout's Theorem I: Geometric Aspects.* J. of the AMS **6 (2)** (1993) 459–501.

[30] S. Smale: *On the Efficiency of Algorithms of Analysis.* Bull. of the AMS **13 (2)** (1985) 87–121.

[31] S. Smale: *Algorithms for Solving Equations. Proceedings of the International Congress of Mathematicians*, Berkeley, California, USA (1986) 172–195.

[32] S. Smale: *On the Topology of Algorithms I.* J. of Complexity **3** (1987) 81–89.

[33] B. L. Van der Waerden: *Moderne Algebra I.* Berlin (1930).

Residues of sparse polynomials

Alicia Dickenstein

This is a summary of my talk at TERA 97. It is a report on recent progress in the study of residues associated to sparse systems of polynomials done in collaboration with Eduardo Cattani, David Cox and Bernd Sturmfels.

Given a finite set $A \subset \mathbb{Z}^n$ (with convex hull Δ), a *sparse (Laurent) polynomial* f with support A (and Newton polytope Δ) is a polynomial of the form:

$$f(t) = \sum_{m \in \Delta \cap \mathbb{Z}^n} u_{jm}\, t^m \in \mathbb{C}[t_1^{\pm 1}, \ldots, t_n^{\pm 1}].$$

Fix polytopes $\Delta_1, \ldots, \Delta_n$ with integer vertices in $\mathbb{R}^n$. If $f_1, \ldots, f_n$ are generic sparse polynomials with respective Newton polytopes $\Delta_1, \ldots, \Delta_n$,

$$V = \{t \in (\mathbb{C}^*)^n : f_1(t) = \cdots = f_n(t) = 0\}$$

is finite and, by Bernstein's Theorem, the number of elements in this zero set is the mixed volume $MV(\Delta_1, \ldots, \Delta_n)$ of the polytopes.
In the case of simple zeroes, we may consider the residue operator:

$$\mathrm{Res} : \mathbb{C}[t_1^{\pm 1}, \ldots, t_n^{\pm 1}]/\langle f_1, \ldots, f_n \rangle \longrightarrow \mathbb{C}$$

$$\mathrm{Res}(g) = \sum_{\xi \in V} \frac{g(\xi)}{J_f(\xi)}$$

which has the following important properties:

- Res is a dualizing linear form, i.e. $g \in I_f \leftrightarrow \mathrm{Res}(g.h) = 0\ \forall \bar{h}$.
- It has functorial properties.
- It admits an extension with the same properties when the zero set V is finite but the zeroes are not necessarily simple.
- It depends polynomially on parameters u for flat families $f_1^u, \ldots, f_n^u$.
- Consider the standard trace

  $$\mathrm{tr} : \mathbb{C}[t_1^{\pm 1}, \ldots, t_n^{\pm n}]/\langle f_1, \ldots, f_n \rangle \longrightarrow \mathbb{C}$$

  $$\mathrm{tr}(g) = \sum_{\xi \in V} g(\xi).$$

  Then, $\mathrm{tr}(g) = \mathrm{Res}(g \cdot J_f)$ (i.e., traces are instances of residues).

- There are singly exponential (in n) time general complexity bounds for the computation of residues.
- There exist "theoretical" applications of residues to the resolution of systems of polynomial equations, membership in ideals, division formulas, computation of real and complex roots, computation of degrees of real proper maps, etc.
- There exist "practical" applications of residues to the resolution of systems of polynomial equations
- For generic sparse polynomials with fixed Newton polytopes, for any $g \in \mathbb{C}[t_1^{\pm 1}, \ldots, t_n^{\pm n}]$, $\mathrm{Res}(g)$ is a rational function of the coefficients of the data.

We studied this rational function with a double perspective: on one side, as a tool for the study of certain multihomogeneous ideals of the homogeneous coordinate ring of a toric variety, and on the other side, as a tool to develop computational techniques which may ultimately enter into the design of algorithms for solving polynomial equations.

The main link between this two aspects is the description of the residue $\mathrm{Res}(g)$ as a coefficient of a normal form of a suitable multi-homogeneization G or g, modulo the ideal I_F generated by (standard "Taylor") polynomials $F_0, \ldots, F_n$, where F_0 depends only on the support of g and $F_1, \ldots, F_n$ are the corresponding multi-homogeneizations of the input polynomials $f_1, \ldots, f_n$.

The effective computation of residues allows to solve, for instance, the following elimination problem: Denote N the total number of indetermined coefficients u in $f_1, \ldots, f_n$. Let

$$V := \{(u,t) \in \mathbb{C}^N \times (\mathbb{C}^*)^n : f_j(u,t) = 0\}.$$

and $H_i(u,t) := t_i$, $(i = 1, \ldots, n)$. Consider the linear (non flat) projection

$$\pi : V \to \mathbb{C}^N \times \mathbb{C},$$

$$\pi(u,t) = (u, H_i).$$

Find a polynomial

$$P(u,Y) = Q(u)Y^{\deg(P)} + \sum_{j<\deg(P)} Q_j(u)Y^j$$

describing the image

$$P(u,t_i) = 0 \text{ for all } (u,t) \in V.$$

The rational functions $Q_j(u)/Q(u)$ are the elementary symmetric functions of the i-th coordinates of the common zeros of $f_1(u,t) = \ldots = f_n(u,t) = 0$, which can be deduced from the computation of traces of monomials.

We described, for each monomial t^m, the denominator of the residue $\mathrm{Res}(t^m)$ as a product of sparse resultants associated to the facets of the Minkowski sum $\Delta := \Delta_1 + \ldots + \Delta_n$ of the input polytopes $\Delta_1, \ldots, \Delta_n$, raised to powers prescribed by the "lattice distances" from m to those facets of Δ which are "visible" from m. We also showed that the denominator appears as a "numerator" of the normal form of explicit polynomials D_m which resemble a Bezoutian, modulo a suitable multihomogeneous ideal I_F. Moreover, we gave bounds for the polynomial dependence of $\mathrm{Res}(t^m)$ on coefficients corresponding to interior monomials.

As a by-product of our methods, we express sparse resultants as determinants of Koszul-type complexes which involve the Jacobian, (or as g.c.d. of maximal minors of explicit matrices). In some special cases, we obtain Sylvester-type formulas. In fact, the numerator of the residue can also be described in this way.

Bibliography

[1] C. Berenstein, R. Gay, A. Vidras, A. Yger: *Residue Currents and Bezout Identities.* Progress in Mathematics **114**, Birkhäuser, Basel Boston Berlin (1993).

[2] E. Cattani, D. Cox, A. Dickenstein: *Residues in Toric Varieties.* Compositio Mathematica **108** (1997) 35–76.

[3] E. Cattani, A. Dickenstein: *A Global View of Residues in the Torus.* Journal of Pure and Applied Algebra **117 & 118** (1997) 119–144.

[4] E. Cattani, A. Dickenstein, B. Sturmfels: *Computing Multidimensional Residues. MEGA 94: Effective Methods in Algebraic Geometry*, Progress in Mathematics **143**, Birkhäuser (1996) 135–164.

[5] E. Cattani, A. Dickenstein, B. Sturmfels: *Residues and Resultants.* Preprint alg-geom 9702001 (1997).

[6] J.-P. Cardinal, B. Mourrain: *Algebraic approach of residues and applications.* Lectures in Applied Math. **32**, Amer. Math. Soc. Press (1996) 189–210.

[7] D. Cox: *Toric residues.* Arkiv för Matematik **34** (1996) 73–96.

[8] I. Gel'fand, M. Kapranov, A. Zelevinsky: *Discriminants, Resultants and Multidimensional Determinants.* Birkhäuser, Basel Boston Berlin (1994).

[9] O. A. Gelfond, A. G. Khovanskii: *Newtonian polyhedrons and Grothendieck residues.* Doklady Mathematics **54** (1996) 700–702.

[10] J.-P. Jouanolou: *Résultant et résidu de Grothendieck.* Preprint IRMA, Strasbourg (1992).

[11] B. Mourrain, V.Y. Pan: *Solving Special Polynomial Systems by Using Structured Matrices and Algebraic Residues.* In: F.Cucker–M.Shub eds., *Foundations of Computational Mathematics*, Springer-Verlag (1997).

[12] A. Tsikh: *Multidimensional Residues and Their Applications.* AMS, Providence (1992).

Energy Landscape and Learning on a Neural Network for the Sum Problem

Leonardo Franco [1], *Sergio A. Cannas* [2]

Abstract. We describe a feed-forward Neural Network for the addition of two natural numbers. We show that the corresponding error function has local minima, flat minima and other properties of the same kind. A couple of optimization strategies are implemented for the learning problem.

Using the local minima structure of the error, we analyze how the selection of examples in the training set influences the learning perfomance.

We obtain an upper bound for the minimal number of examples needed for the addition of two N–bit integers.

Introduction

In 1996 Sprinkhuizen-Kuyper & Boers showed in [1] that the XOR function does not have local minima. In this paper we study more complicated functions and check whether they have local minima and why. This information is then used in order to compare through the learning process different algorithms for addition.

We study the energy function of a specific network that performs the addition of two numbers of N-bit length (observe that the case $N = 1$ is represented by the XOR function).

The Structure of a specific network for the addition of two N–bit numbers

In [2] was introduced a network for the addition of two N–bit numbers. This network is a perceptron with a hidden layer that solves, with determinated weights, the sum problem. The input layer has $2N$ units, the hidden layer has N, and the output layer has $N+1$ neurons. We consider the symmetric case, where synapses connecting one neuron to input bits with the same significant values are equal.

---

[1] Fellow of the Secretaría de Ciencia y Tecnología de la Universidad Nacional de Córdoba. Argentina

[2] Member of the National Research Council (CONICET). Argentina.

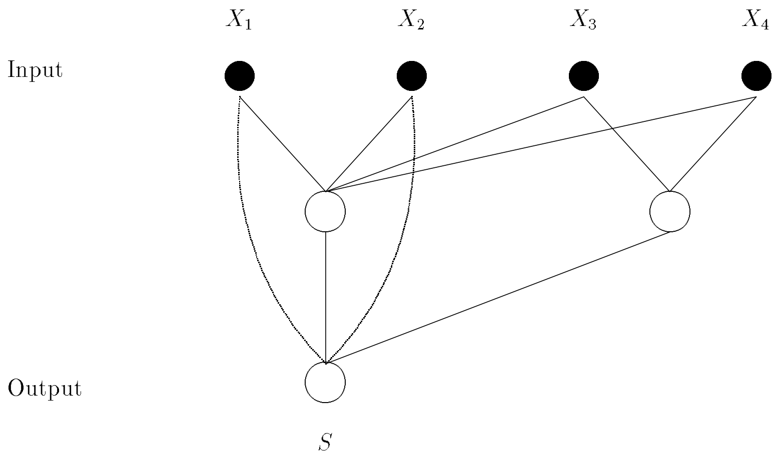


Figure 1: Network Structure for the second output bit for the sum problem of two natural numbers.

The energy of the network, E, is defined as the difference between the output computed by the network and the desired output (target), computed over the training set as follows:

$$E\{w_j, K_i\} = \sum_{i=1}^{K} (S_i\{w_j, K_i\} - O_i\{K_i\})^2, \tag{1}$$

Here K is the number of examples in the training set, K_i is a particular example, and $\{w_j\}$ are the synaptic weights.

The generalization error is defined similarly taking into account all possible examples.

Existence of Local minima

We study the addition of two 2–bit numbers in the symmetric case, restricting ourselves to the second output bit (because the study of the first output bit was exhaustively done in [1]). The structure for computing this second output bit is shown in Figure 1.

The output neuron, S, computes the following function:

$$S = \sigma\Big(w_1{\cdot}(x_1{+}x_2){+}w_2{\cdot}\sigma\Big(w_3{\cdot}(x_1{+}x_2){+}w_4{\cdot}(x_3{+}x_4){-}h_1\Big){+}w_5{\cdot}\sigma\Big(w_6{\cdot}(x_3{+}x_4){-}h_2\Big){-}h_0\Big)$$

Here w_i are the synaptic weights, h_i are the thresholds, and x_i, $(i = 1, .., 4)$ are the input bits, where x_1, x_2 are the bits with significative value 2^1 corresponding to the two input numbers and x_3 and x_4 are the bits with value 2^0. The activation function

σ computed by the neurons is a sigmoid transfer function of the form:

$$\sigma(x) = \frac{1}{1+e^{-x}}.$$

The input neurons (x_i, $i = 1, \ldots, 4$) are fed by binary values $\{0, 1\}$ depending on the example to be evaluated. Since we wish to have binary values as outputs and the output of the network is continuous, we set the targets to fixed values in order to avoid an excessive growth of the synapses. This is done in the following way:

- δ in the case corresponding to an output 0 (Neuron OFF).
- $1 - \delta$ in the case corresponding to an output 1 (Neuron ON).

We select δ to be 0.1 but any small positive number can be used.

For the case of adding two numbers of two-bit length ($N = 2$), the number of different training patterns (examples) is 16, but just 9 of them are independent due to the symmetry imposed to the synapses.
To find the possible local minima of Equation (1) is necessary to solve the following nine differential equations:

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial h_i} = 0$$

In this way we obtain a system of nine differential equations in nine variables which is too complicated to be solved analytically. Therefore we are going to use optimization methods in order to guess local minima fo the energy function. It turns out that they are indeed local minima. For this purpose we compute the eigenvalues of the Hessian matrix corresponding to Equation (1). If a given point is a local minimum all these eigenvalues define a positive definite Hessian. The entries of the Hessian matrix corresponding to Equation (1) have

$$H_{ij} = \frac{\partial^2 E}{\partial^2 w_{ij}}$$

Although numerical difficulties occur when we try to verify the positiveness of the eigenvalues, we have strong evidence that many local minima exist for our problem.

Energy Landscape and the behaviour of the examples

Our optimization algorithms produce characteristic elements of the landscape as Flat minima, Plateaus, High barriers, etc. All these elements turn the optimization procedure very complicated, slowing down the rate of learning and the probability of success of the search for global minima. In Figure 2 we show the energy distribution along the direction between two minima, one being local and the other being global.

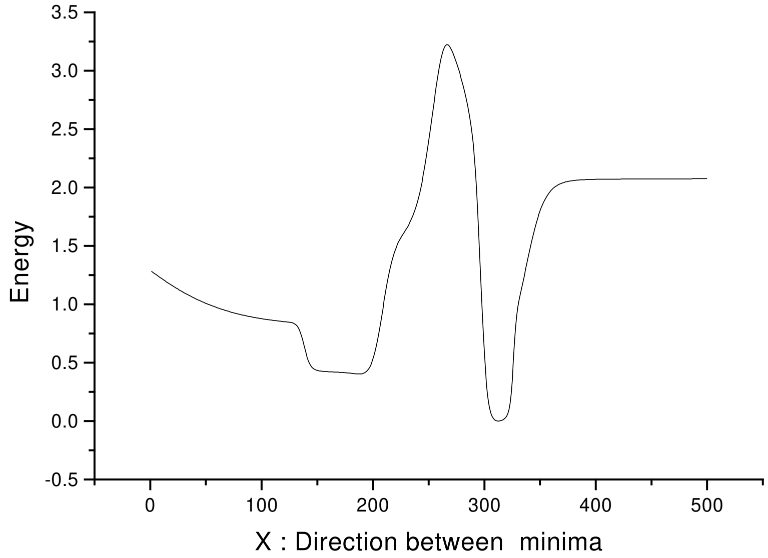


Figure 2: Energy landscape along the direction between a global minimum and a local one.

The energy landscape function corresponding to Equation (1) is the sum of many particular energies of the individual examples. The analysis of these individual energies yields the following characterization:

with respect to the global minima we can clearly distinguish two types of examples (according to its contribution to the energy function): the examples with output 0.1 (OFF) and those with output 0.9 (ON). There exists also examples with higher energy contributions at global minima than others (see Figure 3).

Analyzing local minima we find two cases:
In the first case all the examples have the same residual energy, as shown in Figure 4. In the second case we can clearly distinguish between examples that have been learnt and others that have not.

Our experimental data indicate that examples involving bit-carry are the most difficult ones to be learnt.

Improving Generalization: Selection of Examples

It was shown by Kinzel & Ruján [2] that selection of examples influences generalization. With the information obtained from the study of the energy landscape we made a selection of examples for the sum problem. For the sake of simplicity we consider a network with binary neurons, instead of the continuos ones, and we study the case $N = 3$. For the case $N = 3$ we analyze the 27 equations for every particular energy of the examples, finding that 10 selected examples are enough to obtain generalization.

Our results allow conclusions for the case of N arbitrary. We find *analytically* an up-

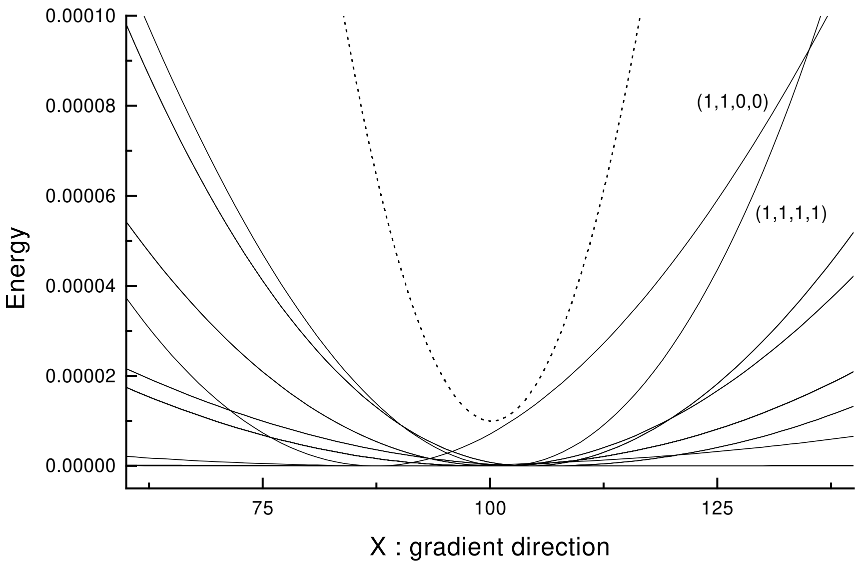


Figure 3: Behaviour of the examples near a global minima. In dotted line the Total energy and in dashed lines the energy corresponding to the examples.

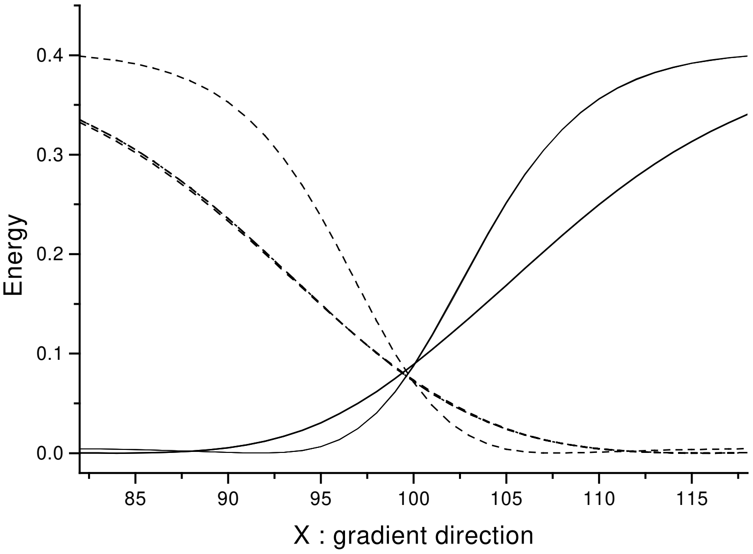


Figure 4: Local minima where all the examples have the same non-zero energy equal to 0.08.

per bound for the minimal number N_m of examples needed for perfect generalization, namely:

$$N_m \leq 2(2N-1).$$

Comparing this number with the total number of synapses N_s of the net for the sum problem, namely

$$N_s = \frac{1}{2}(N^2 + 5N - 2),$$

we find that, for $N \gg 1$, the following asymptotic relation:

$$\frac{N_m}{N_s} \sim \mathcal{O}(\frac{1}{N})$$

.

We also tested the generalization properties through numerical simulations (simulated annealing) for $N = 3, 4, 5$ choosing equidistributed random examples. We found that the *average* minimal number of examples $\langle N_m \rangle$ behaves as:

$$\langle N_m \rangle \sim \mathcal{O}(N^2)$$

.

Conclusions

The energy landscape of the sum problem of two N-bit numbers has for $N > 1$ a very complex structure, with *local minima, flat minima* and *plateaus*. The analysis of this structure leads to the conclusion that some *classes* of examples strongly influence the energy landscape and therefore, they must be included in a selection if we wish to obtain a good generalization.

We find **analytically** an upper bound to the minimal number of examples for optimal generalization, namely $N_m \sim \mathcal{O}(N)$, which compares very well to the average minimal number of randomly chosen examples, namely $\langle N_m \rangle \sim \mathcal{O}(N^2)$.

All these results can be used for the development of *general* criteria for the improvement of the generalization procedure by means of example selection. Work along these lines is in progress.

References

[1] I.G. Sprinkhuizen-Kuyper, E.J.W. Boers: *The Error Surface of the Simplest XOR Network Has Only Global Minima.* Neural Computation **8** (1996) 1301–1320.

[2] S.A. Cannas: *Arithmetic Perceptrons.* Neural Computation **7(1)** (1995) 173–181.

[3] W. Kinzel, P. Ruján: *Improving a Network Generalization Ability by Selecting Examples.* Europhysics Letters **13(5)** (1990) 473–477.

Computing the Dimension of a Projective Variety: the Projective Noether Maple Package

M. Giusti, K. Hägele, G. Lecerf, J. Marchand, B. Salvy

Classical methods to study and solve systems of polynomial equations are based on numerous avatars of Gröbner (or standard) bases algorithms or Riquier-Janet type methods (Ritt-Wu's algorithm). All these methods use implicitly but deeply the dense or sparse representation of multivariate polynomials, which is the computer science counterpart of the expansion of these mathematical objects on the monomial basis of the polynomial algebra. And all these methods can be interpreted as *rewriting* techniques.

Considerable efforts have been made in order to improve both theoretical and practical aspects of these techniques and to produce efficient algorithms and implementations. Restricting to Gröbner bases algorithms and to quote only the most commonly available packages, see the Maple `grobner package` [1], `axiom` [9], `gb` [2], `singular` [6], `macaulay2` [8], `reduce` [7]. Some very recent benchmarks can be found in [3].

The knowledge of a standard basis yields as a simple byproduct the dimension of the algebraic variety defined by such systems. Actually, one can show that focusing on the simpler problem of computing the dimension of a projective algebraic variety will lead to a better worst-case complexity than the whole construction of a standard basis [4].

We consider the *unit cost measure* model, i.e., each arithmetic operation of the ground field is counted as one. Then the complexity is polynomial in the size of the intermediate expressions computed. Its seems that there is no hope to design an algorithm whose complexity is *polynomial* in the size of the input, since the intermediate computations are not. But this observation is only valid if we stay stuck in the dense representation context. A breakthrough was obtained by [5], resulting in the existence of an algorithm with polynomial behaviour, provided one uses a mixed representation for intermediate computations.

Actually the algorithm described in *loc. cit.* computes a little bit more, i.e., a change of coordinates putting the new variables in *Noether position*. Informally speaking, the variables are separated into two subsets of different nature: the independent and dependent ones. Furthermore the polynomials occurring as intermediate computations are coded in the dense representation w.r.t. the dependent variables, while their coefficients are polynomials in the independent ones coded by *arithmetic circuits*, also called *straight-line programs*. This means that these latter polynomials are represented by programs evaluating them at a point (of the ground field) using only additions and multiplications (of the ground field). We will conveniently refer respectively to the *writing data structure* and the *evaluation data structure*.

Mixing these two data structures was successfully used in a series of papers to design a new geometric elimination algorithm (see in the references the joint works by Giusti, Hägele, Heintz, Montaña, Morais, Morgenstern, Pardo 1995–97). An efficient implementation of the complete elimination algorithm will require some time to collect more practical experience with the first experimental prototypes for some of its components. Some basic but important steps were made already, but there is still a lot of work left (on the use of evaluation data structures, see the works by Aldaz Zaragüeta, Castaño, Hägele, Llovet, Martínez, Matera within the Tera project `http://hilbert.matesco.unican.es/tera`).

We present here another step, a Maple program called the *Projective Noether Package* implementing the algorithm from [5]. (The package and its documentation are available at `http://medicis.polytechnique.fr/tera/soft.html`). It turns out that a not so well-known functionality of Maple is the systematic use of the evaluation data structure, which is described in the first part of this article. Consequently we can compare the more traditional algorithms already available within Maple with the new ones, experimenting with several possible strategies. The package contains hybrid code mixing the old and new approaches, thus allowing to investigate the potential of the new techniques. Comparative results on benchmarks for time and space of three different families of multivariate polynomial equation systems are given and we point out both advantages and weaknesses of the different approaches. One of the most encouraging results is provided by an example where our Maple implementation computes an upper bound for the dimension more than fifty times faster than the `gb` system.

References

[1] B.W. Char, K.O. Geddes, G.H. Gonnet, B.L. Leong, M.B. Monagan, S.M. Watt: *Maple V Library Reference Manual.* Springer-Verlag (1991).

[2] J.-C. Faugère: *GB Reference Manual.* LITP (1995).

[3] J.-C. Faugère: *GB: State of GB + Tutorial.*

[4] M. Giusti: *Combinatorial dimension theory of algebraic varieties.* Journal of Symbolic Computation **6(2–3)** (1988) 249–265.

[5] M. Giusti, J. Heintz: *La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial.* Computational algebraic geometry and commutative algebra (D. Eisenbud, L. Robbiano, Eds.), Symposia Matematica vol. XXXIV, Cambridge University Press, Cambridge (1993) 216–256.

[6] G.-M. Greuel, G. Pfister, H. Schönemann: *Singular.* Universität Kaiserslautern, Germany (1997).

[7] A.C. Hearn: *Reduce user's manual, Version 3.3.* The RAND Corporation (1987).

[8] M. Stillman, D. Bayer: *Macaulay 2 User Manual* (1996).

[9] S. Watt, P.A. Broadberry, S.S. Dooley, P. Iglio, S.C. Morrison, J.M. Steinbach, R.S. Stutor: *Axiom Library Compiler.* NAG, March (1995).

On the intrinsic complexity of the arithmetic Nullstellensatz

K. Hägele, J.E. Morais, L.M. Pardo, M. Sombra

Research was partially supported by the following European, French and Spanish grants : CNRS 1026 MEDICIS, and PB93–0472–C02–02.

Keywords. Polynomial system solving, elimination, geometric degree, arithmetic networks, straight–line programs, complexity.

In the sequel we discuss several arithmetic aspects of Hilbert's Nullstellensatz whose origin is closely connected with the classical consistency question for systems of multivariate polynomial equation systems. Let us assume that a system $S = \{f_1, \dots, f_s\} \subseteq \mathbb{Z}[X_1, \dots, X_n]$ of multivariate polynomials with integer coefficients is given. *The consistency problem over* $\mathbb{C}$ *on input* S can be stated as deciding whether the following formula is true or not:

$$\exists x_1 \in \mathbb{C}, \dots, \exists x_n \in \mathbb{C}, f_i(x_1, \dots, x_n) = 0, \text{ for } 1 \leq i \leq s. \tag{1}$$

Hilbert's Nullstellensatz states that this formula is false if and only if an *arithmetic Bézout identity* holds, i.e. if and only there exist a non-negative integer $a \in \mathbb{Z} \setminus \{0\}$ and polynomials $g_1, \dots, g_s \in \mathbb{Z}[X_1, \dots, X_n]$ such that the following identity holds:

$$a = g_1 f_1 + \cdots + g_s f_s \tag{2}$$

This arithmetic presentation of the Hilbert Nullstellensatz is more suitable for the study of the *consistency problem over prime fields of positive characteristic.* More precisely, let us consider a prime number $p \in \mathbb{N}$, $\mathbb{F}_p := \mathbb{Z}/p\mathbb{Z}$ the prime field of *positive* characteristic p, and let K_p be an algebraic closure of $\mathbb{F}_p$. For every polynomial $f \in \mathbb{Z}[X_1, \dots, X_n]$ we denote by $\bar{f}^p \in \mathbb{Z}/p\mathbb{Z}[X_1, \dots, X_n]$ the polynomial obtained by taking residues $\pmod p$ in the coefficients of f (the superscript p will be usually omitted when no confusion arrises). The consistency problem over $\mathbb{F}_p$ on input system S can now be stated as deciding whether the following formula is true or not:

$$\exists x_1 \in K_p, \dots, \exists x_n \in K_p, \bar{f}_i^p(x_1, \dots, x_n) = 0, 1 \leq i \leq s \tag{3}$$

These pages are mainly concerned with the estimation of the number of primes p for a given system S such that the formula (3) is false. In order to formalise this problem, we introduce the following function (which depends on S):

$$\pi_S : \mathbb{Z}_{\geq 0} \longrightarrow \mathbb{Z}_{\geq 0}$$

defined by $\pi_S(x) :=$ the number of primes p such that $p \leq x$ and the formula (3) is true. Exact estimates of this function π_S are not known. For the moment, only upper and lower bounds are available. These pages are a new attempt to show some upper bounds for π_S.

Let us observe that if the system S is inconsistent over $\mathbb{C}$, the arithmetic Bézout identity (2) holds. In that case, π_S is a bounded function. The obvious reason is that if (2) holds, then (3) is false for all those prime numbers p such that $a \not\equiv 0 \pmod{p}$. In particular, we have the bound

$$\pi_S(x) \leq \log_2 |\, a \,| \text{ for all } a \in \mathbb{Z}_{\geq 0},$$

where $|\, a \,|$ denotes the usual absolute value of a. Conversely, if (1) is a valid formula, π_S is an unbounded function. Thus, studies around the arithmetic Bézout identity (2) with estimates for the absolute value of a (and consequently of the coefficients of the polynomials $g_1, \ldots, g_s$) are of central relevance for estimating π_S.

The introduction of the function π_S for complexity considerations in elimination theory is due to [17] In this paper the reduction techniques of [13], [12], [10], [14] and the generalized Riemann Hypothesis are used to prove upper and lower bounds for the function π_S. From the bounds thus obtained one deduces easily that the consistency problem for polynomial equation systems lies in a complexity class below PSPACE (a fact which has been shown already using different techniques).

A discussion of the historical achievements around (arithmetic) Bézout identities follows. Let us assume now that d is an upper bound for the degrees of the polynomials in the input system $S = \{f_1, \ldots, f_s\}$ and, that h is an upper bound for the absolute values of the coefficients of the polynomials in S. First estimates for $\log_2 |\, a \,|$ were obtained by the systematic use of upper bounds for for the degrees of the polynomials $g_1, \ldots, g_s$ that may occur in a Bézout identity like (2).

Let us assume by now that the system S is inconsistent and denote by $D(d,n)$ the minimum of such degrees in the Bézout identity (2).

The first remarkable achievement was G. Hermann's upper bound (cf. [16]) that stated

$$D(d,n) \leq d^{2^n}.$$

We have to wait until the late eighties to see meaningful improvements of Hermann's bound. In [4], [6], [18] the bound was improved to:

$$D(d,n) \leq \max\{3, d\}^n$$

(cf. also [8], [24]). As the case $d = 2$ seemed to escape to their techniques, in the nineties the following bound was obtained (cf. [7], [25])

$$D(d,n) \leq 4nd^n$$

which holds also for the case $d = 2$. The lower bound

$$D(d,n) \geq d^n - d^{n-1}$$

is a consequence of the example due to Mora-Lazard-Masser-Philippon:

$$S := \{X_1^d, X_1 - X_2^d, \ldots, X_{n-2} - X_{n-1}^d, 1 - X_{n-1}X_n^{d-1}\}.$$

One may see (2) as a big system of linear equations where the variables are a and the coefficients of the polynomials $g_1, \ldots, g_s$. This system has at most

$$\binom{sD(d,n)+n}{n}$$

variables and equations. Thus the system S is consistent if and only if this big system of linear equations is inconsistent. Moreover, by means of an appropriate use of Kramer's rule, we may view a as a big determinant of a matrix M with

$$\binom{sD(d,n)+n}{n}$$

rows and columns. We may then conclude:

$$\log_2 |\, a \,| \leq D(d,n)^n s \log_2 h.$$

The previous bounds would imply the bound

$$\pi_s(x) \leq \log_2 |\, a \,| \leq d^{n^2} s \log_2 h.$$

Nevertheless, this upper bound is not good enough. The works of [2], [1] and [19], [20] would provide the following more precise upper bound:

$$\pi_s(x) \leq \log_2 |\, a \,| \leq k(n) d^{8n+3} (\log_2 \eta + d \log_2 d) \tag{4}$$

where $k(n)$ is an effective constant depending only on n. This bound seemed to be a final achievement because of the following adapted example (for $a \notin \mathbb{Z}$):

$$X_1 - a, X_2 - X_1^d, \ldots, X_n - X_{n-1}^d, X_n$$

In this example we observe that $\log_2 |\, a \,| \geq d^n$ for some inconsistent system S.

However, all these approaches had some inconvenience. Their estimates are based on the chosen representation of the input polynomials. More precisely, all these estimates depend strongly on the syntactical representation of the polynomials (degree, number of variables, absolute value of the coefficients) and they are not so precise when we are faced with very simple systems as the following one:

$$S_5 := \{X_1 - 1, X_2 - X_1^d, \ldots, X_n - X_{n-1}^d, X_n\}.$$

In this example, the integer $a \in \mathbb{Z} \setminus \{0\}$ in identity (2) may be chosen to be exactly 1. In other words,

$$\pi_s(x) \leq \log_2 |\, a \,| = 0, \ \forall x \in \mathbb{Z}_{\geq 0}.$$

However, the given "syntactical" upper bounds would yield

$$\pi_s(x) \leq \log_2 |\, a \,| \leq k(n) d^{8n} \log_2 n$$

which is too far from the actual value of a.

Intuition says that we may profit the special features of the input system S in order to reach improvements for the upper bounds of $\log_2 |\, a \,|$. In this sense, the works [9], [19] and [20] were essentially seminal. These works essentially said that the polynomials $g_1, \ldots, g_s$ and the integer $a \in \mathbb{Z} \setminus \{0\}$, occurring in a Bézout identity may be chosen with very special features in terms of expressibility. In fact, those works showed that $g_1, \ldots, g_s$ and a can be expressed in terms of the input system in a better form than expected: they can be evaluated by a (certain kind of) program in a size of order similar to the degree ($d^{O(n)}$). This feature yielded bounds of order equivalent to those in (4). Moreover, these works showed that these easy-to-express polynomials can be effectively computed by a Turing machine running in time which is essentially polynomial in the bound $d^n \log_2 h$.

However the elimination procedures underlying [19], [20] were also of syntactical nature (they were strongly inspired by [11]). The bound is then of syntactical nature (depending on the bound d^n) and this can not profit the special features of the input system (as in the case of S_5).

Several intrinsic complexity elimination procedures were developed since then in [13] etc. These elimination procedures have the following properties:

- they can be adapted to the syntactical representation chosen to encode the polynomials in the input system S.
- an intrinsic invariant (the degree of the system S) rises in the complexity estimate.

Theses methods are essential to the forthcoming main theorem.

The adaptation to the syntactical representation chosen to encode the input polynomials can be explained in the following terms: usual encodings of polynomials is that of viewing the polynomial as a list of coefficients. This is very accurate for “dense polynomial systems”, i.e. when all coefficients of all polynomials are non-zero. However, this is not well-adapted when the input polynomials have very few non-zero coefficients (sparse/fewnomial systems). Even more, this representation is badly-adapted when the polynomials are easy-to-evaluate (easy-to-compute in the current literature). For instance, dense encoding of the polynomials of system S_5 would yield an input length of order nd^n, whereas sparse/fewnomial encoding would yield an $O(n \log_2 d)$ input length. Moreover, all polynomials in S_5 can be evaluated using $O(n \log_2 d)$ arithmetic operations. Another example with similar characteristics is the determinant of a matrix whose entries are treated as variables. The determinant $\det(X_{ij})$ has a dense encoding of length $\binom{2n}{n}$ (i.e. exponential in n), whereas the sparse/fewnomial encoding in n and $\det(X_{ij})$ can be evaluated by $O(n^6)$ arithmetic operations.

In order to represent the sequence of arithmetic operations needed to evaluate a polynomial the computer science literature developed the notion of straight-line program. This notion was originated by a question of A. M. Ostrowski in 1954 [22]. Some reference material on the subject may be seen in [23], [15] or in the books [3], [5].

Basically, a straight–line program is a list of instructions (arithmetic operations) performed to evaluate the polynomial on any input. The size of the straight–line program is the number of arithmetic operations indicated. For instance, if f is a polynomial with N non-zero coefficients of degree at most d, the size of the natural straight–line program that evaluates f is $O(N \log_2 d)$. One section below recompiles some basic facts concerning straight–line programs.

Thus, we say that a polynomial is easy-to-compute if it can be evaluated by a straight–line program of “small” size. In the case of multivariate polynomials, a “small size” could be considered as similar to the degree of the polynomial (or even smaller). In this sense, we say that a polynomial is “easy-to-express” or “easy-to-evaluate” (as used above) if it can be evaluated by s straight–line program of small size. The “easy-to-compute” condition would yield special features in terms of obsolete values of the coefficients.

In [13], [12], [10] the complexity of symbolic solving of multivariate polynomial equation systems (such as S above) was shown to be dependent on some purely geometric invariant δ_S called the geometric degree of the system. In our complexity analysis (as stated in Theorem below) we show that this is also true for the computation if the non-zero integer a and the polynomials $g_1, \ldots, g_s$ occurring in a Bézout identity caused by inconsistent systems.

Whereas δ_S is purely geometric, our study for upper bounds for $\log_2 | a |$ also requires an intrinsic parameter M_S (of arithmetic-geometric nature). Roughly speaking M_S measures the features of the system S to be transformed in a “suite secante”. Whereas δ_S simply measures the (geometric) degree of intermediate varieties occurring during the elimination procedure, the invariant M_S estimates the metric properties (the Mahler measure of the elimination polynomials) of the varieties arising during the same elimination procedure.

Theorem 1 (New bounds for the arithmetic Nullstellensatz) *Let* $S = \{f_1, \ldots, f_s\} \subseteq \mathbb{Z}[X_1, \ldots, X_n]$ *be a system of polynomials with integer coefficients. Let us assume that there exists a well-parallelizable division free straight–line program* Γ *that evaluates all polynomials in* S. *Let* L *be an upper bound for the size of* Γ, d *an upper bound for the degrees of the polynomials in* S *and* h *an upper bound for the absolute values of the coefficients of the polynomials in* S. *Let* δ_S *and* M_S *be the degree and the logarithmic Mahler measure of the system* S. *Then, if the equation system associated to* S *is inconsistent, there exist a non-zero integer* a *and polynomials* $g_1, \ldots, g_s \in \mathbb{Z}[X_1, \ldots, X_n]$ *such that the following Bézout identity is satisfied:*

$$a = g_1 f_1 + \cdots + g_s f_s$$

and the following properties hold:

i) $\deg(g_i) \leq (nd\delta_S)^c$

ii) $\log_2 |\, a\, | \leq (nd\delta_S M_S L)^c$

iii) $M(g_i) \leq (nd\delta_S M_S L)^c$, *for* $1 \leq i \leq s$

iv) $\log_2 wt(g_i) \leq (nd\delta_S M_S L)^c$, *for* $1 \leq i \leq s$

where $c > 0$ *is a universal constant independent of* S, n, d *and* L, $M(g_i)$ *is the Mahler measure of the polynomial* g_i *and* $wt(g_i)$ *is the weight (the sum of the absolute values of the coefficients) of the polynomial* g_i.

Corollary 2 *With the same notations as above, if* S *is inconsistent,*

$$\pi_S(x) \leq (nd\delta_S M_S L)^c$$

These bounds are achieved as a consequence of a complexity study of the division problem in the Nullstellensatz. Moreover, the polynomials $g_1, \ldots, g_s$ and the integer a are easy-to-express in terms of the input system. In fact, we also obtain the following theorem (with the same notations as in Theorem 1):

Theorem 2 (Computational Complexity) *There exists a bounded error probabilistic Turing machine* M *that, taking the system* S *(encoded by* Γ*) as input, performs the following task:*

i) decide whether the system S *defines consistent system of equations,* *ii) if the system S is inconsistent, the machine M outputs the integer $a \in \mathbb{Z} \subseteq \{0\}$ and the polynomials $g_1, \ldots, g_s \in \mathbb{Z}[X_1, \ldots, X_n]$.*

- *The running time of the machine M is polynomial in $ndL\delta_S$.*
- *The number of arithmetic operations performed by M is also polynomial in $ndL\delta_S$.*
- *The output is given by a division free straight–line program of size polynomial in $ndL\delta_S$.*

As final comments on these main statements we make the following remarks: A bounded error probability probabilistic Turing machine M performing task 1 was already obtained as a consequence of the series of papers In [13], [12], [10], [21].

The new outcome is the performance of task 2 (the division problem). Condition (ii) in Theorem 11 can be understood as the polynomials and integers appearing in the Nullstellensatz are easy-to-express in terms of the input system and some intrinsic (purely geometric but not arithmetic) condition number δ_S. The bounds in Theorem 9 and its corollary will also depend on the arithmetic features of the input system (the Mahler measure, for instance, M_S). Finally the distance between the complexity estimates (depending just on the geometry) and the arithmetic estimates (depending also on the arithmetic properties of the system) will open the question about the distance between both approaches. In order to show that these statements are of sufficiently general type we give the proofs for subrings R of numberfields K.

References

[1] C. Berenstein, A. Yger: *Effective Bézout identities in $\mathbb{Q}\,[X_1, \ldots, X_n]$.* Acta. Math. **166** (1991) 69–120.

[2] C. Berenstein, A. Yger: *Une Formule de Jacobi et ses conséquences.* Ann. Sci. E.N.S. **24 (4)** (1991) 363–377.

[3] A. Borodin, J. Munro: *The Computational Complexity of Algebraic and Numeric Problems.* Elsevier (1972).

[4] D. W. Brownawell: *Bounds for the Degree in the Nullstellensatz.* Annals of Math. **126** (1987) 577–591.

[5] P. Bürgisser, M. Clausen, M. Amin Shokrollahi: *Algebraic Complexity Theory.* Grundlehren der mathematischen Wissenschaften **315**, Springer (1997).

[6] L. Caniglia, A. Galligo, J. Heintz: *Borne simple exponentielle pour les degrés dans le théorème des zéros sur un corps de charactéristique quelconque.* C. R. Acad. Sci. Paris **307** (1988) 255–258.

[7] T. W. Dubé: *A combinatorial proof of Effective Nullstellensatz.* J. Symb. Comp. **15** (1993) 277–296.

[8] N. Fitchas, A. Galligo: *Nullstellensatz effectif et Conjecture de Serre (Théorème de Quillen-Suslin) pour le Calcul Formel.* Math. Nachr. **149** (1990) 231–253.

[9] N. Fitchas, M. Giusti, F. Smietanski: *Sur la complexité du théorème des zéros.* In J. G. et al., ed., *Approximation and Optimization in the Caribbean II, Proceedings 2nd Int. Conf. on Non-Linear Optimization and Approximation*, Approximation and Optimization **8**, Peter Lange Verlag, Frankfurt am Main (1995) 247–329.

[10] M. Giusti, K. Hägele, J. Heintz, J. E. Morais, J. L. Montaña, L. M. Pardo: *Lower Bounds for Diophantine Approximation.* In *Proceedings of MEGA'96*, Journal of Pure and Applied Algebra **117,118** (1997) 277–317.

[11] M. Giusti, J. Heintz: *La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial.* In D. Eisenbud and L. Robbiano, eds., *Computational Algebraic Geometry and Commutative Algebra*, Symposia Matematica vol. XXXIV, Cambridge University Press (1993) 216–256.

[12] M. Giusti, J. Heintz, J. E. Morais, J. Morgenstern, L. M. Pardo: *Straight–Line Programs In Geometric Elimination Theory.* To appear in J. of Pure and App. Algebra (1997) 1–46.

[13] M. Giusti, J. Heintz, J. E. Morais, L. M. Pardo: *When Polynomial Equation Systems can be solved fast ?* In G. Cohen, H. Giusti, and T. Mora, eds., *Applied Algebra, Algebraic Algorithms and Error Correcting Codes, Proceedings AAECC-11*, LNCS **948**, Springer (1995) 205–231.

[14] M. Giusti, J. Heintz, J. E. Morais, L. M. Pardo: *Le rôle des structures de données dans les problèmes d'élimination.* To appear in C. R. Acad. Sci. Paris (1997) 1–7.

[15] J. Heintz: *On the computational complexity of polynomials and bilinear mappings. A survey.* In *Proceedings AAECC-5*, LNCS **356**, Springer (1989) 269–300.

[16] G. Hermann: *Die Frage der endlich vielen Schritte in der Theorie de Polynomideale.* Math. Ann. **95** (1926) 736–788.

[17] P. Koiran: *Hilbert's Nullstellensatz Is in the Polynomial Hierarchy.* Journal of Complexity **12 (4)** (1996) 273–286.

[18] J. Kollár: *Sharp effective Nullstellensatz.* J. of the AMS **1** (1988) 963–975.

[19] T. Krick, L. M. Pardo: *Une approche informatique pour l' approximation diophantienne.* C. R. Acad. Sci. Paris **318 (1)** (1994) 407–412.

[20] T. Krick, L. M. Pardo: *A Computational Method for Diophantine Approximation.* In L. González-Vega and T. Recio, eds., *Algorithms in Algebraic Geometry and*

Applications. Proceedings of MEGA'94, Progress in Mathematics **143**, Birkhäuser Verlag (1996) 193–254.

[21] J. E. Morais: *Resolución eficaz de sistemas de ecuaciones polinomiales.* Ph.D. thesis, Universidad de Cantabria, Santander, Spain (1997).

[22] A. M. Ostrowski: *On two problems in abstract algebra connected with Horner's rule.* In *Studies in Math. and Mech. presented to Richard von Mises*, Academic Press (1954) 40–48.

[23] L. M. Pardo: *How lower and upper complexity bounds meet in elimination theory.* In G. Cohen, H. Giusti, and T. Mora, eds., *Applied Algebra, Algebraic Algorithms and Error Correcting Codes. Proceedings of AAECC-11*, LNCS **948**, Springer (1995).

[24] P. Philippon: *Denominators in Hilbert's Nullstellensatz.* Acta Arith. **58** (1991) 1–25.

[25] J. Sabia, P. Solernó: *Bounds for Traces in Complete Intersections and Degrees in the Nullstellensatz.* Applicable Algebra in Engineering Communications and Computing (AAECC Journal) **6** (1996) 353–376.

The virtual world as a real trap for the scientist

Joos Heintz

Even if you may not notice during my talk, my main concern is making publicity for certain ideas of collaboration and joint advancement on goals which computer science and mathematics have in common.

In a first part of my talk I will give you my point of view of the current situation of science in general and based on that in a second part the reason why the TERA project group is doing what it is doing now.

The main concerns of this project group are the origins of algebraic geometry and qualitative theory of differential equations: the search for efficient tools for simplifying and solving polynomial and algebro-differential equation systems. This task is of high practical relevance in particular when the input equation system depends on parameters.

Unfortunately a simple minded approach to this question leads to huge complexity problems. In the past, this situation pushed the field of algebraic geometry away from its origins into a purely theoretical direction: it was easier to develop a structural theory of solution varieties rather than to solve the polynomial equation systems themselves.

This talk is devoted to a description of the state of the art in efficient algorithms and data structures for polynomial equation solving and their computer implementations. This description is complemented by recent results on the *intrinsic* worst case complexity of polynomial equation system solving.

The intrinsic complexity of elimination algorithms and their numerical and topological stability

J. Heintz, G. Matera, L.M. Pardo, R. Wachenchauzer

In this contribution we show that any *general* elimination procedure for parametric polynomial equation systems must *necessarily* be exponential if we require the representation of the output to satisfy certain natural conditions of topological (or numerical) stability.

In fact, the most common parametric elimination problems have solutions which mathematically can be represented as polynomial maps between low–dimensional parameter and high–dimensional function spaces (in fact: jet spaces). In general these solution–maps are topologically (and numerically) highly unstable. Any algorithm which solves the elimination problem under consideration induces an unfolding of the solution map. This unfolding encodes the data type selected to represent the output.

We call an elimination algorithm *geometrically stable* if it produces for any input representation an encoding of the output which depends only on the input *object* but *not* on its *representation.* This (modest) uniformity requirement is satisfied by all known *parametric* elimination procedures.

Moreover this uniformity implies that the algorithm under consideration induces an unfolding of the solution–map which owns a certain topological (and numerical) stability property. This stability property reflects sharply the complexity behaviour of the algorithm.

We construct specific example series of geometric elimination problems with highly singular solution–maps. Any geometrically stable algorithm which solves this problem induces an unfolding which operates on a huge parameter space. This implies the claimed exponentiallity of the output size for any geometrically stable algorithm which solves our problem.

Our proof techniques require only elementary differential geometry and basic knowledge of algebraic complexity theory (see [1]). The method itself is new for lower bound proofs for the evaluation complexity of polynomial maps. A full paper version of this result is submitted to publication [2].

References

[1] P. Bürgisser, M. Clausen, A. Shokrollahi: *Algebraic Complexity Theory.* A Series of Comprehensive Studies in Mathematics **315**, Springer–Verlag (1996).

[2] J. Heintz, G. Matera, L.M. Pardo, R. Wachenchauzer: *About the intrinsic complexity of parametric elimination procedures.* WAIT'97, Buenos Aires, Argentina (1997).

Estudio de los Mecanismos de Crossover en el Problema del Replegado de Proteínas

Natalio Krasnogor, David Pelta, Pablo E. Martínez López, Esteban de la Canal [1]

Resumen: El problema de encontrar la conformación tridimensional de una proteína a partir de la secuencia primaria de aminoácidos es de difícil solución. Esto se debe a dos razones fundamentales: las contribuciones de la energía libre que estabilizan la estructura funcional no han sido totalmente entendidas, y por otro lado, el espacio de configuraciones posibles crece exponencialmente con el tamaño de la instancia. El problema de encontrar el estado de mínima energía en modelos discretos puede ser visto como un problema de optimización combinatoria. Resultados independientes han mostrado que versiones simples de este problema son *NP-completas* [18], [19], [10]. La estructura tridimensional que se pretende inducir a partir de la secuencia primaria de aminoácidos determina la función biológica de la proteína en cuestión. Es por este motivo que el estudio del *Protein Folding* tiene motivaciones tanto teóricas como prácticas.

En [9] se justifica el uso de metaheurísticas como los algoritmos genéticos y meméticos en problemas en los que es posible asociarles una función objetivo con estructura jerárquica, como por ejemplo el de predicción de la estructura terciaria a partir de la secuencia de aminoácidos. En este trabajo se comentarán algunos modelos discretos para el estudio del Protein Folding, como así también algunas técnicas de investigación operativa que han sido aplicadas a este problema. Se pondrá especial énfasis en la utilización de algoritmos genéticos, la evaluación de su desempeño [13] y nuevas líneas de investigación.

1. Introducción

En los últimos años una nueva disciplina ha surgido, la Biología Computacional.

Su objeto de estudio son los problemas surgidos de las ciencias biológicas desde una perspectiva matemática y computacional. El problema del *Plegado de proteínas* (PF en adelante) es uno de los más importantes problemas abiertos en biología.

Las proteínas están compuestas de 20 tipos diferentes de aminoácidos que se ligan unos a otros dando lugar a una larga cadena molecular. Una vez ensamblados los aminoácidos, la molécula resultante se pliega en el espacio hasta alcanzar la llamada

---

[1] URL: `http://www-lifia.info.unlp.edu.ar`

estructura terciaria o estado nativo.

Es esta estructura tridimensional la que determina la función de la proteína en cuestión. El problema del *PF* consiste entonces en poder determinar, dada la cadena de aminoácidos, cual será la estructura final que la proteína adoptará.

Existen diversas formulaciones abstractas de este problema que permiten obtener buenas aproximaciones a lo que realmente acontece con las proteínas dentro de las células o en los cultivos.

Un modelo abstracto del *PF* es el sugerido por Paterson et al., el cual resulta muy interesante desde la perspectiva de las matemáticas discretas.

En su trabajo ellos consideran el problema del *String Folding*, que puede ser definido como sigue:

dada una cadena finita S, *un entero* k *y una grilla* Gr *existe un doblado de* S *en* Gr *con un puntaje no menor a* k*? donde un doblado de* S *en* Gr *se define como un mapeo inyectivo* F*: [1...n]*$\mapsto Gr$*, donde* $n = \mid S \mid$*, y si* $1 \leq i, j \leq n$*, i=j-1 entonces* $F(i)$ *es adyacente a* $F(j)$ *en* Gr*;*

El puntaje de F se computa contando el número de pares de símbolos identicos que se mapean a nodos adyacentes de Gr. Esos pares se denominan *bonds*. En su trabajo Paterson *et al.* muestran que *String Folding* es *NP-completo* en Z^2 y Z^3 mientras que en otras instancias se mantiene *NP-hard*.

Un modelo alternativo más sencillo, el llamado modelo de Dill, considera a los aminoácidos agrupados en solo dos tipos.

En la naturaleza existen 20 aminoácidos distintos, que en general pueden clasificarse segun su hidrofobicidad. Esto es, los hidrofílicos o tipo H , y los hidrofóbicos o tipo P.

Una proteína entonces, sera representada por una secuencia en $(H^*P^*)^*$. Graficamente consideramos a los aminoácidos P de color negro y a los H de color blanco.

La función de energía utilizada, solo tiene en cuenta las interacciones entre vecinos topológicos de tipo P.

Generalmente se considera que la proteína se encuentra en un reticulado (2 o 3 dimensiones), donde cada posición es ocupada por a lo sumo un aminoácido.

La correspondencia entre aminoácidos y posiciones se llama *embedding* de la proteína, y cuando el embedding es inyectivo se denomina *self avoiding*.

El *PF*, bajo estas condiciones, se demostró NP-completo en [3] y en [19]. Por lo tanto, la existencia de un algoritmo polinomial que lo resuelva en forma exacta se considera imposible. Es por esto que el uso de heurísticas y algoritmos de aproximación se convierten en la alternativa más viable para resolverlo, al menos, para algunas de sus instancias.

Existen soluciones a *PF* usando Simulated Annealing [14], algoritmos genéticos y el método de Monte Carlo [18]; también en [12] se presenta un algoritmo polinomial determinístico que sigue las líneas de [18].

Lo que es importante notar es que los modelos estudiados son *NP-hard*, *NP-completo*, etc., y no el problema en si. Este *quid pro quo* es tal vez el origen de la paradoja de Levinthal. En este trabajo no haremos un resumen del estado del arte respecto a estos algoritmos. Lo que los autores buscamos es llamar la atención a un error que se ha venido cometiendo sistemáticamente a la hora de utilizar algoritmos genéticos para obtener soluciones heurísticas a este problema. Y este error es el de no haber tenido en cuenta el carácter altamente epistático de las representaciones cromosomáticas utilizadas. Vamos a mostrar como los aparentemente buenos resultados alcanzados con los algoritmos genéticos desarrollados hasta el momento no son tan buenos y sugeriremos la manera de corregirlos.

2. Algoritmos Genéticos y Protein Folding

En esta sección, comentaremos tres implementaciones diferentes de algoritmos genéticos para resolver el *PFP* y en la sección próxima compararemos brevemente los resultados. Los artículos [18] y [11] representan al universo de trabajos sobre la aplicación de AG al Protein Folding, en el sentido que la gran mayoría de los mismos comparten los problemas que comentaremos mas abajo. Por este motivo, y por ser el grupo de Unger de los primeros en estudiar la complejidad computacional del PF en modelos reducidos, es que se eligieron estos dos trabajos.

2.1. Algoritmo de Unger y Moult

En su trabajo [18], Unger y Moult dicen:

> .. Our application of GAs to the protein folding problem may be regarded as an extension of the more familiar *Monte Carlo (MC)* method to include information exchange between a set of parallel simulations. ...

Describiremos a continuación las características más importantes de esta implementación y en secciones subsiguientes refutaremos el comentario transcripto más arriba.

Unger y Moult adoptan en su trabajo el modelo de Dill en un reticulado bidimensional e implementan un AG para resolver el problema. Dicho algoritmo genético aplica el método de *Simulated Annealing* para los períodos de mutación y crossover.

Simulated Annealing puede sintetizarse de la siguiente manera:

se parte de una conformación inicial aleatoria S_1 con energía E_1. Luego se realiza un pequeño cambio aleatorio en S_1 para generar S_2 y se calcula su energía, E_2. Si $E_2 \leq E_1$ entonces se acepta S_2. En caso contrario se elegirá, en forma no determinística, si el cambio es aceptado o no de acuerdo a alguna función de probabilidad. Generalmente el criterio es:

$$Rnd \leq \exp^{\frac{E_1 - E_2}{c_k}}$$

donde $0 \leq Rnd \leq 1$ y c_k es el factor de enfriamiento que va cambiando en función de k, siendo k el número de pasos de iteración. Si no se acepta, se elimina S_2 y se continúa iterando a partir de S_1 nuevamente. En caso de aceptarse, el proceso continúa desde S_2. Se repiten los pasos previos hasta alcanzar algun criterio de finalización.

El esquema de Unger y Moult es un esquema de mutación, crossover y selección, donde los pasos de mutación se corresponden con un paso del método arriba descripto.

La representación de los individuos de la población es directa, es decir, no se representan las soluciones como cadenas de bits. Un individuo consiste en una palabra $w \in \{U, D, R, L\}^*$ con $\|w\| = \|i\| - 1$, donde i es una instancia en el modelo de Dill. U, D, R, L corresponden a las posiciones relativas que puede tomar un aminoácido con respecto a su anterior.
Para cruzar dos individuos, el algoritmo elige un punto al azar, corta por ese punto cada padre y une las subsecuencias complementarias para generar dos individuos nuevos. Existen tres maneras de unir los substrings, esto es, conectarlos a 0, 90 o 270 grados. Si la energía de estos individuos es menor que el promedio de la energía de los padres entonces son aceptados, en caso contrario se decide no determinísticamente (entre descartarlo y generar otro, o aceptarlo), utilizando el mismo procedimiento que el descripto en Simulated Annealing.

2.2. Algoritmo de Patton et al.

En el trabajo de Patton et al. [11], se critica el carácter híbrido del algoritmo de Unger y Moult y se presenta un AG estándar. Dichas críticas apuntan a:

- La inicialización no random de la población.
- El alto grado de mutaciones.
- El filtrado de los resultados del crossover y las mutaciones mediante iteraciones de tipo *simulated annealing*.
- El efecto en cascada y acoplado de, por un lado, rechazar las configuraciones no válidas, y por otro lado, los filtrados mediante *SA*, desvían la capacidad de exploración y explotación del AG.

- El reemplazo de toda la población, excepto del individuo mejor.
- El encoding directo de la dirección de un aminoácido respecto de su predecesor.

Para corregir estos "errores" los autores presentan un AG estándar con las siguientes características:

- 0.95 de probabilidad de crossover.
- 1-point y 2-point crossover.
- probabilidad de mutación menor a .001.
- tamaño de población variable hasta un 20%.
- reducción del incesto.
- población de 1000 individuos.
- penalización de las posiciones con colisión (no de los péptidos que colisionan).
- encoding en base a movimientos relativos entre el aminoácido i y el $i-1$, que evita algunas colisiones obvias (por ejemplo UD o LR).

Con este algoritmo, los autores consiguen bajar notablemente la cantidad de evaluaciones de la función de energía y a la vez, obtienen resultados de energía libre comparables e incluso, en algunos casos, mejores. Ver tabla 1 en [11]. Sin embargo...

Nuestras Hipótesis

En las secciones anteriores se presentaron dos trabajos en los cuales mediante el uso de algoritmos genéticos se trata de resolver el problema del protein folding.

En ambos trabajos se elogian las ventajas de los AGs frente a otros métodos y en la investigación llevada a cabo por Patton se critican algunos puntos importantes del trabajo de Unger, a la vez que se indica una alternativa que solucione los inconvenientes citados.

Patton et.al., al comparar resultados con los obtenidos por Unger y ver que logran mejorarlos, consideran verificadas sus ideas. Nuestras hipótesis de trabajo suponen que los problemas marcados por Patton en la solución de [18], si bien, importantes, no son condición necesaria y suficiente para el mal desempeño del AG en cuestión.

Para demostrar esto, implementamos nuestro propio AG compartiendo las supuestas "dudosas" características del trabajo cuestionado. Las simulaciones ejecutadas arrojan resultados inesperados. Con esas características perniciosas se obtienen mejores conformaciones que en [18], con un speed up comparable al de [11].

Más aún, suponemos que los problemas que hemos encontrado en el comportamiento de los AG analizados en el presente trabajo se deben a causas más profundas y compartidas por los tres diseños.

2.2.1. Nuestro Algoritmo

Con el objeto de comprobar nuestra hipótesis implementamos una versión propia de AG para resolver *PF*. Reducimos el modelo a 2 dimensiones, ya que los principios a estudiar no se pierden al bajar la dimensionalidad del problema [15], [16], [17]. Las características principales de nuestra implementación se describen a continuación.

El proceso de selección asigna una probabilidad de apareo P_i a cada individuo i de la población. Esta probabilidad se computa de acuerdo al fitness de cada i: $P_i = \frac{f_i}{F}$ donde f_i es el fitness del individuo i y $F = \sum_{i=1}^{i=N} f_i$, siendo N el tamaño de la población. Durante el proceso de cruzamiento dos padres generan solo un hijo. Esto obliga a usar un conjunto de entrecruzamiento de $2 * (N - Z)$ genomas. Z es el tamaño del conjunto de elite.

Una vez que el conjunto de entrecruzamiento ha sido llenado, el crossover propiamente dicho tiene lugar. Dos padres se seleccionan y se cruzan con una probabilidad de P_X. El crossover que se utiliza es el mismo que en [19]. Si el entrecruzamiento no acontece, entonces el padre con mejor fitness se copia a la proxima generación. Además de mutación, el algoritmo cuenta con la posibilidad de realizar macromutaciones. Estas pueden ser de cuatro tipos:

- Se seleccionan dos puntos al azar. La cadena entre este par de puntos se modifica de tal forma de representar un giro de 0, 90, 180 o 270 grados del segmento de proteína en cuestión.
- Dos puntos son elegidos al azar. Entre dichos puntos se modifica el genoma de tal forma de reflejar, vertical u horizontalmente, la configuración de la proteína.
- La proteína se "estira" entre un par de puntos elegidos al azar.
- La dirección de los nucleótidos entre un par de puntos elegidos al azar se selecciona random. Esta es la macromutación con menos estructura de las descriptas.

Además, el AG preserva el mejor individuo y lo copia a la próxima generación.

Para resumir se puede decir que nuestro algoritmo comparte características con los AGs anteriores, por ejemplo:

- La inicialización no random de la población.
- El alto grado de mutaciones.

- Probabilidad de macromutaciones diferente de cero.
- 1-point y 2-point crossover.
- El reemplazo de toda la población, excepto el elite set (reducido).
- Permite configuraciones no válidas.
- Penalización de las colisiones.
- La codificación directa de la dirección de un aminoácido respecto de su predecesor, idéntico al de Unger y Moult.

Es importante notar que para poder probar nuestras hipótesis lo que se hizo fue elegir la codificación de Unger y Moult que permite muchas más configuraciones inválidas que el de Patton pero no filtramos dichas configuraciones, solo las penalizamos.

Es decir, la elección de encoding e individuos validos es la más desfavorable.

Por otro lado hemos utilizado una población mucho más pequeña que las usadas en los otros trabajos, 50 individuos en nuestro caso contra no menos de 100 en los otros trabajos llegando inclusive hasta 1000 en algunas corridas. La inicialización heurística y no aleatoria de la población lejos de ser perjudicial para el AG, como supone Patton, lo ayuda a salir de las regiones de configuraciones infactibles del espacio de búsqueda.

Los valores de energía alcanzados con nuestro AG son mejores que los mostrados en [18] y presentan un speed up comparable a los de [11]. Referirse a dichos trabajos para comparar resultados con las tablas que aquí se presentan.

Por otro lado, el crossover en este trabajo es mucho más sencillo que el de Patton que no es escalable debido al alto costo de decodificación, $O(\|i\| * n^2)$ donde n es la cantidad de posiciones relativas posibles e i, como siempre, una instancia.

Son estas evidencias las que nos inclinan a decir que las críticas de Patton et.al. al trabajo de Unger son sólo superficiales.

3. Experimentos y Resultados Obtenidos

Debido a que nuestro objetivo es mostrar como fueron mal interpretados los resultados de AGs aplicados a modelos reducidos de folding es que decidimos realizar simulaciones extensivas. Las pruebas se hicieron sobre instancias de proteínas extraídas de [18]. Dichas instancias corresponden al modelo de Dill [2], y se muestran en la figura de abajo.

i) `BWBWWBBWBWWBWBBWWBWB`

ii) `BBWWBWWBWWBWWBWWBWWBWWBB`

iii) `WWBWWBBWWWWBBWWWWBBWWWWBB`

iv) `WWWBBWWBBWWWWWBBBBBBBWWBBWWWWBBWWBWW`

v) `WWBWWBBWWBBWWWWWBBBBBBBBBBWWWWWWBBWW`
`BBWWBWWBBBBB`

vi) `BBWBWBWBWBBBBWBWWWBWWWBWWWWBWWWBWWWB`
`WBBBBWBWBWBWBB`

vii) `WWBBBWBBBBBBBBWWWBBBBBBBBBBWBWWWBBBB`
`BBBBBBBBWWWWBBBBBBWBBWBW`

viii) `BBBBBBBBBBBBWBWBWWBBWWBBWWBWWBBWWBBW`
`WBWWBBWWBBWWBWBWBBBBBBBBBBBB`

Para cada una de estas instancias, se determino cual era la terna (P_X, P_{MM}, P_M) para la cual el AG obtuvo los mejores resultados.

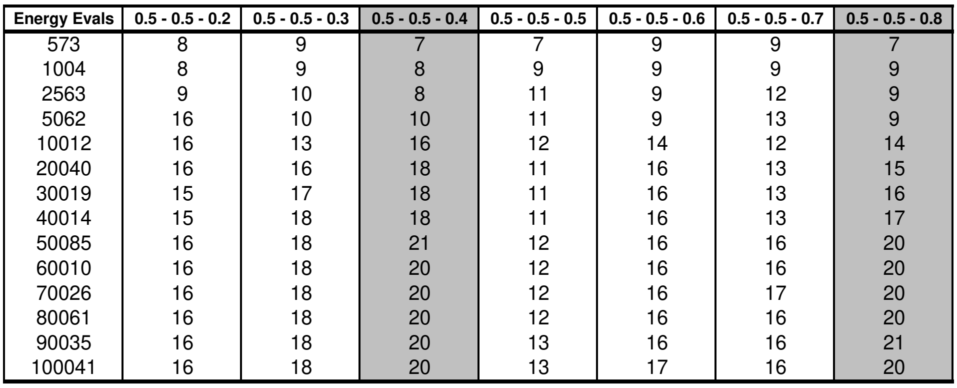
(P_X, P_{MM}, P_M) son las probabilidades de crossover, macromutación y mutación respectivamente;
(P_X, P_{MM}, P_M) $\in$ {0.2,0.3, 0.4, 0.5, 0.6, 0.7, 0.8}.

La calidad del desempeño del AG esta dado obviamente por la minimización de energía libre de una instancia dada, o lo que es lo mismo, la maximización del número de bonds.

Para cada instancia, todas las combinaciones de valores posibles fueron probadas. Esto implicó testear para cada instancia 7^3 posibles parámetros del AG. Una vez obtenidos los resultados de estos experimentos, se filtró para cada instancia la terna que maximizaba el número de bonds. En las tablas 2, 3 y 4 se muestra el resultado del filtrado para las instancias UM6, UM7 y UM8 en base a una inicialización heurística.

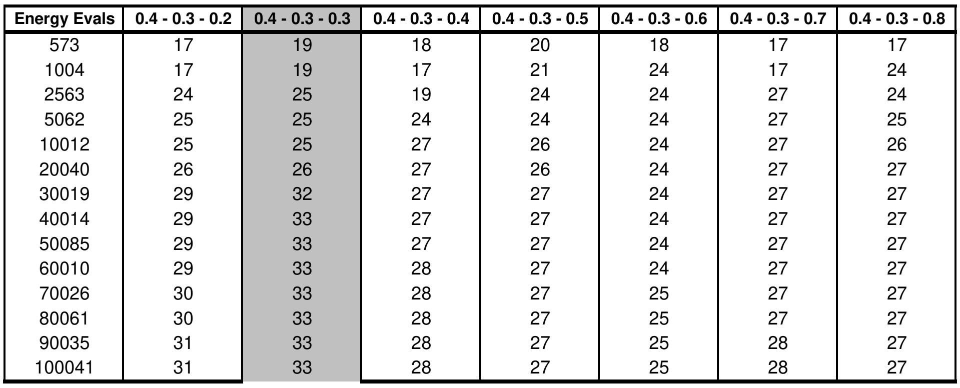
El mismo procedimiento con inicialización random, esta reflejado en las tablas 5 y 6, donde 5 corresponde a la instancia UM6 y la restante a UM8. Para UM7 no se obtuvieron conformaciones válidas. En estas tablas, los valores iniciales altos corresponden a conformaciones inválidas.
Los resultados obtenidos *a priori* eran los que esperabamos encontrar. En todos los casos en los que el número de bonds era máximo, la probabilidad de crossover, P_X, era menor que las otras dos probabilidades, o en el mejor de los casos iguales.

Estos hechos parecerían estar en conflicto con la intuición generalizada respecto del comportamiento de los AG. Es ampliamente aceptado que tanto P_{MM} como P_M deben ser mucho menores que P_X.



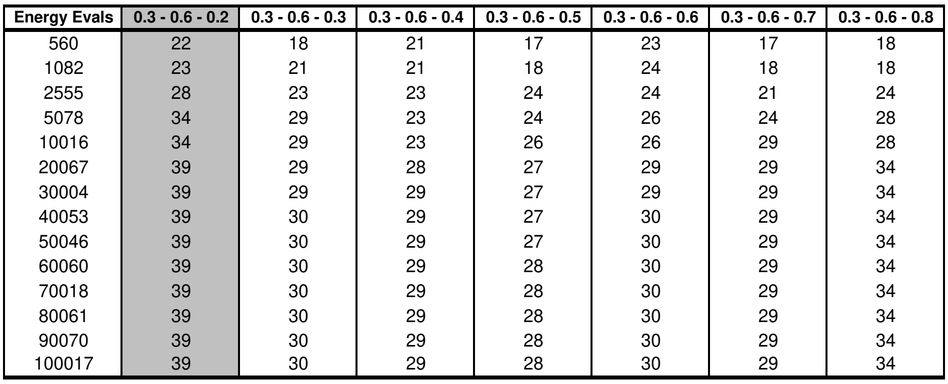
Energy Evals	0.5 - 0.5 - 0.2	0.5 - 0.5 - 0.3	0.5 - 0.5 - 0.4	0.5 - 0.5 - 0.5	0.5 - 0.5 - 0.6	0.5 - 0.5 - 0.7	0.5 - 0.5 - 0.8
573	8	9	7	7	9	9	7
1004	8	9	8	9	9	9	9
2563	9	10	8	11	9	12	9
5062	16	10	10	11	9	13	9
10012	16	13	16	12	14	12	14
20040	16	16	18	11	16	13	15
30019	15	17	18	11	16	13	16
40014	15	18	18	11	16	13	17
50085	16	18	21	12	16	16	20
60010	16	18	20	12	16	16	20
70026	16	18	20	12	16	17	20
80061	16	18	20	12	16	16	20
90035	16	18	20	13	16	16	21
100041	16	18	20	13	17	16	20

Figure 1: Filtrado para la instancia 6



Energy Evals	0.4 - 0.3 - 0.2	0.4 - 0.3 - 0.3	0.4 - 0.3 - 0.4	0.4 - 0.3 - 0.5	0.4 - 0.3 - 0.6	0.4 - 0.3 - 0.7	0.4 - 0.3 - 0.8
573	17	19	18	20	18	17	17
1004	17	19	17	21	24	17	24
2563	24	25	19	24	24	27	24
5062	25	25	24	24	24	27	25
10012	25	25	27	26	24	27	26
20040	26	26	27	26	24	27	27
30019	29	32	27	27	24	27	27
40014	29	33	27	27	24	27	27
50085	29	33	27	27	24	27	27
60010	29	33	28	27	24	27	27
70026	30	33	28	27	25	27	27
80061	30	33	28	27	25	27	27
90035	31	33	28	27	25	28	27
100041	31	33	28	27	25	28	27

Figure 2: Filtrado para la instancia 7



Energy Evals	0.3 - 0.6 - 0.2	0.3 - 0.6 - 0.3	0.3 - 0.6 - 0.4	0.3 - 0.6 - 0.5	0.3 - 0.6 - 0.6	0.3 - 0.6 - 0.7	0.3 - 0.6 - 0.8
560	22	18	21	17	23	17	18
1082	23	21	21	18	24	18	18
2555	28	23	23	24	24	21	24
5078	34	29	23	24	26	24	28
10016	34	29	23	26	26	29	28
20067	39	29	28	27	29	29	34
30004	39	29	29	27	29	29	34
40053	39	30	29	27	30	29	34
50046	39	30	29	27	30	29	34
60060	39	30	29	28	30	29	34
70018	39	30	29	28	30	29	34
80061	39	30	29	28	30	29	34
90070	39	30	29	28	30	29	34
100017	39	30	29	28	30	29	34

Figure 3: Filtrado para la instancia 8

Energy Evals.	0.7 - 0.7 - 0.2	0.7 - 0.7 - 0.3	0.7 - 0.7 - 0.4	0.7 - 0.7 - 0.5	0.7 - 0.7 - 0.6	0.7 - 0.7 - 0.7	0.7 - 0.7 - 0.8
545	138	90	140	146	115	223	131
1040	138	90	140	146	115	223	131
2525	138	90	140	146	115	223	0
5099	12	0	140	146	115	223	7
10049	20	19	140	146	115	223	18
20048	28	27	140	146	115	223	23
30047	28	27	140	146	115	223	25
40046	28	32	140	146	16	223	26
50045	28	32	140	146	23	223	26
60044	28	32	140	146	26	223	26
70043	28	32	140	146	28	223	26
80042	28	32	140	146	28	223	26
90041	28	32	140	146	28	223	26
100040	28	32	140	146	29	223	26

Figure 4: Filtrado para la instancia 8 random

Energy Evals	0.7 - 0.7 - 0.2	0.7 - 0.7 - 0.3	0.7 - 0.7 - 0.4	0.7 - 0.7 - 0.5	0.7 - 0.7 - 0.6	0.7 - 0.7 - 0.7	0.7 - 0.7 - 0.8
545	11	0	1	1	14	1	55
1040	11	3	1	2	14	6	55
2525	11	5	5	5	14	8	55
5099	2	8	9	7	12	10	55
10049	13	11	12	11	13	11	11
20048	13	15	13	12	14	10	12
30047	13	17	13	14	15	11	13
40046	13	17	13	15	15	11	15
50045	13	17	13	15	15	12	15
60044	13	17	13	15	15	13	15
70043	13	18	16	15	15	14	15
80042	13	18	16	15	15	16	15
90041	15	20	16	15	15	17	15
100040	15	20	16	15	16	17	15

Figure 5: Filtrado para la instancia 6 random

Sin embargo en estos experimentos encontramos que los mejores conjuntos de parámetros son aquellos en los cuales las probabilidades de macromutación y la mutación son mayores que la de crossover.

4. Análisis e Interpretación de Resultados

Tradicionalmente en la literatura de algoritmos genéticos se encuentra que la presentación de resultados responde a algunos cánones.

Cuando el algoritmo genético ha sido utilizado como herramienta de optimización combinatoria en algun problema en particular, los autores se conforman con obtener "buenas" soluciones y realizar comparaciones con otros trabajos.

Lo que muy pocas veces se encuentra es un análisis de "cual" fue el camino que recorrio el algoritmo genético para alcanzar los supuestamente buenos resultados.

Una mirada atenta a las tablas que se presentan en los trabajos, permite analizar cual es la dinámica del AG.

En el caso del problema que nos compete, este análisis no se ha realizado. Los autores no han estudiado detenidamente como sus algoritmos alcanzaron dichos resultados. Probablemente por este motivo los AG no han resultado en un beneficio significativo respecto de otras metaheurísticas.

En las figuras 6, 7 y 8 se puede observar la dinámica del algoritmo. Se grafican el número de bonds en función del número de evaluaciones de la función de energía, es decir, bonds en función de tiempo.
Estos gráficos muestran el mismo comportamiento que las corridas de Unger y Moult, como asi tambien las de Patton et.al. En dichas figuras se observa que a medida que aumentan las evaluaciones de energía las conformaciones obtenidas mejoran.

Sin embargo, esta mejora no es en modo alguna continua sino más bien a saltos. Deben pasar muchas generaciones antes de que el AG "descubra" una conformación mejor. Estos saltos evidencian una dinámica muy similar a la que se podría esperar de un método similar a SA o MC.

La cuestión a responder es: como el AG crea nueva información?.

Fundamentalmente existen dos ideas en las que un AG se basa para resolver un problema: la mutación y el crossover.

El crossover básicamente sirve a dos propósitos:

La transferencia de información entre individuos de la población y el mantenimiento y reorganización de genes con el objeto de descubrir los building blocks.

La mutación, en cambio, es utilizada para incorporar material genético nuevo a

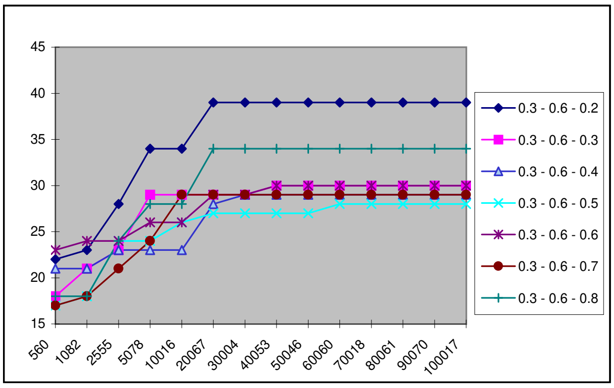


Figure 6: Bonds / Tiempo para UM8

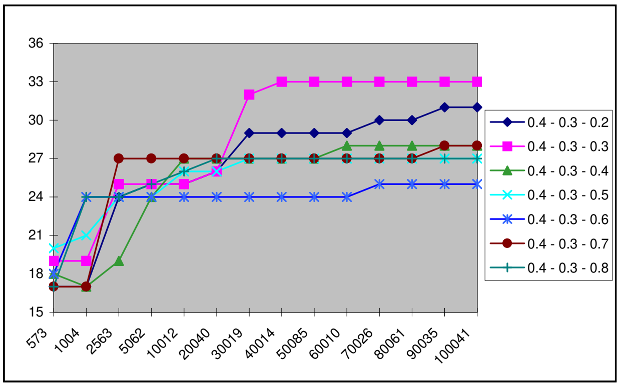


Figure 7: Bonds / Tiempo para UM7

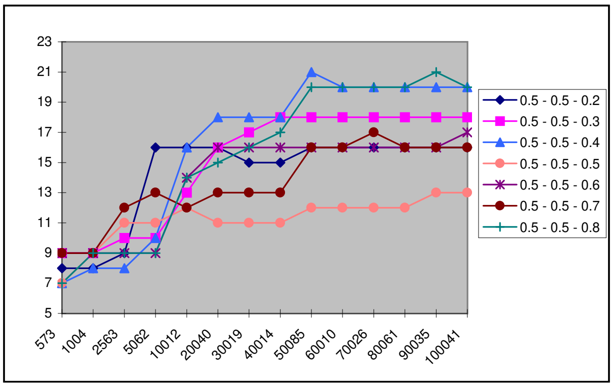


Figure 8: Bonds / Tiempo para UM6

la población, lo que permite obtener individuos que estarian fuera de la clausura transitiva del crossover, y por ende, inalcanzables mediante entrecruzamiento.

Comportamientos como el presentado en las figuras 6 a 8 son fácilmente entendidos al considerar los dos mecanismos arriba mencionados.

Es claro que la evolución de la población hacia óptimos locales mejores se debe al mecanismo de selección. El carácter de a saltos de este proceso es reflejo del descubrimiento por parte de la mutación de genomas que codifican mejores individuos.

Los períodos de planicie corresponden a intervalos de tiempo en donde el crossover esta "tratando" de reorganizar los building blocks. Esta reorganización permitirá más tarde el surgimiento de alguna otra mutación que lleve a la población hacia alguna meseta superior. Sin embargo, estos períodos de mejora son muy lentos, las planicies muy largas, y es por esto que explora el espacio de búsqueda muy lentamente. Este fenómeno ha sido evidenciado en todos los trabajos con AG aplicados al protein folding.

Dichos trabajos fracasan entonces, en el uso correcto del crossover. El mecanismo de entrecruzamiento no esta transmitiendo información entre individuos de la población ni tampoco ha sido diseñado teniendo en cuenta la representación del genoma. Es por esto que al no haber una relación estrecha entre la representación cromosómica y el crossover, las mesetas son muy largas.

El entrecruzamiento se ve imposibilitado de descubrir y transmitir buenos esquemas a la población. En la figura 9 podemos ver un ejemplo de como el $1-$point crossover, no solo no transmite información entre individuos de la población, sino que también destruye los building blocks ya encontrados. Este mismo fenómeno acontece con $N-$point crossover si se mantiene la misma codificación. Es decir, el mal desempeño se debe a la pobre interacción entre la codificación y el crossover elegido. En una codificación lineal como la utilizada en los trabajos analizados, los building blocks son muy grandes y por ende es poco probable que se mantengan intactos luego del crossover.
La figura 9 muestra 2 padres y un hijo de izquierda a derecha. Los bonds en cada proteína se muestran con una línea de puntos. Asi, el primer padre tiene 3 bonds, el segundo solo 1 al igual que el hijo.

Hay que destacar que el bond hallado en el hijo no es transmitido de los padres; esto es, la nueva conformación no hereda nada de sus predecesores. Ese bond esta contruído con los aminoácidos (a_1, e_2). Podemos ver que no hay intercambio útil de información.

El carácter altamente epistático de la codificación, que da motivo al título de este trabajo y que no ha sido notado por los autores de los otros trabajos, determina que los bonds en los padres estén formados por aminoácidos muy lejanos en el cromosoma uno de otros.

En este ejemplo, $\{(a_1, e_1), (a_1, d_1), (b_1, c_1)\}$ para el primer padre, y $\{(a_2, e_2)\}$ para el segundo.

Cuando ocurra una etapa de apareamiento bajo este encoding con un crossover de tipo $N-$point, es muy poco probable que los genes que codifican estos bonds producto de la epistasis, sean transferidos juntos. Más aun, suponiendo que se transfieran juntos es mucho menos probable aún que queden en posiciones relativas que les permitan volver a constituir los bonds en cuestión.

4.1. Diferencia Entre el Mecanismo y la Idea de Crossover

En [5] se distingue entre el concepto o idea de crossover y la mecánica del mismo. Como dijimos en la sección anterior, el crossover cumple dos roles que podemos llamar la *idea* y la *mecánica*.

La *idea* subyacente a la utilización de crossover es que existe alguna razón para creer que cada uno de los padres tiene una posibilidad mejor que el promedio de conferir a sus descendientes un conjunto de building blocks mejores que el promedio. En otras palabras, lo que se pretende al utilizar crossover es que los esquemas prometedores que algun individuo fue descubriendo a lo largo de su evolución sea comunicado a la población.

Se espera intercambio de información; de esta forma una población que inicialmente estaba compuesta por individuos aleatorios, debido a la selección convergerá hacia una población con individuos cuya performance sea mejor que la de una población random.

El *mecanismo* del crossover es una implementación en particular de la idea arriba descripta. Si bien todas las implementaciones se remiten a esta idea, en general pueden ser muy diferentes entre si.

Un aspecto muy importante a tener en cuenta cuando analizamos el comportamiento de un algoritmo genético en algun entorno de explotación es verificar cual es el beneficio verdadero que reporta el uso del crossover. Esto es, verificar la utilidad no solo de la mecánica del mismo sino también, constatar que se este sacando provecho de la idea del entrecruzamiento.

Si la mecánica del crossover no implementa la idea del crossover, su utilización equivale a realizar solo macromutaciones a los genomas.

Entonces el uso de una población no es necesario, pudiendose descartar la misma.

Esto trae aparejado un ahorro en tiempo de cómputo. Al mismo tiempo, simplifica el algoritmo limitándolo a realizar un hillclimbing en base a macromutaciones en un solo individuo.

Terry Jones en su tesis de doctorado sugiere un experimento para dilucidar si el

crossover funciona correctamente en un determinado AG. El experimento consiste en comparar un AG estándar con uno idéntico pero que utilice "random crossover". El random crossover funciona de la siguiente manera:
se eligen dos padres $P1$ y $P2$ de los que se generaran dos hijos $H1$ y $H2$. Para obtener Hi se generan $R1$ y $R2$, individuos aleatorios, y se aparean $P1$ con $R1$ y $P2$ con $R2$. De cada apareamiento se obtienen dos hijos, eligiéndose, nuevamente al azar, sólo uno de ellos, el Hi. Es claro que el crossover aleatorio mantiene la mecánica del crossover pero no refleja la idea del crossover de transmitir información. Si luego de comparar la performance del AG estándar con el AG con random crossover, se observa que el random iguala o supera al crossover natural, entonces podemos descartar la población.

Debería resultar claro al lector, que esta comparación está incluída en la serie de experimentos descriptos en secciones anteriores, al permitir que la probabilidad de macromutación varíe en los mismos rangos que las de mutación y crossover. Nuestros resultados concuerdan con los análisis que realiza Jones en su tesis doctoral.

5. Conclusiones

En este paper hemos demostrado como los, supuestamente, buenos resultados obtenidos con el uso de AG en modelos reducidos de protein folding no son tales, por lo menos en los casos estudiados.

Nuestro resultados nos permiten afirmar:

- Las críticas de Patton al trabajo de Unger y Moult, justificadas en ciertos aspectos, no son condición necesaria y suficiente para que un AG falle al resolver el protein folding.

  Para demostrar esto hemos implementado una versión de AG que posee dichas características "malas" y aún así obtuvimos resultados mejores que en [18] y comparables con [11]. Por ejemplo, hemos sido capaces de obtener una configuración de 39 bonds para UM8, donde el óptimo es 42 y el equipo de Unger obtiene 37, y en menor cantidad de evaluaciones de energía.

- Para los algoritmos descriptos, los mismos resultados se pueden obtener prescindiendo de la población y optimizando con un solo individuo que haga hillclimbing, como sugiere Jones en [5].

  Esta conclusión se fundamenta en el hecho de no existir una buena interrelación entre la codificación utilizada y el crossover elegido. De tal forma que lo único que se está explotando del crossover es su mecánica y no la idea del mismo. Por lo tanto utilizando sólo macromutaciones se obtiene el mismo efecto.

- La afirmación de Unger y Moult citada en página 4, e implícitamente aceptada por el equipo de Patton ha sido claramente refutada con los experimentos realizados y justificada en el punto anterior.

6. Trabajo Futuro

No es nuevo para la comunidad científica dedicada al estudio de los algoritmos genéticos el uso de crossovers multidimensionales. Puede resultar de interés estudiar su utilización en la predicción de la estructura terciaria de proteínas. Por ejemplo se podría adaptar un crossover $N-$dimensional al problema del string folding [10] en N dimensiones.

Un enfoque alternativo es la generación de esquemas $2-$dimensionales a instancias del modelo de Dill en el plano.

La utilización de heurísticas de inicialización de la población está siendo cuidadosamente estudiada [6], [7], [13].

7. Agradecimientos

Los autores quieren agradecer a Roman García por su apoyo constante en la utilización de los recursos computacionales, Pablo Mocciola por los trabajos conjuntos desarrollados con anterioridad, Gabriela Arevalo por su colaboración en la tarea de edición.

Natalio Krasnogor quiere agradecer especialmente a Pablo Moscato y Michael Norman por las innumerables creativas y esclarecedoras discusiones.

Los autores agradecen además a los jueces anónimos de este trabajo.

8. References

[1] Than N. Bui et al.: *On Multi-Dimensional Encoding/Crossover.* Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kauffman (1995) 49–56.

[2] Ken A. Dill: Biochemistry **24** (1985) 1501.

[3] Aviezri S. Fraenkel: *Complexity of Protein Folding.* Bulletin of Mathematical Biology **6** (1993).

[4] W.E. Hart, S. Istrail: *Fast Protein Folding in the Hydrophobic Hydrophilic Model within Three Eighths of Optimal.* Proceedings of the 27th ACM Symposium on Theory of Computation (1995) 157–168.

[5] Terry Jones: *Crossover, Macromutation, and Population based Search.* Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kauffman (1995) 73–80.

[6] N. Krasnogor, P.E. Martínez López, P. Mocciola, D. Pelta: *Protein Folding meets Functional Programming.* First International Conference on Computational molecular Biology poster proceedings and International Conference on Functional Programming SIGPLAN proceedings (1997).

[7] N. Krasnogor, P.E. Martínez López, P. Mocciola, D. Pelta: *Modelling String Folding with G2L Grammars.* First International Conference on Computational molecular Biology poster proceedings and International Conference on Functional Programming (1997).

[8] Byung R. Moon et al.: *Toward More Powerful Recombinations.* Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kauffman (1995) 96–103.

[9] Pablo Moscato: *An Introduction to Population Approaches for Optimization and Hierarchical Objective Functions: A Discussion on the role of Tabu Search.* Annals of Operations Research **41** (1993) 85–121.

[10] Mike Paterson, Teresa Przytycka: *On the Complexity of String Folding.* Proceedings of the First International Conference on Computational molecular Biology (1997).

[11] Arnold L. Patton et al.: *A Standard GA Approach to Native Protein Conformation Prediction.* Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kauffman (1995) 574–581.

[12] D. Pelta, N. Krasnogor, P. Moscato: *Resultados de la Complejidad Computacional en el Problema de Replegado de Proteínas.* II Jornadas de Informática en Investigación Operativa, Montevideo (1995).

[13] N. Krasnogor, D. Pelta, Pablo Mocciola, Pablo E. Martinez Lopes, Esteban de la Canal: *Enhanced Evolutionary Search of Foldings Using Parsed Proteins.* Proceedings del Simposio en Investigación Operativa 97. 26 JAIIO, Buenos Aires, Argentina (1997).

[14] A. Sali, E. Shakhnovich, M. Karplus: *How Does a Protein Fold.* Nature **369** (1994) 248–251.

[15] H. Li, R. Helling, C. Tang, N. Wingreen: *Emergence of Preferred Structures in a Simple Model of Protein Folding.* Science **273** (1996) 666.

[16] M. Kardar: *Wich Came First, Protein Sequence or Structure?.* Science **273** (1996) 610.

[17] L. Holm, C. Sander: *Mapping the Protein Universe.* Science **273** (1996) 595.

[18] R. Unger, J. Moult: *Genetic Algorithms for Protein Folding Simulations.* Journal of molecular Biology **231** (1993) 75–81.

[19] R. Unger, J. Moult: *Finding the Lowest Free Energy Conformation of a Protein is an NP-hard Problem: Proof and Implications.* Center for Advanced Research in Biotechnology, University of Maryland (1993).

[20] R. Unger, J. Moult: *A Genetic Algorithm for 3D Protein Folding Simulations.* Proceedings of the fifth Annual International Conference on Genetic Algorithms (1993) 581–588.

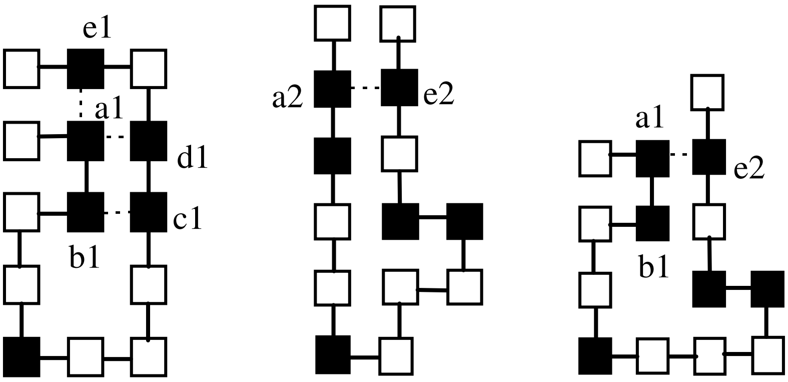


Figure 9: Ejemplo de 1−point crossover

Versión discreta del problema biespectral

Fernando Levstein

En estas notas haremos una breve exposición de lo que se conoce actualmente como *problema biespectral.*

Un problema clásico es el de recuperar la información contenida en una señal $f(t)$ de la cual conocemos su distribución de frecuencias $\hat{f}(\omega)$. Esto se resuelve mediante la Transformada de Fourier.

Sin embargo en la práctica, debido a que la señal se conoce sólo en un intervalo $[-T,T]$ y las frecuencias medibles están restringidas a un intervalo acotado $[\Omega,\Omega]$, lo que se necesita invertir es el operador

$$A : f \to \chi_{[-\Omega,\Omega]} \hat{f} \chi_{[-T,T]},$$

donde los operadores χ se refieren a multiplicar por la función característica del intervalo en cuestión.

El operador A no es inversible lo cual lleva a plantearse el problema de descomposición en valor singular, para lo cual es necesario calcular los autovalores y autovectores de $A^T A$. Este es un operador de núcleo y el problema es muy complicado aún numericamente.

Slepian, Landau y Pollak (ver [10]), obtuvieron una solución muy ingeniosa debida a un hecho inesperado: Encontraron un operador diferencial D de segundo orden de espectro simple, que conmuta con $A^T A$. Esto quiere decir que ambos operadores comparten autovectores, más aún los autovalores de D están espaciados lo cual facilita su cálculo.

A. Grunbaum, considero también algunos casos que le llevaron a pensar que la milagrosa existencia del operador D estaba ligada al hecho que las autofunciones de D son también autofunciones de otro operador **en la variable espectral**.

Teniendo en mente esto, es que logra clasificar junto con H. Duistermaat, ver [5], los operadores D, de segundo orden en la variable x, cuyas autofunciones (que dependen de x y λ), son también autofunciones de un operador B en el parámetro espectral λ.

$$Df(x,\lambda) = g(\lambda) f(x,\lambda)$$
$$Bf(x,\lambda) = h(x) f(x,\lambda)$$

La clasificación resultó estar relacionada con la familia de soluciones racionales de la ecuación de KdV que permanecen racionales por el flujo. Las soluciones se obtienen a partir de soluciones básicas aplicándoles sucesivamente una transformación de Darboux.

Estas transformaciones consisten en descomponer un operador como producto de dos operadores de orden menor y obtener un nuevo operador que resulta de permutar los factores.

$$A = BC \rightarrow \tilde{A} = CB$$

En 1993 G. Wilson se plantea el mismo problema, pero en lugar de considerar operadores toma álgebras de operadores que conmutan entre sí.

Esto le permite obtener una versión algebro-geométrica del resultado de Grunbaum y Duistermaat, cuando los órdenes de los operadores que forman una tal álgebra tienen MCD igual a 1, (rango 1).

En 1995, J. Liberati da una demostración del resultado de Wilson usando transformaciones de Darboux, lo cual le permite dar un ejemplo de cómo generalizar los resultados de Wilson al caso de rango 2.

Durante estos ultimos 3 años, Grunbaum y Haine (ver [6]), tratan el caso semi-discreto donde uno de los operadores es diferencial y el otro es de diferencias. Es así que recuperan el teorema de Bochner [2], sobre los polinomios ortogonales que satisfacen una ecuacion diferencial de segundo orden, como así también los polinomios de Krall [7], que satisfacen ecuaciones de orden cuatro.

La idea principal de estos resultados es que aplicando una transformación de Darboux a la matriz tridiagonal correspondiente a una familia de polinomios ortogonales clásicos se obtiene una matriz tridiagonal dependiente de un parámetro, ya que la descomposición no es única, que corresponde a una familia de polinomios ortogonales de Krall.

Junto a Laura Matusevich (ver [9]), consideramos el caso donde ambos operadores son discretos, lo cual permite plantear el problema en términos de matrices.

Ahora bien, el análogo del teorema de Bochner sobre polinomios que son autofunciones de un operador en diferencias y que considerados como función en la variable espectral son polinomios y son autofunciones de un operador en diferencias, fue resuelto por Leonard en 1984 [8], quien probó que tales polinomios son los llamados de q-Racah de Askey-Wilson (ver [1]).

No es difícil ver que si uno reemplaza “polinomios” por “funciones”, el teorema de Leonard sigue valiendo, es decir, no aparecen nuevas soluciones.

La pregunta que nos planteamos entonces fue:
¿ qué ocurre si aplicamos una transformación de Darboux a uno de los operadores?

La diferencia fundamental con los casos continuo y semidiscreto es que no aparecía ningún parámetro, por ser la descomposición única.

Lo primero que observamos fue que si uno efectuaba la descomposición $A = LU$ con L triangular inferior y U triangular superior y descartaba la última fila y última columna de la matriz $\tilde{A} = UL$ se obtenía la matriz correspondiente a una familia de polinomios de Lloyd-Wilson, los cuales son de q-Racah por un resultado de Delsarte (ver [3]). O sea que este tipo de transformación de Darboux, también llamada de Christoffel, manda una familia de n polinomios de q-Racah en otra familia de $n-1$ polinomios del mismo tipo.

Fue entonces que vislumbramos el parámetro buscado al considerar el conjunto de polinomios ortogonales que van a parar por medio de una transformación de Christoffel a una familia de q-Racah. Este conjunto depende de un único parámetro. En este caso la transformación, también puede interpretarse como el efecto de añadir una masa en el origen con un cierto peso, del cual depende el parámetro, (ver [4]).

Más aún, la transformación puede interpretarse en términos de descomposición de matrices de manera similar a la considerada por Grunbaum y Haine. Para ello, previamente extendemos nuestra matriz $n \times n$ a una matriz $(n+1) \times (n+1)$ añadiendo una última fila y una última columna de ceros. si llamamos $\tilde{A} = UL$ a la matriz extendida, la matriz $\hat{A} = LU$ corresponde a la familia monoparamétrica de polinomios. Esta transformación se conoce con el nombre de "dressing Darboux" o también como transformación de Geronimus.

Para llegar a una completa analogía con los casos tratados por Grunbaum faltaría probar que estos nuevos polinomios que se obtienen (ya no pertenecen a la clase de q-Racah, salvo para un valor del parámetro), satisfacen una ecuación en diferencias de orden cuatro. Esto se encuentra aún a nivel de conjetura.

Referencias

[1] R. Askey, J. Wilson: *A set of orthogonal Polynomials that generalize the Racah coefficients or* $6-j$ *symbols.* SIAM J. Math. Anal. **10(5)** (1979).

[2] S. Bochner: *Über Sturm-Liouvillesche Polynomesysteme.* Math. Z. **29** (1929) 730–736.

[3] L. Chihara: *On the zeros of Akey-Wilson polynomials, with applications to coding theory.* SIAM J. Math. Anal. **18(1)** (1987) 191–207.

[4] T. S. Chihara: *Orthogonal Polynomials and measures with end point masses.* Rocky Mountain Journal of Mathematics **15(3)** (1985).

[5] J. J. Duistermaat, F. A. Grünbaum: *Differential Equations in the Spectral Parameter.* Commun. Math. Physics. **103** (1986).

[6] F. A. Grunbaum, L. Haine: *Orthogonal Polynomials satisfying Differential Equations: The Role of the Darboux Transformation.* Center for Pure and Applied Mathematics, University of California, Berkeley (1994).

[7] H.L. Krall: *On orthogonal polynomials satisfying a certain fourth order differential equation.* Pennsylvania State College Studies **6** (1940).

[8] D. Leonard: *Orthogonal Polynomials, Duality and Association Schemes.* SIAM J. Math. Anal. **13(4)** (1982).

[9] F. Levstein, L. Matusevich: *Discrete version of the bispectral problem.* CRM Proceedings and Lecture Notes, AMS, por aparecer.

[10] D. Slepian, H. O. Pollak: *Prolate spheroidal wave functions, Fourier analysis and uncertainty I.* Bell Systems Tech. J. **40** (1961) 43–63.

Why geometric reasoning requires space?

Guillermo Matera

It is a general belief that minimizing memory space is not meaningful for computer algebra tasks. This belief is due to the results of [3] and [19] showing that every straight–line program can be evaluated using only a constant number of arithmetic registers.

However, there is somenthing wrong with that interpretation. Computer oriented algorithmics often requires the optimization of more than one resource at the same time. When one deals with memory space from a tradeoff point of view it turns out that "fast" algorithms are not necessarily efficient in the management of the memory space (see for example [1]).

The purpose of this talk is the discussion of the requirements of memory space for *real life* procedures dealing with multivariate polynomial elimination problems.

It is well known that algorithmic multivariate polynomial elimination faces serious space complexity problems. As it was shown by E. Mayr and A. Meyer in [18], *algebraic* elimination problems are practically *infeasible* since they are EXPSPACE–complete. This represents a major problem for the currently available software on symbolic computation (mainly based on the so–called Gröbner solving, see [4]), which has an algebraic conception and hence, fails when dealing with realistic sized problems. Quite often, out–of–memory errors arise during the execution of the programs.

Hence, only *geometric* elimination should be considered from a practical point of view. Positive results on the space complexity of geometric elimination have been obtained in the papers [17], [15] and [16], where *feasible* space bounds were shown, even considering the bit related aspects. However, the running time of these procedures turns out to have poor performance (however better than that of the Gröbner solving, see [6], [5], [13]), and this was confirmed by an *ad–hoc* implementation by a team work in the University of San Luis, Argentina (see [11]).

An immediate conclusion that should be drawn from this observation is that memory space cannot be regarded separately of running time, if the aim is the development of efficient procedures for *geometric* elimination able to be implemented on *real life* computers.

Recent *theoretical* work developed by the TERA group (see [10]) allows to tackle the mentioned inconvenients. Algorithmic procedures for geometric elimination with time–space complexity behaviour controlled by *geometric* (rather than *syntactic*) parameters were developed by the TERA group (see [2], [8], [7]).

In the present contribution we analyze, from a time–space tradeoff point of view, the algorithms devised in the cited papers. Applying a result of J. Ja'Ja' [12] we deduce that the algorithms in [2], [8], [7] cannot be implemented using *sublinear* space if a nonpolynomial growth of the time–space tradeoff complexity has to be avoided.

Then, following Schönhage's GOLDEN RULE NUMBER 1: *Do care about the constants!* [20], we comment on some algorithms for linear algebra tasks and an effective shape lemma adapted to the particular features of the procedures of [2], [8] and [7] which, combined with the techniques in the above cited papers, yield algorithms for geometric elimination using *linear* space and *quadratic* time.

References

[1] M. ALDAZ, J. HEINTZ, J.L. MONTAÑA, G. MATERA, L.M. PARDO: *Time–space trade–offs in algebraic complexity theory.* Submitted to Computational Complexity (1996).

[2] B. BANK, M. GIUSTI, J. HEINTZ, R. MANDEL, G. MBAKOP: *Polar Varieties and Efficient Real Equation Solving: The Hypersurface Case.* J. of Complexity **13 (1)** (1997) 5–27.

[3] M. BEN–OR, R. CLEVE: *Computing algebraic formulas using a constant number of registers.* Proceedings 20th Annual ACM Symposium on Theory of Computing (1988) 254–257. Also in SIAM Journal of Computing **21(1)** (1992) 54–58.

[4] B. BUCHBERGER: *Gröbner Bases: An algorithmic method in polynomial ideal theory.* In N. K. B. et al, ed., *Multidimensional System Theory*, Reidel, Dordrecht (1985) 374–383.

[5] T. W. DUBÉ: *The structure of polynomial ideals and Gröbner bases.* SIAM Journal on Computing **19 (4)** (1990) 750–775.

[6] M. GIUSTI: *Complexity of standard basis in projective dimension zero.* Lecture Notes in Comp. Science **378**, Springer–Verlag (1989) 333–335.

[7] M. GIUSTI, K. HÄGELE, J. HEINTZ, J. E. MORAIS, J. L. MONTAÑA, L. M. PARDO: *Lower Bounds for Diophantine Approximation.* In *Proceedings of MEGA'96*, Journal of Pure and Applied Algebra **117& 118** (1997) 277–317.

[8] M. GIUSTI, J. HEINTZ, J. E. MORAIS, J. MORGENSTERN, L. M. PARDO: *Straight–Line Programs In Geometric Elimination Theory.* To appear in J. of Pure and App. Algebra (1997) 1–46.

[9] M. GIUSTI, J. HEINTZ, J. E. MORAIS, L. M. PARDO: *When Polynomial Equation Systems can be solved fast?* In G. Cohen, H. Giusti, and T. Mora, eds., *Applied*

Algebra, Algebraic Algorithms and Error Correcting Codes, Proceedings AAECC-11, LNCS **948**, Springer (1995) 205–231.

[10] M. Giusti, J. Heintz, L. M. Pardo: *Proceedings of TERA'96.* Preprint 7/96 Dep. Matemáticas, Estadística y Computación. Universidad de Cantabria, Santander, `http://hilbert.matesco.unican.es/tera/` (1996).

[11] A. Grosso, N. Herrera, G. Matera, M. E. Stefanoni, J. M. Turull: *Un algoritmo para el cálculo del rango de matrices enteras en espacio polilogarítmico.* In *Proceedings 25th JAIIO* (1996).

[12] J. Ja'Ja': *Time–space tradeoffs for some algebraic problems.* Journal of the Association for Computing Machinery **30(3)** (1983) 657–667.

[13] T. Krick, A. Logar: *Membership Problems, Representation problems and the Computation of the Radical for one-dimensional Ideals.* In T. Mora and C. Traverso, eds., *Effective Methods in Algebraic Geometry*, Progress in Mathematics **94**, Birkhäuser (1991) 203–216.

[14] T. Krick, L. M. Pardo: *A Computational Method for Diophantine Approximation.* In L. González-Vega and T. Recio, eds., *Algorithms in Algebraic Geometry and Applications. Proceedings of MEGA'94*, Progress in Mathematics **143**, Birkhäuser Verlag (1996) 193–254.

[15] G. Matera: *Sobre la complejidad en espacio y tiempo de la eliminación geométrica.* Ph.D. thesis, Universidad de Buenos Aires, Argentina (1997).

[16] G. Matera: *Some probabilistic algorithms for geometric elimination.* Submitted to Journal of Symbolic Computation (1997).

[17] G. Matera, J. M. T. Torres: *The Space Complexity of Elimination: Upper Bounds.* In F.Cucker and M.Shub, eds., *Proceedings Foundations of Computational Mathematics (FOCM'97)*, Springer (1997) 267–276.

[18] E. Mayr, A. Meyer: *The complexity of the word problem for commutative semigroups.* Adv. in Math. **46** (1982) 305–329.

[19] C. Michaux: *Une remarque à propos des machines sur* $\mathbb{R}$ *introduites par Blum, Shub et Smale.* Comptes Rendus de l'Académie des Sciences de Paris **309(I)** (1989) 435–437.

[20] A. Schönhage, F. W. Grotefeld, E. Vetter: *Fast Algorithms: A Multitape Turing Machine Implementation.* B.I. Wissenschaftsverlag (1994).

On the simultaneous approximation of polynomial roots *(Extended abstract)*

Arcadio M. Niell

1. Introduction

We present some comments on the article by Hopkins, Marshall, Schmidt, and Zlobec [1] that analyzed several results concerning a powerful method of Newton type, originally due to Weierstrass in 1891 [2] and often rediscovered, for the simultaneous approximation of all roots of an algebraic polynomial. Some new results are also presented.

Given a (monic, for definiteness) polynomial $p(\lambda) = \prod_{i=1}^{n}(\lambda - r_i)$ with distinct real or complex roots, the method, basically a variant of Newton's, yields a sequence of approximations $x^{(k)}$ to the *root vector* $r = (r_1, r_2, \ldots, r_n)$ or one of its permutations σr. The iterations become ill-defined if any two components of some approximation coincide. However, the method is exceptionally robust, giving convergence for "almost all" initial approximations $x^{(0)}$, not just those close to the exact solution. Furthermore, it even works for polynomials with multiple roots, although with slower convergence.

A remarkable property of the method was demonstrated in [1], namely, that after the first iteration the sum of the components of the approximation to the root vector equals the actual sum of the roots.

2. Review of the method

In this section we summarize some basic definitions and results stated and proved in [1]. Some statements of a purely algebraic nature found here and in later sections are valid for all $n \in \mathbb{N}$, but convergence analysis makes sense only for $n \geq 2$.

The algorithm involves iteration of the function $G : \mathbb{C}^n \to \mathbb{C}^n$ defined by $y = G(x)$, where

$$y_i \overset{\text{def}}{=} x_i - \frac{p(x_i)}{\Pi_i(x)} = x_i - \frac{\prod_{j=1}^{n}(x_i - r_j)}{\prod_{\substack{j = 1 \\ j \neq i}}^{n} (x_i - x_j)} \qquad \text{for } i = 1, \ldots, n, \tag{1}$$

the domain of G being

$$\operatorname{dom}(G) = \{x \in \mathbb{C}^n : x_i \neq x_j \text{ if } i \neq j\}. \tag{2}$$

Let δ denote the smallest distance between roots:

$$\delta \overset{\text{def}}{=} \min\{|r_i - r_j|,\ i, j = 1, \ldots, n,\ i \neq j\} \tag{3}$$

It is an essential assumption that $\delta > 0$.

The study of local convergence in [1] shows that if

$$\left\| x^{(0)} - \mathbb{R} \right\| < \rho(n, \delta) \overset{\text{def}}{=} \frac{\sqrt[n-1]{4} - 1}{2\ \sqrt[n-1]{4} + 1}\, \delta \tag{4}$$

then the iterates $x^{(k)} = G(x^{(k-1)}) = G^k(x^{(0)})$ converge to r quadratically, and the asymptotic convergence rate satisfies

$$\overline{\lim}_{k \to \infty} \frac{\left\| x^{(k+1)} - \mathbb{R} \right\|}{\left\| x^{(k)} - \mathbb{R} \right\|^2} \leq \frac{n-1}{\delta}. \tag{5}$$

3. The sum of the roots: a direct proof

Theorem 1 *Let $p(\lambda)$ be a polynomial with roots $r_1, \ \ldots, \ r_n$, some of which may be repeated. Let $G(x)$ be the iteration function defined in Section 2. Then*

$$\operatorname{im}(G) \subset \left\{ x \in \mathbb{C}^n : \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} r_j \right\}.$$

The statement of this theorem has been taken from [1]. The following proof, however, is different. We start by establishing an algebraic identity.

Lemma 2 *Let $n \in \mathbb{N}, r \in \mathbb{C}^n$, and $x \in \mathbb{C}^n$ such that $x_i \neq x_j$ if $i \neq j$, but otherwise arbitrary. Then:*

$$\sum_{i=1}^{n} \frac{\prod_{j=1}^{n} (x_i - r_j)}{\prod_{\substack{j = 1 \\ j \neq i}}^{n} (x_i - x_j)} = \sum_{i=i}^{n} (x_i - r_i) \tag{6}$$

Proof of the Lemma 2 By induction on n and properties of the divided differences. ■

Proof of Theorem 1 Take the fundamental formula 1 defining the algorithm, sum on i from 1 to n, and apply Lemma 2. ■

Improved local convergence analysis

Using a simpler technique than in [1], we obtain some sharper bounds. The conclusion is that the original method converges as stated in 5 whenever

$$\left\| x^{(0)} - \mathrm{I\!R} \right\| < \tau(n,\delta) \overset{\mathrm{def}}{=} \frac{\sqrt[n-1]{2}-1}{2\,\sqrt[n-1]{2}-1}\,\delta \tag{7}$$

For $n > 2$, the "radius of convergence" is therefore larger than surmised in [1]. Indeed, the ratio $\tau(n,\delta)/\rho(n,\delta)$ does not depend on δ; as a function of n, it equals 1 for $n = 2$ and grows asymptotically to $3/2$ as $n \to \infty$.

Tabla 3 *Comparative values of the asserted radius of convergence.*

n	τ/δ	ρ/δ	τ/ρ
2	0.33333	0.33333	1.00000
4	0.17102	0.14070	1.21547
6	0.11461	0.08780	1.30538
8	0.08615	0.06370	1.35243
10	0.06901	0.04996	1.38122
12	0.05755	0.04109	1.40064
14	0.04936	0.03489	1.41461

4. A modification yielding faster convergence

The modification we propose follows a line of thought very well established in numerical linear algebra, that is, the one that leads from Jacobi's method to that of Gauss-Seidel. The idea is to use at every moment the latest computed components of the approximate solution vector in order to compute the next component, rather than using the "old" approximate solution vector to compute the entire "new" vector.

The following formula defines the iteration stage of order k for the modified algorithm:

$$x_i^{(k+1)} = x_i^{(k)} - \frac{\prod_{j=1}^{n}(x_i^{(k)} - r_j)}{\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k+1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - x_j^{(k)})} \quad (i = 1, \ \ldots, n) \tag{8}$$

The main convergence result (with its wording mostly paraphrasing that of [1]) is the following:

Theorem 4 *Suppose that $p(\lambda)$ has distinct roots and that, in the* **modified** *Weierstrass method defined by 8, the initial approximation $x^{(0)}$ is chosen sufficiently close to a root vector r or, more specifically, that*

$$\left\| x^{(0)} - \mathrm{I\!R} \right\| < \tau(n, \delta) \tag{9}$$

Then

$$\lim_{k\to\infty} x^{(k)} = r \tag{10}$$

and

$$\overline{\lim}_{k\to\infty} \frac{\left\| x^{(k+1)} - \mathrm{I\!R} \right\|}{\left\| x^{(k)} - \mathrm{I\!R} \right\|^t} \leq \frac{1}{\delta^{t-1}} \tag{11}$$

where t is the positive root of the equation

$$t = (t-1)^n, \tag{12}$$

implying $t > 2$.

Remark 5 *According to the last assertion, local convergence is super-quadratic. However, this wording should be interpreted in a broad context that takes into account the number of operations, considering that the parallelism of the original method is lost, and thus avoiding unsubstantiated claims.*

Remark 6 *Loss of parallelism is not the only price to be paid for the improved rate of convergence of the modified method: the "sum of roots" property of the original method (see Theorem 1) is lost, too.*

Tabla 7 *Order of local convergence of the modified method.*

n	*t*	*n*	*t*
2	*2.61803*	*9*	*2.08507*
3	*2.32472*	*10*	*2.07577*
4	*2.22074*	*11*	*2.06830*
5	*2.16730*	*12*	*2.06217*
6	*2.13472*	*13*	*2.05705*
7	*2.11278*	*14*	*2.05271*
8	*2.09698*	*15*	*2.04898*

Remark 8 *The preceding results show that the modified method compares favorably with the original method in what regards* **local** *convergence. The theoretical study of* **global** *convergence is much more involved and still leaves many open questions. It has been briefly touched upon in [1] with reference to the original method. Numerical experiments seem to indicate that the proposed modification yields faster global convergence in most cases, but not all. This is reminiscent of what happens in the field of numerical linear algebra, as mentioned at the beginning of the present section: Gauss-Seidel usually beats Jacobi, but not always.*

Moreover, the modified method converges in some cases for which the original method does not converge at all.

5. Conclusions

On the basis of the presentation and the analysis contained in [1], we have achieved some new results concerning the Weierstrass method for the simultaneous approximation of polynomial roots:

1) a more straightforward proof of the “sum of the roots” property (no Vandermonde determinants or Taylor series expansions are used);

2) a simplification of the instrumental lemma used in the analysis of local convergence;

3) an improvement in the local convergence results: the “radius of convergence” (tolerance on the error of the initial approximation) is shown to be larger, in general, than previously proved;

4) a modification of the method giving faster convergence, with a computer program implementing both methods in parallel, for ease of comparison;

5) the local convergence analysis of the modified method.

6. References

[1] M. Hopkins, B. Marshall, G. Schmidt, S. Zlobec: *On a Method of Weierstraß for the Simultaneous Calculation of the Roots of a Polynomial.* ZAMM (Zeitschrift für angewandte Mathematik und Mechanik) **74(8)** (1994) 295–306.

[2] K. Weierstrass: *Neuer Beweis des Satzes, daß jede ganze rationale Funktion einer Veränderlichen dargestellt werden kann als ein Produkt aus linearen Funktionen derselben Veränderlichen.* Gesammelte Werke **3** (1903) 251–269 (Johnson Reprint Comp., New York 1967).

Eliminación de cuantificadores y straight–line programs

Susana Puddu, Juan Sabia

Introducción

El objetivo de esta charla es comentar un algoritmo de eliminación de cuantificadores sobre cuerpos algebraicamente cerrados que utiliza como estructura de datos la codificación de polinomios por medio de programas de evaluación (*straight-line programs*) (ver [9]). Para esto primero ubicaremos el problema en el contexto del Algebra Conmutativa y la Geometría Algebraica Computacionales, después explicaremos por qué esta estructura de datos puede mejorar las complejidades conocidas y finalmente enunciaremos el resultado que obtuvimos, dando una idea de la demostración.

La eliminación de cuantificadores sobre cuerpos algebraicamente cerrados

Sea k un cuerpo arbitrario y sea K un cuerpo algebraicamente cerrado que lo contiene (por ejemplo, sea $k = \mathbb{Q} \subset K = \mathbb{C}$). Vamos a notar con $\mathcal{L}(k)$ al lenguaje de primer orden de la teoría de los cuerpos algebraicamente cerrados con constantes en k. Para ser más claros, podemos decir que $\mathcal{L}(k)$ es el conjunto de todas las fórmulas que pueden ser escritas como combinaciones booleanas ($\wedge$, $\vee$ y $\neg$) de polinomios a coeficientes en k y variables en K igualados a cero y cuyas variables pueden estar cuantificadas por $\forall$ y por $\exists$. Cabe destacar que en estas fórmulas no pueden aparecer ni subconjuntos ni relaciones cuantificados.

Un hecho conocido es que la teoría de primer orden de los cuerpos algebraicamente cerrados admite eliminación de cuantificadores. Es decir, que toda fórmula φ con $Y_1, \dots, Y_n$ variables libres y $X_1, \dots, X_m$ variables ligadas (cuantificadas) admite una fórmula ψ equivalente sin cuantificadores (donde equivalente significa que los conjuntos

$$\{(y_1, \dots, y_n) \in K^n \ / \ \varphi(y_1, \dots, y_n)\} \quad \text{y} \quad \{(y_1, \dots, y_n) \in K^n \ / \ \psi(y_1, \dots, y_n)\}$$

son iguales.)

Ejemplos

i) Consideremos la fórmula

$$\varphi := (\exists X)(Y_1.X^2 + Y_2.X + Y_3 = 0) \wedge (Y_1 \neq 0) \wedge (2Y_1.X + Y_2 \neq 0)$$

La existencia de dicho X es equivalente a pedir que el polinomio que aparece sea exactamente de grado dos y que no tenga raíces múltiples. Por lo tanto la fórmula φ es equivalente a la fórmula

$$\psi := (Y_1 \neq 0) \wedge (Y_2^2 - 4Y_1.Y_3 \neq 0)$$

que es libre de cuantificadores.

ii) Consideremos ahora una matriz $A \in k^{n \times n}$ y un vector $B \in k^n$. Es claro que la fórmula

$$\varphi := (\exists X_1, \ldots, X_n) \;\; A. \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix} = B$$

aunque esté escrita en notación matricial es una fórmula de $\mathcal{L}(k)$. Pero decidir la existencia del vector $(X_1, \ldots, X_n)$ solución del sistema es equivalente a la siguiente fórmula

$$\psi := rg(A) = rg(A|B)$$

que iguala el rango de la matriz A con el de la matriz ampliada y que es libre de cuantificadores. Este ejemplo va a volver a mencionarse en el caso de la eliminación efectiva de cuantificadores.

Un caso especial. El teorema de los ceros de Hilbert

Una fórmula de particular importancia en la eliminación es la siguiente:

$$\varphi := (\exists X_1, \ldots, X_n)(f_1(X_1, \ldots, X_n) = 0 \wedge f_2(X_1, \ldots, X_n) = 0 \wedge \ldots f_s(X_1, \ldots, X_n) = 0)$$

donde $f_1, \ldots, f_s \in k[X_1, X_2, \ldots, X_n]$ son polinomios en n variables a coeficientes en k. El teorema de los ceros de Hilbert afirma que el conjunto descripto por φ sobre K^n es vacío si y sólo si existen $p_1, \ldots, p_s \in k[X_1, \ldots, X_n]$ tales que $1 = p_1.f_1 + \ldots + p_s.f_s$.

Este resultado ya era conocido por Kronecker que fue el primero en mostrar una eliminación efectiva de cuantificadores aunque la formulación del problema desde el punto de vista de la lógica se debe a Tarski [10] que lo resolvió sobre cuerpos algebraicamente cerrados.

El conocimiento de una cota *a priori* de los grados de los polinomios $p_i (1 \leq i \leq s)$ en función de la cantidad de variables y de los grados de los polinomios $f_i (1 \leq i \leq s)$ permitiría traducir esta fórmula a la condición de existencia de solución para un sistema lineal como en el ejemplo ii) anterior y poder resolverlo efectivamente.

El primer planteo sobre la efectividad de los algoritmos lo hizo Collins en 1975 en el caso real mientras que Wüthrich y Heintz [7,11] plantearon el problema para cuerpos algebraicamente cerrados y notaron que era esencialmente distinto que para los cuerpos reales. Sucesivos trabajos de Brownawell [1], Caniglia-Galligo-Heintz [2] y Kollár [8] demostraron que las cotas óptimas para los grados de los polinomios $p_i(1 \leq i \leq s)$ que aparecen en la escritura del 1 son del orden de d^n donde $d = \max\{3, \deg f_i(1 \leq i \leq s)\}$.

La complejidad de los algoritmos

Si consideramos los algoritmos como grafos acíclicos no orientados, una medida posible de complejidad es la cantidad de nodos del grafo. Una posible forma de codificar polinomios en n variables para que aparezcan como datos de entrada o de salida de un algoritmo es ordenar los distintos monomios de los polinomios y dar los coeficientes según ese orden preestablecido. Es decir, cada polinomio será un vector de números, cada uno representando un coeficiente del polinomio. De esta manera, para representar un polinomio de grado d en n variables con $d > n$, se necesitarán del orden de d^n monomios, y por lo tanto, las cotas obtenidas por las versiones efectivas del teorema de los ceros de Hilbert serán del orden de d^{n^2} como lo mostraron Chistov- Grigor'ev [3] (con un algoritmo no paralelizable que depende de la aritmética del cuerpo en cuestión) y Fitchas-Galligo-Morgenstern [5] (con un algoritmo bien paralelizable y que no depende del cuerpo en cuestión).

Veamos que este orden de complejidad es intrínseco al problema de la eliminación (es decir, no depende de la aplicación del teorema de los ceros) si utilizamos la codificación de polinomios como vectores de coeficientes:

Ejemplo

Sean d, r números naturales tales que $d \geq r$ y sea φ la fórmula:

$$\begin{gathered} \exists X_1 \exists X_2 \ldots \exists X_{r-1} : {X_1}^d . Y_1 - 1 = 0 \wedge {X_2}^d . Y_2 - X_1 = 0 \wedge \\ \wedge {X_3}^d . \ Y_3 - X_2 = 0 \wedge \ldots \wedge {X_{r-1}}^d . Y_{r-1} - X_{r-2} = 0 \wedge {Y_r}^d - X_{r-1} = 0 \end{gathered}$$

En este caso hay r variables libres y $r-1$ variables ligadas y aparecen r polinomios cuyos grados están acotados por $d+1$.

Es fácil ver que φ es equivalente a la siguiente fórmula sin cuantificadores:

$${Y_r}^{d^r} . {Y_{r-1}}^{d^{r-2}} . {Y_{r-2}}^{d^{r-3}} \ldots {Y_2}^d . Y_1 - 1 = 0$$

También puede probarse que, como este último polinomio es irreducible, *cualquier* fórmula libre de cuantificadores equivalente a φ debe involucrarlo y, por lo tanto, la cantidad de números necesarios para codificarlo (del orden de d^{r^2} si pensamos en un vector de coeficientes) es una cota inferior para la complejidad de *cualquier* algoritmo de eliminación.

Al llegar a este punto, se hace evidente que la escritura de un polinomio como vector de sus coeficientes es demasiado "cara" a nivel algorítmico. Pero, ¿por qué es necesario tanto espacio para codificar este polinomio si nosotros pudimos escribirlo en un renglón? Esta idea da origen a otros dos tipos de escritura posibles para polinomios:

i) La escritura rala: cuando un polinomio tiene muchos coeficientes nulos, solo basta codificar los coeficientes no nulos indicando a que monomio acompañan. Esta codificación es esencialmente mala para los algoritmos de Algebra Conmutativa y Geometría Algebraica porque no soporta cambios de variables. Si bien el polinomio X^{100} es fácil de codificar por tener muchos ceros, el polinomio $(X+Y)^{100}$ obtenido del anterior por un simple cambio de variables, no tiene ningún coeficiente nulo.

ii) Los *straight-line programs* (slp): un polinomio puede ser dado por un programa que lo evalúe sin comparaciones ni ramas. Para codificar el polinomio $(X+Y)^{100}$, podríamos presentar las variables y operarlas obteniendo así el siguiente slp:

$$X \rightarrow Y \rightarrow X+Y \rightarrow\rightarrow (X+Y)^2 \rightarrow (X+Y)^4 \rightarrow \ldots \rightarrow (X+Y)^{100}$$

(Notar que este último slp usa sólo del orden de $\log_2(100)$ nodos y que un vector de coeficientes usaría del orden de los 100^2 nodos.)

El resultado

Basándonos principalmente en el algoritmo para calcular la dimensión de una variedad algebraica de Giusti-Heintz [6] obtuvimos este primer resultado que, en base al cambio de codificación de los polinomios, mejora las complejidades conocidas hasta ahora en algoritmos de eliminación:

Sean $1 \leq m \leq n$ *números naturales y sean* $f_1, \ldots, f_s \in k[X_1, \ldots, X_n]$ *polinomios cuyos grados totales en las últimas* m *variables están acotados por un número natural* $d \geq m$ *y cuyos grados totales en las primeras* $n-m$ *variables están acotados por un número natural* d'. *Se supone que los polinomios* $f_1, \ldots, f_s$ *están codificados en forma densa (como vector de coeficientes) en las últimas* m *variables y que sus coeficientes (polinomios en las* $n-m$ *primeras variables) están dados por un slp de longitud* L. *Sea*

$$\mathcal{P} = \{(x_1, \ldots, x_{n-m}) \in K^{n-m} / \exists (\quad x_{n-m+1}, \ldots, x_n) \in K^m : \\ f_1(x_1, \ldots, x_n) = 0 \wedge \ldots \wedge f_s(x_1, \ldots, x_n) = 0\}$$

Entonces, a menos de una preparación previa, existe un algoritmo bien paralelizable y sin divisiones de complejidad secuencial $L + s^{O(1)} d^{O(m)}$, *que permite describir al* *conjunto $\mathcal{P}$ por medio de una fórmula ψ libre de cuantificadores de longitud $L + s^{O(1)}d^{O(m)}$. En ella aparecen a los sumo $s^{O(1)}d^{O(m)}$ polinomios con grados acotados por $d'.d^{O(m)}$ dados por un slp de longitud $L + s^{O(1)}d^{O(m)}$*

(Notar que, en el ejemplo anterior, este algoritmo da una complejidad secuencial del orden de d^r, mejorando por lo tanto cualquier algoritmo que utilice exclusivamente la representación densa de polinomios.)

La idea simplificada de nuestro algoritmo es considerar cada punto de K^{n-m} como fijo y tratar de calcular formalmente la dimensión de la variedad (vía Giusti- Heintz [6]) que definen los polinomios $f_i(1 \leq i \leq s)$ en ese caso, intersecando con hiperplanos genéricos. Esto da origen a una ramificación del algoritmo dependiendo de ciertas condiciones algebraicas que cumpla el punto en cuestión y eso se traduce a condiciones donde las variables a eliminar no aparecen.

También obtuvimos un algoritmo de eliminación en el caso de que aparezcan desigualdades. En este caso, mostramos que el truco de Rabinowicz no es conveniente (en el sentido de que empeora la complejidad) y obtuvimos cotas que discriminan cada vez más los parámetros en cuestión:

Sean $1 \leq m \leq n$ números naturales y sean $f_1, \dots, f_s \in k[X_1, \dots, X_n]$ polinomios cuyos grados totales en las últimas m variables están acotados por un número natural $d \geq m$ y cuyos grados totales en las primeras $n - m$ variables están acotados por un número natural d'. Sean $g_1, \dots, g_{s'} \in k[X_1, \dots, X_n]$ polinomios cuyos grados totales en las últimas m variables están acotados por un número natural $\delta \geq m$ y cuyos grados totales en las primeras $n - m$ variables están acotados por un número natural δ'. Se supone que los polinomios $f_1, \dots, f_s, g_1, \dots, g_{s'}$ están dados por un slp de longitud L. Sea

$$\begin{array}{ll} \mathcal{P} = \{(x_1, \dots, x_{n-m}) \in K^{n-m} / & \exists\, (x_{n-m+1}, \dots, x_n) \in K^m : \\ & f_1(x_1, \dots, x_n) = 0 \ \wedge \dots \wedge \ f_s(x_1, \dots, x_n) = 0 \ \wedge \\ \wedge\, g_1(x_1, \dots, x_n &) \neq 0 \ \wedge \dots \wedge \ g_{s'}(x_1, \dots, x_n) \neq 0\} \end{array}$$

Entonces, a menos de una preparación previa, existe un algoritmo bien paralelizable y sin divisiones de complejidad secuencial $L^2 + (ss'\delta)^{O(1)}d^{O(m)}$, que permite describir al conjunto $\mathcal{P}$ por medio de una fórmula ψ libre de cuantificadores de longitud $L^2 + (ss'\delta)^{O(1)}d^{O(m)}$.
En ella aparecen a los sumo $L^2 + (ss'\delta)^{O(1)}d^{O(m)}$ polinomios con grados acotados por $\delta' d' (s'\delta)^{O(1)}d^{O(m)}$ dados por un slp de longitud $L^2 + (ss'\delta)^{O(1)}d^{O(m)}$

Iterando este algoritmo se encontró uno que elimina cuantificadores de cualquier fórmula prenexa, cuya complejidad también mejora las de los algoritmos conocidos.

Para terminar, cabe destacar que los algoritmos obtenidos no son uniformes, en el sentido de que la construcción de los mismos depende de la elección de conjuntos de vectores y que efectuar dicha elección excede las complejidades obtenidas. Sin embargo, pueden ser considerados probabilísticamente uniformes eligiendo aleatoriamente los conjuntos de vectores en cuestión con baja probabilidad de error.

Bibliografía

[1] D. Brownawell: *Bounds for the degrees in the Nullstellensatz.* Ann. Math. Second Series **126(3)** (1987) 577–591.

[2] L. Caniglia, A. Galligo, J. Heintz: *Borne simple exponentielle pour les degrés dans le théorème des zéros sur un corps de caractéristique quelconque.* C.R. Acad. Sci. Paris **307(I)** (1988) 255–258.

[3] A. L. Chistov, D. Yu. Grigor'ev: *Complexity of quantifier elimination in the theory of algebraically closed fields.* Proc. 11th Symp. MFCS 1984, Springer LN Comp. Sci. **176** (1984) 17–31.

[4] G. E. Collins: *Quantifier elimination for real closed fields by cylindrical algebraic decomposition.* Lecture Notes in Computer Science **33** Springer, Berlin (1975) 134–183.

[5] N. Fitchas, A. Galligo, J. Morgenstern: *Precise sequential and parallel complexity bounds for quantifier elimination over algebraically closed fields.* Journal of Pure and Applied Algebra **67** (1990) 1–14.

[6] M. Giusti, J. Heintz: *La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial.* Computational Algebraic Geometry and Commutative Algebra, Proceedings of the Cortona Conference on Computational Algebraic Geometry and Commutative Algebra, D. Eiseinbud, L. Robbiano, eds., Symposia Matematica vol. XXXIV, Instituto Nazionale di Alta Matematica, Cambridge University Press (1993).

[7] J. Heintz, R. Wüthrich: *An efficient quantifier elimination algorithm for algebraically closed fields.* SIGSAM Bull. **9** (1975) 11.

[8] J. Kollar: *Sharp effective Nullstellensatz.* J. AMS **1** (1988) 963–975.

[9] S. Puddu, J. Sabia: *An effective algorithm for quantifier elimination over algebraically closed fields using straight line programs.* Aparecerá en Journal of Pure and Applied Algebra (1997).

[10] A. Tarski: *A decision method for elementary algebra and geometry.* 2nd ed. Univ. of California Press (1951).

[11] R. Wüthrich: *Ein Quantoreneliminationverfahren für die Theorie der algebraisch abgeschlossenen Körper.* Ph.D. Tesis, University of Zurich (1977).

The Decomposition Method applied to Life Sciences

M.J. Pujol, P. Grimalt

1. Introduction

Mathematical modeling is fundamental for the understanding of complex phenomenon coming from life sciences. For instance the medical treatments in cancerology and for the AIDS disease could be greatly benefit of mathematical developments for determining the optimal way for delivering chemical drugs in order to decrease tumors or quantity of viruses.

Almost all systems coming from life sciences are modeled by differential, partial differential and integral equations. Most of these functional equations are non linear and for solving them we need performing mathematical or numerical methods. In the following we shall present an original method for solving nonlinear functional equations. This technique called decomposition method uses a decomposition of the nonlinear operator and special kinds of polynomials (called Adomian polynomials)

Many important works have been done in MEDIMAT laboratory (Pierre et Marie Curie University Paris VI in France) for proving its convergence and for giving practical formulae to calculate the Adomian polynomials. We can specially quote the works of K. Abbaoui, Y. Cherrault, S Guellal, P. Grimalt and V. Seng. (see references).

2. The Decomposition Method

Let us give the basic ideas of this crafty method first introduced by George Adomian about 1980 (Georgia University). Let us consider the general nonlinear functional equation written in the canonical from:

$$y - N(y) = f \tag{1}$$

where N, the nonlinear operator (spaces Banach, Hilbert) and the function f are given and where y is unknown. The resolution of (1) is obtained by looking for a solution under a series form:

$$y = \sum_{i=0}^{\infty} y_i \tag{2}$$

and by decomposing the nonlinear operators as a series:

$$N(y)=\sum_{i=0}^{\infty}A_i \tag{3}$$

where the A_i's are the Adomian polynomials obtained from the relationship:

$$A_n=\frac{1}{n!}\frac{d^n}{d\lambda^n}\left(N\left(\sum_{i=0}^{\infty}\lambda^i y_i\right)\right)_{\lambda=0} \qquad n=0,1,2,\dots \tag{4}$$

Putting (2) and (3) in (1) leads to:

$$\sum_{i=0}^{\infty}y_i-\sum_{i=0}^{\infty}A_i=f \tag{5}$$

and the Adomian method consists in calculating the series terms y_i by means of the formulae:

$$y_0=f, \qquad y_1=A_0, \qquad \dots \qquad y_i=A_{i-1}, \qquad \dots \tag{6}$$

Each y_i is uniquely defined by (6) because the Adomian polynomials A_i are calculable by means of formulae (4) and depend only on $y_0, y_1, \dots, y_i$.

The convergence of the method has mainly been developed by Y. Cherrault and K. Abbaoui (see references). The convergence can be proved:

- by using the fixed point theorem after giving another formulation of the method adapted to a mathematical study.
- under some hypotheses on $N(u_0)$ and its Frechet derivatives $N^k(u_0), k=1,\dots$ (to see reference [4], [8]).

Generally it is not possible to calculate the sum of the series (2) and thus we shall only obtain an approximated solution of the equation under the form of a truncated series

$$\phi_h=\sum_{i=0}^{h-1}$$

with some reasonable hypothesis on N, K. Abbaoui has found a bound of the error $||y-\phi_h||$ in function on h.

The basic formulae for finding the Adomian polynomials are not easy to use in practice when we have complex nonlinearities. Equivalent and recurrent formulae have been proposed (K. Abbaoui). They permit to calculate the Adomian polynomials by using formal calculus software.

By last, the solution obtained by Adomian method explicitly depend on space and time variables. No discretitation or linearitation is necessary.

3. Application of the Life Sciences

Many applications have been shown for some aspects related to the life sciences. We can enumerate some examples in Biology and Medicine:

- The treatment for certain allergies (Degranulation Basophiles).
- Models of Ecology (Lotka–Voleterra).
- The identification of parameters.
- Optimal control in biological systems (Respiratory System).
- Equations of enzymes (Biochemistry) and the Michaelis–Menten equations.

We can give many more examples in Physics, Economy, Molecular Biology, etc., but we will talk specifically about these applications in Immunology (Marchuk's Models).

4. Marchuk's Model

The knowledge of the new ideas in Immunology of the immunitorial answer (the organism's defense if there are bacterial or virulent attacks) is necessary for the comprehension of illness processes and ways to obtain appropriate treatments. The observation about the development of infections illness has made it possible to obtain significant results about the process of interaction "antigens–antibodies" in the genetic dominion. As a result of this, the mathematical models of immunitarial process can be formed.

5. Simple model of an illness

Around 1979 based on reasonable hypothesis of the elements that intervene in an infections illness, $V(t)$ concentration of virus, $F(t)$ concentration of antibodies, $C(t)$ concentration of plasma cells, and $m(t)$ characteristics of the injured organ, and considering each one in different parts, Marchuk (U.R.S.S. Science Academy) found a non–linear system of differential equations (to see reference [13]):

$$\begin{aligned}
\frac{dV}{dt} &= (\beta - \gamma F)V \\
\frac{dC}{dt} &= \alpha V(t-z)F(t-z) - u_0(C - C^*) \\
\frac{dF}{dt} &= \rho C - (u_j + \eta r V)F \\
\frac{dm}{dt} &= \sigma V - u_m m
\end{aligned}$$

with initial conditions $t = t_0$, or $[t_0 - z, t_0]$ the organism has not got any virus $V(t) = 0$. For $t < t_0$, the conditions will be fixed by the t_0, s_0, $V(t_0) - V_0$, $C(t_0) = C_0$, $F(t_0) = F_0$, $m(t_0) = m_0$.

The system is resolute by the decomposition method considering the variable of the models as an analytic function, the initial conditions are non–negatives, all the parameters $(\beta, \gamma, \alpha, u_0, u_m, \ldots)$ are positive and any results as [4], [5], [11] must be taken for differential equation systems, obtaining a good approximation using MAPLE V without using the classic methods of linearitation and discretitation.

The results we get generalize the results given by G. Adomian [3] and establish a connection between mathematics and immunology to take into account other more complex models.

6. References

[1] G. Adomian: *A review of the decomposition method and some results for nonlinear equations.* Mathematical and Computer Modelling **13(7)** (1996) 17–43.

[2] G. Adomian: *Solving frontier problems of Physics: The decomposition method.* Kluwer (1994).

[3] G. Adomian, G.E. Adomian: *Solution of the Marchuk model of infections disease and immune response.* Mathematical and Computer Modelling (1996) 803–807.

[4] K. Abbaoui, Y. Cherrault: *Convergence of Adomian's method applied to differential equations.* Mathematical and Computer Modelling **28(5)** (1994) 103–109.

[5] K. Abbaoui, Y. Cherrault, M. N'Dour: *The decomposition method applied to differential equation systems.* Kybernetes **24(8)** (1995) 32–40.

[6] Y. Cherrault: *Convergence of Adomian's method.* Kybernetes **18(2)** (1989) 31–38.

[7] Y. Cherrault, G. Adomian: *Decomposition method: a new proof of convergence.* Mathematical and Computer Modelling **18(12)** (1993) 103–106.

[8] Y. Cherrault, G. Adomian, K. Abbaoui, R. Rach *Further remarks on convergence of decomposition.* I.J.B.C. **38** (1995) 89–93.

[9] K. Abbaoui, Y. Cherrault, V. Seng: *Practical formulae for the calculus of multivariate Adomian's polynomials.* Mathematical and Computer Modelling **22(1)** (1995) 89–93.

[10] S. Guellal, Y. Cherrault: *Practical formulae for calculation of Adomian's polynomials and application to the convergence of the decomposition method.* I.J.B.C. **36** (1994) 223–228.

[11] S. Guellal, P. Grimalt, Y. Cherrault: *Numerical study of Lorenz's equations by Adomian's method.* Computer Mathemat. Applied **33(3)** (1997) 25–29.

[12] P. Grimalt: *El método de descomposición y sus aplicaciones a problemas biomédicos.* Revista Ciencias Canaria, Volumen II (1995).

[13] G.I. Marchuk: *Mathematical model in immunology.* 2nd. ed. Optimization Software Inc. Publication Division New York (1987).

Twisted Chow Forms and Intrinsic Elimination Theory

J. Maurice Rojas[1]

Generalized resultants are reemerging as an elegant and efficient means of solving polynomial systems. We will briefly illustrate this through some extensions of the sparse resultant, and the classical Chow form, which dramatically improve prior elimination theoretic algorithms. Our notation follows that of [Roj97a, Section 6.3] and [Roj97c].

Since we will deal with polynomial systems which are sparse in the "specified supports" sense, the classical Chow form proves unreliable for polynomial system solving.[2] Hence we will need to introduce the following generalization of the u-resultant: **Definition 1** *Let* F *be an* $n \times n$ *polynomial system with constant coefficients and* $\mathrm{Supp(F)} = \mathrm{E}$. *Assume further that* E *is such that a generic system* G *with* $\mathrm{Supp(G)} \subseteq \mathrm{E}$ *has positively (and finitely) many roots. Call a facet of a polytope in* $\mathbb{R}^n$ **bad** *iff it has an inner normal* **outside** *the nonnegative orthant. Finally, suppose* $A \subset (\mathbb{N} \cup \{0\})^n$ *satisfies the following properties:*

- *i)* $\mathbf{O} \in A$ *and* $|A| \geq n+1$
- *ii) The convex hull* $\mathrm{Conv(A)}$ *is such that the cone generated by the edges emanating from the vertex* $\mathbf{O}$ *is precisely the nonnegative orthant in* $\mathbb{R}^n$.
- *iii) Every bad facet of* $\mathrm{Conv(A)}$ *is parallel to some facet of the Minkowski sum* $P := \sum_{i=1}^{n} \mathrm{Conv(E_i)}$.

We then call the polynomial $\mathrm{Ch_A(u)} := \mathrm{AffRes_{(E,A)}(F, \sum_{e \in A} u_e x^e)}$ *an* **(affine) twisted Chow form** *of* F. The notation $\mathrm{AffRes}_\star(\cdot)$ refers to the **affine (sparse) resultant** [Roj97c]. The affine resultant extends the benefits the sparse resultant [Stu97] provides (for algebraic geometry in the algebraic torus) to affine space minus an arbitrary union of coordinate hyperplanes, and can be computed quite efficiently.

As an application of the above construction, we give an explicit formula for the degree (and a lower bound for the **geometric** degree) of a zero-dimensional intersection of n algebraic hypersurfaces in K^n, for any algebraically closed field K. **Main Theorem [Roj97c]** *Following the notation of our last definition...*

---

[1] This research was completed at MIT and partially funded by an NSF Mathematical Sciences Postdoctoral Research Fellowship.

[2] There are many simple examples [Roj97b] where the u-resultant [Can86] (and even the **sparse** u-resultant) vanishes identically for a given nontrivial family of $n \times n$ sparse polynomial systems.

i) If $\mathrm{Ch}_{\mathrm{A}}(\mathrm{u})$ *is not identically zero, then the number of roots of* F *in* K^n*, counting multiplicities, is precisely the degree of* $\mathrm{Ch}_{\mathrm{A}}(\mathrm{u})$ *as a polynomial in* $u_{\mathbf{O}}$*.*

ii) Let $\mathrm{Ch}'_{\mathrm{A}}(\mathrm{u})$ *be the square-free part of* $\mathrm{Ch}_{\mathrm{A}}(\mathrm{u})$*. Then if* $\mathrm{Ch}'_{\mathrm{A}}(\mathrm{u})$ *is not identically zero, the number of distinct roots of* F *in* K^n *is at least the degree of* $\mathrm{Ch}'_{\mathrm{A}}(\mathrm{u})$ *as a polynomial in* $u_{\mathbf{O}}$*.*

iii) Suppose further that E *is fixed so that a generic* G *with* $\mathrm{Supp}(\mathrm{G}) \subseteq \mathrm{E}$ *has only finitely many roots in* K^n*. Then the* G *for which* $\mathrm{Ch}_{\mathrm{A}}(\mathrm{u})$ *vanishes identically form a subvariety of codimension* ≥ 2*.*

iv) Suppose further that $n = 2$*. Then* $\mathrm{Ch}_{\mathrm{A}}(\mathrm{u})$ *vanishes identically* $\Longleftrightarrow$ F *has infinitely many roots in* K^2*.*

The nonvanishing of $\mathrm{Ch}_{\mathrm{A}}(\mathrm{u})$ is a very weak assumption and still permits F to have infinitely many roots at toric infinity [Roj97c]. Assertion (3) thus tells us that our above root counting method will fail very rarely. Also, by applying the **toric GCP** [Roj97b], one can **always** efficiently and accurately count roots in K^n, even when Ch_{A} vanishes identically [Roj97c].

Twisted Chow forms can also be used to reduce polynomial system solving to a large eigenvalue problem [Roj97c]. This removes virtually all nonsingularity assumptions from an earlier construction (using the sparse u-resultant) from [Emi94]. In particular, prior applications of the sparse u-resultant never specified how to find roots lying on coordinate hyperplanes.

References

[Can86] John F. Canny: *The Complexity of Robot Motion Planning Problems.* ACM Doctoral Dissertation Award Series, ACM Press (1986).

[Emi94] Ioannis Z. Emiris: *Sparse Elimination and Applications in Kinematics.* Ph.D. dissertation, Computer Science Division, U. C. Berkeley (December, 1994), available on-line at `http://www.inria.fr/safir/SAFIR/Ioannis.html`.

[Roj97a] J. Maurice Rojas: *Toric Intersection Theory for Affine Root Counting.* Journal of Pure and Applied Algebra, to appear (1997).

[Roj97b] ________________: *Twisted Chow Forms and Toric Perturbations for Degenerate Polynomial Systems.* submitted for publication.

[Roj97c] ________________: *Affine Elimination Theory and its Applications to Diophantine Complexity.* preprint, MIT.

[Stu97] B. Sturmfels: *Introduction to Resultants.* Proceedings of the AMS Short Course on Applications of Computational Algebraic Geometry, San Diego, January 6–7, 1997, AMS.

An Artificial Neural Network for approximating functions in robotics

Juan Miguel Santos, Adrián Cristal

Introduction

The functions that define the kinematics and dynamics of real mechanical manipulators are especially interesting for the modelization of Artificial Neural Networks (ANN). The reason for such an approach is that real manipulators can be modelized by non–linear, highly coupled systems with many unknown parameters. Previous work in this direction concerns e.g. the Cerebellar Model Articulation Controller [Albus 1975], [Miller et al. 1993], context networks [Yeung & Beckey 1993], hierarchical networks [Kawato et al. 1987] and ANN in adaptative control schemes [Xianzhong & Kang 1993].

Approximation by means of hyperplanes allows not only the determination of the value of the function under consideration at each point of the training set, but also allows linear interpolation of the remaining points of the given domain. This allows a precise definition of the notion of generalization. With this idea in mind, we introduce a neural architecture and wish a reconstruction algorithm having polinomial complexity (a different approach based on the ANN paradigm can be found in [Funahashi 1989] and [Moody & Darken 1989]). The behaviour of our learning function was tested on ideal and real manipulators.

Architecture and network activation

Suppose we ignore the analytical expression of a given function $f : \mathbb{R}^n \to \mathbb{R}$, but we have a set of points $\big(\xi, \zeta := f(\xi)\big)$ of f, then we can define a training set $X = \{(\xi^\mu, \zeta^\mu); \mu : 1, \ldots, N\}$ where N is the amount of available points.

We wish to construct a network, with the architecture showed in the Figure 1, able to approximate the value of the function f in points that do not belong to X . This network consists of layers C_1 and C_2 that constitute a classification sub-network (CSN) and layers A and S. In its turn the layers A and S constitute an approximation sub-network (ASN). Each unit of the given CSN is a simple perceptron with a step function of activation, and each unit of ASN is a simple perceptron with a linear function of activation.

An input of the network is propagated to ASN as well as is to the CSN. The CSN

classifies the input according to the domain region it belongs.

Each unit of the layer C_2 identifies a subdomain of f, such that only one of them will be active while the rest of units in C_2 will remain with activity zero.

Each unit of layer A receives information from the input and from the layer C_2 too, therefore its activity state will depend on the classification made by the CSN. The network output will be the activation level of the unique unit of the layer S, which will depend directly of the activation state units of the layer A.

The activation state $c_{1,j}$ of each unit $C_{1,j}$ in the layer C_1 is given by

$$c_{1,j} = \varphi\left(\sum_{i=1}^{n} u_{ji}\xi_i - t_{1,j}\right)$$

where (u_{ji}) is the weight matrix of the connections that join the input with the layer C_1 , $t_{1,j}$ is the threshold of the unit $C_{1,j}$ and φ is the step activation function defined by

$$\varphi(x) = \begin{cases} -1 & if \quad x < 0 \\ 1 & if \quad x \geq 0 \end{cases} .$$

The activation state $c_{2,k}$ of each unit $C_{2,k}$ of the layer C_2 is given by

$$c_{2,k} = \phi\left(\sum_{j=1}^{m-1} v_{kj}c_{1j} - t_{2,k}\right)$$

where (v_{kj}) is the weight matrix of the connections that join the layer C_1 with the layer C_2 , $t_{2,k}$ is the threshold of the unit $c_{2,k}$ and ϕ is a step activation function defined by

$$\phi(x) = \begin{cases} 0 & if \quad x < 0 \\ 1 & if \quad x \geq 0 \end{cases} .$$

The activation state of each unit of the layer A is given by

$$a_k = \left(\sum_{i=1}^{n} w_{ki}\xi_i - t_{3,k}\right) c_{2,j}$$

with (w_{kj}) the weight matrix of the connections between the input and the layer A .

The network output is given by the activation state $o = \sum\limits_{k=1}^{m} a_k$ of the unique unit of the layer S.

The connections between the units of layer C_2 with layer A as well as the connections between layer A and layer S, have unitary weight.

Figure 1: Network architecture.The amount of units of the layer C_2 indicates the amount of planes used by the network to approximate a certain given function. The amount of units in the input layer coincides with the dimension of the domain. The units of C_1 are perceptrons that separate the domain in m subdomains: one for each approximation plane.

Fundamental and construction algorithm

We are going to describe an algorithm which does the following: for a given a set $X = \{(\xi^\mu, \zeta^\mu); \mu : 1, \ldots, N\}$, the algorithm contructs a network architecture as introduced in the previous section. For each element of X, the algorithm computes the secant plane p^μ that passes by (ξ^μ, ζ^μ) and its n neighbors and associates a set T^μ of points approximated by p^μ with an error $\epsilon > 0$. Let $P = \{p^\mu; \mu : 1, \ldots, N\}$ with $p^\mu = (p_0^\mu, p_1^\mu, p_2^\mu, \ldots, p_n^\mu) \in \mathbb{R}^{n+1}$ and where the components of p^μ are the coefficients of the hyperplane equation that passes through (ξ^μ, ζ^μ) and the $n+1$ nearest points to (ξ^μ, ζ^μ). For this sake we define a function $\alpha : \{1, \ldots, n+1\} \to \{1, \ldots, N\}$ having the following properties:

i) $\left\|\xi^{\alpha(1)} - \xi^\mu\right\| \leq \left\|\xi^{\alpha(2)} - \xi^\mu\right\| \leq \cdots \leq \left\|\xi^{\alpha(n+1)} - \xi^\mu\right\|$

ii)There exist ξ^r and $r \in \{1, \ldots, N\}$ such that

$$\left\|\xi^{\alpha(i)} - \xi^\mu\right\| \leq \|\xi^r - \xi^\mu\| < \left\|\xi^{\alpha(i+1)} - \xi^\mu\right\|$$

for $1 \leq i \leq n$.

This means that p^μ must satisfy the linear equation system:

$$\left\{\begin{array}{rcl} p_0^\mu + p_1^\mu \xi_1^{\alpha(1)} + p_2^\mu \xi_2^{\alpha(1)} + \cdots + p_n^\mu \xi_n^{\alpha(1)} & = & \zeta^{\alpha(1)} \\ & \vdots & \\ p_0^\mu + p_1^\mu \xi_1^{\alpha(1)} + p_2^\mu \xi_2^{\alpha(1)} + \cdots + p_n^\mu \xi_n^{\alpha(1)} & = & \zeta^{\alpha(1)} \end{array}\right.$$

We define now $T = \left\{T^1, T^2, \ldots, T^N\right\}$ by the property:

$$\forall \left(\xi^t, \zeta^t\right) \in T^\mu \Rightarrow \left| \zeta^t + p_0^\mu + \sum_{j=1}^{n} p_j^\mu \xi_j^t \right| < \epsilon \quad \text{for } t : 1, \ldots, N$$

Now, we wish to find a set of perceptrons that allows domain partition so that each set of the partition contains only points of the training set that belong to just one T^μ.

For this purpose we are going to construct a partition $S = \{S_1, S_2, \ldots, S_m\}$ of the domain fo the function f with the following property:
for any index $1 \leq i \leq m$ there exists exactly one index $1 \leq j \leq N$ such that for any index $1 \leq k \leq N$ the following implication holds: $\xi^k \in S_i \Rightarrow (\xi^k, \zeta^k) \in T^j$. Obviously we have $m \leq N$. The construction of the partition S produces for any S_i a plane where the function f is approximated.

Let us consider the domain D where we want to approximate the function. We define recursively a partition D^* of D in the following way:

i) $D_1^* = D$, $D_2^* = \{D_+, D_-\}$ where $D_+ = \{\mathbf{x} \in D / \mathbf{x} \cdot \mathbf{q} \geq 0\}$, $D_- = \{\mathbf{x} \in D / \mathbf{x} \cdot \mathbf{q} < 0\}$ and $\mathbf{q}$ is a perceptron with step activation function.

ii) $D_{n+1}^* = (D_n^* - \{D_h\}) \cup \{D_{h+}\} \cup \{D_{h-}\}$ with $D_h \in D_n^*$, $D_{h+} = \{\mathbf{x} \in D_h / \mathbf{x} \cdot \mathbf{q}_h \geq 0\}$ and $D_{h-} = \{x \in D_h / \mathbf{x} \cdot \mathbf{q}_h < 0\}$. Here h is a string of symbols $h_1, h_2, \ldots, h_n \in \{+, -\}$ and $\mathbf{q}_h$ is a perceptron which subdivides the domain D_h.

We say that D_m^* is a partition of m sets induced by a $m-1$ perceptrons.

We propose an algorithm in Figure 3 which produces perceptron $C_{1,1}, C_{1,2}, \ldots, C_{1,m-1}$ inducing a partition $D_m^* = \{D_{h_1}, D_{h_2}, \ldots, D_{h_m}\}$ with the following property: there exists an index $1 \leq i \leq m$ such that $\xi^\mu \in D_{h_i}$ implies $\xi^\mu \in S_i$). By the way this algorithm yields a network architecture and the weights of a network that approximates the given function f.

At the beginning this algorithm is called by the following items: a network with only one unit in layer C_2 (threshold= -0.5) connected only with its respective unit in layer A , the index *ind* set to one, the complete training set and the error bound ϵ. When some unit is added, the weights not yet assigned by the algorithm have value zero.

In Steps 1, 2 and 3 we obtain T^{k_0} maximal in the sense that the hyperplane p^{k_0} approximates the largest possible quantity of points of X.

Step 4 is a modification of the Pocket algorithm [Mezard & Nadal 1989]. This modification guarantees that it is always possible to find a perceptron which splits the training set in two non-empty sets. This condition is necessary to prove the termination of the whole procedure.

In Step 5 the weight assignation between layer C_1 and C_2, and threshold assignation

Figure 2: Changes of the architecture of the CSN introduced by Step 5 of the algorithm.

to units of C_2 guarantees the activation of only one unit in C_2 when an input is introduced to the network.

In Figure 2 we can observe the architecture of a CSN with only one node in the layer and two nodes in the layer C_2. The activation state of $C_{1,1}$ belongs to a $\{1, -1\}$ set. Only one unit in layer C_2 has unitary activation and the rest zero activation.

Figure 2 shows the domain points to be learned. The straight line represents the separation hiperplane defined by the $C_{1,1}$ weights and thresholds, and the points belonging to the semiplane defined by $\xi_1 u_{1,1} + \xi_2 u_{1,2} > t_{1,1}$ form a non resolute set. The part b of Figure 2 shows a modified architecture which produces a partition without non resolute sets. The weights and thresholds of the new units $C_{1,2}$ and $C_{2,3}$ are obtained following Step 5 of the algorithm.

Algorithm.

Let R be a network with m units in layers C_2 and A, $m-1$ units in layer C_1;

Let $X = \{(\xi^\mu, \zeta^\mu), \mu : 1, \ldots, N\}$ be a non resolute set, *ind* the index of the unit of layer C_2 which this set and an error bound.

The output of the algorithm is a network which approximates the function according to the points in X.

Step 1. Construct $P = \{p^\mu\}$ such that p^μ verifies (1) for each μ.

Step 2. Construct $T = \left\{T^1, \ldots, T^N\right\}$, $T \subset P(X)$ such that for any $1 \leq k \leq N$ the following equivalence holds: $(\xi^\mu, \zeta^\mu) \in T^k \Leftrightarrow \left|\zeta^\mu - p_0^k + \sum\limits_{i=1}^{n} p_i^k \xi_i^\mu\right| < \epsilon$.

Step 3. Construct a partition $B = \{B_-, B_+\}$ with $B_+ = T^{k_0}$ and $B_- = X - T^{k_0}$ where $T^{k_0} \in T$ and $\#T^{k_0} \geq \#T^k$ holds for all $1 \leq k \neq k_0 \leq N$.

Step 4. Construct a perceptron $\mathbf{q}$ which splits $\mathbb{R}^n$ in two nonempty sets B^*_+ and B^*_-, satisfying the condition $B_- \cap B^*_- \neq \emptyset$.

Step 5. Let $B^* = \left\{B^*_-, B^*_+\right\}$ be the partition induced by $\mathbf{q}$. Add $\mathbf{q}$ as the unit $c_{1,m}$ onto layer C_1 of R.

Add $c_{2,m+1}$ to layer C_2 with the same connections that $c_{2,ind}$, i.e. $v_{m+1,j} = v_{ind,j}$ for $1 \leq j \leq m-1$ and $t_{2,m+1} = t_{2,ind}$

Add 1 to $t_{2,m+1}$ and $t_{2,ind}$.

Set $v_{m+1,m} = 1$ and $v_{ind,m} = -1$.

Add a_{m+1} onto layer A.

Step 6. Let k satisfy the condition $\left(B^*_+ \cap B\right) \subset T^k$. Set a_{m+1} equal to p^k, i.e. $w_{m+1,j} = p^k_j$ for $1 \leq j \leq N$ and $t_{3,m+1} = p^k_0$.

else

Call the algorithm with R ,$\left(B^*_+ \cap B\right)$, $m+1$ and ϵ.

If k' is any index with $\left(B^*_- \cap B\right) \subset T^{k'}$, set a_{ind} equal to p^k, i.e. $w_{ind,j} = p^k_j$ for $1 \leq j \leq N$ and $t_{3,ind} = p^k_0$.

else
Call the algorithm with R, $\left(B^*_- \cap B\right)$, *ind* and ϵ.

Step 7. End.

Figure 3. Algorithm constructing the artificial neural network.

We will prove inductively that Step 5 of algorithm preserves the property of the network that only one unit has unitary activation and the remaining units have zero activation.

First, note how the threshold value $t_{2,i}$ and $\sum\limits_{j=1}^{m-1} v_{1,j} c_{1,j}$ are related. Each unit added to layer C_2 inherits the connections and threshold values of the unit that is representing the non resolute set. Moreover it receives a new connection from the new unit in layer C_1. The threshold values of both units will increase by one. Therefore, the fan-in of the units of layer C_2 is the value of its threshold plus one. Thus

$$(2) \text{ if } \Big(\sum_{j=1}^{m-1} v_{1,j} c_{1,j}\Big) > t_{2,i} \text{ then } \Big(\sum_{j=1}^{m-1} v_{1,j} c_{1,j}\Big) - t_{2,i} = 0.5.$$

(Note: the threshold initial value of the first unit in layer C_2 is -0.5). Let CSN_1 be the network with $m-1$ units in layer C_1 and m units in layer C_2 and CSN_2 the network obtained by Step 5 of the algorithm. In CSN_1 for any pattern present in the input layer, only one unit in the layer C_2 will have unitary activation.

Let $C_{2,ind}$ be a unit that represents a non resolute set. We present a pattern to CSN_1 and suppose that the unit $C_{2,ind}$ has output. That is, $c_{2,ind} = \phi\left(\sum_{j=1}^{m-1} v_{ind,j} - t_{2,ind}\right) = 0$.

If we present the same pattern to CSN_2, we have

$$c'_{2,ind} = \phi\left(\sum_{j=1}^{m} v_{ind,j} c_{1j} - t'_{2,ind}\right) = \phi\left(\sum_{j=1}^{m-1} v_{ind,j} c_{1,j} - t_{2,ind} - 1 + (-1)\, c_{1,m}\right) = 0$$

since $\sum_{j=1}^{m-1} v_{ind,j} c_{1,j} - t_{2,ind} < 0$ and $(-1 - c_{1,m}) \leq 0$ ($c'_{2,ind}$ and $t'_{2,ind}$ correspond to CSN_2). In the same way we see that $c'_{2,m+1}$ has output zero, since $t_{2,m+1} = t_{2,ind}$, $v_{m+1,j} = v_{ind,j}$ for $1 \leq j \leq m-1$ and $-1 + c_{1,m} \leq 0$ holds.

We suppose now that the unit $c_{2,ind}$ has output equal to one when we present some pattern to CSN_1. Then by (2) we have $\sum_{j=1}^{m-1} v_{ind,j} c_{1,j} - t_{2,ind} = 0.5$.

When the unit $c_{1,m}$ has output equal one, we obtain

$$c'_{2,ind} = \phi\left(\sum_{j=1}^{m-1} v_{ind,j} c_{1,j} - t_{2,ind} - 1 + (-1)\, c_{1,m}\right) = 0$$

and

$$c'_{2,ind} = \phi\left(\sum_{j=1}^{m-1} v_{ind,j} c_{1,j} - t_{2,ind} - 1 + c_{1,m}\right) = 1,$$

but when $c_{1,m}$ has output equal -1, $c'_{2,ind}$ and $c'_{2,m+1}$ are one and zero respectively.

We are going to show that our algorithm has worst case complexity $O(n^2)$. The complexity of the Pocket algorithm is kn where n is the cardinal of the given set and k is a constant. The worst case occurs when the cardinality of the set in each recursion step to the algorithm is $n-1$. Such a step requires $k(n-1)$ operations. Hence in $n-1$ steps we obtain $k(n-1)n/2$ operations and thus the complexity of our algorithm is $O(n^2)$. However, if the algorithm in each recursive call splits the set in two equal parts, we obtain a binary tree with kn operations in each level.

From the fact that the height of the complete binary tree is $\log_2 n$, we deduce that the complexity of the algorithm is $O(n \log_2 n)$. This complexity is bounded from above nh, where h is the height of the recursion tree.

Ideal Manipulator

For simulations we use functions that connects acceleration $\ddot{\theta}$ and torque τ (given the position θ and speed $\dot{\theta}$) in the joints of a dynamic model of a two degree of

Figure 4: Average of absolute error on validation set vs. training set cardinal. Validation set has 10000 patterns following the uniform distribution.

freedom manipulator with rotational joints [Craig 1989]. The length of link 1 is $l_1 = 0.4m$, the length of link 2 is $l_2 = 0.2m$, the mass of link 1 and link 2 are $m_1 = 1kg$ and $m_2 = 0.5kg$ respectively. The mass is concentrated at distal end. The function domains are given by $\theta_1 \in (0, \pi)$, $\theta_2 \in (0, \frac{\pi}{2})$, $\dot{\theta}_1, \dot{\theta}_2 \in (-0.5, 0.5)$ and $\ddot{\theta}_1, \ddot{\theta}_2 \in (-0.1, 0.1)$. The points have been obtained in a random way with uniform distribution and their outputs correspond to

$$\begin{aligned}\tau_1 = {} & m_2 l_2^2 \left(\ddot{\theta}_1 + \ddot{\theta}_2\right) + m_2 l_1 l_2 \cos\left(\theta_2\right)\left(2\ddot{\theta}_1 + \ddot{\theta}_2\right) + \left(m_1 + m_2\right) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 \sin\left(\theta_2\right) \ddot{\theta}_2^2 \\ & -2 m_2 l_1 l_2 \sin\left(\theta_2\right) \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g \cos\left(\theta_1 + \theta_2\right) + \left(m_1 + m_2\right) l_1 g \cos\left(\theta_1\right) \\ \tau_2 = {} & m_2 l_1 l_2 \cos\left(\theta_2\right) \ddot{\theta}_1 + m_2 l_1 l_2 \sin(\theta_2) \dot{\theta}_1^2 + m_2 l_2 g \cos\left(\theta_1 + \theta_2\right) + m_2 l_1 l_2^2 \left(\ddot{\theta}_1 + \ddot{\theta}_2\right)\end{aligned}$$

where g is the gravity acceleration.

We constructed 3600 networks from 3600 training set of 100, 200, 400, 800, 1600 and 3200 examples (600 for each one, three hundred for the first function and three hundred for the second one). For each error bound (=0.01, 0.05, 0.1, 0.2, 0.4 and 0.6) we produced fifty networks. We computed the average absolute error with a validation set of 10000 patterns following the uniform distribution. This set was unique for all networks in each function.

The average of the absolute error (aae) of the validation set was bounded by the error bound ϵ during the training while the cardinal of the training set increased (Figure 4). This did not increase significantly the amount of approximation units.

Real Manipulator

We have used a large and flexible manipulator with rotational joints for the experimentation with our system. The robot had four principal degrees of freedom, one of them for vertical axis rotation while the other three joints were planar and redundant. Each joint was moved by means of hydraulic actuators and they were

Figure 5: Control scheme for the evaluation of ANNs (inverse model).

controlled by electro-valves of linear type. At each joint, there was collocated an absolute sensor (potentiometer).

Amplitude [volts]	-1.0	-0.5	-0.2	-0.1	0	0.1	0.2	0.5	1.0
Duration [msec]	250	500	750	1000	1250	1000	750	500	250

Table 1. Test amplitude and duration.

In order to obtain these measurements, we carried out 81 open loop tests. Table 1 shows the amplitudes and durations of the step function applied to electro-valves at joints 1 and 2. For each one of these voltages applied to joint 1, the complete series of voltages was applied to joint 2. We carried out tests at 70 different angular positions by combining 14 angular positions of joint 1 with 5 angular positions of joint 2. The intervals were, for joint 1: $\theta(1) \in (1.91, -1.44)$, and for joint 2: $\theta(2) \in (2.06, 0.95)$. With an interval of 10 msec. the following training set points were obtained: for joint 1 $\{\{\theta_m(1), \theta_m(2), \delta(2), \delta(1)\ \},\{\ V_{EV}(1)\ \}\}$. Hhere $\theta_m(1)$ and $\theta_m(2)$ are the average of the angular position of joint 1 and joint 2 during the complete interval, $\delta(2)$ is the rate of change of the angular position in joint 2, $\delta(1)$ is the rate of change of the angular position in joint 1 and $V_{EV}(1)$ is the voltage applied to the joint 1 and the desired output to be learnt. We made up the training set for joint 2 in a similar way.

Each training set had 10455 examples and we selected randomly 1000 of them to form the validation set.

for a given training set, we supposed the existence of an optimal value for ϵ with minimum average of absolute error *aae* (during validation). We constructed networks with training sets of 100, 200, 400, 800, 1600, 3200, 6400 and 9055 examples. For each one, we varied ϵ to find the network with minimal *aae* and we chose networks trained with a set of 3200 examples, ϵ=0.17 and β=0.05 for joint 1 and 6400 examples, ϵ=0.5 and β=0.1 for joint 2. The ANNs obtained in this way were assumed to be the inverse model of two joints of the real plant. They contain 316 units (105 units in layer C_1) representing joint 2, and 73 units (24 units in layer C_1) representing joint 1.

Joint	Square Mean	Absolute Max.	Absolute Mean	Bound Overshoots
1	0.0019	0.719	0.044	8
2	0.004	0.439	0.11	0

Table 2. Validation of the ANNs

Table 2 shows the results of the validation. The last column contains the quantity of examples approximated with an error larger than the bound (overshoot).

The experimental scheme for evaluating these ANNs given follows the control scheme shown in Figure 5. There were 60 straight trajectories randomly defined in the workspace to be followed by the end effector at constant speed. Table 3 shows the average relative error $e = \frac{1}{m}\sum_{i=1}^{m} \frac{D_i}{dist_i^B}$ of the computation. Here D_i is the distance from the observed point i of the real trajectory to the ideal trajectory, $dist_i^B$ is the distance between the coordinate origin (xy) and the observed point i, m is the observed point quantity of the real trajectory.

Trajectory	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*
$e(\%)$	0.84	0.06	0.91	2.02	0.19	0.95	0.76	0.11

Table 3. Average of relative error.

References

[1] J.S. Albus: *A new approach to manipulator control: The cerebellar Model Articulation Controller.* Trans. of the ASME (1975) 220–227.

[2] John J. Craig: *Introduction to robotics, mechanics and control* (Second ed.). Adison Wesley Pub (1989).

[3] Ken-Ichi Funahashi: *On the approximate Realization of Continuos mappings by neural Networks.* Neural Networks **2** (1989) 183–192.

[4] M. Kawato, Y. Uno, M. Isobe, R. Susuki: *A hierarchical model for voluntary movement and its application to robotics.* Proc. IEEE Conf. of Neural Network (1987) 573–582.

[5] M. Mezard, J.P. Nadal: *Learning in Feedforward Layered Networks: The tiling algorithm.* Journal of Physics (1989).

[6] T.W. Miller III, R.P. Hewes, F.H. Glanz, L.G. Kraft: *Real Time Dynamics Control of an Industrial Manipulator using a Neural Network Based Learning Controller.* IEEE Trans. Robotics Automation **6** (1993) 1–9.

[7] J. Moody, C. Darken: *Fast Learning in Networks of Locally-Tuned Processing Units.* Neural Computation **1** (1989) 281–294.

[8] Dit Yan Yeung, George A. Beckey: *On reducing Learning Time in Context-Dependent Mapping.* IEEE Trans. Neural Network **4** (1993).

[9] Cui Xianzhong, G. Shin Kang: *Direct Control and Coordination Using Neural Networks.* IEEE Trans. Systems, Man and Cybernetics **23 (3)** (1993).

Aproximación de la dinámica inversa de un manipulador robótico mediante un algoritmo estocástico de aprendizaje en redes neuronales

Enrique Carlos Segura

Resumen. Se utiliza el algoritmo SAGA [Segura, 1996] para aproximar la dinámica inversa de un manipulador robótico de dos juntas rotacionales. SAGA (Simulated Annealing + Gradiente + Adaptacion) es una estrategia estocástica para construcción aditiva de una red neuronal artificial del tipo perceptrón bicapa basada en tres elementos esenciales: a) modificación de los pesos de la red mediante información del gradiente de la función de costo; b) aceptación o rechazo del cambio propuesto según una técnica de simulated annealing clásico; y c) crecimiento progresivo de la red neuronal, conforme su estructura resulta insuficiente, según una estrategia conservadora de adición de unidades a la capa oculta.

Se realizan experimentos y se analiza la eficiencia en términos de la relación entre error relativo promedio -en el conjunto de entrenamiento y en el de testeo-, tamaño de la red y tiempo de cálculo.

Se destaca la aptitud de la técnica propuesta para realizar buenas aproximaciones minimizando la complejidad de la arquitectura de la red y, por ende, la memoria computacional requerida. Asimismo se discute la evolución del proceso de minimización conforme la superficie de costo va siendo modificada a medida que la red se agranda.

1. Introducción

Una de las propiedades más popularizadas del Simulated Annealing (SA) clásico y, a la vez, más frecuentadas en la bibliografía [2], [6], [10], es su capacidad para obtener en los problemas de optimización complejos (i.e. dominios de muchas variables y función de costo con muchos mínimos locales y de gradiente muy alto) buenas aproximaciones "gruesas", es decir, acercamiento a un óptimo o subóptimo en tiempos razonables. Como contraparte, el SA suele fallar o ser muy lento en el "ajuste fino", es decir la localización precisa del óptimo buscado. Por otro lado, los algoritmos de minimización (optimización) basados en el gradiente (y en el caso particular de las redes neuronales, el Backpropagation [11]) resultan casi siempre incapaces de escapar de una zona de mínimo local pero muy adecuados para, ya dentro de la región deseada, precisar la ubicación del mínimo. Así surge, naturalmente, la idea de un algoritmo híbrido que concilie cooperativamente las buenas propiedades de ambos enfoques.

El otro ingrediente que se trata de incluir apunta a la búsqueda de la optimalidad. Son bien conocidas [3], [4] las condiciones mínimas que deben imponerse sobre la estructura de una red neuronal de tipo feedforward para garantizar su universalidad en tanto aproximadora de funciones continuas. Atento a ellas, se buscó incorporar a la red propuesta o, mejor dicho, a su estrategia de aprendizaje, un mecanismo de adición de unidades de procesamiento que hiciera al sistema capaz de obtener estructuras de complejidad mínima o cuasi-mínima para aproximar con errores aceptables la función objetivo.

2. Hacia una estructura simple

Muchos investigadores han buscado reducir clases amplias de funciones a combinaciones simples de funciones simples. Uno de los primeros antecedentes es el de Kolmogorov [7] quien, muy alejado del discurso conexionista, estableció la representabilidad (exacta) de cualquier función continua en muchas variables por medio de sumas y composiciones de funciones de una variable. Este resultado, así como sus sucedáneos de unos pocos años después ([1], [13]), puede traducirse, en términos de redes neuronales, como la demostración de que cualquier función continua puede ser representada como una red neuronal de tres capas (si bien lo que Kolmogorov se proponía era, en realidad, responder negativamente al tridécimo problema de Hilbert). Pero estos resultados son presentados de manera por completo no constructiva.

A partir de entonces, distintos trabajos jalonan este camino. En 1989, dos matemáticos presentan, casi simultáneamente, demostraciones de que los perceptrones de tres capas (una de entrada, una oculta, una de salida) con funciones sigmoideas de activación en sus unidades intermedias pueden aproximar con un error arbitrariamente pequeño, en cualquier métrica L^p , una función continua arbitraria ([3], [4]). Enunciamos aquí el resultado en la versión de [4].

Theorem 1 (Funahashi)
Sea ϕ función monótona creciente y acotada; $C \in R^N$ compacto, $f : C \to R^L$ continua.
Dado $\varepsilon > 0$ arbitrario, existe $K \in N$ y constantes reales W_{ij}, u_j $(j = 1,..,K, i = 1,..,L)$ y w_{jk} $(j = 1,...,K;\ k = 1,...,N)$ tales que si
$\tilde{f}(\xi_1,...,\xi_N) = \sum_{j=1}^{K} W_{ij}\phi(\sum_{k=1}^{N} w_{jk}\xi_k - u_j)$
entonces $\left\| f - \tilde{f} \right\|_{L^\infty(C)} = máx_{\xi\in C} \left| f(\xi_1,...,\xi_N) - \tilde{f}(\xi_1,...,\xi_N) \right| < \varepsilon$

Este resultado vale, además, para cualquier métrica L^p , en particular para la cuadrática.

Figure 1: La estructura básica de un perceptrón de tres capas (una oculta). Las unidades intermedias tienen activación sigmoidea; las de entrada y las de salida, lineal.

Resultados más recientes ([5], [8]) han precisado como condición necesaria y suficiente sobre ϕ que no sea un polinomio.

La red neuronal resultante tendrá la forma que exhibe la figura 1.

Nos restringiremos, con evidente conservación de toda generalidad, al caso de una sola variable de salida (L perceptrones en lugar de uno). Luego, el conjunto $\{W_{ij}\}$ de parámetros será simplemente $\{c_i\}$

En esto se inspira la idea de obtener, con la estructura más económica posible, aproximaciones de cierta utilidad. El algoritmo que propondremos realiza, específicamente, una estrategia simple -¿ la más simple?- de agregado de unidades ocultas.

3. El algoritmo

Se presenta a continuación un esquema del algoritmo SAGA (Simulated Annealing + Gradiente + Adaptación). Dicha técnica ya ha sido aplicada, con ligeras diferencias, a problemas de calibración de cámaras de video para robots móviles ([9], [12]).

```
- Obtener un conjunto de entrenamiento (TRS) y otro de testeo (TES) a partir de f (de una sola variable de salida).
- Inicializar la red al azar /* $c_i(0), w_{ij}(0), u_i(0) \approx U(0,1) \quad i = 1,...,K;\ j = 1,...,N$ */
con una adecuada cantidad inicial K de unidades intermedias
- Mientras t < tmáx /* o bien $E_{TRS} > \varepsilon$ o bien $E_{TES} > \varepsilon$ */
{
T← T(t); /* típicamente T(t) = A/log(1+t) */
proponer_cambio(W,W*);
generar p s U(0,1)
Si exp{-E/T} > p; /* $\Delta E = E_x(W^*) - E_x(W)$*/
```

```
W ← W*;
Si no
rechazos++;
Si rechazos > tol y K<Kmáx
agregar(W,K);
t++;
}
proponer_cambio(W,W*)
{
Elegir x ∈ TRS al azar;
Elegir r s U(0,R);      /* R paso máximo fijado heurísticamente */
$W^* \leftarrow W - r\,\frac{\nabla E_x(W)}{|\nabla E_x(W)|}$; /* $E_x(W)$ término de la función de error debido a x */
}
agregar(W,K)
{
K++; /* crea una nueva unidad oculta */
Elegir m entre 1 y K-1; /* típicamente m tal que $|C_m|$ es máximo */
$u_K \leftarrow u_m$;
$w_{Kj} \leftarrow w_{mj} \quad para\ j = 1, ..., N$;
$c_K = c_m \leftarrow \frac{c_m}{2}$;
rechazos ← 0;
}
```

Si bien el algoritmo parece altamente parametrizado, debe destacarse que de las experiencias realizadas surge que los parámetros relevantes son sólo tres: uno para la curva de enfriamiento del SA (variable según la elección de dicha curva), el paso máximo R y el valor de tolerancia (tol) que, al ser superado por el número de rechazos consecutivos, produce la invocación a la rutina de ampliación de la capa oculta de la red (agregar), la que, al tiempo que incrementa en N+2 la dimensión del espacio de los pesos, mantiene el valor de la función de costo obtenida hasta ese momento.

4. Una aplicación

SAGA fue aplicado en el presente trabajo a la aproximación de la dinámica inversa de un manipulador robótico de dos grados de libertad. Es decir, dado un esquema de control como el de la figura 2, donde y(t) es la salida en tiempo t, y*(t) la salida deseada y u(t) la señal de control, se trata de modelar la aplicación inversa: qué señal de control habría correspondido aplicar para obtener la salida deseada. En el caso del manipulador, el modelo inverso relaciona aceleraciones $\ddot{\theta}_1$ y $\ddot{\theta}_2$ en las juntas con torques τ_1 y τ_2 a aplicar (i.e. voltajes en los motores), dadas posiciones angulares θ_1 y θ_2 y velocidades $\dot{\theta}_1$ y $\dot{\theta}_2$.

Figure 2: Esquema de control con modelo inverso

Si bien se conoce una expresión analítica para dicha inversa, dada por las ecuaciones 4.1 y 4.2, estas ecuaciones no representan modelos exactos de la dinámica inversa de ningún manipulador real, la cual debería ser modelada a partir de mediciones experimentales sobre la planta. De todas formas, a los efectos de evaluar la eficiencia del método propuesto, se realizaron simulaciones computacionales utilizando las aproximaciones dadas por dichas ecuaciones.

$$\tau_1 = m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 \cos(\theta_2)(2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 sen(\theta_2)\ddot{\theta}_2^2 -$$

$$-2 m_2 l_1 l_2 sen(\theta_2)\dot{\theta}_1\dot{\theta}_2 + m_2 l_2 g \cos(\theta_1 + \theta_2) + (m_1 + m_2) l_1 g \cos(\theta_1) \quad \mathbf{4.1}$$

$$\tau_2 = m_2 l_1 l_2 \cos(\theta_2)\ddot{\theta}_1 + m_2 l_1 l_2 sen(\theta_2)\dot{\theta}_1^2 + m_2 l_2 g \cos(\theta_1 + \theta_2) + m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) \quad \mathbf{4.2}$$

siendo l_i y m_i la longitud y la masa, respectivamente, del link i.

5. Experimentos

Se aplicó SAGA a la función $\tau = (\tau_1, \tau_2)$ en el dominio de R^6 definido por $\theta_1, \theta_2 \in [0, \pi]$, $\dot{\theta}_1, \dot{\theta}_2 \in [-0.5, 0.5]$ y $\ddot{\theta}_1, \ddot{\theta}_2 \in [-0.1, 0.1]$ con $l_1 = 0.4$ m., $l_2 = 0.2$ m., m_1 =1 kg. y $m_2 = 0.5$ kg.

La función aproximante, es decir la red neuronal, fue de la forma
$\tilde{\tau}_i(x_1, \ldots, x_) = \sum_{j=1}^{K} c_i \phi(\sum_{k=1}^{6} w_{jk} x_k - u_j)$
para y = 1,2, siendo $x_1, \ldots, x_=(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2, \ddot{\theta}_1, \ddot{\theta}_2)$ y $\phi(h) = tanh(\beta h) = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$

Exhibimos aquí algunos de los resultados obtenidos para el caso de τ_1 . La figura 3 muestra la evolución del error relativo promedio para dos conjuntos de entrenamiento independientes. Se observan buenos resultados para 300 ejemplos (6.6 %) y para 400 ejemplos (7.4 %). La figura 2 muestra, para los mismos conjuntos, la evolución del error de entrenamiento a medida que se fueron agregando unidades a la capa oculta. Fijando un máximo de 30 unidades se llega a las aproximaciones precitadas. Debe destacarse, sin embargo, que no siempre una mayor cantidad de ejemplos produce mayor error en la aproximación. Por ejemplo, si hubiésemos decidido que el proceso terminara a las 20000 iteraciones (fig. 1) o en un máximo de 12-14 unidades ocultas (fig. 2), claramente habría resultado más exitoso para el conjunto de 400 ejemplos. Se destaca, de cualquier forma, la gran economía y simplicidad de la estructura de la red neuronal resultante. La figura 5 presenta, para el conjunto anterior de 400 ejemplos, y otro de testeo de 1500, la evolución temporal de los errores (siempre relativos y promedio). Aqui se ilustra con claridad el fenómeno del sobreentrenamiento: a partir de las 50000 iteraciones el error de testeo crece, lo cual sugiere que la fase útil del proceso se ha cumplido.

6. Conclusiones

Se presentó una técnica estocástica para la construcción de redes neuronales artificiales de tipo perceptrón bicapa que permite obtener buenas aproximaciones de funciones continuas cualesquiera mediante estructuras muy simples y número mínimo o cuasi-mínimo de unidades de procesamiento.

La propuesta fue ensayada en el caso de la inversa dinámica de un manipulador robótico de dos grados de libertad (juntas rotacionales), lo que implicó modelar en forma aproximada una función de seis variables de entrada y dos de salida. Si bien los resultados obtenidos pueden considerarse muy buenos, sería recomendable ampliar las experiencias a más grados de libertad, así como a conjuntos (de testeo y de entrenamiento) de mayor cardinalidad. En cuanto al dominio sobre el que se trabajó, puede considerarse bastante cercano a un caso de la realidad.

Reconocimiento: el presente trabajo forma parte del proyecto EX157 de la Secretaría de Ciencia y Técnica de la Universidad de Buenos Aires, que es sostenido con un subsidio de esa institución.

7. Referencias

[1] V.I. Arnold: *On Functions of three Variables.* Dokl. Akad. Nauk SSSR **114** (1957) 679–681.

[2] V. Cerny: *Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm.* J. Opt. Th. and Appl. **45(1)** (1985) 41–51.

[3] G. CYBENKO: *Approximation by superpositions of a sigmoidal function.* Math. Control Signal System **2** (1989) 303–314.

[4] K. FUNAHASHI: *On the approximate realization of continuous mappings by neural networks.* Neural Networks **2** (1989) 183–92.

[5] Y. ITO: *Extension of Approximation Capability of Three Layered Neural Networks to Derivatives.* Proc. IEEE Int. Conf. Neural Networks, San Francisco (1993) 377–381.

[6] S. KIRKPATRICK, C.D. GELATT, M. P. VECCHI: *Optimization by Simulated Annealing.* Science **220** (1983) 671–680.

[7] A.N. KOLMOGOROV: *On the Representation of Functions of Many Variables by Superposition of Continuous Functions of one Variable and Addition.* Am. Math. Soc. Tr. **28** (1963) 55–59.

[8] M. LESHNO, V.Y. LIN, A. PINKUS, S. SCHOCKEN: *Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function.* Neural Networks **6** (1993) 861–867.

[9] A.B. MARTÍNEZ, R.M. PLANAS, E.C. SEGURA: *Disposición anular de cámaras sobre un robot móvil.* Actas XVII Jornadas de Automática Santander 96, Santander (1996).

[10] N. METROPOLIS, A. ROSENBLUTH, M. ROSENBLUTH, A.H. TELLER, E. TELLER: *Equation of State Calculations by Fast Computing Machines.* J. Chem. Phys. **21(6)** (1953) 1087–1091.

[11] D.E. RUMELHART, G.E. HINTON, R.J. WILLIAMS: *Learning representations by back-propagating errors.* Nature **323** (1986) 533–536.

[12] E.C. SEGURA: *A non parametric method for video camera calibration using a neural network.* Proc. Int. Symp. Multi-Technology Information Processing, Hsinchu, Taiwan (1996).

[13] D.A. SPRECHER: *On the Structure of Continuous Functions of Several Variables.* Tr. Am. Math. Soc. **115** (1963) 340–355.

Figure 3: Error de entrenamiento (relativo promedio) en funcion del tiempo (miles de iteraciones).

Figure 4: Error de entrenamiento (relativo promedio) en función del número de unidades ocultas.

Figure 5: Errores en el entrenamiento y en el testeo en función del tiempo (miles de iteraciones).

Bounds for sparseness in the Nullstellensatz

Martín Sombra

We present new bounds for sparseness and degrees of the polynomials in the Nullstellensatz.

Our bounds depend mainly on the unmixed volume of the input polynomial system.

Our degree bounds substantially improve the known ones when the polynomial system is sparse, and they are, in the worst case, simply exponential in terms of the number of variables and the maximum degree of the input polynomials.

Let us denote by k a field and by $\overline{k}$ its algebraic closure. As usual we denote by $\mathbb{A}^n$ the affine space of n dimensions over $\overline{k}$. Let be given polynomials $f_1, \dots, f_s \in k[x_1, \dots, x_n]$ without common zeros in $\mathbb{A}^n$. Classical Hilbert's Nullstellensatz states then that there exist $g_1, \dots, g_s \in k[x_1, \dots, x_n]$ satisfying the Bézout equation

$$1 = g_1 f_1 + \cdots + g_s f_s.$$

The study of this equation is the object of intensive research, due to both its theoretical and practical importance, mainly in the context of computational algebraic geometry and diophantine approximation.

Let d denote the maximum degree of the polynomials $f_1, \dots, f_s$ and assume that $n \geq 2$. In the previous situation we have indeed that there exist polynomials $g_1, \dots, g_s$ verifying the degree bound

$$\deg g_i f_i \leq \max\{3, d\}^n.$$

This result is due to Kollár, and it is optimal provided that $d \geq 3$ [2]. In the case when $d = 2$ and the characteristic of k is different from two, Sabia and Solernó proved that the sharper bound $\deg g_i f_i \leq n\, 2^{n+2}$ holds true [3].

We note that such a degree bound allows us, given polynomials $f_1, \dots, f_s \in k[x_1, \dots, x_n]$, to determine whether this equation is solvable or not, and in the case it is, to actually find a solution, as it reduces the original problem to the solving of a k–linear system of equations.

We are going to introduce the basic notions of sparse elimination theory in order to state our results.

Let be given a Laurent polynomial $f = \sum_{i \in \mathbb{Z}^n} a_i\, x^i \in k[x_1, \dots, x_n, x_1^{-1}, \dots, x_n^{-1}]$. The *support* of f is defined as the set $\{i : a_i \neq 0\}$, that is, the set of exponents of the nonzero monomials of f. More generally, the support of a family of Laurent polynomials $f_1, \ldots, f_s$ is defined as the set of exponents of the nonzero monomials of any of the f_i for $i = 1, \ldots, s$.

The *Newton polytope* $\mathcal{N}(f_1, \ldots, f_s)$ is defined as the convex hull of the support of $f_1, \ldots, f_s$. Let ρ denote the dimension of this polytope. Then the *unmixed volume* $\mathcal{U}(f_1, \ldots, f_s)$ of the family of Laurent polynomials $f_1, \ldots, f_s$ is defined as $\rho!$ times the volume of the polytope $\mathcal{N}(f_1, \ldots, f_s)$.

The degree of a polynomial is bounded by a nonnegative integer d if and only if its Newton polytope is contained in d times the standard simplex of $\mathbb{R}^n$. Thus the notion of Newton polytope gives a sharper characterization of the monomial structure of a polynomial system than just degree.

The sparse aspect in the Nullstellensatz has also been considered by Canny and Emiris, who obtained a sparse effective Nullstellensatz but only for the case of $n+1$ *generic* n–variate Laurent polynomials [1, Theorem 7.2].

We obtain the following result, which in this context can be seen as a bound for the sparseness of the output polynomials in terms of the sparseness of the input system.

Theorem 1. *Let be given polynomials $f_1, \ldots, f_s \in k[x_1, \ldots, x_n]$ without common zeros in $\mathbb{A}^n$. Let $\mathcal{N}$ denote the Newton polytope of the polynomials $x_1, \ldots, x_n, f_1, \ldots, f_s$, and let $\mathcal{U}$ denote the unmixed volume of this polytope. Then there exist $g_1, \ldots, g_s \in k[x_1, \ldots, x_n]$ satisfying*

$$1 = g_1 f_1 + \cdots + g_s f_s,$$

with $\mathcal{N}(g_i f_i) \subseteq n^{n+3}\, \mathcal{U}\, \mathcal{N}$ *for* $i = 1, \ldots, s$.

Let us keep the notations of Theorem 1 and let d denote the maximum degree of f_i for $i = 1, \ldots, s$. We readily derive from the previous result the degree bound

$$\deg g_i f_i \leq n^{n+3}\, d\, \mathcal{U}.$$

We obtain from this the worst–case bound $\deg g_i f_i \leq n^{n+2}\, d^{n+1}$, as the unmixed volume of the polynomials $x_1, \ldots, x_n, f_1, \ldots, f_n$ is bounded by d^n. We show with an example that our degree bound can considerably improve the usual one in the case when the input polynomial system is sparse and $d \geq n$.

Example. Let $f_1, \ldots, f_s \in k[x_1, \ldots, x_n]$ be polynomials without common zeros in $\mathbb{A}^n$. Assume in addition that $\deg_{x_1, \ldots, x_{n-1}} f_i \leq 1$ and that $\deg_{x_n} f_i \leq d$ holds for $i = 1, \ldots, s$. Let $\mathcal{P}_d$ denote the polytope $\{x \in \mathbb{R}^n : x_1, \ldots, x_n \geq 0,\ x_1 + \cdots + x_{n-1} \leq 1,\ x_n \leq d\}$. Then $\mathcal{P}_d$ contains the Newton polytope of the polynomials $1, x_1, \ldots, x_n, f_1, \ldots, f_s$. Thus we are in the hypothesis of Theorem 1 and we can conclude that there exist $g_1, \ldots, g_s \in k[x_1, \ldots, x_n]$ such that

$$1 = g_1 f_1 + \cdots + g_s f_s$$

holds, with $\mathcal{N}(g_i f_i) \subseteq d\, n^{n+4}\, \mathcal{P}_d$ for $i = 1, \ldots, s$, as the volume of $\mathcal{P}_d$ equals $d/(n-1)!$. In particular we get the degree bound $\deg g_i f_i \leq n^{n+4}\, (d+1)^2$, which is much sharper than the estimation $\deg g_i f_i \leq (d+1)^n$ which follows from direct application of the usual degree bound.

We obtain an analogous result for the case of Laurent polynomials.

Theorem 2. *Let be given Laurent polynomials* $f_1, \ldots, f_s \in k[x_1, \ldots, x_n, x_1^{-1}, \ldots, x_n^{-1}]$ *without common zeros in* $(\overline{k}^*)^n$. *Let calN denote the Newton polytope of* $f_1, \ldots, f_s$, *and let* $\mathcal{U}$ *denote the unmixed volume of this polytope. Then there exist* $a \in \mathbb{Z}^n$ *and* $g_1, \ldots, g_s \in k[x_1, \ldots, x_n, x_1^{-1}, \ldots, x_n^{-1}]$ *satisfying*

$$1 = g_1 f_1 + \cdots + g_s f_s,$$

with $a \in n^{2n+3}\,\mathcal{U}^2\,\mathcal{N}$ *and* $\mathcal{N}(g_i f_i) \subseteq n^{2n+3}\,\mathcal{U}^2\,\mathcal{N} - a$ *for* $i = 1, \ldots, s$.

The proof of both results is similar. It takes as its first step the traslation of the original system of equations over the affine space or the torus into a system of linear equations over an appropiate toric variety. The resulting system is then solved by appealing to an effective Nullstellensatz for linear forms in a Cohen–Macaulay graded ring. The main obstacle in the proof of this key lemma is the presence of components at infinite distance in the variety defined by the linear forms. This difficulty is solved by the application of a new technique for eliminating this components at infinite distance.

Besides, we apply these arguments in the context of the usual effective Nullstellensatz and we obtain the following degree bound.

Theorem 3. *Let be given polynomials* $f_1, \ldots, f_s \in k[x_1, \ldots, x_n]$ *without common zeros in* $\mathbb{A}^n$. *Let* d_i *denote the degree of* f_i *for* $i = 1, \ldots, s$ *and assume that* $d_1 \geq \cdots \geq d_s$ *holds. Then there exist* $g_1, \ldots, g_s \in k[x_1, \ldots, x_n]$ *satisfying*

$$1 = g_1 f_1 + \cdots + g_s f_s,$$

with $\deg g_i f_i \leq 2\, d_s \prod_{j=1}^{\min\{n,s\}-1} d_j$ *for* $i = 1, \ldots, s$.

In the case when the polynomials $f_1, \ldots, f_s$ are quadratic this estimation improves the bound $\deg g_i f_i \leq n\, 2^{n+2}$ to $\deg g_i f_i \leq 2^{n+1}$, which is very close to the expected optimal bound 2^n.

References

[1] J. Canny, I. Emiris: *A subdivision–based algorithm for the sparse resultant.* Manuscript (1996).

[2] J. Kollár: *Sharp effective Nullstellensatz.* J. Amer. Math. Soc. **1** (1988) 963–975.

[3] J. Sabia, P. Solernó: *Bounds for traces in complete intersections and degrees in the Nullstellensatz.* AAECC Journal **6** (1995) 353–376.

[4] M. Sombra: *A sparse effective Nullstellensatz.* Manuscript (1997).

Non linear interaction between long waves in the Equator

C. Turner, E. Tabak, R. Rosales, A.Majda, F.Menzaque

Introduction

The change of sign of the Coriolis's force in the Equator produces an effective waveguide for a large class of long atmospheric waves. Among these waves, the longest and slowest are the nondispersive Kelvin and the dispersive Rossby and Yanai —or mixed Rossby-gravity— waves. The most prominent waves observed are very long Kelvin waves [2], with wavenumbers $l = 1$ and $l = 2$ —i.e. spatial periods equal to the whole and a half of the circumference of the Earth— and Yanai waves with $l = 4$ [4]. The occurrence and persistence of these waves raise a number of important questions:

a) Why are these particular waves selected?

b) Why do they not dissipate away? The Kelvin wave, in particular, should rapidly generate shocks, which dissipate energy very efficiently. In addition, one would expect a significant amount of energy transfer to shorter waves, particularly to other Kelvin, Yanai and Rossby modes, and to the faster Poincare waves.

c) Do these various waves interact nonlinearly? And do they feel the equatorial topography and the distribution of land and sea?

In this work, we present a reduced model which sheds light on these questions. In its simplest version, the model has only two waves: an $l = 1$ Kelvin and an $l = 2$ Yanai, interacting through the $l = 3$ mode of the topography, which roughly corresponds to the distribution of the continents. However, the model can be straightforwardly extended to include a larger set of very long waves, which realistically should probably include two Kelvin modes ($l = 1$ and 2) and two to four Yanai and Rossby modes.

For clarity, we present the model in the simplest possible scenario of the β-plane one-layer shallow-water equations with topography and no mean flow. All these simplifying hypothesis can be relaxed, however, in order to build a more realistic model.

The Reduced Model

Basic Equations, Nondimensionalization, and Linear Waves

We start with the non-dimensionalized equations for long waves in a single layer of fluid of constant density in the equatorial waveguide:

$$\begin{aligned} \eta_t + [(1+\eta-h)u]_x + [(1+\eta-h)v]_y &= 0 && (1)\\ u_t + uu_x + vu_y + \eta_x - yv &= 0 && (2)\\ v_t + uv_x + vv_y + \eta_y + yu &= 0. && (3) \end{aligned}$$

We shall propose an ansatz for the solution to (1,2, 3) in which the dependent variables η, u and v and the topography h are all small and of the same order of magnitude ϵ . The linear solutions to the equations above can be divided into two groups: a nondispersive Kelvin wave and an infinite set of dispersive waves, the Rossby, Yanai and Poincaré waves. The Kelvin wave is given by

$$\begin{aligned} \eta &= K(x-t)e^{-\frac{y^2}{2}} && (4)\\ u &= K(x-t)e^{-\frac{y^2}{2}} && (5)\\ v &= 0, && (6) \end{aligned}$$

where K is an arbitrary function. The dispersive waves have the general form

$$\begin{aligned} \eta &= \left[\frac{y}{k-\omega}H_n(y) + \frac{\omega}{\omega^2-k^2}H_n'(y)\right] e^{i(kx-\omega t)}e^{-\frac{y^2}{2}} && (7)\\ u &= \left[\frac{y}{k-\omega}H_n(y) + \frac{k}{\omega^2-k^2}H_n'(y)\right] e^{i(kx-\omega t)}e^{-\frac{y^2}{2}} && (8)\\ v &= iH_n(y)e^{i(kx-\omega t)}e^{-\frac{y^2}{2}}, && (9) \end{aligned}$$

where $H_n(y)$ is the Hermite polynomial of order n, and $\omega = W(k)$ satisfies the dispersion relation

$$-2n + \frac{(\omega+k)(\omega^2-k\omega-1)}{\omega} = 0. \qquad (10)$$

Resonances between Long Waves through Topography

Since the dispersion relations for equatorially trapped waves, are far from self-similar, the nature of resonant interactions depends strongly on the order of magnitude of the wavenumbers in play. We shall concentrate on very long waves, with spatial periods comparable to the circumference of the Earth $P = 40000$ km. In order to see what this means for the wavenumbers k, we need to express P in the spatial unit of our nondimensionalization, i.e. L, which depends on the characteristic speed c.

A Kelvin wave $K(x-t)$, for instance, can be written as a Fourier series

$$K(x-t) = \sum_{l=1}^{\infty} \hat{K}(l) e^{i\alpha l(x-t)},$$

where $\alpha = \frac{2\pi L}{P}$. A wave with $l = 1$, i.e., with period equal to the circumference of the Earth, has a wavenumber $k = \alpha$, and all other wavenumbers are integer multiples of this.

As a Kelvin wave interacts with the topography, it can generate other waves through three-mode resonance. If we denote by k_K, k_T and k^* the wavenumbers of the Kelvin wave, the topography and the dispersive wave, the conditions for resonance are

$$\begin{aligned} k_K + k_T &= k^* && (11) \\ k_K &= \omega^* = W(k^*), && (12) \end{aligned}$$

where we have used the facts that for the Kelvin wave $\omega = k$, and that the topography is time-independent. Since all the wavenumbers have to be multiples of α, the nature of the possible interactions depends on its numerical value. In the next two subsections, we study two typical cases corresponding to the barotropic and first baroclinic modes of the tropical atmosphere.

The Barotropic Mode (Two Wave Interaction)

Let us consider first the case in which c is approximately 300m/s, a typical value for the barotropic mode, so α is close to 0.57. Since the Kelvin wave cannot have a positive wavenumber k_K smaller than $\alpha > 0.3$, we conclude from inspection of the dispersion relation that no Rossby waves can result from the interaction of a Kelvin wave with topography, and the only wave that can be generated by the first two modes of a Kelvin wave is the $k = -1.14$ ($l = -2$) Yanai mode, generated by the $l = 1$ Kelvin wave and the $l = 3$ mode of the topography.

Based on the argument above, we shall propose an ansatz with only two waves, a Kelvin wave and a Yanai mode with $l = -2$. The resulting equations are

$$\begin{aligned} K_\tau + \left(\frac{1}{2}K^2\right)_\theta &= Y(\tau)e^{i\theta} + \bar{Y}(\tau)e^{-i\theta} && (13) \\ Y_\tau &= -\hat{K}(1,\tau) && (14) \end{aligned}$$

where $K(\theta,\tau)$ and $Y(\tau)$ stand for the Kelvin wave and the $l = -2$ mode of Yanai respectively, $\theta = (x - ct)/P$ represents the linear phase of the Kelvin wave, normalized so that it has period 2π, and $\tau = \epsilon t$ represents the slow nonlinear time. The dependent variables K and Y and the slow time τ have been further rescaled so as to normalize to one the coefficients of the nonlinear interaction on the right-hand side, which depend on the projection of the (zonal) mode $l = 3$ of the topography on the (longitudinal) first Hermite polynomial.

Equation (13) for the evolution of the Kelvin wave is a Hopf –or inviscid Burgers– equation, forced on the right-hand side by the interaction of the Yanai wave with the topography. Equation (14), on the other hand, is an ordinary differential equation for the evolution of the Yanai wave, forced by the interaction of the Kelvin wave with the topography.

Properties of the Model

The reduced models discussed above have a number of distinctive features.

Conserved Quantities and Symmetries

The equations (13,14) have two main conserved quantities, the mean of K and the total energy.

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{-\pi}^{\pi} K(\theta,\tau)\,\mathrm{d}\theta = 0. \tag{15}$$

The energy

$$E = \int_0^{2\pi} \frac{K^2}{2}\,\mathrm{d}\theta \; + 2\pi|Y|^2 \tag{16}$$

is conserved while the solution remains smooth. However, energy may be dissipated at shocks.

Traveling Wave Solutions

Equations (13) and (14) have a family of exact solutions where K is a traveling wave. To see this, we propose an ansatz in which

$$K(\theta,\tau) = F(\theta - s\tau)$$

where s is an arbitrary constant.

$$F(z) = s \pm 2\sqrt{-\frac{2\hat{F}_1}{s}}\,|\cos(z/2)\,. \tag{17}$$

$$s = \mp\frac{4}{\pi}\sqrt{\frac{2}{3}} \sim 1.04\,.$$

Thus the wave with a downward peak moves more slowly than a linear Kelvin wave, and the one with the upward peak moves faster The total energy corresponding to this exact solution is readily computed; its value is

$$E = \frac{14\pi}{9} - \frac{32}{3\pi} \sim 1.5\,.$$

References

[1] P. Ripa: *On the theory of non linear wave-wave interactions among geophysical waves.* J. Fluid Mech. **103** (1981) 87–115.

[2] Kouski Wallace: *Observational evidence of Kelvin waves in the tropical stratosphere.* J. Atmos. Sci. **25** (1968) 900–907.

[3] S. Yoden, J. R. Holton: *A new look at Equatorial Quasi-Biennial Oscillation Models.* J. Atmos. Sci. **45** (1988) 2703–2717.

[4] Maruyama Yanai: *Stratospheric wave disturbances propagating over the equatorial Pacific.* J. Meteorol. Soc. Jpn. **44** (1966) 291–294.

[5] G. Strang: *On the Construction and Comparison of Difference Schemes.* SIAM J. Num. Anal. **5** (1968) 506–517.