

Étude statistique de l'activité de la fonction de sélection dans l'algorithme de E-méthode

Romain Michard, Nicolas Veyrat-Charvillon, Arnaud Tisserand

► **To cite this version:**

Romain Michard, Nicolas Veyrat-Charvillon, Arnaud Tisserand. Étude statistique de l'activité de la fonction de sélection dans l'algorithme de E-méthode. RR-5574, INRIA. 2005, pp.10. inria-00070432

HAL Id: inria-00070432

<https://hal.inria.fr/inria-00070432>

Submitted on 19 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

*Étude statistique de l'activité de la fonction de
sélection dans l'algorithme de E-méthode*

Romain Michard, Arnaud Tisserand et Nicolas Veyrat-Charvillon

N° 5574

Mai 2005

Thème SYM



*Rapport
de recherche*

Étude statistique de l'activité de la fonction de sélection dans l'algorithme de E-méthode

Romain Michard, Arnaud Tisserand et Nicolas Veyrat-Charvillon

Thème SYM — Systèmes symboliques
Projet Arénaire

Rapport de recherche n° 5574 — Mai 2005 — 10 pages

Résumé : Ce travail porte sur l'étude statistique de l'activité liée à la fonction de sélection dans l'algorithme d'approximation de polynômes connu sous le nom de E-méthode proposé par M. Ercegovac dans [3, 5]. La latitude de choix dans la fonction de sélection des chiffres du résultat, en représentation redondante, permet d'envisager de limiter l'activité électrique dans certains cas. Cet article présente un début d'étude sur les gains envisageables dans ce cadre.

Mots-clés : arithmétique des ordinateurs, basse consommation d'énergie, évaluation de polynôme, E-méthode.

Statistical study of the activity due to the selection function in the E-method algorithm

Abstract: This work is a statistical study of the activity due to the selection function in the polynomial approximation algorithm called E-method and proposed by M. Ercegovic in [3, 5]. The latitude in the choice of the result digits in the selection function, when using a redundant representation, allows us to consider a reduced electrical activity in some cases. This article presents the beginning of a study on the profits in such a situation.

Key-words: computer arithmetic, low-power consumption, polynomial evaluation, E-method.

1 Introduction

Dans bon nombre d'applications en traitement du signal et des images, en calcul scientifique ou en contrôle numérique, il est nécessaire d'évaluer des fonctions plus ou moins compliquées en utilisant uniquement des opérateurs simples comme l'addition et la multiplication. Les fonctions algébriques ($1/x, x/y, \sqrt{x} \dots$) et les fonctions élémentaires ($\sin(x), \cos(x), \exp(x), \log(x), \arctan(x) \dots$) s'approchent assez bien en utilisant des polynômes [9] déterminés, par exemple, avec l'algorithme de Remes [10]. Des approximations polynomiales des principales fonctions utilisées en calcul scientifique peuvent être trouvées dans [7, 9].

La E-méthode, proposée par M. Ercegovac [3, 5], permet d'évaluer des polynômes avec une itération à base d'additions et de décalages proche de celle utilisée pour la division ou dans l'algorithme CORDIC [11, 12]. Dans cet algorithme, les chiffres du résultat sont déterminés de façon itérative (un chiffre à chaque itération) par la fonction de sélection à partir des chiffres précédents et d'un résidu (équivalent au reste partiel dans la division). Les chiffres issus de la fonction de sélection, représentés en notation redondante, offrent une certaine latitude de choix. Dans ce travail, nous étudions l'activité, d'un point de vue statistique, impliquée par la fonction de sélection. Nous proposons une ébauche de méthode permettant de réduire cette activité en utilisant la connaissance du chiffre du résultat sélectionné à la dernière itération.

La section 2 décrit rapidement l'algorithme de E-méthode proposé par Ercegovac. Nous présentons une fonction de sélection avec mémorisation du chiffre de l'itération précédente à la section 3. Dans la section 4, nous présentons les résultats statistiques des simulations fonctionnelles sur l'activité moyenne de la fonction de sélection. Enfin, nous concluons et présentons quelques perspectives à la section 5.

2 Présentation de la E-méthode

La méthode d'évaluation, ou E-méthode, a été proposée par M. Ercegovac dans les années 70 [3, 5]. Cette méthode permet de résoudre certains systèmes linéaires, à diagonale dominante, à l'aide d'une itération simple et régulière à base d'additions et de décalages. Les systèmes linéaires cibles sont de la forme présentée à l'équation (1). Le principal intérêt de cette méthode est de permettre d'évaluer des polynômes sans aucun multiplieur en utilisant un algorithme à base d'additions et de décalages voisin de l'algorithme de division SRT [4, 6]. Ceci permet donc d'envisager des opérateurs de petite taille permettant de limiter la consommation d'énergie statique. Il existe d'autres algorithmes permettant d'évaluer des fonctions algébrique et/ou élémentaires sans multiplieurs comme la méthode des tables multipartites [2]. Mais, en général, ces méthodes sont dédiées à une fonction donnée.

On note \mathcal{A} la matrice de ce système, b le vecteur second membre et y le vecteur solution. La taille du système est $n + 1$. Après résolution du système (1), la première composante du vecteur solution y est la valeur au point x du polynôme P , de degré n , dont les coefficients sont les composantes de b . C'est à dire que la solution de $\mathcal{A}y = b$ est $y = [y_0, y_1, \dots, y_n]^t$

$$\begin{pmatrix}
 1 & -x & 0 & \cdots & & & 0 \\
 0 & 1 & -x & 0 & \cdots & & 0 \\
 0 & 0 & 1 & -x & 0 & \cdots & 0 \\
 & & \ddots & \ddots & \ddots & \ddots & \vdots \\
 & & & \ddots & \ddots & \ddots & 0 \\
 \vdots & & & & \ddots & \ddots & 0 \\
 0 & & \cdots & & & 1 & -x \\
 & & & & & 0 & 1
 \end{pmatrix}
 \begin{pmatrix}
 y_0 \\
 y_1 \\
 y_2 \\
 \vdots \\
 \vdots \\
 \vdots \\
 y_{n-1} \\
 y_n
 \end{pmatrix}
 =
 \begin{pmatrix}
 p_0 \\
 p_1 \\
 p_2 \\
 \vdots \\
 \vdots \\
 \vdots \\
 p_{n-1} \\
 p_n
 \end{pmatrix}
 \quad (1)$$

telle que

$$y_0 = P(x) = p_n x^n + p_{n-1} x^{n-1} + \cdots + p_0 \quad (2)$$

On trouve dans [3, 5] les limites sur les valeurs possibles pour les coefficients de $P(x)$ et sur l'argument x . Des techniques de mise à l'échelle permettent de limiter l'impact de ces contraintes [1].

Avant de présenter l'itération de la E-méthode, nous devons introduire quelques notations utiles pour la suite. La base du système de représentation des nombres est notée β (en pratique, $\beta = 4$ dans ce travail, mais les résultats peuvent être généralisés dans d'autres bases). Le vecteur des résidus, de taille $n + 1$, est noté w . Les différentes valeurs d'une quantité dans le temps sont représentées avec la notation crochet (comme en traitement du signal). Par exemple $w[j]$ dénote le vecteur des résidus à la j ème itération. Le vecteur de chiffres du résultat trouvé à chaque itération j est noté $d[j]$. En base 4, par exemple, ces chiffres sont dans l'ensemble $\{-3, -2, -1, 0, 1, 2, 3\}$.

L'algorithme de E-méthode est présenté en figure 1. Le vecteur des résidus est initialisé avec les coefficients du polynôme P (les composantes de b). Le premier vecteur de chiffres du résultat est le vecteur nul.

```

1  initialisation :
2    w[0] ← b
3    d[0] ← 0
4  itération :
5    pour j de 1 a m faire
6      w[j] ← β(w[j-1] - Ad[j-1])
7      d[j] ← S(w[j])
8  résultat :
9    y_0[m] = ∑_{i=1}^m d_0[i]β^{-i}

```

FIG. 1 – Algorithme d'évaluation d'un polynôme avec la E-méthode (version vectorielle).

Comme tous les algorithmes à base d'addition et de décalages, la E-méthode produit un chiffre du résultat à chaque itération en commençant par les poids forts. La concaténation des différents chiffres fournit une valeur qui tend vers la valeur mathématique du résultat (à l'infini). Ici, le résultat de chaque itération est un vecteur de chiffres $d[j]$. En pratique, le calcul est décomposé en étages, où le calcul correspondant à une ligne du système linéaire (1) est effectué par un étage. Les chiffres du vecteurs $d[j]$ se propagent donc sur les étages et le résultat final $P(x)$ est la sortie du dernier étage. L'itération est basée sur un calcul similaire à celui d'une division où l'on "diviserait" par la matrice \mathcal{A} (w est analogue à un reste partiel). Pour chaque ligne $i = 0, \dots, n-1$ de la matrice \mathcal{A} , le calcul effectué pour chaque composante est :

$$w_i[j] = \beta(w_i[j-1] - d_i[j-1] + d_{i+1}[j-1]x) \quad (3)$$

Dans le cas $i = n$, le calcul se simplifie en $w_n[j] = \beta(w_n[j-1] - d_n[j-1])$.

On note dans l'équation (3) que le seul produit $d_{i+1}[j-1]x$ est en fait une multiplication d'un chiffre $d_{i+1}[j-1]$ par un nombre x . En pratique, cette petite "multiplication" est faite en sélectionnant le bon multiple de x à ajouter parmi les différents multiples possibles. Ceci est possible car le multiplicande x est constant pendant toute la durée de l'algorithme.

Le calcul des nouveaux termes du résidu n'implique que des additions/soustractions et des produits d'un nombre par un seul chiffre (et petit, ici il est inférieur à 3). A chaque itération, un nouveau vecteur de chiffres du résultat $d[j]$ est produit. Ce calcul se fait en utilisant la fonction de sélection S définie dans un cadre général par l'expression (4), où ρ est la valeur maximum autorisée pour les chiffres du résultat en notation redondante (ici $\rho = 3$). Des détails peuvent être trouvés dans [3, 5].

$$S(x) = \begin{cases} \text{signe } x \lfloor |x| + 1/2 \rfloor, & \text{si } |x| \leq \rho \\ \text{signe } x \lfloor |x| \rfloor, & \text{sinon,} \end{cases} \quad (4)$$

3 Fonction de sélection avec mémoire

Afin de limiter l'activité de l'addition du calcul de $w[j] \leftarrow \beta(w[j-1] - Ad[j-1])$, nous proposons une nouvelle fonction de sélection avec mémoire. Pour les plages de valeurs où la fonction de sélection peut retourner deux valeurs pour le nouveau chiffre du résultat (à l'itération j), on retourne le dernier chiffre utilisé (c.a.d. à l'itération $j-1$) si ce choix est possible (illustration en figure 2,3).

La simplicité de la fonction de sélection permet de l'implanter sous la forme d'un opérateur moins coûteux qu'une table. La fonction de sélection sans mémoire se sert de la redondance pour n'utiliser que la partie entière $E(w)$ du résidu et le premier bit de sa partie fractionnaire $F(w)$. Dans la sélection avec mémoire, on conserve la redondance pour offrir une latitude sur le choix du chiffre, afin de privilégier les cas qui minimisent l'activité ($Sel(w[j]) = Sel(w[j-1])$). Il est alors nécessaire de prendre en compte plus de bits de $F(w)$ (3 pour nos paramètres) lorsque l'on doit borner celui-ci.

FIG. 3 – Fonction de sélection sans (a) et avec (b) mémoire pour $\beta = 4$ et les chiffres dans $\{-3, -2, 1, 0, 1, 2, 3\}$.

FIG. 4 – Redondance possible pour $\beta = 4$ et les chiffres dans $\{-3, -2, -1, 0, 1, 2, 3\}$.

Nous avons écrit un programme réalisant les différents calculs de l'algorithme donné ci-dessus et permettant d'extraire des statistiques sur l'utilisation de la fonction de sélection. Ces statistiques sont faites sur 500 évaluations en différents points avec une précision de 16 bits et un polynôme de degré 4. La table 2 présente ces statistiques. Pour chaque ligne de la table, le chiffre du résultat $d_i[j]$ est sélectionné un certain nombre de fois suivant le chiffre du résultat à l'itération précédente $d_i[j - 1]$ en colonne. Les résultats sont normalisés sur le produit du nombre d'itérations (la précision) et du degré du polynôme. La case en (0,0)

dépasse la valeur 1 car le chiffre 0 apparaît de très nombreuses fois (par exemple lors de l'initialisation) et en particulier, il peut apparaître sur plusieurs lignes de la matrice pour une itération donnée.

		$d_i[j-1]$						
		-3	-2	-1	0	1	2	3
$d_i[j]$	-3	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	-2	0.01	0.02	0.01	0.02	0.03	0.03	0.01
	-1	0.00	0.01	0.10	0.34	0.25	0.02	0.00
	0	0.00	0.01	0.02	2.60	0.45	0.01	0.00
	1	0.03	0.02	0.26	0.58	0.25	0.00	0.01
	2	0.00	0.01	0.04	0.04	0.02	0.00	0.00
	3	0.00	0.01	0.00	0.00	0.00	0.01	0.00

TAB. 2 – Statistiques de la sélection standard.

La table 3 présente les statistiques obtenues pour les mêmes paramètres et valeurs qu'à la section précédente en utilisant la fonction de sélection à mémoire. On constate bien que le nombre de cas où le chiffre du résultat choisi est k à l'itération j alors qu'il valait déjà k à l'itération $j-1$ est augmenté (ce sont les cases de la diagonale). En particulier, la fréquence du cas (0,0) est augmentée ce qui est une bonne chose pour limiter l'activité due au calcul et l'activité parasite. On peut donc espérer un gain sur l'activité dans l'étage de calcul du fait que ce chiffre ne change pas et l'argument x est aussi constant pour chaque point d'évaluation.

		$d_i[j-1]$						
		-3	-2	-1	0	1	2	3
$d_i[j]$	-3	0.00	0.00	0.00	0.02	0.01	0.00	0.00
	-2	0.00	0.01	0.00	0.01	0.07	0.00	0.00
	-1	0.00	0.00	0.02	0.03	0.02	0.02	0.02
	0	0.02	0.00	0.02	3.00	0.31	0.01	0.07
	1	0.00	0.00	0.00	0.53	0.49	0.01	0.03
	2	0.00	0.00	0.03	0.17	0.03	0.14	0.01
	3	0.00	0.00	0.02	0.08	0.01	0.02	0.01

TAB. 3 – Statistiques de la sélection avec mémoire.

5 Conclusion et perspectives

Ce travail sur l'activité de la fonction de sélection dans la E-méthode montre qu'il est possible de diminuer la consommation de l'opérateur. Maintenant, il est clair qu'il ne s'agit

que d'une étude statistique. En particulier, en pratique le léger surcoût lié aux stockage des chiffres $d_i[j - 1]$ (2 à 4 bits suivant la base) et du comparateur va augmenter un peu la surface du circuit.

Nous comptons implanter notre solution sur des technologies ASIC pour pouvoir effectuer des simulations au niveau électrique dans un avenir proche. Nous pensons que cet algorithme est intéressant d'un point de vue de la consommation d'énergie car il présente des implantations avec des surfaces bien moindres qu'en utilisant des solutions à base de multiplieurs. Ceci peut donc être un atout pour les technologies avec des courants de fuites importants. Enfin, nous pensons intégrer cet algorithme dans notre générateur de circuits `divgen` [8].

Références

- [1] N. Brisebarre and J.-M. Muller. Functions approximable by e-fractions. In *38th Conference on signals, systems and computers*, Pacific Grove, California, US, November 2004.
- [2] F. de Dinechin and A. Tisserand. Some improvements on multipartite tables methods. In N. Burgess and L. Ciminiera, editors, *15th International Symposium on Computer Arithmetic ARITH15*, pages 128–135, Vail, Colorado, June 2001. IEEE.
- [3] M. D. Ercegovac. *A general method for evaluation of functions and computation in a digital computer*. PhD thesis, Dept. of Computer Science, University of Illinois, Urbana-Champaign, 1975.
- [4] M. D. Ercegovac and T. Lang. *Division and Square-Root Algorithms : Digit-Recurrence Algorithms and Implementations*. Kluwer Academic, 1994.
- [5] M.D. Ercegovac. A general hardware-oriented method for evaluation of functions and computations in a digital computer. *IEEE Transactions on Computers*, C-26(7) :667–680, 1977.
- [6] M.D. Ercegovac, J.M. Muller, and A. Tisserand. FPGA implementation of polynomial evaluation algorithm. In SPIE, editor, *Field Programmable Gate Arrays for Fast Board Development and Reconfigurable Computing*, volume 2607, pages 177–188, October 1995.
- [7] J.F. Hart. *Computer Approximations*. Wiley, 1968.
- [8] R. Michard, A. Tisserand, and N. Veyrat-Charvillon. Divgen : a divider circuit generator. <http://lipforge.ens-lyon.fr/>, 2004.
- [9] J.-M. Muller. *Elementary Functions : Algorithms and Implementation*. Birkhäuser, Boston, 1997.
- [10] E. Remes. Sur un procédé convergent d'approximations successives pour déterminer les polynômes d'approximation. *C.R. Acad. Sci. Paris*, 198 :2063–2065, 1934.
- [11] J. Volder. The CORDIC computing technique. *IRE Transactions on Computers*, EC-8(3) :330–334, 1959.

- [12] J. Walther. A unified algorithm for elementary functions. In *Joint Computer Conference Proceedings*, 1971. Reprinted in E. E. Swartzlander, *Computer Arithmetic*, Vol. 1, IEEE Computer Society Press Tutorial, Los Alamitos, CA, 1990.



Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399