Επίλυση Αλγεβρικών Συστημάτων Μικρής Διάστασης στους Πραγματικούς

Δημήτρης Διώχνος

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών,
Μετ/κό πρόγραμμα Λογικής, Θεωρίας Αλγορίθμων και Υπολογισμού

Περίληψη: Ασχολούμαστε με επίλυση καλώς ορισμένων πολυωνυμικών συστημάτων με ακέραιους συντελεστές. Χρησιμοποιούμε πολυωνυμικές ακολουθίες υπολοίπων και τον αλγόριθμο του Sturm. Οι λύσεις των μεθόδων μας είναι πραγματικοί αλγεβρικοί αριθμοί που αναπαρίστανται υπό μορφή διαστημάτων απομόνωσης. Το προηγούμενο φράγμα με αυτή την αναπαράσταση ήταν $\widetilde{\mathcal{O}}_B(N^{30})$ και το βελτιώνουμε σε $\widetilde{\mathcal{O}}_B(N^{12})$. Τέλος, υλοποιήσαμε και τις τρεις προτεινόμενες μεθόδους σε MAPLE και παραθέτουμε πειράματα για την αποδοτικότητά τους σε σχέση με άλλες δημοφιλείς υλοποιήσεις.
Λέξεις κλειδιά: επίλυση στους πραγματικούς, πολυωνυμικά συστήματα, αλγόριθμος Sturm, διαστήματα απομόνωσης.

1 Εισαγωγή

Το πρόβλημα επίλυσης καλώς ορισμένων αλγεβρικών συστημάτων είναι καίριας σημασίας. Οι περισσότεροι αλγόριθμοι ασχολούνται με τη γενική περίπτωση ή αναζητούν λύσεις σε κλειστά αλγεβρικά σώματα [3, 16, 21, 22, 20]. Η παρούσα διπλωματική βασίζεται στα αποτελέσματα [5, 6] και επικεντρώνεται στην επίλυση στους πραγματικούς σε διμετάβλητα συστήματα. Έτσι βρίσκουμε ακριβή φράγματα πολυπλοκότητας γι' αυτή την περίπτωση και μελετούμε διάφορους αλγορίθμους στην πράξη. Οι αλγόριθμοι είναι επέκταση των αντίστοιχων μεθόδων στο [31]. Σχετικές εργασίες είναι [15, 9, 23]. Εναλλακτικές προσεγγίσεις βρίσκονται στα [19, 26]. Τέλος, ο προσδιορισμός της τοπολογίας μιας πραγματικής αλγεβρικής καμπύλης πρέπει να υπολογίσει εν κατακλείδι τις πραγματικές λύσεις ενός συστήματος δύο εξισώσεων σε δύο μεταβλητές, δείτε [2, 10, 34, 13, 1].

2 Προκαταρκτικά

Σε ότι ακολουθεί, με $\mathcal{O}_B$ εννοούμε πολυπλοκότητα bit και με $\widetilde{\mathcal{O}}_B$ αγνοούμε (πολυ)-λογαριθμικούς παράγοντες. Για $f \in \mathbb{Z}[y_1, \ldots, y_k, x]$, το $\mathsf{deg}(f)$ δείχνει το συνολικό βαθμό, ενώ το $\deg_x(f)$ δηλώνει το βαθμό του πολυωνύμου ως προς x. Με $\mathcal{L}(f)$ φράσσουμε το δυαδικό μήκος των συντελεστών του f (συμπεριλαμβάνοντας ένα bit για το πρόσημο). Υποθέτουμε $\mathcal{L}(\mathsf{deg}(f)) = \mathcal{O}(\mathcal{L}(f))$.

*Επιβλέπων: Ιωάννης Ζ. Εμίρης. Συνεργασία με τον Ηλία Τσιγαρίδα.

2.1 Πραγματικοί Αλγεβρικοί Αριθμοί

Επιλέγουμε να αναπαραστήσουμε τους πραγματικούς αλγεβρικούς αριθμούς $\alpha \in \mathbb{R}_{alg}$ με αναπαράσταση *διαστημάτων απομόνωσης* (*isolating interval representation*). Η αναπαράσταση περιλαμβάνει ένα χωρίς-τετράγωνα (square-free) πολυώνυμο f το οποίο έχει ρίζα το α και ένα διάστημα με άκρα ρητούς το οποίο περιέχει το α και *καμία άλλη ρίζα*. Αν α η μοναδική ρίζα του f στο διάστημα $\mathfrak{I} = [I_L, I_R]$, όπου $I_L, I_R \in \mathbb{Q}$, το συμβολίζουμε ως: $\alpha \simeq [f, \mathfrak{I}] = [f, [I_L, I_R]]$.

2.2 Πολυωνυμικές Ακολουθίες Υπολοίπων

Καίρια σημασία στις μεθόδους μας έχει ο υπολογισμός του ΜΚΔ δύο πολυωνύμων. Στις εφαρμογές ενδιαφερόμαστε για τις ρίζες του ΜΚΔ. Αρκεί λοιπόν να υπολογιστεί ο ΜΚΔ μέχρι ομοιότητα. Έτσι, χρησιμοποιούμε ακολουθίες υπολοίπων που *μοιάζουν* με την ακολουθία που προκύπτει από την ψευδο-ευκλείδια διαίρεση των δύο πολυωνύμων. Οι παραλλαγές που χρησιμοποιούνται στην πράξη είναι οι *προσημασμένες* ακολουθίες υπολοίπων (βλ. [2, 33, 35] και αναφορές εκεί). Εδώ ασχολούμαστε με προσημασμένες Subresultant και Sturm-Habicht ακολουθίες ($\mathbf{SR}(f,g)$ και $\mathbf{StHa}(f,g)$ αντίστοιχα). Έτσι υπολογίζουμε μια ακολουθία που μοιάζει με την $R_0 = f, R_1 = g, R_2 = -\,\mathsf{prem}\,(f,g), \ldots, R_k = -\,\mathsf{prem}\,(R_{k-2}, R_{k-1})$, όπου $\mathsf{prem}\,(R_i, R_{i+1})$ το υπόλοιπο της αντίστοιχης ευκλείδιας ψευδο-διαίρεσης και $\mathsf{prem}\,(R_{k-1}, R_k) = 0$. Στην περίπτωσή μας, το πολυώνυμο g είναι η παράγωγος του f· βλ. [11, 2]. Ακολουθούν τα πιο σημαντικά αποτελέσματα σχετικά με τον υπολογισμό και την αποτίμηση αυτών των ακολουθιών. Με $\mathbf{sr}(f,g)$ συμβολίζουμε την ακολουθία των πρωτεύοντων συντελεστών υποαπαλοιφουσών (*principal subresultant coefficients*), με $\mathbf{SRQ}(f,g)$ τη μπότα πηλίκο (quotient boot) και με $\mathbf{SR}(f,g;\mathsf{a})$ την ακολουθία αποτιμημένη πάνω στο $\mathsf{a} \in \mathbb{Q}$.

Πρόταση 2.1. *[17, 18, 27] Έστω $p \geq q$. Η $\mathbf{SR}(f,g)$ υπολογίζεται σε χρόνο $\widetilde{\mathcal{O}}_B(p^2 q\tau)$ με $\mathcal{L}(\mathbf{SR}_j(f,g)) = \mathcal{O}(p\tau)$. Η μπότα πηλίκο, οποιοδήποτε πολυώνυμο στην $\mathbf{SR}(f,g)$, η απαλοίφουσα και ο ΜΚΔ υπολογίζονται σε χρόνο $\widetilde{\mathcal{O}}_B(pq\tau)$.*

Λήμμα 2.2. *[17, 27] Έστω $p \geq q$. Μπορούμε να υπολογίσουμε την $\mathbf{SR}(f,g;\mathsf{a})$, όπου $\mathsf{a} \in \mathbb{Q} \cup \{\pm\infty\}$ και $\mathcal{L}(\mathsf{a}) = \sigma$, σε χρόνο $\widetilde{\mathcal{O}}_B(pq\tau + q^2\sigma + p^2\sigma)$, όπου $\tau = \max\{\mathcal{L}(f), \mathcal{L}(g)\}$. Αν το $f(\mathsf{a})$ γνωστό, τότε το φράγμα γίνεται $\widetilde{\mathcal{O}}_B(pq\tau + q^2\sigma)$.*

Ορισμός 2.3. *Έστω L μια λίστα πραγματικών αριθμών. Με $\mathtt{VAR}(L)$ συμβολίζουμε το πλήθος (πιθανώς τροποποιημένων, δείτε [2, 11]) εναλλαγών προσήμου.*

Πόρισμα 2.4. *Για οποιαδήποτε f, g, το $\mathtt{VAR}(\mathbf{SR}(f,g;\mathsf{a}))$ υπολογίζεται σε χρόνο $\widetilde{\mathcal{O}}_B(pq\tau + \min\{p,q\}^2\sigma)$, δεδομένου πως το πρόσημο $\mathrm{sign}(f(\mathsf{a}))$ είναι γνωστό.*

2.3 Πολυώνυμα μίας μεταβλητής

Οι μέθοδοι της ενότητας 3 στηρίζονται στην επίλυση πολυωνύμων μιας μεταβλητής στους πραγματικούς. Εδώ θα αναφέρουμε τα σημαντικότερα αποτελέσματα τα οποία χρειαζόμαστε για τη συνέχεια. Για περισσότερα δείτε [8].

Πρόταση 2.5 (Αλγόριθμος Sturm)**.** *[7, 8] Έστω $f \in \mathbb{Z}[x]$ με βαθμό p και $\mathcal{L}(f) \leq \tau_f$. Μπορούμε να υπολογίσουμε τις πραγματικές ρίζες και τις πολλαπλότητες του f υπό μορφή διαστημάτων απομόνωσης σε χρόνο $\widetilde{\mathcal{O}}_B(p^6 + p^4\tau_f^2)$. Τα άκρα των διαστημάτων έχουν δυαδικό μήκος που φράσσεται από $\mathcal{O}(p^2 + p\,\tau_f)$ και $\mathcal{L}(f_{red}) = \mathcal{O}(p + \tau_f)$, όπου f_{red} το χωρίς-τετράγωνα μέρος της f.*

Πόρισμα 2.6 (Υπολογισμός προσήμου - SIGN_AT)**.** *[2, 8] Δοθέντος ενός πραγματικού αλγεβρικού αριθμού $\alpha \cong (f, [\mathsf{a}, \mathsf{b}])$, με $\mathcal{L}(\mathsf{a}) = \mathcal{L}(\mathsf{b}) = \mathcal{O}(p\tau_f)$, και ενός $g \in \mathbb{Z}[x]$, τέτοιου που $\deg(g) = q, \mathcal{L}(g) = \tau_g$, υπολογίζουμε το πρόσημο $\operatorname{sign}(g(\alpha))$ σε χρόνο (bit-πολυπλοκότητα) $\widetilde{\mathcal{O}}_B(pq\max\{\tau_f, \tau_g\} + p\min\{p, q\}^2\tau_f)$.*

Λήμμα 2.7 (Aggregate separation)**.** *Δοθέντος $f \in \mathbb{Z}[x]$, το άθροισμα των δυαδικών μηκών όλων διαστημάτων απομόνωσης των πραγματικών ριζών του f είναι $\mathcal{O}(p^2 + p\,\tau_f)$.*

Πόρισμα 2.8 (Intermediate Points)**.** *Δοθείσης μιας λίστας με τις πραγματικές ρίζες του f υπό μορφή διαστημάτων απομόνωσης, υπολογίζουμε ρητούς μεταξύ τους σε χρόνο $\widetilde{\mathcal{O}}_B(p^2 + p\tau_f)$.*

2.4 Πολυώνυμα πολλών μεταβλητών

Με πολυώνυμα σε πολλές μεταβλητές χρησιμοποιούμε την τεχνική της *δυαδικής κατάτμησης* (*binary segmentation*) [27]. Μια εναλλακτική προσέγγιση βρίσκεται στο [14]. Έστω $f, g \in (\mathbb{Z}[y_1, \ldots, y_k])[x]$ με $\mathcal{L}(f), \mathcal{L}(g) \leq \tau$, $\deg_x(f) = p \geq q = \deg_x(g)$, $\deg_{y_i}(f) \leq d_i$ και $\deg_{y_i}(g) \leq d_i$. Έστω ακόμη $d = \prod_{i=1}^{k} d_i$.

Πρόταση 2.9. *[27] Υπολογίζουμε την $\mathbf{SRQ}(f, g)$, οποιοδήποτε πολυώνυμο της $\mathbf{SR}(f, g)$ και την απαλοίφουσα $\mathtt{res}(f, g)$ σε χρόνο $\widetilde{\mathcal{O}}_B(q(p+q)^{k+1}d\tau)$.*

Λήμμα 2.10. *Η $\mathbf{SR}(f, g)$ υπολογίζεται σε χρόνο $\widetilde{\mathcal{O}}_B(q(p+q)^{k+2}d\tau)$.*

Θεώρημα 2.11. *Αποτιμούμε την $\mathbf{SR}(f, g)$ στο $x = \alpha$, όπου $\mathsf{a} \in \mathbb{Q} \cup \{\infty\}$ και $\mathcal{L}(a) = \sigma$, σε χρόνο $\widetilde{\mathcal{O}}_B(q(p+q)^{k+1}d\max\{\tau, \sigma\})$.*

Πόρισμα 2.12. *Υπολογίζουμε την $\mathbf{SR}(f, g)$ σε χρόνο $\widetilde{\mathcal{O}}_B(pq(p+q)^2d\tau)$. Για οποιοδήποτε πολυώνυμο $\mathbf{SR}_j(f, g)$ της $\mathbf{SR}(f, g)$, ισχύει $\deg_x(\mathbf{SR}_j(f, g)) = \mathcal{O}(\max\{p, q\})$, $\deg_y(\mathbf{SR}_j(f, g)) = \mathcal{O}(\max\{p, q\}d)$ και $\mathcal{L}(\mathbf{SR}_j(f, g)) = \mathcal{O}(\max\{p, q\}\tau)$.*

Πόρισμα 2.13. *Υπολογίζουμε την $\mathbf{SRQ}(f, g)$, οποιοδήποτε πολ/μο της $\mathbf{SR}(f, g)$, και την απαλοίφουσα $\mathtt{res}(f, g)$ σε χρόνο $\widetilde{\mathcal{O}}_B(pq\max\{p, q\}d\tau)$.*

Πόρισμα 2.14. *Υπολογίζουμε την $\mathbf{SR}(f, g\,;\, \mathsf{a})$, όπου $\mathsf{a} \in \mathbb{Q} \cup \{\infty\}$ και $\mathcal{L}(\mathsf{a}) = \sigma$, σε χρόνο $\widetilde{\mathcal{O}}_B(pq\max\{p, q\}d\max\{\tau, \sigma\})$. Για τα πολυώνυμα $\mathbf{SR}_j(f, g\,;\, \mathsf{a}) \in \mathbb{Z}[y]$, εκτός των f, g, έχουμε $\deg_y(\mathbf{SR}_j(f, g\,;\, \mathsf{a})) = \mathcal{O}((p+q)d)$ και $\mathcal{L}(\mathbf{SR}_j(f, g\,;\, \mathsf{a})) = \mathcal{O}(\max\{p, q\}\tau + \min\{p, q\}\sigma)$.*

2.4.1 Υπολογισμός προσήμου πολυων. δύο μεταβλητών

Ανάγουμε τον υπολογισμό του προσήμου του $f \in \mathbb{Z}[x, y]$ πάνω στο $(\alpha, \beta) \in \mathbb{R}^2_{alg}$ σε υπολογισμό προσήμου πάνω σε πολλά σημεία στο $\mathbb{Q}^2$. Έστω $\deg_x(f) = \deg_y(f) = n_1$, $\mathcal{L}(f) = \sigma$ και $\alpha \cong (A, [\mathsf{a}_1, \mathsf{a}_2])$, $\beta \cong (B, [\mathsf{b}_1, \mathsf{b}_2])$, όπου $A, B \in \mathbb{Z}[X]$, $\deg(A) = \deg(B) = n_2$, $\mathcal{L}(A) = \mathcal{L}(B) = \sigma$. Η ιδέα είναι πως υπολογίζουμε την ακολουθία $\mathbf{SR}(A, f)$ ως προς x, και δημιουργούμε δύο αντίγραφα. Αποτιμούμε το ένα αντίγραφο στο αριστερό άκρο a_1 και το άλλο αντίγραφο στο δεξί άκρο a_2, και κάθε μια ακολουθία αποτιμάται πάνω στο β με το πόρ. 2.6. Τελικά καταμετρώντας τις εναλλαγές προσήμου μπορούμε να αποφανθούμε για το πρόσημο της f πάνω στο (α, β). Έτσι ο αλγόριθμος γενικεύει την περίπτωση μιας μεταβλητής, [29, 8, 35] (πόρ. 2.6). Για τα A και B, υποθέτουμε πως ξέρουμε τις τιμές τους στα $\mathsf{a}_1, \mathsf{a}_2$ και $\mathsf{b}_1, \mathsf{b}_2$ αντίστοιχα.

Θεώρημα 2.15 (BIVARIATE-SIGN_AT). *Έστω $f \in \mathbb{Z}[x,y]$ τέτοιο που $\deg_x(f) = \deg_y(f) = n_1$ και $\mathcal{L}(f) = \sigma$ και δύο πραγμ. αλγ. αριθμοί $\alpha \cong (A, \mathcal{I}_\alpha) = [\mathsf{a}_1, \mathsf{a}_2]$, $\beta \cong (B, \mathcal{I}_\beta) = [\mathsf{b}_1, \mathsf{b}_2]$ όπου $A, B \in \mathbb{Z}[X]$, $\deg(A) = \deg(B) = n_2$, $\mathcal{L}(A) = \mathcal{L}(B) = \sigma$ και $\mathcal{I}_\alpha, \mathcal{I}_\beta \in \mathbb{Q}^2$. Τότε αποτιμούμε το πρόσημο του f πάνω στα α και β με πολυπλοκότητα $\widetilde{\mathcal{O}}_B(n_1^2 n_2^3 \sigma)$, υποθέτοντας $n_1 \le n_2$.*

3 Αλγόριθμοι

Εδώ παρουσιάζουμε τις μεθόδους μας για επίλυση 2×2 συστημάτων.

3.1 Αλγόριθμος GRID

Η μέθοδος GRID είναι η άμεση προσέγγιση, δείτε επίσης [9]. Υπολογίζουμε τις πραγματικές λύσεις ως προς x και ως προς y των απαλοιφουσών $\mathtt{res}_x(f,g)$ και $\mathtt{res}_y(f,g)$. Εν συνεχεία, τις ταιριάζουμε με τον αλγόριθμο BIVARIATE-SIGN_AT (θεώρ. 2.15) εξετάζοντας όλα τα παραλληλόγραμμα στο επαγόμενο πλέγμα. Η έξοδος είναι μια λίστα ζευγών πραγματικών αλγεβρικών αριθμών, η οποία αναπαρίσταται υπό μορφή διαστημάτων απομόνωσης. Τα άκρα ορίζουν παραλληλόγραμμα με μοναδική λύση στο εσωτερικό τους.

Η μέθοδος είναι ελκυστική μιας και είναι απλή, αλλά ο υπολογισμός προσήμου είναι πολύ ακριβός. Ο αλγόριθμος δεν απαιτεί γενική θέση· στη συνέχεια παρουσιάζουμε μια γενική μέθοδο στρέβλωσης η οποία φέρνει το σύστημα σε γενική θέση προκειμένου να υπολογιστούν και οι πολλαπλότητες στον ίδιο ασυμπτωτικό χρόνο. Ο αλγόριθμος επιτρέπει τη χρήση ευρετικών όπως είναι ο μικτός όγκος, ή η απαρίθμηση των ριζών σε συγκεκριμένη τετμημένη (βλ. ενότητα 4).

Θεώρημα 3.1. *Η απομόνωση όλων των πραγματικών ριζών του συστήματος $f = g = 0$ χρησιμοποιώντας τη μέθοδο* GRID *έχει πολυπλοκότητα $\widetilde{\mathcal{O}}_B(n^{14} + n^{13}\sigma)$, δεδομένου $\sigma = \mathcal{O}(n^3)$· ή $\widetilde{\mathcal{O}}_B(N^{14})$, όπου $N = \max\{n, \sigma\}$.*

Στη συνέχεια εξετάζουμε την πολλαπλότητα μιας ρίζας (α, β) του συστήματος. Παρόμοιες εργασίες περιλαμβάνουν [10, 30, 34]. Η μέθοδος μας ανάγεται σε διμετάβλητο υπολογισμό προσήμου και δεν απαιτεί παραγοντοποίηση.

3.1.1 Ντετερμινιστική στρέβλωση (Deterministic shear)

Βρίσκουμε μια επαρκή (οριζόντια) στρέβλωση τέτοια που η εξίσωση

$$R_t(x) = \mathtt{res}_y\left(f(x + ty, y), g(x + ty, y)\right), \tag{1}$$

έχει απλές ρίζες και οι οποίες αντιστοιχούν στις προβολές των λύσεων του συστήματος $f(x,y) = g(x,y) = 0$, όπου $t \mapsto t_0 \in \mathbb{Z}$, και ο βαθμός των πολυωνύμων παραμένει ο ίδιος. Για μια διαφορετική προσέγγιση δείτε [12, 2].

Λήμμα 3.2. *Ο υπολογισμός ενός $t_0 \in \mathbb{Z}$, τέτοιου που η αντίστοιχη στρέβλωση να είναι επαρκώς γενική, έχει πολυπλοκότητα $\widetilde{\mathcal{O}}_B(n^{10} + n^9\sigma)$.*

Θεώρημα 3.3. *Υπό τις προϋποθέσεις του θεωρήματος 3.1, έχοντας απομονώσει όλες τις πραγματικές ρίζες του $f = g = 0$, είναι πιθανό να προσδιορίσουμε τις πολλαπλότητές τους σε χρόνο $\widetilde{\mathcal{O}}_B(n^{12} + n^{11}\sigma + n^{10}\sigma^2)$.*

3.2 Ο αλγόριθμος M_RUR

Η μέθοδος M_RUR υποθέτει πως τα πολυώνυμα είναι σε Γενική Θέση: διαφορετικές ρίζες προβάλλονται σε διαφορετικές τετμημένες και οι συντελεστές ως προς y δεν έχουν κοινές ρίζες.

Πρόταση 3.4. *[10, 2] Έστω f, g πολυώνυμα πρώτα μεταξύ τους, σε γενική θέση. Αν $\mathbf{SR}_j(x,y) = \mathbf{sr}_j(x)y^j + \mathbf{sr}_{j,j-1}(x)y^{j-1} + \cdots + \mathbf{sr}_{j,0}(x)$, και (α, β) είναι μια πραγματική λύση του συστήματος $f = g = 0$, τότε υπάρχει k, τέτοιο που $\mathbf{sr}_0(\alpha) = \cdots = \mathbf{sr}_{k-1}(\alpha) = 0$, $\mathbf{sr}_k(\alpha) \neq 0$ και $\beta = -\frac{1}{k}\frac{\mathbf{sr}_{k,k-1}(\alpha)}{\mathbf{sr}_k(\alpha)}$.*

Αυτό εκφράζει την τεταγμένη μιας λύσης με ρητή αναπαράσταση πολυωνύμου μιας μεταβλητής (Rational Univariate Representation (RUR)) ως προς την τετμημένη. Η RUR εφαρμόζεται σε αλγεβρικά συστήματα πολλών μεταβλητών [4, 28, 2] και γενικεύει τη μέθοδο του Kronecker. Ο αλγόριθμός είναι παρόμοιος με τους [12, 10]. Τροποποιήσαμε τον αλγόριθμο [9], ώστε η έξοδος να περιλαμβάνει διαστήματα απομόνωσης, εξού και το όνομα τροποποιημένος-RUR (modified RUR - M_RUR). Η πιο σημαντική διαφορά με το [10] είναι πως εκείνοι χρησιμοποιούν αναπαράσταση Thom για τους πραγματικούς αλγεβρικούς αριθμούς. Προβάλλουμε στους άξονες x και y και για κάθε πραγματική λύση στον άξονα x υπολογίζουμε την τεταγμένη χρησιμοποιώντας την πρότ. 3.4. Αρχικά υπολογίζουμε την ακολουθία $\mathbf{SR}(f,g)$ ως προς y σε χρόνο $\widetilde{\mathcal{O}}_B(n^5\sigma)$ (πόρ. 2.12).

Η πρώτη φάση (προβολή) είναι παρόμοια με τον GRID. Η πολυπλοκότητα καθορίζεται από την επίλυση στους πραγματικούς των απαλοιφουσών, δηλαδή $\widetilde{\mathcal{O}}_B(n^{12} + n^{10}\sigma^2)$. Έστω α_i, αντίστοιχα β_j, να είναι οι πραγματικές συντεταγμένες. Υπολογίζουμε τους ρητούς q_j μεταξύ των β_j σε χρόνο $\widetilde{\mathcal{O}}_B(n^5\sigma)$, μέσω της συνάρτησης INTERMEDIATE_POINTS(P_y):

$$q_0 < \beta_1 < q_1 < \beta_2 < \cdots < \beta_{\ell-1} < q_{\ell-1} < \beta_\ell < q_\ell, \tag{2}$$

όπου $\ell \leq 2\,n^2$. Κάθε β_j αντιστοιχεί σε ένα *μοναδικό* α_i. Η πολλαπλότητα του α_i είναι η πολ/τα μιας πραγμ. λύσης του συστήματος που το έχει σαν τετμημένη. Εν συνεχεία, υπολογίζουμε σε χρόνο $\widetilde{\mathcal{O}}_B(n^9 + n^8\,\sigma)$ ένα k τέτοιο που η πρόταση 3.4 να ικανοποιείται, βλ. [24, 10]. Τέλος, εκμεταλλευόμαστε τη γενική θέση και την εξίσωση (2), ώστε να ταιριάξουμε τις πραγματικές λύσεις του R_x με αυτές του R_y, σε συνολικό χρόνο (όλες οι επαναλήψεις) $\widetilde{\mathcal{O}}_B(n^{10} + n^9\sigma)$.

Θεώρημα 3.5. *Απομονώνουμε όλες τις πραγματικές ρίζες του συστήματος $f = g = 0$, αν τα f, g είναι σε γενική θέση, με τον αλγόριθμο* M_RUR *σε χρόνο $\widetilde{\mathcal{O}}_B(n^{12} + n^{10}\sigma^2)$· η απλά $\widetilde{\mathcal{O}}_B(N^{12})$, όπου $N = \max\{n, \sigma\}$.*

Η υπόθεση γενικής θέσης είναι χωρίς βλάβη της γενικότητας αφού μπορούμε πάντα να τοποθετήσουμε το σύστημα σε τέτοια θέση εφαρμόζοντας στρέβλωση· δείτε ενότητα 3.1.1 και επίσης [2, 10]. Το δυαδικό μήκος των συντελεστών των πολυωνύμων του (στρεβλωμένου) συστήματος γίνεται $\widetilde{\mathcal{O}}(n+\sigma)$ [10] και δεν αλλάζει το φράγμα του θεωρ. 3.5. Απομένει να εκφραστούν οι πραγματικές λύσεις στο αρχικό σύστημα συντεταγένων· κάτι μη τετριμμένο στην πράξη.

3.3 Ο αλγόριθμος G_RUR

Σε αυτή την ενότητα παρουσιάζουμε έναν αλγόριθμο που χρησιμοποιεί μερικές ιδέες από τον M_RUR και στηρίζεται σε υπολογισμούς ΜΚΔ πολυωνύμων με συντελεστές σε σώμα επέκτασης προκειμένου να είναι αποδοτικός (εξού και το όνομα

G_RUR από το gcd). Για αυτούς τους υπολογισμούς ΜΚΔ χρησιμοποιούμε τον αλγόριθμο και την υλοποίηση σε MAPLE των [32].

Τα πρώτα βήματα είναι παρόμοια με αυτά των προηγούμενων αλγορίθμων: Προβάλλουμε στους άξονες, επιλύουμε στους πραγματικούς και υπολογίζουμε τα ενδιάμεσα σημεία στον άξονα y. Η μέθοδος έχει πολυπλοκότητα $\widetilde{\mathcal{O}}_B(n^{12}+n^{10}\sigma^2)$. Για κάθε τετμημένη x, έστω α, υπολογίζουμε το χωρίς τετράγωνα μέρος των $f(\alpha,y)$ και $g(\alpha,y)$, έστω $\bar{f}$ και $\bar{g}$. Υποθέτοντας ταχείς αλγορίθμους πολλαπλασιασμού, κάτι το οποίο στο [32] δεν είχε γίνει, επαναλαμβάνουμε αυτή τη διαδικασία $\mathcal{O}(n^2)$ φορές, με αποτέλεσμα το συνολικό κόστος να είναι $\widetilde{\mathcal{O}}_B(n^6\sigma^2)$. Τώρα για κάθε α, υπολογίζουμε το $h=\gcd(\bar{f},\bar{g})$ με συνολικό κόστος $\widetilde{\mathcal{O}}_B(n^8+n^6\sigma^2)$. Οι πραγματικές λύσεις του h αντιστοιχούν στις λύσεις του συστήματος με τετμημένη α. Η καρδιά της μεθόδου βρίσκεται στο γεγονός πως το h αλλάζει πρόσημο *μόνο* πάνω στο διάστημα το οποίο περιέχει τις πραγματικές του ρίζες. Προκειμένου να ελέγξουμε αυτά τα πρόσημα, αρκεί να αντικαταστήσουμε το y στο h από τα ενδιάμεσα σημεία, παίρνοντας έτσι ένα πολυώνυμο στο $\mathbb{Z}[\alpha]$, βαθμού $\mathcal{O}(n)$. Τώρα κοιτάμε αυτό το πολυώνυμο στον $\mathbb{Z}[x]$ και το αποτιμούμε πάνω στο α με κόστος $\widetilde{\mathcal{O}}_B(n^6+n^5\sigma+n^4 s_j)$. Αθροίζοντας πάνω σε $\mathcal{O}(n^2)$ σημεία και από το λήμμα 2.7 παίρνουμε $\widetilde{\mathcal{O}}_B(n^8+n^7\sigma)$, με συνολική πολυπλοκότητα $\widetilde{\mathcal{O}}_B(n^{10}+n^9\sigma)$.

Θεώρημα 3.6. *Απομονώνουμε τις πραγματικές λύσεις του συστήματος $f=g=0$, με τη μέθοδο* G_RUR *σε χρόνο $\widetilde{\mathcal{O}}_B(n^{12}+n^{10}\sigma^2)$· ή $\widetilde{\mathcal{O}}_B(N^{12})$, όπου $N=\max\{n,\sigma\}$.*

4 Εφαρμογές, Υλοποίηση και Πειράματα

Οι μέθοδοι που παρουσιάσαμε βρίσκουν εφαρμογή στην απαρίθμηση πραγμ. ριζών πολυωνύμων με συντελεστές σε σώμα επέκτασης, σε πολλές ανισώσεις με ακέραιους συντελεστές σε δύο μεταβλητές, καθώς επίσης και στην πολυπλοκότητα υπολογισμού τοπολογίας αλγεβρικής καμπύλης στον $\mathbb{R}^2$. Για περισσότερα δείτε [6, 5]. Στη συνέχεια παρουσιάζουμε την υλοποίησή μας ανοιχτού κώδικα σε MAPLE[1] και αναδεικνύει τις δυνατότητές του συγκριτικά με άλλα πακέτα. Παρέχουμε μεθόδους για *προσημασμένες* ακολουθίες υπολοίπων, επίλυση στους πραγματικούς μέσω του αλγορίθμου του Sturm, υπολογισμούς με έναν και δύο πραγματικούς αλγεβρικούς αριθμούς, όπως ο υπολογισμός προσήμου και η σύγκριση, και επίλυση 2×2 συστημάτων με ακέραιους συντελεστές.

4.1 Οι αλγόριθμοί μας

Εξετάσαμε τα πολυωνυμικά συστήματα που παρουσιάζονται στα [6, 5]. Τα συστήματα R_i, M_i, D_i είναι από το [9], τα C_i από [12] και τα $W_i, i=1,\ldots,4$, είναι τα C_i αφού εναλλάξουμε τους ρόλους των x,y. Για υπολογισμούς ΜΚΔ σε σώμα επέκτασης, χρησιμοποιήσαμε το [32]. Οι βέλτιστοι αλγόριθμοι για τον υπολογισμό και την αποτίμηση ακολουθιών υπολοίπων *δεν* έχουν υλοποιηθεί ακόμη.

Τα κύρια πειραματικά αποτελέσματά μας φαίνονται στον πίνακα 1 κάτω από τον τίτλο SLV (Sturm soLVer). Η απόδοση όλων των υλοποιήσεων είναι ο μέσος όρος 10 εκτελέσεων σε MAPLE 9.5 σε γραμμή εντολών σε έναν 2GHz AMD64@3K+ επεξεργαστή με 1GB RAM. Ο G_RUR είναι επικρατέστερος αφού είναι γρηγορότερος από τον GRID και τον M_RUR σε 17 από τις 18 περιπτώσεις. Ίσως αυτό να μην

[1] `www.di.uoa.gr/~erga/soft/SLV_index.html`

ισχύει όταν ο βαθμός στο σώμα επέκτασης είναι μεγάλος. Όλοι οι αλγόριθμοι χρησιμοποιούν φίλτρα προκειμένου να είναι αποδοτικοί. Το κυριότερο από αυτά είναι η αριθμητική διαστημάτων προκειμένου να αποφεύγουμε τις χρονοβόρες αποτιμήσεις ακολουθιών πολυωνύμων όπου αυτό είναι δυνατό. Για περισσότερες πληροφορίες δείτε το [6]. Αναλύοντας τους χρόνους παρατηρούμε σε γενικές γραμμές: ο GRID ξοδεύει περίπου το 73% του χρόνου εκτέλεσης στο ταίριασμα λύσεων, ο M_RUR το 45-50% του χρόνου εκτέλεσης στο ταίριασμα και ένα 24-27% στα φίλτρα και τέλος ο G_RUR το 55-80% του χρόνου στο ταίριασμα, συμπεριλαμβάνοντας το χρόνο για υπολογισμούς ΜΚΔ σε σώμα επέκτασης. Τα ποσοστά αυτά είναι ελαφρά αυξημένα όταν ο GRID και ο G_RUR επιλύουν στρεβλωμένα συστήματα.

4.2 Άλλες υλοποιήσεις

Το FGB/RS[2] [28] πραγματοποιεί επίλυση στους πραγματικούς χρησιμοποιώντας βάσεις Gröbner και RUR, μέσω του περιβάλλοντος που παρέχει σε MAPLE· επιπλέον ρυθμίσεις μπορούν να βελτιώσουν το χρόνο απόκρισης κατά 20-30%. Ελέγξαμε επίσης τρία πακέτα της SYNAPS[3]: ο STURM είναι μια απλοϊκή έκδοση του GRID [9]· ο SUBDIV υλοποιεί το [23], χρησιμοποιώντας βάση Bernstein και αριθμητική διπλής ακριβείας. Απαιτεί ένα αρχικό πλαίσιο και το $[-10, 10] \times [-10, 10]$ χρησιμοποιήθηκε. Ο NEWMAC [25] είναι γενικός και βασίζεται σε ιδιοδιανύσματα με χρήση του πακέτου LAPACK και υπολογίζει όλες τις μιγαδικές λύσεις.

MAPLE υλοποιήσεις: Ο INSULATE υλοποιεί το [34] για τον υπολογισμό της τοπολογίας μιας πραγματικής αλγεβρικής καμπύλης και ο TOP υλοποιεί το [12]. Οι δύο αυτές υλοποιήσεις μας παραχωρήθηκαν από τους αντίστοιχους συγγραφείς. Προσπαθήσαμε να τις τροποποιήσουμε ώστε να τερματίζουν μόλις υπολογίζουν τις πραγματικές λύσεις που αντιστοιχούν στο επαγόμενο διμετάβλητο σύστημα. Δεν ήταν όμως εύκολο να κάνουμε τέτοιες τροποποιήσεις ώστε να αντιμετωπίζουν γενικά συστήματα και για το λόγο αυτό δεν υπάρχουν χρόνοι στο πρώτο σετ πειραμάτων. Ο TOP έχει μια παράμετρο που καθορίζει την αρχική ακρίβεια (δεκαδικά ψηφία)· δεν υπάρχει εύκολος τρόπος να επιλέξει κανείς μια καλή αρχική τιμή. Έτσι, ακολουθήσαμε το [13] και καταγράψαμε την απόδοσή του για ακρίβεια 60 και 500 ψηφίων.

Συγκεντρωτικά αποτελέσματα εμφανίζονται στον πίνακα 1. Στις περιπτώσεις που κάποια υλοποίηση δεν κατάφερε να βρει σωστό πλήθος πραγματικών λύσεων σημαδεύουμε με ένα *. Σημειώστε πως ο NEWMAC χρειάζεται ένα ακόμη βήμα ώστε να διαχωρίσει κανείς τις πραγματικές λύσεις μεταξύ όλων των μιγαδικών.

Ο G_RUR είναι γρηγορότερος από το FGB/RS σε 8 από τις 18 περιπτώσεις, συμπεριλαμβάνοντας το C_5. Είναι επίσης γρηγορότερος από τον STURM σε 6 από τα 18 πειράματα. Συγκριτικά με τον SUBDIV ο G_RUR είναι γρηγορότερος στις μισές περιπτώσεις. Παρατηρήστε πως ο SUBDIV συμπεριφέρεται περίεργα στα C_1 και W_1. Συγκριτικά με τον NEWMAC, ο G_RUR τα πάει καλύτερα στα M_4, D_1 και W_3 και είναι συγκρίσιμος στα R_1 και R_3. Σε κάποιες περιπτώσεις όμως ο NEWMAC δεν υπολογίζει όλες τις πραγματικές λύσεις. Σχετικά με τις υλοποιήσεις τοπολογίας, ο G_RUR είναι ταχύτερος από τον INSULATE σε όλα τα συστήματα εκτός του W_2. Συγκριτικά με τον TOP ξεκινώντας με 60 ψηφία, ο G_RUR είναι ταχύτερος σε όλα τα συστήματα πλην του W_2. Με 500 ψηφία, ο TOP εξακολουθεί να είναι ταχύτερος στο W_2. Καθώς η διάσταση των πολυωνυμικών συστημάτων

[2] `http://www-spaces.lip6.fr/index.html`

[3] `http://www-sop.inria.fr/galaad/logiciels/synaps/`

(Σ)	βαθμός		λύσεις	Μέσος Χρόνος (msecs)									
				ΕΠΙΛΥΣΗ ΣΕ ΔΥΟ ΜΕΤΑΒΛΗΤΕΣ							ΤΟΠΟΛΟΓΙΑ		
				SLV			FGB/RS	SYNAPS			INSULATE	TOP	
	f	g		GRID	M_RUR	G_RUR		STURM	SUBDIV	NEWMAC		60	500
R_1	3	4	2	5	9	5	26	2	2	5	–	–	–
R_2	3	1	1	66	21	36	24	1	1	1	–	–	–
R_3	3	1	1	1	2	1	22	1	2	1	–	–	–
M_1	3	3	4	87	72	10	25	2	1	2	–	–	–
M_2	4	2	3	4	5	4	24	1	289*	2	–	–	–
M_3	6	3	5	803	782	110	30	230	5,058*	7	–	–	–
M_4	9	10	2	218	389	210	158	90	3*	447	–	–	–
D_1	4	5	1	6	12	6	28	2	5	8	–	–	–
D_2	2	2	4	667	147	128	26	21	1*	2	–	–	–
C_1	7	6	6	1,896	954	222	93	479	170,265*	39	524	409	1,367
C_2	4	3	6	177	234	18	27	12	23*	4	28	36	115
C_3	8	7	13	580	1,815	75	54	23	214*	25	327	693	2,829
C_4	8	7	17	5,903	80,650	370	138	3,495	217*	190*	1,589	1,624	6,435
C_5	16	15	17	$> 20'$	60,832	3,877	4,044	$> 20'$	6,345*	346*	179,182	91,993	180,917
W_1	7	6	9	2,293	2,115	247	92	954	55,040*	39	517	419	1,350
W_2	4	3	5	367	283	114	29	20	224*	3	27	20	60
W_3	8	7	13	518	2,333	24	56	32	285*	25	309	525	1,588
W_4	8	7	17	5,410	77,207	280	148	4,086	280*	207*	1,579	1,458	4,830

Πίνακας 1: Απόδοση του λογισμικού μας και άλλων πακέτων.

αυξάνει, ο G_RUR φαίνεται να είναι πιο αποτελεσματικός και από τις δύο αυτές υλοποιήσεις.

Αναφορές

[1] D. Arnon and S. McCallum. A polynomial time algorithm for the topological type of a real algebraic curve. *JSC*, 5:213–236, 1988.

[2] S. Basu, R. Pollack, and M-F. Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, 2nd edition, 2006.

[3] J. Canny. *The Complexity of Robot Motion Planning*. ACM – MIT Press Doctoral Dissertation Award Series. MIT Press, Cambridge, MA, 1987.

[4] J. Canny. Some algebraic and geometric computations in PSPACE. In *STOC*, 460–467, 1988.

[5] D.I. Diochnos, I.Z. Emiris, and E.P. Tsigaridas. On the Complexity of Real Solving Bivariate Systems. In *ISSAC*, 2007.

[6] D.I. Diochnos, I.Z. Emiris, and E.P. Tsigaridas. On the asymptotic and practical complexity of solving bivariate systems over the reals. *Journal of Symbolic Computation*, 2008.

[7] A. Eigenwillig, V. Sharma, and C.K. Yap. Almost tight recursion tree bounds for the Descartes method. In *ISSAC '06: Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation*, 71–78, New York, NY, USA, 2006. ACM Press.

[8] I.Z. Emiris, B. Mourrain, and E.P. Tsigaridas. Real Algebraic Numbers: Complexity Analysis and Experimentation. In Hertling, P. and Hoffmann,

C. and Luther, W. and Revol, N., editors, *Reliable Implementations of Real Number Algorithms: Theory and Practice*, volume 5045 of *LNCS*, 57–82. Springer Verlag, 2008. also available in www.inria.fr/rrrt/rr-5897.html.

[9] I.Z. Emiris and E.P. Tsigaridas. Real solving of bivariate polynomial systems. In V. Ganzha and E. Mayr, editor, *Proc. Computer Algebra in Scientific Computing (CASC)*, vol. 3718 of *LNCS*, 150–161. Springer, 2005.

[10] L. González-Vega and M. El Kahoui. An Improved Upper Complexity Bound for the Topology Computation of a Real Algebraic Plane Curve. *J. Complexity*, 12(4):527–544, 1996.

[11] L. González-Vega, H. Lombardi, T. Recio, and M-F. Roy. Sturm-Habicht Sequence. In *ISSAC*, 136–146, 1989.

[12] L. González-Vega and I. Necula. Efficient topology determination of implicitly defined algebraic plane curves. *Computer Aided Geometric Design*, 19(9):719–743, Dec 2002.

[13] M. Kerber. Analysis of Real Algebraic Plane Curves. Diploma thesis, MPI Saarbrücken, 2006.

[14] J. Klose. Binary Segmentation for Multivariate Polynomials. *J. Complexity*, 11(3):330–343, 1995.

[15] K.H. Ko, N.M. Patrikalakis, and T. Sakkalis. Resolution of multiple roots of nonlinear polynomial systems. *International J. of Shape Modeling*, 11(1):121–147, 2005.

[16] D. Lakshman, Y.N. and Lazard. On the complexity of zero-dimensional algebraic systems. In T.M̃ora and C. Traverso, editors, *Effective Methods in Algebraic Geometry*, volume 94 of *Progress in Mathematics*, 217–225, Boston, 1991. Birkhäuser. (Proc. MEGA '90, Livorno, Italy).

[17] T. Lickteig and M-F. Roy. Sylvester-habicht sequences and fast Cauchy index computation. *J. Symb. Comput.*, 31(3):315–341, 2001.

[18] H. Lombardi, M-F. Roy, and M. Safey El Din. New Structure Theorem for Subresultants. *J. Symb. Comput.*, 29(4-5):663–689, 2000.

[19] P.S. Milne. On the solution of a set of polynomial equations. In B. Donald, D. Kapur, and J. Mundy, editors, *Symbolic and Numerical Computation for Artificial Intelligence*, 89–102. Academic Press, 1992.

[20] B. Mourrain. A new criterion for normal form algorithms. *Proc. AAECC*, vol. 1719 of *LNCS*, 430–443, 1999.

[21] B. Mourrain and V.Y. Pan. Solving special polynomial systems by using structured matrices and algebraic residues. In F. Cucker and M. Shub, editors, *Proc. Workshop on Foundations of Computational Mathematics*, 287–304, Berlin, 1997. Springer-Verlag.

[22] B. Mourrain and V.Y. Pan. Asymptotic acceleration of solving polynomial systems. In *STOC*, 488–496. ACM Press, New York, 1998.

[23] B. Mourrain and J-P. Pavone. Subdivision methods for solving polynomial equations. Technical Report RR-5658, INRIA Sophia-Antipolis, 2005.

[24] B. Mourrain, S. Pion, S. Schmitt, J-P. Técourt, E. Tsigaridas, and N. Wolpert. Algebraic issues in computational geometry. In Jean-Daniel Boissonnat and Monique Teillaud, editors, *Effective Computational Geometry for Curves and Surfaces*, 117–155. Springer-Verlag, Mathematics and Visualization, 2006.

[25] B. Mourrain and Ph. Trébuchet. Solving projective complete intersection faster. In *ISSAC*, 231–238. ACM Press, New York, 2000.

[26] P. Pedersen, M-F. Roy, and A. Szpirglas. Counting real zeros in the multivariate case. In F.Ẽyssette and A. Galligo, editors, *Computational Algebraic Geometry*, vol. 109 of *Progress in Mathematics*, 203–224. Birkhäuser, Boston, 1993.

[27] D. Reischert. Asymptotically Fast Computation of Subresultants. In *ISSAC*, 233–240, 1997.

[28] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Journal of Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, 1999.

[29] T. Sakkalis. Signs of algebraic numbers. *Computers and Mathematics*, 131–134, 1989.

[30] T. Sakkalis and R. Farouki. Singular Points of Algebraic Curves. *J. Symb. Comput.*, 9(4):405–421, 1990.

[31] Elias P. Tsigaridas. *Algebraic Algorithms and Applications to Geometry*. PhD thesis, Dept. of Informatics and Telecommunications, University of Athens, 2006.

[32] M. van Hoeij and M. Monagan. A modular GCD algorithm over number fields presented with multiple extensions. In *ISSAC*, 109–116, July 2002.

[33] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge Univ. Press, Cambridge, U.K., 2nd edition, 2003.

[34] N. Wolpert and R. Seidel. On the Exact Computation of the Topology of Real Algebraic Curves. In *SoCG*. ACM, 2005.

[35] C.K. Yap. *Fundamental Problems of Algorithmic Algebra*. Oxford University Press, New York, 2000.