



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**«Κατηγορήματα για τον υπολογισμό του
Απολλώνιου Διαγράμματος ελλείψεων»**

ΓΕΩΡΓΙΟΣ ΤΖΟΥΜΑΣ

ΑΜ: Μ491

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ

ΙΩΑΝΝΗΣ ΕΜΙΡΗΣ

ΑΘΗΝΑ

ΜΑΪΟΣ 2005

*Ευχαριστώ τον καθηγητή μου Ιωάννη Εμίρη, καθώς
και τους συνεργάτες του Μενέλαο Καραβέλα και
Ηλία Τσιγαρίδα για την υπέροχη συνεργασία.*

Περίληψη

Στη διπλωματική αυτή μας ενδιαφέρει ο υπολογισμός του Απολλώνιου διαγράμματος ενός συνόλου ελλείψεων οι οποίες δεν τέμνονται μεταξύ τους, ούτε περιέχεται μία μέσα σε κάποια άλλη. Πρόκειται για μία γενίκευση του διαγράμματος Voronoi. Το επίπεδο χωρίζεται σε περιοχές τα σημεία των οποίων βρίσκονται πλησιέστερα σε μία έλλειψη. Ο υπολογισμός του διαγράμματος γίνεται χρησιμοποιώντας έναν αυξητικό αλγόριθμο, ο οποίος όμως για τη λειτουργία του χρειάζεται τα εξής κατηγορήματα: (κ_0) τον υπολογισμό ενός ρητού σημείου εσωτερικό της έλλειψης· (κ_1) τον προσδιορισμό της έλλειψης που βρίσκεται πιο κοντά σε ένα δεδομένο σημείο από οποιαδήποτε άλλη· (κ_2) δοθέντων τριών ελλείψεων, τον προσδιορισμό της σχετικής θέσης της τρίτης ως προς μια προκαθορισμένη εξωτερική κοινή εφαπτομένη των δύο πρώτων· (κ_3) δοθέντων τεσσάρων ελλείψεων, τον προσδιορισμό της σχετικής θέσης της τέταρτης ως προς τον εξωτερικό Απολλώνιο κύκλο των τριών πρώτων. Για να ολοκληρωθεί ο αυξητικός αλγόριθμος είναι απαραίτητος και ο υπολογισμός της «περιοχής σύγκρουσης» μιας νέας έλλειψης που προστίθεται στο τρέχον διάγραμμα.

Μας ενδιαφέρει ο υπολογισμός των παραπάνω κατηγορημάτων αποτελεσματικά και με πλήρη ακρίβεια. Μία πλήρης λύση για το (κ_0) προκύπτει άμεσα. Προτείνουμε μία αποτελεσματική αναπαράσταση της απόστασης σημείου από έλλειψη θεωρώντας έναν παραμετρικό κύκλο που εφάπτεται στην έλλειψη. Η απόσταση εκφράζεται από την ύπαρξη πολλαπλής ρίζας του χαρακτηριστικού πολυωνύμου των δύο σχημάτων. Ο μηδενισμός της διακρίνουσας δίνει έναν βέλτιστο αλγόριθμο και μία αποτελεσματική υλοποίηση (χρησιμοποιώντας τη βιβλιοθήκη SYNAPS) για το (κ_1). Μια παρόμοια προσέγγιση με διακρίνουσες χρησιμοποιείται και για το (κ_2) το οποίο εμφανίζεται και σε προβλήματα ορατότητας. Ο τρόπος που αντιμετωπίζουμε τις ειδικές περιπτώσεις όπου υπάρχουν κάθετες εφαπτομένες, οδηγεί στον αποτελεσματικό προσδιορισμό όλων των εφαπτομένων. Όσον αφορά το (κ_3), μελετάμε την πολυπλοκότητά του με τη θεωρία της αραιής (ή τορικής) απαλοιφής και δίνουμε το πλήθος των μιγαδικών κύκλων που εφάπτονται από κοινού. Τέλος, για το τελευταίο κατηγορήμα παρέχουμε ορισμένες ημιαλγεβρικές συνθήκες και γεωμετρικούς περιορισμούς που μπορούν να οδηγήσουν σε μια αποτελεσματική υλοποίηση πραγματικού χρόνου.

Περιεχόμενα

0 Κατηγορήματα για τον υπολογισμό του Απολλώνιου διαγράμματος ελλείψεων	6
0.1 Εισαγωγή	6
0.1.1 Η γεωμετρία της έλλειψης	9
0.2 Απόσταση σημείου από έλλειψη	11
0.2.1 Σχετική θέση έλλειψης ως προς κοινή εφαπτομένη ελλείψεων	13
0.3 Εξωτερικός εφαπτόμενος Απολλώνιος κύκλος	14
0.4 Υλοποίηση	17
0.5 Επίλογος	18
Μέρος Β΄	19
1 Preliminaries	20
1.1 The Ellipse	24
2 Distance between point and ellipse	29
2.1 Parametric approach	30
2.2 Algebraic approach	33
2.2.1 The resultant	33
2.2.2 The discriminant	36
2.3 Bitangent and query ellipse	40
3 External tritangent Voronoi circle	44
3.1 The discriminant system	44
3.2 The bisector of two ellipses	47
3.3 The cartesian system	47
3.4 Semi-algebraic conditions	48
3.5 Geometric constraints	49
3.6 The optimal discriminant system	51
3.7 The upper bound	51

3.8 Solving the system	52
4 Implementation	53
4.1 Source code	54
5 Conclusion	65
Βιβλιογραφία	66

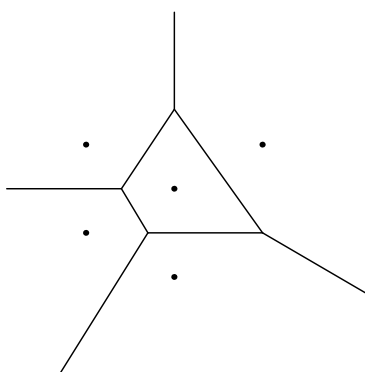
Κεφάλαιο 0

Κατηγορήματα για τον υπολογισμό του Απολλώνιου διαγράμματος ελλείψεων

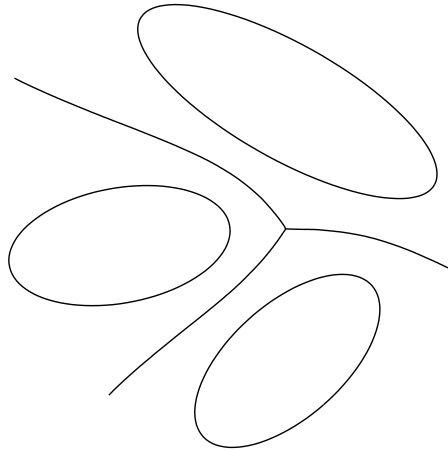
0.1 Εισαγωγή

Τα διαγράμματα Voronoi είναι γεωμετρικές κατασκευές που παρουσιάζουν ιδιαίτερο ενδιαφέρον. Βρίσκουν εφαρμογές στο σχεδιασμό κίνησης ρομπότ, την ανίχνευση συγκρούσεων καθώς και την αναγνώριση προτύπων. Έστω ένα σύνολο σημειακών εστιών p_i στο επίπεδο. Ως περιοχή (ή κελί) Voronoi μίας εστίας ορίζουμε τα σημεία του επιπέδου που βρίσκονται κοντινότερα στην εστία αυτή παρά σε οποιαδήποτε άλλη. Το σύνορο των περιοχών Voronoi των εστιών αποτελεί το διάγραμμα Voronoi των εστιών (βλ. σχήμα 1).

Αν και τα διαγράμματα Voronoi έχουν μελετηθεί εκτενώς [7], η πλειοψηφία των εργασιών



Σχήμα 1: Διάγραμμα Voronoi σημείων



Σχήμα 2: Απολλώνιο διάγραμμα τριών ελλείψεων

αφορούν εστίες-σημεία είτε γραμμικές εστίες, όπως ευθύγραμμα τμήματα [22] και πολύγωνα [32]. Πρόσφατα έγιναν κάποιες προσπάθειες να επεκταθούν τα διαγράμματα Voronoi στην περίπτωση που οι εστίες είναι καμπύλες (π.χ. [1, 3, 6]) είτε έχουν μη κενό εσωτερικό [10, 13, 9]. Ως πρόδρομος της εργασίας αυτής μπορεί να θεωρηθεί το διάγραμμα Voronoi κύκλων π.χ. [4, 24, 27]. Όταν οι εστίες είναι κλειστά σχήματα, το διάγραμμα Voronoi ονομάζεται και Απολλώνιο¹ διάγραμμα.

Ο σκοπός μας είναι να υπολογίσουμε το Απολλώνιο διάγραμμα ελλείψεων (βλ. σχήμα 2) αποτελεσματικά και με πλήρη ακρίβεια. Με τον όρο «πλήρη ακρίβεια» εννοούμε ότι ο αλγόριθμος θα αποφασίζει πάντα σωστά χωρίς περιθώριο σφάλματος. Συμμετρικές περιπτώσεις στην είσοδο δίνουν συμμετρικά αποτελέσματα, κάτι που δε συμβαίνει πάντα με τις αριθμητικές προσεγγίσεις. Οι αλγόριθμοι πλήρους ακρίβειας εξαρτώνται μόνο από το μέγεθος της εισόδου (σε bit) και είναι πιο αργοί από τους αριθμητικούς.

Εστιάζουμε το ενδιαφέρον μας στην περίπτωση που οι ελλείψεις δεν τέμνονται μεταξύ τους. Η προσέγγισή μας βασίζεται στο γενικό πλαίσιο των αφηρημένων διαγραμμάτων Voronoi, π.χ. [28]. Πιο συγκεκριμένα ακολουθούμε την προσέγγιση του [25], όπου προτείνεται ένας δυναμικός αλγόριθμος για την κατασκευή του διαγράμματος Voronoi κυρτών αντικειμένων των οποίων οι περιφέρειες τέμνονται (ανά δύο) σε δύο σημεία το πολύ. Ένα σύνολο μη τεμνόμενων ελλείψεων μπορεί να ενταχθεί σε αυτό το πλαίσιο. Η προσέγγιση αποτελείται από δύο φάσεις. Πρώτα εντοπίζουμε ένα κελί Voronoi που μεταβάλλεται λόγω της έλλειψης που εισάγεται. Έπειτα βρίσκουμε κι ενημερώνουμε όλα τα κελιά αυτής της κατηγορίας. Ο αλγόριθμος στηρίζεται στα παρακάτω κατηγορήματα:

(κ_0) δοθείσας μιας έλλειψης, να βρεθεί εσωτερικό σημείο αυτής

¹Για την προέλευση του ονόματος βλ. παρακάτω.

- (κ_1) δοθέντων δύο ελλείψεων κι ενός σημείου εξωτερικό και των δύο, να προσδιοριστεί ποια έλλειψη βρίσκεται πλησιέστερα στο σημείο υπό την Ευκλείδεια μετρική
- (κ_2) δοθέντων τριών ελλείψεων, να προσδιοριστεί η σχετική θέση της τρίτης ως προς την κοινή εξωτερική εφαπτομένη των δύο πρώτων
- (κ_3) δοθέντων τεσσάρων ελλείψεων, να αποφασιστεί η σχετική θέση της τέταρτης ως προς τον εξωτερικό Απολλώνιο κύκλο των τριών πρώτων.

Το (κ_0) χρησιμοποιείται σε συνδυασμό με όλα τα υπόλοιπα κατηγορήματα και είναι απλό στην υλοποίηση. Το κατηγορήμα (κ_1) αρκεί για την πρώτη φάση του αλγορίθμου, όπου αρκεί να βρούμε την έλλειψη του διαγράμματος που βρίσκεται πλησιέστερα σε οποιοδήποτε σημείο της νέας έλλειψης· το σημείο αυτό παρέχεται από το (κ_0). Τα κατηγορήματα (κ_2) και (κ_3) χρησιμοποιούνται στη δεύτερη φάση. Η ενημέρωση των κελιών που μεταβάλλονται μπορεί να απαιτήσει κατασκευές τις οποίες δεν πραγματευόμαστε στην εργασία αυτή.

Για το κατηγορήμα (κ_1) θεωρούμε έναν κύκλο με κέντρο το σημείο και άγνωστη ακτίνα η οποία αντιστοιχεί στην απόσταση που θέλουμε να συγκρίνουμε. Η ύπαρξη σημείου επαφής ανάμεσα στον κύκλο και την έλλειψη ισοδυναμεί με το μηδενισμό της διακρίνουσας του χαρακτηριστικού πολυωνύμου των δύο σχημάτων. Το χαρακτηριστικό πολυώνυμο ισούται με την ορίζουσα της δέσμης που ορίζουν ο κύκλος και η έλλειψη. Η συνεισφορά μας για τον υπολογισμό αυτού του κατηγορήματος είναι μια μέθοδος η οποία χρησιμοποιεί αλγεβρικούς αριθμούς βαθμού 4, το οποίο είναι βέλτιστο. Αξίζει να σημειώσουμε ότι αποφεύγουμε να εκφράσουμε τις συντεταγμένες του σημείου επαφής.

Για το κατηγορήμα (κ_2) θεωρούμε το σύστημα ευθείας και έλλειψης του οποίου ο μηδενισμός της διακρίνουσας δίνει πάλι μια αποτελεσματική συνθήκη για το πότε μια ευθεία εφάπτεται σε μια έλλειψη. Όλες οι εφαπτομένες χαρακτηρίζονται από ένα σύστημα διακρινουσών. Οι ειδικές περιπτώσεις ύπαρξης κοινής κατακόρυφης εφαπτομένης εξετάζονται ξεχωριστά. Η μελέτη τους δίνει παράλληλα κι έναν αποτελεσματικό τρόπο για τον χαρακτηρισμό όλων των κοινών εφαπτομένων ως εσωτερικών ή εξωτερικών. Και τα δύο παραπάνω κατηγορήματα έχουν υλοποιηθεί σε C++ με πλήρη ακρίβεια χρησιμοποιώντας τη βιβλιοθήκη SYNAPS.

Ας επικεντρωθούμε τώρα στο κατηγορήμα (κ_3). Στην περίπτωση τριών δίσκων, το πλήθος των κύκλων που εφάπτονται και στους τρεις είναι 8 και το αντίστοιχο κατηγορήμα έχει αλγεβρικό βαθμό 2 [15]. Το πρόβλημα αυτό είναι γνωστό ως «Το δέκατο πρόβλημα του Απολλώνιου», διότι πρώτα μελετήθηκε από τον *Άπολλώνιο έκ Πέργης της Παμφυλίας*, περίπου το 250 π.Χ. Παρότι το πρόβλημα είναι γνωστό από την αρχαιότητα, η γενίκευση για ελλείψεις δεν έχει λυθεί ακόμη αποτελεσματικά.

Ούτε καν το πλήθος των κύκλων που εφάπτονται ταυτόχρονα σε τρεις ελλείψεις δεν είναι γνωστό. Το πρόβλημα περιλαμβάνει εξισώσεις υψηλού βαθμού και η εύρεση ακριβούς λύσης δεν είναι προφανής ([3]). Στο [31] γίνεται μια προσπάθεια να αντιμετωπισθεί το πρόβλημα αυτό, ωστόσο ο ακριβής υπολογισμός με την προτεινόμενη μέθοδο δεν ολοκληρώνεται

και ο συγγραφέας αρκείται σε αριθμητικές μεθόδους. Δεν αναφέρονται φράγματα για την πολυπλοκότητα ούτε για το πλήθος των κύκλων.

Εφαρμόζουμε τη μέθοδο από το κατηγορήμα (κ_1) μαζί με πρόσφατα αποτελέσματα από τις αραιές (ή τις τορικές) απαλοίφουσες για να προβάλλουμε όλες τις κοινές ρίζες στις ρίζες ενός πολυωνύμου μιας μεταβλητής. Αυτό οδηγεί στο πρώτο ενδιαφέρον άνω φράγμα για το πλήθος των εφαπτόμενων κύκλων. Ο μεικτός όγκος δίνει επίσης το ίδιο φράγμα όπως και ένας συλλογισμός πραγματικής αλγεβρικής γεωμετρίας.

Η πρότασή μας είναι ότι ο γεωμετρικός λογισμός και τα ημιαλγεβρικά εργαλεία θα πρέπει να ενισχύσουν τις αμιγώς αλγεβρικές μεθόδους για μια αποτελεσματική υλοποίηση πλήρους ακρίβειας. Σε αυτή την κατεύθυνση δίνουμε νέες συνθήκες βασιζόμενοι στο κανόνα του Descartes, και αναλύουμε τη γεωμετρική σημασία τους. Επιπρόσθετα, δίνουμε μερικούς γεωμετρικούς περιορισμούς για το κέντρο και την ακτίνα του Απολλώνιου κύκλου. Στόχος μας είναι να διακρίνουμε περιπτώσεις των οποίων η αλγεβρική πολυπλοκότητα είναι μικρότερη του γενικού προβλήματος, αναμένοντας αυτό μαζί με τεχνικές υποδιαίρεσης να οδηγήσει σε μία υλοποίηση πραγματικού χρόνου. Προς το παρόν λύνουμε το αντίστοιχο αλγεβρικό σύστημα αριθμητικά χρησιμοποιώντας το PHSpack [34].

Η διατριβή οργανώνεται ως εξής. Πρώτα εισάγουμε μερικές ιδιότητες της έλλειψης και υπολογίζουμε το (κ_0). Στη συνέχεια παρουσιάζουμε μια αποτελεσματική αναπαράσταση της απόστασης σημείου από έλλειψη για να λύσουμε το (κ_1). Έπειτα, εξετάζουμε το (κ_2) και τέλος μελετάμε τον κύκλο του Απολλώνιου. Κλείνουμε παραθέτοντας τους σχετικούς χρόνους από την υλοποίηση των (κ_1) και (κ_2) καθώς και την αριθμητική επίλυση του (κ_3). Το κεφάλαιο αυτό αποτελεί μια ελληνική περίληψη των πιο βασικών αποτελεσμάτων της διατριβής, με τις επόμενες ενότητες να αντιστοιχούν στα κεφάλαια που ακολουθούν στα αγγλικά. Μερικά από τα αποτελέσματα αυτά έχουν δημοσιευτεί στο [17].

0.1.1 Η γεωμετρία της έλλειψης

Έλλειψη καλείται ο γεωμετρικός τόπος των σημείων του επιπέδου, των οποίων το άθροισμα των αποστάσεων από δύο σταθερά σημεία είναι σταθερό και ίσο με $2a$. Τα δύο σταθερά σημεία καλούνται εστίες της έλλειψης και βρίσκονται σε απόσταση 2γ . Το μήκος του μεγάλου και μικρού άξονα είναι $2a$ και $2b$ αντίστοιχα, με $\beta^2 = a^2 - \gamma^2$. Έστω (x_c, y_c) το κέντρο της έλλειψης και u η γωνία μεταξύ του άξονα x και του μεγάλου άξονα της έλλειψης. Αν $x_c = y_c = u = 0$, λέμε ότι η έλλειψη βρίσκεται σε ορθογώνια θέση, διαφορετικά λέμε ότι βρίσκεται σε γενική θέση. Η έλλειψη είναι μια κωνική τομή με εξίσωση:

$$E(x, y) := ax^2 + 2bxy + cy^2 + 2dx + 2ey + f \quad (1)$$

Οι παράμετροι της έλλειψης (a, b, c, d, e, f) συνδέονται με το κέντρο, τους άξονες και την εστιακή απόσταση. Για τις σχετικές εξισώσεις παραπέμπουμε στην ενότητα 1.1. Η εξίσωση

της έλλειψης μπορεί να γραφεί και με την παρακάτω παραμετρική μορφή:

$$\begin{aligned}x(z) &= \frac{-\alpha(1-w^2)z^2 - 4\beta wz + \alpha(1-w^2)}{(1+w^2)(1+z^2)} + x_c \\y(z) &= \frac{2(-\alpha wz^2 + \beta(1-w^2)z + \alpha w)}{(1+w^2)(1+z^2)} + y_c\end{aligned}$$

όπου το $w \in (-\infty, \infty)$ αντιστοιχεί στην γωνία στροφής των αξόνων², (x_c, y_c) είναι οι συντεταγμένες του κέντρου και $z \in (-\infty, \infty)$. Οι παρακάτω ποσότητες αποτελούν αναλλοίωτες υπό περιστροφή και μετατόπιση:

$$J_1 = a + c = \alpha^2 + \beta^2, \quad J_2 = \begin{vmatrix} a & b \\ b & c \end{vmatrix} = \alpha^2\beta^2$$

$$J_3 = \begin{vmatrix} a & b & d \\ b & c & e \\ d & e & f \end{vmatrix} = -J_2^2$$

Η παρακάτω ποσότητα αποτελεί αναλλοίωτη υπό περιστροφή. Η έκφραση χρησιμοποιεί το λήμμα που ακολουθεί.

$$J_4 = (a+c)f - d^2 - e^2 = J_2(x_c^2 + y_c^2 - J_1)$$

Έστω L_y, L_x οι ευθείες που ενώνουν το αριστερότερο με το δεξιότερο σημείο της έλλειψης, και το ψηλότερο με το χαμηλότερο αντίστοιχα.

Λήμμα 0.1.1 Έστω μία έλλειψη της μορφής (1). Το κέντρο της είναι ρητή συνάρτηση των παραμέτρων της και συμπίπτει με το σημείο τομής των L_x, L_y , με

$$x_c = \frac{be - dc}{J_2}, \quad y_c = \frac{bd - ae}{J_2}$$

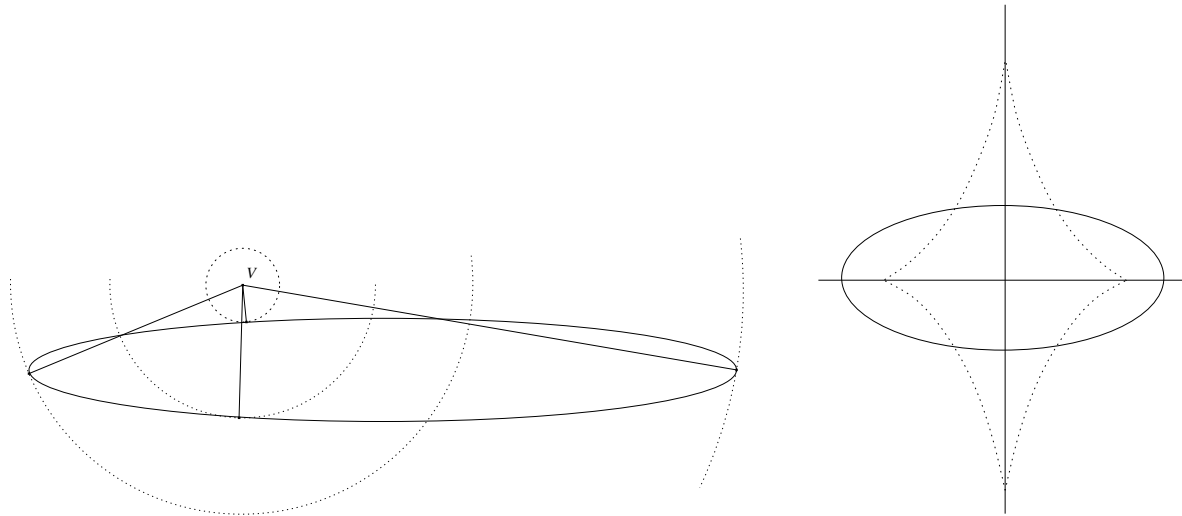
Η απόδειξη δίνεται στην ενότητα 1.1.

Δοθέντος ενός σημείου V εξωτερικό μιας έλλειψης, πόσες καθέτους³ μπορούμε να φέρουμε στην έλλειψη; Το σύνολο των περιοχών όπου το πλήθος των καθέτων που μπορούμε να φέρουμε αλλάζει, είναι η εξειλιγμένη της έλλειψης η οποία είναι ένα αστροειδές (βλ. σχήμα 3). Για μια έλλειψη σε ορθογώνια θέση, κάθε σημείο (x, y) της εξειλιγμένης ικανοποιεί την εξίσωση:

$$(\alpha x)^{\frac{2}{3}} + (\beta y)^{\frac{2}{3}} = \gamma^{\frac{4}{3}}$$

² $w = \tan(\frac{u}{2})$, όπου u η γωνία στροφής των αξόνων της έλλειψης.

³Κάθετος σε καμπύλη είναι η κάθετος στην εφαπτομένη της καμπύλης στο σημείο επαφής.



Σχήμα 3: Αριστερά: Ένα σημείο από το οποίο φέρουμε 4 καθέτους. Δεξιά: Η εξειλιγμένη της έλλειψης.

Πρόταση 0.1.2 Υπάρχουν 4, 3, ή 2 κάθετοι από ένα σημείο σε μια έλλειψη, ανάλογα με το αν το σημείο βρίσκεται μέσα στην εξειλιγμένη, πάνω στην εξειλιγμένη αλλά όχι σε άκρο, ή αντίστοιχα το σημείο είναι άκρο ή βρίσκεται εκτός της εξειλιγμένης.

Το παραπάνω λήμμα δίνει ένα κάτω φράγμα για την αλγεβρική πολυπλοκότητα του υπολογισμού της απόστασης σημείου από έλλειψη, αφού οποιαδήποτε συνθήκη πάνω στην άγνωστη απόσταση θα έχει βαθμό ≥ 4 .

0.2 Απόσταση σημείου από έλλειψη

Έστω μια έλλειψη E κι ένα σημείο $V = (v_1, v_2)$ εξωτερικό της E . Έστω C κύκλος με κέντρο V και ακτίνα \sqrt{s} , για s θετικό. Εκφράζουμε την ευκλείδια απόσταση $\delta(V, E)$ του V από την E με τη μικρότερη θετική τιμή του \sqrt{s} για την οποία ο C εφάπτεται στην E . Για τη σύγκριση αποστάσεων, αρκεί να θεωρούμε το τετράγωνο της απόστασης s .

Ο στόχος μας είναι να αποφύγουμε να υπολογίσουμε τα σημεία επαφής απαιτώντας το σύστημα $E = C = 0$ να έχει πολλαπλή ρίζα. Καταλήγουμε σε ένα απλό πολυώνυμο εφαρμόζοντας τη θεωρία των χαρακτηριστικών πολυωνύμων και των δέσμεων [30, 36]. Ας συμβολίσουμε την κωνική τομή ως $[x, y, 1]M[x, y, 1]^T$ για έναν κατάλληλο πίνακα M . Τότε οι E, C αντιστοιχούν στα

$$A = \begin{bmatrix} a & b & d \\ b & c & e \\ d & e & f \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & -v_1 \\ 0 & 1 & -v_2 \\ -v_1 & -v_2 & v_1^2 + v_2^2 - s \end{bmatrix}$$

Η δέσμη της E και του C είναι $\lambda A + B$ και το χαρακτηριστικό τους πολυώνυμο είναι

$$\phi(\lambda) = |\lambda A + B| = J_2^2 \lambda^3 + c_2 \lambda^2 + c_1 \lambda + s$$

όπου

$$\begin{aligned} c_2(s) &= J_2 s - T(v_1, v_2) \\ c_1(s) &= J_1 s - E(v_1, v_2) \\ T(v_1, v_2) &= J_2[(v_1 - x_c)^2 + (v_2 - y_c)^2 - J_1] \end{aligned}$$

Η διακρίνουσα $\Delta(s)$ του $\phi(\lambda)$ είναι $4^{\text{ου}}$ βαθμού:

$$\begin{aligned} \Delta(s) &= J_2^2(J_1^2 - 4J_2) s^4 + \\ &2J_2(9J_1J_2^2 - J_1^2T + 6J_2T - 2J_1^3J_2 - J_1J_2E) s^3 + \\ &(-18J_2^3E + 4J_1J_2ET - 27J_2^4 + J_1^2T^2 - 18J_1J_2^2T + J_2^2E^2 + 12J_1^2J_2^2E - 12J_2T^2) s^2 + \\ &2(2T^3 - J_1ET^2 - 6J_1J_2^2E^2 + 9J_2^2ET - J_2E^2T) s + \\ &E^2(T^2 + 4J_2^2E) \end{aligned}$$

Η σχετική θέση του κύκλου και της έλλειψης εντάσσεται ανάμεσα σε 9 περιπτώσεις (κεφ. 2), οι οποίες σχετίζονται με το πρόσημο και την πολλαπλότητα των πραγματικών ριζών του $\phi(\lambda)$ [36, θεώρ.8]. Όταν το $\phi(\lambda)$ έχει τουλάχιστον μία πολλαπλή ρίζα, η έλλειψη και κύκλος έχουν τουλάχιστον ένα σημείο επαφής. Σημειώστε ότι το $\phi(\lambda)$ έχει τουλάχιστον μία αρνητική ρίζα, αφού το γινόμενο των ριζών του ισούται με $-s < 0$.

Επιλέγοντας τη μικρότερη θετική ρίζα του $\Delta(s)$, είμαστε βέβαιοι ότι το $\phi(\lambda)$ θα έχει τουλάχιστον μία ρίζα με πολλαπλότητα μεγαλύτερη του 1, αφού Δ είναι η διακρίνουσα του ϕ . Έστω ότι ο κύκλος με κέντρο το V «φρουσώνει» μέχρι να «ακουμπήσει» την E . Αφού το V είναι εξωτερικό της E , η μικρότερη θετική ρίζα του $\Delta(s)$ αντιστοιχεί στην απόσταση $\delta(V, E)$.

Πρόταση 0.2.1 Δοθέντων μιας έλλειψης E και ενός σημείου V εξωτερικό αυτής, η $\delta(V, E)$ αντιστοιχεί στην τετραγωνική ρίζα της μικρότερης θετικής ρίζας του $\Delta(s)$, το οποίο είναι πολυώνυμο μιας μεταβλητής $4^{\text{ου}}$ βαθμού. Ο βαθμός των συντελεστών του $\Delta(s)$ είναι 6, 8, 10, 12 και 14 ως προς το αυξανόμενο εκθέτη s , τα v_1, v_2 και τις παραμέτρους της E . Στην περίπτωση που το (v_1, v_2) είναι κέντρο μιας τρίτης έλλειψης E' , ο βαθμός των συντελεστών του $\Delta(s)$ είναι ακριβώς 22 ως προς τις παραμέτρους των E, E' .

Πόρισμα 0.2.2 Δοθέντων δύο ελλείψεων E_1, E_2 κι ενός σημείου V εξωτερικό και των δύο, μπορούμε να αποφασίσουμε ποια έλλειψη βρίσκεται πλησιέστερα στο V συγκρίνοντας δύο αλγεβρικούς αριθμούς βαθμού 4. Ο βαθμός αυτός είναι βέλτιστος, σύμφωνα με όσα αναφέρθηκαν στην προηγούμενη ενότητα.

0.2.1 Σχετική θέση έλλειψης ως προς κοινή εφαπτομένη ελλείψεων

Δοθέντων δύο ελλείψεων και μιας εξωτερικής κοινής εφαπτομένης τους, το κατηγορήμα (κ_2) αποφασίζει απο ποια πλευρά της εφαπτομένης βρίσκεται μια τρίτη έλλειψη (σχήμα 2.5). Σε περίπτωση που η τρίτη έλλειψη τέμνει την εφαπτομένη, υπολογίζει και το πλήθος των σημείων τομής. Όπως και παραπάνω, μελετάμε ελλείψεις που δεν τέμνονται. Για τον υπολογισμό του κατηγορήματος υπολογίζουμε και χαρακτηρίζουμε ως εσωτερικές ή εξωτερικές όλες τις κοινές εφαπτομένες των δύο ελλείψεων. Στη συνέχεια επιλέγουμε την εφαπτομένη που μας ενδιαφέρει και τέλος βρίσκουμε τη σχετική θέση της τρίτης έλλειψης ως προς αυτή.

Έστω μία μη κατακόρυφη ευθεία $L : y = ux + v$ και μία έλλειψη E . Αντικαθιστώντας το y στην E παίρνουμε ένα τετραγωνικό πολυώνυμο ως προς x το οποίο είναι η απαλοίφουσα ως προς y των E, L . Για να εφάπτεται η L στην E , η διακρίνουσα Λ της απαλοίφουσας θα πρέπει να μηδενίζεται:

$$\Lambda(u, v) = (e^2 - cf)u^2 + 2(cd - be)uv + (b^2 - ac)v^2 + 2(de - bf)u + 2(bd - ae)v - af + d^2$$

Για την κοινή εφαπτομένη ισχύει $\Lambda_1 = \Lambda_2 = 0$, όπου το Λ_i αντιστοιχεί στην i -οστή έλλειψη. Το σύστημα έχει το πολύ 4 πραγματικές ρίζες που αντιστοιχούν στις 4 κοινές εφαπτομένες δύο ελλείψεων.

Λήμμα 0.2.3 Έστω u_1, u_2, u_3, u_4 οι συντελεστές διεύθυνσης των τεσσάρων κοινών εφαπτομένων σε αύξουσα σειρά, όπου $u_i \in \mathbb{R} \cup \{\infty\}$. Έστω μ μια εσωτερική εφαπτομένη και ϵ μια εξωτερική. Τότε τα (u_1, u_2, u_3, u_4) αντιστοιχούν σε μια κυκλική μετάθεση των $(\mu\epsilon\epsilon\mu)$. Ονομάζουμε αυτή τη μετάθεση διάταξη-II.

Η απόδειξη δίνεται στην ενότητα 2.3.

Πόρισμα 0.2.4 Για να προσδιορίσουμε τη διάταξη-II των κοινών εφαπτομένων δύο ελλείψεων, αρκεί να προσδιορίσουμε τον τύπο ακριβώς δύο εφαπτομένων.

Η προηγούμενη ανάλυση δε λαμβάνει υπόψιν τις κατακόρυφες εφαπτομένες οι οποίες αντιστοιχούν σε ρίζες στο άπειρο του συστήματος $\Lambda_1 = \Lambda_2 = 0$. Σε περίπτωση κατακόρυφη εφαπτομένης, ο βαθμός της απαλοίφουσας γίνεται μικρότερος του 4. Μια τέτοια εφαπτομένη έχει εξίσωση $x = v$. Αν αντικαταστήσουμε την τιμή του x στην εξίσωση της έλλειψης, παίρνουμε ένα τριώνυμο ως προς y η διακρίνουσα του οποίου είναι $K(v) = (b^2 - ac)v^2 + 2(eb - dc)v + e^2 - fc$. Έτσι, δύο ελλείψεις έχουν κοινή κατακόρυφη εφαπτομένη αν και μόνο αν $K_1(v)$ και $K_2(v)$ έχουν κοινή ρίζα.

Λήμμα 0.2.5 Η διάταξη των ριζών του $K_1(v)$ και του $K_2(v)$, μαζί με την τεταγμένη των σημείων επαφής των κατακόρυφων εφαπτομένων, δίνει τη διάταξη-II όλων των εφαπτομένων, κατακόρυφων και μη.

Για την απόδειξη βλ. ενότητα 2.3.

Δοθείσας μιας εξωτερικής εφαπτομένης $\bar{L} : y = \bar{u}x + \bar{v}$, μπορούμε να προσδιορίσουμε τη σχετική θέση μιας τρίτης έλλειψης E_3 ως προς αυτήν υπολογίζοντας τη διακρίνουσα των \bar{L} και E_3 , $\Lambda(\bar{u}, \bar{v})$. Η Λ_3 είναι αρνητική, μηδέν ή θετική αν και μόνο αν η E_3 έχει 0, 1 ή 2 κοινά σημεία με την \bar{L} αντίστοιχα. Στις πρώτες 2 περιπτώσεις, το πρόσημο της τιμής της $\bar{L}(x, y)$ όταν τα x, y αντιστοιχούν στο κέντρο της E_3 , καθορίζει σε ποια πλευρά της \bar{L} βρίσκεται η E_3 .

Πόρισμα 0.2.6 Το κ_2 υλοποιείται εκτελώντας το πολύ τις παρακάτω πράξεις:

- i) επίλυση συστήματος 2 εξισώσεων 2 μεταβλητών 2^{ου} βαθμού (υπολογισμός κοινών εφαπτομένων),
- ii) επίλυση 2 τριωνύμων και το πολύ 3 συγκρίσεις μεταξύ αλγεβρικών αριθμών 2^{ου} βαθμού (υπολογισμός διάταξησ-II),
- iii) υπολογισμός τιμής 1 γραμμικού πολυωνύμου 2 μεταβλητών (επιλογή κατάλληλης εξωτερικής εφαπτομένης), και
- iv) υπολογισμός τιμής 1 τετραγωνικού πολυωνύμου 2 μεταβλητών (υπολογισμός σχετικής θέσης εφαπτομένης και έλλειψης).

Όλα τα πολώνυμα έχουν τετραγωνικούς συντελεστές ως προς τις παραμέτρους εισόδου.

0.3 Εξωτερικός εφαπτόμενος Απολλώνιος κύκλος

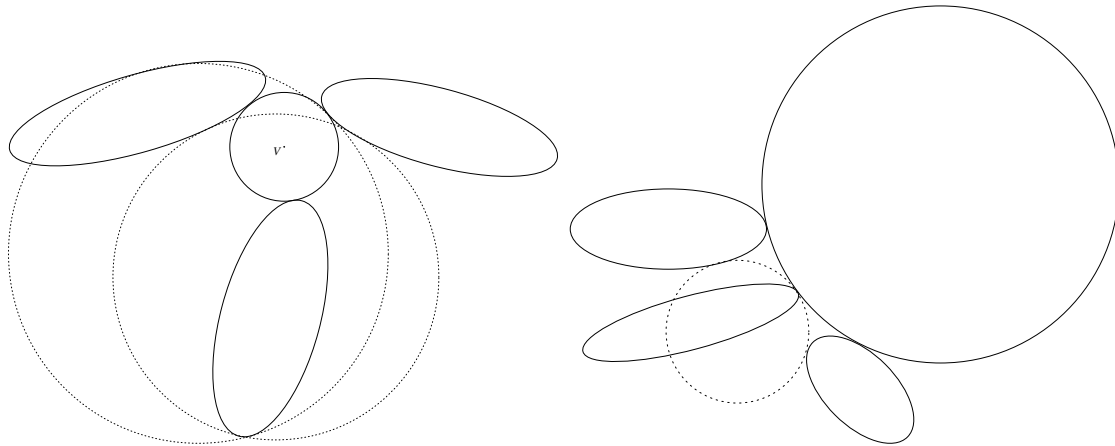
Δοθέντων τριών ελλείψεων της μορφής (1), θέλουμε να βρούμε τον εξωτερικό εφαπτόμενο κύκλο, όπως φαίνεται στο σχήμα 4. Τελικά μας ενδιαφέρει να αποφασίσουμε τη σχετική θέση μιας τέταρτης έλλειψης και του κύκλου. Σημαντικό ανοιχτό ερώτημα παραμένει το μέγιστο πλήθος των πραγματικών εφαπτόμενων κύκλων σε 3 ελλείψεις.

Έστω \sqrt{s} η ακτίνα του εφαπτόμενου κύκλου και (v_1, v_2) το κέντρο του. Παίρνοντας τη διακρίνουσα, όπως παραπάνω, για καθεμία έλλειψη έχουμε:

$$\Delta_1(v_1, v_2, s) = \Delta_2(v_1, v_2, s) = \Delta_3(v_1, v_2, s) = 0. \quad (2)$$

Ανάμεσα στις λύσεις του συστήματος, ο εξωτερικός εφαπτόμενος κύκλος δεν έχει απαραίτητα τη μικρότερη ακτίνα· δείτε τις αντίστοιχες περιπτώσεις στο σχήμα 4.

Εφαρμόζουμε τη θεωρία της αραιής (ή τορικής) απαλοιψής, χρησιμοποιώντας τις ιδιότητες της διακρίνουσας και του μεικτού όγκου. Δίνεται ένα σύστημα $n + 1$ πολυωνύμων f_i με n μεταβλητές, με συντελεστές c_{ij} . Η απαλοιφούσα αυτών των πολυωνύμων είναι ένα νέο πολώνυμο $R \in \mathbb{Z}[c_{ij}]$ τέτοιο ώστε όταν πάρουν τιμές τα c_{ij} , $R = 0 \iff \exists a : \forall i f_i(a) = 0$.



Σχήμα 4: Εφαπτόμενοι κύκλοι σε 3 ελλείψεις· μόνο ένας εφάπτεται εξωτερικά

Αν οι ρίζες a βρίσκονται σε μια προβολική (αντίστ. τορική) ποικιλότητα, τότε αναφερόμαστε στην προβολική (αντίστ. τορική) απαλοίφουσα. Δοθέντων n πολυωνύμων n μεταβλητών, ο μεικτός όγκος αυτού του πολυωνυμικού συστήματος είναι μια συνάρτηση της υποστήριξης⁴ (πολύτοπο του Νεύτωνα) κάθε πολυωνύμου. Ο μεικτός όγκος παρέχει ένα άνω φράγμα για τις ρίζες (πλήθος) του συστήματος στο $(\mathbb{C}^*)^n$. Για περισσότερες πληροφορίες δείτε το [11].

Κάθε Δ_i είναι συνολικού βαθμού 8 στα v_1, v_2, s και 4 στο s . Ο μεικτός όγκος του συστήματος (2) είναι 256 – πολύ υψηλός. Είναι γνωστό ότι αυτό το φράγμα μπορεί να μην είναι σφιχτό, καθώς μπορεί να να απαριθμεί μιγαδικές ρίζες και ρίζες στο «άπειρο».

Στην πραγματικότητα είναι δυνατό να ελαττώσουμε το μεικτό όγκο του παραπάνω συστήματος και να πετύχουμε καλύτερο άνω φράγμα. Θέτουμε

$$q := v_1^2 + v_2^2 - s \quad (3)$$

Σε αυτή την περίπτωση, ο πίνακας B που ορίστηκε σε προηγούμενη ενότητα, περιέχει μόνο γραμμικούς όρους ως προς v_1, v_2, q . Η διακρίνουσα του χαρακτηριστικού πολυωνύμου είναι συνολικού βαθμού 6 στα v_1, v_2, q και 4 στο q : οι συντελεστές των $1, q, q^2, q^3, q^4$ είναι πολυώνυμα ως προς v_1, v_2 βαθμού 6,5,4,2,1 αντιστοίχως. Τώρα το σχετικό σύστημα διακρινουσών έχει μεικτό όγκο 184. Σημειώστε ότι για να λύσουμε ως προς v_1, v_2, s χρειάζεται η εξίσωση (3). Ο μεικτός όγκος του συστήματος των Δ_i επαυξημένο με αυτή την εξίσωση είναι πάλι 184, ως προς v_1, v_2, s, q .

Πρόσφατα αποτελέσματα στη μορφή πίνακα της απαλοίφουσας μας επιτρέπουν να υπολογίζουμε την απαλοίφουσα συγκεκριμένων συστημάτων 3 πολυωνύμων δύο μεταβλητών ως μία απλή ορίζουσα. Μία κατηγορία τέτοιων συστημάτων είναι εκείνα με την ίδια υποστήριξη

⁴Υποστήριξη ενός συστήματος είναι οι βαθμοί των όρων κάθε εξίσωσης. Για παράδειγμα $3x^4y^4 + 2xz = 0$, έχει όρους τα $\{x^4y^4, xz\}$ και η υποστήριξη ως προς (x, y, z) είναι $\{(4, 4, 0), (1, 0, 1)\}$.

[26]. Ο αντίστοιχος πίνακας είναι υβριδικός, περιέχει μπλοκ τύπου Sylvester και μπλοκ τύπου Bézout. Η κατασκευή του πίνακα έχει υλοποιηθεί σε Maple από τον A. Khetan. Το σύστημά μας (2) εντάσσεται σε αυτή την κατηγορία, θεωρώντας μεταβλητές τα v_1, v_2 , αφού «κρύψουμε» το q στους συντελεστές. Η απαλοίφυσά του είναι ένα πολυώνυμο ως προς q και ισούται, γενικά, με την ορίζουσα ενός 58×58 πίνακα. Συμβολίζουμε τον πίνακα αυτό με K .

Για να πάρουμε μια ιδέα για τις ποσότητες που περιλαμβάνονται, μελετήσαμε ένα συγκεκριμένο παράδειγμα τριών ελλείψεων σε τυχαία θέση, όπως στο αριστερό σχήμα 4. Οι παράμετροι εισόδου είναι προσημασμένοι ακέραιοι 10-bit. Τα στοιχεία του K είναι είτε 0 είτε πολυώνυμο ως προς q βαθμού 0–10. Ο υπολογισμός της ορίζουσας του K γίνεται με παρεμβολή. Η ορίζουσα του K είναι ένα πολυώνυμο ως προς q , το οποίο συμβολίζουμε με $d(q)$. Αντικαθιστώντας διαφορετικές τιμές του q στο K απαλοίφουμε τους αγνώστους, έτσι ώστε ο υπολογισμός της ορίζουσας να είναι απλό ζήτημα. Κάνοντας 200 αντικαταστάσεις (στην πραγματικότητα αρκούν 185) πήραμε 200 ζεύγη $\langle q, d(q) \rangle$. Προκύπτει ότι υπάρχει μοναδικό πολυώνυμο παρεμβολής βαθμού 184 ως προς q που περνά από αυτές τις τιμές και είναι ακριβώς η απαλοίφουσα του παραδείγματός μας. Έτσι, σε αυτό το παράδειγμα το πλήθος μιγαδικών ριζών ταιριάζει με το άνω φράγμα που δίνει ο μεικτός όγκος. Οι συντελεστές της απαλοίφουσας είναι κατα μέσο όρο ακέραιοι 1385 ψηφίων (4603-bit). Δεν έχουμε καταφέρει ακόμη να επιλύσουμε την απαλοίφουσα αποτελεσματικά και ακριβώς. Πάντως, σε μια προκαταρκτική προσέγγιση εφαρμόσαμε τη μέθοδο Aberth (υλοποίηση στο [8]) για να επιλύσουμε το πολυώνυμο αριθμητικά. Ο αλγόριθμος έδωσε 8 πραγματικές ρίζες σε λιγότερο από ένα δευτερόλεπτο.

Ενώ η αριθμητική επίλυση του πολυωνύμου της απαλοίφουσας γίνεται πολύ γρήγορα, ο υπολογισμός (των συντελεστών) του πολυωνύμου παίρνει αρκετό χρόνο. Ωστόσο, μπορούμε να επιλύσουμε αριθμητικά το ίδιο το σύστημα ως εξής: Θεωρούμε ένα σύστημα ισοδύναμο με το (2) και το οποίο έχει τον ίδιο μεικτό όγκο. Κάθε $\Delta_i(v_1, v_2, q)$ είναι η διακρίνουσα του $\phi_i(\lambda, v_1, v_2, q)$. Επειδή ο μηδενισμός της διακρίνουσας αντιστοιχεί στο μηδενισμό της πρώτης παραγώγου, παίρνουμε το 6×6 σύστημα:

$$\begin{aligned} \phi_i(\lambda_i, v_1, v_2, q) &= 0 \\ \frac{\partial}{\partial \lambda_i} \phi_i(\lambda_i, v_1, v_2, q) &= 0 \end{aligned} \quad (4)$$

όπου $i = 1, 2, 3$. Το σύστημα αυτό έχει πολύ πιο μικρούς συντελεστές από το (2) και περιλαμβάνει τρεις εξισώσεις συνολικού βαθμού 3 και τρεις συνολικού βαθμού 2. Τώρα είναι εύκολο να επιλύσουμε το σύστημα αριθμητικά με το PHECrack [34], το οποίο εκμεταλλεύεται την αραιότητα και χρησιμοποιεί μεθόδους συνέχειας της ομοτοπίας.

Θεώρημα 0.3.1 *Το σύστημα (2) έχει το πολύ 184 μιγαδικές ρίζες. Το φράγμα αυτό είναι σφιχτό, δηλαδή υπάρχουν τριάδες ελλείψεων με αυτό το πλήθος μιγαδικών Απολλώνιων κύκλων.*

Το αποτέλεσμα γενικεύεται για όλες τις κωνικές τομές. Σύμφωνα με το [33], υπάρχουν 184 μιγαδικοί κύκλοι στη χειρότερη περίπτωση που εφάπτονται σε 3 δοθείσες κωνικές τομές στο επίπεδο. Ωστόσο παραμένει ακόμη ανοιχτό το ερώτημα πόσοι πραγματικοί κύκλοι υπάρχουν. Τα δικά μας πειράματα έδωσαν μέχρι 23 τέτοιους κύκλους.

Για να ελαττώσουμε την αλγεβρική πολυπλοκότητα του προβλήματος, μπορούμε να θεωρήσουμε διάφορες ημι-αλγεβρικές συνθήκες. Αυτές οι ανισότητες χρησιμεύουν στο να επιλέγουμε τις κατάλληλες ρίζες όταν μας δίνεται μια (αριθμητική) λύση του συστήματος, μιας και ενδιαφερόμαστε μόνο για τον εξωτερικό Απολλώνιο κύκλο. Στο κεφάλαιο 3 γίνεται μια πιο λεπτομερής ανάλυση του προβλήματος και δίνουμε ημι-αλγεβρικές συνθήκες καθώς και περιορισμούς που προκύπτουν από τη γεωμετρία του προβλήματος. Αναμένουμε οι συνθήκες αυτές να οδηγήσουν σε έναν αποτελεσματικό αλγόριθμο υποδιαίρεσης.

0.4 Υλοποίηση

Υλοποιήσαμε τα κατηγορήματα κ_1 και κ_2 χρησιμοποιώντας τη SYNAPS [14], μια βιβλιοθήκη C++ για συμβολικούς-αριθμητικούς υπολογισμούς. Η SYNAPS παρέχει μια αποτελεσματική υλοποίηση βασισμένη στους αλγόριθμους του [16] για λογισμό με αλγεβρικούς αριθμούς μικρού βαθμού, επίλυση πολυωνυμικών συστημάτων 2 μεταβλητών, και υπολογισμό του προσήμου της τιμής πολυωνύμου δύο μεταβλητών πάνω σε δύο αλγεβρικούς αριθμούς. Χρησιμοποιήσαμε εκτεταμένη ακέραια αριθμητική GMP⁵.

Για την αριθμητική επίλυση του συστήματος που αντιστοιχεί στο τρίτο κατηγορήμα, χρησιμοποιήσαμε το PHCrack με την επιλογή «μαύρο κουτί», αφού πρώτα κλιμακώσαμε κατάλληλα τους συντελεστές.

Για κάθε τεστ δημιουργήσαμε 1000 τυχαία παραδείγματα (σημείο και 2 ελλείψεις ή 3 ελλείψεις) με συντελεστές ομοιόμορφα κατανομημένους μεταξύ 0 και 2^B , $B \in \{10, 30, 100, 300\}$. Τα τεστ έτρεξαν σε έναν P4 2.6GHz-CPU με 1GB RAM, με Debian Linux πυρήνα 2.6.10. Ο πίνακας 1 συνοψίζει τους μέσους χρόνους· για τα κ_1 και κ_2 , οι χρόνοι εκτέλεσης αυξάνονται υποτετραγωνικά ως προς B . Σημειώστε πως για το κ_2 , ο μισός χρόνος αναλώνεται στην επίλυση του συστήματος και ο άλλος μισός για τον υπολογισμό της σχετικής θέσης της τρίτης έλλειψης. Συγκρινόμενο με το [19], το κ_2 είναι πιο ισχυρό. Ενώ εκεί υπολογίζεται το (χ_1) σε συνολικό χρόνο 6ms, το κ_2 , στον ίδιο χρόνο, παρέχει περισσότερη πληροφορία.

Για το κ_3 , οι χρόνοι αναφέρονται στην επίλυση του αντίστοιχου συστήματος. Είναι σχεδόν ανεξάρτητοι του μεγέθους της εισόδου σε bit, αφού χρησιμοποιείται αριθμητική διπλής ακρίβειας. Ακόμα δεν είμαστε σε θέση να εκτιμήσουμε την αριθμητική ακρίβεια, ούτε τον ακριβή αριθμό πραγματικών λύσεων. Στο [3] το προτεινόμενο σύστημα για το κ_3 έχει υποβέλτιστο μεικτό όγκο και συνεπώς μεγαλύτερη πολυπλοκότητα. Αν και εκεί υπολογίζονται

⁵www.swox.com/gmp/

B	κατηγορήμα κ_1 [ms]	κατηγορήμα κ_2 [ms]	κατηγορήμα κ_3 [sec]
10	0.45	6.15	23.85
30	0.94	16.46	34.52
100	3.68	73.21	38.36
300	17.3	396.82	38.44

Πίνακας 1: Μέσοι χρόνοι της υλοποίησής μας

επίσης προσεγγιστικές αριθμητικές λύσεις, οι μέθοδοι πινάκων που χρησιμοποιούνται φαίνονται πιο αργές από τις δικές μας.

0.5 Επίλογος

Ο υπολογισμός του διαγράμματος Νογοποι καμπύλων αντικειμένων μπορεί να απαιτήσει πολύπλοκα κατηγορήματα. Σε αυτή τη διατριβή μελετήσαμε μερικά από τα κατηγορήματα που απαιτούνται στην περίπτωση που τα καμπύλα αντικείμενα είναι ελλείψεις.

Δώσαμε έναν απλό τρόπο υπολογισμού ενός ενός εσωτερικού σημείου μιας έλλειψης. Προτείναμε μια αποτελεσματική αναπαράσταση της απόστασης μεταξύ σημείου και έλλειψης η οποία οδήγησε σε μια αποτελεσματική υλοποίηση του αντίστοιχου κατηγορήματος. Ακολουθώντας μια παρόμοια προσέγγιση, μπορέσαμε να διακρίνουμε τη σχετική θέση μιας τρίτης έλλειψης ως προς την κοινή εφαπτομένη δύο άλλων. Ο τρόπος που υπολογίζεται το κατηγορήμα μας επιτρέπει να χαρακτηρίζουμε πλήρως όλες τις κοινές εφαπτομένες.

Βασισμένοι σε αυτά τα αποτελέσματα μελετήσαμε το πρόβλημα του Απολλώνιου κύκλου και την πολυπλοκότητά του χρησιμοποιώντας εργαλεία όπως ο μεικτός όγκος και η απαλοίφουσα. Είμαστε οι πρώτοι που δώσαμε ένα σφιχτό άνω φράγμα για το πλήθος των μιγαδικών Απολλώνιων κύκλων.

Αν και λύνουμε το αντίστοιχο σύστημα αριθμητικά, παρέχουμε ημιαλγεβρικές συνθήκες και γεωμετρικούς περιορισμούς ώστε να μπορούμε να αποκλείσουμε περιπτώσεις με κάποιον αλγόριθμο υποδιαίρεσης. Σε μελλοντική εργασία, ένας τέτοιος αλγόριθμος θα μπορούσε να οδηγήσει σε μια αποτελεσματική και ακριβή υλοποίηση πραγματικού χρόνου.

Μέρος Β'

Chapter 1

Preliminaries

Voronoi diagrams are interesting constructs with numerous applications which include motion planning, collision detection, and pattern matching. They have been studied extensively [7], however the bulk of the existing work in the plane concerns point-sites or linear sites such as segments [22] and polygons [32]. More recently, some efforts have been undertaken to extend Voronoi diagrams to the case where the sites are curves (e.g. [1, 3, 6]) or have non-empty interior [10, 13, 9]. The most direct precursor to our work is the diagram of circles, e.g. [4, 24, 27]. For additively-weighted diagrams, the implementation of [15] is now part of the standard distribution of the CGAL library; an important step was the use of efficient predicates from [23]. Among those were specialised versions of our predicates (κ_2) and (κ_3) , whereas (κ_0) and (κ_1) become trivial in the case of circles.

A related class of problems concerns the computation of visibility maps among ellipses, which generalizes [2]. In these problems, the computation and the characterisation (as external or internal) of all bitangents of two ellipses is of independent interest. Three crucial predicates for visibility problems are χ_2 , χ_1 and χ_0 .

(χ_2) computes the sign of the determinant of two oriented bitangents.

(χ_1) is a specialization of χ_2 , in case where both bitangents are tangent to the same ellipse.

(χ_0) takes three bitangents, all of them are tangent to the same ellipse, and computes the circle order of their tangency points to this ellipse.

These are discussed in [2, 15] for circles and, currently, in [19] for ellipses. We can apply our approach to answer χ_i 's or vice versa. However, our approach is more general, since we compute and characterize all the bitangents, without losing in efficiency.

Recent works derive (semi-)algebraic conditions for characterizing the relative position of conics in the plane or certain quadrics in space, e.g. [18, 35, 36]. We draw upon those and propose some simpler conditions in our specific setting.

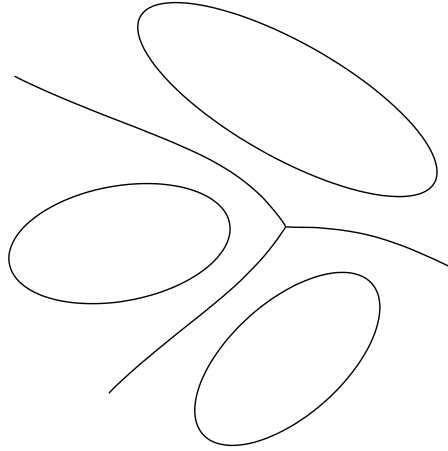


Figure 1.1: Voronoi diagram of three ellipses

Our ultimate goal is to compute efficiently and exactly the Voronoi (or Apollonius) diagram of arbitrary sets of ellipses in the plane, under the Euclidean metric. Figure 1.1 shows such a diagram of three random ellipses.

We assume that the ellipses are given algebraically, or implicitly. This is clearly a harder problem than the diagram of circles or the visibility map among ellipses, hence the need for higher degree algebraic operations. As a first step, we study the case of *non-intersecting* ellipses.

Our approach relies on the general framework of *abstract* Voronoi diagrams, e.g. [28]. More specifically we follow [25], which proposes a dynamic algorithm for computing the Voronoi diagram of convex objects where the boundaries of any two of them intersect at most twice. Non-intersecting ellipses enter in this framework. The approach has two *phases*. First, locate a Voronoi cell which changes due to the inserted ellipse and, second, detect and update all such cells. The algorithm depends on the following construction and 3 predicates:

- (κ_0) determine a point inside a given ellipse
- (κ_1) given two ellipses and a point outside of both, decide which is the ellipse closest to the point, under the Euclidean metric
- (κ_2) given three ellipses, decide the relative position of the third one with respect to a specific external bitangent of the first two
- (κ_3) given 4 ellipses, decide the relative position of the fourth one with respect to the external tritangent Voronoi circle of the first three

Construction (κ_0) is used in conjunction with all other predicates and is straightforward to implement. Predicate (κ_1) suffices for the first phase of the algorithm, where it is enough to find the ellipse of the diagram closest to any point of the new ellipse; this point is provided by (κ_0) . Predicates (κ_0) and (κ_3) are used in the second phase. Updating the cells that change may also require some constructions, which shall be addressed in a future work.

For predicate (κ_1) we consider a circle, centered at the point, with unknown radius, which corresponds to the distance to be compared. The existence of a tangency point between the circle and the ellipse is equivalent to the vanishing of the discriminant of the characteristic polynomial of the conics. It is equal to the determinant of the pencil defined by the circle and the ellipse. One of our main contributions is a method for (κ_1) that uses algebraic numbers of degree 4, which is optimal. Note that we avoid expressing the coordinates of the tangency point.

In order to decide κ_2 we compute and characterize all the bitangents of two ellipses as roots of a bivariate quadratic polynomial system. We avoid to compute the tangency points, as in κ_1 . Our approach is exact, optimal in the algebraic degree and handles efficiently all degenerate cases, i.e. vertical bitangents. We provide efficient implementation in SYNAPS ([14]) for both predicates and we answer them in an exact way.

Let us focus on predicate (κ_3) . In the case of 3 disks, the number of tritangent circles is 8 and the corresponding predicate is of algebraic degree 2 [15]. This problem is also known as *Apollonius Tenth problem*, because it was first addressed by *Apollonius of Perga*, in about 250 BC. While this has been known since antiquity, the generalization to ellipses is yet to be solved efficiently.

Even the number of tritangent circles to 3 ellipses is not known. The problem involves equations of high degree and obtaining an exact solution is nontrivial ([3]). [31] attempts to deal with this problem, but exact computation with the proposed method is not completed and the author reverts to numerical methods. No bounds on the complexity are given, nor on the number of tritangent circles.

We apply the method from predicate (κ_1) and recent advances in sparse (or toric) resultants in order to project all common roots to those of a univariate equation. This leads to the first interesting bound on the number of tritangent circles. Mixed volume also gives this bound as does a real algebraic geometry argument [33].

Our proposal is that geometric reasoning and semi-algebraic tools must enhance the purely algebraic methods, towards an efficient and exact implementation. In this direction, we contribute new conditions, based on Descartes' rule. We analyze their geometric significance in order to distinguish cases whose algebraic complexity is lower than the general problem, expecting that this, enhanced with novel subdivision techniques like [32], shall lead to a real-time algorithmic solution. Currently we solve the corresponding system numerically using PHCpack ([34]).

This master's thesis is organised as follows. First, we introduce some of the ellipse's

properties and settle construction (κ_0) . Then we propose an efficient representation for the Euclidean distance between a point and an ellipse and we apply this idea to solve predicate (κ_1) . After that we deal with the bitangent line, and finally with the external tritangent Apollonius circle. At the end, we present the implementation of (κ_1) and (κ_2) .

1.1 The Ellipse

An *ellipse* is the locus of points in the plane the sum of whose distances from two fixed points F_1, F_2 is a given positive constant 2α . The points F_1 and F_2 are called the *foci* of the ellipse and we'll assume that they are separated by a distance of 2γ . The ellipse is a *conic* section. In cartesian coordinates, let $P = (x, y)$ be a point on the ellipse, $F_1 = (-\gamma, 0)$ and $F_2 = (\gamma, 0)$, that is we consider an ellipse along the x -axis, centered at the origin $O(0, 0)$. Then the following equation holds:

$$\frac{x^2}{\alpha^2} + \frac{y^2}{\beta^2} = 1 \quad (1.1)$$

where $\beta^2 = \alpha^2 - \gamma^2$. In this case, where the center of the ellipse is the origin of the coordinate system and the foci lie on the horizontal axis, we say that the ellipse is in *orthogonal* position.

Let us now consider the ellipse in *generic* position, where $\overline{F_1F_2}$ is not parallel to the x -axis, but has been rotated counter-clockwise by u radians¹, while its center lies at (x_c, y_c) . In this case we consider a new coordinate system centered at the center of the ellipse and with axes parallel to those of the ellipse. Every point (x, y) of the initial coordinate system maps to a new point (X, Y) in the new system where:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} \cos(u) & \sin(u) \\ -\sin(u) & \cos(u) \end{pmatrix} \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix} \quad (1.2)$$

It is obvious from the cartesian equation 1.1 that an ellipse in orthogonal position can be written in a simple parametric form:

$$\begin{aligned} x(t) &= \alpha \cos(t) \\ y(t) &= \beta \sin(t) \end{aligned}$$

where $t \in [0, 2\pi)$.

To find the parametric form of an ellipse in generic position, we multiply the rotation matrix

$$\begin{pmatrix} \cos(u) & -\sin(u) \\ \sin(u) & \cos(u) \end{pmatrix}$$

by the vector $(x(t), y(t))^T$ adding the offset $(x_c, y_c)^T$ and get

$$\begin{aligned} x(t) &= \alpha \cos(u) \cos(t) - \beta \sin(u) \sin(t) + x_c \\ y(t) &= \alpha \sin(u) \cos(t) + \beta \cos(u) \sin(t) + y_c \end{aligned}$$

¹Throughout this document all angles and arcs will be measured in radians, so the unit will not be mentioned.

Let $z = \tan(\frac{t}{2})$, $w = \tan(\frac{u}{2})$. Then $\sin(t) = \frac{2z}{1+z^2}$, $\cos(t) = \frac{1-z^2}{1+z^2}$. We can now rewrite the parametric equations as follows

$$\begin{aligned} x(z) &= \frac{-\alpha(1-w^2)z^2 - 4\beta wz + \alpha(1-w^2)}{(1+w^2)(1+z^2)} + x_c \\ y(z) &= \frac{2(-\alpha wz^2 + \beta(1-w^2)z + \alpha w)}{(1+w^2)(1+z^2)} + y_c \end{aligned}$$

where $z \in (-\infty, \infty)$.

Now assume that $(dx, dy) = (0, 0)$. The equation of the large axis of E is

$$\begin{vmatrix} x & y & 1 \\ 0 & 0 & 1 \\ x(0) & y(0) & 1 \end{vmatrix} = 0$$

which simplifies to

$$e_x(x, y) = yw^2 + 2xw - y = 0 \quad (1.3)$$

Similarly, the equation of the small axis of E is

$$\begin{vmatrix} x & y & 1 \\ 0 & 0 & 1 \\ x(1) & y(1) & 1 \end{vmatrix} = 0$$

which simplifies to

$$e_y(x, y) = xw^2 - 2yw - x = 0 \quad (1.4)$$

From equations 1.1 or 1.2 we can rewrite the equation of an ellipse in the generic form that every conic satisfies:

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0 \quad (1.5)$$

where

$$\begin{aligned} a &= \alpha^2 \sin^2 u + \beta^2 \cos^2 u \\ b &= -\frac{\gamma^2}{2} \sin(2u) \\ c &= \alpha^2 \cos^2 u + \beta^2 \sin^2 u \\ d &= \frac{\gamma^2}{2} y_c \sin(2u) + \gamma^2 x_c \cos^2 u - \alpha^2 x_c \\ e &= \frac{\gamma^2}{2} x_c \sin(2u) - \gamma^2 y_c \cos^2 u - \beta^2 y_c \\ f &= \gamma^2 (y_c^2 - x_c^2) \cos^2 u - \gamma^2 x_c y_c \sin(2u) + \alpha^2 x_c^2 + \beta^2 y_c^2 - \alpha^2 \beta^2 \end{aligned}$$

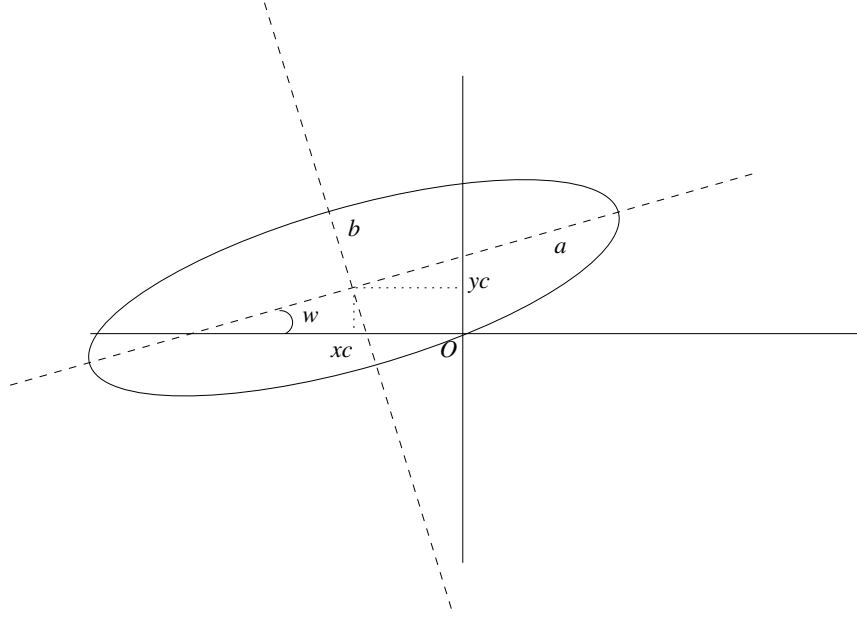


Figure 1.2: An ellipse in generic position

The above equations show the relation between the six parameters of the ellipse $\{a, b, c, d, e, f\}$ and its basic geometric characteristics – centre (x_c, y_c) , rotation (u) and axes $\{\alpha, \beta\}$.

The following quantities remain constant, no matter what transformations (rotation, translation) are applied:

$$\begin{aligned}
 J_1 &= a + c &= \alpha^2 + \beta^2 \\
 J_2 &= \begin{vmatrix} a & b \\ b & c \end{vmatrix} &= \alpha^2 \beta^2 \\
 J_3 &= \begin{vmatrix} a & b & d \\ b & c & e \\ d & e & f \end{vmatrix} \\
 &= 2bed - e^2a - d^2c - b^2f + acf = -\alpha^4 \beta^4 = -J_2^2
 \end{aligned}$$

The following quantity is invariant under rotation; its expression uses the lemma below.

$$\begin{aligned}
 J_4 &= (a + c)f - d^2 - e^2 \\
 &= -\alpha^2 \beta^2 (\alpha^2 + \beta^2 - x_c^2 - y_c^2) \\
 &= J_2 (x_c^2 + y_c^2 - J_1).
 \end{aligned}$$

Lemma 1.1.1 Consider an ellipse E of the form (1.5). Let L_y be the line connecting the leftmost with the rightmost point of the ellipse. Let L_x be the line connecting the topmost

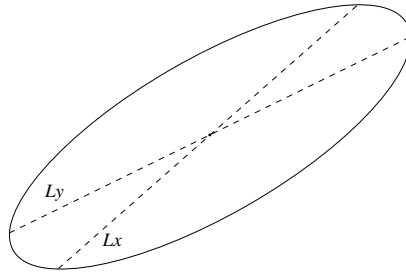


Figure 1.3: The lines intersect at the center of the ellipse

with the bottom-most point. Then L_x and L_y intersect at the center of the ellipse (x_c, y_c) (fig. 1.3) and its coordinates are a rational expression of the parameters of the ellipse:

$$x_c = \frac{be - dc}{J_2}, \quad y_c = \frac{bd - ae}{J_2}$$

Proof. $L_x = 0 \iff \frac{\partial E}{\partial x} = 0 \iff ax + by + d = 0$. Similarly $L_y = 0 \iff \frac{\partial E}{\partial y} = 0 \iff bx + cy + e = 0$. Solving the linear system $\{L_x = 0, L_y = 0\}$ we obtain the solution $\{x = \frac{be - dc}{ac - b^2}, y = \frac{bd - ae}{ac - b^2}\}$. Moreover, from the relation between $\{a, b, c, d, e\}$ and $\{x_c, y_c, \alpha^2, \beta^2, \sin(2u), \cos^2 u\}$ it is obvious that the center of the ellipse (x_c, y_c) satisfies both $L_x = 0$ and $L_y = 0$.² \square

The above lemma leads to a straightforward implementation of construction (κ_0) . In the case of disks we may also pick their centers, though here the latter has coordinates of degree 2 in the input data (a, b, c, d, e, f) .

Given a point V outside an ellipse, how many normals are there to the ellipse? Let us count normal *segments*, defined as the segment of a line normal to the ellipse at some point Q ; the segment's endpoints are Q, V . The boundary of the regions where the number of normals changes is the *evolute*, which is a stretched astroid [21]. For an ellipse in orthogonal position, each point (x, y) on the evolute satisfies:

$$(\alpha x)^{\frac{2}{3}} + (\beta y)^{\frac{2}{3}} = \gamma^{\frac{4}{3}}$$

Proposition 1.1.2 *There are 4, 3 or 2 normals of a point to an ellipse, depending on whether the point lies inside the evolute, lies on the evolute but not at a cusp or, respectively, the point is a cusp or outside the evolute.*

This yields a lower bound on the algebraic complexity of computing the distance of an external point to the ellipse, since any condition on the unknown distance has degree ≥ 4 .

²An alternative way to obtain the centre is that d and e must vanish after translation. This gives a 2×2 system.

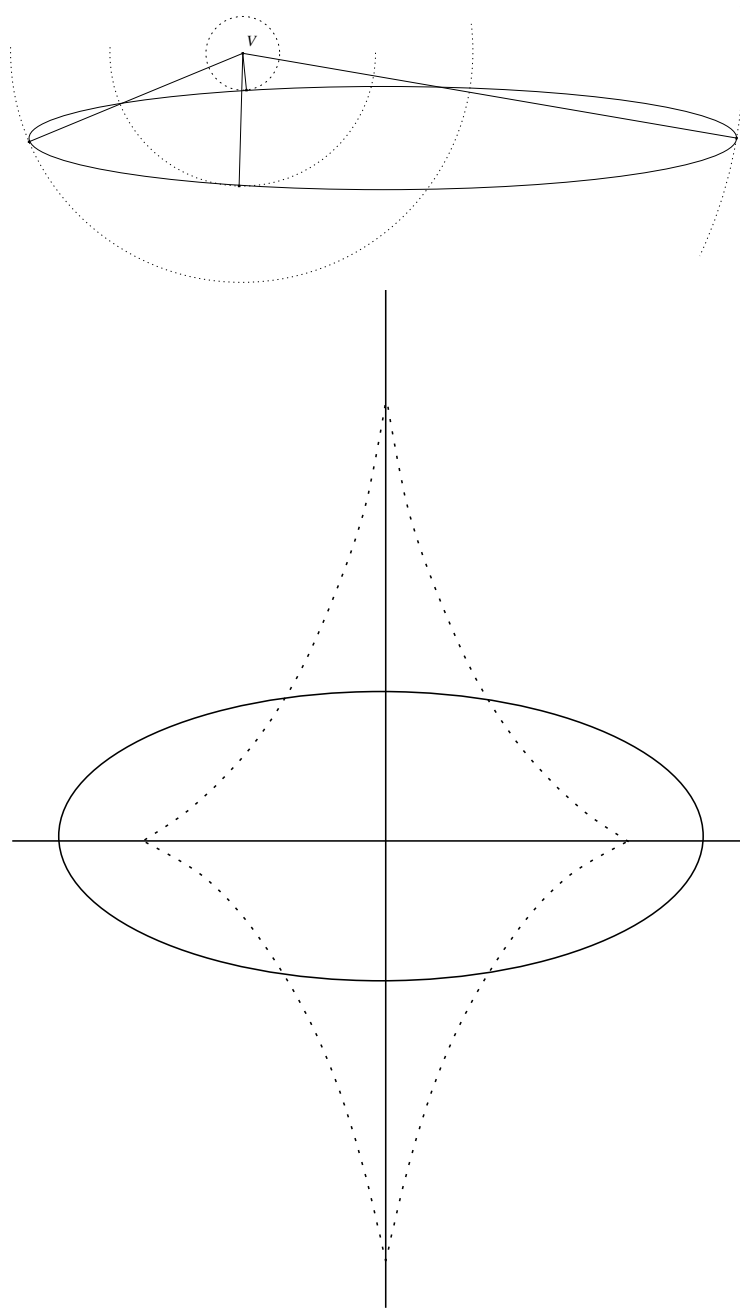


Figure 1.4: Top: an example of a point with 4 normals. Bottom: the evolute of an ellipse.

Chapter 2

Distance between point and ellipse

Consider an ellipse E and a point $V = (v_1, v_2)$ outside E . Let C be a circle centered at V with radius equal to \sqrt{s} , for a real $s > 0$. We shall express the *Euclidean distance* $\delta(V, E)$ between V and E by the *smallest* positive value of \sqrt{s} for which C is tangent to E . In comparing distances, it is sufficient to consider squared distance s^1 .

It would be possible to find all tangency points $P(x, y)$ by solving the system:

$$\begin{cases} E(x, y) = 0 \\ \left[\vec{\nabla} E(x, y) \times (\vec{P} - \vec{V}) \right]_3 = 0 \end{cases} \quad (2.1)$$

and then choosing the appropriate solution, where the 2nd equation constraints the vector $\vec{P} - \vec{V}$ to be parallel to $\vec{\nabla} E$ at (x, y) (and thus normal to E) by vanishing the third coordinate of their cross-product. Consider system (2.1) with an additional equation: $\|\vec{P} - \vec{V}\|^2 = s$. The resultant of the 3 polynomials with respect to x, y is precisely the

¹Note that for the voronoi diagram the definition of the distance is extended to points interior to the ellipse, as well. In this case, the distance from the ellipse is assumed to be negative.

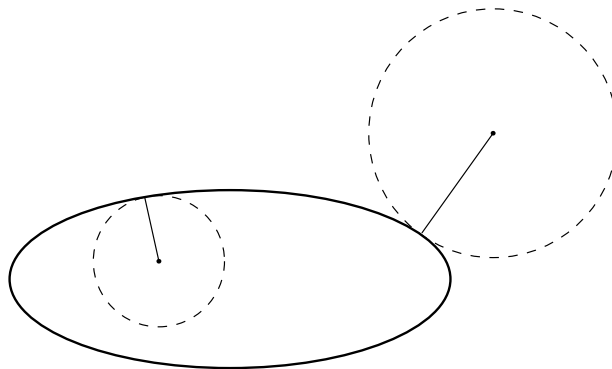


Figure 2.1: Distance between point and ellipse

polynomial $\Delta(v_1, v_2, s)$ to be defined below by an alternative manner. It is the algebraic representation of the offset curve to E at distance s .

From now on, we 'll use a more compact representation for system 2.1:

$$E(x, y) = \det[\nabla E(x, y), (x, y) - V] = 0 \quad (2.2)$$

2.1 Parametric approach

We shall now try to compute $\delta(V, E)$ using the parametric equations of E .

Lemma 2.1.1 *Let E be an ellipse centered at O , V a point on the plane outside E and $P \in E$ such that $\delta(V, E) = |\overrightarrow{VP}|$. Then P and V do not lie on different sides of the large axis of E .*

Proof Assume that P and V lie on different sides of the large axis of E which we will denote by O_x . Obviously the segment PV crosses O_x at some point X . If X is interior to the ellipse E , then the segment XV crosses the boundary of E at some point Y and we have $|PV| = |PX| + |XY| + |YV|$. Consequently, $|YV| < |PV|$ which contradicts our assumption that P is the point on E that is closest to V . If X is exterior to E then we consider two cases:

- i) Both P and V are on the same side of the small axis O_y of E . In this case, we consider the tangent of E that is perpendicular to O_x and crosses O_y on the point A that lies on the same side of O_y together with V and P . The angle \widehat{VAP} is greater than $\pi/2$ radians, and therefore $|AV| < |PV|$ which contradicts our assumption.
- ii) P and V lie on different sides of O_y . We analogously consider the tangent of E perpendicular to O_x crossing O_y on the point A that lies on the same side of O_y together with V . Again, angle \widehat{VAP} is greater than $\pi/2$ radians, and therefore we get $|AV| < |PV|$, which is a contradiction. \square

Let us now compute P . We assume that the coordinates of P are $(x(z), y(z))$ as defined in the previous section and the coordinates of V are (v_1, v_2) . Then we have²

$$\begin{aligned} \delta(z) = |\overrightarrow{VP}| = & \\ & \left(v_1 - \frac{-\alpha(1-w^2)z^2 - 4\beta wz + \alpha(1-w^2)}{(1+w^2)(1+z^2)} \right)^2 \\ & + \left(v_2 - \frac{2(-\alpha wz^2 + \beta(1-w^2)z + \alpha w)}{(1+w^2)(1+z^2)} \right)^2 \end{aligned} \quad (2.3)$$

²We actually compute the square of the distance.

The function $\delta(z)$ is differentiable and therefore it gets its minimum value at some root ζ of its derivative $\delta'(z)$.

By Lemma 2.1.1 we have that

$$(v_2w^2 + 2v_1w - v_2)(y(\zeta)w^2 + 2x(\zeta)w - y(\zeta)) \geq 0$$

or simply

$$-\zeta e_x(v_1, v_2) \geq 0 \tag{2.4}$$

The derivative of $\delta(z)$ is

$$\begin{aligned} \delta'(z) = & \frac{4}{(1+z^2)^3(1+w^2)} \left(-\beta(v_2w^2 + 2v_1w - v_2)z^4 + \right. \\ & 2[(\alpha^2 - \beta^2 - v_1\alpha)w^2 + 2v_2\alpha w + \alpha^2 - \beta^2 + v_1\alpha]z^3 + \\ & 2[-(\alpha^2 - \beta^2 - v_1\alpha)w^2 + 2v_2\alpha w - (\alpha^2 - \beta^2) + v_1\alpha]z + \\ & \left. b(v_2w^2 + 2v_1w - v_2) \right) \end{aligned}$$

It is clear from the above equation that $\delta'(z)$ has at least one positive and one negative root, as the product of the roots of $\delta'(z)$ is negative. Thus, we can always pick either the smallest or the largest root of $\delta'(z)$, so that equation (2.4) holds.

Lemma 2.1.2 *Exactly one root ζ of $\delta'(z)$ satisfies equation (2.4).*

Proof From equation (2.4) it is obvious that the sign of (1.3) with x and y replaced by $x(z)$ and $y(z)$ is $-z$. Positive and negative values of z correspond to points on the ellipse that lie on different sides of its large axis. For our proof we will consider the case where there is no rotation, that is $w = 0$. If $w \neq 0$ then we could rotate everything the inverse way (by $-w$) so that the horizontal axis x becomes the large axis of the ellipse. When $w = 0$, $\delta'(z)$ has the simpler form:

$$\delta'(z) = \frac{4}{(1+z^2)^3} \left(\beta v_2 z^4 + 2[\alpha^2 - \beta^2 + v_1\alpha]z^3 + 2[-(\alpha^2 - \beta^2) + v_1\alpha]z - b v_2 \right)$$

and equation (2.4) becomes:

$$\zeta v_2 \geq 0$$

- $v_2 > 0$. We shall prove that $\delta'(z)$ has exactly one positive root. Considering the sign variations of the coefficients of $\delta'(z)$ and using Descartes's rule we have the following cases:

sign	upper bound of positive roots
(+, -, -, -)	1
(+, +, -, -)	1
(+, -, +, -)	3
(+, +, +, -)	1

Since $\delta'(z)$ has at least one positive root, it suffices to prove that the sign variations of the third case, $(+, -, +, -)$ cannot exist. Let C_n be the coefficient of z^n in $\delta'(z)$ divided by the positive quantity $\frac{4}{(1+z^2)^3}$. Then $C_3 = 2[\alpha^2 - \beta^2 + v_1\alpha]$ and $C_1 = 2[-(\alpha^2 - \beta^2) + v_1\alpha]$. If $C_3 < 0$ then we have equivalently:

$$\begin{aligned} C_3 &< 0 \\ \alpha^2 - \beta^2 + v_1\alpha &< 0 \\ -2(\alpha^2 - \beta^2) + \alpha^2 - \beta^2 + v_1\alpha &< 0 \\ -(\alpha^2 - \beta^2) + v_1\alpha &< 0 \\ C_1 &< 0 \end{aligned}$$

This means that there cannot be $C_3 < 0$ and $C_1 > 0$ at the same time.

- $v_2 < 0$. We can easily prove working in a similar manner that $\delta'(-z)$ has exactly one positive root. \square

We can now implement a function that given an ellipse E and a point V outside E , it computes the point $P(z)$ on E so that $|\overrightarrow{VP}| = \delta(V, E)$. The algorithm computes the sign of $e_x(v_1, v_2)$ in order to determine which side of the large axis of E , V lies on. A special case exists when $e_x(v_1, v_2) = 0$, that is when V lies *on* the large axis of E . In this case, it can be easily shown that P lies on the same side of the small axis of E as V and P equals to either $P(0)$ or $P(\infty)$. Thus if $e_y(v_1, v_2)e_y(x(0), y(0)) > 0$, then we have equivalently that $e_y(v_1, v_2) < 0$.

```

function FindPz( $a, b, w, v_1, v_2$ )
1       $S_x := v_2w^2 + 2v_1w - v_2$ ;
2       $S_y := v_1w^2 - 2v_2w - v_1$ ;
3      if  $S_x = 0$  then
4          if  $S_y < 0$  then
5              return 0;
          else
6              return  $\infty$ ;
          end if
7      else if  $S_x < 0$  then
8           $rootindex := 3$ ;
9      else if  $S_x > 0$  then
10          $rootindex := 0$ ;
      end if
11      $A := (\alpha^2 - \beta^2 - v_1\alpha)w^2 + \alpha^2 - \beta^2$ ;

```



```

12      B := 2v2αw + v1α;
13      dz := -βS_x z^4 + 2(A + B)z^3 - 2(A - B)z + βS_x;
14      return RootOf[dz, rootindex];
end function

```

2.2 Algebraic approach

In this section we follow an algebraic approach in order to compute $\delta(V, E)$. The idea is to avoid explicit computation of the tangency points.

2.2.1 The resultant

The simplest way is the following. The system $E = C = 0$ must have a multiple root for the circle to be tangent to the ellipse. Thus, their resultant with respect to x (or y), which is of degree 4, has a multiple root. The discriminant system (e.g. [16]) expresses this condition effectively.

In the special case where V lies on E we define $\delta(V, E) := 0$. We can easily see that $\delta(V, E) = (b - v_2)^2$, when $v_1 = 0$. We'll now compute $\delta(V, E)$ when $v_1 \neq 0$ and $v_2 \neq 0$ and V does not lie on E , taking the latter in orthogonal position.

Consider the equation system of E and C :

$$\begin{aligned}
E: \quad & \beta^2 x^2 + \alpha^2 y^2 - \alpha^2 \beta^2 = 0 \\
C: \quad & (x - v_1)^2 + (y - v_2)^2 - s = 0
\end{aligned}$$

The resultant of the above system (with respect to y) is a polynomial of x of degree 4:

$$\begin{aligned}
p(x) = & (\alpha^2 - \beta^2)^2 x^4 \\
& - 4\alpha^2 v_1 (\alpha^2 - \beta^2) x^3 \\
& + 2\alpha^2 (-\beta^2 v_1^2 + \beta^2 s + 3\alpha^2 v_1^2 + \beta^2 v_2^2 - \alpha^2 s + \alpha^2 v_2^2 + \alpha^2 \beta^2 - \beta^4) x^2 \\
& - 4\alpha^4 v_1 (\beta^2 + v_2^2 + v_1^2 - s) x \\
& + \alpha^4 ((v_1^2 + v_2^2 + \beta^2 - s)^2 - 4\beta^2 v_2^2)
\end{aligned}$$

Now let us examine some useful properties of the system:

Proposition 2.2.1 $p(x) = 0$ if and only if C and E have a common point.

Proof It is obvious from the definition of the resultant (and because the coefficient of y^2 in E is nonzero). \square

Proposition 2.2.2 If $s < 0$ then $p(x)$ has no real roots.

Proof Since s is the square of the radius of a circle, it must be non-negative, or we 'll have a complex circle. Obviously, if we have no real circle, $p(x)$ will have no real roots. \square

Proposition 2.2.3 $\delta(C, E) = s \implies$ all real roots of $p(x)$ are of even multiplicity

Proof Let K be the center of the circle. It is clear that there exists at least one real root x , since there is a common point P of C and E such that $\overline{PK}^2 = s$. If there existed a root of odd multiplicity, then the ellipse would cross the circle at that point. But then there would exist a point P' of the ellipse interior to the circle. In that case we would have $\overline{P'K}^2 < s$ which is a contradiction. So x (and any other real root that may exist) is a root of even multiplicity. \square

Proposition 2.2.4 As s grows from 0 to infinity, the first time that $p(x)$ has real roots, all of them are of even multiplicity. In that case, $\delta(C, E) = s$.

Proof It is obvious that as s grows, the corresponding circle C grows and it just touches the ellipse E before crossing it. \square

From the above properties the following theorem obviously holds:

Theorem 2.2.5 The smallest value of s for which $p(x)$ has roots of only even multiplicity is equal to $\delta(C, E)$.

There are 8 cases according to the multiplicity of the real roots of $p(x)$ that are shown in figure 2.2.

Due to the previous theorem, we are only interested in the cases 5, 6 and 8, where $p(x)$ has only even multiplicity roots.

According to [16], the quantities Δ_1, Δ_2, T, R define the real roots of $p(x)$. Δ_1, Δ_2, T, R are s -polynomials of degrees 4, 1, 2 and 0 respectively. Moreover, $R = 2\beta^2\alpha^4v_1v_2^2(\alpha^2 - \beta^2)^3 > 0$, since $\alpha > \beta > 0$. For the cases 5, 6 and 8 to be true, the following equation should hold:

$$\Delta_1 = 0 \wedge (T < 0 \vee (T = 0 \wedge \Delta_2 \geq 0))$$

The equation $\Delta_1(s) = 0$ alone covers cases 2, 4, 5, 6, 7, 8, that is for the values of s where $\Delta_1(s) = 0$ any of these cases might be true. However, if we sort the values of s in increasing order, cases 2, 5, 6 and 8 will be true before cases 4 and 7.

Now we just pick³ the smallest root of $\Delta_1(s)$ that makes $\Delta_2(s) \geq 0$ or $T \neq 0$.

³We strongly believe that case 2 never occurs on the roots of $\Delta_1(s)$ and we just have to pick the smallest root of $\Delta_1(s)$. However, we've not come up with a proof yet.

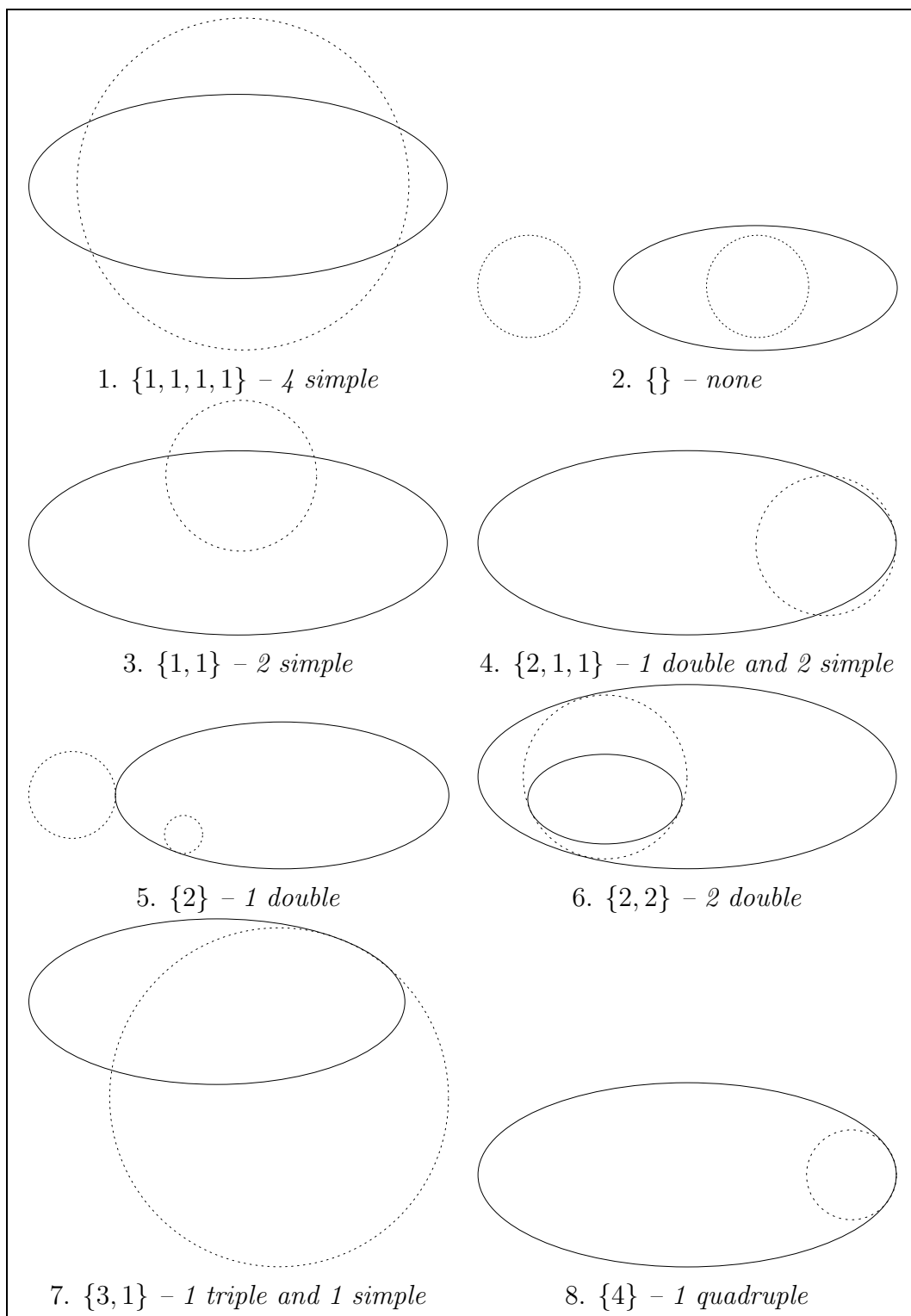


Figure 2.2: Relative position of a circle and an ellipse, according to the real roots of the resultant

2.2.2 The discriminant

We can derive the same conditions using the pencil and geometric arguments ([30, 36]), which are more intuitive.

Let us express a conic as $[x, y, 1]M[x, y, 1]^T$, for an appropriate matrix M . Then E, C correspond to

$$A = \begin{pmatrix} a & b & d \\ b & c & e \\ d & e & f \end{pmatrix}, \quad B(s) = \begin{pmatrix} 1 & 0 & -v_1 \\ 0 & 1 & -v_2 \\ -v_1 & -v_2 & v_1^2 + v_2^2 - s \end{pmatrix}$$

The *pencil* of E and C is $\lambda A + B$, and their characteristic polynomial is

$$\phi(s, \lambda) = |\lambda A + B(s)| = J_2^2 \lambda^3 + c_2(s) \lambda^2 + c_1(s) \lambda + s \quad (2.5)$$

where

$$\begin{aligned} c_2(s) &= J_2 s - T(v_1, v_2) \\ c_1(s) &= J_1 s - E(v_1, v_2) \\ T(v_1, v_2) &= J_2[(v_1 - x_c)^2 + (v_2 - y_c)^2 - J_1] \end{aligned}$$

It is interesting that T stands for a circle centered at the center of E with squared radius equal to J_1 . Note that the coefficients of $\phi(\lambda)$ depend on the invariants of E ; its discriminant $\Delta(s)$ is of degree 4:

$$\begin{aligned} \Delta(s) &= J_2^2(J_1^2 - 4J_2) s^4 + \\ &2J_2(9J_1J_2^2 - J_1^2T + 6J_2T - 2J_1^3J_2 - J_1J_2E) s^3 + \\ &(-18J_2^3E + 4J_1J_2ET - 27J_2^4 + J_1^2T^2 - 18J_1J_2^2T + J_2^2E^2 + 12J_1^2J_2^2E - 12J_2T^2) s^2 + \\ &2(2T^3 - J_1ET^2 - 6J_1J_2^2E^2 + 9J_2^2ET - J_2E^2T) s + \\ &E^2(T^2 + 4J_2^2E) \end{aligned} \quad (2.6)$$

The relative position of a circle and an ellipse falls into one of 9 cases ([36, 18]), shown in fig. 2.3, where we omit the case of the circle coinciding with the ellipse. These cases are related to the multiplicity and signs of the real roots of $\phi(\lambda)$, as in table 2.1, which is based upon [36, thm.8] and [18, sec.4]. Note that $\phi(\lambda)$ has at least one negative root because the product of roots equals $-s < 0$.

To express $\delta(V, E)$, s must be such that the circle and the ellipse are in position 6, 7, or 8 (bold in fig. 2.3). By picking the smallest positive root of $\Delta(s) = 0$, we assure that $\phi(\lambda)$ has at least one root with multiplicity greater than one, since Δ is the discriminant of ϕ . Hence, we get positions 4, 5, 6, 7, 8 or 9; then, we must exclude positions 4, 5 and 9. Assume that the circle centered at V grows until it touches E (and then it might continue to grow until it fully contains E). Since V is outside E , the smallest positive root of $\Delta(s)$ corresponds to $\delta(V, E)$, because position 6 occurs first.

real roots of $\phi(\lambda)$	rel. pos. (fig. 2.3)
$\{-1\}$	2
$\{-1, 2\}$	6
$\{-1, 1, 1\}$	3
$\{-1, -2\}$	4, 5, 7, 8
$\{-1, -1, -1\}$	4, 1
$\{-3\}$	7, 9

Table 2.1: Relative position of circle and ellipse. The multiplicity and sign of the real roots of ϕ is shown in braces, i.e. $\{-1, 2\}$ stands for one negative and one positive double root.

Proposition 2.2.6 *Given an ellipse E and a point V outside E , $\delta(V, E)$ is the square-root of the smallest positive zero of $\Delta(s)$; the latter is a univariate polynomial of degree 4. The degree of the coefficients of $\Delta(s)$ is 6, 8, 10, 12, and 14, in order of decreasing power in s , in v_1, v_2 and the parameters of E . In case (v_1, v_2) is the center of another ellipse E' the degree of the coefficients of $\Delta(s)$ is exactly 22 in the parameters of E, E' .*

Corollary 2.2.7 *Predicate (κ_1) is implemented as follows. Given ellipses E_1, E_2 and point V outside both of them, we can decide which ellipse is closest to V by comparing two algebraic numbers of degree 4. This degree is optimal.*

Optimality follows from the corresponding results on the evolute, as mentioned in the previous chapter. This is a harder predicate than with circles, where it reduces to computing the distance of V to the centre and subtracting the radius.

Alternatively, since we want equation (2.5) to have a multiple root, with respect to λ , the vanishing of the discriminant $\Delta(s)$ corresponds to the solution of the system

$$\begin{cases} \phi(\lambda, s) & = 0 \\ \frac{\partial}{\partial \lambda} \phi(\lambda, s) & = 3J_2^2 \lambda^2 + 2c_2(s)\lambda + c_1(s) = 0 \end{cases} \quad (2.7)$$

However for this predicate we choose not to use this approach, since in [16] all the quantities concerning computations with quartic algebraic numbers are precomputed and clearly faster.

Transitions of the relative position of C and E

Assume that the circle centered at V grows until it touches E and then continues to grow until it fully contains E . Depending on whether V is inside E or not, the initial position is 6 or 4 respectively. The transitions of the different positions are as follows⁴

⁴The bold numbers denote the cases where the circle is tangent to the ellipse. As expected, depending on the relative position of V and the evolute, there are 2, 3 or 4 such cases.

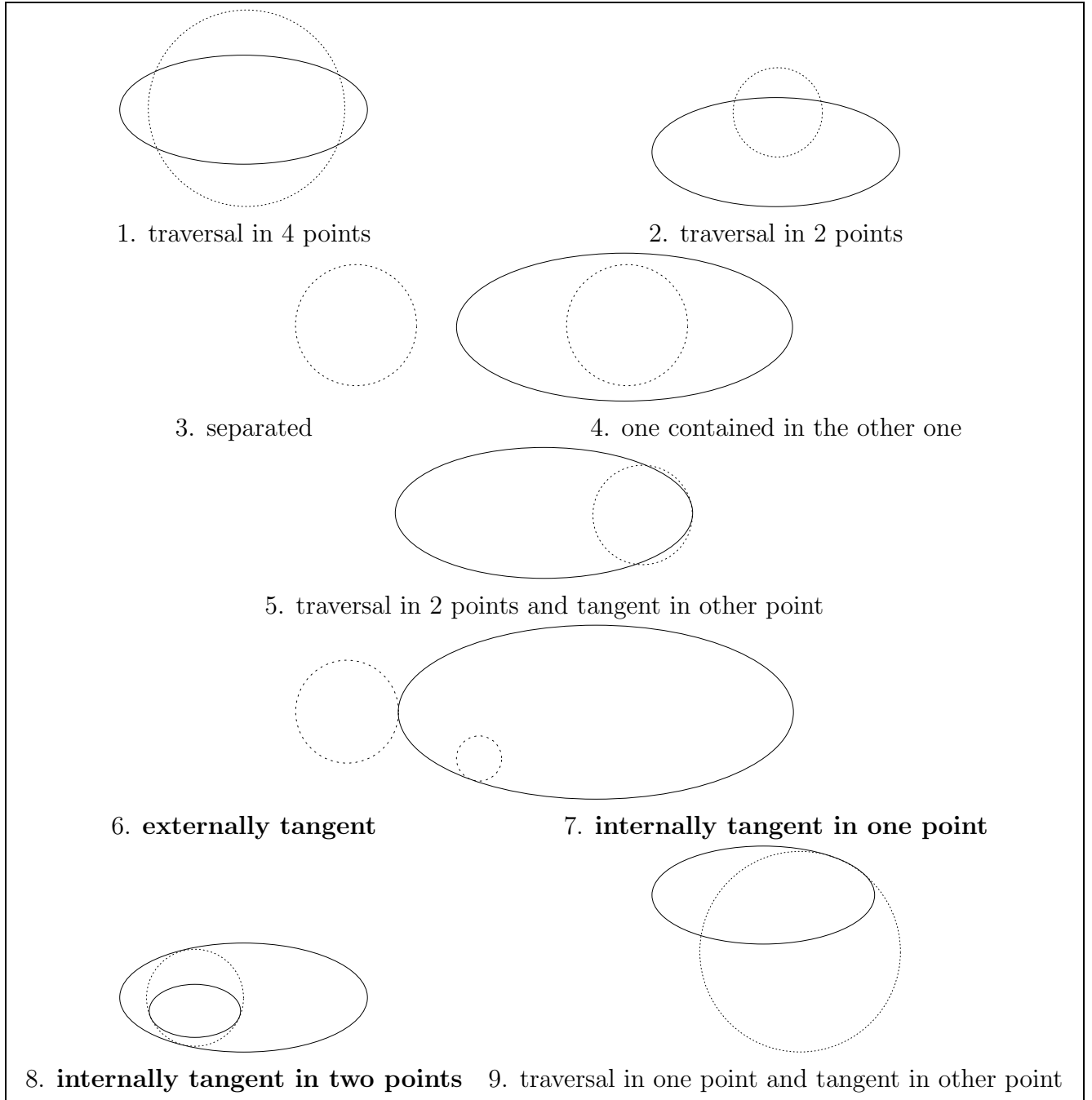


Figure 2.3: Relative position of a circle and an ellipse on the plane

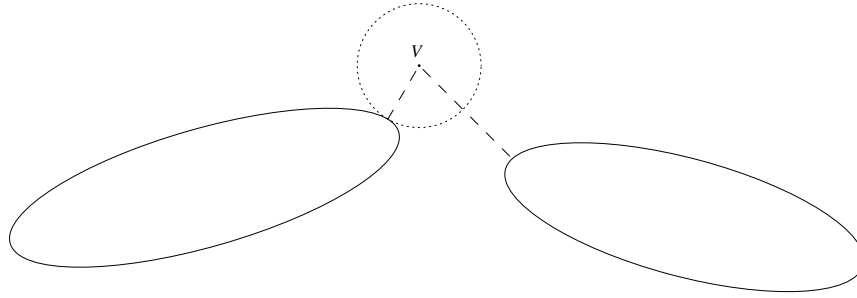


Figure 2.4: Predicate (κ_1)

- *V outside E*

- *V outside* evolute: $3 \rightarrow \mathbf{6} \rightarrow 2 \rightarrow \mathbf{7} \rightarrow 4$

- *V on* evolute: $3 \rightarrow \mathbf{6} \rightarrow 2 \rightarrow \mathbf{9} \rightarrow 2 \rightarrow \mathbf{7} \rightarrow 4$

- *V inside* evolute: $3 \rightarrow \mathbf{6} \rightarrow 2 \rightarrow \mathbf{5} \rightarrow 1 \rightarrow \mathbf{5} \rightarrow 2 \rightarrow \mathbf{7} \rightarrow 4$

- *V inside E*

- *V outside* evolute or *on* cusp of evolute: $4 \rightarrow \mathbf{7} \rightarrow 2 \rightarrow \mathbf{7} \rightarrow 4$

- *V on* evolute: $4 \rightarrow \mathbf{7} \rightarrow 2 \rightarrow \mathbf{9} \rightarrow 2 \rightarrow \mathbf{7} \rightarrow 4$

- *V on* center: $4 \rightarrow \mathbf{8} \rightarrow 1 \rightarrow \mathbf{8} \rightarrow 4$

- *V on* large axis and *in* evolute: $4 \rightarrow \mathbf{8} \rightarrow 1 \rightarrow \mathbf{5} \rightarrow 2 \rightarrow \mathbf{7} \rightarrow 4$

- *V on* small axis and *in* evolute: $4 \rightarrow \mathbf{7} \rightarrow 2 \rightarrow \mathbf{5} \rightarrow 1 \rightarrow \mathbf{8} \rightarrow 4$

- *V inside* evolute: $4 \rightarrow \mathbf{7} \rightarrow 2 \rightarrow \mathbf{5} \rightarrow 1 \rightarrow \mathbf{5} \rightarrow 2 \rightarrow \mathbf{7} \rightarrow 4$

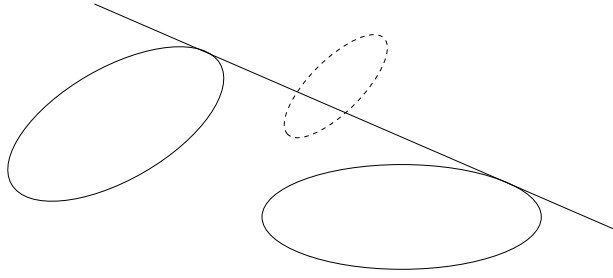


Figure 2.5: The query ellipse has 2 common points with a bitangent

2.3 Bitangent and query ellipse

Given two ellipses, predicate κ_2 , decides on which side of one of their external bitangents, lies a third ellipse (see fig. 2.5). It also decides whether it has common points with the bitangent and, if so, how many. In what follows we assume that the ellipses are separated (i.e. they do not share common points nor one is contained in the other one). To decide the predicate we compute and characterize all the bitangents of the two ellipses, we pick the external bitangent of interest and eventually we compute the relative position of the third ellipse with respect to this bitangent.

Consider a (non-vertical) line $L : y = ux + v$ and an ellipse E . If we substitute y in E we get a quadratic polynomial in x which is the resultant of E, L with respect to y . For L to be tangent to E , the discriminant Λ of the resultant should vanish:

$$\Lambda(u, v) = (e^2 - cf)u^2 + 2(cd - be)uv + (b^2 - ac)v^2 + 2(de - bf)u + 2(bd - ae)v - af + d^2.$$

For a bitangent line, we have $\Lambda_1 = \Lambda_2 = 0$, where Λ_i corresponds to the i -th ellipse. The system has ≤ 4 real roots corresponding to the 4 bitangents.

Lemma 2.3.1 *Let u_1, u_2, u_3, u_4 be the slopes of the four bitangents in increasing order, where $u_i \in \mathbb{R} \cup \{\infty\}$. Let μ be an internal bitangent and ϵ be an external one. Then (u_1, u_2, u_3, u_4) correspond to a cyclic permutation of $(\mu\epsilon\epsilon\mu)$. We call this permutation P -order.*

Proof. Consider, for each bitangent, the corresponding unit vector with the same slope. Given two ellipses, we can easily pick a coordinate system that gives the permutation $(\mu\epsilon\epsilon\mu)$. Now by continuously rotating the coordinate system we get cyclic permutations of $(\mu\epsilon\epsilon\mu)$, for each u_i grows until it reaches $+\infty$ and jumps to $-\infty$. \square

Corollary 2.3.2 *In order to determine the P -order of the bitangents of two ellipses, it suffices to determine the type of exactly two bitangents.*

The previous analysis does not take into account the vertical bitangents which corresponds to roots at infinity of the system $\Lambda_1 = \Lambda_2 = 0$. A vertical bitangent is identified by the degree of the resultant, which becomes < 4 . A vertical bitangent would be $x = v$. If we substitute this value of x in Eq. (1.5), we get a quadratic polynomial in y , whose discriminant is $K(v) = (b^2 - ac)v^2 + 2(eb - dc)v + e^2 - fc$. A vertical bitangent exists iff $K_1(v)$ and $K_2(v)$ have a common root.

Lemma 2.3.3 *The ordering of the roots of $K_1(v)$ and of $K_2(v)$, together with the y -coordinates of the tangency points of the vertical tangents, yields the P -order of all bitangents, both vertical and non-vertical.*

Proof Let $a_1 < a_2$ and $b_1 < b_2$ be the roots of $K_1(v)$ and $K_2(v)$ respectively. Then there exist 13 cases for the total ordering of $\{a_1, a_2, b_1, b_2\}$.

1. $a_1 < a_2 < b_1 < b_2$: The ellipses are separated by a vertical line (fig. 2.6). Let E_1 be the ellipse on the left and E_2 the ellipse on the right. Consider the two vertical tangents of E_1 . We start rotating them clockwise around E_1 , keeping them parallel to each other. Then they become tangent to E_2 in increasing order of their slopes. First, we get one internal bitangent μ , then two external ones ϵ , and finally one internal one. Thus the P -order is $(\mu\epsilon\epsilon\mu)$ and is unique.
2. $b_1 < b_2 < a_1 < a_2$: Symmetric to the previous case.
3. $a_1 < a_2 = b_1 < b_2$: In this case there is vertical bitangent. Thus the first term in the P -order is μ . Let a_i^y, b_i^y be the y -coordinates of the tangency points of a_i and b_i respectively. If $a_2^y > b_1^y$, then (by using the rotation argument as above) the P -order is $((\mu)\epsilon\epsilon\mu)$. If $a_2^y < b_1^y$ then the P -order is $((\mu)\mu\epsilon\epsilon)$.
4. $b_1 < b_2 = a_1 < a_2$: Symmetric to the previous case.
5. $a_1 < b_1 < a_2 < b_2$: If $a_2^y > b_1^y$ then the P -order is $(\epsilon\epsilon\mu\mu)$, otherwise it is $(\mu\mu\epsilon\epsilon)$. The proof is analogous.
6. $b_1 < a_1 < b_2 < a_2$: Symmetric to the previous case.
7. $a_1 < b_1 < a_2 = b_2$: The P -order is $((\epsilon)\epsilon\mu\mu)$ if $a_1^y > b_2^y$, otherwise it is $((\epsilon)\mu\mu\epsilon)$.
8. $b_1 < a_1 < b_2 = a_2$: Symmetric to the previous case.
9. $a_1 < b_1 < b_2 < a_1$: The P -order is $(\epsilon\mu\mu\epsilon)$.
10. $b_1 < a_1 < a_2 < b_2$: Symmetric to the previous case.
11. $a_1 = b_1 < a_2 < b_2$: If $a_1^y > b_2^y$ then the P -order is $((\epsilon)\epsilon\mu\mu)$, otherwise it is $((\epsilon)\mu\mu\epsilon)$.

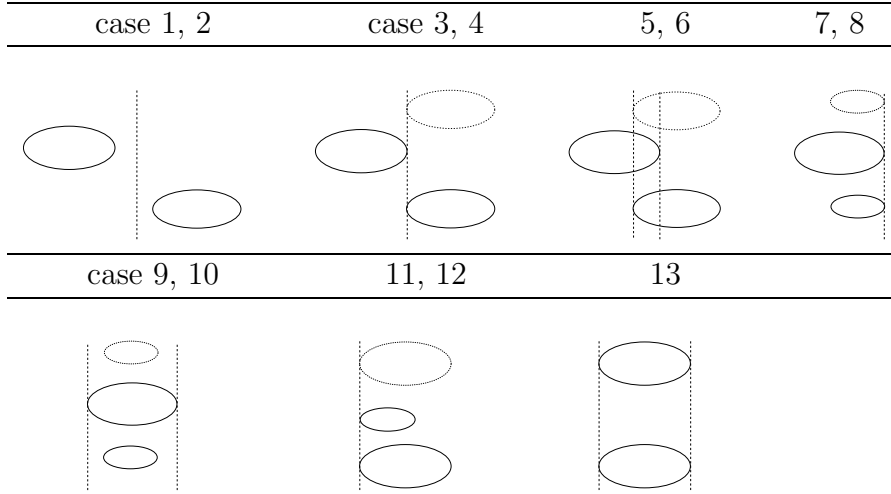


Figure 2.6: Cases of the proof of lemma 2.3.3.

sign of $\Lambda_0(\bar{u}, \bar{v})$	query ellipse
-	on either side of the bitangent
0	tangent to the bitangent
+	crosses the bitangent

Table 2.2: Relative position of query ellipse and bitangent

12. $a_1 = b_1 < b_2 < a_2$: Symmetric to the previous case.
13. $a_1 = b_1 < a_2 = b_2$: In this case there are two external vertical bitangents and two internal ones. Thus the P -order is $((\epsilon)\mu\mu(\epsilon))$.

□

Given an external bitangent $\bar{L} : y = \bar{u}x + \bar{v}$, we can determine the relative position of a third ellipse E_3 wrt this line by computing the discriminant of \bar{L} and E_3 , $\Lambda_3(\bar{u}, \bar{v})$. Λ_3 is negative, zero, or positive iff E_3 has 0, 1 or 2 common points with \bar{L} respectively (table 2.2). In the first 2 cases, the sign of $\bar{L}(x, y)$ evaluated at the center of E_3 specifies on which side of \bar{L} lies E_3 .

Corollary 2.3.4 κ_2 is implemented by, at most, the following operations:

- i) solving a system of 2 bivariate polynomials (computation of bitangents),*
- ii) solving 2 quadratic polynomials and at most 3 comparisons between quadratic algebraic numbers (computation of P -order),*

*iii) evaluating 1 linear bivariate polynomial (choosing appropriate external bitangent),
and*

iv) evaluating 1 quadratic bivariate polynomial (computation of relative position of bitangent and ellipse).

All polynomials have quadratic coefficients in the input parameters.

Chapter 3

External tritangent Voronoi circle

Given 3 ellipses in the form of equation (1.5) we want to find an external tritangent circle, as shown in fig. 3.2. Eventually, we are interested in deciding on the relative position of a fourth ellipse and the circle. An important open question is: what is the maximum number of real tritangent circles to 3 ellipses? Given the discussion in section 1.1, we expect that there are at least $4^3 = 64$ such circles. In the case of 3 disks, the number of tritangent circles is 8 and the corresponding predicate is of algebraic degree 2 [15].

3.1 The discriminant system

Let \sqrt{s} be the radius of the tritangent circle and (v_1, v_2) its center. Using the discriminant, as in equation (2.6), for each of the 3 ellipses, we get

$$\begin{aligned}\Delta_1(v_1, v_2, s) &= 0 \\ \Delta_2(v_1, v_2, s) &= 0 \\ \Delta_3(v_1, v_2, s) &= 0\end{aligned}\tag{3.1}$$

Among the solutions of this system, the external tritangent circle of interest may or may not have the smallest radius; cf. the respective cases in the two figures 3.2.

Proposition 3.1.1 *A solution $(\bar{v}_1, \bar{v}_2, \bar{s})$ of system (3.1) corresponds to an external tritangent circle iff \bar{s} is the smallest positive root of all $\Delta_i(\bar{v}_1, \bar{v}_2, s)$, $i = 1, 2, 3$.*

If s_0^-, s_0^+ are the smallest and largest positive roots of $\Delta_0(\bar{v}_1, \bar{v}_2, s)$, then:

- $\bar{s} \leq s_0^- \iff$ the query ellipse is outside the tritangent circle; tangent iff $\bar{s} = s_0^-$
- $\bar{s} \in (s_0^-, s_0^+) \iff$ the query ellipse crosses the tritangent circle
- $\bar{s} \geq s_0^+ \iff$ the query ellipse is inside the tritangent circle; tangent iff $\bar{s} = s_0^+$

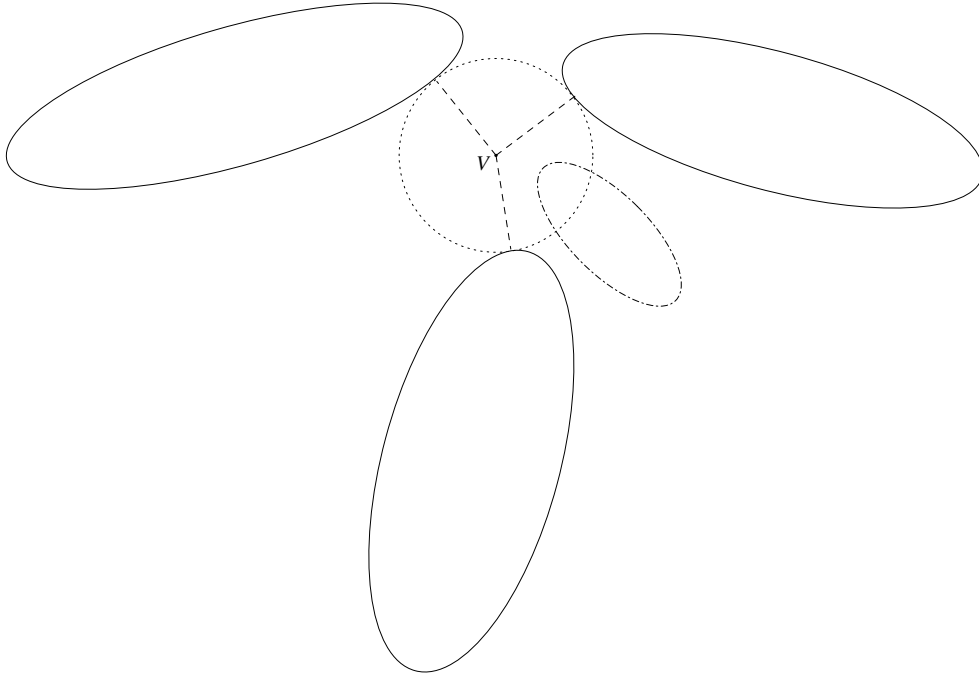


Figure 3.1: A circle externally tangent to three ellipses and a query ellipse that crosses it.

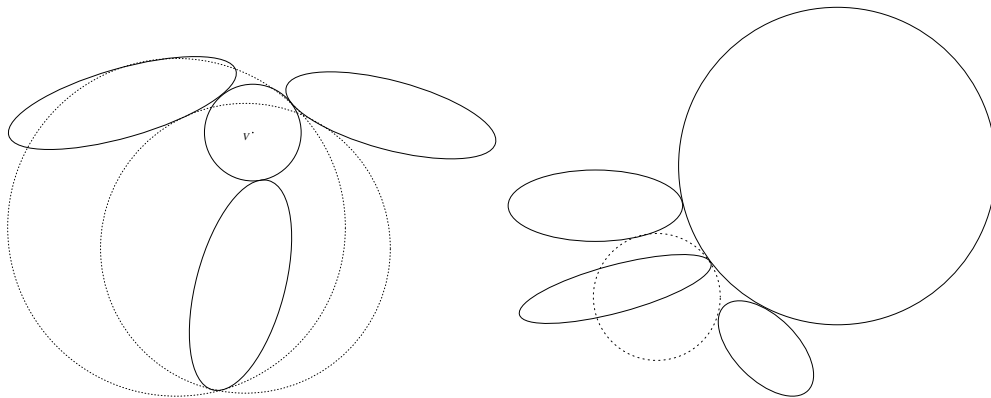


Figure 3.2: Tritangent circles to 3 ellipses; only one is externally tangent

We apply sparse (or toric) elimination theory, using the properties of the resultant and the mixed volume. Given a system of $n + 1$ polynomials f_i in n variables, with coefficients c_{ij} , the *resultant* of these polynomials is a new polynomial $R \in \mathbb{Z}[c_{ij}]$ such that when c_{ij} are specialized, $R = 0 \iff \exists a : \forall i f_i(a) = 0$. If the roots a lie in a projective (resp. toric) variety, then we refer to the projective (resp. toric) resultant. Given n polynomials in n variables, the *mixed volume* of this polynomial system is a function of the support (Newton polytope) of each polynomial. The mixed volume provides an upper bound on the roots of the system in $(\mathbb{C}^*)^n$. For more information see [11].

Each Δ_i is of total degree 8 in v_1, v_2, s and 4 in s ; the coefficient of $s^i, 0 \leq i \leq 4$ is a completely dense polynomial in v_1, v_2 , of degree $8 - 2i$. The mixed volume is 256, which is too high for practical computation. It is known that this bound may not be tight, as it may count complex roots and roots at “infinity”.

Recent advances in matrix formulae for the resultant allow us to compute the resultant of certain systems of 3 bivariate polynomials as a single determinant. One class of such systems are those with identical supports [26]; see also [11]. The corresponding matrix is of hybrid type, i.e., it contains blocks of Sylvester and of Bézout type. The matrix construction has been implemented in Maple by A. Khetan. Our system (3.1) falls in this class, considered with variables v_1, v_2 , after hiding s in the field of coefficients. Its resultant shall be a polynomial in s and equals, generically, the determinant of a 108×108 matrix. We denote this matrix by K . The elements of this matrix are either 0 or polynomials in s of degree 0–10.

We have studied a specific example of three ellipses, in random position as in the left-hand side figure 3.2. The parameters are signed 10-bit integers. The first obstacle is that the assumption of genericity does no longer hold when we specialize the coefficients of system (3.1). In particular, the matrix determinant is the zero polynomial in s . To avoid this degeneracy, it is possible to perturb the original polynomials by adding to each one a random polynomial in v_1, v_2 , of the same support, multiplied by a positive infinitesimal variable t [12]. The corresponding perturbed determinant becomes:

$$d_8(s)t^8 + d_9(s)t^9 + \dots$$

How do we find the exponent of the smallest nonzero term, denoted $d_k(s)t^k$ (here $k = 8$)? It is computed probabilistically by taking random integer values of $t = \tau$ and finding the value of the determinant mod τ^i . When this value vanishes, $k \leq i$; otherwise, $k > i$. In practice, we also specialized s to random values.

Now, it suffices to study $d_8(s)$, which is the most significant coefficient since $t \rightarrow 0^+$. This coefficient is known as a *projection operator* and offers a generalization of the resultant, in the sense that every common root of the original system projects to some root of $d_8(s)$ [12]. In order to understand the complexity of $d_8(s)$, we computed its values for specific

integers $s = \sigma$. Let $D(\sigma, \tau)$ be the determinant for specific values of s, t . We have

$$[D(\sigma, \tau)/\tau^8] \bmod \tau \equiv d_8(\sigma) \bmod \tau.$$

It is possible to use several values of τ , then the Chinese remainder theorem (CRT) in order to interpolate $d_8(\sigma)$. The knowledge of $d_8(\sigma)$ for sufficiently many σ shall yield $d_8(s)$.

Application of CRT is a way to compute different values of $d_8(s)$. Given these values, one could use some method based on interpolation in order to determine the degree of $d_8(s)$ and finally solve it.

It is interesting to see that the above methods failed in practice with our specific example. The involved computations made the involved quantities grow too fast. $d_8(0)$ was found to have more than 10000 digits, so we abandoned this approach.

3.2 The bisector of two ellipses

We may alternatively consider system Δ_1, Δ_2 in v_1, v_2 , with mixed volume = 64. We eliminate s by computing the system's (Sylvester) resultant, and get a completely dense polynomial $B(v_1, v_2)$ as the *bisector* of the two ellipses, of total degree 28.

Given three ellipses E_1, E_2, E_3 , the intersection points of their bisectors define the centre of an Apollonius circle. Let B_{ij} be the bisector of E_i and E_j . Then system $B_{12} = B_{13} = 0$ provides an alternative formulation to the problem. It involves two bivariate polynomials of degree 28. However, our experiments indicate that the corresponding curve (real part) has a much lower degree.

3.3 The cartesian system

In chapter 2 we considered system (2.1) with the additional equation expressing distance and remarked that its resultant is precisely $\Delta(v_1, v_2, s)$. It has degree 8, even for fixed s , which agrees with the degree of the offset curve at distance s [5, 3]. Now, $\Delta(v_1, v_2, s) = 0$ is irreducible, although the offset curve has two components in the real plane. Therefore, $\Delta(v_1, v_2, s)$ is, in a sense, the best we can do.

Let us forget the discriminants and consider system (2.1). It leads to a better upper bound on the number of tritangent circles. Take system

$$\begin{aligned} E_i(x_i, y_i) &= \det[\nabla E_i(x_i, y_i), (x_i, y_i) - V] = 0, \\ \|(x_1, y_1) - V\|^2 &= \|(x_2, y_2) - V\|^2 = \|(x_3, y_3) - V\|^2 \end{aligned}$$

where $i = 1, 2, 3$. Its Bézout bound is 256, as the previous system, but the mixed volume, with respect to unknowns $(x_i, y_i), i = 1, 2, 3, v_1, v_2$, is 192. An alternative formulation is to

consider first the distance equations subsystem. This subsystem is linear with respect to the coordinates of $V(v_1, v_2)$. Solving this linear system for v_1, v_2 and substituting, yields a 6×6 system with Bézout bound 512 and mixed volume 416. These bounds are higher than the best-so-far 256 and 192 respectively.

3.4 Semi-algebraic conditions

In order to reduce the algebraic complexity of the problem, one may consider semi-algebraic conditions. These conditions come handy in selecting the appropriate roots given a (numeric) solution of the system, since we are only interested in the *externally* tangent Apollonius circle. Moreover, they could make a subdivision-based (full-precision) algorithm more efficient. General conditions are presented in [18], however our conditions are based on Descartes' rule ([16]) and are simpler.

Proposition 3.4.1 (Descartes) *The number of sign variations in the coefficients of a univariate polynomial exceeds the number of positive real roots by an even quantity, possibly 0.*

Corollary 3.4.2 *For a polynomial of degree d , the number of sign variations in its coefficient sequence gives precisely the number of positive roots, assuming no root equals zero and there are d real roots.*

Now apply this corollary to the characteristic polynomial $\phi(\lambda)$. Its sign sequence is $(+, \text{sgn}(c_2), \text{sgn}(c_1), +)$, where the constant coefficient is $s > 0$. The sequence of $\phi(-\lambda)$ is $(-, \text{sgn}(c_2), -\text{sgn}(c_1), +)$. For the (parametric) circle to be external to the ellipse, $\phi(\lambda)$ must have a double positive root and one negative root (table 2.1).

Proposition 3.4.3 *The (parametric) circle is externally tangent to the ellipse, iff the coefficients of $\phi(\lambda)$ satisfy $\Delta = 0$ and one of the following: either $c_2 \geq 0$ and $c_1 < 0$, or $c_2 < 0$.*

Proof. [\Leftarrow] $\Delta = 0$ ensures a multiple root, and 2 sign variations ensure a double root. [\Rightarrow] External tangency enforces existence of one negative root and one positive double root. Now, $\Delta = 0$ since there is at least one multiple root, hence all 3 roots of ϕ are real; thus, cor. 3.4.2 applies. The case we wish to describe corresponds to 2 sign changes for $\phi(\lambda)$ and one for $\phi(-\lambda)$. There are two cases for c_2 . If $c_2 \geq 0$, then the sign sequences of $\phi(\lambda), \phi(-\lambda)$ become

$$(+, \geq 0, \text{sgn}(c_1), +) \text{ and } (-, \geq 0, -\text{sgn}(c_1), +),$$

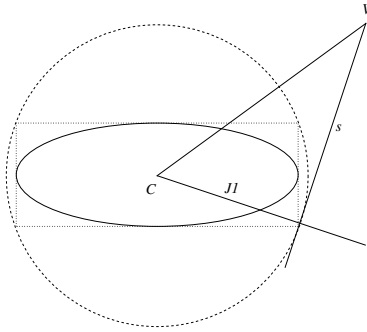


Figure 3.3: The geometric meaning of $c_2 < 0$

so we must have $c_1 < 0$. If $\text{sgn}(c_2) < 0$, then the sequences are

$$(+, -, \text{sgn}(c_1), +) \text{ and } (-, -, -\text{sgn}(c_1), +),$$

so anything goes for $\text{sgn}(c_1)$. □

Now $c_2 < 0 \Leftrightarrow (v_1 - x_c)^2 + (v_2 - y_c)^2 > J_1 + s$ which means that (v_1, v_2) lies outside the circle $T(x, y)$, cf. fig. 3.3. On the other hand, $c_1 < 0 \Leftrightarrow E(v_1, v_2) - J_1 s > 0$. For fixed s , this means V is outside the ellipse $E = J_1 s$, which has the same foci as E but different axes. The complementary condition $c_2 \geq 0$ is equivalent to (v_1, v_2) lying inside or on the circle $T(x, y)$.

3.5 Geometric constraints

In this section we provide geometric constraints that allow us to derive bounds for the center of the tritangent circle as well as its radius.

Lemma 3.5.1 *An external bitangent circle of two smooth convex objects (eg. ellipses), touches each one at a point which lies inside their convex hull.*

Proof. Since each object is smooth, a normal line can be drawn at every point on its boundary. At the points of tangency (A and B resp.), two oriented normal lines (rays), towards the external region, a and b resp., should meet at the center of the external bitangent circle (V). Since the object is convex, the slope of the normal line increases monotonically as we move clockwise around the object. The same holds for the convex hull (CH) of the objects as well. Therefore, two normals on the CH, do not have an intersection point outside CH. So, at least one of A and B should lie inside CH. Let A lie inside the convex hull and B lie not inside it (i.e. on the boundary). If the normal rays intersect, then V lies outside CH, since ray b does not have any common points with CH

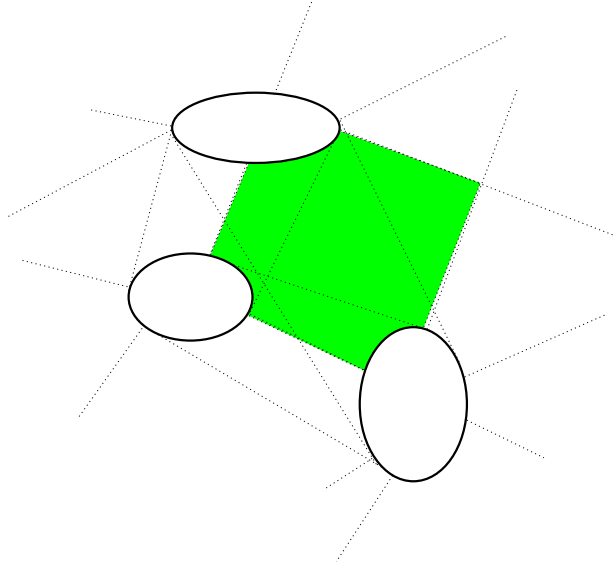


Figure 3.4: The C -region of three ellipses.

(except A). Now, consider the tangent line b' at B , which is perpendicular to b and lies outside CH . Then \overrightarrow{VA} intersects b' , therefore $|\overrightarrow{VB}| < |\overrightarrow{VA}|$. This means that V cannot be the center of the bitangent circle. Thus, both A and B should lie inside CH . \square

Corollary 3.5.2 *Given n disjoint smooth convex objects, their external Apollonius circle (tangent to all of them), if exists, touches each one at a point which lies inside the intersection of the convex hulls of every pair of them. The center of the Apollonius circle lies inside the intersection of the corresponding regions defined by the normals at the external bitangents of every pair of objects (cf. fig 3.4). We call this intersection C -region.*

Corollary 3.5.3 *Given 3 ellipses, where none of them is fully contained in the convex hull of the other two, the C -region is a single bounded region, otherwise it consists of two bounded regions.*

Proposition 3.5.4 *Consider two ellipses E_1, E_2 and a point V outside of both. Let δ_1, δ_2 be the distances between V and each one of the ellipses. Assume $\delta_1 < \delta_2$. Then the center of an external bitangent circle does not lie in the circle $C(V, \frac{\delta_2 - \delta_1}{2})$.*

Proof. Consider a circle centered at V with radius r . Let V' be a point inside this circle. Now it is $\delta_1 - r \leq \delta(V', E_1) \leq \delta_1 + r$ and $\delta_2 - r \leq \delta(V', E_2) \leq \delta_2 + r$. Since $\delta_1 \leq \delta_2$, we get $\delta_1 + r \leq \delta_2 - r$ or equivalently $r \leq \frac{\delta_2 - \delta_1}{2}$. \square

3.6 The optimal discriminant system

It is possible to reduce the mixed volume of the discriminant system, in order to obtain a better upper bound. We set

$$q := v_1^2 + v_2^2 - s \tag{3.2}$$

In this case, the matrix B defined previously, contains only linear terms with respect to v_1, v_2, q . The discriminant of the characteristic polynomial is of total degree 6 in v_1, v_2, q and 4 in q ; the coefficients of $1, q, q^2, q^3, q^4$ are polynomials in v_1, v_2 of degree 6, 5, 4, 2, 1 respectively. The corresponding system has mixed volume 184. Note that solving for v_1, v_2, s requires the use of equation (3.2). The mixed volume of the system of Δ_i with this additional equation, with respect to v_1, v_2, s, q , is still 184.

To get an idea of the quantities involved, we have studied a specific example of three ellipses, in random position as in the left-hand side figure 3.2. The input parameters are signed 10-bit integers. The elements of K are either 0 or polynomials in q of degree 0–10. The computation of the determinant of K is done by interpolation. The determinant of K is a polynomial in q , which we denote by $d(q)$. By substituting different values of q into K we eliminate all indeterminates making the computation of the determinant a trivial task. By making 200 such replacements (in fact, 185 suffice) we obtain 200 pairs of $\langle q, d(q) \rangle$. It turns out that there is a unique interpolating polynomial of degree 184 in q through these values which is exactly the resultant of our example. Hence, in this example the number of complex solutions matches the upper bound given by the mixed volume.

The coefficients of this resultant are, on average, 1385-digit (4603-bit) integers. We've not yet managed to solve this resultant efficiently and exactly. However, as a preliminary approach we have applied the Aberth method (implemented in [8]) to solve the polynomial numerically. This algorithm yielded 8 real roots in less than a second.

3.7 The upper bound

According to [33], there are 184 complex circles in the worst case that are tangent to 3 given conics in the plane. The idea is to consider a manifold (space of complete conics) whose cohomology ring ([20]) has two generators:

$$\begin{aligned} p &:= \text{a conic contains a fixed, but general point} \\ l &:= \text{a conic is tangent to a fixed, but general line} \end{aligned}$$

In this ring, conjunction of conditions is multiplication, and every degree-5 monomial is associated to an integer:

$$\begin{aligned}
p^5 &\mapsto 1 && \text{(there is a unique conic through 5 points)} \\
p^4l &\mapsto 2 && \text{(there are 2 conics through 4 points and tangent to a line)} \\
p^3l^2 &\mapsto 4 \\
p^2l^3 &\mapsto 4 \\
pl^4 &\mapsto 2 \\
l^5 &\mapsto 1
\end{aligned}$$

The condition of tangency to a conic is $2(p+l)$ and a circle contains the two circular points at infinity $(x_1 : x_2 : x_0) = (i : \pm 1 : 0)$. Now it suffices to evaluate the expression $p^2[2(p+l)]^3$:

$$p^2[2(p+l)]^3 = 8[p^5 + 3p^4l + 3p^3l^2 + p^2l^3] = 8 * 23$$

Thus $p^2[2(p+l)]^3 \mapsto 184$ which means that there are at most 184 complex circles tangent to 3 conics in the plane. An open question is how many of these circles can be real.

3.8 Solving the system

Let us now consider a system equivalent to (3.1) in terms of the mixed volume. Each $\Delta_i(v_1, v_2, q)$ is the discriminant of $\phi_i(\lambda, v_1, v_2, q)$. Following the construction of the system in equation (2.7) we get the 6×6 system:

$$\begin{aligned}
\phi_i(\lambda_i, v_1, v_2, q) &= 0 \\
\frac{\partial}{\partial \lambda_i} \phi_i(\lambda_i, v_1, v_2, q) &= 0
\end{aligned} \tag{3.3}$$

where $i = 1, 2, 3$. This system has substantially smaller coefficients than (3.1) and contains three equations of total degree 3 and three equations of total degree 2. We solved the system numerically with `PHCpack` [34], which exploits sparseness, and implements homotopy continuation methods to compute numerically approximations to all isolated solutions of a system of n polynomial equations in n unknowns.

Additionally we chose the scaling parameters of `PHCpack`, in order to avoid ill-conditioned systems without losing any real solutions. Timings are reported in the next chapter.

Chapter 4

Implementation

We have implemented predicates κ_1 and κ_2 using SYNAPS [14], a C++ algebraic library for symbolic-numeric computations. SYNAPS provides efficient implementation based on the algorithms of [16] for computation of algebraic numbers of small degree, solution of bivariate polynomial systems and computation of the sign of a bivariate polynomial evaluated over two real algebraic numbers. We used extended integer arithmetic GMP¹. We also performed tests with the filtered number type `Lazy_exact_nt`², but we do not present the results here because they were sufficiently slower, since the bit size of the various quantities is very big and the filter failed almost always. This gives us an indication that geometric filters must be used in order to further accelerate the computations.

As for the numerical solution of system corresponding to the third predicate we used `PHCpack` based on its black-box mode, except a scaling in the coefficients.

For each test we randomly generated 1000 instances (point and 2 ellipses, or 3 ellipses), with the coefficients uniformly distributed between 0 and 2^B , $B \in \{10, 30, 100, 300\}$. The tests ran on a P4 2.6GHz-CPU with 1GB of RAM, running Debian Linux with a 2.6.10 kernel. Table 4.1 summarizes average timings; for κ_1 and κ_2 runtimes grow sub-quadratically in B . Note that for predicate κ_2 half of the time spent for the solution of the bivariate system and the other half spent for the computation of the the relative position of the third ellipse. Compared to [19], κ_2 is more powerful. While they compute (χ_1) in 6ms total time, κ_2 in the same time provides more information.

For predicate κ_3 , the timings refer to the solution of the corresponding system. They are almost independent of the bit size, since we scaled the coefficients in order to fit machine accuracy. Currently, we are not able to estimate neither numerical accuracy nor the actual number of real solutions. In [3] the system proposed for κ_3 has suboptimal mixed volume, hence large complexity. Though they also calculate approximate numerical solutions, their matrix methods seem slower than ours.

¹www.swox.com/gmp/

²www.cgal.org

B	predicate κ_1 [ms]	predicate κ_2 [ms]	predicate κ_3 [sec]
10	0.45	6.15	23.85
30	0.94	16.46	34.52
100	3.68	73.21	38.36
300	17.3	396.82	38.44

Table 4.1: Average timings of our implementation

4.1 Source code

```

/*
  National University of Athens
  Department of Informatics and Telecommunications

  Implementation of the following algebraic predicates
  using the SYNAPS library:

  0) Find a rational point inside an dellipse

  1) Given a point and two ellipses , find out which ellipse is closest
  to the point.

  2) Given three ellipses and an external bitangent of the fist two,
  decide the relative position of the third one with respect to the
  bitangent.

  (c) 2004–2005 George M. Tzoumas

  THIS SOURCE CODE IS PROVIDED WITHOUT ANY WARRANTY.
  YOU ARE FREE TO USE THIS CODE, BUT PROPER CREDITS MUST BE GIVEN.
*/

#include "synaps/upol.H"
#include "synaps/linalg.H"
#include "synaps/resultant.H"
#include "synaps/arithm/root_of.H"
// #include "synaps/arithm/ZZ.H"
#include "synaps/util/Rand.H"
#include <vector>

#include<stdlib.h>
#include<time.h>

using namespace SYNAPS;

typedef ZZ RT;
typedef UPolDse<RT> Poly;
typedef ALGEBRAIC::MPoly_2_2<RT> BPoly;

```

```

typedef ALGEBRAIC::root_of<RT> Rootof;
typedef std::pair< Rootof, Rootof > RootofPair;

Rand<RT> *Rnd = NULL;

using std::endl;
using std::cout;
using std::cin;
using std::cerr;

void Randomize(int bits)
{
    RT b = (RT(1) << bits) - 1;
    if (Rnd != NULL) delete Rnd;
    Rnd = new Rand<RT>(-b, b);
}

class TTimer {
private:
    clock_t t0;
    clock_t tot_clocks;
    int nops;
    char *name;

public:
    TTimer(char *aname): name(aname), nops(0), tot_clocks(0) { }

    void Start() {
        t0 = clock();
    }

    void Stop() {
        tot_clocks += clock() - t0;
        nops++;
    }

    void Stats() {
        double sec = tot_clocks*1.0/CLOCKS_PER_SEC;
        cerr << name << ":_ " << sec << "_sec,_" << nops << "_ops,_" <<
            sec*1000/nops << "_ms/op" << endl;
    }
};

TTimer t_pc("precompute"), t_bext("find_ext"), t_sol("solve"),
    t_bspec("find_spec"), t_relp("relpos");

class TPoint {
public:
    RT x, y;

    TPoint() { };
};

```

```

TPoint(RT x_, RT y_): x(x_), y(y_) { };
void PickRandom() {
    x = (*Rnd)();
    y = (*Rnd)();
}
};

class TEllipse {
private:
    RT E, J1, J2, R;
    Poly p;
    int recompute;

    void compute_poly_coeff();
    void compute_vars(RT& v1, RT& v2);

public:
    RT a, b, c, d, e, f;
    TPoint *V;

    TEllipse() { p.resize(5); }
    TEllipse(RT a_, RT b_, RT c_, RT d_, RT e_, RT f_):
        a(a_), b(b_), c(c_), d(d_), e(e_), f(f_), V(NULL) { p.resize(5); };
    void SetPoint(TPoint *V_);
    void PickRandom();
    RT EvalV() { return E; }
    Rootof DistV();

};

class TBitangent {
private:
    Poly d1, d2; // degenerate;
    BPoly linep1, linep2, qlinep; /* line bpolys */
    RT cache_J2, cache_X, cache_Y;

    void compute_disc_coeff(TEllipse& e, BPoly& p0) {
        RT c[6];
        c[0] = e.d*e.d - e.a*e.f;
        c[1] = 2*(e.b*e.d - e.a*e.e);
        c[2] = 2*(e.d*e.e - e.b*e.f);
        c[3] = 2*(e.c*e.d - e.b*e.e);
        c[4] = e.b*e.b - e.a*e.c;
        c[5] = e.e*e.e - e.c*e.f;
        BPoly p(c);
        p0 = p;
    }

    void compute_disc_coeff(TEllipse& e, Poly& p) { // degenerate

```



```

    p.resize(3);
    p[0] = e.e*e.e - e.c*e.f;
    p[1] = 2*(e.e*e.b - e.c*e.d);
    p[2] = e.b*e.b - e.a*e.c;
}

void compute_line_poly(BPoly& disc , BPoly& linep) {
    RT c[6];
    c[0] = disc[1];
    c[1] = 2*disc[4];
    c[2] = disc[3];
    c[3] = 0; c[4] = 0; c[5] = 0;
    BPoly p(c);
    linep = p;
}

void compute_qline_poly(BPoly& linep) {
    RT c[6];
    c[0] = cache_Y;
    c[1] = cache_J2;
    c[2] = -cache_X;
    c[3] = 0; c[4] = 0; c[5] = 0;
    BPoly p(c);
    linep = p;
}

int eval_line_cent(BPoly& linep , Rootof& u, Rootof& v) {
int s = sign(sign_at(linep , u, v));
// cerr << "sign = " << s << ":" << u << ":" << v << endl;
return s;
}

int eval_line_cent(Poly& disc , Rootof& v) { // degenerate
    Poly p(2, AsSize());
    p[0] = disc[1];
    p[1] = 2*disc[2];
    return sign(sign_at(p, v));
}

int eval_line_qell(Rootof& u, Rootof& v) {
    return sign(sign_at(qlinep , u, v));
}

int eval_line_ell(Rootof& v) {
    Poly p(2, AsSize());
    p[0] = cache_X;
    p[1] = cache_J2;
    return sign(sign_at(p, v));
}

```

```

int is_external(BPoly& linep1 , BPoly& linep2 ,
                Rootof& u, Rootof& v, int& signell) {
    int s = eval_line_cent(linep1 , u, v);
    if (s == eval_line_cent(linep2 , u, v)) {
        signell = s;
        return 1;
    } else return 0;
}

#define SEARCH_FOR_ONE(f) \
    int qsign = eval_line_qell(sol[f].first , sol[f].second); \
    if (sign1 * qsign == signprod) { \
        u = sol[f].first; v = sol[f].second; \
        found = true; \
    } \

#define PICK_ONE(f) \
    u = sol[f].first; v = sol[f].second; \
    found = true

public:
    int signprod;
    Rootof u, v;
    int degenerate; /* when zero means that we are not looking for a
                    degenerate root;
                    when non-zero it is equal to the # of degenerate
                    roots that exist (we shall choose one of them) */
    BPoly p1,p2; /* discriminants */
    bool found;
    int nsols; /* number of solutions */

    TBitangent(int asignprod): signprod(asignprod) , found(false) {
    }

    // return the number of bitangents (2-4), used for random data generation
    int BCount(TEllipse& e1 , TEllipse& e2) {
        compute_disc_coeff(e1 , p1);
        compute_disc_coeff(e2 , p2);
        Seq<RootofPair> sol = Solve(p1 , p2 , ALG_STURM<RT>());
        return sol.size();
    }

    void Compute(TEllipse& e1 , TEllipse& e2 , TEllipse& e3) {

        t_pc.Start();
    //    t_bext.Start();
        t_bspec.Start();
        t_sol.Start();

```

```

cache_J2 = e3.b*e3.b - e3.a*e3.c;
cache_X = e3.e*e3.b - e3.c*e3.d;
cache_Y = e3.b*e3.d - e3.a*e3.e;
compute_disc_coeff(e1, p1);
compute_disc_coeff(e2, p2);
compute_line_poly(p1, linep1);
compute_line_poly(p2, linep2);
compute_qline_poly(qlinep);
t_pc.Stop();

degenerate = 0;
found = false;
int sign1 = 0, sign2 = 0;
Seq<RootofPair> sol = Solve(p1, p2, ALG_STURM<RT>());
t_sol.Stop();
nsols = sol.size();

/* classify external/internal bitangents; there are 4 cases:
   0. i-i-e-e
   1. i-e-e-i
   2. e-i-i-e
   3. e-e-i-i */
int bcase1 = is_external(linep1, linep2,
                        sol[0].first, sol[0].second, sign1);
/*   int bcase2 = is_external(linep1, linep2,
                        sol[1].first, sol[1].second, sign2); */
//   t_bext.Stop();

if (bcase1) { // high bit
    // 2 or 3
    // eval center on candidate bitangent to pick appropriate roots
    SEARCHFOR_ONE(0);
    if (!found) {
        int bcase2 = is_external(linep1, linep2,
                                sol[1].first, sol[1].second, sign2);
        if (bcase2 == 0) { // 2
            if (nsols == 4) {
                PICK_ONE(3);
            }
        } else { // 3
            PICK_ONE(1);
        }
    }
} else {
    // 0 or 1
    SEARCHFOR_ONE(1);
    if (!found) {
        int bcase2 = is_external(linep1, linep2,
                                sol[1].first, sol[1].second, sign2);
        if (bcase2 == 0) { // 0

```

```

SEARCHFOR_ONE(2);
  if (!found && nsols == 4) {
    PICK_ONE(3);
  }
} else if (nsols >= 3) {
  PICK_ONE(2);
}
}
}
t_bspec.Stop();

if (!found) degenerate = 4 - nsols;
if (degenerate > 0) { // vertical bitangent
  compute_disc_coeff(e1, d1);
  compute_disc_coeff(e2, d2);
  Seq<Rootof> rs = Solve(d1, ALG_STURM<RT>());
  Seq<Rootof> qs = Solve(d2, ALG_STURM<RT>());
  int s00 = compare(rs[0], qs[0]);
  if (s00 == 0) {
    if (degenerate == 1) {
      v = rs[0];
      found = true;
    } else {
      int qsign = eval_line_ell(rs[0]);
      cerr << "qsign = " << qsign << endl;
      int sign1 = eval_line_cent(d1, rs[0]);
      if (sign1 * qsign == signprod &&
          eval_line_cent(d2, qs[0]) == sign1) {
        v = rs[0];
        found = true;
      } else {
        v = rs[1];
        found = true;
      }
    }
  }
} else if (s00 == -1) {
  if (compare(rs[1], qs[0]) > 0 && compare(rs[1], qs[1]) == 0) {
    v = rs[1];
    found = true;
  }
} else {
  if (compare(rs[0], qs[1]) > 0 && compare(rs[1], qs[1]) == 0) {
    v = rs[1];
    found = true; // merge with above
  }
}
};
// assert(found);
};

int RelPos(TEllipse& e) {

```

```

    if (degenerate == 0) {
        RT c[6];
        c[0] = e.a*e.e - e.b*e.d;
        c[1] = e.a*e.c - e.b*e.b;
        c[2] = e.b*e.e - e.d*e.c;
        c[3] = 0; c[4] = 0; c[5] = 0;
        BPoly p(c);
        return sign(sign_at(p,u,v));
    } else {
        Poly p(2, AsSize());
        p[0] = e.b*e.e - e.d*e.c;
        p[1] = e.b*e.b - e.a*e.c;
        return sign(sign_at(p, v));
    }
};

```

```

void TEllipse::SetPoint(TPoint *V_)
{
    V = V_;
    compute_vars(V->x, V->y);
    compute_poly_coeff();
}

```

```

void TEllipse::compute_vars(RT& v1, RT& v2)
{
    RT t20 = 2 * v1;
    RT t19 = 2 * v2;
    J1 = a + c;
    J2 = a * c - b * b; /* recomputed for timekeeping purposes */
    RT t3 = v1 * v1;
    RT t8 = v2 * v2;
    E = a * t3 + f + c * t8 + e * t19 + (d + b * v2) * t20;
    R = J2 * (t3 + t8) + (a * e - b * d) * t19 + (c * d - b * e) * t20 +
        J1 * f - d * d - e * e;
}

```

```

void TEllipse::compute_poly_coeff()
{
    RT t2 = J2 * J2;
    RT t6 = t2 * E;
    RT t4 = J2 * E * J1;
    RT t1 = E * E;
    RT t3 = t6;
    RT t5 = R * R;
    p[0] = t1 * (4 * t3 + t5);
    RT t14 = t1 * t2;
    p[1] = 2 * (-6 * t14 - t5 * E) * J1 +
        2 * (2 * t5 + 9 * t3 - t1 * J2) * R;
}

```

```

RT t27 = J1 * J1;
RT t34 = t2 * J1;
p[2] = t14 - 27 * t2 * t2 + (-18 * t6 - 12 * t5) * J2 +
      (4 * t4 - 18 * t34) * R + (12 * t3 + t5) * t27;
p[3] = -2 * J2 * (-9 * t34 + t4 - 6 * J2 * R +
      (2 * J2 * J1 + R) * t27);
p[4] = -t2 * (-t27 + 4 * J2);
}

void TEllipse::PickRandom()
{
    RT J3;

    while (1) {
        a = (*Rnd)(); b = (*Rnd)(); c = (*Rnd)();
        d = (*Rnd)(); e = (*Rnd)(); f = (*Rnd)();
        J2 = a * c - b * b;
        if (J2 <= 0) {
            /* cerr << "bad: j2 <= 0" << endl; */
            continue;
        }
        J3 = 2*b*e*d - e*e*a - d*d*c - b*b*f + a*c*f;
        if (a*J3 > 0) {
            /* cerr << "bad: a*j3 > 0" << endl; */
            continue;
        }
        if ((a == c) || (b == 0)) {
            /* cerr << "bad: circle" << endl; */
            continue;
        }
        break;
    }
    p.resize(5);
    /* cerr << "(" << a << ") * X^2 + (" << 2*b << ") * X * Y + ("
       << c << ") * Y^2 + ("
       << 2*d << ") * X + (" << 2*e << ") * Y + (" << f << ")" << endl; */
}

Rootof TEllipse::DistV()
{
    int i;
    Seq<Rootof> sol = Solve(p, ALG_STURM<RT>());
    for (i = 0; i < sol.size(); i++) if (sign(sol[i]) >= 0) break;
    return sol[i];
}

int main(int argc, char **argv)
{
    if (argc == 3) {

```

```

    int bits = atoi(argv[1]);
    int num = atoi(argv[2]);
    int i;
    Randomize(bits);
    TEllipse e1, e2, e3;
    TPoint v;
    TBitangent btan(1);

    std::cout << num << endl;
    for (i = 0; i < num; i++) {
/*
        v.PickRandom();
        while (1) {
            e1.PickRandom();
            e2.PickRandom();
            e1.SetPoint(ℰv);
            e2.SetPoint(ℰv);
            if (e1.EvalV() >= 0 && e2.EvalV() >= 0) break;
        }*/
        while (1) {
            e1.PickRandom();
            e2.PickRandom();
            e3.PickRandom();
            if (btan.BCount(e1, e2) == 4) break;
        }
        std::cout << e1.a << " " << e1.b << " " << e1.c << " "
            << e1.d << " " << e1.e << " " << e1.f << endl;
        std::cout << e2.a << " " << e2.b << " " << e2.c << " "
            << e2.d << " " << e2.e << " " << e2.f << endl;
        std::cout << e3.a << " " << e3.b << " " << e3.c << " "
            << e3.d << " " << e3.e << " " << e3.f << endl;
/*
        std::cout << v.x << " " << v.y << endl;*/
    }
} else {
    int i, num;
    RT a, b, c, d, e, f;
    TEllipse e1, e2, e3;
    TPoint v;
    TBitangent btan(1);

    std::cin >> num;
    for (i = 0; i < num; i++) {
        std::cin >> e1.a >> e1.b >> e1.c >> e1.d >> e1.e >> e1.f;
        std::cin >> e2.a >> e2.b >> e2.c >> e2.d >> e2.e >> e2.f;
        std::cin >> e3.a >> e3.b >> e3.c >> e3.d >> e3.e >> e3.f;
/*
        std::cin >> v.x >> v.y;
        e1.SetPoint(ℰv); e2.SetPoint(ℰv);
        //std::cout << compare(e1.DistV(), e2.DistV()) << endl;
        compare(e1.DistV(), e2.DistV());*/
        btan.Compute(e1, e2, e3);
/*
        std::cerr << "pos = " << btan.RelPos(e3)
            << " degen = " << btan.degenerate << " u = "; */
    }
}

```

```
        t_relp.Start();
        btan.RelPos(e3);
        t_relp.Stop();
    }
}
t_pc.Stats();
//    t_bext.Stats();
t_sol.Stats();
t_bspec.Stats();
t_relp.Stats();
return 0;
}
```


Chapter 5

Conclusion

The computation of the Voronoi diagram of curved objects may require complex algebraic predicates. In this thesis we have investigated the corresponding predicates in the case where the curved objects are ellipses.

We presented a simple way to compute a point inside an ellipse. We proposed an efficient representation of the distance between a point and an ellipse, that has lead to an efficient implementation of the corresponding predicate. Following a similar approach, we have been able to decide the relative position of a third ellipse with respect to a bitangent of two other ellipses. The way this predicate is computed allows us to completely characterize all bitangents.

Upon these results we studied the problem of the Apollonius circle and bounded its complexity using tools such as the mixed volume and sparse (or toric) resultants. We were the first to obtain a tight upper bound on the number of tritangent circles.

Although we solved the corresponding system numerically, we provide semi-algebraic and geometric constraints in order to prune cases in some subdivision-based algorithm. In a future work, such an algorithm could lead to an efficient and exact real time solution.

Bibliography

- [1] H. Alt and O. Schwarzkopf. The Voronoi diagram of curved objects. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 89–97, 1995.
- [2] P. Angelier. *Algorithmique des graphes de visibilité*. PhD thesis, Université Paris VII, 2002.
- [3] F. Anton. *Voronoi diagrams of semi-algebraic sets*. PhD thesis, The University of British Columbia, January 2004.
- [4] F. Anton, J.-D. Boissonnat, D. Mioc, and M. Yvinec. An exact predicate for the optimal construction of the additively weighted Voronoi diagram. In *Europ. Workshop Comput. Geom.*, 2002.
- [5] F. Anton, I.Z. Emiris, B. Mourrain, and M. Teillaud. The degree of the offset to planar algebraic curves. Manuscript.
- [6] F. Anton and D. Mioc. A certified predicate for maintaining the topology of the voronoi diagram for conics. In *Proc. X Encuentros de Geometría Computacional*, pages 180–186, Sevilla, Spain, 2003.
- [7] F. Aurenhammer. Voronoi diagrams: A survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, September 1991.
- [8] D.A. Bini and G. Fiorentino. Design, analysis, and implementation of a multiprecision polynomial rootfinder. *Numerical Algorithms*, 23:127–173, 2000.
- [9] I. Boada, N. Coll, and J.A. Sellarès. Multiresolution approximations of generalized voronoi diagrams. In *Proc. Intern. Conf. Comp. Science*, 2004.
- [10] J.-D. Boissonnat and M. Karavelas. On the combinatorial complexity of Euclidean Voronoi cells and convex hulls of d-dimensional spheres. In *Proc. SODA*, pages 305–312, 2003.

- [11] D. Cox, J. Little, and D. O’Shea. *Using Algebraic Geometry*. Number 185 in Graduate Texts in Mathematics. Springer-Verlag, New York, 1998.
- [12] C. D’Andrea and I.Z. Emiris. Computing sparse projection operators. In E.L. Green, S. Hosten, R.C. Laubenbacher, and V.A. Powers, editors, *Symbolic Computation: Solving Equations in Algebra, Geometry, and Engineering*, volume 286 of *Contemporary Mathematics*, pages 121–139, Providence, Rhode Island, 2001. AMS.
- [13] C. Delage. Diagrammes de Möbius en dimension 2, 2003. Master’s thesis, INRIA Sophia-Antipolis.
- [14] G. Dos Reis, B. Mourrain, R. Rouillier, and P. Trébuchet. An environment for symbolic and numeric computation. In *Proc. Int. Conf. Math. Software*, World Scientific, pages 239–249, 2002.
- [15] I.Z. Emiris and M.I. Karavelas. The predicates of the Apollonius diagram: algorithmic analysis and implementation. *Comp. Geom.: Theory & Appl., Spec. Issue*, 2005. To appear.
- [16] I.Z. Emiris and E.P. Tsigaridas. Computing with real algebraic numbers of small degree. In *Proc. ESA, LNCS*, pages 652–663. Springer Verlag, 2004.
- [17] I.Z. Emiris and G.M. Tzoumas. Algebraic study of the apollonius circle of three ellipses. In *Proc. European Workshop Computat. Geometry*, pages 147–150, Eindhoven, Holland, March 2005.
- [18] F. Etayo, L. Gonzalez-Vega, and N. del Rio. A new approach for characterising the relative position of two ellipses depending on one or several parameters. 2004. Submitted.
- [19] L. Habert and M. Piocchola, November 2004. Private communication.
- [20] J. Harris. *Algebraic Geometry (A First Course)*. Springer-Verlag, Berlin, Germany, 1992.
- [21] F. Hartmann and R. Jantzen. Apollonius’ ellipse and evolute revisited, September 2003. Manuscript.
<http://www42.homepage.villanova.edu/frederick.hartmann/resume.html>
- [22] M.I. Karavelas. A robust and efficient implementation for the segment voronoi diagram. In *Proc. Int. Symp. Voronoi Diagrams*, pages 51–62, 2004.
- [23] M.I. Karavelas and I.Z. Emiris. Root comparison techniques applied to the planar additively weighted Voronoi diagram. In *Proc. SODA*, pages 320–329, January 2003.

- [24] M.I. Karavelas and M. Yvinec. Dynamic additively weighted Voronoi diagrams in 2D. In *Proc. ESA 2002*, pages 586–598, September 2002.
- [25] M.I. Karavelas and M. Yvinec. Voronoi diagram of convex objects in the plane. In *Proc. ESA*, pages 337–348, 2003.
- [26] A. Khetan. The resultant of an unmixed bivariate system. *J. Symb. Comput.*, 36:425–442, 2003.
- [27] D.-S. Kim, D. Kim, and K. Sugihara. Voronoi diagram of a circle set from Voronoi diagram of a point set: II. Geometry. *CAGD*, 18:563–585, 2001.
- [28] R. Klein, K. Mehlhorn, and S. Meiser. Randomised incremental construction of abstract Voronoi diagrams. *Comput. Geom. Theory & Appl.*, 3(3):157–184, 1993.
- [29] T. Koufogiorgos. *Lecture notes on analytical geometry*. University of Ioannina, 1996.
- [30] H. Levy. *Projective and Related Geometries*. McMillan Co., 1967.
- [31] R. Lewis and S. Bridget. Polynomial equations arising from apollonius problems in biochemistry and pharmacology. In *Proceedings of Applications of Computer Algebra*, 2001.
- [32] M. McAllister, D. Kirkpatrick, and J. Snoeyink. A compact piecewise-linear Voronoi diagram for convex sites in the plane. *Discrete Comput. Geom.*, 15:73–105, 1996.
- [33] F. Sottile. Personal communication. 2004.
- [34] J. Verschelde. PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. 25(2):251–276, 1999.
- [35] W. Wang and R. Krasauskas. Interference analysis of conics and quadrics. In *Algebraic geometry and geometric modeling*, volume 334 of *Contemporary Mathematics*, pages 25–36. AMS, 2003.
- [36] W. Wang, J. Wang, and M. Kim. An algebraic condition for the separation of two ellipsoids. *Comp. Aided Geom. Design*, 18:531–539, 2001.