

# Het Oplossen van Stelsels Veeltermvergelijkingen

Simon Telen

Departement Computerwetenschappen  
KU Leuven

Marc Van Barel

Departement Computerwetenschappen  
KU Leuven

3 juni 2016

## Samenvatting

We stellen een nieuwe methode voor om 0-dimensionale bivariate veeltermstelsels op te lossen gebruik makend van numerieke lineaire algebratools. Er wordt een graadsuitbreiding toegepast op een 2-parameter eigenwaardenprobleem die resulteert in een resultant equivalent met die van Sylvester. Alle oplossingen (reële en complexe) worden berekend met de juiste multipliciteit. De methode slaagt erin om zonder Newton-Raphson verfijningen of uitbreidingen naar hogere precisie nauwkeurige resultaten te bekomen, ook voor meervoudige nulpunten.

## 1 Inleiding

Multivariate veeltermssystemen duiken op in vele ingenieursdisciplines. Ze vinden hun toepassingen typisch in problemen die een intrinsiek veeltermkarakter hebben of problemen die niet nauwkeurig genoeg beschreven worden door lineaire modellen. Een aantal voorbeelden van toepassingsgebieden zijn chemische ingenieurstechnieken, filterontwerp, computer aided design, robotica, . . . . Vrij recent zijn er verschillende algoritmische methodes ontwikkeld om veeltermssystemen op te lossen. Groebner basismethodes [12, 13, 5], homotopiemethodes [2, 16] en methodes gebaseerd op resultanten [9, 7, 3, 11, 4] zijn daarvan de belangrijkste klassen. In dit artikel ligt de focus op het tweedimensionale geval. Noteer de ring van bivariate veeltermen met  $\mathcal{P}^2$ . Het probleem wordt als volgt geformuleerd.

**Probleem 1.** *Vind alle paren  $(x, y) \in \mathbb{C}^2$  die voldoen aan*

$$\begin{cases} p(x, y) = 0 \\ q(x, y) = 0 \end{cases} \quad (1)$$

met  $p, q \in \mathcal{P}^2$ .

We zullen er steeds vanuit gaan dat (1) een eindig aantal geïsoleerde oplossingen heeft (of ook, de beschouwde systemen zijn 0-dimensionaal). Er wordt

in dit artikel een methode gebaseerd op numerieke lineaire algebra voorgesteld om (1) op te lossen. We streven ernaar om niet enkel alle (reële en complexe) oplossingen van (1) te vinden maar ook om hun multipliciteit in rekening te brengen. De methode combineert de voorstelling van (1) als een twee-parameter eigenwaardeprobleem [10] met een graadsuitbreiding (die ook eigen is aan Macaulay resultant-gebaseerde methodes [7, 3]) om een lineair vierkant pencil te verkrijgen in slechts 1 veranderlijke. Er kan aangetoond worden dat de eigenwaarden van dit pencil dezelfde zijn als die van de Sylvesterresultant [13, 12, 5, 1]. Zo kunnen de  $x$ -waarden (of  $y$ -waarden) gevonden worden als de eigenwaarden van een veralgemeend eigenwaardeprobleem (GEP). In dit artikel illustreren we de methode aan de hand van een eenvoudig voorbeeld en geven we de belangrijkste resultaten. Daarna worden er enkele numerieke experimenten toegelicht en wordt de snelheid en nauwkeurigheid ten opzichte van bestaande oplossingsmethodes geïllustreerd. Voor bewijzen, uitbreidingen naar andere basissen en meer details verwijzen we de lezer naar [14]. In [14] wordt ook een uitbreiding naar meerdere variabelen voorgesteld in bijlage.

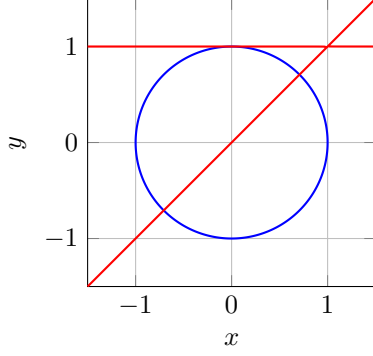
## 2 Een Eenvoudig Voorbeeld

Beschouw het probleem

$$\begin{cases} p(x, y) = x^2 + y^2 - 1 = 0 \\ q(x, y) = -x + y + xy - y^2 = 0 \end{cases} \quad (2)$$

met als oplossingen  $(\sqrt{2}/2, \sqrt{2}/2)$ ,  $(-\sqrt{2}/2, -\sqrt{2}/2)$ ,  $(0, 1)$  en  $(0, 1)$ <sup>1</sup>. De reële nulverzamelingen van  $p$  en  $q$  zijn afgebeeld in Figuur 1. Een eerste stap van de oplossingsmethode is de overgang van (2) naar de

<sup>1</sup>Hier is  $(0, 1)$  een oplossing met multipliciteit 2. Voor meer uitleg over de multipliciteitstructuur van de oplossingen van multivariate veeltermproblemen verwijzen we naar [12].



Figuur 1: Reële nulverzamelings van de veeltermen  $p$  (—) en  $q$  (—) uit (2).

matrixformulering

$$\underbrace{\begin{pmatrix} -1 & 0 & 1 & | & 0 & 0 & | & 1 \\ 0 & -1 & 0 & | & 1 & 1 & | & -1 \end{pmatrix}}_{\Phi} \mathbf{v}(x, y) = \begin{pmatrix} p(x, y) \\ q(x, y) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (3)$$

met  $\mathbf{v}(x, y) = (1 \ x \ x^2 | y \ xy | y^2)^\top$  de vector van monomialen van graad  $\leq 2$  en  $\Phi$  de *coëfficiëntenmatrix*. Het is duidelijk dat elke oplossing  $(x^*, y^*)$  een richting genereert in de rechtse nulruimte van  $\Phi$ . Inderdaad, als  $p(x^*, y^*) = q(x^*, y^*) = 0$  dan is  $\Phi \mathbf{v}(x^*, y^*) = \mathbf{0}$ . Echter, niet elke vector in  $\text{null}(\Phi)$  geeft informatie over een oplossing van het stelsel. Beschouw bijvoorbeeld de vector  $\mathbf{w} = (1 \ 0 \ 1 | 0 \ 0 | 0)^\top$ . Het is duidelijk dat  $\Phi \mathbf{w} = \mathbf{0}$ , maar uit  $\mathbf{w}$  kunnen we geen informatie halen over een oplossing van (2). Er bestaat immers geen complex koppel  $(x^*, y^*)$  zodanig dat de vector  $\mathbf{w}$  een veelvoud is van  $\mathbf{v}(x^*, y^*)$ . Met andere woorden,  $\mathbf{w}$  heeft geen *Vandermondestructuur*.

**Definitie 2.1.** Een vector  $\mathbf{w}$  heeft een (bivariate) Vandermondestructuur in de monomiaalbasis gegeven door de elementen van  $\mathbf{v}(x, y)$  als er een koppel  $(x^*, y^*) \in \mathbb{C}^2$  bestaat zodanig dat  $\mathbf{w} = C \mathbf{v}(x^*, y^*)$  met  $C \in \mathbb{C}_0$ .

Dit leidt tot de volgende stap, waarin we aan (3) twee nieuwe blokrijen toevoegen die de Vandermonde structuur opleggen aan de nulvectoren van  $\Phi$ :

$$\underbrace{\begin{pmatrix} -1 & 0 & 1 & | & 0 & 0 & | & 1 \\ 0 & -1 & 0 & | & 1 & 1 & | & -1 \\ -x & 1 & & & & & & \\ & -x & 1 & & & & & \\ & & & -x & 1 & & & \\ -y & & & 1 & & & & \\ & & & -y & & & 1 & \end{pmatrix}}_{L(x, y)} \begin{pmatrix} 1 \\ x \\ x^2 \\ y \\ xy \\ y^2 \end{pmatrix} = \begin{pmatrix} p(x, y) \\ q(x, y) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (4)$$

Merk op dat in (4) de kolommen van  $L(x, y)$  verdeeld zijn in blokken die overeenkomen met monomialen van toenemende graad in  $y$ . De rijen zijn verdeeld in blokrijen als volgt. De bovenste twee rijen worden gegeven door de coëfficiëntenmatrix  $\Phi$ . Het volgende blok bestaat uit drie rijen die de variabele  $x$  bevatten. We gebruiken de notatie  $\mathcal{B}_x - x\mathcal{C}_x$  voor deze deelmatrix. Voor de laatste twee rijen gebruiken we de analoge notatie  $\mathcal{B}_y - y\mathcal{C}_y$ . Elke vector in de nulruimte van  $L(x^*, y^*)$  (verschillend van  $\mathbf{0}$ ),  $(x^*, y^*) \in \mathbb{C}^2$  heeft een Vandermonde structuur omwille van de twee onderste blokrijen. Elk zo'n vector komt overeen met een oplossing van (1). Immers,  $L(x^*, y^*)C\mathbf{v}(x^*, y^*) = \mathbf{0}$ ,  $C \in \mathbb{C}_0$  impliceert dat  $p(x^*, y^*) = q(x^*, y^*) = 0$ .

**Theorema 2.1.** De oplossingen van (2) in  $\mathbb{C}^2$  zijn de koppels  $(x^*, y^*)$  waarvoor  $L(x^*, y^*)$  niet van volle kolomrang is.

De onderste twee blokrijen moeten opgevat worden als een 'basisdefinitie' van de klassieke monomiaalbasis. Er zijn verschillende mogelijkheden om zo een basisdefinitie op te stellen. Er is gekozen om  $x$  en  $y$  enkel lineair te laten voorkomen in het resulterende pencil en om zoveel mogelijk het gebruik van  $y$  te vermijden. We noteren

$$\begin{cases} p(x, y) = 0 \\ q(x, y) = 0 \end{cases} \xrightarrow{C} L(x, y)$$

waar de  $C$  staat voor de constructie van het lineaire pencil  $L(x, y)$ . Om de veranderlijken te scheiden staan er in  $L(x, y)$  nog niet genoeg vergelijkingen in  $x$ . De bovenste twee blokrijen zijn immers niet van volle kolomrang, en dit voor elke mogelijke waarde van  $x$ . Een volgende stap is de graadsuitbreiding. Hier maken we gebruik van de volgende equivalentie

$$\begin{cases} p(x, y) = 0 \\ q(x, y) = 0 \end{cases} \Leftrightarrow \begin{cases} p(x, y) = 0 \\ q(x, y) = 0 \\ yp(x, y) = 0 \\ yq(x, y) = 0 \end{cases}, \forall (x, y) \in \mathbb{C}^2. \quad (5)$$

De veeltermen  $yp(x, y)$  en  $yq(x, y)$  geven aanleiding tot de coëfficiëntenmatrix

$$\Psi = \begin{pmatrix} 0 & 0 & 0 & 0 & | & -1 & 0 & 1 & | & 0 & 0 & | & 1 \\ 0 & 0 & 0 & 0 & | & 0 & -1 & 0 & | & 1 & 1 & | & -1 \end{pmatrix}$$

in de monomiaalbasis

$$(1 \ x \ x^2 \ x^3 | y \ xy \ x^2y | y^2 \ xy^2 | y^3)^\top.$$

In het uitgebreide pencil  $\hat{L}(x, y)$  wordt de eerste blokrij gevormd door de coëfficiëntenmatrix  $\hat{\Phi}$  van  $p$  en  $q$

in de uitgebreide monomiaalbasis. De tweede blokrij is  $\Psi$  en de volgende twee blokrijen vormen de basis-definitie, opnieuw opgedeeld in een  $x$ - en een  $y$ -deel. Het resultaat is

$$\hat{L}(x, y) = \left( \begin{array}{cccc|cccc|ccc} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & -1 \\ \hline -x & 1 & & & & & & & & \\ & -x & 1 & & & & & & & \\ & & -x & 1 & & & & & & \\ \hline & & & & -x & 1 & & & & \\ & & & & & -x & 1 & & & \\ & & & & & & -x & 1 & & \\ \hline -y & & & & 1 & & & & & \\ & & & & -y & & & & & \\ & & & & & & 1 & & & \\ & & & & & & -y & & & 1 \end{array} \right).$$

We gebruiken de notatie  $L(x, y) \xrightarrow{E} \hat{L}(x, y)$  waar  $E$  staat voor de uitbreiding (of ‘extensie’) van het pencil. Merk op dat de monomiaal  $x^3$  niet voorkomt in de vergelijkingen van het uitgebreide stelsel (5). Inderdaad, we hebben de graad uitgebreid door te vermenigvuldigen met  $y$ . De hoogste graad in  $x$  is nog steeds 2. Daarom kunnen we de vierde kolom en de zevende rij uit  $\hat{L}(x, y)$  schrappen. Het resultaat noemen we  $\hat{L}_r(x, y)$ , we vinden

$$\hat{L}_r(x, y) = \left( \begin{array}{cccc|cccc|ccc} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & -1 & \\ \hline -x & 1 & & & & & & & & \\ & -x & 1 & & & & & & & \\ & & & -x & 1 & & & & & \\ & & & & -x & 1 & & & & \\ & & & & & -x & 1 & & & \\ \hline -y & & & 1 & & & & & & \\ & & & -y & & & 1 & & & \\ & & & & & & -y & & & 1 \end{array} \right)$$

en we noteren  $\hat{L}(x, y) \xrightarrow{R} \hat{L}_r(x, y)$ , waar  $R$  staat voor de reductie door het verwijderen van de gepaste rijen en kolommen.

**Theorema 2.2.** *De oplossingen van (2) in  $\mathbb{C}^2$  zijn de koppels  $(x^*, y^*)$  waarvoor  $\hat{L}_r(x^*, y^*)$  niet van volle kolomrang is.*

Noteer de eerste drie blokrijen van  $\hat{L}_r(x, y)$  met  $\hat{\Pi}_{x,r}(x)$  (in dit deel van het pencil komt de variabele  $y$  niet voor). Een nodige voorwaarde opdat  $\hat{L}_r(x, y)$  niet van volle kolomrang is, is dat  $\hat{\Pi}_{x,r}(x)$  niet van volle kolomrang is. Het probleem ‘zoek de waarden

van  $x$  waarvoor  $\hat{\Pi}_{x,r}(x)$  niet van volle kolomrang is’ is een vierkant veralgemeend eigenwaardeprobleem. Immers,  $\hat{\Pi}_{x,r}(x^*) \in \mathbb{C}^{9 \times 9}, \forall x^* \in \mathbb{C}$ .

**Theorema 2.3.** *Beschouw het probleem (1). Noteer de graad van  $p$  in  $y$  als  $\delta_p^y$  en die van  $q$  als  $\delta_q^y$ . Construeer het pencil  $\hat{L}_r(x, y)$  waarbij voor de graadsuitbreiding van  $L(x, y)$  alle vergelijkingen  $\{y^i p(x, y) = 0\}_{1 \leq i \leq \delta_p^y - 1}$  en  $\{y^i q(x, y) = 0\}_{1 \leq i \leq \delta_q^y - 1}$  worden toegevoegd aan het stelsel (1). Dan is het resulterende gereduceerde  $x$ -pencil  $\hat{\Pi}_{x,r}(x)$  vierkant en er geldt*

$$\det \hat{\Pi}_{x,r}(x) = \gamma \text{res}^{p,q}(x)$$

met  $\gamma \in \{-1, 1\}$  en  $\text{res}^{p,q}(x)$  de resultant van Sylvester geassocieerd met (1).

Theorema 2.3 impliceert dat de  $x$ -coördinaten van de oplossingen van het probleem (1) eigenwaarden zijn van  $\hat{\Pi}_{x,r}(x)$ , geconstrueerd zoals voor dit eenvoudige voorbeeld. Meer nog, de multipliciteit van een eindige eigenwaarde  $x^*$  van  $\hat{\Pi}_{x,r}(x)$  komt overeen met de som van de multipliciteiten van alle oplossingen van (1) van de vorm  $(x^*, y)$  (waarbij ook rekening gehouden dient te worden met oplossingen van de vorm  $(x^*, \infty)$ ). De numerieke waarden voor  $x$  zijn voor ons voorbeeldprobleem (gebruik makend van Matlab)

$$\begin{aligned} & -8.896452425993\text{e-}09 + 7.205292475198\text{e-}10\text{i} \\ & 8.896453574757\text{e-}09 - 7.205292872444\text{e-}10\text{i} \\ & -7.071067811865\text{e-}01 + 0.000000000000\text{e+}00\text{i} \\ & 7.071067811865\text{e-}01 + 1.357635865057\text{e-}17\text{i} \end{aligned}$$

waarnaar we zullen verwijzen als  $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3$  en  $\tilde{x}_4$  respectievelijk.

Om de bijhorende  $y$ -waarden te vinden zijn er verschillende benaderingen mogelijk. In dit artikel illustreren we een methode die alle mogelijke  $y$ -waarden berekent en dan een koppeling vindt tussen de set  $x$ - en  $y$ -waarden. De methode slaagt er goed in om de juiste informatie wat betreft de multipliciteiten te achterhalen. De koppeling is gebaseerd op een groepering van alle gevonden  $x$ - en  $y$ -waarden.

We willen de numerieke benaderingen van eenzelfde  $x$ - of  $y$ -waarde in dezelfde groep onderbrengen. Dit gebeurt op basis van de veronderstelling dat  $x$ -waarden in dezelfde groep een klein residu<sup>2</sup> hebben

<sup>2</sup>Het residu is gedefiniëerd met een gemengd absoluut en relatief criterium:

$$r(x^*, y^*) = \frac{|p(x^*, y^*)|}{|p(|x^*|, |y^*|) + 1} + \frac{|q(x^*, y^*)|}{|q(|x^*|, |y^*|) + 1}$$

met  $|p|(x, y) \triangleq \sum_{i=0}^{\delta} \sum_{j=0}^{\delta-i} |p_{ij}| x^j y^i$  en  $|q|(x, y) \triangleq \sum_{i=0}^{\delta} \sum_{j=0}^{\delta-i} |q_{ij}| x^j y^i$ .

gekoppeld met dezelfde set  $y$ -waarden. Voor  $y$  vinden we de numerieke waarden

```
1.000000000000e+00 + 0.000000000000e+00i
9.999999999999e-01 + 0.000000000000e+00i
-7.071067811865e-01 + 0.000000000000e+00i
7.071067811865e-01 + 0.000000000000e+00i
```

waarnaar we zullen verwijzen als  $\tilde{y}_1, \tilde{y}_2, \tilde{y}_3$  en  $\tilde{y}_4$  respectievelijk. De residumatrix voor dit probleem is

$$R(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) = \begin{matrix} & \tilde{x}_1 & \tilde{x}_2 & \tilde{x}_3 & \tilde{x}_4 \\ \tilde{y}_1 & \tilde{0} & \tilde{0} & \times & \times \\ \tilde{y}_2 & \tilde{0} & \tilde{0} & \times & \times \\ \tilde{y}_3 & \times & \times & \tilde{0} & \times \\ \tilde{y}_4 & \times & \times & \times & \tilde{0} \end{matrix}.$$

Daarbij stelt  $R(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}})_{ij}$  het residu voor van het koppelpel  $(x_j, y_i)$ . Het symbool  $\tilde{0}$  stelt een positief getal voor dat kleiner is dan een bepaalde (kleine) drempelwaarde  $\epsilon$  (in dit geval geeft  $\epsilon = 10^{-15}$  al de gewenste opdeling). Een ‘ $\times$ ’ staat voor een getal  $> \epsilon$ . Uit de residumatrix besluiten we dat  $\tilde{x}_1$  en  $\tilde{x}_2$  tot eenzelfde groep behoren (ze zorgen voor kleine residu’s met dezelfde set  $y$ -waarden). Hetzelfde geldt voor  $\tilde{y}_1$  en  $\tilde{y}_2$ . Alle andere  $x$ - en  $y$ -waarden worden apart gegroepeerd. Uit de residumatrix kunnen we afleiden dat er een koppeling moet zijn tussen de groep  $\{\tilde{x}_1, \tilde{x}_2\}$  en de groep  $\{\tilde{y}_1, \tilde{y}_2\}$  met multipliciteit 2. De andere koppelingen zijn  $(\tilde{x}_3, \tilde{y}_3)$  en  $(\tilde{x}_4, \tilde{y}_4)$ . Merk op dat de enkelvoudige oplossingen gevonden zijn tot op machinenauwkeurigheid. De tweevoudige oplossing is ‘opgesplitst’ in twee oplossingen die op een afstand van orde  $10^{-8}$  van de originele oplossing verwijderd liggen. Nemen we echter het gemiddelde van beide groepen  $\{\tilde{x}_1, \tilde{x}_2\}$  en  $\{\tilde{y}_1, \tilde{y}_2\}$  dan vinden we het punt  $(0, 1)$  terug tot op machineprecisie.

### 3 Numerieke Experimenten

In deze paragraaf bespreken we eerst kort de resultaten van de vergelijking van onze methode met een aantal bestaande oplossingsmethodes. Daarna worden er enkele interessante numerieke voorbeelden gegeven.

#### 3.1 Een vergelijking

De voorgestelde methode is getest op een set problemen en vergeleken met andere solvers. Een tool die uiterst geschikt is om snel een idee te geven van de prestatie van een bepaalde solver in vergelijking

met andere solvers is een *performantieprofiel* [6]. Beschouw een set  $S$  van kandidaten (solvers) en een set  $P$  van problemen. De performantiecurve voor een solver  $s \in S$  is gegeven door

$$\rho_s(\tau) = \frac{|\{p \in P \mid t_{p,s} \leq 2^\tau (\min_{s \in S} t_{p,s})\}|}{|P|} \quad (6)$$

met  $|\cdot|$  de cardinaliteit van een verzameling en  $t_{p,s}$  de tijd die een solver  $s$  nodig had om het probleem  $p$  ‘succesvol’ op te lossen. Met ‘succesvol’ wordt hier bedoeld dat er evenveel oplossingen zijn gevonden als het aantal oplossingen in de referentieset<sup>3</sup> en dat aan het volgende voldaan is. Er moet een bijectieve afbeelding  $b : \mathcal{S}_{\text{ref}} \rightarrow \tilde{\mathcal{S}}$  bestaan zodat voor elke oplossing  $\mathbf{s} \in \mathcal{S}_{\text{ref}}$  geldt dat

$$\|\mathbf{s} - \tilde{\mathbf{s}}\|_2 \leq 10^{-2}(1 + \|\mathbf{s}\|_2),$$

met  $\tilde{\mathbf{s}} \triangleq b(\mathbf{s})$ . Deze test is geïmplementeerd met behulp van `bipartite_matching` van de `gaimc` Matlab toolbox voor graafalgoritmen [8]. Het performantieprofiel voor de vergelijking van onze methode met PHClab [16, 15], Bertini [2] en PNLA [7, 3] is te zien in Figuur 2 (links). Er is voor elke methode gebruik gemaakt van de standaardinstellingen, zonder variabele precisie of verfijningsopties voor de oplossingen<sup>4</sup>. Voor een eerste vergelijking is een set van 60 lage graadsproblemen met uitdagende meervoudige oplossingen gebruikt. Bemerkt dat onze methode alle problemen succesvol oplost, rekening houdend met de multipliciteiten en dat voor meer dan 90% van de problemen op de snelste manier doet. De homotopiemethodes en PNLA lossen minder dan 80% van de problemen succesvol op. Een tweede vergelijking is gebeurd op basis van een set random problemen van graad 1 tot en met 40. Met een ‘random’ probleem van graad  $\delta$  bedoelen we een probleem waarbij  $p$  en  $q$  normaalverdeelde coëfficiënten met gemiddelde 0 en standaardafwijking 1 hebben bij alle monomialen van graad  $\leq \delta$ . Alle oplossingen van zo een generiek stelsel zijn enkelvoudig en er zijn er  $\delta^2$  volgens de stelling van Bézout. Het criterium voor succes is in dit geval dat er meer dan 99% van alle  $\delta^2$  oplossingen moet gevonden zijn met een residu  $< 10^{-6}$ . Het performantieprofiel is rechts op Figuur 2 weergegeven. De testen voor PNLA zijn niet volledig voltooid omdat ze te veel tijd vergen. Voor het probleem van graad 25 waren er ongeveer 25 uren nodig. Wat opvalt is

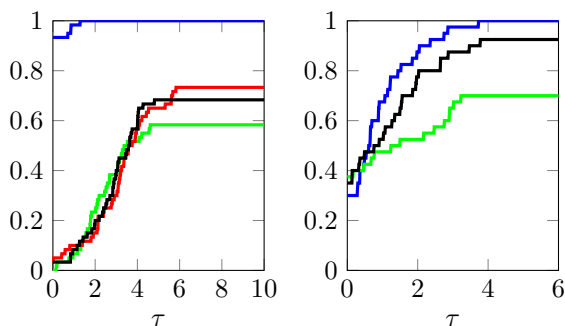
<sup>3</sup>De referentieset bestaat uit de oplossingen gevonden door Bertini in variabele precisie, een zeer betrouwbare solver. Omwille van de variabele precisie is deze oplossingsmethode wel beduidend trager dan de andere (met standaardinstellingen).

<sup>4</sup>Voor Bertini is `MPTYPE: 0` gebruikt, omdat er default `MPTYPE: 1` (adaptive precision) wordt gebruikt. Voor PNLA gebruiken we de `sparf` functie.

$\Re(x)$	$\Im(x)$	$\Re(y)$	$\Im(y)$
2	$4.592 \cdot 10^{-15}$	2	$6.480 \cdot 10^{-15}$
2	$4.592 \cdot 10^{-15}$	2	$6.480 \cdot 10^{-15}$
2	$4.592 \cdot 10^{-15}$	2	$6.480 \cdot 10^{-15}$
$1.571 \cdot 10^0$	$1.207 \cdot 10^{-15}$	$-1.429 \cdot 10^{-1}$	$-1.523 \cdot 10^{-16}$
1	$-5.990 \cdot 10^{-16}$	1	$9.605 \cdot 10^{-16}$
1	$-5.990 \cdot 10^{-16}$	1	$9.605 \cdot 10^{-16}$
1	$-5.990 \cdot 10^{-16}$	1	$9.605 \cdot 10^{-16}$
1	$-5.990 \cdot 10^{-16}$	1	$9.605 \cdot 10^{-16}$
1	$-5.990 \cdot 10^{-16}$	1	$9.605 \cdot 10^{-16}$

Tabel 1: Numerieke oplossingen van (7).

dat voor hogere graadsproblemen onze methode het niet haalt van Bertini of PHClab wat de snelheid betreft. Er is echter wel telkens 100% van alle oplossingen gevonden met een residu  $< 10^{-6}$  terwijl alle berekeningen gebeuren in dubbele precisie en er geen Newton-Raphson verfijning gebruikt is.



Figuur 2: Links: Performance profiel voor de vergelijking van onze methode (—), PHClab (—), Bertini (—) en PNLA (—) voor de set van 60 testproblemen. Rechts: analoog voor de random problemen van graad 1 tot en met 40.

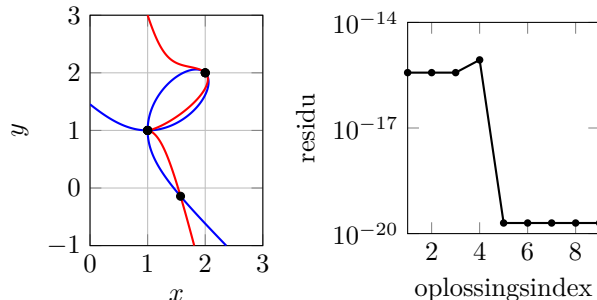
### 3.2 Enkele voorbeelden

- Beschouw het systeem gegeven door

$$\begin{cases} p(x, y) = -4 + 5x - 3x^2 + x^3 + 5y - 2xy - 3y^2 \\ \quad + y^3 = 0 \\ q(x, y) = -4 + x - 2x^2 + 2x^3 + 9y + 2xy - 4x^2y \\ \quad - 8y^2 + 3xy^2 + y^3 = 0 \end{cases} \quad (7)$$

dat enkel reële oplossingen heeft, waaronder een drievoudige in het punt  $(2, 2)$  en een vijfvoudige in  $(1, 1)$ . De reële nulverzamelingen van  $p$  en  $q$  zijn weergegeven in Figuur 3. De numerieke oplossingen gevonden door onze solver zijn gegeven in Tabel 1. Bemerk dat zelfs de vijfvoudige oplossing tot op machineprecisie is teruggevonden.

- Ter illustratie zijn in Figuur 4 twee voorbeelden gegeven van reële nulverzamelingen en de gevonden numerieke reële oplossingen. Voor het eerste probleem (bovenaan op Figuur 4) is  $p$  van

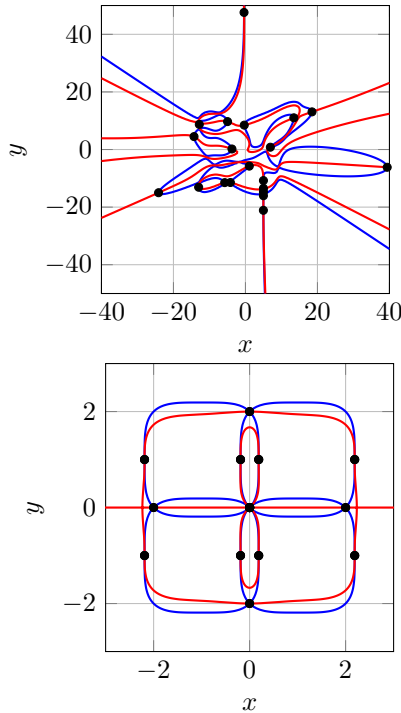


Figuur 3: Links: Reële nulverzamelingen van  $p$  (—) en  $q$  (—) uit (7) en het reële deel van de gevonden numerieke oplossingen ( $\bullet$ ). Rechts: residu voor alle 9 numerieke oplossingen.

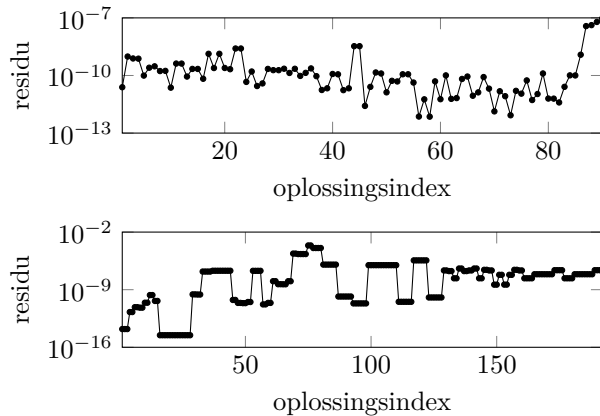
graad 10 en  $q$  van graad 9. Er zijn 90 oplossingen die allemaal worden gevonden met een klein residu. Voor het tweede probleem heeft  $p$  graad 16 en  $q$  graad 15. Alle oplossingen zijn meervoudig en er zijn er in totaal 192 (meervoudigheden meegeteld). Het punt  $(0, 0)$  is bijvoorbeeld een 16-voudig nulpunt. Resultaten zijn weergegeven in Figuur 5. De residu's liggen hoger voor het tweede probleem ten gevolge van de hogere graad en de meervoudigheden. Voor beide problemen voldoet de numerieke oplossingsverzameling aan het succes criterium dat beschreven werd voor het opstellen van de performantieprofielen.

## 4 Besluit

De solver die in dit artikel wordt voorgesteld gebruikt numerieke lineaire algebra tools om de geïsoleerde oplossingen van bivariate stelsels veeltermvergelijkingen te vinden. Er wordt op een intuïtieve manier een algoritme opgesteld om een lineair pencil te construeren waarvan de eigenwaarden de  $x$ - of  $y$ -coördinaten zijn van de oplossingen. Inderdaad, we hebben aangetoond dat de determinant van dit intuïtief bekomen pencil sterk verbonden is met de Sylvesterresultant. Die link maakt het mogelijk voor onze solver om ook informatie over de multipliciteit van de oplossingen te geven. De coëfficiënten van de gegeven veeltermen komen ongemaneuleerd voor in het veralgemeend eigenwaardeprobleem, wat de nauwkeurigheid ten goede komt. Uit de numerieke experimenten blijkt dat de solver er voor lagere graden ( $\leq 20$ ) in slaagt om snel nauwkeurige oplossingen te bekomen met de juiste multipliciteiten. Voor generieke problemen met een graad tot zeker 40 worden alle oplossingen gevonden met een kleine residu en binnen een aanvaardbare tijd. Het residu groeit echter met



Figuur 4: Reële nulverzamelingen van twee voorbeeldproblemen. De zwarte punten geven de reële numerieke oplossingen aan.



Figuur 5: Bovenaan: residu voor alle 90 oplossingen van het probleem bovenaan op Figuur 4. Onderaan: residu voor alle 192 oplossingen van het probleem onderaan op Figuur 4.

de graad en homotopiegebaseerde methodes blijken sneller voor graden  $> 15$ . Deze methodes slagen er echter niet in om alle oplossingen terug te vinden voor generieke problemen van graad  $> 35$  (bij gebruik van de standaardinstellingen en dubbele precisie).

## Referenties

- [1] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*. Springer, 2006.
- [2] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. *Numerically solving polynomial systems with Bertini*, volume 25 of *Software, Environments and Tools*. SIAM, 2013.
- [3] K. Batselier. *A Numerical Linear Algebra Framework for Solving Problems with Multivariate Polynomials*. KU Leuven - Faculty of Engineering Science, 2013. PhD thesis, promotor: Bart De Moor.
- [4] L. Busé, H. Khalil, and B. Mourrain. *Resultant-Based Methods for Plane Curves Intersection Problems*. Springer-Verlag Berlin Heidelberg, 2016.
- [5] D. Cox, J. Little, and D. O’Shea. *Ideals, Varieties and Algorithms*. Springer, 2007.
- [6] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91:201–213, 2002.
- [7] P. Dreesen. *Back to the Roots*. KU Leuven - Faculty of Engineering Science, 2013. PhD thesis, promotor: Bart De Moor.
- [8] D. Gleich. `gaimc`: Graph algorithms in matlab code. Matlab File Exchange, <http://www.mathworks.com/matlabcentral/fileexchange/24134-gaimc---graph-algorithms-in-matlab-code>.
- [9] Y. Nakatsukasa, V. Noferini, and A. Townsend. *Computing the common zeros of two bivariate functions via Bézout resultants*. Springer-Verlag Berlin Heidelberg, 2014.
- [10] B. Plestenjak and M. E. Hochstenbach. Roots of bivariate polynomial systems via determinantal representations. *SIAM J. Sci. Comput.*, 38(2):A765–A788, 2015.
- [11] L. Sorber, M. Van Barel, and L. De Lathauwer. Numerical solution of bivariate and polyanalytic polynomial systems. *SIAM J. Num. Anal.* 52, pages 1551–1572, 2014.
- [12] H. J. Stetter. *Numerical Polynomial Algebra*. Society for Industrial and Applied Mathematics, 2004.
- [13] B. Sturmfels. *Solving Systems of Polynomial Equations*. Number 97 in CBMS Regional Conferences. Amer. Math. Soc., 2002.
- [14] S. Telen. *Solving Systems of Polynomial Equations*. KU Leuven, Department of Computer Science, 2016. Master thesis, promotor: Marc Van Barel.
- [15] J. Verschelde. *Homotopy Continuation Methods for Solving Polynomial Systems*. KU Leuven - Faculty of Engineering Science, 1996. PhD thesis, promotor: Ann Haegemans.
- [16] J. Verschelde. Algorithm 795: Phepack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software Vol. 25, No. 2*, pages 251–276, 1999.