Computation Sequences for Series and Polynomials

Yiming Zhang
*The University of Western Ontario*

Supervisor
Robert M. Corless
*The University of Western Ontario*

Graduate Program in Applied Mathematics
A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy
© Yiming Zhang 2013

Follow this and additional works at: [https://ir.lib.uwo.ca/etd](https://ir.lib.uwo.ca/etd)

Recommended Citation
[https://ir.lib.uwo.ca/etd/1683](https://ir.lib.uwo.ca/etd/1683)

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlswadmin@uwo.ca.
COMPUTATION SEQUENCES FOR SERIES AND POLYNOMIALS
(Thesis Format: Integrated-Article)

by

Yiming Zhang

Graduate Program in Applied Mathematics

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Copyright Yiming Zhang 2013
Abstract

Approximation to the solutions of non-linear differential systems is very useful when the exact solutions are unattainable. Perturbation expansion replaces the system with a sequences of smaller problems, only the first of which is typically linear. This works well by hand for the first few terms, but higher order computations are typically too demanding for all but the most persistent. Symbolic computation is thus attractive; however, symbolic computation of the expansions almost always encounters intermediate expression swell, by which we mean exponential growth in subexpression size or repetitions. A successful management of spatial complexity is vital to compute meaningful results.

This thesis contains two parts. In the first part, we investigate a heat transfer problem where two-dimensional buoyancy-induced flow between two concentric cylinders is studied. Series expansion with respect to Rayleigh number is used to compute an approximation of a solution, using a symbolic-numeric algorithm. Computation sequences are used to help reduce the size of intermediate expressions. Up to 30\textsuperscript{th} order solutions are computed. Accuracy, validity and stability of the computed series solution are studied.

In the second part, Hilbert’s 16\textsuperscript{th} problem is investigated to find the maximum number of limit cycles of certain systems. Focus values of the systems are computed using perturbation theory, which form multivariate polynomial systems. The real roots of such systems leads to possible limit cycle conditions. A modular regular chains approach is used to triangularize the polynomial systems and help to compute the real roots. A system with 9 limit cycles is constructed using the computed real roots.
Keywords: perturbation theory, large expression management, computation sequences, heat transfer, free convection, concentric cylinders, singularities, Quotient-Difference method, Hilbert’s 16th problem, limit cycles, focus values, regular chains, modular algorithm
Co-Authorship Statement

Chapter 2 is co-authored with Rob Corless, and has been submitted for publication. Rob Corless provided guidance through this work, and supervised the research. Rob Corless also reviewed and revised drafts of the paper.

Chapter 3 is co-authored with Rob Corless, Pei Yu, Marc Moreno Maza and Changbo Chen, which is accepted for publication. Rob Corless proposed the subject and supervised throughout the whole period. He also reviewed the final draft of the paper. Pei Yu provided the guidance on Hilbert’s 16th problem and the techniques to compute limit cycles symbolically. He also provided the focus values for the computation and revised many drafts of the paper. Marc Moreno Maza suggested the Regular Chains method to solve the multivariate polynomial system and introduced the modular technique. He also reviewed and revised the final draft of the paper. Changbo computed the real roots using real root isolation algorithm, and contributed a draft of section 3.3.1.
Acknowledgements

First of all, I would like to thank Rob Corless who guided me through this four wonderful years of my life and together we journey through the beautiful arts of applied mathematics. He has been helping me on every aspect of my research work. He shows great patience and trust, and he is always supportive and encouraging. It would not be possible for me to finish this thesis without his inspiration and constant help.

Within the department, I would like to thank Pei Yu for pouring his knowledge of dynamical system to me, and his willingness to help me whenever I meet problems. He devoted large amount of time on reviewing and rewriting almost every part of my draft work on the Hilbert 16th problem. There would be no such a paper without his contribution. I would like to thank Greg Reid for his very interesting super symmetry seminars. We had much fun marching to the field of super symmetry. I am thankful for David Jeffrey’s insightful discussions on the Painlevé transcendents. Even though that work couldn’t ended as a paper, but I get a much deeper understanding of singularity structures of PDEs. I also want to thank Chris Essex for sharing me his knowledge on teaching, who trusted me on taking his position to give lectures when he is occupied.

In our reach group, I would like to thank Marc Moreno Maza for teaching me the regular chains method from zero, and providing the cutting edge programming techniques. He was very patient on my naive questions and explained thoroughly all the technique details to make sure I understand. I also want to thank Changbo Chen for the good discussion and wonderful experience working together on the regular chains programs and drafts. I would like
to thank Rong Xiao for his help on writing and compiling MAPLE procedures.

Back at Brock, I want to thank Thomas Wolf who introduce me to the area of computer algebra. If it wasn’t for him, I would never had the opportunity to meet Rob and pursue my Ph.D. at Western.

Last but not least, I want to thank my wife Chunlei Hua for her support, and the encouragement when I meet obstacles. Together with our son, she brought happiness, joy, and love to my family and the things I would like to strive for.
# Contents

Abstract ii

Co-Authorship Statement iv

Acknowledgements v

List of Tables viii

List of Figures ix

1 Introduction 1

1.1 Motivation 1

1.2 Outline 5

2 High-accuracy series solution for two-dimensional convection in a horizontal concentric cylinder∗ 10

2.1 Introduction 10

2.2 Model Equations 12

2.3 Solution by computation sequences: Perturbation in Rayleigh number 14

2.3.1 Direct integration method 16

2.3.2 The method of undetermined coefficients 20

2.4 Cost of computation 26

2.5 The accuracy of the series solution 27

2.5.1 Stability of the solution 35

2.6 Conclusions 36

*A version of this chapter has been submitted to SIAM Journal on Applied Mathematics.
3 An application of regular chain theory to the study of limit cycles†

3.1 Introduction ................................................. 40
3.2 Incremental solving ........................................... 44
3.3 The regular chains method .................................... 46
  3.3.1 Some definitions and examples for triangular decompo-
        sition ......................................................... 47
  3.3.2 Triangular decomposition algorithm ........................ 50
  3.3.3 A method based on modular techniques for computing
        triangular decomposition ................................. 52
  3.3.4 Isolating real roots of a regular chain .................... 56
3.4 Limit cycle and focus value ................................... 57
3.5 Application to limit cycle computation ......................... 59
  3.5.1 Generic quadratic system ............................... 60
  3.5.2 A special cubic system .................................. 64
3.6 Conclusion .................................................... 70

4 Concluding Remarks ............................................ 76
  A Proof of the shape of the general form ....................... 79
  B QD algorithm ............................................... 89
  C Perturbation method and multiple time scale algorithm .... 92
  D An example of focus value computation ....................... 95
  E Flaws in the paper of Lloyd and Pearson .................... 98
  F Maple input for the quadratic example ..................... 100
  G Maple input for the cubic example ......................... 101

Curriculum Vitae ................................................. 106

†A version of this chapter has been accepted by the International Journal of Bifurcation and Chaos.
List of Tables

2.1 Table form of $T_{31}$, where, the $K$’s are coefficients of homogeneous solution, $C$’s are coefficients of particular solution. For example, $K_1$ is the coefficient of $K_1 r^{-1}$, where $\alpha = -1$ and $\beta = 0$; $C_3$ is the coefficient of $C_3 r^{-1} \ln(r)$, where $\alpha = -1$ and $\beta = 1$.

2.2 Table form of $R_{31}$. $R$’s are dummy variables representing the coefficients of each term. For example, $R_2$ is the coefficient of $R_2 r^{-5} \ln(r)$, where $\alpha = -5$ and $\beta = 1$.

2.3 Accuracy analysis on the examples with nearby defects.

2.4 Nearest pole locations of Rayleigh number $A$, starting from the origin.

A1 The non-zero component of the $T_k^m$ table. In side the boundary, the $\times$ represents some non-zero coefficient $C_{T_k^m,\alpha,\beta}$, while outside the boundary all elements are zeros.

A2 The $\nabla^2 T_k^m$ table in “stair” shape.
# List of Figures

2.1 Sketch of the concentric cylinders ........................................... 13
2.2 Computation routine ................................................................. 18
2.3 Left figure is the growth of number of terms in $T_k$ and $\psi_k$ ($O(k^3)$); the right one is the growth of number of entries in the computation sequence for $T_k$ and $\psi_k$ ($O(k^6)$). ........................................... 27
2.4 Left figure is the log plot of residuals on $T_k$ and $\phi_k$ with $R = 2$ $P = 0.02 \ k \leq 30$, the right one is the log of magnitude of $T_k$ and $\phi_k$ with same parameter. ........................................... 28
2.5 Residual of $T_{30}$. ................................................................. 29
2.6 The singularites and zeros of $R = 2$, $P = 0.02$ case (left), and $R = 2$, $P = 0.7$ case (right) in the complex plane ......................... 31
2.7 The stream contours of $R = 2, P = 0.7$ case .............................. 33
2.8 The stream contours of $R = 7/6, P = 0.2$ case ......................... 34
2.9 The ratios of $\frac{\|\Delta T, \Delta \psi\|}{\|R(r, \theta), S(r, \theta)\|}$ where $A = 2000, R = 2$ and $P = 0.7$ ................................. 36
3.1 The incremental solving of (3.2) ............................................... 46
Chapter 1

Introduction

1.1 Motivation

The integration of multivariate non-linear differential systems is a very important but challenging subject in computer algebra. Exact solutions of many such systems, especially those with complicated nonlinear terms, are beyond the reach of today’s techniques. A popular workaround in computer algebra is to solve a nearby problem as an approximation with good accuracy. Perturbation theory is one such technique, which has a long history, and still remains popular [15, 23, 20, 22]. Other works on perturbation theory in practice include [25, 24, 13, 16]. Due to their complicated structure, it is very natural to use computer algorithms to solve perturbation problems which usually involve the handling of large expressions. Thanks to the advances in both hardware and software techniques, we are able to compute perturbation expansions for systems that could not be solved before. In chapter 2, we use perturbation theory to solve the systems describing the two-dimensional heat convection of a fluid contained in two concentric cylinders. In chapter 3, we use perturbation theory to compute focus values which helps to identify the number of limit cycles on Hilbert’s 16th problem. In both applications, large expression management techniques such as computation sequences and modular methods are the key technique.

In regular perturbation theory, the equations in the target system are ex-
panded with respect to some parameter to form the series expansions. For example, to find the root of
\[ x^3 + x - \varepsilon \]  
that goes to zero as \( \varepsilon \to 0 \).

We can expand \( x \) into Taylor series of \( \varepsilon \), as follows:
\[ x := \sum_{k \geq 1} a_k \varepsilon^k \]  
(1.2)

By substitution, we obtain a sequence of equations for the \( a_k \),
\[
\begin{align*}
    a_1 - 1 &= 0 \\
    a_2 &= 0 \\
    a_1^3 + a_3 &= 0 \\
    3a_1^2 a_2 + a_4 &= 0 \\
    2a_2 a_1 + a_1(2a_1 a_3 + a_2^2) + a_3 a_1^2 + a_5 &= 0 \\
    \cdots
\end{align*}
\]  
(1.3)

We solve these equations one after another and obtain the sequence of the coefficients \( \{a_1, a_2, a_3, \cdots\} \) as
\[
\{1, 0, -1, 0, 3, 0, -12, 0, 55, 0, -273, 0, 1428, 0, -7752, \cdots\}.
\]  
(1.4)

Therefore we arrive at the following approximation to the solution using the perturbation series:
\[ x = \varepsilon - \varepsilon^3 + 3\varepsilon^5 - 12\varepsilon^7 + 55\varepsilon^9 - 273\varepsilon^{11} + 1428\varepsilon^{13} - 7752\varepsilon^{15} + \cdots \]  
(1.5)

At this moment, several questions arise. How accurate is the solution? What is the maximum value that \( |\varepsilon| \) could be? We usually expect that a series solution is more accurate when truncated with higher order, with a small \( \varepsilon \). The values of \( \varepsilon \) must be inside the radius of convergence. In this example, we
can determine the radius of convergence directly. Here, the $a_k$ can be written as
\[
a_k := \begin{cases} 
0 & \text{if } k \text{ is even}, \\
\frac{(-1)^m}{2m+1} \binom{3m}{m} & \text{if } k \text{ is odd}, 
\end{cases}
\]
(1.6)
where $m = (k - 1)/2$. Then the radius of convergence is
\[
\lim_{k \to \infty} \frac{|a_k|}{a_{k+1}} = \lim_{k \to \infty} \sqrt{|\frac{a_{2k+1}}{a_{2k+3}}|} = \lim_{m \to \infty} \sqrt{|\frac{a_m}{a_{m+1}}|} = \lim_{m \to \infty} \sqrt{\frac{(3m)^2}{2m+1} \frac{2m+3}{(3m+3)^2}} = \lim_{m \to \infty} \sqrt{\frac{(2m+3)(m+1)(2m+1)(2m+2)}{(2m+1)(3m+1)(3m+2)(3m+3)}} = 2/3\sqrt{3} \\
\approx 0.385.
\]
(1.7)

If we set $d\varepsilon/dx$ to zero, which is $3x^2 + 1 = 0$, we obtain $x = \pm i/\sqrt{3}$. At these points $dx/d\varepsilon = \infty$, therefore $\varepsilon = \mp i/3\sqrt{3} \pm i/\sqrt{3} = \pm 2/3\sqrt{3}i$, are the exact locations of the singularities, which match the radius of convergence. If we feed the series to MAPLE, it returns
\[
\sum_{m=0}^{\infty} \frac{(-1)^m}{2m+1} \binom{3m}{m} \varepsilon^{2m+1} = 3\varepsilon \cosh \left( \frac{2}{3} \text{arccosh} \left( \frac{3\varepsilon}{2} \right) \right) + \frac{2\sqrt{3}}{9} \left( -3 - \frac{81}{4} \varepsilon^2 \right) \sinh \left( \frac{2}{3} \text{arcsinh} \left( \frac{3\sqrt{3}\varepsilon}{2} \right) \right) \\
\sqrt{1 + \frac{27}{4} \varepsilon^2},
\]
(1.8)
which confirms the singular points to be $\pm 2/3\sqrt{3}i$. In this simple example, the solving process for the series coefficients is easy. In more general cases, for example when solving differential equations, the perturbation series are usually tremendously complicated with a fast growth in size of the resulting expressions. The perturbation series for many non-linear systems, such as the ones presented in Chapter 2 and 3, are so complicated that any direct
translating of the perturbation expansion will quickly run out of memory. We consider the spatial complexity to be the number one issue to overcome.

Accuracy is another problem we encounter when computing series solutions using perturbation technique for nonlinear differential equations. The dependency of higher order solutions on lower ones can amplify the numerical error from lower order solutions. For example, a term in a lower order solution which should be zero, might be stored as some small number because of numerical error. After several steps of integration, when the program carries this error term to higher order solution, it may result in many terms that shouldn’t exist. When these terms are integrated, even more error terms show up. These errors could go quickly out of control. Therefore, we used a pure symbolic approach that computes the series solutions where the coefficients of each term are symbols. However, in such symbolic integration processes the expression swell is also a difficulty. Large-expression management techniques are needed to control the rapid growth of space usage, thereafter help us to arrive at high order solutions. Eventually, numerical evaluation does take place. In the event that expressions are ill-conditioned, higher precision must be used. [25, 12, 7]

An even more important problem is to make sure the computed series solutions truly represent the exact solution. As demonstrated in the previous example, series solutions must be bounded by the radius of convergence. We declare a series solution to be valid when the expansion parameter $\varepsilon$ is inside the radius of convergence. If the radius of convergence of the series solution is very small, no useful information of the system but the expansion point can be found immediately. Unlike the previous example, the radius of convergence is not always easily obtainable. A method that helps determine the radius of convergence is needed as well. In many cases, the solutions of nonlinear differential systems posses complicated singularity structures such as movable poles, essential singularities, branch cuts etc. These singularity structures have a great impact on the radius of convergence of the series solution. By Darboux’s principle [11, 5, 6], the convergence of a series expansion is determined by the nearest singularity. The distance between the point of expansion (usually the origin) and the nearest pole is the maximum range that the expansion
parameter \( \varepsilon \) should be used in. To deal with this problem, singularity detection techniques such as the Quotient-Difference (QD) algorithm (please see appendix B), are needed to ensure the validity of such solutions. In the previous example, we input the series solution to the QD algorithm, the estimated nearest pole location is 0.395 which is within 3% of the true value.

1.2 Outline

In the first part of this thesis, we investigate the heat transfer of fluids contained in the annulus between two horizontally placed concentric cylinders. Two dimensional flow behavior for free convection\(^*\) is studied. We used the perturbation expansion with respect to Rayleigh number \( \mathcal{A} \), following the work of Mack & Bishop [19] who computed series solution of the second order by hand. Corless \textit{et al.} pushed the series solution to 10\(^{th}\) order [7]. They introduced computation sequences to simplify the intermediate expression swell. However at this order not many conclusions could be drawn firmly. We extended the work of Corless \textit{et al.} [7], optimized the computation sequences and reprogrammed the symbolic-numerical solver, thereby pushing the result to 16\(^{th}\) order. The solver applies a simplified direct integration method. During the computation of order \( k^{th} \) solution, the algorithm computes particular solutions according to each term of the inhomogeneous parts. The solutions are collected after all terms of the inhomogeneous parts are taken into consideration and then combined with the general solutions. At this point the coefficients in many terms of the solution are very complicated. We use new symbols to substitute these coefficients, and record the evaluation relation of these symbols in the computation sequences. These coefficients are not evaluated until the end of the symbolic stage when the desired order solutions are computed symbolically.

For a second, greatly improved algorithm, we recognized the pattern of solution of each order and summarized it into a general form. Applying the general form we designed a more efficient algorithm using the method of un-

\(^*\)The fluid is only influenced by gravity.
known coefficients. This greatly reduces the size of intermediate expressions. The new algorithm decreased the spatial complexity from $O(n^7)$ to $O(n^4)$, where the solution is truncated at $n^{\text{th}}$ order. We take advantage from this efficiency and successfully computed solution to the 30$^{\text{th}}$ order. With this high order solution, reliable information of the system can be extracted. As pointed out by Y.F Chang [8], 30$^{\text{th}}$ order solutions allow good estimates of nearby singularities and their properties. Thereafter the series provides the range on Rayleigh number $A$, where within the range the solutions are valid.

The QD method [14, 3, 9, 10, 1] is the main technique used here to detect singularities. Comparing to other methods, such as Padé approximants [3], the QD method has many advantages. It does not require information on the singularity structure a priori. It can handle the cases where defects happen. It works well with a small radius of convergence, where singularities are very close to the origin.

The errors of the computed series solutions are estimated using residual tests. The stability of the solution is also analyzed by perturbing the system with additional nonlinear terms. We observed the difference between the solutions and original ones compared to the size of the additional terms.

In the second part of the thesis, normal form theory and perturbation expansion are used to identify the number of limit cycles of quadratic and cubic planar polynomial systems. We investigated Hilbert’s 16$^{\text{th}}$ problem, which asks for an upper bound of number on the limit cycles for a system in the form of

$$
\dot{x} = F(x, y), \quad \dot{y} = G(x, y),
$$

(1.9)

where $F(x, y)$ and $G(x, y)$ are degree $k$ polynomials of variables $x$ and $y$, with real coefficients. The problem is narrowed to the case of small-amplitude limit cycles bifurcating from a center at the origin. In this case, the number of such limit cycles can be obtained by focus value computations. This problem has been solved for generic quadratic systems [4], where at most three such limit cycles could exist. For cubic systems, James and Lloyd obtained [18] a

\[^{1}\text{A defect in Padé approximants is the case where a nearby singularity is accompanied by a close zero}\]
special cubic system with eight limit cycles. Yu and Corless [2009] showed the existence of nine limit cycles with the help of a numerical method for another special cubic system. We will symbolically compute the case of 9 limit cycles.

In this work, the focus values are computed using perturbation theory on multiple time scales. The parameters of the system becomes the variable of the output focus values, which are multivariate polynomial equations. The real solutions of these equations will provide possible condition that the system consists certain number of limit cycles.

In order to find the \( n \) limit cycles in a cubic system, there must be at least \( n \) free parameters, and \( n + 1 \) focus values to be computed (one more focus value is needed to distinguish between the limit cycle conditions and the center conditions). Due to the rapid growth in size of the higher-order focus values, as expected, it is very hard to compute symbolic solutions of these equations. Direct solving on such system fails for the built-in MAPLE solver. The more powerful tools such as the Gröbner bases package in MAPLE quickly ran out of memory as well. Instead, we applied a modular technique [17] on the regular chains [21, 2] method to compute the triangular decomposition of the focus value system. Please refer to Section 3.3.2 for an example of the regular chains method. Some large enough primes are used during the computation process to decrease the size of intermediate expressions. The result of the modular triangular decomposition is then verified using another prime with similar size, and lifted using the first prime. The lifting process provides regular chains in a triangular shape which have the same common zeros as the input system. All the real roots are isolated and represented by intervals where each interval contains one and only one real root. The size of the intervals can be made arbitrarily small on demand. This interval representation is commonly viewed as a symbolic solution since it is fully compatible with the symbolic procedures. With one set of roots as an example, we constructed the cubic system that contains nine small limit cycles. To our knowledge it is the best symbolic result so far, and provides a rigorous proof for the existence of nine limit cycles in a cubic system.
Bibliography


Chapter 2

High-accuracy series solution for two-dimensional convection in a horizontal concentric cylinder

2.1 Introduction

Heat transfer via natural convection in horizontal concentric cylinders has attracted much attention, due to its wide practical application and interesting dynamical behavior. Following the first comprehensive study by Liu, et al. (1961) [17] using air as the fluid, many experiments were conducted in the 1960’s by Bishop & Carley [5], Grigull & Hauf [12] and Lis [16] with different diameter ratio of cylinders and different Grashof number. Powe et al. [19, 20] summarized their results on the convective flow of air and categorized the flow pattern into steady flow, oscillatory flow, three-dimensional spiral flow and multicellular flow. Labonia & Guj [15] conducted experiments using large Rayleigh number \( A \in [0.9E5, 3.3E5] \) and observed chaos (as one might expect nowadays).

In terms of computational studies, Mack and Bishop [18] applied a perturbation expansion in the Rayleigh number \( A \) and obtained a series solution of second order. They suggested an approximation for a limiting value \( A_{lim} \) above which their solution was not to be trusted; by implication, it was con-

* A version of this chapter has been submitted to SIAM Journal on Applied Mathematics.
sidered trustworthy for $A < A_{lim}$. We will pursue this solution method to very high order in this present work, and give reliable accurate estimates for $A_{lim}$.


Most numerical studies are conducted using finite difference methods. Each study chooses some specified settings of Prandtl number (type of the fluent), Rayleigh number (heat difference of the cylinders) and radius ratio (shape of the concentric cylinder). In the case of series solution, Mack & Bishop [18] gave a second order series solution valid for low Rayleigh number $A$. Further, their estimate of the upper limit of validity of their solution, which they called $A_{lim}$, was of unknown reliability. They used a perturbation expansion with respect to Rayleigh number to obtain the steady state solution of the stream and heat equations. Corless et al. [7] investigated the problem in a computer algebra point of view. In [7] the series solution of the problem was extended to $10^{th}$ order by computer algebra using the then-novel technique of Large Expression Management. The principal concern of that work was efficient computer algebra.

In this article, we will extend that series solution to very high order in the Rayleigh number for arbitrary values of the parameters. The choice of parameters do influence the accuracy of the series solution, which will be discussed in section 5. We provide a reliable method for assessing precisely how small $A$ must be for the solution to be valid.

Following Mack & Bishop’s method, the same expansion is applied with an additional Fourier expansion to remove the $\theta$ dependence. A direct integration algorithm is developed to systematically solve the differential equations.
generated from the double expansions. The pattern of the symbolic solution is recognized as some general form, and applied to develop a much more efficient algorithm using the method of unknown coefficients.

The current approach generates a symbolic program that, given values for Prandtl number $P$, and radius ratio $R$, can evaluate all terms up to $O(A^{30})$ exactly or in arbitrary high precision. Error analysis for the latter is discussed in section five as well. At this high order, reliable techniques for detecting and locating singularities become available. Here, we use the Quotient-Difference (QD) method to identify the structure of the singularities of the computed series solution, and thereafter provide an estimate on the validity of the serious solution.

2.2 Model Equations

Following the discussion of Mack and Bishop [18], the model contains two equations:

$$\nabla^4 \psi = A \cdot L(T) + \frac{1}{P \cdot r} \frac{\partial (\nabla^2 \psi, \psi)}{\partial (r, \theta)},$$  

$$\nabla^2 T = \frac{1}{r} \frac{\partial (T, \psi)}{\partial (r, \theta)},$$

where $\psi$ is the stream function, $T$ is temperature, $P$ is the Prandtl number and $A$ is the Rayleigh number, and

$$L(T) = \sin(\theta) \frac{\partial T}{\partial r} + \cos(\theta) \frac{\partial T}{\partial \theta},$$

$$\frac{\partial (T, \psi)}{\partial (r, \theta)} = \frac{\partial T}{\partial r} \frac{\partial \psi}{\partial \theta} - \frac{\partial T}{\partial \theta} \frac{\partial \psi}{\partial r},$$

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2}, \quad \nabla^4 = \nabla^2 (\nabla^2).$$

Note that both the stream function $\psi$ and temperature $T$ are nondimensional quantities. They have the following relationship with the dimensional
Figure 2.1: Sketch of the concentric cylinders

quantities (all dimensional quantities are marked with primes).

\[ r = \frac{r'}{r_i}, \quad r' \in [r'_i, r'_o], \]
\[ T = \frac{T'_i - T'_o}{T'_i - T'_o} \quad T' \in [T'_i, T'_o], \]
\[ \psi = \frac{\psi'}{\alpha'}, \]
\[ P = \frac{\nu'}{\alpha'}, \]
\[ A = g\beta' (T'_i - T'_o) r'_i^3. \]

(2.3)

Here \( r'_i \) is the radius of the inner cylinder, \( r'_o \) is the radius of the outer cylinder, and their ratio is \( R = \frac{r'_o}{r'_i} \). We define \( r = \frac{r'}{r_i} \), where \( r'_i \leq r' \leq r'_o \) such that \( 1 \leq r \leq R \). \( T'_i \) and \( T'_o \) represent the temperature of inner and outer boundary respectively. \( \alpha' = \frac{k'}{\rho' C'_p} \) is the fluid thermal diffusivity, \( k' \) is the thermal conductivity, \( \rho' \) is the density and \( C'_p \) is the specific heat capacity. \( \nu' \) is the fluid kinematic viscosity, \( g' \) is the acceleration due to gravity and \( \beta' \) is coefficient of volumetric expansion.
The equation (2.1) and (2.2) obey the following boundary conditions:

\[ T(1, \theta) = 1 , \quad (2.4) \]

\[ T(R, \theta) = 0 , \quad (2.5) \]

\[ \psi(1, \theta) = \psi(R, \theta) = \frac{\partial \psi}{\partial r}(1, \theta) = \frac{\partial \psi}{\partial r}(R, \theta) = 0 , \quad (2.6) \]

\[ \frac{\partial T}{\partial \theta}(r, 0) = \psi(r, 0) = \frac{\partial^2 \psi}{\partial \theta^2}(r, 0) = 0 , \quad (2.7) \]

\[ \frac{\partial T}{\partial \theta}(r, \pi) = \psi(r, \pi) = \frac{\partial^2 \psi}{\partial \theta^2}(r, \pi) = 0 . \quad (2.8) \]

The boundary condition (2.4) and (2.5) define the temperatures of the annulus boundaries. The condition (2.6) ensures no flow passes through the boundaries. The initial conditions (2.7) and (2.8) define the flow to be symmetric with respect to the vertical line of \( \theta = 0 \) and \( \theta = \pi \).

### 2.3 Solution by computation sequences: Perturbation in Rayleigh number

Assume that \( T \) and \( \psi \) can be expanded in a convergent power series with respect to the Rayleigh number \( A \),

\[ T = \sum_{j=0}^{\infty} A^j T_j(r, \theta) , \quad (2.9) \]

\[ \psi = \sum_{j=1}^{\infty} A^j \psi_j(r, \theta) . \quad (2.10) \]

Substitute these power series into equations (2.1) and (2.2), and isolate the coefficients of the same powers of \( A \). This yields two infinite sets of equations,

\[ \nabla^2 T_k = \frac{1}{r} \sum_{j=0}^{k-1} \frac{\partial (T_j, \psi_{k-j})}{\partial (r, \theta)} , \quad k = 0, 1, 2, \ldots \quad (2.11) \]
\[ \nabla^4 \psi_k = \frac{1}{P \cdot r} \sum_{j=1}^{k-1} \frac{\partial(\nabla^2 \psi_j, \psi_{k-j})}{\partial (r, \theta)} + L(T_{k-1}), \quad k = 1, 2, 3, \ldots \]  

(2.12)

According to the series expansion, the boundary conditions become

\[ T_0(1, \theta) = 1, \]  

(2.13)

\[ T_0(R, \theta) = 0, \]  

(2.14)

\[ T_j(1, \theta) = T_j(R, \theta) = 0, \quad j = 1, 2, 3, \ldots, \]  

(2.15)

\[ \frac{\partial T_j}{\partial \theta}(r, 0) = \frac{\partial T_j}{\partial \theta}(r, \pi) = 0, \quad j = 0, 1, 2, \ldots, \]  

(2.16)

\[ \psi_j(1, \theta) = \frac{\partial \psi_j}{\partial r}(1, \theta) = \psi_j(R, \theta) = \frac{\partial \psi_j}{\partial r}(R, \theta) = 0, \quad j = 1, 2, 3, \ldots, \]  

(2.17)

\[ \psi_j(r, 0) = \frac{\partial^2 \psi_j}{\partial \theta^2}(r, 0) = \psi_j(r, \pi) = \frac{\partial^2 \psi_j}{\partial \theta^2}(r, \pi) = 0, \quad j = 1, 2, 3, \ldots. \]  

(2.18)

We further expand \( \psi_k \) and \( T_k \) in Fourier series with respect to \( \theta \),

\[ T_k(r, \theta) = \sum_{m=0}^{k} T^m_k(r) \cos(m\theta), \quad k = 0, 1, 2, \ldots, \]  

(2.19)

\[ \psi_k(r, \theta) = \sum_{m=0}^{k} \psi^m_k(r) \sin(m\theta), \quad k = 1, 2, 3, \ldots, \]  

(2.20)

to remove the \( \theta \) dependence. In the Fourier series, the odd numbered terms are zero if \( k \) is even, and even numbered terms are zero if \( k \) is odd. Substituting the Fourier series into equations (2.11) and (2.12) yields two infinite sequences of ordinary differential equations for functions \( T^m_k(r) \) and \( \psi^m_k(r) \) which only depend on \( r \). These new equations are of Euler type:

\[ \left( \frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{m^2}{r^2} \right) T^m_k(r) = R^m_k(r), \]  

(2.21)

\[ \left( \frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{m^2}{r^2} \right) \left( \frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{m^2}{r^2} \right) \psi^m_k(r) = S^m_k(r), \]  

(2.22)
where the inhomogeneous parts \( R^m_k(r) \) and \( S^m_k(r) \) are in terms of lower order \( T^m_k(r) \) and \( \psi^m_k(r) \), and always have the form \( \sum C_i r^\alpha \ln^\beta(r) \). \( C_i \) \( i = 0, 1, 2 \ldots \) form a computation sequence, because each of them is defined in terms of previously computed \( C_k \) or \( K_\ell \), where \( k, \ell < i \). The general solutions of these Euler type equations are summations of homogeneous solutions and particular solutions. The homogeneous solutions are as follows,

\[
T^m_{H,k} = \begin{cases} 
K_1 + K_2 \ln(r) & \text{if } m = 0, \\
K_1 r^{-m} + K_2 r^m & \text{if } m \neq 0,
\end{cases}
\]

\[
\psi^m_{H,k} = \begin{cases} 
K_1 + K_2 \ln(r) + K_3 r^2 + K_4 \ln(r) r^2 & \text{if } m = 0, \\
K_1/r + K_2 r + K_3 r^3 + K_4 \ln(r) r & \text{if } m = 1, \\
K_1 r^{-m} + K_2 r^{-m+2} + K_3 r^m + K_4 r^{m+2} & \text{if } m \neq 0, m \neq 1,
\end{cases}
\]

where \( K_1, K_2, K_3 \) and \( K_4 \) are unknown coefficients directly solvable the boundary conditions. The particular solutions given the inhomogeneous parts in terms of \( C_{\alpha,\beta} r^\alpha (\ln(r))^\beta \) are always computable. We use MAPLE to do the bookkeeping of the inhomogeneous terms and compute the particular solutions of the equations (2.21) and (2.22). Observe that in (2.22) the operator \((\frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - m^2)\) is applied twice, so a program that computes the particular solution of (2.21) can be used to find the solution of (2.22) as well. In the following section we will give an algorithm that computes the particular solution of (2.21).

### 2.3.1 Direct integration method

Equation (2.21) can be integrated using substitutions and an “integrating factor”. For an inhomogeneous term in general form \( C r^\alpha (\ln(r))^\beta \), we have

\[
\left( r^2 \frac{d^2}{dr^2} + r \frac{d}{dr} - m^2 \right) T = C r^\alpha (\ln(r))^\beta. \tag{2.25}
\]
By linearity we may take $C = 1$ without loss of generality. We apply the substitution $x = \ln r$ to factorize the operator on $T$.

$$\left(\frac{d^2}{dx^2} - m^2\right)T = \left(\frac{d}{dx} + m\right)\left(\frac{d}{dx} - m\right)T = e^{\alpha x}x^\beta. \tag{2.26}$$

The equation is separated into two similar ones,

$$\left(\frac{d}{dx} + m\right)v = e^{\alpha x}x^\beta, \tag{2.27}$$

$$\left(\frac{d}{dx} - m\right)T = v, \tag{2.28}$$

and (2.27) is integrated first (order does not matter since these operators commute\(^\dagger\)). The second substitution $v = ue^{\alpha x}$ is introduced such that $v' = (u' + \alpha u)e^{\alpha x}$ and (2.27) becomes

$$\frac{du}{dx} + (\alpha + m)u = x^\beta. \tag{2.29}$$

Suppose $\alpha + m \neq 0$, (2.29) is integrated using the integration factor $e^{(\alpha + m)x}$,

$$u = e^{-(\alpha + m)x} \int x^\beta e^{(\alpha + m)x} \, dx = e^{-(\alpha + m)x} \sum_{k=0}^{\beta} \gamma_k x^{(\beta-k)} = e^{-(\alpha + m)x} \sum_{k=0}^{\beta} \gamma_k r^{(\beta-k)},$$

where $\gamma_0 = \frac{1}{\alpha + m}$, $\gamma_j = -\frac{\gamma_{j-1}(\beta-j+1)}{\alpha + m}$, $j = 1, 2, \ldots, \beta$. Thus, according to $v = ue^{\alpha x}$ and $x = \ln(r)$,

$$v = e^{\alpha x} \sum_{k=0}^{\beta} \gamma_k x^{(\beta-k)} = e^{\alpha x} \sum_{k=0}^{\beta} \gamma_k r^{(\beta-k)}, \tag{2.31}$$

$$\gamma_0 = \frac{1}{\alpha + m}, \quad \gamma_j = -\frac{\gamma_{j-1}(\beta-j+1)}{\alpha + m}, \quad j = 1, 2, \ldots, \beta.$$

\(^\dagger\)That is, we could equally well have chosen to integrate instead the pair $\left(\frac{d}{dx} - m\right)u = e^{\alpha x}x^\beta$ followed by $\left(\frac{d}{dx} - m\right)T = u$ in that order.
For the special case \( \alpha + m = 0 \), equation (2.29) degenerates into \( u' = x^\beta \) which has the solution

\[
\begin{align*}
u &= \frac{1}{\beta + 1} x^{\beta+1}, \\
v &= \frac{1}{\beta + 1} x^{\beta+1} e^{\alpha x} = \frac{1}{\beta + 1} \ln(r) x^{\beta+1} r^\alpha.
\end{align*}
\]

(2.32)

Observe that \( v \) is also in the form of \( \sum C r^\alpha \ln^\beta(r) \), so (2.28) can be solved similarly to obtain \( T \) which is the particular solution of (2.21).

A Maple procedure has been written to systematically solve \( T_k^m \) and \( \psi_k^m \) following a specific order of the equations (see Figure 2.2). The solution of \( T_0^0(r) \) is computed when the program starts. Then for each higher order \( k > 0 \), \( \psi_k^m \) are computed first followed by \( T_k^m \), where \( m \) is increasing.

The program contains two stages. In the first stage, the program computes the solution symbolically, where the evaluation sequence of each unknown constant is collected rather than solved. The symbolic solution is computed to an order specified by the call. In the second stage, all the evaluation sequences are evaluated following the same routine as in Figure 2.2 to determine the
unknown coefficients.

The first several symbolic solutions generated by our program are printed here:

\[ T_0 = K_1 + K_2 \ln(r) \]

\[ \psi_1 = \frac{K_3}{r} + 1/16 K_2 r^3 \ln(r) + K_6 r \ln(r) - 1/32 C_1 r^3 + K_5 r \]

\[ T_1 = -1/4 \frac{C_4}{r} - 1/2 \frac{K_2 K_3}{\ln(r)} r + \frac{1}{128} K_2 r^3 \ln(r) + 1/4 C_3 r^2 (\ln(r))^2 + 1/4 K_2 K_6 r (\ln(r))^2 - 1/512 C_2 r^3 + K_5 r \]

\[ \psi_2 = K_{10} - 1/128 C_1 r^2 - 1/9216 C_{17} r^4 + 1/128 K_9 r^2 - 1/12288 1/P K_2 r^6 (\ln(r))^2 + 1/147456 C_{11} r^6 \ln(r) - 1/3538944 C_{12} r^6 + 1/128 C_{15} r^2 \ln(r) - 1/64 C_9 r^2 (\ln(r))^2 + 1/2304 C_{13} r^4 \ln(r) - 1/384 C_7 r^4 (\ln(r))^2 \]

\[ T_2 = \frac{1}{24576} K_2 r^3 (\ln(r))^2 - \frac{1}{294912} K_1 r^6 \ln(r) + \frac{1}{884736} C_1 r^6 \ln(r) + \frac{1}{512} K_6 r^4 (\ln(r))^3 + \frac{1}{2048} C_2 r^4 (\ln(r))^2 - \frac{1}{4096} C_1 r^4 \ln(r) - \frac{1}{16384} C_2 r^4 + 1/16 K_2 K_6 r^2 (\ln(r))^3 + \frac{1}{128} C_{23} r^2 (\ln(r))^2 + \frac{1}{512} C_2 r^2 \ln(r) - \frac{1}{2048} C_{25} r^2 + 1/16 C_6 (\ln(r))^2 - 1/24 K_2 K_3 K_6 (\ln(r))^3 + 1/8 K_2 K_3^2 r^2 (\ln(r)) + 1/8 C_2^2 C_6 (\ln(r))^2 + K_13 + K_14 r \ln(r) \]

\[ T_2 = -1/32 C_{37} + K_1 r^2 - 1/331776 C_{32} r^4 + 1/64 C_4 r^2 - 1/384 C_{33} r^2 (\ln(r))^3 - 1/768 K_2 K_6 r^4 (\ln(r))^3 + 1/196608 K_2^2 r^6 (\ln(r))^2 - 1/16 C_3 r^6 + 3/16 K_2 K_3 K_6 (\ln(r))^2 + \frac{1}{2359296} C_2 r^6 \ln(r) - \frac{1}{56623104} C_{29} r^6 - 1/1024 C_3 r^6 \ln(r) + 1/512 C_4 r^6 (\ln(r))^2 - 1/55296 C_{31} r^4 \ln(r) - 1/4608 C_{30} r^4 (\ln(r))^2 \]

The unknown constants \( K_1, K_2, \ldots \) are introduced when computing the general solutions of the homogeneous equations, and they are efficiently computable from boundary conditions. The \( C \)'s are introduced during the computation of particular solutions, and they generally depend on the radius ratio.
$R$, Prandtl number $P$ and previously defined $C'$s and $K'$s. Maple does all the bookkeeping of the evaluation information contained in computation sequences of these unknown constants. Each time a symbolic solution of certain $T^m_k$ or $\psi^m_k$ is computed, the coefficients of the $r^\alpha \ln(r)^\beta$ terms are examined. Those coefficients that contain more than one monomial are substituted using a new $C$, and recorded in the appropriate computation sequence. In this way, the size of the input for the direct solving procedure are kept under control, which makes it possible to compute symbolic solutions to higher orders.

During the computation process, some $K$ may share the same value as $C$, but we only keep their relationship and never substitute using the $C$, since the computational efficiency gained by doing so will be offset by more complicated bookkeeping.

This algorithm successfully computed solution to the 18th order, which contains totally 111557 terms, 560 $K'$s, 83286 $C'$s, used about 22 hours and 41140.2MB memory.

### 2.3.2 The method of undetermined coefficients

With a careful investigation and verification we found the pattern of the symbolic solutions of each order, so that we can attack the expression swell in the intermediate steps. For $k \geq 1$, $T^m_k$ and $\psi^m_k$ obey the following general form

\begin{equation}
T^m_k = \sum_{\alpha=-k/2}^{-m/2} \sum_{\beta=0}^{1+2\alpha+k} C^m_{T^m_k,2\alpha,\beta} r^{2\alpha} \ln^\beta r + \sum_{\alpha=-m/2+1}^{m/2} \sum_{\beta=0}^{k-m/2+\alpha+1} C^m_{T^m_k,2\alpha,\beta} r^{2\alpha} \ln^\beta r \\
+ \sum_{\alpha=m/2+1}^{3k/2-1} \sum_{\beta=0}^{k+1} C^m_{T^m_k,2\alpha,\beta} r^{2\alpha} \ln^\beta r + \sum_{\beta=0}^{k} C^m_{T^m_k,3k,\beta} r^{3k} \ln^\beta r ,
\end{equation}

(2.33)

---

\textsuperscript{1}The uncommon half integers in $\alpha$ are used to summarize the even and odd case equations into one general form.
\[
\psi^m_k = \sum_{\alpha=-k/2+1}^{-m/2-1} \sum_{\beta=0}^{2\alpha+k-1} C_{\psi^m_k,2\alpha,\beta} r^{2\alpha} \ln^\beta r + \sum_{\alpha=-m/2}^{m/2} \sum_{\beta=0}^{k-m/2+\alpha} C_{\psi^m_k,\alpha,\beta} r^{2\alpha} \ln^\beta r + \sum_{\alpha=m/2+1}^{3k/2} \sum_{\beta=0}^k C_{\psi^m_k,2\alpha,\beta} r^{2\alpha} \ln^\beta r.
\]

For a proof of this general form, please refer to Appendix A.

Using (2.33) and (2.34), we designed a new algorithm that starts from the known form of the symbolic solution and evaluates the coefficients according to that. Similar to the direct integration method, the coefficients are still distinguished, where the \(K\)’s are general solution coefficients, and the \(C\)’s are particular solution coefficients. This is not shown in the above general form, but is used in the algorithm to construct the \(T_k^m\) and \(\psi_k^m\). It is natural to distinguish \(K\) and \(C\), since when the solutions are substituted to (2.21) and (2.22), the terms that contain \(K\)’s vanish and those contain \(C\)’s equal the inhomogeneous terms. Therefore, \(K\)’s are evaluated using the boundary conditions and \(C\)’s are computed using the unknown coefficient method.

In the new algorithm, solutions of \(T_k^m\), \(\psi_k^m\) and their corresponding right hand sides \(R_k^m\), \(S_k^m\) are stored in tables. The table structure helps to demonstrate how the terms containing \(C\)’s are mapped to the inhomogeneous parts. We also take advantage from less storage space and faster accessing time instead of using an explicit polynomial data structure. In the tables, the row index is \(\alpha\) and column index is \(\beta\) which are powers of \(r\) and \(\ln(r)\) respectively. The entries are the corresponding coefficients \(C\) or \(K\). The procedure only needs to write down the coefficients of each entry other than the whole expression containing \(r^\alpha \ln(r)^\beta\). To demonstrate the table structure, we take the equation

\[
\left( \frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{1}{r^2} \right) T_3^1 = R_3^1,
\]

(2.35)
Table 2.1: Table form of \( T_3^1 \), where, the \( K \)'s are coefficients of homogeneous solution, \( C \)'s are coefficients of particular solution. For example, \( K_1 \) is the coefficient of \( K_1 r^{-1} \), where \( \alpha = -1 \) and \( \beta = 0 \); \( C_3 \) is the coefficient of \( C_3 r^{-1} \ln(r) \), where \( \alpha = -1 \) and \( \beta = 1 \).

as an example where

\[
T_3^1 = K_1 r^{-1} + K_2 r + C_1 r^{-3} + C_2 r^{-3} \ln^2(r) + C_3 r^{-1} \ln(r) + C_4 r^{-1} \ln^2(r) + C_5 r^{-1} \ln^3(r) \\
+ C_6 r \ln(r) + C_7 r \ln^2(r) + C_8 r \ln^3(r) + C_9 r \ln^4(r) + C_{10} r^3 + C_{11} r^3 \ln(r) + C_{12} r^3 \ln^2(r) \\
+ C_{13} r^3 \ln^3(r) + C_{14} r^3 \ln^4(r) + C_{15} r^3 + C_{16} r^3 \ln(r) + C_{17} r^3 \ln^2(r) + C_{18} r^3 \ln^3(r) \\
+ C_{19} r^5 \ln^4(r) + C_{20} r^7 + C_{21} r^7 \ln(r) + C_{22} r^7 \ln^2(r) + C_{23} r^7 \ln^3(r) + C_{24} r^7 \ln^4(r) \\
+ C_{25} r^9 + C_{26} r^9 \ln(r) + C_{27} r^9 \ln^2(r) + C_{28} r^9 \ln^3(r) ,
\]

(2.36)

and

\[
R_3^1 = R_1 r^{-5} + R_2 r^{-5} \ln(r) + R_3 r^{-3} + R_4 r^{-3} \ln(r) + R_5 r^{-3} \ln^2(r) + R_6 r^{-1} + R_7 r^{-1} \ln(r) \\
+ R_8 r^{-1} \ln^2(r) + R_9 r^{-1} \ln^3(r) + R_{10} r + R_{11} r \ln(r) + R_{12} r^2 \ln^2(r) + R_{13} r^3 \ln^3(r) + R_{14} r^4 \ln^4(r) \\
+ R_{15} r^3 + R_{16} r^3 \ln(r) + R_{17} r^3 \ln^2(r) + R_{18} r^3 \ln^3(r) + R_{19} r^3 \ln^4(r) + R_{20} r^5 + R_{21} r^5 \ln(r) \\
+ R_{22} r^5 \ln^2(r) + R_{23} r^5 \ln^3(r) + R_{24} r^5 \ln^4(r) + R_{25} r^7 + R_{26} r^7 \ln(r) + R_{27} r^7 \ln^2(r) \\
+ R_{28} r^7 \ln^3(r) ,
\]

(2.37)

are written in Table 2.1 and Table 2.2 respectively.

In order to compute the inhomogeneous coefficients \( C \)'s, the solution is substituted into (2.35). A term \( C_r^\alpha \ln(r)^\beta \) in \( T_k^m \) will be mapped by the
\[
\begin{array}{c|ccccc}
\alpha \backslash \ln(r)^eta & \beta = 0 & \beta = 1 & \beta = 2 & \beta = 3 & \beta = 4 \\
\hline
\alpha = -5 & R_1 & R_2 & 0 & 0 & 0 \\
\alpha = -3 & R_3 & R_4 & R_5 & 0 & 0 \\
\alpha = -1 & R_6 & R_7 & R_8 & R_9 & 0 \\
\alpha = 1 & R_{10} & R_{11} & R_{12} & R_{13} & R_{14} \\
\alpha = 3 & R_{15} & R_{16} & R_{17} & R_{18} & R_{19} \\
\alpha = 5 & R_{20} & R_{21} & R_{22} & R_{23} & R_{24} \\
\alpha = 7 & R_{25} & R_{26} & R_{27} & R_{28} & 0 \\
\end{array}
\]

Table 2.2: Table form of \(R_1^3\). R’s are dummy variables representing the coefficients of each term. For example, \(R_2\) is the coefficient of \(R_2 r^{-5} \ln(r)\), where \(\alpha = -5\) and \(\beta = 1\).

\[
\frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{1}{r^2}
\]

operator into

\[
\left( \frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{1}{r^2} \right) (Cr^\alpha \ln(r)^\beta)
\]

\[
= C(\alpha^2 - m^2) r^{\alpha-2} \ln(r)^\beta + 2C\alpha\beta r^{\alpha-2} \ln(r)^{\beta-1} + C\beta(\beta - 1) r^{\alpha-2} \ln(r)^{\beta-2}.
\]

(2.38)

Therefore, the elements in row \(\alpha\) of the Table 2.1 will be mapped into row \(\alpha-2\) of Table 2.2. This mapping can be written into a matrix form. For instance, we write the \(C\)’s in the fourth row \((\alpha = 3)\) of Table 2.1 as a vector \(C = \langle C_{10}, C_{11}, C_{12}, C_{13}, C_{14} \rangle^T\), and the fourth row \((\alpha = 1)\) of Table 2.2 into a vector \(R = \langle R_{10}, R_{11}, R_{12}, R_{13}, R_{14} \rangle^T\). Then the mapping has the matrix form \(M_1 C = R\), where

\[
M_1 = \begin{bmatrix}
\alpha^2 - m^2 & 0 & 0 & 0 & 0 \\
2\alpha\beta & \alpha^2 - m^2 & 0 & 0 & 0 \\
\beta(\beta - 1) & 2\alpha(\beta + 1) & \alpha^2 - m^2 & 0 & 0 \\
0 & (\beta + 1)(\beta) & 2\alpha(\beta + 2) & \alpha^2 - m^2 & 0 \\
0 & 0 & (\beta + 2)(\beta + 1) & 2\alpha(\beta + 3) & \alpha^2 - m^2
\end{bmatrix},
\]

(2.39)

\(\alpha = 3, \beta = 1\) and \(m = 1\). This linear system can be generalized for any row of \(T_k^m\) and \(R_k^m\) table. Suppose that there are \(j\) nonzero elements in row \(\alpha\) of given \(T_k^m\) table, then the linear system between this row and corresponding
$\alpha-2$ row in $R^m_k$ will have the same matrix form $M_1C = R$, where

$$
M_1 = \begin{bmatrix}
\alpha^2 - m^2 & 0 & 0 & 0 & 0 & 0 \\
2\alpha\beta & \alpha^2 - m^2 & 0 & 0 & 0 & 0 \\
\beta(\beta - 1) & 2\alpha(\beta + 1) & \ddots & 0 & 0 & 0 \\
0 & (\beta + 1)(\beta) & \ddots & \ddots & 0 & 0 \\
0 & 0 & \ddots & \ddots & \alpha^2 - m^2 & 0 \\
0 & 0 & 0 & \ddots & 2\alpha(\beta + j - 2) & \alpha^2 - m^2
\end{bmatrix}, \quad (2.40)
$$

and $\beta$ is the powers of $\ln(r)$ of the first element in the $\alpha$ row of $T^m_k$ table.

We can store the solutions of the linear system $M_1C = R$ for further numerical evaluation. However, it is not efficient to do so. A better way that consumes less space is to store the matrix $M_1$ and vector $R$. When evaluating $C$’s, the vector $R$ is substituted as numerical values.

Similarly, $(\frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{1}{r^2})(\frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{1}{r^2})$ maps $Cr^\alpha \ln(r)^\beta$ into

$$
\left(\frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{1}{r^2}\right) \left(\frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{1}{r^2}\right) (Cr^\alpha \ln(r)^\beta)
$$

$$
= (\alpha^2 - m^2)((\alpha - 2)^2 - m^2)r^{\alpha - 4} \ln^\beta r
+ [2(\alpha^2 - m^2)(\alpha - 2)\beta + 2\alpha\beta((\alpha - 2)^2 - m^2)]r^{\alpha - 4} \ln^{\beta - 1} r
+ [(\alpha^2 - m^2)\beta(\beta - 1) + 4\alpha(\alpha - 2)\beta(\beta - 1) + \beta(\beta - 1)((\alpha - 2)^2 - m^2)]r^{\alpha - 4} \ln^{\beta - 2} r
+ (4\alpha - 2)\beta(\beta - 1)(\beta - 2)r^{\alpha - 4} \ln^{\beta - 3} r
+ \beta(\beta - 1)(\beta - 2)(\beta - 3)r^{\alpha - 4} \ln^{\beta - 4} r
$$

(2.41)

Therefore, the row $\alpha$ in $\psi^m_k$ table is mapped into $\alpha-4$ row in the $S^m_k$ table. A similar linear system can be constructed as $M_2C = S$, where $C$ is the vector consisting of $j$ elements from row $\alpha$ in $\psi^m_k$, and $S$ is the corresponding row in
$S^m_k$ table. The matrix $M_2$ has the following form,

$$
M_2 = \begin{bmatrix}
 a_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 b_1 & a_2 & 0 & 0 & 0 & 0 & 0 \\
 c_1 & b_2 & a_3 & 0 & 0 & 0 & 0 \\
 d_1 & c_2 & b_3 & \cdot & \cdot & 0 & 0 & 0 \\
 e_1 & d_2 & c_3 & \cdot & \cdot & \cdot & 0 & 0 \\
 0 & e_2 & d_3 & \cdot & \cdot & \cdot & \cdot & 0 \\
 0 & 0 & e_3 & \cdot & \cdot & \cdot & \cdot & a_{j-1} \\
 0 & 0 & 0 & \cdot & \cdot & \cdot & b_{j-1} & a_j
\end{bmatrix}, \quad (2.42)
$$

and

$$
\begin{align*}
 a_i &= (\alpha^2 - m^2)((\alpha - 2)^2 - m^2), \\
b_i &= (\alpha^2 - m^2)2(\alpha - 2)(\beta + i - 1) + 2\alpha(\beta + i - 1)((\alpha - 2)^2 - m^2), \\
c_i &= (\alpha^2 - m^2)(\beta + i - 1)(\beta + i - 2) + 4\alpha(\alpha - 2)(\beta + i - 1)(\beta + i - 2) \\
&\quad + (\beta + i - 1)(\beta + i - 2)((\alpha - 2)^2 - m^2), \\
d_i &= (4\alpha - 2)(\beta + i - 1)(\beta + i - 2)(\beta + i - 3), \\
e_i &= (\beta + i - 1)(\beta + i - 2)(\beta + i - 3)(\beta + i - 4).
\end{align*}
$$

(2.43)

Here $i = 1, 2, \ldots, j$ and $\beta$ is the power of $\ln(r)$ in the first element in the $\alpha$ row of $\psi^m_k$ table.

With the help of the table structure, evaluating the inhomogeneous coefficients now turns into the process of solving a series of small linear system. We computed the condition number for all the matrices $M$'s up to 30th order. The average condition number is around 3.7 and maximum one is 10.4. Therefore, solving such linear systems numerically gives accurate solutions.

The new algorithm successfully computed series solution to the 30th order, which totally used about 24 hours and 14196.6MB memory.
2.4 Cost of computation

Despite the space management techniques being used here, the size of the solutions and the corresponding computation sequences grow very fast. For example, in the direct integration method, $T_8$ contains 715 terms with 67092 entries contained in the computation sequence for those terms; $\psi_8$ contains 496 terms with 19796 entries in its corresponding computation sequence. Demonstrated in the left graph of Figure 2.3, the growth in number of terms is $O(k^3)$. In the right graph of Figure 2.3, the number of entries in the computation sequence used in each order $T_k$ and $\psi_k$ have a growth rate of $O(k^6)$. In fact, the construction of the coefficients in $T_k$ and $\psi_k$ involves $O(k^6)$ operations. Each operation using exact rational arithmetic has a cost that depends on the length of the integers involved, but for this problem the growth is modest and at 10th order the longest integers are about 100 digits long. Therefore a solution truncated at order $N$, for example computing $T_N = \sum_{k=0}^{N} A^k T_k(r, \theta)$ has spatial complexity $O(1^3 + 2^3 + \cdots + N^6) = O(N^7)$.

The difference between the size of $k^{th}$ order solutions and the number of entries involved to compute them means the intermediate expressions during the $k^{th}$ order computation are much larger than the actual size of the same order solution. In fact, there are many terms that share the same monomial of $r^\alpha \ln^\beta(r)$ in the intermediate expression. If these terms can be condensed without the loss of information, the solution process could be much improved. The new algorithm using unknown coefficients is motivated by this intention.

In the new algorithm, solutions are constructed using the general form (2.33) and (2.34). The redundant computations in the first algorithm are eliminated since the coefficients $C$ of the solutions have one to one mappings with the coefficients $R$ in the corresponding right hand sides. Solving the $C$’s in the new algorithm does not involve the integration process which produces the redundant terms; instead it only requires the evaluation of several linear systems. For each $T^m_k$ or $\psi^m_k$ there are $2k$ lower triangular matrices that need

---

§It can be directly computed from the general form.
¶The growth of the computation sequence is obtained based on observation.
Figure 2.3: Left figure is the growth of number of terms in $T_k$ and $\psi_k$ ($O(k^3)$); the right one is the growth of number of entries in the computation sequence for $T_k$ and $\psi_k$ ($O(k^6)$).

to be solved, where the maximum size of the matrices is $k \times k$. The spatial cost of computing $T_k^m$ or $\psi_k^m$ is then $O(2k \times k^2)$ which is $O(k^3)$. Then a solution truncated at order $N$, has spatial complexity $O(1^3 + 2^3 + \cdots + N^3) = O(N^4)$. Compared to the complexity of the direct integration method which is $O(N^7)$, the new algorithm is seen to be much better.

2.5 The accuracy of the series solution

Both of the algorithms are based on the series expansion with respect to Rayleigh number $A$, where

$$T = \sum_{k=0}^{\infty} A^k T_k(r, \theta), \quad \psi = \sum_{k=1}^{\infty} A^k \psi_k(r, \theta).$$

One question is how accurate is the solution being computed. We need to identify the error on the $T_k$ and $\psi_k$. Further, given the coefficients being accurate, one also wants to know how well the series solutions represent the actual solutions.
Recall that in the above symbolic-numerical approaches, each $T_k$ and $\phi_k$ are computed or written down symbolically in first step. These symbolic solutions are exact, since they strictly satisfy the equations (2.11), (2.12), and corresponding boundary conditions. The round off errors are introduced during the evaluation on the unknown coefficients $K$’s and $C$’s. Since higher order solutions dependent on lower order ones, the error accumulates during the evaluation process.

![Figure 2.4: Left figure is the log plot of residuals on $T_k$ and $\phi_k$ with $R = 2$ $P = 0.02$ $k \leq 30$, the right one is the log of magnitude of $T_k$ and $\phi_k$ with same parameter.](image)

We use the residual of equation (2.11) and (2.12) where the numerical solutions are substituted in, to estimate the numerical error. For $N$th order truncated solutions $\hat{T}_N$ and $\hat{\psi}_N$, the residuals are defined as

\[
\varepsilon_{\psi_k} := \nabla^4 \hat{\psi}_N - A \cdot L(\hat{T}_N) - \frac{1}{P \cdot r} \frac{\partial (\nabla^2 \hat{\psi}_N, \hat{\psi}_N)}{\partial (r, \theta)},
\]

\[
\varepsilon_{T_k} := \nabla^2 \hat{T}_N - \frac{1}{r} \frac{\partial (\hat{T}_N, \hat{\psi}_N)}{\partial (r, \theta)}.
\]

The size of the residual varies according to solution with different order, Prandtl number $P$, and radius ratio $R$. However they are all similar in size given the same digits of accuracy defined by reserved word `Digits` in `Maple`. 28
In the left graph of Figure 2.4, the log residual of $T_k$ and $\phi_k$ are plotted for the $R = 2, P = 0.02$ case. Note that the residuals are functions in $r$ and $\theta$, a local maximum of the residual is computed in the range of $1 \leq r \leq R$, $0 \leq \theta \leq \pi$. From Figure 2.4 we can see that the residuals are very small, and the errors are slowly adding up. In Figure 2.5, the residual of $T_{30}$ is computed with $r \in [1, 2]$ and $\theta \in [0, \pi]$. The residual oscillates within a bound when $\theta$ varies, but increases dramatically when $r$ increases. However, the maximum size of the residual is very small.

![Figure 2.5: Residual of $T_{30}$](image)

The residual analysis confirms the computation to be very accurate, but the series solution may not always represent the real solution. Traditionally, one more higher order solution is considered to be better, when higher-order terms add a small correction to the overall sum (Custer & Shaughnessy [8]). It is also quite common to truncate just before the smallest terms. However, we need to point out that the series solution may or may not converge in either case. We consider a series solution to be valid when $A$ is inside the radius of convergence $r_A$, controlled by nearest pole location. According to Darboux’s principle [6], given a series expansion of a meromorphic function, the radius of convergence
convergence is determined by the nearest pole in the complex plane. Therefore, the pole locations of the series solution needs to be identified.

A possible way of obtaining the pole locations is by Padé approximants [4]. Padé approximants provides a rational function approximation based on the series expansion. The denominator of the approximation gives the information on pole locations. However, since the numbers of the poles and their structures are not known, it is hard to distinguish between the poles and noise. Padé approximants also faces severe difficulty when a defect happens where a pole is accompanied by a nearby zero [1]. Unfortunately, based on the results from the Padé approximants, the computed series solutions has many defects (as shown in Figure 2.6).

The QD method is used here to locate the nearby poles of the series expansion with respect to the Rayleigh number $A$. The QD algorithm does not require any information on the poles a priori [13, 2, 9]. Unlike the Padé approximants, it provides a mechanism that extracts pole location from the series input. Further more, the defect of nearby pole and zero have no significant influence on QD method.

In order to demonstrate the performance of the QD algorithm on the defect, 10 input equations with a pole and a nearby zero are constructed as the input. The equations used to generate the series are as follows,

$$
\frac{e^x(x - 0.999 \cdot 10^{(-k)})}{x - 1 \cdot 10^{(-k)}}, \quad k = 1, 2, \ldots, 10, \quad (2.45)
$$

where $k$ controls how close the defect is from the origin, and $1 \cdot 10^{(-k)}$ is the exact location of the pole. The distance between the defects and origin decreases by $10^{(-1)}$ for each $k$. The noise $e^x$ is multiplied which doesn’t affect the nature of poles and zeros. Each of the equations in (2.45) is then expanded into Maclaurin series to the 10th order, which are the input of the QD algorithm. The computed pole locations and errors of the QD algorithm are presented in Table 2.3. Apparently, a defect near the expansion point has no impact on the ability of the QD algorithm to accurately locate the pole. In addition

---

30

---

The roots of the input series can also be computed if needed.
Table 2.3: Accuracy analysis on the examples with nearby defects

<table>
<thead>
<tr>
<th>k</th>
<th>QD output pole location</th>
<th>computational error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.010000053</td>
<td>0.53230320 \cdot 10^{-7}</td>
</tr>
<tr>
<td>2</td>
<td>0.001000001</td>
<td>0.55376742 \cdot 10^{-10}</td>
</tr>
<tr>
<td>3</td>
<td>0.00010000000</td>
<td>0.55593272 \cdot 10^{-13}</td>
</tr>
<tr>
<td>4</td>
<td>0.000010000000</td>
<td>0.55614944 \cdot 10^{-16}</td>
</tr>
<tr>
<td>5</td>
<td>0.1000000000 \cdot 10^{-5}</td>
<td>0.55617112 \cdot 10^{-19}</td>
</tr>
<tr>
<td>6</td>
<td>0.1000000000 \cdot 10^{-6}</td>
<td>0.55617329 \cdot 10^{-22}</td>
</tr>
<tr>
<td>7</td>
<td>0.1000000000 \cdot 10^{-7}</td>
<td>0.55617350 \cdot 10^{-25}</td>
</tr>
<tr>
<td>8</td>
<td>0.1000000000 \cdot 10^{-8}</td>
<td>0.55617352 \cdot 10^{-28}</td>
</tr>
<tr>
<td>9</td>
<td>0.1000000000 \cdot 10^{-9}</td>
<td>0.55617353 \cdot 10^{-31}</td>
</tr>
<tr>
<td>10</td>
<td>0.1000000000 \cdot 10^{-10}</td>
<td>0.55617353 \cdot 10^{-34}</td>
</tr>
</tbody>
</table>

The QD algorithm is very accurate even when the radius of convergence of the series expansion is almost zero. It has been shown [13] that the accuracy of the QD algorithm will only suffer when there are multiple poles in the same location or share the same moduli, for example: an essential singularity. In this case, the essential singularities can be mapped to nonessential singularities by logarithmic derivative.

Figure 2.6: The singularites and zeros of $R = 2, P = 0.02$ case (left), and $R = 2, P = 0.7$ case (right) in the complex plane

The QD algorithm requires a series input. Both the series solution of temperature $T$ and stream equation $\psi$ in power series of $A$ can be used. We
can also use the equivalent conductivity, which is independent of \( r \) as the input. It has been shown by Mack & Bishop [18], the equivalent conductivity

\[
k_{eq} = 1 - A^2 \left[ \ln(R) \frac{\partial T_2}{\partial r} \bigg|_{r=1} \right] - A^4 \left[ \ln(R) \frac{\partial T_4}{\partial r} \bigg|_{r=1} \right] - \cdots ,
\]

(2.46)

consists of the only terms that contributes to the overall heat transfer. The three different input series gives similar results as expected. We chose the series solution of \( T \) as the input where \( \theta \) and \( r \) are chosen randomly from the interval \( \theta \in [0, \pi] \) and \( r \in [1, R] \). The distance between the nearest pole and the origin is presented in Table 2.4, where Prandtl number is set to \( p = 0.02 \), \( p = 0.7 \) and \( p = 7 \) meaning mercury, air and water respectively. The corresponding radius ratio \( 2/(R-1) \) changes from 2 to 12. Comparing to the example of Mack & Bishop [18] where \( R = 2, P = 0.02 \), the QD algorithm returns 125 as the nearest conjugate pair of poles around the origin \( (A = 0) \). This is smaller than the 170 using only the second order series solution in Mack & Bishop’s estimation.

Nearest pole location are increasing with a increased radius ratio \( 2/(R-1) \). This means the solution can be evaluated using larger Rayleigh number in the narrow cylinders cases. Since the radius of convergence is estimated and the residual analysis shows some good accuracy, we are confident in the series

<table>
<thead>
<tr>
<th>( 2/(R-1) )</th>
<th>( P = 0.02 )</th>
<th>( P = 0.7 )</th>
<th>( P = 7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>125</td>
<td>2272</td>
<td>2880</td>
</tr>
<tr>
<td>3</td>
<td>441</td>
<td>5570</td>
<td>9433</td>
</tr>
<tr>
<td>4</td>
<td>1142</td>
<td>21193</td>
<td>39545</td>
</tr>
<tr>
<td>5</td>
<td>2460</td>
<td>69229</td>
<td>96243</td>
</tr>
<tr>
<td>6</td>
<td>4716</td>
<td>176510</td>
<td>196394</td>
</tr>
<tr>
<td>7</td>
<td>8367</td>
<td>239256</td>
<td>356767</td>
</tr>
<tr>
<td>8</td>
<td>14052</td>
<td>488791</td>
<td>597300</td>
</tr>
<tr>
<td>9</td>
<td>22599</td>
<td>904625</td>
<td>941098</td>
</tr>
<tr>
<td>10</td>
<td>34988</td>
<td>1163326</td>
<td>1414369</td>
</tr>
<tr>
<td>11</td>
<td>52315</td>
<td>1705722</td>
<td>2046395</td>
</tr>
<tr>
<td>12</td>
<td>75768</td>
<td>2418297</td>
<td>2869505</td>
</tr>
</tbody>
</table>

Table 2.4: Nearest pole locations of Rayleigh number \( A \), starting from the origin.
solutions within the radius. We now finally present some streamlines and isotherms that we believe to be highly accurate.

In Figure 2.7 where $R = 2$, $P = 0.7$, the Rayleigh number increased from 1750 to 2000. For this set of parameters, the nearest pole is a conjugate pair with a distance of 2272 from the origin. We trust the series solution for $T$ and

Figure 2.7: The stream contours of $R = 2, P = 0.7$ case
ψ with $A \leq 2000$. We observed that when $A$ is near the critical point around 1800, the one cell in the center starts to split into three. When these multi-cell emerged, saddle points formed in between each pair of nearby cells. At these saddle points fluids are static but unstable. When the Rayleigh number is further increased around 1950, counter rotating cells emerges in between the
three clockwise rotating cells near the saddle points.

2.5.1 Stability of the solution

A more important (and traditional) question: is the computed solution stable? If we perturb equation (2.1) into,

\[ \nabla^4 \psi = AL(T) + \frac{1}{P \cdot r} \frac{\partial (\nabla^2 \psi, \psi)}{\partial (r, \theta)} + \varepsilon f(r, \theta), \tag{2.47} \]

and \( T = \tilde{T}_0 + \varepsilon \tilde{T}_1, \psi = \tilde{\psi}_0 + \varepsilon \tilde{\psi}_1 \), then \( \tilde{T}_1 \) and \( \tilde{\psi}_1 \) can be computed by solving the following new system,

\[ \nabla^4 \tilde{\psi}_1 = A \left[ \frac{\partial L}{\partial T}(\tilde{T}_0) \right] + \frac{1}{P \cdot r} \left[ \frac{\partial (\nabla^2 \tilde{\psi}_0, \tilde{\psi}_1)}{\partial (r, \theta)} + \frac{\partial (\nabla^2 \tilde{\psi}_1, \tilde{\psi}_0)}{\partial (r, \theta)} \right] + \varepsilon f(r, \theta), \tag{2.48} \]

\[ \nabla^2 \tilde{T}_1 = \frac{1}{r} \left[ \frac{\partial (\tilde{T}_0, \tilde{\psi}_1)}{\partial (r, \theta)} + \frac{\partial (\tilde{T}_1, \tilde{\psi}_0)}{\partial (r, \theta)} \right], \tag{2.49} \]

This system is more complicated than the original system. A faster way is to change the basic solution \( T_0^0 \) to \( T_0^0 + \varepsilon f \) symbolically, for example \( f = (r - 1)(R - r)r^\alpha \ln(r)^\beta \) such that it still obeys the boundary conditions. Then we generate the solutions as \( T = T_0 + \varepsilon \Delta T \) and \( \psi = \psi_0 + \varepsilon \Delta \psi \). We substitute these solutions back to the original system and compute the residual containing \( \varepsilon \)

\[ \varepsilon \tilde{R}(r, \theta) = \nabla^4 \psi - AL(T) - \frac{1}{P \cdot r} \frac{\partial (\nabla^2 \psi, \psi)}{\partial (r, \theta)}, \tag{2.50} \]

\[ \varepsilon \tilde{S}(r, \theta) = \nabla^2 T - \frac{1}{r} \frac{\partial (T, \psi)}{\partial (r, \theta)}, \tag{2.51} \]

The solution is stable if \( \| \Delta T, \Delta \psi \| \approx \| \tilde{R}(r, \theta), \tilde{S}(r, \theta) \| \). It is ill-conditioned if \( \| \Delta T, \Delta \psi \| \gg \| \tilde{R}(r, \theta), \tilde{S}(r, \theta) \| \). In the case of \( A = 2000, R = 2 \) and \( P = 0.7 \), the maximum ratio of \( \frac{\| \Delta T, \Delta \psi \|}{\| \tilde{R}(r, \theta), \tilde{S}(r, \theta) \|} \) is around 0.07, which means the solution is stable.
2.6 Conclusions

Computation sequences and the discovery of a general form expression improve the algorithm so that we can compute very high order series solutions. Obviously the method used here is slower than numerical methods. Still, with such solutions, much more information can be extracted, such as the singularity structures, and therefore the radius of convergence. The accuracy and stability of the series solutions are also verified.

The series solutions computed here are very accurate inside the radius of convergence, which truly represent the exact solutions. Solutions at Rayleigh number larger than the radius of convergence are not to be trusted. This does not necessarily mean that we can not pass the radius of convergence. The technique of analytic continuation could be used to compute series solutions beyond the boundary. This is not quite straightforward, one will encounter many practical issues when programming using this technique. For example,
the algorithm takes many steps to go around a pole on the real axis especially when there are nearby poles on the complex plane. Moreover, we lose the economical general form expression, which is valid only near $A = 0$. On the plus side, one can still guarantee accuracy using analytic continuation by using small steps and computing the residual. If the overall errors can be controlled then much more interesting results could be discovered, such as going around poles using different paths, and a better understanding of the solution. This will be examined in future work.

Within the radius of convergence, the series solution computed here offers a useful reference for the existing results computed from pure numerical methods such as the finite difference method. Due to its high accuracy many details of the flows can be obtained. The high order solution give us the chance to observe the singularity structures more clearly, which could be used as an “map” for future work. Algorithms such as analytic continuation will certainly benefit from this.

Finally, the spatial management techniques, especially computation sequences can be applied to other similar problems which use perturbation series to describe the solutions. At the most simple, the same problem with different boundary conditions. We have only investigated the simplest of these. For flow in porous media, other boundary conditions are of interest. We remark that the technique of computation sequences is by no means restricted to this PDE, however.

**Bibliography**


Chapter 3

An application of regular chain theory to the study of limit cycles

3.1 Introduction

In the field of dynamical systems, an interesting topic is the study of the number of limit cycles of a given system. For example, Hilbert’s 16th problem asks for an upper bound of the number of limit cycles for the system

\[ \dot{x} = F(x, y), \quad \dot{y} = G(x, y), \]

(3.1)

where \( F(x, y) \) and \( G(x, y) \) are degree \( k \) polynomials of variables \( x \) and \( y \), with real coefficients. No results are established for generic cubic systems.

In the case of finding small-amplitude limit cycles bifurcating from an elementary center or a focus point based on focus value computation, the problem has been completely solved only for generic quadratic systems [3], which can have three limit cycles in the vicinity of such a singular point. For cubic systems, James and Llyod obtained [25] a formal construction, via symbolic computation, of a special cubic system with eight limit cycles. In [52], Yu and Corless showed the existence of nine limit cycles with the help of a numerical

\[ A \text{ version of this chapter has been accepted by the International Journal of Bifurcation and Chaos.} \]
Very recently, Lloyd and Pearson [32] claimed to be the first to obtain a formal construction, via symbolic computation, of a new cubic system with nine limit cycles. A key step of their derivation is to show that two bivariate polynomials $R_1$ and $R_2$ have real solutions. They found that the resultant of $R_1$ and $R_2$ had a real solution and then concluded that $R_1$ and $R_2$ would have a real common solution. This is not always true. In fact, the existence of a real solution of the resultant of two bivariate polynomials does not necessarily imply the existence of a common real solution for the original two polynomial equations. For example, given $R_1 = y^2 + x + 1$ and $R_2 = y^2 + 2x + 1$ with $x < y$, the resultant of $R_1$ and $R_2$ in $y$ is $x^2$, which has a real solution $x = 0$. However the two equations $R_1 = R_2 = 0$ actually do not have common real solutions. In addition, a similar flawed conclusion was made by the authors when they were claiming that the existence of real solutions for $R_1 = R_2 = 0$ was implying the existence of real solutions for a trivariate polynomial systems $\Psi_1 = \Psi_2 = \Psi_3 = 0$. Therefore, the proof given by Lloyd and Pearson in [32] is not complete. (For a more complete explanation, please refer to Appendix E.)

In the present paper, we formally prove that a specific cubic dynamical system has nine limit cycles. Our strategy is as follows. Given a cubic dynamical system, we reduce the fact that this system has (at least) nine limit cycles to testing whether a given semi-algebraic set is empty or not. This test is based on a symbolic procedure capable of producing an exact representation for each real solution of any system of polynomial equations and inequalities. Once one such real solution has been found, then this procedure can be halted and non-emptiness has been formally established. Therefore, our approach does not have the flaws of [32].

The symbolic computation of small limit cycles involves finding the common roots of a non-linear polynomial system consisting of $n$ focus values $v_0(\gamma_1, \ldots, \gamma_m), \ldots, v_{n-1}(\gamma_1, \ldots, \gamma_m)$, where the variables $\gamma_1, \ldots, \gamma_m$ are the parameters of the original system. With the help of algorithmic and software tools from symbolic computation, we are able to compute nine limit cycles.
symbolically, using the same system as that used by Yu & Corless [52]. Unlike the methods used in previous studies which usually depend on good choices of free parameters and the values of dependent parameters, the new method introduces a systematic procedure to symbolically find the maximum number of limit cycles for a given system. It also provides a symbolic proof on the existence of the computed number of limit cycles. In addition, center conditions may be obtained as a by-product.

Symbolic methods for studying and solving non-linear polynomial systems are of great interest due to their wide range of applications, for example, in theoretical physics, dynamical systems, biochemistry, to name a few. They are very powerful tools that surpass numerical methods by giving exact solutions, whether the number of solutions is finite or not, and by identifying which solutions have real coordinates.

There are two popular families of symbolic methods, based on different algebraic concepts: Gröbner bases [7, 4, 6], and regular chains [26, 46, 35, 2, 10]. Gröbner bases methods have gained much attention during the past four decades due to their simpler algebraic structure: the input polynomial system, say $F$, is replaced by another polynomial system, say $G$, such that both $F$ and $G$ have the same solution set and geometrical information (dimension, number of solutions) can easily be read from $G$.

Methods based on regular chains are relatively new, and have many advantages compared to Gröbner bases methods. For example, they tend to produce much smaller output [18, 11] in terms of number of monomials and size of coefficients. In addition, regular chain methods can proceed in an incremental manner, that is, by solving one equation after another, against the previously solved equations. This allows for more efficient implementation and makes the processing of inequality constraints much easier. These advantages will be further explained later in this paper.

Given a multivariate polynomial system $F$ in a polynomial ring, for example $\mathbb{Q}[x]$ over $\mathbb{Q}$, regular chains methods compute the algebraic variety (or zero set - the set of common complex solutions) of $F$ in the form of a list of finitely many polynomial sets. Each of these sets is a polynomial system in triangular
shape and with remarkable algebraic properties; for these reasons, it is called a regular chain. The algebraic variety of the input system $F$ is given by the union of the common complex roots of the output regular chains. The notion of a regular chain was introduced independently by Kalkbrener [26] and, by Yang and Zhang [46] as an enhancement for notion of a triangular set. Indeed, the regular chain is a special type of triangular set which avoids possible degenerate cases that lead to empty solution [11].

One of the main successes of the Computer Algebra community in the last 30 years is the discovery of algorithms, called modular methods, that allow to keep the swell of the intermediate expressions under control. Even better: with these methods, almost all intermediate (polynomial or matrix) coefficients fit in a machine word, making these methods competitive in terms of running time with numerical methods. Modular methods have been well developed for solving problems in linear algebra and for computing greatest common divisors (GCDs) of polynomials [43]. They extend the range of accessible problems that can be solved using exact algorithms. In the area of polynomial system solving, the development of those methods is quite recent. They have been applied to Gröbner bases [1, 42] and primitive element representations [23, 24]. Thanks to sharp size estimates [18], the application of modular methods to polynomial system solvers based on regular chains has been very successful in both practice and theory, see [19], opening the door to using fast polynomial arithmetic [28] and parallelism [36] in the implementation of those solvers. The modular method of [19] is available in the RegularChains package in Maple.

The rest of the paper is organized as follows. The advantages of incremental solving are further explained in the next section. The theory of regular chains and a modular method for solving polynomial systems by means of regular chains are presented in the third section, together with a number of examples and related Maple commands. The relationship of limit cycles and focus values is presented in the fourth section, with an example of focus value computation using a perturbation method. Then, in the fifth section, the regular chains method is applied to a generic quadratic system to show three small-amplitude limit cycles around the origin and to obtain center con-
ditions. Moreover, with a modular method based on regular chain theory, a special cubic system is presented to show nine small-amplitude limit cycles in the vicinity of the origin.

### 3.2 Incremental solving

The nature of the algebraic problem posed by this application to the study of dynamic systems and, more precisely, the study of limit cycles require that the supporting algebraic tools provide the following specifications and properties.

**Incremental solving of polynomial systems.** Given a polynomial system of equations, \( f_1 = \ldots = f_m = 0 \), one would like to solve one equation after another against the previously solved equations. To be more precise, we first choose a format for the solutions. Here we consider regular chains. Thus, we can assume that the common solutions of \( f_1, \ldots, f_j \), for \( 1 \leq j < m \), are given by finitely many regular chains \( T_1, \ldots, T_e \). Then the common solutions of \( f_1, \ldots, f_{j+1} \) are obtained by taking the union of the regular chains computed by executing a procedure called \texttt{Intersect} and applied to \( f_{j+1} \) and \( T_1, \ldots, T_e \) successively.

The advantages of this approach are numerous. First of all, from a theoretical point of view, if \( \{f_1, \ldots, f_m\} \) is a regular sequence, then incremental solving is known to be a very effective process \[27, 41, 11, 21\].

There are also practical reasons. For instance, information (such as dimension, existence of real solutions) may be extracted before completing the solving of the entire system \( f_1 = \ldots = f_m = 0 \).

**Incremental processing of inequality constraints.** Given a component of the solution set of a system of polynomial equations, one would like to extract from that component the points that satisfy an inequality constraint, either of the type \( f \neq 0 \) or of the type \( f > 0 \). For example, in the application to limit cycles, one requires the first several focus values vanish, \( v_0 = \cdots = v_{n-1} = 0 \), but the last one \( v_n \neq 0 \). Regular chains provide this facility \[9, 12\]. That is, for a component encoded by one or several regular chains, one can extract the points of that component that satisfy a given inequality constraint. Moreover, the output of this refinement process is again given by a special
flavor of regular chains, called regular semi-algebraic systems [8]. Therefore, incremental solving can also be used with inequality constraints.

**Practical efficiency.** With respect to other algebraic tools for describing solution sets of polynomial systems, regular chains have an advantage in terms of size [17]. In addition, there are sharp size estimates about the representation of the solutions of polynomial systems when this representation is done with regular chains. This is essential in order to design efficient algorithms to compute these representations.

Moreover, these efficient algorithms are able to take advantages of modular techniques. We use a standard example to introduce the principle of those techniques. Consider a square matrix \(A\) with integer entries and for which its determinant \(d\) is to be computed exactly. It is well-known that using multi-precision rational arithmetic will only solve examples of moderate size due to intermediate expression swell. Let \(B\) be a bound on the absolute value of \(d\) and let \(p_1, \ldots, p_s\) be prime numbers such that their product exceeds \(2B\) and each of these primes is of machine word size. One computes the determinant \(d_i\) of \(A\) modulo the prime number \(p_i\). Then, the determinant \(d\) is obtained by applying the Chinese remainder theorem (CRT) to the residues \(d_1, \ldots, d_s\) and the moduli \(p_1, \ldots, p_s\). This approach not only avoids intermediate expression swell, but it allows for using efficient algorithms over finite fields and efficient implementation techniques in fixed single precision. Last but not least, the complexity of this modular computation process is less than that of the direct approach for computing the determinant of \(A\) via Gaussian Elimination (or LU decomposition, etc.) [22].

The following example is introduced to demonstrate the idea of incremental solving. Given the system

\[
F = \begin{cases}
x, \\
x + y^2 - z^2, \\
y - z^3,
\end{cases}
\] (3.2)

we want to find the real common roots. The incremental solving algorithm
processes one additional equation at a time. So it takes the first equation $x = 0$ and find the real roots, in this case the whole $y$-$z$ plane (left graph of Fig. 1). In the second step, the next equation $x + y^2 - z^2$ is taken into computation to obtain the common roots $x = 0, y = \pm z$ (middle graph of Fig. 1). At the last step, $y - z^3$ is added to compute the final answer \{x = 0, y = 0, z = 0\}, \{x = 0, y = 1, z = 1\}, \{x = 0, y = -1, z = -1\} (right graph of Fig. 1).

![Figure 3.1: The incremental solving of (3.2)](image)

### 3.3 The regular chains method

Similarly to a linear system which can be transformed to a triangular system by Gaussian elimination, a non-linear polynomial system can be transformed into one or finitely many systems, such that each of them is in a triangular shape. Such a system is called a triangular set, in that the main (or leading) variables of different polynomials are distinct. The notion of a triangular set was introduced in [39, 44], with the purpose of representing and computing the set of the common zeros of a given polynomial system. Since a triangular set is already in triangular form, it is ready to be solved by evaluating the unknowns one after another using a back-substitution process, as for triangular linear
systems. For example, the system

\[ F = \begin{cases} 
  x_4^2 - 2x_3 + x_1, \\
  x_3^3 + 2x_2, \\
  x_2^2x_1 - 2x_1 + 3, \\
  2x_1^2 + x_1, 
\end{cases} \]  

(3.3)

with ordered variables \( x_1 < x_2 < x_2 < x_4 \), is a triangular set since the polynomials in it have distinct main variables, which are here \( x_4, x_3, x_2, x_1 \), respectively.

The backward solving process of a triangular set could sometimes lead to an empty solution set. In the above example, one solution of the last equation is \( x_1 = 0 \), which leads to no solution for \( x_2 \). To avoid such degenerate cases, the notion of a regular chain was introduced. A regular chain is a type of triangular set which guarantees the success of the backward solving process. Regular chains are constructed by the insight that every algebraic variety is uniquely represented by some generic points of their irreducible components [2]. These generic points are given by certain polynomial sets, called regular chains. The common complex roots of any given multivariate polynomial system can be described by some finite union of regular chains. Such a family of regular chains is called a triangular decomposition of the input system.

### 3.3.1 Some definitions and examples for triangular decomposition

Before demonstrating the regular chains method, some definitions are given, followed by illustrative examples. Throughout this section, let \( \mathbb{Q} \) denote the rational number field and \( \mathbb{C} \) the complex number field. Let \( \mathbb{Q}[x] \) denote the ring of polynomials over \( \mathbb{Q} \), with ordered variables \( x = x_1 < \cdots < x_n \). Let \( p \) be a polynomial of the polynomial ring \( \mathbb{Q}[x] \) and let \( F \subset \mathbb{Q}[x] \) be a finite subset. We denote by \( V(F) \) the algebraic variety defined by \( F \), that is, the set of points in \( \mathbb{C}^n \) which are common solutions of the polynomials of \( F \).
Definition 1. If the polynomial \( p \in \mathbb{Q}[x] \) is not a constant, then the greatest variable appearing in \( p \) is called the main variable (or leading variable) of \( p \), denoted by \( \text{mvar}(p) \). Furthermore, the leading coefficient and leading monomial of \( p \), regarded as a univariate polynomial in \( \text{mvar}(p) \), are called the initial and the rank of \( p \), denoted by \( \text{init}(p) \) and \( \text{rank}(p) \), respectively.

Example 1. Let \( p := (x_1 + 1)x_2^2 + 1 \in \mathbb{Q}[x_1, x_2] \), where \( x_1 < x_2 \). Then, \( \text{mvar}(p) = x_2 \), \( \text{init}(p) = x_1 + 1 \) and \( \text{rank}(p) = x_2^2 \).

Definition 2. Let \( T \subset \mathbb{Q}[x] \) be a triangular set, that is, a set of non-constant polynomials with pairwise distinct main variables. The quasi-component of \( T \), denoted by \( W(T) \), is the set of points in \( \mathbb{C}^n \) which vanish all polynomials in \( T \), but none of the initials of polynomials in \( T \). The minimal algebraic variety containing \( W(T) \), denoted by \( \overline{W(T)} \), is called the Zariski closure of \( W(T) \). Note that \( W(T) \) is a subset of \( V(T) \), but may not equal \( V(T) \).

Example 2. Consider the polynomial ring \( \mathbb{Q}[x, y, z] \), where \( x < y < z \). Then, the set \( T := \{ y - x, yz^2 - x \} \) is a triangular set. The quasi-component \( W(T) \) is \( \{ (x, y, z) \in \mathbb{C}^3 \mid x \neq 0, y = x, z^2 - 1 = 0 \} \). The Zariski closure \( \overline{W(T)} \) is \( \{ (x, y, z) \in \mathbb{C}^3 \mid y = x, z^2 - 1 = 0 \} \). The variety \( V(T) \) is \( \{ x = 0, y = 0 \} \cup W(T) \).

Definition 3. Let \( T \) be a triangular set. A polynomial \( p \) is said to be zero modulo \( T \) if \( W(T) \subseteq V(p) \) holds. A polynomial \( p \) is said to be regular modulo \( T \) if the dimension of the variety \( V(p) \cap \overline{W(T)} \) is strictly less than that of \( \overline{W(T)} \).

Example 3. Let \( T := \{ y - x, yz^2 - x \} \). The polynomial \( y - x \) is zero modulo \( T \) since we have \( W(T) \subseteq V(p) \). On the other hand, the polynomial \( z - x \) is regular modulo \( T \) since \( V(p) \cap \overline{W(T)} \) is the set of points \( \{ (x, y, z) \in \mathbb{C}^3 \mid x^2 - 1 = 0, y = x, z^2 - 1 = 0 \} \), whose dimension is zero, that is, less than the dimension of \( \overline{W(T)} \).

Definition 4. A triangular set \( T \subset \mathbb{Q}[x] \) is a regular chain if one of the following two condition holds:

\(^{\dagger}\)The dimension of the empty set is defined as \(-1\).
(i) $T$ is empty or consists of a single polynomial;

(ii) $T \setminus \{T_{\text{max}}\}$ is a regular chain, where $T_{\text{max}}$ is the polynomial in $T$ with largest main variable, and the initial of $T_{\text{max}}$ is regular modulo $T \setminus \{T_{\text{max}}\}$.

**Example 4.** The triangular set $T := \{y - x, yz^2 - x\}$ is a regular chain since $\{y - x\}$ is a regular chain and $y$ is regular modulo $\{y - x\}$.

**Definition 5.** Let $F \subset \mathbb{Q}[x]$ be finite, and $\mathcal{T} := \{T_1, \ldots, T_e\}$ be a finite set of regular chains of $\mathbb{Q}[x]$. We call $\mathcal{T}$ a triangular decomposition of $V(F)$ if we have $V(F) = \bigcup_{i=1}^e W(T_i)$. We denote by `Triangularize` a function for computing such decompositions.

**Example 5.** Let $F := \{y - x, yz^2 - x\}$, $T_1 := \{y - x, z^2 - 1\}$ and $T_2 := \{x, y\}$. Then, $\{T_1, T_2\}$ is a triangular decomposition of $V(F)$.

The corresponding MAPLE program is as follows:

```maple
with(RegularChains):
F:=[y-x, y*z^2-x];
R:=PolynomialRing([z, y, x]);
dec:=Triangularize(F, R, output=lazard);
map(Equations, dec, R);
```

which returns,

$$[[z-1, y-x], [z+1, y-x], [y, x]]$$

**Definition 6.** Let $T$ be a regular chain, and $p$ be a polynomial of $\mathbb{Q}[x]$. Let $\mathcal{T} := \{T_1, \ldots, T_e\}$ be a finite set of regular chains of $\mathbb{Q}[x]$. We call $\mathcal{T}$ a regular split of $T$ w.r.t. $p$ if (1) $W(T) = \bigcup_{i=1}^e W(T_i)$ and (2) the polynomial $p$ is either zero or regular modulo $T_i$, for $i = 1, \ldots, e$. We denote by `Regularize` a function for computing such decompositions.

**Example 6.** Let $p := z - 1$ and $T := \{y - x, yz^2 - x\}$. Let $T_1 := \{y - x, z + 1\}$ and $T_2 := \{y - x, z - 1\}$. Then $\{T_1, T_2\}$ is a regular split of $T$ w.r.t. $p$.

The MAPLE program for this example is given by,
with(ChainTools):
p:=z-1;
T := Chain([y-x, y*z^2-x], Empty(R), R);
reg, sing := op(Regularize(p, T, R));
map(Equations, reg, R);
map(Equations, sing, R);

which returns,

\[
[[z+1, y-x]]
[[z-1, y-x]]
\]

### 3.3.2 Triangular decomposition algorithm

In this section, we illustrate how to obtain a triangular decomposition of an input polynomial system.

Given an input set of polynomials \( F = [P_1, \ldots, P_m] \subset \mathbb{Q}[x] \), we would like to compute a triangular decomposition of \( V(F) \), that is, regular chains \( T_1, \ldots, T_e \subset \mathbb{Q}[x] \) such that we have \( V(F) = W(T_1) \cup \cdots \cup W(T_e) \). The algorithm presented here works in an incremental manner, that is, by solving one input equation after another, against the solutions of the previously solved equations. The core routine of this algorithm is denoted as \texttt{Intersect}. It takes a regular chain \( T \) and a polynomial \( P \) as input, and returns regular chains \( T_1, \ldots, T_e \), such that we have

\[
V(P) \cap W(T) \subseteq W(T_1) \cup \cdots \cup W(T_e) \subseteq V(p) \cap \overline{W(T)}. \tag{3.4}
\]

We choose a polynomial \( p_1 \) with minimum rank from \( F \) and remove it from \( F \). Then, it is intersected with the empty regular chain, and obtain the regular chain \( T \) as \( p_1 \) itself. Next, the polynomial \( p_2 \) with minimum rank from the remaining \( F \) is chosen and removed. Then, \( p_2 \) and the regular chain \( T \) are the input for \texttt{Intersect}, which returns a list of regular chains \( T_1, \cdots, T_e \) that satisfy (3.4). Further, \( p_3 \) with the minimum rank from the remaining input \( F \) is intersected with each \( T_i, i \in 1, \cdots, e \), and will give more regular chains.
which also satisfy (3.4). The algorithm will go on until $F$ is empty. A more
detailed description of the algorithm can be found in [11].

In order to illustrate this triangular decomposition process, we compute
the triangular decomposition of $V(F)$ for the following example. Let $F =
[p_1, p_2, p_3]$, where

$$
p_1 := z + y + x^2 - 1,
\quad p_2 := z + y^2 + x - 1,
\quad p_3 := z^2 + y + x - 1,
$$

(3.5)

with a order $x < y < z$.

Firstly, $p_1$ is picked and removed from $F$ as the lowest rank polynomial
within the three polynomials, and then is a regular chain $T_0 = p_1$ by definition.

Secondly, $p_2$ with the lowest rank is chosen from the remaining two polyno-
mials. Now $p_2$ and $T_0$ are the input of Intersect, which computes $V(z + y + x^2 - 1, z + y^2 + x - 1)$. The procedure Intersect works as follows. By computing the
resultant of $z + y + x^2 - 1$ and $z + y^2 + x - 1$, $z$ is eliminated and we obtain a bivariate polynomial $(y - x)(y + x - 1)$. Then $T_1 := \{(y - x)(y + x - 1), z + y + x^2 - 1\}$
is a regular chain\(^4\), with $V(z + y + x^2 - 1, z + y^2 + x - 1) = W(T_1)$. Since
the GCD of $z + y + x^2 - 1$ and $z + y^2 + x - 1$ modulo $(y - x)(y + x - 1)$ is $z + y + x^2 - 1$, which is obtained by MAPLE’s command RegularGcd. Note that
$(y - x)(y + x - 1)$ has two factors. By factorizing it\(^5\), we obtain two regular chains $T_{11} := \{y - x, z + y + x^2 - 1\}$ and $T_{12} := \{y + x - 1, z + y + x^2 - 1\}$
such that we have $V(z + y + x^2 - 1, z + y^2 + x - 1) = W(T_{11}) \cup W(T_{12})$.

In the third step, the variety $V(p_1, p_2, p_3)$ is finally computed. This is
equivalent to computing the union of $V(p_3) \cap W(T_{11})$ and $V(p_3) \cap W(T_{12})$.

Let us consider how to compute $V(p_3) \cap W(T_{11})$. To this end, we first
compute the resultant of $z^2 + y + x - 1$ and $z + y + x^2 - 1$ and obtain

\[
\text{resultant}(z^2 + y + x - 1, z + y + x^2 - 1, z) = (y + x^2 + x - 1)(y + x^2 - x).
\]

We then compute the resultant of $(y + x^2 + x - 1)(y + x^2 - x)$ and $y - x$, and obtain

\(^4\)For this particular regular chain, one can check that $W(T_1) = V(T_1)$. But this does not
always hold unless the regular chain is zero-dimensional.

\(^5\)Irreducible factorization over $\mathbb{Q}$ is not necessary for computing triangular decomposition.
However, factorization often helps to improve the practical efficiency of polynomial system
solvers based on triangular decomposition.
resultant((y + x^2 + x - 1)(y + x^2 - x), y - x, x) = (x^2 + 2x - 1)x^2. Since the GCD of \((y + x^2 + x - 1)(y + x^2 - x)\) and \(y - x \mod (x^2 + 2x - 1)x^2\) is \(y - x\), and the GCD of \(z^2 + y + x - 1\) and \(z + y + x^2 - 1 \mod \{(x^2 + 2x - 1)x^2, y - x\}\) is \(z + y + x^2 - 1\), we know that \(V(p_3) \cap W(T_{11})\) is the union of zero sets of \(\{x^2 + 2x - 1, y - x, z + y + x^2 - 1\}\) and \(\{x, y - x, z + y + x^2 - 1\}\), which could be further simplified as \(\{x^2 + 2x - 1, y - x, z - x\}\) and \(\{x, y, z - 1\}\).

Similarly, \(V(p_3) \cap W(T_{12})\) can be decomposed into a union of zero sets of two regular chains \(\{x, y - 1, z\}\) and \(\{x - 1, y, z\}\).

To summarize, we have the following triangular decomposition to represent the zero set of \(F\):

\[
\begin{align*}
  z - x &= 0, \\
  y - x &= 0, \\
  x^2 + 2x - 1 &= 0.
\end{align*}
\]

3.3.3 A method based on modular techniques for computing triangular decomposition

For challenging input polynomial systems, the method described in the previous section may require vast amounts of computing resources (time and space). This situation can be improved in a spectacular manner by means of so-called modular techniques, which, broadly speaking, means computing by homomorphic images instead of computing directly in the original polynomial ring. We present below such an improvement for the case of input zero-dimensional systems whose coefficients are in \(\mathbb{Q}\).

Let \(F = \{p_1, \ldots, p_n\} \subset \mathbb{Q}[x]\). Recall that \(x\) stands for \(n\) ordered variables \(x_1 < \cdots < x_n\). We assume that the variety \(V(F)\) is finite and that the Jacobian matrix of \(F\) is invertible at any point of \(V(F)\). This latter assumption allows the use of Hensel lifting techniques. The algorithm proposed in [19] computes a triangular decomposition of \(V(F)\) via the following two-step process:

1. For some prime number \(\wp\), compute a triangular decomposition of \(V(F \mod \wp)\),
2. Apply Hensel lifting to recover a triangular decomposition of $V(F)$ from that of $V(F \mod \wp)$.

Some precautions need to be taken before the algorithm produces correct answers. In fact, extraneous factorizations or recombinations could occur when working modulo some “unlucky” prime numbers. Since the same input system $F$ could admit different triangular decompositions, it is possible that a regular chain obtained modulo $\wp$ does not match the modular image of any regular chains in a triangular decomposition $T_1, \ldots, T_e$ of $V(F)$. In [19], the following example is considered. Let $F = [p_1, p_2]$ where $p_1 := 326x_1 - 10x_2^6 + 51x_2^5 + 17x_2^4 + 306x_2 + 102x_2^3 + 12$, with $x_1 < x_2$. We have the following triangular decomposition of $V(F)$, that is, over $\mathbb{Q}$:

$$T_1 = \begin{cases} x_1 - 1 = 0, \\ x_3^2 + 6 = 0, \\ x_2^3 + 2 = 0, \\ x_2^2 + x_1 = 0. \end{cases} \quad (3.7)$$

Computing the regular chains that describe $V(F \mod 7)$ yields

$$t_1 = \begin{cases} x_2^2 + 6x_2x_1^2 + 2x_2 + x_1 = 0, \\ x_1^3 + 6x_1^2 + 5x_1 + 2 = 0, \\ x_2 + 6 = 0, \\ x_1 + 6 = 0. \end{cases} \quad (3.8)$$

which are not the images of $T_1, T_2$ modulo 7. In order to overcome this difficulty, the notion of *equiprojectable decomposition* was introduced in [19].

For a given ordering of the coordinates, the equiprojectable decomposition of a zero-dimensional (that is, with finitely many points) variety $V$ is a canonical decomposition of $V$ into components, each of which being the zero set of a regular chain. This notion can be defined as follows. Consider the projection $\pi := V \subset \mathbb{A}^n(\bar{k}) \rightarrow \mathbb{A}^{n-1}(\bar{k})$ that forgets the last coordinate, say $x$. We define $N(\alpha) := \#\pi^{-1}(\pi(\alpha)), \quad \alpha \in V$, that is, the number of the points that share the same coordinate with $\alpha$ in the $x$-axis.

The variety $V$ is split into $V_1, \ldots, V_d$ such that each $V_i, i = 1, \ldots, d$, consists of the point $\beta \in V$ such that $N(\beta) = i$. Then, a similar decomposition process is applied to each $V_i$ by considering the second last coordinate. Continuing in this manner yields a partition of $C_1 \cup \cdots \cup C_d = V$, which is a equiprojectable
decomposition. The key point is that each equiprojectable component $C_j$ is the zero set of a regular chain $T_j$, which can be made unique by requiring that each of its initials is equal to one. Together, those regular chains $T_1, \ldots, T_d$ form now a canonical triangular decomposition of $V$.

In the last example, the triangular decomposition, $t_1, t_2$ of $V(F \mod 7)$, is not an equiprojectable decomposition, as shown in the left graph of Fig. 2, since for the points which share the same $x_1$ coordinate, only the left and middle columns have the same number of points (which is two), while the right column has three points. So the decomposition is rearranged such that the left and middle columns are represented by one regular chain $t'_2$, and the last column is another regular chain $t'_1$ (the right graph of Fig. 2). One can use the MAPLE’s procedure EquiprojectableDecomposition to compute the regular chains $t'_1, t'_2$ from $t_1, t_2$, and thus to obtain the equiprojectable decomposition of the input system.

$$t'_1 = \begin{cases} x_1 - 1 = 0, \\ x_2^3 + 6 = 0 \end{cases}, \quad t'_2 = \begin{cases} x_2^2 + 2 = 0, \\ x_2^2 + x_1 = 0 \end{cases}. \quad (3.9)$$

It is obvious that $t'_1, t'_2$ are equal to $T_1, T_2 \mod 7$.

![Fig. 2: Equiprojectable decomposition](image)

Now the modular triangular decomposition will only be lifted after the equiprojectable decomposition is applied. Another key feature of this approach based on modular techniques is the size of the prime number $\wp$. The following theorem provides an approach for selecting good primes so as to avoid unlucky reductions.

**Definition 7.** The height of a non-zero number $a \in \mathbb{Z}$, is $H(a) := \log(|a|)$. For a rational number $P/Q \in \mathbb{Q}$, $\gcd(P, Q) = 1$, the height is $\max(H(P), H(Q))$. 

54
Finally, the height of a polynomial system \( F \in \mathbb{Z}[x_1, \ldots, x_m] \) is the maximum height of a non-zero coefficient in a polynomial of \( F \).

**Theorem 1** (Theorem 1 in [19]). Let \( F = [p_1, \ldots, p_m] \subset \mathbb{Q}[x] \) where each polynomial has degree at most \( d \) and height at most \( h \), Let \( T = T_1, \ldots, T_e \) be the equiprojectable decomposition of \( V(F) \). There exists an \( A \in \mathbb{N} - \{0\} \), with \( H(A) \leq a(m, d, h) \), and, for \( m \geq 2 \),

\[
a(m, d, h) = 2m^2d^{2m+1}(3h + 7\log(m + 1) + 5m\log d + 10),
\]

such that, if a prime number \( \wp \) does not divide \( A \), then \( \wp \) cancels none of the denominators of the coefficients of \( T \), and the regular chains \( T_1, \ldots, T_e \) reduced mod \( \wp \) define the equiprojectable decomposition of \( V(F \mod \wp) \).

Therefore, the set of unlucky primes is finite. Moreover, one can always find a large enough \( \wp \) that guaranties the success of the modular algorithm sketched above.

Once the equiprojectable decomposition using some good prime \( \wp \) is computed, the result is ready to be lifted in the sense of Hensel lifting. According to Hensel’s lemma [20], a simple root \( r \) of a polynomial \( f \mod \wp^k \) can be lifted to root \( s \) of \( f \mod \wp^{k+m} \), which also holds in the multivariate case. Using this lemma, given a polynomial system \( F \), its modular triangular decomposition \( t = t_1, \ldots, t_e \) over \( V(F \mod \wp) \) is lifted to \( t^k = t_1^k, \ldots, t_e^k \), which is the triangular decomposition of \( V(F \mod \wp^{2k}) \) [40]. Then, rational reconstruction is used to recover the regular chains with coefficients in \( \mathbb{Q} \).

Here, a probabilistic method is implemented which uses two primes \( \wp_1, \wp_2 \) that satisfy the condition of Theorem 1. The use of a probabilistic algorithm is a very common technique to compute values modulo primes, and then reconstruct the result to integers or rationals. It is very useful when the deterministic bound is not available or, like in our case, very high. The algorithm usually terminates when the result does not change for several primes. The output could be incorrect, but the probability of such failure is very small and controllable. In MAPLE many procedures are implemented using, probabilis-
tic algorithms including the commands \texttt{Determinant}, \texttt{LinearSolve}, \texttt{CharacteristicPolynomial}, \texttt{Eigenvalues}, \texttt{resultant} etc.

In our case, the algorithm works as follows.

1. Compute the equiprojectable triangularizations $T$ and $U$ for $\varphi_1$ and $\varphi_2$, respectively.

2. Lift $T$ to $T^k = T_1^k, \ldots, T_e^k$ in $Z(F \bmod \varphi_1^{2k})$, where $k$ starts from 1.

3. $T^k$ is taken as the input of the rational reconstruction to obtain $N^k = N_1^k, \ldots, N_e^k$ over $\mathbb{Q}$.

4. The algorithm terminates if $N^k \bmod \varphi_2$ equals $U$, and $N^k$ is returned as the triangular decomposition of $F$ over $\mathbb{Q}$.

5. Otherwise, $k$ is incremented by 1 and computations resume from Step 2.

Assume that $N$ is the correct equiprojectable triangular decomposition of the input system $F$. The algorithm fails when $N^k \bmod \varphi_2$ equals $U$ (the modular image of $N^k$ w.r.t $\varphi_2$), but $N^k \neq N$. It is also possible that either one of $\varphi_1, \varphi_2$ divides $A$ or both, so $N^k$ modulo $\varphi_2$ may never agree with $N$ modulo $\varphi_2$. However, the choices of $\varphi_1, \varphi_2$ that lead to those bad cases are finite and controllable. See Theorem 2 in [19] for details. In MAPLE, the \texttt{Triangularize} command offers this modular method. With the option \texttt{probability} = \texttt{prob}, the algorithm applies the probabilistic approach using the input probability of success \texttt{prob}, which control the size of the prime numbers $\varphi_1, \varphi_2$.

### 3.3.4 Isolating real roots of a regular chain

In this section, we briefly review how to obtain the real roots of a regular chain. Let $T$ be a regular chain of $\mathbb{Q}[x_1 < \cdots < x_n]$. A Cartesian product of $n$ intervals is called a box of $\mathbb{Q}[x_1 < \cdots < x_n]$. Let $L$ be a list of boxes. We say $L$ isolates the real roots of $T$ if

- The boxes in $L$ are pairwise disjoint;
- Each real root of $T$ belongs to one element of $L$;
• Every element of $L$ contains a real root of $T$.

**Example 7.** Let $T := \{x^2 - 2, y^2 - x\}$. Then, the MAPLE output of a real root isolation of $T$ is as follows:

\[
\begin{align*}
\{ & 19 \\
\{ & y = [-19, 5/4] \\
\{ & 16 \\
\{ & y = [16, -5/4] \\
\{ & 16 \\
\{ & x = [-16, -16] \\
\{ & x = [16, 16] \\
\{ & 181 91 \\
\{ & 181 91 \\
\{ & x = [181 91, 181 91] \\
\{ & x = [181 91, 181 91] \\
\{ & 128 64 \\
\{ & 128 64 \\
\end{align*}
\]

There are several existing algorithms and implementations [33, 45, 16, 5] for isolating the real roots of regular chains. However, they all rely on Maple’s univariate real root isolation routine, which is not efficient enough for our particular problem. Instead, we adapt a hybrid routine. The univariate polynomial in the regular chain $T$ is isolated by a parallel and cache optimal Collins-Akritas algorithm implemented in Cilk++ [13]. The obtained intervals are used to isolate the rest of the polynomials in $T$ by a sleeve-polynomials like algorithm [16], implemented in MAPLE.

### 3.4 Limit cycle and focus value

In system (3.1), suppose that $F(x, y)$ and $G(x, y)$ contain $m$ parameters $\gamma_1, \ldots, \gamma_m$, and there is a Hopf critical point at the origin, then the normal form of the system can be written in polar form up to the $(2n + 1)$-th order as [47],

\[
\frac{dr}{dt} = r(v_0 + v_1 r^2 + v_2 r^4 + \cdots + v_n r^{2n}) ,
\]

\[
r \frac{d\theta}{dt} = r \left(1 + \frac{d\phi}{dt}\right) = r \left(1 + \omega + t_1 r^2 + t_2 r^4 + \cdots + t_n r^{2n}\right) ,
\]

where each $v_k, k = 0, 1, \ldots, n$ is the $k$th-order focus value of the origin. Note that there are only $r^{2k}$ ($k = 0, 1, \ldots, n$) terms, since the odd power terms
vanish. Each of the focus values $v_k$ is a polynomial of the parameters $\gamma_j$, $(j = 1, 2, \ldots, m)$ of the original system.

The small-amplitude limit cycles near the origin can be determined from the equation,

$$\frac{dr}{dt} = 0 = r(v_0 + v_1 r^2 + v_2 r^4 + \cdots + v_n r^{2n}),$$  \hspace{1cm} (3.12)

then the right hand side of the equation (3.10) needs to be manipulated such that there are $n$ (and at most $n$) positive real roots for $r^2$.

Assuming the first $n+1$ focus values $v_0, v_1, \ldots, v_{n-1}, v_n$ are computed, we will find a combination of parameters such that the first $n$ focus values $v_0, v_1, \ldots, v_{n-1}$ all vanish except the $v_n$. This can generate at most $n$ limit cycles. Then, proper perturbations on the zeros of the $n$ focus values yields $n$ limit cycles. More precisely, a theorem on the relationship between the number of limit cycles and the focus values has been established in [51], which is given here for convenience.

**Theorem 2.** Suppose the origin is an elementary center of (3.1). If the first $n$ focus values associated with the origin depend on $n$ parameters $\{\gamma_j\}$, $j = 1, 2, \ldots, n$ such that

$$v_0 = v_1 = \cdots = v_{n-1} = 0, \quad v_n \neq 0,$$  \hspace{1cm} (3.13)

then there are at most $n$ small-amplitude limit cycles in the vicinity of the origin. Further suppose that $v_k(\Gamma)$, $k = 0, 1, \ldots, n - 1$, $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$, has some positive real solution $\Gamma = C$, $C = \{c_1, \ldots, c_n\}$ such that $v_k(C) = 0$ and the following condition holds,

$$\det \left[ \frac{\partial (v_0, v_1, \ldots, v_{n-1})}{\partial (\gamma_1, \gamma_2, \ldots, \gamma_n)} \right]_{\Gamma = C} \neq 0,$$  \hspace{1cm} (3.14)

then there are exactly $n$ small-amplitude limit cycles around the origin.

Accordingly, in order to compute $n$ small limit cycles near the origin, one
needs to find the common roots of a multivariate polynomial system:

\[ v_0(\gamma_1, \ldots, \gamma_n) = \cdots = v_{n-1}(\gamma_1, \ldots, \gamma_n) = 0, \quad (3.15) \]

where the variables \( \gamma_1, \ldots, \gamma_n \) are parameters of the original system. Once the common roots of \( v_0, \ldots, v_{n-1} \) are computed, the next focus value \( v_n \) will be evaluated at these roots. If some of the common roots does not make \( v_n \) vanish, then this set of roots will lead to \( n \) limit cycles, given their Jacobian to be non-zero. Otherwise, the common roots leading to \( v_n = 0 \) will be the candidate conditions for the origin to be a center.

There are many commonly used methods to compute focus values, including the perturbation method based on multiple time scales\[48, 49, 47, 50, 38, 37\], the singular point method \[31, 29, 14, 15\], and Poincare-Takens method \[51\]. In this article, we apply the perturbation method to compute the focus values.

### 3.5 Application to limit cycle computation

In this section, we apply the results presented in previous sections to compute limit cycles bifurcating from an isolated singular point (the origin of the system). Without loss of generality, suppose system (3.1) has at most \( n \) limit cycles. Then the first \( n+1 \) focus values need to be computed. \( v_0, \ldots, v_{n-1} \) are taken as the input for the triangular decomposition and \( v_n \) is used to verify if the output regular chains represent limit cycles. Two examples are given in this section. In the first example, we use the general quadratic system (D1) to illustrate how to use the regular chains method to find the limit cycle conditions and center conditions, respectively. It is actually a simple case where small limit cycles have already been thoroughly studied \[52\] using variable elimination method. The regular chains method computes all the possible common complex roots of the input system, and provides a systematical procedure of analyzing the properties of the outputs. If a regular chain \( T \) makes \( v_n \) vanish, then it is a candidate of center condition; if \( v_n \) does not vanish on
then it is a limit cycle condition. This can be checked by calling the built-in Maple procedure Regularize.

In the second example, we follow the work of [52] on a special cubic system that yields nine limit cycles with the help of numerical computation. Unlike the case of quadratic system, the existence of nine limit cycles for this cubic system has not been confirmed by purely symbolic algorithm. Due to the large input focus value system, the modular method based on regular chain theory is applied.

### 3.5.1 Generic quadratic system

Consider the general quadratic system [52], which is the system (3.23) truncated at 3rd-order terms,

\[
\begin{align*}
\dot{x} &= \alpha x + y + x^2 + (b + 2d)xy + cy^2, \\
\dot{y} &= -x + \alpha y + dx^2 + (e - 2)xy - dy^2,
\end{align*}
\]

where \( \alpha, b, c, d \) and \( e \) are independent parameters. It has been proved [3] that this system has three small-amplitude limit cycles near the origin. \( \alpha \) is set to zero to make the zero-order focus value \( v_0 = 0 \), then the rest focus values up to \( v_4 \) are obtained using the perturbations method,

\[
\begin{align*}
v_1 &= -(1/8)b(c + 1) \\
v_2 &= -(1/288)(c + 1)(20bc^2 + 19bce - 18bc + 30dce + 18b + 5b^3 + 3be + 56d^2b - 6de^2 \\
&\quad -be^2 + 34b^2d + 30de) \\
v_3 &= -(1/663552)(c + 1)(112800dec^2 - 33564bec^2 + 68944b^2d^2c + 1054be^3c + 10224dc^2c^2 \\
&\quad +151200dec + 4746bc^2c + 52320dce + 238080d^3ec - 140062d^2c^2 + 7776dec^3 \\
&\quad +26409be^2c^2 + 104160dec^3c + 71500be^3c + 98304bd^2c + 1764be^2c + 130176bd^2e \\
&\quad -15568bd^2c^2 + 22510b^3ec + 36288b^2de + 250112bd^2c^2 - 82464b^2dc + 267136bd^2ec \\
&\quad +126464b^2dce + 87156b + 88344bc^2 - 1071be^2 - 30132be + 292608d^2b - 99792dce^2 \\
&\quad +142560de + 118800b^2d - 82128bc^3 - 35526b^3c - 37248d^3c^2 + 27640b^4d + 127536b^3d^2 \\
&\quad -94b^3c^2 + 222208bd^4 - 1968de^4 - 83be^4 + 270208b^2d^3 + 47566b^4c^2 + 7985b^3c^2 \\
&\quad +1110be^3 + 7014b^3e + 238080d^3c + 24096de^3 + 40176bc + 4473b^3 + 2293b^5),
\end{align*}
\]
The existence of three small-amplitude limit cycles requires that the focus values $v_0$, $v_1$, $v_2$ vanish, while $v_3 \neq 0$ [51]. Since $v_0$ is already zero, the triangular
decomposition of $v_1$ and $v_2$ gives the following regular chains.

$$c + 1 = 0, \quad \begin{cases} d = 0, \\ b = 0, \end{cases} \quad \begin{cases} e = 0, \\ b = 0, \end{cases} \quad \begin{cases} e - 5c - 5 = 0, \\ b = 0. \end{cases} \quad (3.19)$$

Note that these regular chains represent the common roots of $v_1$ and $v_2$. They are candidates of center conditions or the conditions for the existence of three limit cycles, depending on whether $v_3$ vanishes on them or not. In this case, it is easy to check by directly substituting each regular chain into $v_3$. However, in a more general case with a large input system, regular chains obtained by triangular decomposition are not simple. It can not be substituted into higher-order focus values. Therefore, two different methods are introduced to verify the properties of the regular chains. The first method involves the triangular decomposition using one or few more higher-order focus values, while the second method uses the Regularize procedure to check whether the input regular chains make the next focus value vanish implicitly.

In the first method, another triangular decomposition using all three focus values $v_1, v_2$ and $v_3$ is conducted. The newly generated regular chains are then compared with the ones obtained using only $v_1$ and $v_2$. The triangular decomposition of $v_1, v_2$ and $v_3$ gives the new regular chains,

$$c + 1 = 0, \quad \begin{cases} d = 0, \\ b = 0, \end{cases} \quad \begin{cases} e = 0, \\ b = 0, \end{cases} \quad \begin{cases} d^2 + 2c^2 + c = 0, \\ e - 5c - 5 = 0, \\ b = 0. \end{cases} \quad (3.20)$$

Comparing with the regular chains in (3.19) generated from $v_1$ and $v_2$, the first three regular chains $\{c + 1 = 0\}, \{d = 0, b = 0\}, \{e = 0, b = 0\}$ are identical. This indicates that on these three regular chains $v_3$ vanishes as well, therefore they are center conditions. Now consider the fourth regular chain, $d^2 + 2c^2 + c$ must also be zero in order to make $v_3$ vanishes on $\{e - 5c - 5 = 0, b = 0\}$. Therefore $\{e - 5c - 5 = 0, b = 0, d^2 + 2c^2 + c \neq 0\}$ is a condition for the existence of three limit cycles, while $\{e - 5c - 5 = 0, b = 0, d^2 + 2c^2 + c = 0\}$ is a possible center condition.
To further verify the result, one can conduct the triangular decomposition with one additional focus value $v_4$, which yields,

$$
c + 1 = 0, \quad \begin{cases} d = 0, \\ b = 0,\quad \begin{cases} e = 0, \\ b = 0,\quad \begin{cases} \frac{d^2}{2} + 2c^2 + c = 0, \\ e - 5c - 5 = 0, \\ b = 0. \end{cases} \end{cases} \end{cases} \tag{3.21}$$

These are exactly the same regular chains as that given in (3.20). So $v_4$ vanishes on the regular chain \{ $e - 5c - 5 = 0, b = 0, d^2 + 2c^2 + c = 0$ \}, which confirms that it is a center condition.

The advantage of this method is easy to see how the results are verified. However, the triangular decomposition computation with additional higher-order focus values could be very heavy, and sometimes impossible to compute. Therefore, we introduce another method which is less illustrative but computationally efficient.

The second method uses the built-in MAPLE procedure Regularize. Recall from Example 6, Regularize takes a polynomial $p$ and a regular chain $T$ as input, in this case the polynomial is $v_3$, and $T$ is chosen from (3.19). It returns two lists. The first one consists of the regular chain $T_r$ such that $p$ is regular modulo $T_r$. The second list consists of the regular chain $T_z$ such that $p$ is zero (or singular) modulo $T_z$. If the first list is empty, then $p$ is zero modulo the input regular chain $T$, implying that $T$ will make $v_3$ vanish. If the second list is empty, then $p$ is regular modulo $T$, which implies that this regular chain will make $p \neq 0$.

After the triangular decomposition of $v_1$ and $v_2$ the regular chains in (3.19) are then used to regularize $v_3$. The Regularize process shows that for the first three regular chains in (3.19), the first output list is empty, implying that the first three regular chains make $v_3$ vanish. For the last regular chain, the second output of the Regularize procedure is empty, indicating that the last regular chain makes $v_3 \neq 0$. One can also use Regularize on $v_4$ with respect to each regular chain in (3.19) as well to further verify, which gives exactly the same result as that obtained using the first method. Compared to the first method, the Regularize procedure takes much less time in computation. We shall apply
the Regularize method in the next subsection to compute nine limit cycles for a special cubic system.

### 3.5.2 A special cubic system

A general normalized cubic system with a fixed point at the origin has the form:

\[
\begin{align*}
\dot{x} &= a_{10}x + a_{101}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{30}x^3 + a_{21}x^2y + a_{12}xy^2 + a_{03}y^3, \\
\dot{y} &= b_{10}x + b_{101}y + b_{20}x^2 + b_{11}xy + b_{02}y^2 + b_{30}x^3 + b_{21}x^2y + b_{12}xy^2 + b_{03}y^3,
\end{align*}
\]  

(3.22)

where \(a_{ij}\)'s and \(b_{ij}\)'s are parameters. According to [52], the system can be simplified into

\[
\begin{align*}
\dot{x} &= \alpha x + y + ax^2 + (b + 2d)xy + cy^2 + fx^3 + gx^2y + (h - 3p)xy^2 + ky^3, \\
\dot{y} &= -x + \alpha y + dx^2 + (e - 2a)xy - dy^2 + lx^3 + (m - h - 3f)x^2y + (n - g)xy^2 + py^3,
\end{align*}
\]  

(3.23)

where \(a\) can be an arbitrary nonzero constant, usually set to \(a = 1\) by a proper scaling.

It has been proved [30] that \(\alpha = b = d = e = h = n = m = 0\) is a center condition for the origin. In order to find nine limit cycles we need \(v_0 = \cdots = v_8 = 0\), but \(v_9 \neq 0\). We follow the set-up of [52] and set the following 5 parameters to be zero:

\[
\alpha = b = d = e = h = 0 .
\]  

(3.24)

By the perturbation method, eight focus values are computed, with \(v_1\) given by

\[
m/8 ,
\]  

(3.25)

which obviously indicates that \(m = 0\) to ensure \(v_1 = 0\). With this new condition, the second focus value \(v_2\) becomes

\[
-1/8fn + 1/8pn .
\]  

(3.26)
Note that \( nf \) is a factor in \( v_3 \) and all higher-order focus values. This indicates that either \( n = 0 \), leading to the center condition \([30]\), or a new candidate condition for center: \( \alpha = b = d = e = h = m = f = p = 0 \). So, in the following, we assume \( nf \neq 0 \). Thus, the only choice of making \( v_2 = 0 \) for existence of limit cycles is

\[
p = f .
\]  
(3.27)

Under this condition, \( v_3 \) has the following form:

\[
-1/192 fn(3n + 15 \ell - 30c + 45 - 35c^2 + 15k) .
\]  
(3.28)

Since \( nf \neq 0 \), an easy choice of making \( v_3 \) vanish is

\[
n = -5 \ell + 10c - 15 + \frac{35}{3} c^2 - 5k .
\]  
(3.29)

Now there are 5 free parameters, \( c, k, \ell, f, g \),

(3.30)

remaining in the five focus values \( v_4, v_5, \ldots, v_8 \). Using the above results and removing the common factor \( nf \) and a constant factor in the resulting focus values we obtain

\[
v_4 = 648 - 162c - 516c^2 + 72\ell + 81k + 45g - 30gc - 434c^3 + 60c\ell + 54ck - 168c^4 + 56c^2\ell - 24k^2 - 6gk - 7c^2g - 6g\ell - 30k\ell - 6\ell^2 - 21kc^2 ,
\]  
(3.31)

\[
v_5 = 231336 - 265836c^3k + 37350kc^2\ell + 6174c^2g\ell - 4428gk\ell + 1764c^2gk - 66204gc
- 184098c^2\ell + 40392gk - 133182c^2g + 25002g\ell + 74610k\ell - 361344kc^2 + 270c^2\ell^2 - 14448c^4\ell - 101871kc^4 - 1944k\ell^2 - 7506k^2\ell + 24165k^2c^2 - 13587c^4g - 1575g^2c^2 - 540g^2\ell - 864f^2k - 13860f^2c^2 - 864f^2\ell - 540g^2k - 3618gk^2 - 810g\ell^2 - 34296c^3\ell - 156888ck - 135828c\ell - 4590g^2c + 360c\ell^2 + 40104ck\ell + 6912gce - 3348gck - 41580c^3g - 11880f^2c + 38394ck^2 - 497556c^2 + 655080c^3 + 548132c^4 + 60525k^2 + 16110c^2 + 187306c^5 - 270c^3 + 54152c^6 - 5832k^3 + 6885g^2 + 17820f^2 - 607122c + 115398\ell + 363339k + 131625g ,
\]  
(3.32)
\[ v_0 = 323074872 g k c^2 \ell + 46434531132 c^3 k - 10614656412 k c^2 \ell - 4323518316 c^2 g \ell \\
+ 1747144728 g k t - 8537169420 c^2 g k + 477367776 k c^2 \ell + 242856468 k c^2 \ell^2 \\
+ 512185086 k c^4 \ell - 4214700 g c^2 \ell^2 - 103297626 g c^4 \ell - 762314922 g c^4 \\
- 34795656 g k t^2 - 93514176 g k c^2 - 234557856 g k c^2 + 191130624 f c^2 \ell \\
- 80777088 f^2 k k - 26967924 g k c^2 - 33543720 g^2 k k + 29212704 g^2 c^2 \ell \\
- 428849856 f^2 k c^2 - 189314496 f^2 g c^2 - 18942336 f^2 g k - 18942336 f^2 g k^2 \\
+ 3496808634 c^5 k - 12158345106 g c + 494477136 g c k k - 564839272 c^2 \ell \\
+ 6530829606 k g - 8063653761 c^2 g + 2727654102 g k + 5077228878 k c^2 \ell \\
- 14369006205 k c^2 - 2308784724 c^2 \ell^2 + 11211047880 c^4 \ell + 26955499191 k c^4 \\
+ 847752156 k c^2 + 2178967392 k c^2 - 10546897392 k c^2 + 11692092699 c k c^4 g \\
- 1454945976 g c^2 c^2 + 324725760 g^2 c^2 + 1155995712 f^2 k - 320686872 f^2 c^2 \\
+ 492687460 f^2 \ell + 522334332 g^2 k + 1571957280 g k^2 + 409782564 g k^2 \\
+ 1168019685 k c^6 - 20942712 k c^6 + 499013568 k c^6 - 53343360 k c^6 \ell \\
- 22915872 k c^2 + 1728185544 k c^4 + 11043864 c^2 \ell^3 - 527082024 c^6 \ell \\
+ 144765594 c^4 \ell^2 - 26181792 g c^2 - 7361928 g c^2 - 29683332 g^2 c^4 \\
- 66407040 g k^3 + 512530473 g c^6 - 6788520 g c^6 - 11975040 f^2 \ell^2 \\
- 287050824 f^2 c^4 - 68802048 f^2 k^2 - 15098076 g c^3 c - 3143448 g c^3 k \\
- 3143448 g c^3 - 617404032 k c^2 c - 44934652 c^2 c + 1785600194 c^2 c \\
- 422269027 k c^2 - 181543032 g c^3 c + 2704428702 g c^5 - 1408703616 f^2 c^3 \\
- 31447268118 c k - 38578680 g^2 c - 12575398716 c k - 909902808 g^2 c \\
+ 1301328 c^2 \ell^2 + 166743356 c^3 - 1886860656 c k - 1001160712 c k \ell \\
- 463425688 g k c + 233217792 f^2 c k + 337929408 g k c^2 + 1314575496 c k^3 \\
- 247755826 c k \ell + 365912640 c k^2 \ell - 561043368 g c^3 k + 150984 g c^2 k \\
+ 260877592 g c^3 k - 208987776 f^2 c k + 40376880 g^2 c k - 91362168 g^2 c k \\
- 251475840 f^2 g c + 14327069940 c^3 g + 377213760 f^2 g - 1438591104 f^2 c \\
- 7726593888 c^2 k^2 + 2289360096 + 111869219888 c^2 + 49162020900 c^2 \\
- 4045102440 c^4 + 7440988536 k c^2 + 1963517274 f^2 - 4687436278 c^5 \\
+ 17692680 \ell^3 - 19564392796 c^3 + 1527553728 k^3 + 1314588204 g^2 c^2 \\
+ 3474845508 f^2 - 369870578 c^7 - 47900160 k^4 + 176215256 c^8 \\
- 3470040 c^4 + 5786020 g^3 - 17873296866 c + 4874228136 c \\
+ 5523913665 k + 3624801597 g .
\]
The other 2 polynomials,
\[ v_7 = v_7(c, f, g, k, \ell), \quad v_8 = v_8(c, f, g, k, \ell), \]  
(3.34)
with degrees 10 and 12, are too large to be presented here. These five focus values are input to the triangular decomposition algorithm. To simplify the computing process, a better order was generated before the triangular decomposition (by using the built-in MAPLE procedure `SuggestVariableOrder`),
\[ f > g > \ell > k > c. \]  
(3.35)

According to the size of the input system, a sufficiently large prime,
\[ \varphi := 304166505300000047, \]  
(3.36)
with 2\(^{58}\) bits, is chosen to conduct the modular triangular decomposition. Note that the prime chosen here guarantees the success of modular algorithm.

The program was successfully executed to generate seven regular chains. In order to be lifted, they are mapped into two equiprojectable regular chains. The first one is omitted since it contains \( f = 0 \). The second regular chain is
\[
\begin{align*}
    f_2 + Q_1(c) + 109048982804251206, \\
g + Q_2(c) + 213759544982554218, \\
\ell + Q_3(c) + 212357665370487176, \\
k + Q_4(c) + 235643319065695752, \\
Q_5(c) + 249698644301675923,
\end{align*}
\]  
(3.37)
where \( Q_1(c), Q_2(c), \ldots, Q_5(c) \) are polynomials in \( c \) with order 425, 425, 425, 425 and 426, respectively. This regular chain is lifted using the same prime given in (3.36) to obtain,
\[
T = \begin{cases} 
R_1(c)f^2 + S_1(c) + P_1, \\
R_2(c)g + S_2(c) + P_2, \\
R_3(c)\ell + S_3(c) + P_3, \\
R_4(c)k + S_4(c) + P_4, \\
S_5(c) + P_5, 
\end{cases}
\]  
(3.38)
where \( R_1(c), \ldots, R_4(c), S_1(c), \ldots, S_5(c) \) are polynomials in \( c \), with order 426 in
$S_5(c)$ and 425 in the rest; $P_1, \ldots, P_5$ are big constant terms, and approximately equal to

$$\begin{align*}
P_1 & \approx 0.9531642255 \cdot 10^{2755}, \\
P_2 & \approx 0.6286620222 \cdot 10^{1432}, \\
P_3 & \approx 0.6286809511 \cdot 10^{1432}, \\
P_4 & \approx -0.2811943803 \cdot 10^{1428}, \\
P_5 & \approx -0.1285851059 \cdot 10^{517}.
\end{align*}$$

(3.39)

Since these constants are long, only their first 10 digits and their size are presented. In order to check if $v_9$ vanishes or not on the common roots of $T$, one can follow the quadratic example, and use the `Regularize` procedure. However, since $T$ is very large, we check this by the following steps instead. Firstly, we compute $T_p = T \mod \wp$, and check if $T_p$ is a regular chain which turns out to be true. Secondly, we take $v_9 \mod \wp$ and $T_p$ as the input for `Regularize`, and find out that $v_9 \mod \wp$ does not vanish on $T_p$. According to the specialization property of resultants [34] (or Theorem 4 in [12]), this is a sufficient condition for $v_9 \neq 0$ on $T$. Therefore, we have found the conditions such that $v_1 = v_2 = \cdots = v_8 = 0$ but $v_9 \neq 0$, indicating that there exist at most nine limit cycles. Note that one requirement during the lifting procedure is that the Jacobian to be nonzero, which satisfies the condition of Theorem 2. This implies that all the positive real roots of the second regular chain lead to nine limit cycles.

By isolating the real roots of the obtained regular chain, we found that it has 78 real roots. The computer outputs of the intervals for the first several ones are shown below:

$$\begin{align*}
f & = [-11/32, -41/128], \\
g & = [-93359084781/1073741824, \\
& \quad -186718169557/2147483648], \\
l & = [1244408533/67108864, 39821073059/2147483648], \\
k & = [64099524509/68719476736, 128199049023/137438953472], \\
c & = [-1217904753311153071896572523085646692445709943373514406698511/ \\
& \quad 6204186867199659306166505138577217441/3417579257473456131832034/ \\
& \quad 72987128338364327235770644431915266672515551561249024880036739/ \\
& \quad 3390985216, -3805952354097235334967678913464264591389284357304/ \\
& \quad 22325209328488138083959989353317703285580538045/10679935179604/ \\
& \quad 550411975108530847760573013522611783263849735208039111098628903/ \\
& \quad 20275011481043468288]].
\end{align*}$$

68
\[ f = [41/128, 11/32], \quad g = [-93359084781/1073741824, -186718169557/2147483648], \]
\[ l = [1244408533/67108864, 39821073059/2147483648], \]
\[ k = [64099524509/68719476736, 128199049023/137438953472], \]
\[ c = [-121790475331115301869572523085666692445709943373514406698511\]
   \quad 6204186867199659306166505138577217441/3417579257473456131832034\]
   \quad 72987128338336432723577064443191526657251551561249024880036739\]
   \quad 3390985216, -38059523540972353349676789134642645913892843573042\]
   \quad 232520932848138083959989353317703285580538045/106799351796045\]
   \quad 50411975108530847760573013522611783263849735208039110986289032\]
   \quad 02750114810434688288] \]

\[ f = [-19/4, -35/8], \quad g = [-5239003/262144, -83824045/4194304], \]
\[ l = [292265139/16777216, 1169060569/67108864], \]
\[ k = [-247962889/134217728, -991851547/536870912], \]
\[ c = [-115680680925314261355705483664130489454918902845080066745722072\]
   \quad 8245158022295965108754679631530185/579177248617/36695977858411\]
   \quad 441857731343248333910527450398266924979798014214301907660174157\]
   \quad 56929120296849762010984873984, -1446008511566428266946318545801\]
   \quad 631118186486285663508343215259103064475278699563859433495394127\]
   \quad 3197397156077/4586997231980143023221641790604173881593129978336\]
   \quad 562247475177678773845752176969616140037106220251373109248] \]

\[ \ldots \]

The total time used for the modular triangular decomposition is 1622615.24 sec (almost 19 days), on a computer with Intel(R) Core(TM)2 Quad CPU Q9550 @ 2.83GHz and 8G of memory. Isolating the real roots of the regular chain takes about nine hours in Maple on one node of a cluster. The node has 4 processors, each of which is a 12-core AMD Opteron(tm) 6168 @ 0.8GHz processor, and total memory of 250 GB.

To illustrate the critical focus values, we take one solution with 1000 sig-
significant figures (only the first 50 decimals are printed for convenience):

\[ \alpha = b = d = e = h = m = 0, \]
\[ p = f, \]
\[ n = -5\ell + 10c - 15 + \frac{35}{3}c^2 - 5k, \]
\[ c = -3.5636474286524271074464850122360152178067239603615\cdots, \]
\[ f = -0.33257083410940510824128708562052896225706851485676\cdots, \]
\[ g = -86.9474232009343774198056958113440830986000366046486\cdots, \]
\[ l = 18.543132142599506651625032427714327516815314466604\cdots, \]
\[ k = 0.93277084686805751726888595860136166253862306463035\cdots. \]

which yields the following approximations for critical focus values:

\[ v_0 = 0, \quad v_4 = -0.2628637706 \cdot 10^{-1088}, \quad v_9 = 0.9410263940 \cdot 10^{19}. \]
\[ v_1 = 0, \quad v_5 = -0.3957953881 \cdot 10^{-1078}, \]
\[ v_2 = 0, \quad v_6 = -0.5385553132 \cdot 10^{-1076}, \]
\[ v_3 = 0, \quad v_7 = -0.5135260069 \cdot 10^{-1074}, \]
\[ v_8 = -0.4251758871 \cdot 10^{-1072}, \]

and the determinant of the Jacobian matrix is \(-0.4633625957 \cdot 10^{1259}\). This clearly indicates the existence of nine limit cycles. By increasing the precision used to 2000 digits, the size of \(v_4, \ldots, v_8\) is reduced to \(\mathcal{O}(10^{-2000})\). These numbers are zero in actuality. By having constructed isolating intervals for the real root earlier, this was proved. The numerical computation here merely illustrates the computer-assisted proof given earlier.

### 3.6 Conclusion

Quantitative analysis of polynomial dynamical systems, such as determining the number of small-amplitude limit cycles around the origin, naturally leads to solve systems of multivariate polynomial equations and inequalities. Proving formally that such a semi-algebraic system is consistent, and, if it is, computing
all its solutions or a sample of them, are goals that make the use of symbolic and exact methods desirable.

In this paper, we have demonstrated that the theory of regular chains possesses powerful algorithmic tools to achieve those goals. We have applied to large input focus value systems an algorithm for computing triangular decompositions of polynomial systems via modular techniques. From these calculations, we have obtained conditions for the existence of limit cycles and potential center conditions. One example, in particular, exhibiting nine limit cycles shows the computational power and efficiency of these tools from regular chain theory.

These tools, available in the RegularChains library in MAPLE can be applied to solve other polynomial systems arising from real physical or engineering systems.

Bibliography


Chapter 4

Concluding Remarks

The works presented here are conducted with the belief that series expansions provide precise information for the given system around the expansion point. In Chapter 2 we demonstrated the use of series solution of the two dimensional heat transfer of some fluids between two horizontal concentric cylinders. The high order solutions do provide very accurate approximation to the actual solutions within the radius of convergence. The stability of the computed series solutions is also studied, which turns out be quite stable.

The radius of convergence of the computed solutions is determined by the nearest singularity according to Darboux’s principle. The singularity properties and their locations are determined also using the high order solutions. We utilized the QD algorithm to compute the nearby poles, and this algorithm is more reliable compared to other methods such as Padé approximants in our case. We find out that there is no evidence of essential singularities, but there are many cases the poles are accompanied by nearby zeros, which can cause the Padé approximant algorithm to be inaccurate.

In Chapter 3 we investigated the Hilbert’s 16th problem which asked the number of limit cycles of a system. We narrow the question to the case of small-amplitude limit cycles bifurcating from a center at the origin of the quadratic and cubic planar polynomial systems. We perturbed the system using multiple time scale method to compute focus values of the input systems. The real solutions of the focus values are possible limit cycle conditions. These focus values are multivariate polynomials of the parameters in the original system,
and they are very complicated both in size and degree. The modular regular chains approach is used to triangularize the polynomial system and give an input for the real root isolation procedure. Using the computed real roots we constructed the target system which process 9 small limit cycles near the origin.

In both works, large expression management is the key technique that makes the solutions computable. In the concentric cylinder problem, the computation sequences are applied to reduce the size of symbolic expressions. They are used here due to the observations that many terms of the intermediate expressions share the same monomials. By collecting the complicated coefficients and substituting them with new variables, the algorithm is able to compute some high order solution (16th order). The shape of the solutions is then recognized and applied in a new algorithm which reduces the intermediate expression even more. The computation sequences could be used in many other perturbation problems which involves symbolic computation of perturbation series. In the Hilbert’s 16th problem, symbolic computation of limit cycles is transformed into a solving process of dense multivariate polynomial systems. The modular technique reduces the size of the coefficients of each expression during the computation, without which it is impossible to solve such system symbolically. The modular approach is quite popular in recent years, and it is suitable for those systems whose complexity are mostly caused by the size of the coefficients.

**Future work**

In the concentric cylinder problem, the boundary conditions in our study are symmetric, with the inside cylinder being hotter than the outer one. It is worth investigating with changed boundary conditions, for example, the outside cylinder being hotter, or a non-uniformly distributed temperature on the boundaries. These new boundary conditions requires different Fourier transformations, but the series solutions will in some cases have similar forms to the ones presented in this work.
The series solutions of the concentric cylinder problem are only valid within the radius of convergence. This boundary could be extended using analytic continuation, which can go around the poles and reach the point that otherwise couldn’t. The observed singularity structures can be used as an guidance for the analytic continuation algorithm. A problem is the increased spatial complexity due to more complicated structure and steps. Another issue is how to find the optimal path through the “mine field” filled with poles, that maximizes the accuracy and minimizes the number of steps. This is a technique with great potential due to the fact that the solutions of most partial differential equations possess singularities.

The modular regular chains algorithm will be applied to similar cubic systems with more independent parameters which could have more small limit cycles. Since there are at most 12 such parameters in cubic systems, it is commonly believed that the maximum number of small limit cycles is 12. We will further optimize the focus value computation program and the modular regular chains algorithm to reach this target by symbolic computations.

Other systems in dynamical system problems that ask the number of limit cycles or require the solving of polynomial systems can also be studied using the modular regular chains approach. There are several systems available that we will try to solve using the this method.
Appendix

A Proof of the shape of the general form

In this Appendix, we give an inductive proof of the shape of the general solution of $T^m_k$. We rewrite (2.33), (2.34) and (2.11) here for convenience,

\[
T^m_k = \sum_{\alpha=-k/2}^{m/2} \sum_{\beta=0}^{1+2\alpha+k} C_{T^m_k,2\alpha,\beta} r^{2\alpha} \ln^\beta r + \sum_{\alpha=-m/2+1}^{m/2} \sum_{\beta=0}^{k-m/2+\alpha+1} C_{T^m_k,2\alpha,\beta} r^{2\alpha} \ln^\beta r \\
+ \sum_{\alpha=m/2+1}^{3k/2-1} \sum_{\beta=0}^{k+1} C_{T^m_k,2\alpha,\beta} r^{2\alpha} \ln^\beta r + \sum_{\alpha=m/2+1}^{3k/2-1} \sum_{\beta=0}^{k} C_{T^m_k,3k,\beta} r^{3k} \ln^\beta r ,
\]

\[
\psi^m_k = \sum_{\alpha=-k/2+1}^{-m/2-1} \sum_{\beta=0}^{2\alpha+k-1} C_{\psi^m_k,2\alpha,\beta} r^{2\alpha} \ln^\beta r + \sum_{\alpha=-m/2}^{m/2} \sum_{\beta=0}^{k-m/2+\alpha} C_{\psi^m_k,2\alpha,\beta} r^{2\alpha} \ln^\beta r \\
+ \sum_{\alpha=m/2+1}^{3k/2} \sum_{\beta=0}^{k} C_{\psi^m_k,2\alpha,\beta} r^{2\alpha} \ln^\beta r .
\]

\[
\nabla^2 T_k = \frac{1}{r} \sum_{j=0}^{k-1} \frac{\partial (T_j, \psi_{k-j})}{\partial (r, \theta)} , \quad k = 0, 1, 2, \ldots .
\]

One can easily verify that $T^1_1$ and $\psi^1_1$ satisfy (2.33) and (2.34). Recall the Fourier series

\[
T_k(r, \theta) = \sum_{m=0}^{k} T^m_k(r) \cos(m\theta) , \quad k = 0, 1, 2, \ldots , \quad (A1)
\]

\[
\psi_k(r, \theta) = \sum_{m=0}^{k} \psi^m_k(r) \sin(m\theta) , \quad k = 1, 2, 3, \ldots , \quad (A2)
\]

79
are substituted into (2.11), such that

\[ \nabla^2 \sum_{m=0}^{k} T^m_k(r) \cos(m\theta) \]

\[ = \frac{1}{r} \sum_{j=0}^{k-1} \left[ \sum_{p=0}^{j} \left( \frac{\partial T_j^p(r)}{\partial r} \cos(p \theta) \sum_{q=0}^{k-j} \psi_{k-j}^q(r) \cos(q \theta) + \sum_{p=0}^{j} \sum_{q=0}^{k-j} \frac{\partial \psi_{k-j}^q(r)}{\partial r} \sin(q \theta) \right) \right] \]

\[ = \frac{1}{r} \sum_{p=0}^{k-1} \sum_{q=0}^{k-j} \left( \frac{q \partial T_j^p(r)}{\partial r} \psi_{k-j}^q(r) [\cos(p - q) \theta + \cos(p + q) \theta] \right) \]

\[ + \frac{p}{2} T_j^p(r) \frac{\partial \psi_{k-j}^q(r)}{\partial r} [\cos(p - q) \theta - \cos(p + q) \theta] \]  \hspace{1cm} (A3)

yields

\[ \nabla^2 T^m_k = \frac{1}{r} \sum_{j=0}^{k-1} \sum_{p=0}^{j} \sum_{q=0}^{k-j} \left( \frac{q \partial T_j^p(r)}{\partial r} \psi_{k-j}^q(r) + \frac{p}{2} T_j^p(r) \frac{\partial \psi_{k-j}^q(r)}{\partial r} \right) \] \hspace{1cm} |_{p-q=m}, \hspace{1cm} (A4)

or

\[ \nabla^2 T^m_k = \frac{1}{r} \sum_{j=0}^{k-1} \sum_{p=0}^{j} \sum_{q=0}^{k-j} \left( \frac{q \partial T_j^p(r)}{\partial r} \psi_{k-j}^q(r) - \frac{p}{2} T_j^p(r) \frac{\partial \psi_{k-j}^q(r)}{\partial r} \right) \] \hspace{1cm} |_{p+q=m}, \hspace{1cm} (A5)

The table structure of \( T^m_k \) provides a guide line for the proof by induction. We will prove equation (2.33) for some \( k \) and \( m \) completely fits the Table A1 with the assumption that all solutions with smaller \( k \) and \( m \) satisfy (2.33) and (2.34). According to the shape of the table structure, the tables are separated into four parts (see Table A1), where the upper and middle part have a stair shape, the lower part is in a block shape and the last row has one less element compared to the lower part. In order to prove the general form of \( T^m_k \), we need to ensure both the left and right hand sides of (A4) and (A5) have the same monomials in \( r^\alpha \ln^\beta r \). In other words, if we put both sides into table form similar to Table A1, their shape must match. Firstly, we will find the general
Table A1: The non-zero component of the $T^m_k$ table. Inside the boundary, the $\times$ represents some non-zero coefficient $C^m_{T_k,\alpha,\beta}$, while outside the boundary all elements are zeros.

\begin{equation}
\nabla^2 T^m_k = \sum_{\alpha=-m/2}^{-m/2} \sum_{\beta=0}^{1+2\alpha+k} C^m_{T_k,2\alpha,\beta} \nabla^2 \left( r^{2\alpha} \ln^\beta r \right) \\
+ \sum_{\alpha=-m/2+1}^{-m/2+\alpha+1} \sum_{\beta=0}^{m/2} C^m_{T_k,2\alpha,\beta} \nabla^2 \left( r^{2\alpha} \ln^\beta r \right) \\
+ \sum_{\alpha=m/2+1}^{3k/2-1} \sum_{\beta=0}^{k+1} C^m_{T_k,2\alpha,\beta} \nabla^2 \left( r^{2\alpha} \ln^\beta r \right) \\
+ \sum_{\beta=0}^{3k/2} C^m_{T_k,3k,\beta} \nabla^2 \left( r^{3k} \ln^\beta r \right),
\end{equation}

(A6)

We name the four terms on the right hand side to be (1), (2), (3) and (4). Now we expand the first term of the right hand side (1),

\begin{align}
(1) &= \sum_{\alpha=-k/2}^{-m/2} \sum_{\beta=0}^{1+2\alpha+k} C^m_{T_k,2\alpha,\beta} \nabla^2 \left( r^{2\alpha} \ln^\beta r \right) \\
&= \sum_{\alpha=-k/2}^{-m/2} \sum_{\beta=0}^{1+2\alpha+k} C^m_{T_k,2\alpha,\beta} \left[ (4\alpha^2-m^2)r^{2\alpha-2} \ln^\beta r + 4\alpha r^{2\alpha-2} \ln^{\beta-1} r + \beta(\beta-1)r^{2\alpha-2} \ln^{\beta-2} r \right] \\
&= \sum_{\alpha=-k/2}^{-m/2} \sum_{\beta=0}^{1+2\alpha+k} C^m_{T_k,2\alpha,\beta} \left[ (4\alpha^2-m^2)r^{2\alpha-2} \ln^\beta r + 4\alpha r^{2\alpha-2} \ln^{\beta-1} r + \beta(\beta-1)r^{2\alpha-2} \ln^{\beta-2} r \right].
\end{align}

(A7)
\[
\begin{align*}
&= \sum_{\alpha=-k/2}^{m/2-1+2\alpha+k} \sum_{\beta=0}^{-m/2-1+2\alpha+k} C_{T_k^m,2\alpha,\beta}(4\alpha^2-m^2)r^{2\alpha-2} \ln^\beta r + \sum_{\alpha=-k/2}^{m/2-1+2\alpha+k} \sum_{\beta=1} \nabla_2 C_{T_k^m,2\alpha,\beta} 4\alpha \beta r^{2\alpha-2} \ln^{\beta-1} r \\
&+ \sum_{\alpha=-k/2}^{m/2-1+2\alpha+k} \sum_{\beta=2}^{1+2\alpha+k} C_{T_k^m,2\alpha,\beta}(\beta-1)r^{2\alpha-2} \ln^{\beta-2} r + \sum_{\beta=1}^{1+2\alpha+k} (4\alpha^2-m^2)r^{2\alpha-2} \ln^\beta r \\
&+ \sum_{\beta=1}^{1+2\alpha+k} 4\alpha \beta r^{2\alpha-2} \ln^{\beta-1} r \bigg|_{\alpha=-m/2} + \sum_{\beta=2}^{1+2\alpha+k} \beta(\beta-1)r^{2\alpha-2} \ln^{\beta-2} r \bigg|_{\alpha=-m/2} \\
&= \sum_{\alpha=-k/2}^{m/2-1+2\alpha+k} \sum_{\beta=0}^{-m/2-1+2\alpha+k} C_{T_k^m,2\alpha,\beta}(4\alpha^2-m^2)r^{2\alpha-2} \ln^\beta r + \sum_{\alpha=-k/2}^{m/2-1+2\alpha+k} \sum_{\beta=2}^{2\alpha+k-1} C_{T_k^m,2\alpha,\beta+2}(\beta+2)(\beta+1)r^{2\alpha-2} \ln^\beta r + \sum_{\beta=0}^{2\alpha+k} 4\alpha(\beta+1)r^{2\alpha-2} \ln^\beta r \\
&+ \sum_{\beta=0}^{2\alpha+k-1} (\beta+2)(\beta+1)r^{2\alpha-2} \ln^\beta r \bigg|_{\alpha=-m/2} \\
&= \sum_{\alpha=-k/2}^{m/2-1+2\alpha+k} \sum_{\beta=0}^{2\alpha+k} \tilde{C}_{T_k^m,2\alpha,\beta} r^{2\alpha-2} \ln^\beta r + \sum_{\beta=0}^{2\alpha+k} \tilde{C}_{T_k^m,2\alpha,\beta} r^{2\alpha-2} \ln^\beta r \bigg|_{\alpha=-m/2},
\end{align*}
\]

where \( \alpha \in \left[-k/2, -m/2\right] \) and

\[
\tilde{C}_{T_k^m,2\alpha,\beta} = \begin{cases} 
C_{T_k^m,2\alpha,\beta}(4\alpha^2-m^2) + C_{T_k^m,2\alpha,\beta+1}4\alpha(\beta+1) + C_{T_k^m,2\alpha,\beta+2}(\beta+2)(\beta+1) & \text{if } \beta \leq 2\alpha+k-1 \\
C_{T_k^m,2\alpha,\beta}(4\alpha^2-m^2) + C_{T_k^m,2\alpha,\beta+1}4\alpha(\beta+1) & \text{if } \beta = 2\alpha+k \\
C_{T_k^m,2\alpha,\beta}(4\alpha^2-m^2) & \text{if } \beta = 2\alpha+k+1.
\end{cases}
\]

We collect the “old” \( C \) and use \( \tilde{C} \) for the easiness of notations. Similarly, the rest of \( \nabla^2 T_k^m \) can be written down as

\[
2 = \sum_{\alpha=-m/2+1}^{m/2-1} \sum_{\beta=0}^{k-m/2+\alpha+1} \tilde{C}_{T_k^m,2\alpha,\beta} r^{2\alpha-2} \ln^\beta r + \sum_{\beta=0}^{k-m/2+\alpha} \tilde{C}_{T_k^m,2\alpha,\beta} r^{2\alpha-2} \ln^\beta r \bigg|_{\alpha=m/2},
\]

(A9)
where \( \alpha \in [-m/2 + 1, m/2] \) and

\[
\tilde{C}_{T^n,2\alpha,\beta} = \begin{cases} 
    C_{T^n,2\alpha,\beta}(4\alpha^2 - m^2) + C_{T^n,2\alpha,\beta+14\alpha(\beta+1) + C_{T^n,2\alpha,\beta+2}(\beta+1) & \text{if } \beta \leq k-m/2+\alpha+1 \\
    C_{T^n,2\alpha,\beta}(4\alpha^2 - m^2) + C_{T^n,2\alpha,\beta+14\alpha(\beta+1) & \text{if } \beta = k-m/2+\alpha \\
    C_{T^n,2\alpha,\beta}(4\alpha^2 - m^2) & \text{if } \beta = k-m/2+\alpha-1 .
\end{cases}
\]  
\tag{A10}

\[
\sum_{\alpha=m/2+1}^{3k/2-1} \sum_{\beta=0}^{k+1} \tilde{C}_{T^n,2\alpha,\beta} r^{2\alpha-2} \ln^\beta r , 
\tag{A11}
\]

where \( \alpha \in [m/2 + 1, 3k/2 - 1] \) and

\[
\tilde{C}_{T^n,2\alpha,\beta} = \begin{cases} 
    C_{T^n,2\alpha,\beta}(4\alpha^2 - m^2) + C_{T^n,2\alpha,\beta+14\alpha(\beta+1) + C_{T^n,2\alpha,\beta+2}(\beta+1) & \text{if } \beta \leq k+1 \\
    C_{T^n,2\alpha,\beta}(4\alpha^2 - m^2) + C_{T^n,2\alpha,\beta+14\alpha(\beta+1) & \text{if } \beta = k \\
    C_{T^n,2\alpha,\beta}(4\alpha^2 - m^2) & \text{if } \beta = k-1 .
\end{cases}
\]  
\tag{A12}

\[
\sum_{\beta=0}^{k} \tilde{C}_{T^n,2\alpha,\beta} r^{2\alpha-2} \ln^\beta r \bigg|_{\alpha=3k/2} , 
\tag{A13}
\]

where

\[
\tilde{C}_{T^n,2\alpha,\beta} = \begin{cases} 
    C_{T^n,2\alpha,\beta}(4\alpha^2 - m^2) + C_{T^n,2\alpha,\beta+14\alpha(\beta+1) + C_{T^n,2\alpha,\beta+2}(\beta+1) & \text{if } \beta \leq k \\
    C_{T^n,2\alpha,\beta}(4\alpha^2 - m^2) + C_{T^n,2\alpha,\beta+14\alpha(\beta+1) & \text{if } \beta = k-1 \\
    C_{T^n,2\alpha,\beta}(4\alpha^2 - m^2) & \text{if } \beta = k-2 .
\end{cases}
\]  
\tag{A14}

To summarize

\[
\nabla^2 T^n_k = \sum_{\alpha=-m/2+1}^{-m/2+1} \sum_{\beta=0}^{2\alpha+k} \tilde{C}_{T^n,2\alpha,\beta} r^{2\alpha-2} \ln^\beta r + \sum_{\beta=0}^{2\alpha+k} \tilde{C}_{T^n,2\alpha,\beta} r^{2\alpha-2} \ln^\beta r 
\bigg|_{\alpha=-m/2} 
+ \sum_{\alpha=-m/2+1}^{m/2-1} \sum_{\beta=0}^{k-m/2+\alpha+1} \tilde{C}_{T^n,2\alpha,\beta} r^{2\alpha-2} \ln^\beta r + \sum_{\beta=0}^{k-m/2+\alpha} \tilde{C}_{T^n,2\alpha,\beta} r^{2\alpha-2} \ln^\beta r 
\bigg|_{\alpha=m/2} 
+ \sum_{\alpha=m/2+1}^{3k/2-1} \sum_{\beta=0}^{k+1} \tilde{C}_{T^n,2\alpha,\beta} r^{2\alpha-2} \ln^\beta r + \sum_{\beta=0}^{k} \tilde{C}_{T^n,2\alpha,\beta} r^{2\alpha-2} \ln^\beta r 
\bigg|_{\alpha=3k/2} ,
\tag{A15}
\]
have a table form (see Table A2) similar to Table A1. The only difference is that the last rows of upper and middle part have one less element (the zeros in dashed cell). With the $\nabla^2 T_k^m$ expanded, we now consider the right hand sides of (A4) and (A5). We consider the general term $\frac{\partial T^p(r)}{\partial r} \psi^q_{k-j}(r)$ first.

$\frac{\partial T^p(r)}{\partial r}$ has the following general form,

$$
\frac{\partial T^p(r)}{\partial r} = \sum_{\alpha_1=-j/2}^{-p/2} \sum_{\beta_1=0}^{1+2\alpha_1+j} C_{T^p_j,2\alpha_1,\beta_1} r^{2\alpha_1-1} \ln^{\beta_1} r
$$

$$
+ \begin{cases}
\sum_{\alpha_1=-p/2+1}^{p/2} \sum_{\beta_1=0}^{j-p/2+\alpha_1+1} C_{T^p_j,2\alpha_1,\beta_1} r^{2\alpha_1-1} \ln^{\beta_1} r & \text{if } p \text{ is odd} \\
\sum_{\alpha_1=-p/2+1}^{p/2} \sum_{\beta_1=0}^{j-p/2+\alpha_1+1} C_{T^p_j,2\alpha_1,\beta_1} r^{2\alpha_1-1} \ln^{\beta_1} r & \text{if } p \text{ is even}
\end{cases}
$$

(A16)

$$
- C_{T^p_j,2\alpha_1,\beta_1} r^{2\alpha_1-1} \ln^{\beta_1} r \bigg|_{\alpha_1=0, \beta_1=j-p/2+1}
$$

$$
+ \sum_{\alpha_1=p/2+1}^{3/2j-1} \sum_{\beta_1=0}^{j+1} C_{T^p_j,2\alpha_1,\beta_1} r^{2\alpha_1-1} \ln^{\beta_1} r + \sum_{\beta_1=0}^{j} C_{T^p_j,2\alpha_1,\beta_1} r^{2\alpha_1-1} \ln^{\beta_1} r \bigg|_{\alpha_1=3j/2}
$$

Table A2: The $\nabla^2 T_k^m$ table in “stair” shape.
Here the $\overline{C}$ is used to distinguish from the $C$ used by $T_j^p(r)$. We name each term as (5), (6), (7) and (8). Now for $\psi_{k-j}^q(r)$ we have

$$\psi_{k-j}^q(r) = \sum_{\alpha_2=(k-j)/2+1}^{q/2-1} \sum_{\beta_2=0}^{2\alpha_2+(k-j)-1} C_{\psi_{k-j}^q,2\alpha_2,\beta_2} r^{2\alpha_2} \ln^{\beta_2} r$$

$$+ \sum_{\alpha_2=-q/2}^{q/2} \sum_{\beta_2=0}^{\alpha_2+(k-j)-q/2} C_{\psi_{k-j}^q,2\alpha_2,\beta_2} r^{2\alpha_2} \ln^{\beta_2} r$$

$$+ \sum_{\alpha_2=q/2+1}^{3/2(k-j)} \sum_{\beta_2=0}^{k-j} C_{\psi_{k-j}^q,2\alpha_2,\beta_2} r^{2\alpha_2} \ln^{\beta_2} r,$$

(A17)

and we name each term (9), (10) and (11). We will prove that the upper part of Table A2 can be constructed from

$$\frac{1}{r} \frac{\partial T_j^p(r)}{\partial r} \psi_{k-j}^q(r) = \frac{1}{r} (5 + 9 + 7 + 8) \times (9 + 10 + 11),$$

(A18)

but the boundary will never be passed. Naively, the first row of the upper part seems to be the product of $\frac{1}{r} \overline{C}$ and (9); however it is not the case. Actually, when $q < k - j - 2$, (9) does not vanish, since $\frac{1}{r} \overline{C} \times (9)$ is

$$\sum_{\alpha=-k/2+1}^{-m/2-1} \sum_{\beta=0}^{2\alpha+k} \overline{C}_{T_j^p,2\alpha_1,\beta_1} C_{\psi_{k-j}^q,2\alpha_2,\beta_2} r^{2\alpha-2} \ln^{\beta} r,$$

(A19)

where $\alpha = \alpha_1 + \alpha_2$ and $\beta = \beta_1 + \beta_2$. Comparing to the first term of (A15), $\frac{1}{r} \overline{C}$ and (9) do not have the very top row $\alpha = -k/2$ and the last element $\beta = 1 + 2\alpha + k$ of every row is missing. Therefore, when $q < k - j - 2$ the very top row of the product (A18) will never reach $-k/2$. Since $q$ and $k - j$ must have the same parity, then the only case left is $q = k - j$, where (9) does not vanish.

In order to reach the boundary of the upper part, we need to separate the very first row from the upper part. For the first row $\alpha = -k/2$ of the upper part, we take $q = k - j$. Since $q = k - j$, (9) vanishes, we take the first row of
\( (\alpha_1 = j/2) \) and consider \( \frac{1}{r} 5 \times 10. \)

\[
\begin{align*}
&= \sum_{\beta_1 = 0}^{1+2\alpha_1+j} \tilde{C}_{T_j, \alpha_1, \beta_1, r^{2\alpha_1-2} \ln{\beta_1}} \cdot \sum_{\alpha_2 = -q/2}^{q/2} \sum_{\beta_2 = 0}^{(k-j)-q/2+\alpha_2} C_{\psi_{\alpha_2, \beta_2, r^{2\alpha_2} \ln{\beta_2}}} \\
&= \sum_{\alpha_2 = -q/2}^{q/2} \sum_{\beta_2 = 0}^{(k-j)-q/2+\alpha_2} C_{T_j, -j, \alpha_2, \beta_2, r^{2\alpha_2-2} \ln{\beta_2}} \\
&= \frac{(k-j)-q/2+\alpha_2}{(q-j)/2} k/2 + \alpha \left( \sum_{\alpha_2 = -q/2}^{q/2} \sum_{\beta_2 = 0}^{(k-j)-q/2+\alpha_2} C_{T_j, -j, \alpha_2, \beta_2, r^{2\alpha_2-2} \ln{\beta_2}} \right)
\end{align*}
\]

where \( \alpha = \alpha_2 - j/2, \beta = \beta_2 + 1 \) and

\[
\tilde{C}_{T_j, \alpha, \beta} = \begin{cases} 
\tilde{C}_{T_j, -p, 0, \alpha_2, \beta, r^{2\alpha_2-2} \ln{\beta}} & \text{if } \beta = 0 \\
\tilde{C}_{T_j, -p, 0, \alpha_2, \beta, r^{2\alpha_2-2} \ln{\beta}} + \tilde{C}_{T_j, -p, 1, \alpha_2, \beta-1, r^{2\alpha_2-2} \ln{\beta}} & \text{if } 0 < \beta < k/2 + \alpha + 1 \\
\tilde{C}_{T_j, -p, 1, \alpha_2, \beta-1, r^{2\alpha_2-2} \ln{\beta}} & \text{if } \beta = k/2 + \alpha + 1.
\end{cases}
\]

We collect the \( \tilde{C} \) and use \( \tilde{C} \) for the easiness of notations. Observe the first row \( \alpha = -k/2 \),

\[
\begin{align*}
\sum_{\alpha = -k/2}^{(q-j)/2} \sum_{\beta_1 = 0}^{k/2 + \alpha + 1} \tilde{C}_{T_j, \alpha_2, \beta_2, r^{2\alpha_2-2} \ln{\beta}} &= \tilde{C}_{T_j, -k, 0, r^{-k-2} \ln{r} + \tilde{C}_{T_j, -k, 1, r^{-k-2} \ln{r}} (A22)
\end{align*}
\]
which matches the first row in the upper part of $\nabla^2 T^m_k$:

\[
\sum_{\alpha=-k/2}^{-m/2-1} \sum_{\beta=0}^{1+2\alpha+k} \tilde{C}_{T^m_\alpha,2\alpha,\beta} r^{2\alpha-2} \ln^2 r \bigg|_{\alpha=-k/2} = \tilde{C}_{T^p,-k,0} r^{-k-2} + \tilde{C}_{T^p,-k,1} r^{-k-2} \ln(r). \tag{A23}
\]

Note, in the $q = k - j$ case (A20), only the first row matches the one in the upper part of $\nabla^2 T^m_k$ and all other rows have less elements. In fact, $-k/2 \leq \alpha \leq k/2$ such that $\alpha + k/2 \geq 0$, the max element in each row is $\beta = k/2 + \alpha + 1 \leq 1 + 2\alpha + k - (k/2 + \alpha) \leq 1 + 2\alpha + k$. The equality holds when $\alpha + k/2 = 0$, which is the first row.

Now we need the find the “double step stair” shape boundary case for the upper part without the first row. A special case is needed, $q = k - j - 2$, where the first row of $\psi^{q}_{k-j}$ contains three elements, such that the product with the upper part of $T^p_j$ matches the target boundary. We want to maximize $j - p$ such that the upper part of $T^p_j$ contains the maximized number of rows. The largest $j$ that satisfies $q = k - j - 2$ is $q = 1, k - j = 3$ such that $j = k - 3$. The smallest $p$ is where we take $m = p + q$. In this case, the first row of $\psi^{q}_{k-j}$ is

\[
\sum_{\alpha_2=-q/2}^{q/2} \sum_{\beta_2=0}^{(k-j)-q/2+\alpha_2} C_{\psi^q_{k-j},2\alpha_2,\beta_2} r^{2\alpha_2} \ln^2 r \bigg|_{\alpha_2=-q/2=-1/2} = \sum_{\beta_2=0}^2 r^{-1} C_{\psi^{q}_{k-j},-1,\beta_2} \ln^2 r \tag{A24}
\]

Now the product of $\frac{1}{r^3}$ and the first row of $\psi^{q}_{k-j}$ is

\[
\frac{1}{r} \sum_{\alpha_1=-j/2}^{-p/2} \sum_{\beta_1=0}^{1+2\alpha_1+j} C_{T^p_{\alpha_1},2\alpha_1,\beta_1} r^{2\alpha_1-1} \ln^2 r \sum_{\beta_2=0}^2 r^{-1} C_{\psi^{q}_{k-j},-1,\beta_2} \ln^2 r = \sum_{\beta_1=0}^{-(m-1)/2} \sum_{\alpha_1=-j/2}^{-p/2} \sum_{\beta_1=0}^{1+2\alpha_1+j+2} C_{T^p_{\alpha_1},2\alpha_1,\beta_1} C_{\psi^q_{k-j},-1,\beta_2} r^{2\alpha_1-3} \ln^2 r \tag{A25}
\]

\[
= \sum_{\alpha=-j/2-1/2}^{-m/2} \sum_{\beta=0}^{1+2\alpha+k+3} \tilde{C}_{T^p_{\alpha_1},2\alpha_1,\beta_1} r^{2\alpha-2} \ln^2 r
\]

where $\alpha = \alpha_1 - 1/2$, $\beta = \beta_1$ and $\tilde{C}_{T^p_{\alpha_1},2\alpha_1,\beta_1} = \tilde{C}_{T^p_{\alpha_1},2\alpha_1,\beta_1} C_{\psi^q_{k-j},-1,\beta_2}$. This is
almost exactly the target upper part without the first row \( \alpha = -k/2 \),

\[
-m/2 - 1 + 2\alpha + k \sum_{\alpha = -k/2} \sum_{\beta = 0} T_{k}^{m,2\alpha,\beta} r^{2\alpha-2} \ln^{2} r \nabla_{r}, \tag{A26}
\]

except there is one more term \( \alpha = -m/2 \). Now we will show the last element of this last row is zero. Recall that when \( p + q = m \) on the right (A5) there may be a chance that the two terms cancel. In fact, take an arbitrary term of

\[
q \frac{\partial T_{j}^{p}(r)}{\partial r} \psi_{k-j}(r) - p T_{j}^{p}(r) \frac{\partial \psi_{k,j}(r)}{\partial r},
\]

\[ q T_{j}^{p,\alpha_{1,\beta_{1}} r^{\alpha_{1}\ln \beta_{1} r}} C_{\psi_{k-j},\alpha_{2,\beta_{2}} r^{\alpha_{2} \ln \beta_{2} r}} \ln \beta_{1} r - p C_{T_{j}^{p,\alpha_{1,\beta_{1}} r^{\alpha_{1} \ln \beta_{1} r}} C_{\psi_{k-j},\alpha_{2,\beta_{2}} r^{\alpha_{2} \ln \beta_{2} r}} r^{\alpha_{2} \ln \beta_{2} r}
\]

\[= q C_{1} C_{2} (\alpha_{1} r^{\alpha_{1} \ln \beta_{2} r} r + \beta_{1} r^{\alpha_{1} \ln \beta_{2} r}) r^{\alpha_{2} \ln \beta_{2} r}
\]

\[= p C_{1} C_{2} (\alpha_{2} r^{\alpha_{2} \ln \beta_{2} r} r + \beta_{2} r^{\alpha_{2} \ln \beta_{2} r})
\]

\[= q C_{1} C_{2} (\alpha_{1} r^{\alpha_{1} \ln \beta_{2} r} r + \beta_{1} r^{\alpha_{1} \ln \beta_{2} r})
\]

\[= p C_{1} C_{2} (\alpha_{2} r^{\alpha_{2} \ln \beta_{2} r} r + \beta_{2} r^{\alpha_{2} \ln \beta_{2} r})
\]

\[. \tag{A27}
\]

For \( \alpha_{1} r^{\alpha_{1} \ln \beta_{1} \ln \beta_{2} r} \) term to vanish one needs \( q \alpha_{1} = p \alpha_{2} \), and for \( \alpha_{1} r^{\alpha_{1} \ln \beta_{1} \ln \beta_{2} r} \) term to vanish one needs \( q \beta_{1} = p \beta_{2} \). Any element in this row except the last element will never give a zero in the target table, since it can easily be compensated from the derivatives of the next element. For the last element in the boundary case, if the \( \alpha_{1} r^{\alpha_{1} \ln \beta_{1} \ln \beta_{2} r} \) vanishes, it will reshape the boundary of the outcome table, since no higher order term exists. In the previous case we have \( p + q = m \), \( \alpha_{2} = -q/2 = -1/2 \) when \( \alpha = -m/2 \) which is the last row, \( \alpha_{1} = -m/2 + 1/2 = -p/2 - q/2 + 1/2 = -p/2 \) such that \( q \alpha_{1} = p \alpha_{2} \). This will make the last element vanish, so the product of \( r \) and the first row of \( \partial_{r} \) is

\[
-m/2 - 1 + 2\alpha + k \sum_{\alpha = -k/2+1} \sum_{\beta = 0} \tilde{T}_{k}^{m,2\alpha,\beta} r^{2\alpha-2} \ln^{2} r + \sum_{\alpha = -m/2} \sum_{\beta = 0} \tilde{T}_{k}^{m,2\alpha,\beta} r^{2\alpha-2} \ln^{2} r \nabla_{r}. \tag{A28}
\]

where the first component completely matches the target (A26) and the second component matches the shape of the last row of upper component. Now we have proved the upper part of right hand side of (A4) and (A5) must have the
shape in table A2. Similarly one can prove the middle and lower part. The same idea can be applied to the $\psi_k^m$ to verify the general form. □

B QD algorithm

Given a power series expansion of the meromorphic function $f(z) = \sum_{i=0}^{\infty} c_i z^i$, the QD algorithm relies on the computation of QD scheme,

\[
\begin{array}{cccc}
q_1^0 & e_1^0 \\
0 & 0 & q_1^1 & q_2^0 \\
0 & 0 & e_1^1 & e_2^0 \\
q_1^2 & q_2^1 & \ddots & \\
0 & e_1^2 & e_2^1 & \ddots \\
& \vdots & \vdots & \ddots \\
\end{array}
\] (B1)

where the alternate columns containing $q_m^n$ and $e_m^n$ are called Q-columns and E-columns respectively. Entries of the QD scheme are defined by

\[
q_m^n = \frac{H_{m+1}^n H_{m-1}^n}{H_m^n H_{m-1}^{n+1}},
\]
\[
e_m^n = \frac{H_{m+1}^n H_{m-1}^{n+1}}{H_m^n H_{m+1}^{n-1}},
\] (B2)

where the $H_m^n$ is the Hankel determinant of the input power series,

\[
H_m^n = \begin{vmatrix}
c_n & c_{n+1} & \cdots & c_{n+m-1} \\
c_{n+1} & c_{n+2} & \cdots & c_{n+m} \\
\vdots & \vdots & \ddots & \vdots \\
c_{n+m-1} & c_{n+m} & \cdots & c_{n+2m+2}
\end{vmatrix}.
\] (B3)
Suppose we have one and only one pole in the meromorphic function. Then
the radius of convergence is the absolute value of the pole $Z_S$, defined as
$|Z_S| = \lim_{n \to \infty} \left| \frac{c_n}{c_{n+1}} \right|$. In multiple pole situations, the relationship between
Hankel determinants and pole locations can be viewed as an extension of the
definition of radius of convergence in the case of one pole. Suppose there are
$m$ poles $z_1, z_2, \ldots, z_m$ with distinct moduli in the complex plane, then the
product of the reciprocal of each pole location is the ratios of corresponding
Hankel determinants when $n \to \infty$,

$$\prod_{i=1}^{m} u_i = \lim_{n \to \infty} \frac{H_{m+1}^{n+1}}{H_m^n}. \quad (B4)$$

Even though the $q$ and $e$ elements are defined using Hankel determinants,
the QD scheme is not computed this way, since the computation is very ineffi-
cient and ill-conditioned [7]. By manipulating (B2) one can have the following
rhombus rule,

$$q_{m+1}^n = \left( e_m^n - e_{m-1}^{n+1} \right) + q_m^n,$$
$$e_{m+1}^n = \frac{q_{m+1}^{n+1}}{q_m^n} e_m^n. \quad (B5)$$

The QD scheme is computed using these identities and the following initial
values,

$$q_m^{-m+1} = 0, \quad m = 2, 3, \ldots,$$
$$q_1^{-1} = \frac{c_1}{c_0},$$
$$e_m^{-m+1} = \frac{b_{m+1}}{b_m} m = 1, 2, \ldots. \quad (B6)$$

Here the $b$'s are the coefficients of the power series expansion of the reciprocal
of $f(z)$, $1/f(z) = b_0 + b_1 z + b_2 z^2 + \cdots$ computed also by recurrence relation.

For a meromorphic function $f(z)$ that contains $m$ poles, where

$$0 < |z_1| \leq |z_2| \leq \cdots \leq |z_m|, \quad (B7)$$

any $k$ such that $|z_k| < |z_{k+1}|$ is called a critical index. On the QD scheme,
any critical index will make the corresponding E-column go to zero. In other
words, if \( k \) is a critical index, then

\[
\lim_{n \to \infty} e^n_k = 0. \tag{B8}
\]

If all poles are distinct, every \( k \in \{1, 2, \ldots, m\} \) is a critical index, and each Q-column will converge to the reciprocal of the corresponding pole location,

\[
\lim_{n \to \infty} q^n_k = u_k = 1/z_k. \tag{B9}
\]

If there are poles with equal moduli, for example

\[
u_k = u_{k+1} = \cdots = u_{k+j}, \tag{B10}
\]

then the nearby critical index will be \( k \) and \( k+j \). The pole location information in between the \( q_k \) and \( q_{k+j} \) column are not that obvious. To extract the pole information in the equal moduli case, the Hadamard polynomial \([7, 1, 5, 6]\) is used,

\[
p^n_{m+1}(u) = u p^n_{m+1}(u) - q^n_{m+1} p^n_m(u), \quad p^n_0(u) = 1. \tag{B11}
\]

Given the Q-columns between any two nearby critical index, as in the example \((B10)\) the Hadamard polynomials have the following property:

\[
\lim_{n \to \infty} p^n_j(u) = \prod_{i=1}^{j} (u - u_i), \quad i = k, k+1, \ldots k+j. \tag{B12}
\]

Therefore, the locations of poles with same moduli can be computed by Hadamard polynomials. If we construct the matrix

\[
M^n_m = \begin{bmatrix}
g^n_1 & h^n_1 & 0 \\
1 & g^n_2 & h^n_2 \\
& 1 & g^n_3 & h^n_3 \\
& & \ddots & \ddots & \ddots \\
0 & & & 1 & g^n_m
\end{bmatrix}, \tag{B13}
\]

where \( g^n_{k+1} = q^n_{k+1} + e^n_{k+1}, h^n_m = q^n_m e^n_m \), then the Hadamard polynomial \( p^n_m \) is
the characteristic polynomial of the matrix $M_n^m$. Therefore, the eigenvalues of $M_n^m$ are the zeros of the Hadamard polynomial, which is also the location of the poles. Finding eigenvalues using, for example the QR algorithm is a stable method of finding these poles. The QD algorithm contains two parts

1. Construct the QD scheme using the input power series coefficients.

2. Identify the critical indexes and extract pole locations using the Hadamard matrices.

Note that the successful computation of pole locations in the multiple pole case depend on the number of higher order coefficients available. A more accurate estimations on the locations of the poles requires higher order input. But the higher order input does not guarantee the accuracy. The number of poles and their structures have great impact on the accuracy. For example, if the solution has many poles that either share the same location (multiple poles) or same moduli, then the accuracy of the QD algorithm suffers greatly [7]. If the output of the QD algorithm contains several poles near each other, then it may be case they are actually on the same location but just biased by the QD algorithm. Nevertheless, some high order input for example the 30th order in Chapter 2, is always appreciated.

\section*{C Perturbation method and multiple time scale algorithm}

There are many commonly used methods to compute focus values, including the perturbation method based on multiple time scales [14, 15, 13, 16, 12, 11], the singular point method [9, 8, 3, 4], and Poincaré-Takens method [17]. In this work, we apply the perturbation method. Suppose system (3.1) has an elementary center at the origin. It can always be written in the following form:

\begin{align}
\dot{x} &= y + F(x,y), \\
\dot{y} &= -x + G(x,y), \\
\end{align}

(C1)
where $F$ and $G$ are polynomials in $x$ and $y$, consisting of only non-linear terms. Space variables $x, y$ are expanded using $\varepsilon$:

$$
\begin{align*}
x &= \sum_{k=0}^{\infty} x_k \varepsilon^{k+1}, \\
y &= \sum_{k=0}^{\infty} y_k \varepsilon^{k+1},
\end{align*}
$$

\hspace{1cm} (C2)

where a scalar transformation $z \rightarrow \varepsilon z$ has been applied to $x$ and $y$ prior to the expansion. Time variable $t$ is replaced by an infinite sequence of time variables $T_k$, which is defined as:

$$
T_k = \varepsilon^k t, \quad k = 0, 1, 2, \ldots
$$

\hspace{1cm} (C3)

Now the derivative with respect to $t$ is expressed as a series expansion in powers of $\varepsilon$

$$
\frac{d}{dt} = \sum_{k=0}^{\infty} \varepsilon^k D_k,
$$

\hspace{1cm} (C4)

where $D_k = d/dT_k$. $T_k$’s are called multiple time scales. Substituting the series expansions for both space and time (C2), (C3) and (C4) into (C1) and balancing all the coefficients of like powers of $\varepsilon$, yields an infinite set of equations,

$$
\begin{align*}
D_0 x_k - y_k &= F_k, \quad k = 0, 1, \ldots \\
D_0 y_k + x_k &= G_k, \quad k = 0, 1, \ldots
\end{align*}
$$

\hspace{1cm} (C5, C6)

where $F_0 = G_0 = 0$, $F_k$ and $G_k$, $k > 0$ are determined by the terms found at lower orders and always in the form of

$$
C + \sum A_m r^m \cos(mT_0 + \phi(T_1, T_2, \ldots)) + B_m r^m \sin(mT_0 + \phi(T_1, T_2, \ldots)).
$$

\hspace{1cm} (C7)
The solutions of these differential equations are always solvable and computable. Applying $D_0$ on (C5) plus (C6) gives

$$D_0^2(x_k) + x_k = D_0(F_k) + G_k. \quad (C8)$$

Note that the right hand side of (C8) is still in the form of (C7). A particular solution can be obtained by the method of undetermined coefficients:

$$x_k = A_0^k + \sum_{j=1}^{k} \left[ A_j^k \cos(jT_0 + \phi(T_1, T_2, \ldots)) + B_j^k \sin(jT_0 + \phi(T_1, T_2, \ldots)) \right], \quad (C9)$$

where $A_k, k = 0, 1, \ldots$, and $B_k, k = 1, 2, \ldots$ are coefficients to be determined. This $x_k$ is substituted back into (C8) to determine these coefficients. In the solution, some terms are unbounded when $t \to \infty$, which are called secular terms. These terms need to be eliminated because the true solution is known to be bounded. To eliminate the secular terms the coefficients of $\cos(T_0 + \phi(T_1, T_2, \ldots))$ and $\sin(T_0 + \phi(T_1, T_2, \ldots))$ terms must be zero, which yields

$$D_k r + H(p_1, p_2, \ldots, p_m) = 0,$$
$$D_k \phi + I(p_1, p_2, \ldots, p_m) = 0, \quad (C10)$$

where $H$ and $I$ are polynomials in the parameters of the original system. Solving (C10) provides the focus values of each order $v_k = D_k r = -H(p_1, p_2, \ldots, p_m)$. Then, the normal form of the system, given in polar coordinates, can be rewritten in the form of:

$$\frac{dr}{dt} = \frac{\partial r}{\partial T_0} \frac{dT_0}{dt} + \frac{\partial r}{\partial T_1} \frac{dT_1}{dt} + \frac{\partial r}{\partial T_2} \frac{dT_2}{dt} + \cdots = D_0 r + D_1 r + D_2 r + \cdots, \quad (C11)$$

$$\frac{d\phi}{dt} = \frac{\partial \phi}{\partial T_0} \frac{dT_0}{dt} + \frac{\partial \phi}{\partial T_1} \frac{dT_1}{dt} + \frac{\partial \phi}{\partial T_2} \frac{dT_2}{dt} + \cdots = D_0 \phi + D_1 \phi + D_2 \phi + \cdots, \quad (C12)$$

94
where $D_1 r$ and $D_1 \phi$ are both uniquely determined by the parameters of the original system.

\section*{D An example of focus value computation}

The focus value algorithm using the perturbation method is better explained using an example. Consider the general quadratic system \cite{18}, which is the system (3.23) truncated at 3rd-order terms,

\begin{align*}
\dot{x} &= \alpha x + y + x^2 + (b + 2d)xy + cy^2, \\
\dot{y} &= -x + \alpha y + dx^2 + (e - 2)xy - dy^2,
\end{align*}

(D1)

where $\alpha, b, c, d$ and $e$ are independent parameters. It has been proved \cite{2} that this system has at most three small-amplitude limit cycles near the origin. This system is introduced here to illustrate the computation of focus values.

The space variable $x$ and $y$ are firstly scaled by $x \rightarrow \varepsilon x, \quad y \rightarrow \varepsilon y,$ and then expanded into (C2). While the time variable $t$ is expanded using scales $T_k = \varepsilon^k t, \quad k = 0, 1, 2, \ldots,$ and the derivative with respect to $t$ is then replaced by (C4). Once (D1) is expanded according to the previous space and time expansions, coefficients of the like powers of $\varepsilon$ are then collected, which gives infinitely many pairs of equations,

\begin{align*}
D_0 x_0 - y_0 &= 0, \\
D_0 y_0 + x_0 &= 0, \\
D_0 x_1 - y_1 + D_1 x_0 - x_0^2 - (b + 2d)x_0 y_0 - cy_0^2 &= 0, \\
D_0 y_1 + x_1 + D_1 y_0 + dx_0^2 + (e - 2)x_0 y_0 - dy_0^2 &= 0,
\end{align*}

(D2)

\begin{align*}
D_0 x_2 - y_2 + D_2 x_0 + D_1 x_1 - 2x_0 x_1 - (b + 2d)x_0 y_1 - (b + 2d)x_1 y_0 - 2cy_0 y_1 &= 0, \\
D_0 y_2 + x_2 + D_2 y_0 + D_1 y_1 + 2dx_0 x_1 + (e - 2)x_0 y_1 + (e - 2)x_1 y_0 - 2dy_0 y_1 &= 0.
\end{align*}

(D3)

(D4)

\ldots

95
Consider the first pair (D2). Applying $D_0$ on the first equation plus the second one will eliminate $y_k$ and give a second-order linear differential equation,

$$D_0^2 x_0 + x_0 = 0,$$  \hspace{1cm} (D5)

for which the solution is $x_0 = r \cos(\theta)$, \( \theta = T_1 + \phi(T_2, T_3, \ldots) \), and accordingly $y_0 = D_0 x_0 = -r \sin(\theta)$. Now the second pair of equations in (D3) becomes

$$D_0 x_1 - y_1 + D_1 r \cos(\theta) - r^2 \cos(\theta)^2 + (b + 2d)r^2 \cos(\theta) \sin(\theta) - cr^2 \sin(\theta)^2 = 0,$$
$$D_0 y_1 + x_1 - D_1 r \sin(\theta) + br^2 \cos(\theta)^2 - (e - 2)r^2 \cos(\theta) \sin(\theta) - dr^2 \sin(\theta)^2 = 0,$$  \hspace{1cm} (D6)

and is treated in the same way as the first pair to obtain an equation containing only one unknown $x_1$:

$$D_0^2 x_1 + x_1 - 2 \sin(\theta) D_1 r - 2r \cos(\theta) D_1 \phi - r^2 \sin(\theta)^2 - (b + 2d)r^2 \cos(\theta) \sin(\theta) - cr^2 \cos(\theta)^2 + dr^2 \cos(\theta)^2 - (e - 2)r^2 \cos(\theta) \sin(\theta) - dr^2 \sin(\theta)^2 = 0.$$  \hspace{1cm} (D7)

This new equation is then simplified using trigonometric identities to reduce the power of the sin and cos terms, yielding

$$D_0^2 x_1 + x_1 + 1/2r^2 \left[ (-2d - b + e - 2) \sin(2\theta) + (1 - 2d - c) \cos(2\theta) - 1 - c \right] - 2 \sin(\theta) D_1 r - 2r \cos(\theta) D_1 \phi = 0.$$  \hspace{1cm} (D8)

In order to eliminate the secular terms in the solution $x_1$, the $\sin(\theta)$ and $\cos(\theta)$ terms must be dropped, so that the coefficients of the secular terms must be zero,

$$D_1 r = 0,$$
$$D_1 \phi = 0,$$  \hspace{1cm} (D9)

yielding $v_0 = D_1 r = 0$, as expected. Then, equation (D8) becomes

$$D_0^2 x_1 + x_1 + 1/2r^2 \left[ (-2d - b + e - 2) \sin(2\theta) + (1 - 2d - c) \cos(2\theta) - 1 - c \right] = 0.$$  \hspace{1cm} (D10)

According to the form in the square bracket of the above equation, $x_1$ is
assumed to be
\[ x_1 = r^2(A_0 + A_2 \cos(2\theta) + B_2 \sin(2\theta)) , \quad (D11) \]
to compute a particular solution by the method of undetermined coefficients. Substituting (D11) back to (D8), to balance the coefficients of cos, sin and constant terms, yields the solution of each unknown coefficients,
\[ A_0 = 1/2 + 1/2c , \]
\[ A_2 = 1/6 - 1/3d - 1/6c , \]
\[ B_2 = -1/3d - 1/6b + 1/6e - 1/3 , \] (D12)
yielding
\[ x_1 = r^2[(1/2 + 1/2c) + (1/6 - 1/3d - 1/6c) \cos(2\theta) \]
\[ + (-1/3d - 1/6b + 1/6e - 1/3) \sin(2\theta)] , \] (D13)
which is substituted into (D3) to obtain
\[ y_1 = r^2[(-1/3 + 1/2b + 5/3d + 1/3c) \sin(2\theta) \]
\[ + (1/3e - 2/3d - 1/3b + 1/2c - 7/6) \cos(2\theta) - 1/2 - 1/2c] . \] (D14)

The computation procedure of the second order and even higher ones are exactly the same. It is always solvable, since at each order, \( x_n \) and \( y_n \) only depend on previously computed \( x_k, y_k, D_kr, D_k\phi \). The term that contains un-evaluated focus values are always linear and have the form of \( D_nD_0x_0 + D_ny_0 \), which then leads to the secular terms in solutions. Therefore, the focus values are always computed by eliminating the secular terms in solutions.

The most expensive part of the computation is the trigonometric transformation which combines the high order sin and cos terms. The Maple routine \texttt{combine[trig]} is relatively slow and inefficient. Instead, a new procedure was written to take advantage of repeating trigonometric terms in each equation. A table is created to memorize new trig-identities, and retrieved when an input term contains existing identities. The new procedure boosted the speed of the whole algorithm by more than five times.

It should be noted that although we have discussed and applied the pertur-
bation method for two-dimensional system (3.1), it actually can be applied to consider general $n$-dimensional system with a Hopf critical point at the origin. More details can be found in [13].

E Flaws in the paper of Lloyd and Pearson

In [10], Lloyd and Pearson made the following claim: "Our example appears to be the first to have been obtained without recourse to some numerical calculation". We think that this conclusion is not correct since their method has some flaws and is not complete. In their calculation, they reduce the original problem to checking whether or not the system $\Psi_1 = \Psi_2 = \Psi_3 = 0, \Phi_4 \neq 0$. The polynomials $\Phi_i, i = 1, ..., 4$ are polynomials in the three rational variables $b_1 < m < b_6$.

To achieve their goal, they did the computations in two steps

Step.1 They show that if the system $\Phi_1 = \Phi_2 = \Phi_3 = 0$ holds, then $\Phi_4 \neq 0$ also holds

Step.2 They try, but fail, to show that the system $\Phi_1 = \Phi_2 = \Phi_3 = 0$ has real solutions.

We explain below why this Step.2 has significant flaws.

To see this, let us have a look at the author’s main computation steps.

1. They compute the following resultants, for which they give patterns that we reproduce below on the right hand sides of the equalities,

   \[
   \text{resultant}(\Phi_1, \Phi_2, b_6) = #m^{144}(b_1^2 + 1)^{188} \star R_1 \\
   \text{resultant}(\Phi_1, \Phi_3 \ mod \ \Phi_1, b_6) = #m^{186}(b_1^2 + 1)^{248} \star R_2 \\
   \text{resultant}(\Phi_1, \Phi_4 \ mod \ \Phi_1, b_6) = #m^{228}(b_1^2 + 1)^{280} \star R_3 \\
   \text{resultant}(R_1, R_2, m) = #(b_1^2 + 1)^{40} \star Z,
   \]

   where the pound sign denotes (large) integer factors while $R_1, R_2, R_3$ are bivariate polynomials in $m, b_1$ and $Z$ are univariate polynomials in $b_1$.  

98
2. They correctly prove that resultant\((R_1, R_2, m)\) and resultant\((R_1, R_3, m)\) cannot have a non-trivial common factor, which implies that Step.1 is properly handled. This is done by

\(\text{(a) computing resultant}(R_1, R_3, m)\mod 44449\) and resultant\((R_1, R_2, m)\mod 44449\), and

\(\text{(b) finding that they have no non-trivial common factors}\)

Indeed, if resultant\((R_1, R_3, m)\) and resultant\((R_1, R_2, m)\) would have a non-trivial common factor over the rationals, they would have one non-trivial common factor modulo 44449. Moreover, the fact that resultant\((R_1, R_3, m)\) and resultant\((R_1, R_2, m)\) have non-trivial common factors would imply that the system \(\Phi_1 = \Phi_2 = \Phi_3 = \Phi_4 = 0\) is inconsistent. (Resultant\((R_1, R_2, m)\) and resultant\((R_1, R_3, m)\) both belong to the ideal generated by \(\Phi_1, \Phi_2, \Phi_3, \Phi_4\)).

3. They try to prove that \(\Phi_1 = \Phi_2 = \Phi_3 = 0\) has real common solutions. First they view \(Z\) as a polynomial in \(w = b_1^2\) and find that \(Z = 0\) has a real positive zero \(w^*\) in the interval \((0.6, 0.7)\). Then they claim that “When \(b_1^2 = w^*\) holds there is a value of the variable \(m\) satisfying \(R_1 = R_2 = 0\), but \(R_3 \neq 0\), and hence a value of \(b_6\) such that \(\Phi_1 = \Phi_2 = \Phi_3 = 0\), with \(\Phi_4 \neq 0\).”

This deduction is wrong for the following reasons.

\(\text{(a) The fact that the univariate equation resultant}\((R_1, R_2, m)\) = 0 has a real solution does not imply that the bivariate system \(R_1 = R_2 = 0\) has real solutions.}\n
For example \(R_1 := y^2 + x + 1\) and \(R_2 := y^2 + 2x + 1\) are two bivariate polynomials in \(x < y\) where the resultant in \(y\) is \(x^2\), which has real solutions, while the system \(R_1 = R_2 = 0\) does not.

\(\text{(b) Even if the system } R_1 = R_2 = 0 \text{ has solutions (complex or real) this does not imply that the } \Phi_1 = \Phi_2 = \Phi_3 = 0 \text{ system has solutions.}\)

For example \(\Phi_1 := z^2 + x + y + 1, \Phi_2 := z^2 + 2x + 2y + 1\) and \(\Phi_3 := z^2 + 3x - 3y + 1\). We have \(R_1 := \text{resultant}(\Phi_1, \Phi_2, z) = (x+y)^2\)
and $R_2 := \text{resultant}(\Phi_1, \Phi_3, z) = (2x - 4y)^2$. Obviously, $R_1$ and $R_2$ have common real solutions, but the system $\Phi_1 = \Phi_2 = \Phi_3 = 0$ has no real solutions.

F Maple input for the quadratic example

read "focusvalues_quadric": # Read in the focus values
eqs := [v2, v3];
vars := SuggestVariableOrder(eqs);
# Suggest a best order for the variables
R := PolynomialRing(vars); # Construct the polynomial ring
dec := Triangularize(eqs, R, output=lazard);
# Compute the triangular decomposition
Info(dec, R);
# Display the output which contains four regular chains,
# $[[c+1], [d, b], [e-5*c-5, b], [e, b]]$;

# Now we check if $v4$ vanishes on each of the regular chains
# Method1: using Regularize.

Regularize(v4, dec[1], R);
# $[[], \text{regular\_chain}]$
# This output shows that $v4$ vanishes on zeros of $\text{dec}[1]$;
# This is equivalent to say that $\text{dec}[1]$ is a center condition.

Regularize(v4, dec[2], R);
# Same as above

Regularize(v4, dec[4], R);
# Same as above

Regularize(v4, dec[3], R);
# The output is [[regular_chain], []],
# which says that v4 does not vanish on all the zeros of dec[3]

# Method2:
dec2 := Triangularize([v2, v3, v4], R, output=lazard);
Info(dec2, R);
# [[c+1], [d, b], [e, b], [d^2+2*c^2+c, e-5*c-5, b]]
# According the result from dec (v2, v3 only),
# [c+1], [d, b], [e, b] are center conditions, since v4 vanishes
# on them. d^2+2*c^2+c must be zero in order to make v4 vanishes
# at [e-5*c-5, b]. Thus, [e-5*c-5=0, b=0], but d^2+2*c^2+c<>0 is
# condition for limit cycle.

dec3 := Triangularize([v2, v3, v4, v5], R, output=lazard);
Info(dec3, R);
# [[c+1], [d, b], [e, b], [d^2+2*c^2+c, e-5*c-5, b]]
# By dec2, all the components from dec2 makes v5 vanishes,
# which means [d^2+2*c^2+c, e-5*c-5, b] is a new center
# condition.

G Maple input for the cubic example

read "focusvalues_cubic";
with(RegularChains);
F:= [F1, F2, F3, F4, F5];
R:= PolynomialRing[vars];  # Construct the polynomial ring
vars:= SuggestVariableOrder(F);
# Suggest a best order for the variables
p := 304166505300000047;  # Pick a large enough prime
Rp := PolynomialRing(vars, p);
# Construct the polynomial ring mod p
dec := Triangularize(F, Rp);
# Compute the triangular decomposition modulo p
map(NumberOfSolutions, dec, Rp);

# Check the number of solutions of each output regular chain
# [474, 214, 112, 34, 18, 1, 1]

ndec := [seq(op(NormalizeRegularChain(rc, Rp,
     'normalized'='strongly')), rc=dec)];
# Normalize each regular chain
edec := [op(EquiprojectableDecomposition(ndec, Rp))];
# Compute the equiprojectable decomposition, which contains
# two regular chains edec[1], edec[2]
with(MatrixTools);
jm1 := JacobianMatrix(F, edec[1], Rp);  # Jacobian of edec[1]
MatrixTools:-MatrixInverse(jm1, edec[1], Rp);
# Check if the Jacobian is invertable, which returns false
jm2 := JacobianMatrix(F, edec[2], Rp);  # Jacobian of edec[2]
MatrixTools:-MatrixInverse(jm2, edec[2], Rp);

# The Jacobian of edec[1] is zero
Equation(edec[1], Rp);
# Show the equations in edec[1], which contains f=0
# This is a known center condition

# The Jacobian of edec[2] is non-zero
Lift(F, R, edec[2], 10, p);  # Lift the edec[2]
eqn0 := Equations(dec, Rp);
# Extract the equations from edec[2]
# check if the five equations is initial is 0 mod p
expand(Initial(eqn0[1], R)) mod p;
expand(Initial(eqn0[2], R)) mod p;
expand(Initial(eqn0[3], R)) mod p;
expand(Initial(eqn0[4], R)) mod p;
expand(Initial(eqn0[5], R)) mod p;

# check if still a regular chain mod p;
eqp := map(x->expand(x) mod p, eq0);

rc := Empty(Rp);
rc := Chain(eqp[5..-1], rc, Rp);
# Reconstruct the regular chain mod p
Regularize(Initial(eqp[4], Rp), rc, Rp);
# [[[regular_chain], []]]
rc := Chain(ListTools:-Reverse(eqp[4..-1]), Empty(Rp), Rp);
Regularize(Initial(eqp[3], Rp), rc, Rp);
# [[[regular_chain], []]]
rc := Chain(ListTools:-Reverse(eqp[3..-1]), Empty(Rp), Rp);
Regularize(Initial(eqp[2], Rp), rc, Rp);
# [[[regular_chain], []]]
rc := Chain(ListTools:-Reverse(eqp[2..-1]), Empty(Rp), Rp);
Regularize(Initial(eqp[1], Rp), rc, Rp);
# [[[regular_chain], []]]
rc := Chain(ListTools:-Reverse(eqp), Empty(Rp), Rp);
# It turns out that it is still a regular chains mod p

read "v9": # Read the next focus value v9
Regularize(v9, rc, Rp);
# Check if the regular chain makes v9 vanish
# [[[regular_chain], []]]
# v9 does not vanish on the regular chain, so the eq0 deals
# to limit cycles
Bibliography


# Curriculum Vitae

**Name:** Yiming Zhang  

**Education**

**The University of Western Ontario**  
PhD candidate in Applied Mathematics  
supervised by Robert M. Corless  
2009-2013

**Brock University**  
MSc in Applied Mathematics  
supervised by Thomas Wolf  
2007-2009

**Publications**


**Related Work**

**Teaching Assistant and Research Assistant Experience:**  
University of Western Ontario  
2009-2013

Teaching Assistant  
Brock University  
2007-2009