



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Instituto de Física Armando Dias Tavares

João Paulo Coutinho Eiras

Sistemas dinâmicos em três dimensões com integrais primeiras

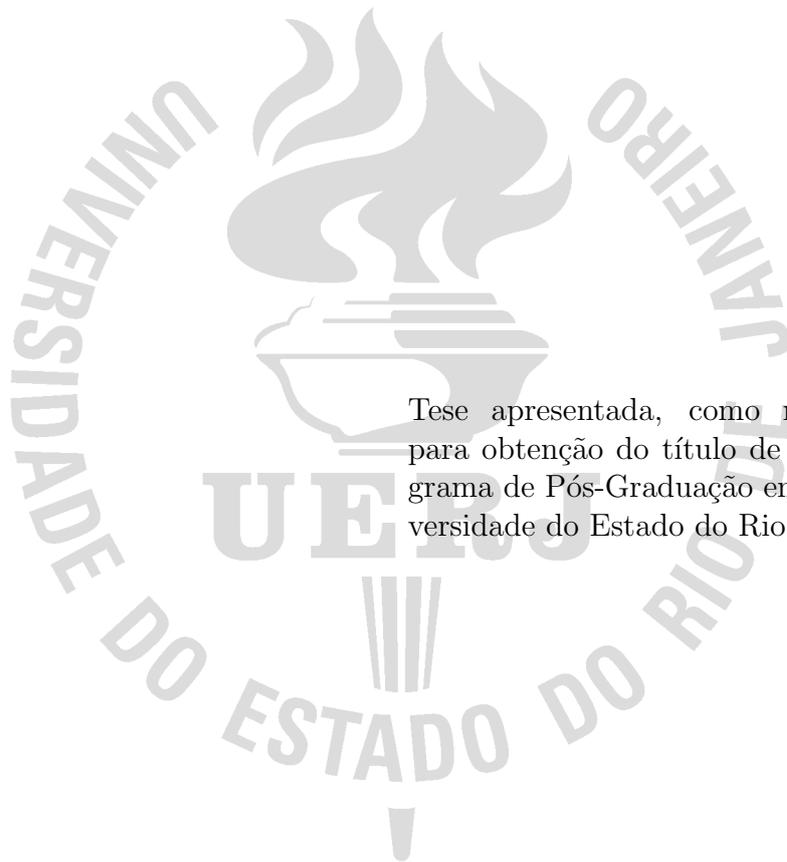
Liouvillianas: abordagens concretas

Rio de Janeiro

2021

João Paulo Coutinho Eiras

**Sistemas dinâmicos em três dimensões com integrais primeiras Liouvillianas:
abordagens concretas**



Tese apresentada, como requisito parcial para obtenção do título de Doutor, ao Programa de Pós-Graduação em Física, da Universidade do Estado do Rio de Janeiro.

Orientador: Prof. Dr. Luís Antônio Campinho Pereira da Mota

Coorientador: Prof. Dr. Luiz Guilherme Silva Duarte

Rio de Janeiro

2021

CATALOGAÇÃO NA FONTE
UERJ/ REDE SIRIUS / BIBLIOTECA CTC/D

E35s

Eiras, João Paulo Coutinho.

Sistemas dinâmicos em três dimensões com integrais primeiras Liouvillianas: abordagens concretas / João Paulo Coutinho Eiras. – 2021.

81 f.

Orientador: Luís Antônio Campinho Pereira da Mota.

Coorientador: Luiz Guilherme Silva Duarte.

Tese (doutorado) - Universidade do Estado do Rio de Janeiro, Instituto de Física Armando Dias Tavares.

1. Integrais (Matemática) – Teses. 2. Sistemas dinâmicos diferenciais – Teses. 3. Equações diferenciais ordinárias – Teses. 4. Física matemática – Teses. I. Mota, Luís Antônio Campinho Pereira da. II. Duarte, Luiz Guilherme Silva. III. Universidade do Estado do Rio de Janeiro. Instituto de Física Armando Dias Tavares. IV. Título.

CDU 517.3

Bibliotecária: Teresa da Silva CRB7/5209

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta tese, desde que citada a fonte.

Assinatura

Data

João Paulo Coutinho Eiras

**Sistemas dinâmicos em três dimensões com integrais primeiras Liouvillianas:
abordagens concretas**

Tese apresentada, como requisito parcial para obtenção do título de Doutor, ao Programa de Pós-Graduação em Física, da Universidade do Estado do Rio de Janeiro.

Aprovada em 13 de Dezembro de 2021.

Banca Examinadora:

Prof. Dr. Luís Antônio Campinho Pereira da Mota (Orientador)
Instituto de Física Armando Dias Tavares - UERJ

Prof. Dr. Luiz Guilherme Silva Duarte (Coorientador)
Instituto de Física Armando Dias Tavares - UERJ

Prof. Dr. Cesar Augusto Linhares da Fonseca Júnior
Instituto de Física Armando Dias Tavares - UERJ

Prof. Dr. Rafael Fernandes Aranha
Instituto de Física Armando Dias Tavares - UERJ

Prof. Dr. Fernando Pereira Paulucio Reis
Universidade Federal do Espírito Santo

Prof. Dr. Jair Koiller
Universidade Federal do Rio Janeiro

Prof. Dr. Rafael De Sousa Dutra
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca

Prof. Dr. Sergio Eduardo Silva Duarte
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca
Rio de Janeiro

DEDICATÓRIA

Aos meus pais, Mario Wilson e Maria de Nazaré (*in memoriam*), a minha irmã, Marcele e ao meu sobrinho Daniel.

RESUMO

EIRAS, J.P.C. *Sistemas dinâmicos em três dimensões com integrais primeiras Liouvillianas: abordagens concretas*. 2021. 80 f. Tese (Doutorado em Física) – Instituto de Física Armando Dias Tavares, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2021.

Nesta tese apresentamos um método semi algorítmico para determinar *integrais primeiras Liouvillianas* de sistemas dinâmicos polinomiais em três dimensões. O procedimento que apresentamos aqui é uma extensão do que construímos em minha dissertação de mestrado (onde buscávamos integrais primeiras elementares) e segue, inicialmente, uma linha análoga. Contudo, no caso de integrais primeiras Liouvillianas não elementares, a estrutura do fator integrante muda e, assim, para determiná-lo, se torna necessário a determinação de *fatores exponenciais*. Dessa maneira, neste trabalho, aprimoramos o método mostrado em (1) e desenvolvemos dois algoritmos (semi): O primeiro é uma extensão do método mostrado em (1, 2) e necessita do cômputo de *polinômios de Darboux* e *fatores exponenciais* do *campo vetorial associado* ao sistema dinâmico; o segundo é baseado em uma técnica análoga à que foi desenvolvida em (3) e se apoia na determinação de campos vetoriais em duas dimensões associados ao campo vetorial 3D original. Também fazemos duas aplicações teóricas do segundo algoritmo para dois casos importantes:

1. Equações diferenciais ordinárias racionais de segunda ordem (2EDOs racionais) – Melhoramos a eficiência do algoritmo desenvolvido em (3) e demonstramos que a determinação dos campos vetoriais associados se constitui em um algoritmo completo.
2. Equações diferenciais ordinárias de primeira ordem (1EDOs) com uma função elementar – Ampliamos o algoritmo desenvolvido em (4) para incluir a resolução de 1EDOs com funções algébricas.

Os procedimentos que desenvolvemos foram implementados em um pacote computacional (*LeapS3Dim*) em Maple. O pacote *LeapS3Dim* possui vários comandos que realizam os cálculos intermediários necessários para a obtenção da integral primeira tornando-o, assim, útil para a pesquisa em física e matemática.

Palavras-chave: Sistemas dinâmicos em três dimensões. Equações diferenciais ordinárias de primeira e segunda ordens. Campos vetoriais associados. Integrais primeiras Liouvillianas.

ABSTRACT

EIRAS, J.P.C. *Dynamic systems in three dimensions with Liouvillian first integrals: concrete approaches*. 2021. 80 f. Tese (Doutorado em Física) – Instituto de Física Armando Dias Tavares, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2021.

In this thesis we present a semi-algorithmic method to determine *Liouvillian first integrals* of polynomial dynamical systems in three dimensions. The procedure we present here is an extension of what we built in my master's thesis (where we were looking for elementary first integrals) and, initially, follows a similar line. However, in the case of non-elementary Liouvillian first integrals, the structure of the integrating factor changes and, thus, to determine it, it is necessary to determine *exponential factors*. Thus, in this work, we improve the method shown in (1) and develop two algorithms (semi): The first is an extension of the method shown in (2, 1) and requires the computation of *Darboux polynomials* and *exponential factors* of the *vector field associated* to the dynamical system; the second is based on a technique analogous to the one developed in (3) and is based on the determination of vector fields (in two dimensions) associated with the original 3D vector field. We also make two theoretical applications of the second algorithm (for two important cases):

1. Second order rational ordinary differential equations (2 rational ODEs) – We improve the efficiency of the algorithm developed in (3) and demonstrate that the determination of the associated vector fields constitutes a complete algorithm.
2. First order ordinary differential equations (1ODEs) with an elementary function – We extend the algorithm developed in (4) to include the resolution of 1ODEs with algebraic functions.

The procedures we developed in this thesis were implemented in a computational package (*LeapS3Dim*) in Maple. The *LeapS3Dim* package has several commands that perform the intermediate calculations necessary to obtain the first integral, thus making it useful for research in physics and mathematics.

Keywords: Dynamic systems in three dimensions. Ordinary differential equations of first and second orders. Associated vector Fields. Liouvillian first integrals.

SUMÁRIO

	INTRODUÇÃO	8
1	CAMPOS VETORIAIS POLINOMIAS EM 3D (3CVPS) COM INTEGRAIS PRIMEIRAS LIOUVILLIANAS	13
1.1	A abordagem Darbouxiana e o método da função-S	13
1.1.1	<u>Os métodos de Prelle-Singer (PS) e de Christopher-Singer(CS)</u>	15
1.1.2	<u>O método da função-S para sistemas</u>	18
1.2	Campos vetoriais polinomias em 3D com uma integral primeira Liouvilliana	19
1.3	Campos vetoriais polinomiais no plano associados com \mathfrak{X}	22
1.4	Possíveis algoritmos para determinar integrais primeiras	25
1.4.1	<u>Procedimento 1: Determinando um fator integrante</u>	25
1.4.2	<u>Procedimento 2: Calculando \mathfrak{X}_i</u>	29
2	1EDOS COM UMA FUNÇÃO ELEMENTAR	31
2.1	Um aprimoramento de P_{RO2} para uma classe de 1EDOs	35
2.2	Calculando (Q, P, N) linearmente	41
3	2EDOS RACIONAIS	43
4	O PACOTE <i>LEAPS3DIM</i>	48
4.1	Os comandos do pacote <i>LeapS3Dim</i>	50
4.1.1	<u>Comando: Dx</u>	51
4.1.2	<u>Comando: XX</u>	51
4.1.3	<u>Comando: Tr3Dsys</u>	52
4.1.4	<u>Comando: Tr1ode</u>	52
4.1.5	<u>Comando: Tr2ode</u>	53
4.1.6	<u>Comando: Xi2D</u>	54
4.1.7	<u>Comando: Darboux</u>	54
4.1.8	<u>Comando: FindA</u>	55
4.1.9	<u>Comando: InFa</u>	56
4.1.10	<u>Comando: InFa2</u>	57
4.1.11	<u>Comando: Invar</u>	58
4.1.12	<u>Comando: Solde</u>	59
4.2	Exemplos da utilização dos comandos do pacote	60
4.2.1	<u>Lidando com 3CVPs gerais</u>	60
4.2.2	<u>Lidando com 1EDOs com um função elementar</u>	62
4.2.3	<u>Lidando com 2EDOs racionais</u>	64
5	DESEMPENHO	65
5.1	Procedimento 1 (P_{RO1}) \times Procedimento 2 (P_{RO2})	65

5.2	O Algoritmo $P_{RO3.2}$ – 1EDOs com uma função elementar	69
5.3	O Algoritmo $P_{RO4.2}$ – 2EDOs racionais	71
5.4	Uma breve discussão sobre os algoritmos	73
5.5	Os procedimentos 1 e 2	73
5.6	Os procedimentos 3.1 e 3.2	74
5.7	Os procedimentos 4 e 4.2	75
	CONCLUSÃO	76
	REFERÊNCIAS	77

INTRODUÇÃO

A determinação de uma integral primeira Liouvilliana para um campo vetorial é um problema antigo e difícil. Embora essa questão tenha sido colocada há muito tempo atrás, foi apenas em 2017 que G. Chèze e T. Combot (17) resolveram (de maneira geral) o problema para campos vetoriais polinomiais no plano (\mathbb{R}^2). No caso de campos vetoriais em três variáveis, a complexidade aumenta consideravelmente e, portanto, o problema geral permanece em aberto. A questão da integrabilidade está entre as principais dificuldades que surgem, pois, no caso de campos vetoriais no \mathbb{R}^3 , a existência de integrais primeiras analíticas globais não é garantida.

Uma das primeiras tentativas para abordar o problema da integrabilidade remonta à segunda metade do século 19, quando Darboux (5) iniciou o desenvolvimento de uma teoria (atualmente conhecida como teoria da integrabilidade de Darboux) que buscava relacionar a integrabilidade de sistemas polinomiais de 1EDOs com a existência de curvas algébricas invariantes sobre as soluções do sistema. Essa teoria se mostrou muito frutífera e tem sido desenvolvida / ampliada / generalizada por diversos pesquisadores ao longo dos anos. Podemos destacar: Jouanolou, Schlomiuk, Singer, Christopher, Llibre, Zhang, Giacomini, Pantazi, Weil, Ferragut, Gasull, Chèze, Combot entre outros (veja os trabalhos ‘pioneiros’: (6, 7, 8, 9, 10)).

Nessa tese vamos apresentar dois procedimentos (semi-algorítmicos)¹ para calcular integrais primeiras Liouvillianas (IPL) de campos vetoriais polinomiais no \mathbb{R}^3 (3CVP): o primeiro deles é uma extensão do método apresentado em (1, 2) e busca determinar um fator integrante para o 3CVP; o segundo consiste em uma generalização do método da função-S (veja (3)). Em resumo, o primeiro procedimento se assemelha à abordagem Darbouxiana para encontrar as integrais primeiras e o outro tenta evitar os problemas envolvidos no cálculo dos polinômios de Darboux, usando para isso, campos vetoriais polinomiais no plano (2CVP) associados ao campo vetorial 3D original². Os métodos apresentados aqui usam semi-algoritmos não lineares. Em dois casos, podemos aprimorar o segundo procedimento (e em um desses dois casos, podemos construir um algoritmo linear).

Estes são os nossos principais resultados e a tese será organizada da seguinte forma:

¹ Semi-algorítmico significa que o procedimento (como um todo) pode não terminar, isto é, o procedimento não é um algoritmo completo.

² Mostramos que, com o conhecimento de um desses campos vetoriais 2D associados, o problema da determinação de uma integral primeira Liouvilliana para um 3CVP torna-se equivalente ao problema de se encontrar uma integral primeira Liouvilliana para um campo vetorial polinomial no plano (e mais algumas quadraturas ocasionais).

1. Nessa introdução, apresentamos a estrutura deste trabalho e introduzimos (brevemente) a notação usada.
2. No capítulo 1, desenvolvemos o arcabouço teórico que usaremos no restante deste trabalho:
 - (a) Na primeira seção, estabelecemos as bases e os resultados necessários para construir as extensões ao que foi realizado em minha dissertação (1). Apresentamos os principais conceitos envolvidos e alguns teoremas-chave para os desenvolvimentos posteriores:
 - i. Começamos por apresentar os conceitos centrais à abordagem Darbouxiana: *campo vetorial associado, polinômio de Darboux; integral primeira Liouvilliana; fator integrante etc.*
 - ii. Em seguida apresentamos os resultados principais envolvidos na abordagem Darbouxiana: os teoremas de Prelle-Singer, Singer e Christopher e alguns métodos que podem ser construídos com esses resultados.
 - iii. O método da função-S aplicado a sistemas (do modo como foi desenvolvido em (1)).
 - (b) Na segunda seção, com base nos resultados obtidos por Singer e Christopher em (9, 10) (que se aplicam a campos vetoriais polinomiais no plano) nós provamos um resultado para campos vetoriais polinomiais em três variáveis (3CVPs): resumidamente, considere o campo vetorial dado por

$$\mathfrak{X} \equiv F \partial_x + G \partial_y + H \partial_z,$$

onde F, G, H são polinômios coprimos em $\mathbb{C}[x, y, z]$. Mostramos que a existência de uma integral primeira Liouvilliana $I(x, y, z)$ para o campo \mathfrak{X} com derivadas da forma $I_x = RQ, I_y = RP, I_z = RN$ (onde R é uma função Liouvilliana em (x, y, z) e Q, P, N são polinômios coprimos em $\mathbb{C}[x, y, z]$), implica na existência de um fator integrante de Darboux, isto é, em um fator integrante que é da forma

$$R = \exp(A/B) \prod_i p_i^{n_i},$$

onde A, B são polinômios coprimos em $\mathbb{C}[x, y, z]$, os p_i são polinômios irreduzíveis em $\mathbb{C}[x, y, z]$ e os n_i são constantes. Além disso, os polinômios p_i e B são polinômios de Darboux do campo vetorial \mathfrak{X} . Esse resultado nos permite construir duas abordagens distintas para resolver o problema:

- i. Procedimento 1 – Uma abordagem mais clássica no estilo do procedimento original de Darboux, i.e, o cálculo de polinômios de Darboux para construir

um fator integrante. Nesta vertente, construímos um sub-procedimento que tornou prático um método que abandonamos em (1) por ser cúbico nos indeterminados. Em relação à ideia central, é uma generalização do semi-algoritmo apresentado em (13, 14, 15, 16) e, em termos gerais, muito semelhante ao procedimento seguido em (2) (que se aplica a 3CVPs com integrais primeiras elementares).

- ii. Procedimento 2 – Basicamente, uma generalização do método da função-S para sistemas (como foi construído em (1)), isto é, mostramos que é possível construir campos vetoriais polinomiais no plano (2CVPs) que apresentam como uma integral primeira Liouvilliana (IPL) a mesmo IPL admitida pelo campo vetorial 3D original (\mathfrak{X}). Essa técnica é uma extensão do processo apresentado em (3) para lidar com 2EDOs racionais.

Ao final de cada seção, os passos do método desenvolvido são aplicados a um exemplo para esclarecer seu funcionamento.

3. No capítulo 2, apresentamos um aperfeiçoamento do método apresentado no capítulo 1, que se aplica a campos vetoriais tridimensionais associados a 1EDOs que possuem uma função elementar e que podem ser escritas como

$$y' = \frac{dy}{dx} = \phi(x, y, \theta), \quad (1)$$

onde ϕ é uma função racional de (x, y, θ) , i.e., $\phi \in K = \mathbb{C}(x, y, \theta)$, $\theta = \theta(x, y)$. Se (1) apresenta uma solução geral Liouvilliana $I(x, y, \theta) = c$ de tal modo que as derivadas parciais de I (relacionadas às variáveis x, y e θ) possam ser expressas como

$$\begin{aligned} I_x &= RQ, \\ I_y &= RP, \\ I_\theta &= RN, \end{aligned}$$

onde R é uma função Liouvilliana de (x, y, θ) e $Q, P, N \in C[x, y, \theta]$, então é possível construir um algoritmo linear para determinar ³ campos vetoriais polinomiais \mathfrak{X}_i no plano (2CVPs) que apresentam $I(x, y, z)$ como integral primeira. Em outras palavras, podemos construir um algoritmo linear que reduz o problema original ao de encontrar uma integral primeira Liouvilliana de um campo vetorial polinomial no plano (problema que foi recentemente resolvido por Chèze e Combot

³ As funções S definem os 2CVPs que ‘compartilham’ a integral primeira Liouvilliana I com o campo vetorial tridimensional original.

(17)). Apresentamos em seguida um exemplo e fazemos alguns comentários sobre o desempenho do algoritmo.

4. No capítulo 3, tratamos de campos vetoriais associados a 2EDOs racionais que apresentem ao menos uma integral primeira Liouvilliana $I(x, y, z)$ tal que, para a função I , valem as mesmas condições descritas no capítulo 2. Mostramos que, neste caso, dependemos da determinação de apenas um dos polinômios Q, P, N (e não de dois como acontece para os procedimentos desenvolvido no capítulo 2). Desenvolvemos um procedimento que melhora o método apresentado em (3) e o torna muito menos oneroso (do ponto de vista computacional). Também mostramos que o procedimento para determinar os 2CVPs associados se torna, neste caso, um algoritmo completo. Apresentamos, ao final da seção, dois exemplos comparativos.
5. No capítulo 4 apresentamos uma implementação computacional dos algoritmos desenvolvidos. Criamos o pacote *LeapS3Dim – Liouvillian, Elementary, Algebraic, Polynomial Solutions of 3D Systems*.
 - (a) Na primeira seção, mostramos um resumo dos comandos que compõe o pacote *LeapS3Dim* e sua funcionalidade e, em seguida, uma descrição mais detalhada de cada um dos comandos que o compõe.
 - (b) Na segunda seção, apresentamos exemplos do uso dos comandos do pacote aplicados a exemplos específicos.
6. No capítulo 5, apresentamos alguns exemplos para melhor ilustrar os procedimentos e discutir o desempenho dos algoritmos:
 - (a) Na primeira seção, mostramos uma comparação entre os métodos apresentados no capítulo 1 (para 3CVPs em geral) e outros métodos disponíveis: comparamos os tempos gastos pelo nossos procedimentos (PRO1 e PRO2) em uma ‘implementação teste’ em Maple⁴.
 - (b) Na segunda seção, apresentamos alguns exemplos de 1EDOs com funções elementares e sua solução com o algoritmo de ‘redução’ linear (PRO3). Também mostramos os tempos gastos e a quantidade de memória utilizada em cada caso.
 - (c) Na terceira subseção, comparamos o método apresentado no capítulo 3 com seu ‘predecessor’ (o método apresentado em (3)) e com o Maple (o comando

⁴ Maple é um sistema de computação algébrica de propósito geral (CAS) que é também uma linguagem de programação.

`dsolve`) para várias 2EDOs racionais. Também mostramos, como nas subseções anteriores, os tempos de CPU gastos e a quantidade de memória RAM utilizada.

- (d) Na última seção, discutimos um pouco sobre as vantagens e desvantagens, limitações e generalizabilidade dos algoritmos apresentados.

7. Por último, apresentamos nossas conclusões e possíveis desenvolvimentos.

1 CAMPOS VETORIAIS POLINOMIAIS EM 3D (3CVPS) COM INTEGRAIS PRIMEIRAS LIOUVILLIANAS

Neste capítulo vamos construir a base teórica necessária para lidar com campos vetoriais polinomiais em três variáveis. O capítulo terá a seguinte estrutura:

- Na primeira seção vamos apresentar alguns conceitos e os principais teoremas relevantes para a abordagem Darbouxiana e para o método da função-S.
- Na seção 2, vamos desenvolver mais formalmente as ideias apresentadas em (1) e construir dois procedimentos para buscar integrais primeiras Liouvillianas de sistemas diferenciais polinomiais em três dimensões.

1.1 A abordagem Darbouxiana e o método da função-S

Em fins do século XIX, G. Darboux (5) iniciou o desenvolvimento de uma teoria (atualmente conhecida como teoria da integrabilidade de Darboux) que era centrada no conceito de certos polinômios especiais hoje chamados de polinômios de Darboux⁵. Darboux mostrou em (18) que se um campo vetorial polinomial \mathfrak{X} possui pelo menos $\binom{n+d-1}{n} + 1$ polinômios de Darboux irredutíveis (onde d é o grau do sistema e n a sua dimensão), então \mathfrak{X} tem uma integral primeira que pode ser expressa por meio desses polinômios. Em 1983 (em (8)), Prelle and Singer mostraram ser possível construir um fator integrante algébrico para um campo vetorial polinomial 2D (2CVP) se este possuísse uma integral primeira elementar. Este artigo indicava a possibilidade de se construir um semi algoritmo para determinar a integral primeira de um 2CVP. A ideia central, portanto, era calcular os polinômios de Darboux e construir um fator integrante e, a partir dele (ou diretamente), uma integral primeira. Assim, o primeiro passo neste caminho é determinar esses autopolinômios.⁶

⁵ Existem muitos nomes diferentes na literatura para polinômios de Darboux como, por exemplo: integrais especiais, autopolinômios, hipersuperfícies algébricas invariantes, polinômios especiais ou integrais segundas.

⁶ Na realidade, as coisas são um pouco mais complicadas do que isso porque é necessário saber (em primeiro lugar) se o campo vetorial em questão possui polinômios de Darboux. Para pesquisas neste campo, veja (19). Além disso, também é necessário saber se existe uma integral primeira Liouvilliana uma vez que os algoritmos envolvidos são, na verdade, semi-algoritmos. Assim, além dos desenvolvimentos que buscam diretamente por melhores algoritmos para computar polinômios de Darboux, há uma extensa pesquisa que estuda a integrabilidade de campos vetoriais apresentando integrais primeiras Liouvillianas (integrabilidade de Darboux). Veja, por exemplo, (20, 21, 22, 23, 24, 25, 26, 27).

No entanto, este passo inicial também é o mais ‘difícil’: determinar polinômios de Darboux é, como regra, computacionalmente caro. Mesmo em graus baixos (na prática, em torno de quatro) a determinação dos polinômios de Darboux pelo método dos coeficientes indeterminados (MCI) começa a se tornar impossível na prática. Por este motivo, tem havido, nas últimas duas décadas, uma extensa busca por procedimentos para melhorar esta situação, ou seja, determinar os polinômios de Darboux em menos tempo de processamento computacional ou menos uso de memória. Ou seja, uma busca por algoritmos mais eficientes (mais rápidos e / ou menos onerosos). Nesse sentido, destacamos os trabalhos de G. Chèze, A. Ferragut, A. Gasull, A. Bostan, T. Cluzeau, J.A. Weil, H. Giacomini, T. Combot, C. Christopher, J. Llibre, C. Pantazi, S. Walcher, C. Galindo, F. Monserrat, X. Zhang entre outros:

- Em (22, 24) C. Christopher, J. Llibre, C. Pantazi e S. Walcher desenvolveram uma abordagem algorítmica para determinar os campos vetoriais polinomiais no plano que apresentam um determinado polinômio de Darboux. Seu método pode fornecer uma expressão explícita para esses campos vetoriais se todos os pontos singulares (finitos) das curvas algébricas (definidas pelos polinômios de Darboux) forem não-degenerados.
- Em (28), A. Ferragut e H. Giacomini desenvolveram um algoritmo rápido para calcular as integrais primeiras racionais de um campo vetorial polinomial no plano. O algoritmo também permite o cálculo das curvas notáveis associadas à integral primeira racional.
- Em (29) G. Chèze mostrou como calcular (para um campo vetorial polinomial no plano, $D \equiv N \partial_x + M \partial_y$ de grau d) todos os polinômios Darboux irredutíveis de grau menor que k com $\mathcal{O}((dk \log(\mathcal{H}))^{\mathcal{O}(1)})$ operações binárias⁷. Ele baseou seu método na fatoração da curva extática (sugestão de J.V. Pereira (23)).
- Em (30) A. Bostan, G. Chèze, T. Cluzeau e J.A. Weil, com base na ideia de Ferragut e Giacomini (28), construíram algoritmos para calcular integrais primeiras racionais por meio de sistemas de equações lineares em vez de sistemas de equações quadráticas.
- Em (31) A. Ferragut e A. Gasull constroem um método que melhora (e muito) o método dos coeficientes indeterminados (MCI) para a determinação dos polinômios de Darboux de um campo vetorial polinomial.

⁷ $\mathcal{H} = \max(\|M\|_\infty, \|N\|_\infty)$; $d = \max(\deg(M), \deg(N))$.

- Em (32) A. Ferragut, C. Galindo e F. Monserrat desenvolveram um método para calcular as integrais primeiras integrais Darbouxianas do tipo $\prod_i p_i^{\alpha_i}$ onde os p_i são polinômios de Darboux.
- Em (17), G. Chèze e T. Combot generalizam (para o caso Darbouxiano, Liouvilliano e de Riccati) a curva extática introduzida por J. Pereira em (33) e constroem novos algoritmos para computação de integrais primeiras racionais, Darbouxianas, Liouvillianas ou de Riccati de um campo vetorial polinomial no plano (2CVP).

Na subseção seguinte apresentaremos os procedimentos clássicos (usando o método dos coeficientes indeterminados – MCI) para encontrar integrais primeiras elementares (método de Prelle-Singer) e integrais primeiras Liouvillianas não elementares (método de Christopher-Singer) para campos vetoriais polinomiais no plano.

1.1.1 Os métodos de Prelle-Singer (PS) e de Christopher-Singer(CS)

Considere um sistema diferencial polinomial em duas dimensões dado por

$$\begin{cases} \dot{x} = N_0(x, y), \\ \dot{y} = M_0(x, y), \end{cases} \quad (2)$$

onde M_0 e N_0 são polinômios coprimos em $\mathbb{C}[x, y]$. Uma integral primeira I do sistema (2) é uma função que é constante nas soluções de (2).

Definição 1.1. *Seja $L(x, y)$ uma extensão Liouvilliana do corpo de funções $\mathbb{C}(x, y)$. Uma função $I \in L(x, y)$ é dita ser uma **integral primeira Liouvilliana** do sistema (2) se $\mathfrak{X}_{2D}(I) = 0$, onde $\mathfrak{X}_{2D} \equiv N_0 \partial_x + M_0 \partial_y$ é o **campo vetorial associado** (ou o **operador de Darboux associado**) com o sistema (2).*

Observação 1.1. *Se $I \in E(x, y)$, onde $E(x, y)$ é uma extensão elementar⁸ do corpo $\mathbb{C}(x, y)$, I é dita ser uma **integral primeira elementar**.*

Observação 1.2. *Como o sistema (2) é autônomo, podemos dividir \dot{y} por \dot{x} para obter uma 1EDO racional*

$$y' \equiv \frac{dy}{dx} = \frac{M_0(x, y)}{N_0(x, y)}. \quad (3)$$

Para esta 1EDO racional, a função I define sua solução geral na forma implícita: $I(x, y) = c$. Além disso, a 1-forma $\gamma \equiv M_0 dx - N_0 dy$ é nula sobre as soluções da 1EDO (3) e,

⁸ Para uma definição formal de extensão Liouvilliana ou elementar de um corpo, veja (11).

portanto, a 1-forma exata $\omega \equiv dI$ é proporcional a γ , ou seja, $\omega = R\gamma$. A função R é comumente chamada de **fator integrante**.

O método PS pode ser facilmente compreendido se prestarmos atenção aos seguintes resultados obtidos por Prelle e Singer (8):

Teorema 1.1 (Prelle-Singer).:

Se o sistema (2) apresenta uma integral primeira elementar, então:

(i) O sistema (2) apresenta uma integral primeira da forma $I = W_0 + \sum_{j=1}^K c_j \ln(W_j)$, W 's são funções algébricas.

(ii) A 1-forma $M_0 dx - N_0 dy$ apresenta um fator integrante da forma $R = \prod_i p_i^{n_i}$, onde os p_i são polinômios irredutíveis e os n_i são números racionais.

Prova Para uma prova veja (8).

Definição 1.2. Seja $p(x, y) \in \mathbb{C}[x, y]$. Diz-se que o polinômio p é um **polinômio de Darboux** do campo vetorial \mathfrak{X}_{2D} , se $\mathfrak{X}_{2D}(p) = qp$, onde q é um polinômio em $\mathbb{C}[x, y]$ que é chamado de **cofator** de p .

Corolário 1.1. Se a hipótese do teorema 1.1 for satisfeita, então

$$\sum_i n_i \frac{\mathfrak{X}_{2D}(p_i)}{p_i} = -\mathbf{div}(\mathfrak{X}_{2D}),$$

onde \mathfrak{X}_{2D} é o campo vetorial associado a (2), p_i são polinômios de Darboux de \mathfrak{X}_{2D} e \mathbf{div} significa divergente.

Prova: Para uma prova, veja a seção 2 de (12).

O corolário 1.1 é a chave para o método PS:

Procedimento (esboço):

1. Construa os candidatos para p e q com coeficientes indeterminados. Vamos chamá-los de: p_c e q_c . Substitua-os na equação $\mathfrak{X}_{2D}(p_c) - q_c p_c = 0$.
2. Colete a equação nas variáveis (x, y) obtendo um conjunto de equações (quadráticas) para os coeficientes dos candidatos. Resolva este conjunto de equações para os coeficientes indeterminados.
3. Substitua as soluções na equação $\sum_i n_i q_i + N_{0x} + M_{0y} = 0$ (ver corolário 1.1) e reúna a equação nas variáveis (x, y) . Resolva o conjunto de equações para n_i .
4. Construa o fator integrante $R = \prod_i p_i^{n_i}$ e encontre (por quadraturas) a integral primeira elementar I .

O método CS segue diretamente dos resultados obtidos por M. Singer (9) e C. Christopher (10):

Teorema 1.2 (Singer).:

Se a 1-forma $M_0 dx - N_0 dy$ associada ao sistema (2) apresenta uma integral primeira Liouwilliana, então ela tem um fator integrante da forma

$$R(x, y) = \exp \left[\int U(x, y) dx + V(x, y) dy \right], \quad (4)$$

onde U e V são funções racionais com $U_y = V_x$ de forma que a integral de linha (4) é bem definida.

Prova: Para uma prova veja (9).

Teorema 1.3 (Christopher).:

Se o sistema (2) tem um fator integrante da forma (4) onde U e V são funções racionais com $U_y = V_x$, então existe um fator integrante para o sistema (2) da forma

$$R(x, y) = \exp \left(\frac{A}{B} \right) \prod_i p_i^{n_i}, \quad (5)$$

onde A , B e os p_i são polinômios em x e y .

Prova: Para uma prova veja (10, 14).

Corolário 1.2. B e os p_i são polinômios de Darboux do campo vetorial \mathfrak{X}_{2D} e $\mathfrak{X}_{2D} \left(\frac{A}{B} \right)$ é um polinômio.

Prova: Para uma prova veja (10, 14).

O corolário 1.2 permite a construção de um semi-algoritmo na linha do método PS:

Procedimento (esboço):

1. Construa os candidatos para p e q com coeficientes indeterminados. Vamos chamá-los de: p_c e q_c . Substitua-os na equação $\mathfrak{X}_{2D}(p_c) - q_c p_c = 0$.
2. Colete a equação nas variáveis (x, y) obtendo um conjunto de equações (quadráticas) para os coeficientes dos candidatos. Resolva este conjunto de equações para os coeficientes indeterminados.
3. Construa todos os candidatos possíveis para B (até certo grau) e um candidato para A . Substitua os candidatos na equação $B_c \mathfrak{X}_{2D}(A_c) - A_c \mathfrak{X}_{2D}(B_c) + B_c^2 (\sum_i n_i q_i + N_{0x} + M_{0y}) = 0$ (veja (14)) e colete a equação nas variáveis (x, y) . Resolva o conjunto de equações para n_i e para os coeficientes de A_c .

4. Construa o fator integrante $R = \exp(A/B) \prod_i p_i^{n_i}$ e encontre (por quadraturas) a integral primeira Liouvillianiana I .

Apresentaremos a seguir, o método da função-S aplicado a campos vetoriais polinomiais em 3D desenvolvido em (1).

1.1.2 O método da função-S para sistemas

Considere o sistema diferencial polinomial abaixo

$$\begin{cases} \dot{x} = F(x, y, z) \\ \dot{y} = G(x, y, z) \\ \dot{z} = H(x, y, z), \end{cases} \quad (6)$$

onde F, G, H são polinômios coprimos em $\mathbb{C}[x, y, z]$. Uma integral primeira I do sistema (6) é uma função de (x, y, z) cujo valor é constante sobre todas as soluções de (9).

Definição 1.3. *Seja $I(x, y, z)$ uma integral primeira do sistema (6) e seja S uma função de (x, y, z) definida por $S \equiv I_y/I_z$. Chamamos S de função-S associada ao sistema (6) através de I .*

Teorema 1.4. *Considere que o sistema (6) apresenta uma IPL I e uma função-S racional associada ao sistema através de I . Além disso, seja $\mathfrak{X} \equiv F \partial_x + G \partial_y + H \partial_z$ seu operador de Darboux. Então S obedece à equação*

$$\mathfrak{X}(S) = S^2 \left(\frac{FG_z - GF_z}{F} \right) + S \left(\frac{GF_y - FG_y + FH_z - HF_z}{F} \right) - \left(\frac{FH_y - HF_y}{F} \right). \quad (7)$$

Prova: Para uma prova veja (1).

Teorema 1.5. *Considere que a equação (1.4) possui uma solução racional P/N (P e N polinômios coprimos). Então, o polinômio $p_s \equiv P(x, y, z)s - N(x, y, z)$ é um polinômio de Darboux do operador definido por $D_s \equiv FD + \Phi \partial_s$, onde $\Phi \equiv (FG_z - GF_z)s^2 + (GF_y - FG_y + FH_z - HF_z)s - (FH_y - HF_y)$.*

Prova: Para uma prova veja (1).

Teorema 1.6. *Considere que o sistema (6) apresenta uma IPL I e uma função-S racional associada ao sistema através de I . Considere ainda que a equação (1.4) possui uma solução racional $S = P/N$ (P e N polinômios coprimos). Então, $I(x, y, z) = c$ é uma solução geral para a 1EDO racional definida por*

$$\frac{dz}{dy} = -\frac{P(x, y, z)}{N(x, y, z)}. \quad (8)$$

Prova: Para uma prova veja (1).

Com esses resultados pudemos elaborar uma estratégia da qual apresentaremos (a seguir) apenas a ideia central:

Procedimento: (esboço)

1. Usando o resultado do teorema 1.5, podemos buscar polinômios de Darboux da forma $P s - Q$ do operador D_s . Ao encontrarmos um teremos determinado P e N .
2. Resolva a 1EDO racional

$$\frac{dz}{dy} = -\frac{P(x, y, z)}{N(x, y, z)},$$

e encontre a integral primeira I .

Alguns comentários:

Observação 1.3. *Os polinômios de Darboux do operador D_s são funções de 4 variáveis (x, y, z, s) e, portanto, ‘difíceis’ de computar.*

Observação 1.4. *O fato de que $I(x, y, z) = c$ é uma solução geral para a 1EDO racional definida por $\frac{dz}{dy} = -\frac{P(x, y, z)}{N(x, y, z)}$, não implica que uma solução geral desta 1EDO será uma integral primeira para o sistema (6). Isso pode acontecer porque x , que é uma variável em relação ao sistema (6), é apenas um parâmetro em relação à 1EDO $dz/dy = -P/N$ (veja (3, 1) para mais detalhes).*

Observação 1.5. *Na próxima seção vamos apresentar um algoritmo mais eficiente (e mais geral) para computar integrais primeiras Liouvillianas de sistemas diferenciais polinomiais em três dimensões. O aperfeiçoamento do método está ligado a uma generalização dos teoremas 1.4, 1.5 e 1.6 e a um detalhe envolvendo o teorema 5.1 de (1) (que serão detalhados na seção seguinte).*

1.2 Campos vetoriais polinomiais em 3D com uma integral primeira Liouvilliana

Em 1999, Christopher mostrou (baseado no trabalho de Singer (9)) que uma integral primeira Liouvilliana de um campo vetorial polinomial (no plano) estava ligada à existência de um fator integrante Darbouxiano⁹ (veja (10)). Esse resultado foi estendido

⁹ Uma função de Darboux é uma função da forma $e^{Z_0} \prod_i p_i^{n_i}$, onde os n_i são constantes, Z_0 é uma função racional e os p_i são funções polinomiais (nos seus argumentos). Um fator integrante Darbouxiano é um fator integrante que é, também, uma função de Darboux.

para mais dimensões (veja, por exemplo, os trabalhos de J. Llibre e X. Zhang (27, 34, 35)) e foi mostrado que, para campos vetoriais polinomiais em n dimensões, a existência de $(n - 1)$ integrais primeiras Liouvillianas (IPLs) implicam na existência de um multiplicador de Jacobi Darbouxiano (veja (27)). Assim, para 3CVPs ($n = 3$), duas IPLs I_1 e I_2 implicam em um multiplicador de Jacobi Darbouxiano J . O uso prático do multiplicador de Jacobi é, em geral, o seguinte: a partir do conhecimento de uma das IPLs (suponha que I_1 seja conhecida), podemos eliminar uma das variáveis (por exemplo, z) e obter um sistema reduzido. Assim, se conhecemos um multiplicador de Jacobi J , podemos construir um fator integrante R para o sistema reduzido: $R = J/I_{1z}$ (U_z denota $\partial_z U$). Porém, se o 3CVP em questão possuir apenas uma IPL I (que ainda não foi determinada) o método mencionado não se aplica e, o que normalmente se faz nesses casos, é buscar um fator integrante R que permita determinar I . Como o fator integrante R não é um multiplicador de Jacobi (no caso tridimensional), nessa tese, o termo *fator integrante* não será utilizado para designar multiplicadores de Jacobi. Vamos, a seguir, fazer algumas definições e estabelecer uma notação adequada ao que desenvolvemos neste trabalho.

Considere o sistema dinâmico polinomial abaixo

$$\begin{cases} \dot{x} = F(x, y, z) \\ \dot{y} = G(x, y, z) \\ \dot{z} = H(x, y, z), \end{cases} \quad (9)$$

onde F, G, H são polinômios coprimos em $\mathbb{C}[x, y, z]$. Uma integral primeira I do sistema (9) é uma função de (x, y, z) cujo valor é constante sobre todas as soluções de (9).

Definição 1.4. *Seja L uma extensão Liouvillianas do corpo de funções racionais $\mathbb{C}(x, y, z)$. A função $I(x, y, z) \in L$ é dita ser uma **integral primeira Liouvillianas (IPL)** do sistema (9) se $\mathfrak{X}(I) = 0$, onde $\mathfrak{X} \equiv F \partial_x + G \partial_y + H \partial_z$ é o **campo vetorial associado** ao sistema diferencial (9).*

Definição 1.5. *Seja $p(x, y, z) \in \mathbb{C}[x, y, z]$. O polinômio p é dito ser um **polinômio de Darboux** do campo vetorial \mathfrak{X} se $\mathfrak{X}(p) = qp$, onde q é um polinômio em $\mathbb{C}[x, y, z]$ o qual é chamado de **cofator** de p .*

Definição 1.6. *Seja $I(x, y, z)$ uma IPL do sistema (9) e considere que suas derivadas podem ser escritas como*

$$I_x = RQ, \quad (10)$$

$$I_y = RP, \quad (11)$$

$$I_z = RN, \quad (12)$$

onde R é uma função Liouvillianas de (x, y, z) e Q, P, N são polinômios coprimos em

$\mathbb{C}[x, y, z]$. Então dizemos que I é um membro do conjunto L_S e que R é um **fator integrante** associado à integral primeira Liouvilliana I .

Definição 1.7. Seja $J(x, y, z)$ uma função Liouvilliana tal que¹⁰ $\langle \nabla, J \mathfrak{X} \rangle = 0$ (i.e., $\partial_x(JF) + \partial_y(JG) + \partial_z(JH) = 0$). Dizemos que J é um **multiplicador de Jacobi Liouvilliano** do sistema (9).

Definição 1.8. Seja $I \in L_S$ uma integral primeira do sistema (9) e que suas derivadas sejam escritas como em (10,11,12). O campo vetorial \mathfrak{J} definido por $\mathfrak{J} \equiv Q \partial_x + P \partial_y + N \partial_z$ será chamado **campo vetorial associado** à integral primeira I .

Definição 1.9. Seja γ uma 1-forma polinomial definida por $\gamma \equiv Q dx + P dy + N dz$ onde Q, P, N são polinômios coprimos em $\mathbb{C}[x, y, z]$. A função R é dita ser um **fator integrante** para a 1-forma γ se a 1-forma $R\gamma$ for exata.

Alguns comentários:

Observação 1.6. Se o sistema (9) apresenta uma integral primeira Liouvilliana I como descrita na definição 1.6 (isto é, $I \in L_S$), então a função R , que é um fator integrante para (9) associado com I , é também um fator integrante para a 1-forma $\gamma \equiv Q dx + P dy + N dz$.

Observação 1.7. Um multiplicador de Jacobi J do sistema (9) é comumente chamado de fator integrante. Isto, provavelmente, se deve a dois fatos:

1. Para duas dimensão ($n = 2$), ou seja, para campos vetoriais polinomiais no plano, um multiplicador de Jacobi J também é um fator integrante (do tipo descrito na observação (1.6)).
2. Para dimensões $n \geq 3$, um multiplicador de Jacobi J está diretamente associado ao fator integrante do sistema reduzido (obtido após usar os $n - 2$ integrais primeiras conhecidas - veja (36) para uma descrição detalhada do processo). Além disso, para tornar as coisas ainda mais confusas, o fator integrante do sistema reduzido ($n = 2$) é um multiplicador de Jacobi (conforme observado no item 1 desta observação).

Portanto, a fim de evitar termos ambíguos, neste trabalho usaremos o termo **fator integrante** para denotar fatores integrantes do tipo descrito na definição 1.6.

Observação 1.8. O campo vetorial \mathfrak{X} associado com o sistema (9) é chamado também de **operador de Darboux** associado com (9).

¹⁰ No que se segue, os operadores $\nabla(\cdot)$, $\langle \nabla, \cdot \rangle$, $\nabla \wedge \cdot$ significam **grad**, **div**, **rot**, respectivamente.

Observação 1.9. Se $I \in L_S$ é uma integral primeira do sistema (9) escrita como na definição 1.6, então a condição $\mathfrak{X}(I) = 0$ é equivalente a $\langle \mathfrak{X}, \mathfrak{J} \rangle = FQ + GP + HN = 0$.

Observação 1.10. Em princípio, para determinar uma integral primeira I , precisaríamos determinar o fator integrante R e os polinômios Q, P, N . Porém, como veremos, se pudermos calcular dois polinômios do conjunto Q, P, N , teremos à nossa disposição outras maneiras de determinar a integral primeira I . Uma delas é encontrar uma integral primeira Liouvilliana de um campo vetorial polinomial no plano (2D) ‘associado’ ao sistema (9) através de I .

Observação 1.11. Um dos campos vetoriais polinomiais 2D ‘associados’ ao sistema (9) (referido na observação acima) está diretamente relacionado à 1EDO racional (8) (veja o teorema 1.6 acima). A representação que usaremos é bem mais clara que a usada em (1).

A seguir, mostraremos como construir um campo vetorial polinomial 2D (2CVP) que “compartilha” a integral primeira Liouvilliana I com o campo vetorial \mathfrak{X} .

1.3 Campos vetoriais polinomiais no plano associados com \mathfrak{X}

Lema 1.1. Seja $I \in L_S$ uma integral primeira do sistema (9) escrita como na definição 1.6 e seja \mathfrak{J} o campo vetorial associado a I . Então, o campo vetorial definido por $\mathfrak{L} \equiv \nabla \wedge \mathfrak{J}$ obedece à condição $\langle \mathfrak{L}, \mathfrak{J} \rangle = 0$.

Prova. Da hipótese do teorema $I_x = RQ, I_y = RP, I_z = RN$. Então, podemos escrever \mathfrak{J} como $\frac{\nabla(I)}{R}$ que leva a

$$\mathfrak{L} = \nabla \wedge \mathfrak{J} = \frac{1}{R} \underbrace{\nabla \wedge \nabla(I)}_{=0} + \nabla \left(\frac{1}{R} \right) \wedge \nabla(I).$$

Logo, $\langle \mathfrak{L}, \mathfrak{J} \rangle = \langle \nabla \left(\frac{1}{R} \right) \wedge \nabla(I), \frac{\nabla(I)}{R} \rangle = 0$. \square

Lema 1.2. Seja $I \in L_S$ uma integral primeira Liouvilliana do sistema (9) e sejam $\mathfrak{X}, \mathfrak{J}$ e \mathfrak{L} campos vetoriais polinomiais definidos como acima. Então $\mathfrak{X} \wedge \mathfrak{L} = \frac{\mathfrak{X}(R)}{R} \mathfrak{J}$.

Prova. Do lema 1.1 temos que $\mathfrak{L} = \nabla \left(\frac{1}{R} \right) \wedge \nabla(I)$. Então ¹¹

¹¹ Usando a identidade $A \wedge (B \wedge C) = \langle A, C \rangle B - \langle A, B \rangle C$.

$$\mathfrak{X} \wedge \mathfrak{L} = \mathfrak{X} \wedge \left(\nabla \left(\frac{1}{R} \right) \wedge \nabla(I) \right) = \underbrace{\langle \mathfrak{X}, \nabla(I) \rangle}_{\mathfrak{X}(I)} \nabla \left(\frac{1}{R} \right) - \langle \mathfrak{X}, \nabla \left(\frac{1}{R} \right) \rangle \nabla(I).$$

$$\text{Como } \mathfrak{X}(I) = 0, \quad \mathfrak{X} \wedge \mathfrak{L} = -\langle \mathfrak{X}, -\frac{\nabla(R)}{R^2} \rangle \nabla(I) = \frac{\mathfrak{X}(R)}{R} \frac{\nabla(I)}{R} = \frac{\mathfrak{X}(R)}{R} \mathfrak{J}. \quad \square$$

Agora, podemos enunciar o seguinte teorema:

Teorema 1.7. *Seja $I \in L_S$ uma integral primeira do sistema (9) e sejam \mathfrak{X} , \mathfrak{J} e \mathfrak{L} campos vetoriais polinomiais definidos como acima. Então, para os campos vetoriais polinomiais no plano definidos por $\mathfrak{X}_1 \equiv N \partial_y - P \partial_z$, $\mathfrak{X}_2 \equiv -N \partial_x + Q \partial_z$, $\mathfrak{X}_3 \equiv P \partial_x - Q \partial_y$, as seguintes afirmações são válidas:*

- (a) $\mathfrak{X}_1(I) = \mathfrak{X}_2(I) = \mathfrak{X}_3(I) = 0$,
(b) $\frac{\mathfrak{X}_i(R)}{R} = -\langle \nabla, \mathfrak{X}_i \rangle = -\langle \mathfrak{L}, \partial_i \rangle$, ($i \in \{1, 2, 3\}$).

Prova. A afirmação (a) segue diretamente da definição:

$$\mathfrak{X}_1(I) = N \partial_y(I) - P \partial_z(I) = N R P - P R N = 0 ;$$

$$\mathfrak{X}_2(I) = -N \partial_x(I) + Q \partial_z(I) = -N R Q + Q R N = 0 ;$$

$$\mathfrak{X}_3(I) = P \partial_x(I) - Q \partial_y(I) = P R Q - Q R P = 0.$$

Agora vamos provar (b): temos que

$$\nabla \wedge \nabla(I) = \nabla \wedge (R \mathfrak{J}) = \begin{vmatrix} \hat{x} & \hat{y} & \hat{z} \\ \partial_x & \partial_y & \partial_z \\ R Q & R P & R N \end{vmatrix} = 0. \quad (13)$$

Isso implica em

$$\partial_y(R N) - \partial_z(R P) = R_y N + R N_y - R_z P - R P_z = 0 \Rightarrow \mathfrak{X}_1(R) + R(N_y - P_z) = 0,$$

$$\partial_z(R Q) - \partial_x(R N) = R_z Q + R Q_z - R_x N - R N_x = 0 \Rightarrow \mathfrak{X}_2(R) + R(Q_z - N_x) = 0,$$

$$\partial_x(R P) - \partial_y(R Q) = R_x P + R P_x - R_y Q - R Q_y = 0 \Rightarrow \mathfrak{X}_3(R) + R(P_x - Q_y) = 0.$$

Da definição de \mathfrak{L} e \mathfrak{X}_i , temos

$$\mathfrak{L} \equiv \nabla \wedge \mathfrak{J} = (N_y - P_z) \partial_x + (Q_z - N_x) \partial_y + (P_x - Q_y) \partial_z,$$

$$\mathfrak{X}_1 \equiv N \partial_y - P \partial_z \Rightarrow \langle \nabla, \mathfrak{X}_1 \rangle = N_y - P_z = \langle \mathfrak{L}, \partial_x \rangle,$$

$$\mathfrak{X}_2 \equiv -N \partial_x + Q \partial_z \Rightarrow \langle \nabla, \mathfrak{X}_2 \rangle = Q_z - N_x = \langle \mathfrak{L}, \partial_y \rangle,$$

$$\mathfrak{X}_3 \equiv P \partial_x - Q \partial_y \Rightarrow \langle \nabla, \mathfrak{X}_3 \rangle = P_x - Q_y = \langle \mathfrak{L}, \partial_z \rangle.$$

Substituindo essas definições nas equações obtidas a partir da identidade $\nabla \wedge \nabla(I) = 0$, a afirmação (b) segue diretamente. \square

Algumas observações:

Observação 1.12. Como $\langle \mathfrak{X}, \mathfrak{J} \rangle = FQ + GP + HN = 0$, a partir de qualquer par de polinômios no conjunto $\{Q, P, N\}$, podemos obter o terceiro.

Observação 1.13. $\mathfrak{X} \wedge \mathfrak{L} = \frac{\mathfrak{x}(R)}{R} \mathfrak{J} \Rightarrow \frac{\mathfrak{x}(R)}{R} \in \mathbb{C}[x, y, z]$.

Observação 1.14. $\langle \mathfrak{X}, \mathfrak{J} \rangle = \langle \mathfrak{L}, \mathfrak{J} \rangle = 0 \Rightarrow \langle [\mathfrak{X}, \mathfrak{L}], \mathfrak{J} \rangle = 0$.

Observação 1.15. Observe que os campos vetoriais associados \mathfrak{X}_i são 2CVPs e apresentam uma integral primeira Liouvilliana I (que é a mesma IPL admitida pelo campo vetorial \mathfrak{X}). Portanto, pelos resultados de Singer (9) e Christopher (10), eles apresentam um fator integrante Darbouxiano.

Observação 1.16. Devido ao fato de que os campos vetoriais polinomiais $\mathfrak{X}_1, \mathfrak{X}_2, \mathfrak{X}_3$ têm I como uma primeira integral Liouvilliana, podemos considerar o cálculo desta integral primeira I a partir do conhecimento de um desses campos vetoriais.

Definição 1.10. Seja $I \in L_S$ uma integral primeira do sistema (9). Então, as funções definidas por $S_k = P_i/P_j$ (onde $P_1 = Q, P_2 = P, P_3 = N, i, j, k = 1, \dots, 3$ and $i \leq j$) são chamadas **funções-S** associadas ao campo vetorial \mathfrak{X} através da integral primeira I .

Desenvolvendo o que foi apontado nas observações 1.15 e 1.16, podemos inferir os seguintes resultados:

Teorema 1.8. Seja $I \in L_S$ uma integral primeira do sistema (9). Então o 3CVP \mathfrak{X} tem um fator integrante Darbouxiano.

Prova. O caminho seguido para demonstrar este teorema vem dos resultados de Singer e Christopher (SC) para campos vetoriais polinomiais no plano¹²: A existência de uma integral primeira Liouvilliana admitida por um campo vetorial polinomial (no plano) implica na existência de um fator integrante Darbouxiano.

Da hipótese do teorema ($I \in L_S$) e da afirmação (b) do teorema 1.7 ($\frac{\mathfrak{x}_i(R)}{R} = -\langle \nabla, \mathfrak{X}_i \rangle$) segue diretamente que existem campos vetoriais polinomiais no plano ($\mathfrak{X}_1, \mathfrak{X}_2, \mathfrak{X}_3$) admitindo R (um fator de integrante do campo vetorial \mathfrak{X}) como fator integrante. Usando o resultado SC, temos que os campos vetoriais \mathfrak{X}_i (para qualquer $i \in \{1, 2, 3\}$) admitem funções de Darboux R_i como fatores integrantes. Logo, eles podem ser escritos como $R_i = \mathcal{F}_i(I) R$, onde $\mathcal{F}_i(I)$ são funções da integral primeira I . Portanto, os R_i também são fatores integrantes do campo vetorial \mathfrak{X} . Como cada R_i é um fator integrante Darbouxiano (em um dos pares (x, y) , (x, z) ou (y, z)) e todos são fatores integrantes para o campo

¹² Veja (9) e (10).

vetorial \mathfrak{X} , então $R_i = \mathcal{F}_{ij}(I) R_j$, onde $\mathcal{F}_{ij}(I)$ são funções da integral primeira I . Isso implica que: ou existe uma integral primeira Darbouxiana (uma vez que $\mathcal{F}_{ij}(I) = \frac{R_i}{R_j}$) ou um fator integrante Darbouxiano (no caso em que $\mathcal{F}_{ij}(I) = k_{ij}$, ou seja, $R_i = k_{ij} R_j$ onde k_{ij} são constantes) para o campo vetorial \mathfrak{X} . \square

1.4 Possíveis algoritmos para determinar integrais primeiras

O teorema 1.8 aponta possíveis estratégias para determinar uma integral primeira Liouvillianiana para o sistema (9), devido à existência de um fator integrante de Darboux. Mostraremos (na presente seção) que as condições que os polinômios $\{Q, P, N\}$ devem satisfazer podem ser escritas como equações diferenciais parciais de primeira ordem (1EDPs). Essas 1EDPs serão a base para os nossos métodos. A primeira delas é construída de forma semelhante à apresentada em (15) e, a seguir, iremos estabelecê-la.

1.4.1 Procedimento 1: Determinando um fator integrante

Vmos começar com o lema 1.2 (junto com a observação 1.13) para obter:

Teorema 1.9. *Seja $I \in L_S$ uma integral primeira do sistema (9) escrita como na definição 1.6 e seja \mathfrak{X} o campo vetorial associado ao sistema. Então, os polinômios P e N obedecem às seguintes 1EDPs:*

$$P \frac{\mathfrak{X}(R)}{R} + F \partial_y \left(\frac{G}{F} \right) P + F \partial_y \left(\frac{H}{F} \right) N + \mathfrak{X}(P) = 0, \quad (14)$$

$$N \frac{\mathfrak{X}(R)}{R} + F \partial_z \left(\frac{G}{F} \right) P + F \partial_z \left(\frac{H}{F} \right) N + \mathfrak{X}(N) = 0. \quad (15)$$

Prova. Para evitar cálculos desnecessários, indicaremos apenas o processo de obtenção do resultado desejado: A partir das hipóteses do teorema temos que $\langle \mathfrak{X}, \mathfrak{J} \rangle = FQ + GP + HN = 0$. Resolvendo a equação $FQ + GP + HN = 0$ para o polinômio Q e substituindo o resultado na equação vetorial $\mathfrak{X} \wedge \mathfrak{L} = \frac{\mathfrak{X}(R)}{R} \mathfrak{J}$ (lema 1.2), obtemos (para os componentes y e z , após reorganizar os termos) as condições expressas pelas 1EDPs (14,15) para os polinômios P e N . \square

Observação 1.17. *A condição expressa pela equação que resulta da componente x é dependente das equações (14,15).*

Observação 1.18. *Existem condições análogas para os outros dois pares $\{Q, N\}$ e $\{Q, P\}$ que resultam da resolução da equação $FQ + GP + HN = 0$ para P e N , respectivamente, e da substituição dos resultados na equação $\mathfrak{X} \wedge \mathfrak{L} = \frac{\mathfrak{X}(R)}{R} \mathfrak{J}$. Esses pares de equações são:*

$$Q \frac{\mathfrak{X}(R)}{R} + G \partial_x \left(\frac{F}{G} \right) Q + G \partial_x \left(\frac{H}{G} \right) N + \mathfrak{X}(Q) = 0, \quad (16)$$

$$N \frac{\mathfrak{X}(R)}{R} + G \partial_z \left(\frac{F}{G} \right) Q + G \partial_z \left(\frac{H}{G} \right) N + \mathfrak{X}(N) = 0, \quad (17)$$

e

$$Q \frac{\mathfrak{X}(R)}{R} + H \partial_x \left(\frac{F}{H} \right) Q + F \partial_x \left(\frac{G}{H} \right) P + \mathfrak{X}(Q) = 0, \quad (18)$$

$$P \frac{\mathfrak{X}(R)}{R} + H \partial_y \left(\frac{F}{H} \right) Q + F \partial_y \left(\frac{G}{H} \right) P + \mathfrak{X}(P) = 0. \quad (19)$$

Observação 1.19. *Uma vez que o fator integrante R é um fator integrante de Darboux (ou seja, $R = \exp(Z_0) \prod_i p_i^{n_i}$, onde $Z_0 = A/B$ e A, B são polinômios coprimos) e $\frac{\mathfrak{X}(R)}{R}$ é um polinômio, isso implica que p_i e B são polinômios de Darboux de \mathfrak{X} (Para uma prova, consulte (14, 15)).*

Usando a estrutura de R , podemos escrever $\frac{\mathfrak{X}(R)}{R}$ como

$$\frac{\mathfrak{X}(R)}{R} = \frac{\mathfrak{X}(e^{Z_0} \prod_i p_i^{n_i})}{e^{Z_0} \prod_i p_i^{n_i}} = \mathfrak{X} \left(\frac{A}{B} \right) + \sum_i n_i \underbrace{\frac{\mathfrak{X}(p_i)}{p_i}}_{q_i}, \quad (20)$$

onde os q_i são os cofatores dos polinômios de Darboux p_i . Substituindo (20) nas equações (14,15) e multiplicando ambas as equações por B , teremos

$$P \left(\mathfrak{X}(A) - A \frac{\mathfrak{X}(B)}{B} + B \sum_i n_i q_i \right) + B \left(F \partial_y \left(\frac{G}{F} \right) P + F \partial_y \left(\frac{H}{F} \right) N + \mathfrak{X}(P) \right) = 0, \quad (21)$$

$$N \left(\mathfrak{X}(A) - A \frac{\mathfrak{X}(B)}{B} + B \sum_i n_i q_i \right) + B \left(F \partial_z \left(\frac{G}{F} \right) P + F \partial_z \left(\frac{H}{F} \right) N + \mathfrak{X}(N) \right) = 0. \quad (22)$$

Como B é um polinômio de Darboux de \mathfrak{X} , podemos escrevê-lo como $\sum_j p_{b_j}^{m_j}$, onde p_{b_j} são polinômios de Darboux irreduzíveis e os m_j são inteiros positivos. Usando este conhecimento nas equações (21,22), podemos construir um possível (semi) algoritmo para determinar um fator integrante R para o sistema (9):

Procedimento 1 (esboço): (P_{RO1})

1. Calcule os polinômios de Darboux p_i de \mathfrak{X} (o 3CVP associado ao sistema (9)) até um certo grau.
2. Defina o grau (inicial) para os polinômios A e B . Construa um polinômio A_c com coeficientes indeterminados e um polinômio B_c com os polinômios de Darboux p_i encontrados na etapa anterior. Observe que, após definir o grau (inicial) de B , ficamos com um conjunto finito de possibilidades $\{\{m_j\}_1, \{m_j\}_2, \dots, \{m_j\}_k\}$ para os inteiros m_j .
3. Defina os graus (iniciais) para os polinômios Q , P e N . Construa os polinômios Q_c , P_c e N_c com coeficientes indeterminados e substitua na equação $FQ + GP + HN = 0$. Colete a equação polinomial resultante nas variáveis (x, y, z) obtendo um conjunto de equações lineares para os coeficientes indeterminados. Resolva este conjunto de equações e substitua a solução nos candidatos Q_c , P_c e N_c .
4. Para cada conjunto possível de $\{m_j\}$, substitua os candidatos em uma das equações de base ((21) ou (22)) e colete a equação polinomial resultante nas variáveis (x, y, z) obtendo um conjunto de equações para os coeficientes indeterminados. Tente resolver este conjunto de equações. Se uma solução for encontrada, construa um fator integrante R e determine as derivadas $I_x = RQ$, $I_y = RP$, $I_z = RN$ de uma integral primeira I .
5. De I_x , I_y , I_z , calcule uma integral primeira I por quadraturas.

Observação 1.20. *Os Termos $P\left(\mathfrak{X}(A) - A\frac{\mathfrak{X}(B)}{B} + B\sum n_i q_i\right)$ e $N\left(\mathfrak{X}(A) - A\frac{\mathfrak{X}(B)}{B} + B\sum n_i q_i\right)$ presente nas equações (21,22) dão origem a monômios quadráticos nos coeficientes indeterminados. As equações para os coeficientes podem se tornar (na prática) muito difíceis de resolver (computacionalmente falando). Podemos usar o fato de que $\mathfrak{X}\left(\frac{A}{B}\right)$ é um polinômio para determinar os coeficientes do candidato A_c antes de substituí-lo nas equações (21,22). Isso divide o problema de resolver o sistema algébrico para coeficientes em duas partes, uma das quais é linear. Portanto, podemos melhorar muito nosso algoritmo adicionando a seguinte etapa entre as etapas 2 e 3:*

Etapa intermediária - Construa um polinômio \mathcal{P} com coeficientes indeterminados. Para cada conjunto possível de $\{m_j\}$, colete a equação $\mathfrak{X}(A_c) - A_c\frac{\mathfrak{X}(B_c)}{B_c} - B_c\mathcal{P} = 0$ nas variáveis (x, y, z) obtendo um conjunto de equações para os coeficientes indeterminados (que é linear). Tente resolver este conjunto de equações. Se uma solução for encontrada, podemos determinar um candidato A_{cl} (para este conjunto particular $\{m_j\}_l$).

Vamos ver este procedimento na prática:

Exemplo 1.1. – Considere o sistema representado pelo campo vetorial

$$\mathfrak{X} = (x^3y^2 - yx^3 + 2xz)\partial_x + (-2x^2y^3 + 2y^2z + 2y)\partial_y + (2x^2y^2z + 2x^2y + 2z^2)\partial_z. \quad (23)$$

- Passo 1 – Calcule os polinômios de Darboux (PDs) de \mathfrak{X} : Até o grau 3, os PDs encontrados são $p_1 = x$, $p_2 = y$ e $p_3 = x^2y - z$, e os respectivos cofatores são $q_1 = x^2y^2 - x^2y + 2z$, $q_2 = -2x^2y^2 + 2yz + 2$ e $q_3 = -2x^2y + 2z$.
- Passo 2 – Para grau 3, os conjuntos possíveis para o polinômio B são $x^3 \rightarrow \{m_1 = 3, m_2 = 0, m_3 = 0\}$, $x^2y \rightarrow \{m_1 = 2, m_2 = 1, m_3 = 0\}$, $xy^2 \rightarrow \{m_1 = 1, m_2 = 2, m_3 = 0\}$, $y^3 \rightarrow \{m_1 = 0, m_2 = 3, m_3 = 0\}$, $x^2y - z \rightarrow \{m_1 = 0, m_2 = 0, m_3 = 1\}$.¹³
- Passo intermediário – Para $B = x^2y - z$, podemos fazer $A = \text{constante} = 1$, $\mathcal{P} = 2$.
- Passos 3 e 4 – Para os graus 7, 8, e 5 para os polinômios Q , P e N , podemos obter:

$$Q = -2y^2x(x^2y^2 - yz - 1), \quad (24)$$

$$P = -x^4y^3 + x^4y^2 + y^2x^2z - 2x^2yz + x^2y + z^2, \quad (25)$$

$$N = y(x^2y^2 - yz - 1), \quad (26)$$

$$R = -\frac{e^{(x^2y-z)^{-1}}}{(x^2y - z)^2 y^2}. \quad (27)$$

- Passo 5 – Finalmente, podemos obter

$$I = \frac{e^{(x^2y-z)^{-1}}}{y} + Ei\left(1, -(x^2y - z)^{-1}\right) \quad (28)$$

Observação 1.21. *A obtenção do polinômio de Darboux $p_3 = x^2y - z$ no passo 1 e o passo 4 são de longe os processos mais custosos (do ponto de vista computacional) do procedimento como um todo. Discutiremos o tempo de CPU e o uso da memória do computador no capítulo 5.*

Observação 1.22. *O tempo gasto no passo 4 (ao resolver para os coeficientes) depois de substituir os candidatos em uma das equações de base ((21) ou (22)) não depende (neste exemplo particular) de qual equação base usamos (o tempo de CPU e o uso de memória são praticamente os mesmos nos dois casos).*

¹³ Para os graus 1 e 2 nenhuma solução foi encontrada.

1.4.2 Procedimento 2: Calculando \mathfrak{X}_i

A outra maneira (citada acima) de calcular uma primeira integral para um 3CVP é usar o fato expresso no teorema 1.7, ou seja, que os campos vetoriais associados \mathfrak{X}_i compartilham a integral primeira I do sistema (9) com \mathfrak{X} . Para fazer isso, podemos usar as equações $\langle \mathfrak{X}, \mathfrak{J} \rangle = \langle \mathfrak{L}, \mathfrak{J} \rangle = 0$ para obter os polinômios Q , P e N . A partir deles, podemos construir os campos vetoriais associados \mathfrak{X}_i e usar um deles para construir a integral primeira I usando, por exemplo, o método da função-S (veja (3)), onde podemos encontrar a integral primeira I de \mathfrak{X} construindo uma integral primeira para um dos 2CVPs \mathfrak{X}_i . Então, começamos com o resultado:

Teorema 1.10. *Seja $I \in L_S$ uma integral primeira do sistema (9) de modo que suas derivadas sejam escritas como em (10, 11, 12) e seja \mathfrak{X} o campo vetorial associado ao sistema. Então, os polinômios Q , P , N obedecem às seguintes 1EDPs:*

$$\mathfrak{X} \left(\frac{P_j}{P_k} \right) = F_i \partial_k \left(\frac{F_j}{F_i} \right) \left(\frac{P_j}{P_k} \right)^2 + F_i \left(\partial_k \left(\frac{F_k}{F_i} \right) - \partial_j \left(\frac{F_j}{F_i} \right) \right) \left(\frac{P_j}{P_k} \right) - F_i \partial_j \left(\frac{F_k}{F_i} \right), \quad (29)$$

onde $P_1 = Q$, $P_2 = P$, $P_3 = N$, $F_1 = F$, $F_2 = G$, $F_3 = H$, $\partial_i \equiv \frac{\partial}{\partial x_i}$, $x_1 = x$, $x_2 = y$, $x_3 = z$, e (i, j, k) é qualquer permutação do conjunto $\{1, 2, 3\}$.

Prova. A partir das hipóteses do teorema, temos que $\langle \mathfrak{X}, \mathfrak{J} \rangle = \langle \mathfrak{L}, \mathfrak{J} \rangle = 0$, (onde $\mathfrak{L} \equiv \nabla \wedge \mathfrak{J}$). Essas condições implicam nas seguintes equações:

$$\begin{aligned} \langle \mathfrak{X}, \mathfrak{J} \rangle &= FQ + GP + HN = 0, \\ \langle \mathfrak{L}, \mathfrak{J} \rangle &= (N_y - P_z)Q + (Q_z - N_x)P + (P_x - Q_y)N = 0. \end{aligned}$$

Como na prova do teorema 1.9 (para evitar cálculos desnecessários), indicamos apenas o caminho: resolvemos a equação $FQ + GP + HN = 0$ para um dos polinômios Q , P , N e substituímos a solução na equação $(N_y - P_z)Q + (Q_z - N_x)P + (P_x - Q_y)N = 0$. Obtemos (dependendo de qual polinômio escolhermos) as condições expressas pelas 1EDPs (29) para as outras duas funções polinomiais. \square

Como as equações (29) são escritas em termos de apenas um par de polinômios, podemos usá-los diretamente para construir um algoritmo para determinar os campos vetoriais associados. Um possível algoritmo seria:

Procedimento 2 (esboço): (P_{RO2})

1. Defina o grau (inicial) para os polinômios P_i e P_j . Construa dois polinômios P_{ic} e P_{jc} com coeficientes indeterminados e substitua-os na equação (29). Colete a equação polinomial resultante nas variáveis (x, y, z) obtendo um conjunto de equações quadráticas para os coeficientes indeterminados. Resolva este conjunto de equações e substitua a solução nos candidatos P_{ic} e P_{jc} para obter P_i e P_j .

2. Use algum procedimento (o algoritmo Chèze-Combote (17), o método da função-S (3), etc) para encontrar uma integral primeira \mathcal{I}_k para o campo vetorial $\mathfrak{X}_k \equiv P_j \partial_i - P_i \partial_j$ (onde (i, j, k) é qualquer permutação do conjunto $\{1, 2, 3\}$).
3. Substitua $I = \mathcal{F}(x_k, \mathcal{I}_k)$ em $\mathfrak{X}(I) = 0$ para obter a integral primeira I (veja (3)).

Agora, vamos ver como este segundo procedimento se comporta quando aplicado ao exemplo anterior (exemplo 1.1 – campo vetorial (23)):

- Passo 1 – Para os graus 2 e 0 para os candidatos Q_c e N_c encontramos $\frac{Q}{N} = -2xy$. Usando a equação $FQ + GP + HN = 0$ obtemos:

$$Q = -2y^2x(x^2y^2 - yz - 1),$$

$$P = -x^4y^3 + x^4y^2 + x^2y^2z - 2x^2yz + x^2y + z^2,$$

$$N = y(x^2y^2 - yz - 1).$$

- Passo 2 – $\mathfrak{X}_1 = N \partial_y - P \partial_z$; resolvendo a 1EDO

$$\frac{dz}{dy} = \frac{x^4y^3 - x^4y^2 - x^2y^2z + 2x^2yz - x^2y - z^2}{y(x^2y^2 - yz - 1)}, \quad (30)$$

(associada com \mathfrak{X}_1) temos

$$\mathcal{I} = Ei(1, -(x^2y - z)^{-1}) + \frac{e^{(x^2y - z)^{-1}}}{y}. \quad (31)$$

- Passo 3 – Não precisamos usar esta etapa porque $\mathcal{I} = I$, ou seja, neste exemplo \mathcal{I} já é a integral primeira que estávamos procurando.

Observação 1.23. *As equações (29), bem como os pares de equações (14, 15), (16, 17) e (18, 19) são equações quadráticas nos indeterminados, implicando que as equações para os coeficientes indeterminados (em cada um dos procedimentos) também são quadráticas. Dessa forma, nenhuma das abordagens tem uma vantagem geral sobre a outra, ou seja, haverá casos em que um dos métodos será provavelmente muito mais vantajoso do que o outro e vice-versa.*

Observação 1.24. *Como $\frac{Q}{N} = -2xy$, o procedimento encontra Q/N quase instantaneamente. O exemplo 1.1 mostra um dos casos descritos na observação acima: uma das funções-S é muito simples e, portanto, o procedimento 2 leva vantagem em relação ao procedimento 1. No capítulo 5 apresentaremos mais alguns casos comparativos e comentaremos (brevemente) as possíveis razões para essas diferenças de desempenho entre os algoritmos.*

2 1EDOS COM UMA FUNÇÃO ELEMENTAR

Neste capítulo, vamos apresentar um caso em que o procedimento 2 (veja seção 1.4.2) pode se tornar muito mais eficiente: 1EDOs que apresentam uma função elementar. Na primeira seção mostraremos uma equivalência entre um certo tipo de 1EDOs e campos vetoriais polinomiais em três variáveis. Na segunda, generalizaremos o método construído em (4) (construiremos uma versão aprimorada do procedimento 2). Na terceira, mostramos que, assim como na tese (4), também para esta generalização que fizemos, podemos encontrar os campos associados \mathfrak{X}_i linearmente.

Começamos definindo o tipo de 1EDO que podemos resolver usando uma associação com um 3CVP.

Definição 2.1. *Considere uma 1EDO que pode ser escrita na forma*

$$y' = \frac{dy}{dx} = \phi(x, y, \theta) = \frac{M_0(x, y, \theta)}{N_0(x, y, \theta)} \quad (32)$$

onde ϕ é uma função racional de (x, y, θ) , i.e., M_0, N_0 são polinômios coprimos em $\mathbb{C}[x, y, \theta]$. Além disso, a função $\theta(x, y)$ está em uma extensão elementar \mathcal{E} do corpo diferencial $\mathbb{C}(x, y)$ tal que a torre definidora¹⁴ de \mathcal{E} tem comprimento 1, ou seja, a torre definidora é $\mathbb{C}(x, y) = K_0 \subset K_1 = \mathcal{E}$. Suponha ainda que a solução geral da 1EDO (32) pode ser escrita (na forma implícita) como $I(x, y, \theta) = c$, onde I é uma função Liouvilliana de (x, y, θ) tal que suas derivadas parciais (relacionados às variáveis x, y e θ) possam ser expressas como

$$I_x = RQ, \quad (33)$$

$$I_y = RP, \quad (34)$$

$$I_\theta = RN, \quad (35)$$

onde R é uma função Liouvilliana de (x, y, θ) e $Q, P, N \in \mathbb{C}[x, y, \theta]$. Então dizemos que a 1EDO (32) é um membro do conjunto \mathcal{L}_S .

Podemos escrever um sistema 2D associado¹⁵ com a 1EDO (32) como

$$\begin{cases} \dot{x} = N_0(x, y, \theta), \\ \dot{y} = M_0(x, y, \theta). \end{cases} \quad (36)$$

¹⁴ Veja a definição em (9).

¹⁵ No sentido de que a função Liouvilliana I que representa a solução geral de (32) é uma integral primeira para o sistema (36).

Observação 2.1. Das condições da 1EDO (32), i.e., do fato de que as derivadas de I podem ser expressas como (33,34,35) e que as derivadas de θ estão em \mathcal{E} , podemos, a partir do sistema (36), definir um campo vetorial polinomial tridimensional (3CVP) cuja integral primeira $I(x, y, z)$ pode ser obtida da integral primeira do sistema (36) simplesmente substituindo θ por z .

O processo pode ser descrito pelas seguintes etapas:

1. Calcular o campo vetorial 2D \mathfrak{X}_{2D} associado com o sistema (36):

$$\mathfrak{X}_{2D} = N_0 \partial_x + M_0 \partial_y.$$

2. Aplicar \mathfrak{X}_{2D} na integral primeira I usando a regra da cadeia:

$$\begin{aligned} \mathfrak{X}_{2D}(I) &= N_0 \partial_x(I) + M_0 \partial_y(I) = N_0 \partial_x(I) + N_0 \partial_\theta(I) \theta_x + M_0 \partial_y(I) + M_0 \partial_\theta(I) \theta_y = 0 \\ \Rightarrow \mathfrak{X}_{2D}(I) &= N_0 \partial_x(I) + M_0 \partial_y(I) + (N_0 \theta_x + M_0 \theta_y) \partial_\theta(I) = 0. \end{aligned}$$

3. Como θ_x and θ_y são funções racionais podemos multiplicar a equação $\mathfrak{X}_{2D}(I) = 0$ por¹⁶ $\delta_\theta \equiv \text{mmc}(\text{den}(N \theta_x), \text{den}(M \theta_y))$ obtendo

$$\delta_\theta \mathfrak{X}_{2D}(I) = \delta_\theta N_0 \partial_x(I) + \delta_\theta M_0 \partial_y(I) + \delta_\theta (N_0 \theta_x + M_0 \theta_y) \partial_\theta(I) = 0.$$

4. Defina $\mathfrak{X} \equiv T(\delta_\theta \mathfrak{X}_{2D})$, onde $T = \{\theta = z\}$, i.e., defina $F \equiv T(\delta_\theta N_0)$, $G \equiv T(\delta_\theta M_0)$, $H \equiv T(\delta_\theta (N_0 \theta_x + M_0 \theta_y))$ e $\mathfrak{X} \equiv F \partial_x + G \partial_y + H \partial_z$.

Definição 2.2. Considere uma 1EDO da forma (32) em \mathcal{L}_S . Dizemos que o campo vetorial polinomial 3D definido por $\mathfrak{X} \equiv T(\delta_\theta \mathfrak{X}_{2D})$ é o 3CVP associado à 1EDO (32).

Observação 2.2. Observe que, pela forma como foi construído, o campo vetorial \mathfrak{X} tem, como integrais primeiras, $I(x, y, z)$ (que, fazendo a substituição $z \rightarrow \theta(x, y)$ torna-se a integral primeira de \mathfrak{X}_{2D}) e $z - \theta(x, y)$. Então, se pudermos encontrar a IPL $I(x, y, z)$ para o campo vetorial \mathfrak{X} , teremos encontrado a IPL de \mathfrak{X}_{2D} (e, portanto, a solução geral da 1EDO (32)).

A partir da observação acima (observação 2.2), podemos inferir que uma maneira possível de encontrar uma integral primeira Liouvillianiana para o sistema (36) (ou equivalentemente, para determinar uma solução geral Liouvillianiana para uma 1EDO em \mathcal{L}_S) é realizar a seguinte sequência de etapas:

¹⁶ Onde *den* significa denominador.

Procedimento 3 (esboço): (P_{RO3})

1. Determinar o campo vetorial polinomial 3D associado \mathfrak{X} .
2. Encontrar a integral primeira Liouvilliana usando os algoritmos P_{RO1} ou P_{RO2} .
3. Substituir z por $\theta(x, y)$.

Alguns comentários:

- Para o caso em que o campo vetorial \mathfrak{X}_{2D} apresenta uma integral primeira elementar, o uso do procedimento 1 (P_{RO1}) seria basicamente o método usado por Y.-K. Man em (37). Nesse caso o procedimento 2 (P_{RO2}) fornece uma forma alternativa para calcular a integral primeira.
- No caso onde \mathfrak{X}_{2D} apresenta uma integral primeira Liouvilliana não elementar, os dois algoritmos (P_{RO1} e P_{RO2}) são uma extensão do procedimento sugerido por Shtokhamer *et al* em (38) (veja (37)).

A seguir, vamos mostrar que o procedimento 2 (o caminho que elimina o termo $\frac{\mathfrak{X}(R)}{R}$ das 1EDPs) é a chave para (no caso de campos vetoriais que representam 1EDOs em \mathcal{L}_S) fazermos um aprimoramento do método, aprimoramento este que nos permitirá tornar linear o cômputo dos campos vetoriais 2D associados. Vamos começar com o seguinte exemplo:

Considere a 1EDO dada por

$$\frac{dy}{dx} = \frac{e^x ((x^2 + 2x)(e^x)^2 - (2x^2y + x^2 + 4xy)e^x + x^2y^2 + 2xy^2 + y)}{(e^x)^2 + (-x^2 - 2y)e^x + y^2 + y}. \quad (37)$$

Vamos seguir as etapas do procedimento 3 (P_{RO3}) e usar P_{RO2} (na etapa 2) depois de determinar o 3CVP \mathfrak{X} associado com a 1EDO (37):

1. $\mathfrak{X} = F\partial x + G\partial y + H\partial z$, onde

$$F = zx^2 - y^2 + 2zy - z^2 - y, \quad (38)$$

$$G = -z(x^2y^2 - 2x^2yz + z^2x^2 - zx^2 + 2xy^2 - 4xyz + 2xz^2 + y), \quad (39)$$

$$H = z(zx^2 - y^2 + 2zy - z^2 - y). \quad (40)$$

2. P_{RO2} :

- (a) Para os graus $d_Q = 4$, $d_P = 3$ e $d_N = 4$, temos:

$$Q = 2xz(y - z)^2, \quad (41)$$

$$P = zx^2 - y^2 + 2yz - z^2 - y, \quad (42)$$

$$N = x^2y^2 - 2x^2yz + x^2z^2 - zx^2 + y. \quad (43)$$

(b) Usando o método da função-S, obtemos a seguinte integral primeira:

$$I = \frac{-\ln(zx^2 - y)z + \ln(zx^2 - y)y - 1}{y - z}. \quad (44)$$

3. Substituindo z por θ (no caso, $\theta = e^x$) obtemos a solução geral da 1EDO (37):

$$\frac{-\ln(e^x x^2 - y)e^x + \ln(e^x x^2 - y)y - 1}{y - e^x} = c. \quad (45)$$

Observação 2.3. Comparando as componentes (F, G, H) de \mathfrak{X} com as componentes (Q, P, N) de \mathfrak{J} , podemos notar que:

$$H = zF, \quad F = P. \quad (46)$$

Ou seja, F e H só diferem pela variável multiplicativa z e, além disso, a segunda equação nos fornece informações sobre um dos polinômios (neste caso, P) ligados à derivada da integral primeira I (lembre que $I_y = RP$) e, em princípio, desconhecido.

Observação 2.4. Se soubéssemos (a priori) que $P = F$, calcular N (ou Q) usando o algoritmo P_{RO2} seria um processo muito mais eficiente, pois, nesse caso, apenas um dos polinômios (N ou Q) seria desconhecido na equação base relevante (29) (onde $P_i = P$, $P_j = N$ ou $P_i = Q$, $P_j = P$).

Agora, considere a 1EDO

$$\frac{dy}{dx} = \frac{\left((x^4 + 2x^2y - 2x^2 + y^2) \ln\left(\frac{x}{y}\right) + x^4 + 2x^2y + 2xy + y^2 \right) y}{\left(x^5 + x^4y + 2x^3y + 2x^2y^2 + \ln\left(\frac{x}{y}\right) xy + xy^2 + y^3 - y^2 \right)}. \quad (47)$$

e vamos repetir o procedimento que aplicamos ao exemplo anterior:

1. $\mathfrak{X} = F\partial x + G\partial y + H\partial z$, onde

$$F = (x^5 + x^4y + 2x^3y + 2x^2y^2 + xy^2 + xyz + y^3 - y^2)x, \quad (48)$$

$$G = (x^4z + x^4 + 2zx^2y + 2x^2y - 2zx^2 + zy^2 + 2xy + y^2)yx, \quad (49)$$

$$H = -(x^4 + 2x^2y - 2x^2 + y^2 - y)(xz - y). \quad (50)$$

2. P_{RO2} :

(a) Para os graus $d_Q = 5$, $d_P = 4$ e $d_N = 5$ temos que

$$Q = x^4z + 2zx^2y - 2zx^2 + zy^2 + 2xy, \quad (51)$$

$$P = -x^4 - 2x^2y - xz - y^2 + y, \quad (52)$$

$$N = (x^2 + y)^2 x. \quad (53)$$

(b) Usando o método da função-S, obtemos a seguinte integral primeira:

$$I = e^{(x^2+y)^{-1}} (xz - y). \quad (54)$$

3. Substituindo z por $\ln\left(\frac{x}{y}\right)$ temos a solução geral da 1EDO (47):

$$e^{(x^2+y)^{-1}} \left(\ln\left(\frac{x}{y}\right) x - y \right) = c. \quad (55)$$

Observação 2.5. *Desta vez, nenhum dos componentes de \mathfrak{J} é um componente de \mathfrak{X} . A diferença entre os exemplos 1 e 2 reside no fato de que, no exemplo 1, a função elementar presente na 1EDO (37) é uma função de apenas uma das variáveis (e^x , neste caso), enquanto no exemplo 2 a função θ na 1EDO (47) é uma função das duas variáveis ($\ln\left(\frac{x}{y}\right)$).*

Na próxima seção, mostraremos que o fato observado ($F = P$) na 1EDO (37) não é apenas um acaso. Podemos, para uma grande classe de 1EDOs, conhecer um dos polinômios (Q, P, N) antes de usar as equações (29). Este resultado (descrito na sentença anterior) não é uma novidade; ele aparece na tese (4). A novidade é que podemos generalizar o resultado principal (que mostraremos na seção a seguir) para incluir funções algébricas.

2.1 Um aprimoramento de P_{RO2} para uma classe de 1EDOs

Vamos começar definindo uma classe de 1EDOs para a qual podemos tornar o algoritmo P_{RO2} mais eficiente:

Definição 2.3. *Seja ϑ_1 uma 1EDO em \mathcal{L}_S e seja $\theta = \theta(r)$, onde $r \in \mathbb{C}(x, y)$. Seja T a transformação $\{u = r(x, y), v = y\}$ (ou $\{u = x, v = r(x, y)\}$) e seja $T^{-1} = \{x = r_1(u, v), y = v\}$ (resp. $T^{-1} = \{x = u, y = r_2(u, v)\}$) a sua inversa. Se r_i está em $\mathbb{C}(x, y)$ ($i = 1$ ou 2) então dizemos que ϑ_1 é um membro do conjunto \mathcal{L}_{S_r} .*

Observação 2.6. *Se uma 1EDO está em \mathcal{L}_{S_r} , então, com uma transformação simples, podemos levá-la em outra 1EDO da mesma classe onde, agora, θ é uma função de apenas uma das variáveis.*

Agora vamos mostrar que a igualdade $F = P$ no 1EDO (37) não é um mero acaso.

Teorema 2.1. *Considere uma 1EDO (32) em \mathcal{L}_S e seja \mathfrak{X} seu 3CVP associado. Assuma também, que os campos vetoriais \mathfrak{X} e \mathfrak{I} são definidos como acima. Então, as seguintes afirmações são verdadeiras:*

- a) Se $\theta = e^x$ então $F|P$, $H = zF$ e $G|(Q + zN)$.
- b) Se $\theta = \ln(x)$ então $H|P$, $F = xH$ e $G|(xQ + N)$.
- c) Se $\theta = e^y$ então $G|Q$, $H = zG$ e $F|(P + zN)$.
- d) Se $\theta = \ln(y)$ então $H|Q$, $G = yH$ e $F|(yP + N)$.

Prova.

a): Se $\theta = e^x$ então $\theta_x = z$ e $\theta_y = 0$ assim

$$F = N_0(x, y, z), \quad G = M_0(x, y, z), \quad H = zN_0(x, y, z). \quad (56)$$

Temos que $\mathfrak{X}(I) = FI_x + GI_y + HI_z = FRQ + GRP + HRN$ (por hipótese) $= R(FQ + GP + HN) = 0$. Substituindo (56) e notando que R não é nulo, podemos escrever $N_0Q + M_0P + zN_0N = 0$ implicando que $N_0(Q + zN) = -M_0P$. Portanto, podemos escrever

$$Q + zN = -\frac{M_0P}{N_0} \quad \text{e} \quad P = -\frac{N_0(Q + zN)}{M_0}. \quad (57)$$

Uma vez que M_0 e N_0 são coprimos, temos que $N_0|P \Rightarrow F|P$ e $M_0|(Q + zN) \Rightarrow G|(Q + zN)$.

b): Se $\theta = \ln(x) \Rightarrow \theta_x = 1/x$ e $\theta_y = 0$. Temos dois casos:

$$F = xN_0(x, y, z), \quad G = xM_0(x, y, z), \quad H = N_0(x, y, z), \quad (58)$$

se x não é um fator de N_0 ou

$$F = x^n Pol(x, y, z), \quad G = M_0(x, y, z), \quad H = x^{n-1} Pol(x, y, z), \quad (59)$$

onde $Pol = N_0/x^n$ é um polinômio, n é um inteiro positivo e x e Pol são coprimos.

Primeiro caso (x e N_0 são coprimos): $N_0(xQ + N) = -xM_0P$. Portanto, $N_0|P \Rightarrow H|P$ e $xM_0|(xQ + N) \Rightarrow G|(xQ + N)$.

Segundo caso ($x|N_0$): $x^{n-1}Pol(xQ + N) = -M_0P$. Uma vez que M_0 e N_0 são coprimos, temos que $x^{n-1}Pol (= H)$ e M_0 são coprimos. Portanto, $H|P$ e $G|(xQ + N)$.

c): Se $\theta = e^y$ então $\theta_x = 0$ e $\theta_y = z$ assim

$$F = N_0(x, y, z), \quad G = M_0(x, y, z), \quad H = zM_0(x, y, z). \quad (60)$$

Podemos escrever $N_0 Q + M_0 P + z M_0 N = 0$ implicando que $N_0 Q = -M_0 (P + z N)$. Portanto, podemos escrever

$$P + z N = -\frac{N_0 Q}{M_0} \text{ and } Q = -\frac{M_0 (P + z N)}{N_0}. \quad (61)$$

Uma vez que M_0 e N_0 são coprimos, temos que $N_0 | (P + z N) \Rightarrow F | (P + z N)$ e $M_0 | Q \Rightarrow G | Q$.

d): Se $\theta = \ln(y) \Rightarrow \theta_x = 0$ e $\theta_y = 1/y$. Temos dois casos:

$$F = y N_0(x, y, z), \quad G = y M_0(x, y, z), \quad H = M_0(x, y, z), \quad (62)$$

se y não é um fator de M_0 ou

$$F = N_0(x, y, z), \quad G = y^n Pol(x, y, z), \quad H = y^{n-1} Pol(x, y, z), \quad (63)$$

onde $Pol = M_0/y^n$ é um polinômio, n é um inteiro positivo e y e Pol são coprimos.

Primeiro caso (y e M_0 são coprimos): $N_0 y Q = -M_0 (y P + N)$. Portanto, $M_0 | Q \Rightarrow H | Q$ e $y N_0 | (y P + N) \Rightarrow F | (y P + N)$.

Segundo caso ($y | M_0$): $y^{n-1} Pol (y P + N) = -N_0 Q$. Uma vez que M_0 e N_0 são coprimos, temos que $y^{n-1} Pol (= H)$ e N_0 são coprimos. Portanto, $H | Q$ e $F | (y P + N)$. \square

Observação 2.7. *A partir da definição 2.3 e teorema 2.1 temos que para qualquer 1EDO $\in \mathcal{L}_{S_r}$ podemos realizar uma transformação de variável que leva $\theta(x, y)$ em $\exp(x)$, $\ln(x)$, $\exp(y)$ ou $\ln(y)$. Portanto, para qualquer 1EDO em \mathcal{L}_{S_r} , podemos melhorar o algoritmo P_{RO3} (que usa o algoritmo P_{RO2} na segunda etapa), já que para esta classe precisamos calcular apenas um polinômio nas equações (29).*

O teorema 2.1 foi apresentado em (4). Podemos aprimorar o resultado usando o fato que $\theta(x, y)$ é uma função algébrica ou transcendental de uma função racional r de (x, y) cuja inversa é racional. Portanto, podemos usar uma transformação de coordenadas para transformar $r(x, y)$ em y ou x e, assim, transformar $\theta(r(x, y))$ em $\theta(x)$ ou $\theta(y)$. Baseado nesse fato, podemos enunciar o seguinte resultado (que abrange as funções algébricas):

Teorema 2.2. *Seja ϑ_1 uma 1EDO em $\mathcal{L}_{\mathcal{S}}$ e seja \mathfrak{X} seu 3CVP associado. Considere também, que os campos vetoriais \mathfrak{X} e \mathfrak{I} são definidos como acima. Então, existe uma transformação de variáveis T elementar (com inversa T^{-1} elementar), tal que para a 1EDO $\vartheta_T = T(\vartheta_1)$ vale $F_T | P_T$, $H_T = z F_T$ e $G_T | (Q_T + z N_T)$.*

Prova.

Das hipóteses do teorema ($I \in \mathcal{L}_{\mathcal{S}}$), existe, por definição, uma transformação de variáveis T_0 (com inversa racional) que leva $\theta(r(x, y))$ em $\theta(x)$ ou $\theta(y)$. Uma vez que θ é uma

função elementar, podemos fazer outra transformação (elementar com inversa elementar) que leva $\theta(x)$ ou $\theta(y)$ em $\exp(x)$. Assim, a conclusão segue diretamente da afirmação (a) do teorema 2.1. \square

Observação 2.8. $F|P$ não implica que $F = P$, i.e., os polinômios P e $Q + zN$ podem ter um fator em comum. No entanto, com uma simples mudança de variáveis podemos determinar um possível fator multiplicativo polinomial, uma vez que o fator comum não pode ser comum aos três polinômios Q , P e N . O fator comum é construído a partir de uma soma de polinômios (dada por $Q + zN$) que não se transforma diretamente por uma mudança de variáveis. Mais claramente, se fizermos uma transformação de variáveis T o polinômio Q_{tr} que corresponde ao novo Q **não será** $T(Q)$, pois, uma vez que $I_x = RQ$, temos que $I_{tr} = T(I)$, implicando em:

$$T = \begin{cases} u = u(x, y, z) \\ v = v(x, y, z) \\ w = w(x, y, z), \end{cases} \quad T^{-1} = \begin{cases} x = x(u, v, w) \\ y = y(u, v, w) \\ z = z(u, v, w). \end{cases} \quad (64)$$

Logo,

$$I_u = I_x \frac{\partial u}{\partial x} + I_y \frac{\partial u}{\partial y} + I_z \frac{\partial u}{\partial z} = RQ \frac{\partial u}{\partial x} + RP \frac{\partial u}{\partial y} + RN \frac{\partial u}{\partial z} = R_{tr} Q_{tr}, \quad (65)$$

$$I_v = I_x \frac{\partial v}{\partial x} + I_y \frac{\partial v}{\partial y} + I_z \frac{\partial v}{\partial z} = RQ \frac{\partial v}{\partial x} + RP \frac{\partial v}{\partial y} + RN \frac{\partial v}{\partial z} = R_{tr} P_{tr}, \quad (66)$$

$$I_w = I_x \frac{\partial w}{\partial x} + I_y \frac{\partial w}{\partial y} + I_z \frac{\partial w}{\partial z} = RQ \frac{\partial w}{\partial x} + RP \frac{\partial w}{\partial y} + RN \frac{\partial w}{\partial z} = R_{tr} N_{tr}. \quad (67)$$

Dessa maneira, uma transformação simples deve servir para a maioria dos casos.

Observação 2.9. Em relação à observação anterior (obs. 2.8): “uma transformação simples deve servir para a maioria dos casos” – se uma particular transformação não resolver, então qualquer outra (também muito simples) deve fazer o trabalho. Isso nos permite gerar um algoritmo probabilístico: se uma dada transformação não resultar uma outra bastante simples deve bastar.

Observação 2.10. Ainda em relação à observação 2.8, a única exceção é o caso em que a variável z é um fator de P e Q , pois nesse caso os polinômios P e $Q + zN$ têm o fator z em comum.

Procedimento 3.1 (esboço): ($P_{RO3.1}$)

Passos:

1. Use uma transformação de variável T para obter uma $1EDO_{tr}$ tal que $\theta = \exp(x)$.

2. Construa o campo vetorial \mathfrak{X} associado à 1EDO_{tr} , onde $F = N_0(x, y, z)$, $G = M_0(x, y, z)$, $H = z N_0(x, y, z)$,
3. Escolha $P = N_0$ ou $P = z N_0$ e construa um candidato polinomial POL em (x, y, z) com coeficientes arbitrários e substitua-o na equação (29). Colete a equação polinomial resultante em (x, y, z) obtendo um sistema (quadrático) para os coeficientes de Pol . Resolva o sistema quadrático para os coeficientes indeterminados.
4. A partir da solução, obtenha os 2CVPs associados e calcule (com algum método) a integral primeira I .
5. Aplique a transformação inversa $T^{-1}(I)$ para obter a solução geral do 1EDO original.

Observação 2.11. *O procedimento $P_{RO3.1}$ (descrito acima) é apenas uma maneira possível de implementar um algoritmo para gerar uma melhoria no procedimento 3. Poderíamos, por exemplo, transformar a função θ em $\exp(y)$ em vez de $\exp(x)$. O principal ponto a enfatizar é que, qualquer que seja a escolha do θ transformado, estaremos procurando apenas 1 e não 2 polinômios. Essa diferença se traduz em uma diminuição exponencial no tempo de CPU gasto e na quantidade de memória RAM utilizada no processo (consulte o capítulo 5).*

Considere a 1EDO

$$\frac{dy}{dx} = -\frac{((e^y)^2 x^4 + 2 e^y x^3 - 4 x^2 e^y y - 2 xy + y^2) y}{x ((e^y)^2 x^4 y + e^y x^3 y - 3 x^2 e^y y^2 - e^y x^3 + x^2 e^y y - xy^2 + y^3 + xy)}. \quad (68)$$

Agora, vamos aplicar os procedimentos P_{RO3} e $P_{RO3.1}$:

P_{RO3} :

1. $\mathfrak{X} = F\partial x + G\partial y + H\partial z$, onde

$$F = x (z^2 x^4 y + z x^3 y - 3 x^2 z y^2 - x^3 z + x^2 z y - x y^2 + y^3 + x y), \quad (69)$$

$$G = -y (x^4 z^2 + 2 x^3 z - 4 x^2 z y - 2 x y + y^2), \quad (70)$$

$$H = -y z (x^4 z^2 + 2 x^3 z - 4 x^2 z y - 2 x y + y^2). \quad (71)$$

2. P_{RO2} :

- (a) Para os graus $d_Q = 8$, $d_P = 6$ e $d_N = 8$ teremos:

$$Q = y z (x^4 z^2 + 2 x^3 z - 4 x^2 z y - 2 x y + y^2), \quad (72)$$

$$P = -x^2 z (z x^2 - x z y - y), \quad (73)$$

$$N = x (x^4 z^2 + x^3 z - 3 x^2 z y - x y + y^2) y. \quad (74)$$

(b) Usando o método da função-S, obtemos a seguinte integral primeira:

$$I = xz \exp\left(\frac{y}{zx^2 - y}\right) + Ei\left(1, -\frac{y}{zx^2 - y}\right). \quad (75)$$

3. Substituindo z por $\theta(x, y)$ obtemos, finalmente, a solução geral da 1EDO (68):

$$xe^y \exp\left(\frac{y}{zx^2 - y}\right) + Ei\left(1, -\frac{y}{e^y x^2 - y}\right) = c. \quad (76)$$

$P_{RO3.1}$:

1. A mudança de variáveis dada por $\{x = y, y = x\}$ transformam a 1EDO (68) na 1EDO

$$\frac{dy}{dx} = -\frac{y((e^x)^2 y^4 x + e^x y^3 x - 3y^2 e^x x^2 - e^x y^3 + y^2 e^x x - yx^2 + x^3 + xy)}{x((e^x)^2 y^4 + 2e^x y^3 - 4y^2 e^x x - 2xy + x^2)}. \quad (77)$$

2. $\mathfrak{X}_{tr} = F_{tr}\partial x + G_{tr}\partial y + H_{tr}\partial z$, onde

$$F_{tr} = (z^2 y^4 - 4xy^2 z + 2zy^3 + x^2 - 2xy) x, \quad (78)$$

$$G_{tr} = -y(xy^4 z^2 - 3x^2 z y^2 + y^3 x z + xy^2 z - zy^3 + x^3 - yx^2 + xy), \quad (79)$$

$$H_{tr} = z(z^2 y^4 - 4xy^2 z + 2zy^3 + x^2 - 2xy) x. \quad (80)$$

3. Escolhendo $P_{tr} = z F_{tr}$, teremos:

$$Q_{tr} = y^2 z (yzx - y^2 z + x), \quad (81)$$

$$P_{tr} = z(z^2 y^4 - 4xy^2 z + 2zy^3 + x^2 - 2xy) x, \quad (82)$$

$$N_{tr} = (z^2 y^4 - 3xy^2 z + zy^3 + x^2 - xy) xy. \quad (83)$$

4. Da etapa 3, obtemos a seguinte primeira integral:

$$I_{tr} = yz \exp\left(\frac{x}{y^2 z - x}\right) + Ei\left(1, -\frac{x}{y^2 z - x}\right). \quad (84)$$

5. Usando a transformação inversa $\{x = y, y = x\}$ e substituindo z por $\theta(x, y)$ obtemos, a solução geral da 1EDO original.

Observação 2.12. *Em um teste de implementação usando o Maple¹⁷, a 1EDO (68) foi resolvida por P_{RO3} gastando 15,3 segundos de tempo de CPU (e usando aproximadamente 216 Mb de memória RAM). O algoritmo $P_{RO3.1}$ fez o trabalho em 0,1 segundos (usando aproximadamente 4 Mb de memória RAM)¹⁸.*

Observação 2.13. *Para esta 1EDO (68), caímos no referido caso da observação 2.10, ou seja, $P_{tr} = z F_{tr}$.*

2.2 Calculando (Q, P, N) linearmente

Nesta seção, mostraremos que podemos tirar mais proveito da classe de campos vetoriais polinomiais associados a 1EDOs em \mathcal{L}_{Sr} . Começamos chamando a atenção para o fato de que cada 1EDO desta classe pode ser colocada em uma forma ‘canônica’ onde $\theta = \exp(x)$. Portanto, pelo teorema 2.1, o 3CVP associado será dado por $\mathfrak{X} = P \partial x + G \partial y + z P \partial z$ ou $\mathfrak{X} = (P/z) \partial x + G \partial y + P \partial z$ (ver observação 2.8). Agora, olhando para as equações (29) e escolhendo a configuração adequada ($k = 2, i = 3, j = 1$), obtemos

$$\mathfrak{X} \left(\frac{Q}{P} \right) = H \partial_y \left(\frac{F}{H} \right) \left(\frac{Q}{P} \right)^2 + H \left(\partial_y \left(\frac{G}{H} \right) - \partial_x \left(\frac{F}{H} \right) \right) \left(\frac{Q}{P} \right) - H \partial_x \left(\frac{G}{H} \right). \quad (85)$$

Como $F = P$ (ou $F = P/z$) e $H = zF$ (resp. $H = P$), o termo que multiplica $\left(\frac{Q}{P}\right)^2$ é $\partial_y \left(\frac{F}{H}\right) = \partial_y \left(\frac{1}{z}\right) = 0$. Portanto, a equação (85) se torna:

$$\mathfrak{X} \left(\frac{Q}{P} \right) = z P \partial_y \left(\frac{G}{zP} \right) \left(\frac{Q}{P} \right) - z P \partial_x \left(\frac{G}{zP} \right). \quad (86)$$

Observação 2.14. *A equação (86) é linear em Q e, além disso, todos os outros polinômios presentes (incluindo P) são conhecidos. Portanto, a equação é linear em todos os coeficientes desconhecidos.*

A observação acima nos aponta para um possível procedimento linear para determinar os campos vetoriais 2D associados \mathfrak{X}_i . A equação (86) pode ser escrita como

$$PQ_x + GQ_y + zPQ_z + (-P_z z - G_y - P_x)Q - GP_x + G_x P = 0, \quad (87)$$

¹⁷ Maple é um sistema de álgebra computacional de uso geral (CAS), ou seja, uma plataforma para cálculos simbólicos e, ao mesmo tempo, uma linguagem de programação. Os principais CAS de propósito geral são Axiom, Maxima, Magma, Maple, Mathematica e SageMath.

¹⁸ Apresentamos mais algumas informações deste tipo no capítulo 5.

e usada como base para o método.

Procedimento 3.2 (esboço): ($P_{RO3.2}$)

Etapas: O procedimento 3.2 é idêntico ao procedimento 3.1 em todas as etapas, a menos da etapa 3 que será substituída por:

- Escolha $P = F$ ou $P = zF$ e construa um candidato polinomial P_{OL} em (x, y, z) com coeficientes arbitrários. Substitua-o na equação (87). Colete a equação polinomial resultante em (x, y, z) obtendo um sistema linear para os coeficientes de P_{OL} . Resolva o sistema linear para os coeficientes indeterminados.

3 2EDOS RACIONAIS

Neste capítulo, mostramos como melhoramos a técnica desenvolvida em (3) para encontrar uma integral primeira Liouvillianiana de uma 2EDO racional.

Definição 3.1. *Considere a 2EDO racional dada por*

$$y'' = \phi(x, y, y') = \frac{M_0(x, y, y')}{N_0(x, y, y')}, \quad (88)$$

onde ϕ é uma função racional de (x, y, y') , i.e., M_0, N_0 são polinômios coprimos em $\mathbb{C}[x, y, y']$; além disso, $y' \equiv \frac{dy}{dx}$ e $y'' \equiv \frac{dy'}{dx}$. Suponha ainda que uma integral primeira da 2EDO (88) pode ser escrita como $I(x, y, y')$, onde I é uma função Liouvillianiana de (x, y, y') tal que suas derivadas parciais (relacionadas às variáveis x, y e y') possam ser expressas como

$$I_x = RQ, \quad (89)$$

$$I_y = RP, \quad (90)$$

$$I_{y'} = RN, \quad (91)$$

onde R é uma função Liouvillianiana de (x, y, y') e $Q, P, N \in \mathbb{C}[x, y, y']$. Então dizemos que a 2EDO (88) é um membro do conjunto \mathcal{L}_{FI} .

Sobre as soluções de (88), a derivada total $\frac{d}{dx}$ é equivalente ao operador $D_x \equiv \partial_x + z \partial_y + \phi \partial_z$. Portanto, para obter um 3CVP, basta multiplicar D_x por N_0 .

Definição 3.2. *Considere uma 2EDO racional escrita como em (88) e seja \mathfrak{X} o 3CVP definido por $\mathfrak{X} \equiv N_0(x, y, z) \partial_x + z N_0(x, y, z) \partial_y + M_0(x, y, z) \partial_z$. Dizemos que \mathfrak{X} é o 3CVP associado com a 2EDO (88).*

Observação 3.1. *Claramente, se $I(x, y, y')$ é uma integral primeira da 2EDO (88) então $I(x, y, z)$ é uma integral primeira do vetor campo \mathfrak{X} (uma vez que $\mathfrak{X} = N_0 D_x$).*

O campo vetorial \mathfrak{X} associado a um 2EDO racional que tem uma integral primeira Liouvillianiana carrega (como no caso de 1EDOs com uma função elementar) mais informações do que o caso geral: temos (antecipadamente) conhecimento da componente N do campo vetorial \mathfrak{J} . Desta forma, as equações (29) para os pares (P, N) e (Q, N) se tornam condições para apenas um polinômio desconhecido. Veremos que, como consequência, o procedimento para determinar os campos vetoriais 2D associados (\mathfrak{X}_i) se torna um algoritmo completo.

Teorema 3.1. *Seja ϑ_2 uma 2EDO (88) em \mathcal{L}_{FI} e seja \mathfrak{X} seu 3CVP associado. Além disso, suponha que o campo vetorial \mathfrak{J} seja definido como acima. Se N e $Q + zP$ são*

polinômios coprimos, então o cálculo dos campos vetoriais 2D associados (\mathfrak{X}_i) se torna um algoritmo completo.

Prova. A partir das hipóteses do teorema ($\vartheta_2 \in \mathcal{L}_{FI}$, N e $Q + zP$ são coprimos) ϑ_2 apresenta uma integral primeira Liouvilliana I e, além disso, a 2EDO ϑ_2 pode ser escrita como

$$y'' = -\frac{I_x + zI_y}{I_z} = -\frac{Q + zP}{N},$$

implicando que $M_0 = -(Q + zP)$ e $N_0 = N$. Escolhendo $i = 1$, $j = 2$, $k = 3$ em (29) temos

$$\mathfrak{X} \left(\frac{P}{N_0} \right) = N_0 \partial_z \left(\frac{zN_0}{N_0} \right) \left(\frac{P}{N_0} \right)^2 + N_0 \left(\partial_z \left(\frac{M_0}{N_0} \right) - \partial_y \left(\frac{zN_0}{N_0} \right) \right) \left(\frac{P}{N_0} \right) - N_0 \partial_y \left(\frac{M_0}{N_0} \right),$$

levando a $\mathfrak{X}(P) = P^2 + (N_x + zN_y + M_z)P + MN_y - M_yN$, ($N \equiv N_0$, $M \equiv M_0$). Isolando P^2 no lado esquerdo, temos

$$P^2 = NP_x + zNP_y + MP_z - (N_x + zN_y + M_z)P - MN_y + M_yN. \quad (92)$$

O grau do termo P^2 no lado esquerdo da equação (92) é $2 \deg_P$ (uma vez que o termo é um quadrado). Os graus (máximos) dos seguintes termos (no lado direito) são, respectivamente, $\deg_P + \deg_N - 1$, $\deg_P + \deg_N$, $\deg_P + \deg_M - 1$, $\deg_P + \deg_N - 1$, $\deg_P + \deg_N$, $\deg_P + \deg_M - 1$, $\deg_M + \deg_N - 1$, $\deg_M + \deg_N - 1$.

Existem, portanto, quatro casos distintos:

caso 1: $\deg_P \leq \deg_N - 1$;

caso 2: $\deg_P \leq \deg_N$;

caso 3: $\deg_P \leq \deg_M - 1$;

caso 4: $\deg_P \leq \frac{\deg_M + \deg_N - 1}{2}$.

O caso 1 implica o caso 2 e os casos 2 e 3 (combinados) implicam o caso 4. Portanto, se $\deg_M > \deg_N + 1$ então $\deg_P \leq \deg_M - 1$, caso contrário, se $\deg_M \leq \deg_N + 1$ então $\deg_P \leq \deg_N$. Em ambos os casos, o grau de P (\deg_P) tem um limite superior e, portanto, o procedimento termina. \square

Corolário 3.1. *Seja ϑ_2 uma 2EDO (88) em \mathcal{L}_{FI} e seja \mathfrak{X} seu 3CVP associado. Além disso, assumamos que o campo vetorial \mathfrak{J} é definido como acima e que N e $Q + zP$ são polinômios coprimos. Se R é um fator integrante para \mathfrak{X} então $\frac{\mathfrak{X}(R)}{R} = -(\langle \nabla, \mathfrak{X} \rangle + P)$.*

Prova. Das hipóteses ($\vartheta_2 \in \mathcal{L}_{FI}$, N e $Q + zP$ são coprimos) temos que $M_0 = -(Q + zP)$ e $N_0 = N$. Do lema 1.2, temos que $\mathfrak{X} \wedge \mathfrak{L} = \frac{\mathfrak{X}(R)}{R} \mathfrak{J}$. Assim, a terceira componente desta equação vetorial implica que $F(-N_x + Q_z) - G(N_y - P_z) = \frac{\mathfrak{X}(R)}{R} N \Rightarrow \frac{\mathfrak{X}(R)}{R} = -N_x - zN_y + Q_z + zP_z = -N_x - zN_y - M_{0z} - P$. \square

Procedimento 4 (esboço): (P_{RO4})

Etapas:

1. Construa o campo vetorial \mathfrak{X} associado à 2EDO (88), onde $F = N_0(x, y, z)$, $G = z N_0(x, y, z)$, $H = M_0(x, y, z)$,
2. Escolha $N = N_0$ e construa um candidato polinomial P_c de grau $d_P = \max\{d_{M_0} - 1, d_{N_0}\}$ em (x, y, z) com coeficientes indeterminados.
3. Substitua $P = P_c$ na equação $P^2 = N P_x + z N P_y + M P_z - (N_x + z N_y + M_z) P - M N_y + M_y N$. Colete a equação polinomial resultante em (x, y, z) obtendo um sistema (quadrático) para os coeficientes de P_c .
4. Resolva o sistema quadrático para os coeficientes indeterminados. Substitua a solução em P_c obtendo P .
5. Substitua P na equação $M_0 = -(Q + z P)$ para obter Q .
6. De posse dos 2CVPs associados, calcule (com algum método) a integral primeira I .

Observação 3.2. *Se N e $Q + z P$ têm um fator polinomial comum, podemos encontrá-lo simplesmente fazendo uma transformação simples ($\{x = y, y = x\}$, por exemplo, por uma razão análoga à razão mencionada na observação 2.8).*

Não podemos aplicar o procedimento que acabamos de descrever para a 2EDO¹⁹

$$y'' = -\frac{x^2 y^3 - 3x^2 y z + y^2 - z}{x^2} \quad (93)$$

porque $N_0 = x^2$ mas $N = -(2xy + y + 2)x^2$. Isto acontece porque $Q + zP = -(2xy + y + 2)(x^2 y^3 - 3x^2 y z + y^2 - z)$, ou seja, $Q + zP$ e N têm o fator $(2xy + y + 2)$ em comum.

Aplicando a transformação $T = \{x = y, y = x\}$ para 2EDO (93) nós achamos

$$N_{0tr} = -(2xy + x + 2)y^2,$$

$$M_{0tr} = -(2x^4 y^3 z + x^4 y^2 z - 4x^2 y^3 z^2 + 2x^3 y^2 z - 2x^2 y^2 z^2 + 2y^3 z^3 + 2x^3 y z - 2x^2 y^3 - 2xy^2 z^2 + y^2 z^3 + x^3 z - x^2 y^2 - 2xy z^2 - 2y^3 z + 2x^2 z - 4xy^2 - xz^2 - y^2 z - 2z^2).$$

Agora, podemos aplicar o procedimento 4 a essa 2EDO transformada:

P_{RO4} :

1. $\mathfrak{X} = N_{0tr} \partial x + z N_{0tr} \partial y + M_{0tr} \partial z$, (N_{0tr} e M_{0tr} definidos logo acima).

¹⁹ 2EDO não linear do livro do Kamke (40) com $f(x) = 1/x^2$.

2. , 3. , 4. , 5. Para o grau $d_P = 7$ encontramos $P = -(x^2 - z)(2x^2y^3 + x^2y^2 - 2y^3z + 2xy^2 - y^2z + 2xy + x + 2)$, $Q = (2x^2y + x^2 + 2yz + 4x + z)y^2$.

$$3. \quad I = \frac{e^{\frac{1}{y}}(x^2 - z)}{x^2y^2 - y^2z + 2xy + x + 2}.$$

Aplicando a transformação inversa, obtemos: $\frac{e^{\frac{1}{x}}(y^2 - z)}{x^2y^2 - x^2z + 2xy + y + 2}$, que é a integral primeira da 2ODE original (93).

Podemos usar mais algumas consequências das hipóteses do teorema 3.1 para aprimorar ainda mais o método. Vamos usar o seguinte corolário:

Corolário 3.2. *Considere que as hipóteses do teorema 3.1 valem. Se nenhum dos termos monomiais de zP e Q se cancelam, então todos os monômios de P estão em M_{0z} .*

Prova. A conclusão decorre diretamente do fato de que $M_0 = -(Q + zP)$. \square

Portanto, se as condições do corolário 3.2 se aplicarem, podemos usar apenas os monômios de M_{0z} para construir o candidato P_c :

Procedimento 4 (esboço): (P_{RO4})

Etapas:

1. Construa o campo vetorial \mathfrak{X} associado à 2EDO (88), onde $F = N_0(x, y, z)$, $G = zN_0(x, y, z)$, $H = M_0(x, y, z)$,
2. Escolha $N = N_0$ e construa um candidato polinomial P_c usando os monômios de M_{0z} com coeficientes indeterminados. Substitua-os na equação (29) por $k = 1$, $j = 2$, $i = 3$. Colete a equação polinomial resultante em (x, y, z) obtendo um sistema (quadrático) para os coeficientes de P_c .
3. Resolva o sistema quadrático para os coeficientes indeterminados. A partir da solução, obtenha os 2CVPs associados.
4. Calcule (com algum método) a integral primeira I .

Observação 3.3. *No caso em que alguns dos termos monomiais de zP e Q se cancelam, podemos seguir uma estratégia semelhante àquela que foi descrita na observação 2.8 para determinar esses monômios: podemos fazer uma transformação de variáveis simples que muda os polinômios P e Q de formas diferentes, dessa maneira, desfazer os cancelamentos.*

Considere a 2EDO

$$y'' = -\frac{2x^2z^4 + 4xy^2z^3 + 2y^4z^2 - xyz^3 + y^3z^2 + y^2z^2 - 3xyz - y^3 - xz}{x^2yz^2 + 2xy^3z + y^5 + xy^2z - x^2}. \quad (94)$$

Para esta 2EDO, temos que $Q = y^3z^2 - x^2z^2 - 2xy^2z - y^4 - xyz$ e $zP = 2x^2yz^4 + 4xy^3z^3 + 2y^5z^2 - xy^2z^3 + y^4z^2 + x^2z^2 - xy^2z$. Uma segunda olhada nos mostra que o monômio x^2z^2 aparece com o sinal alterado Q e zP e, portanto, cancela. Aplicando a transformação $\{x = x^2\}$, os polinômios zP e Q tornam-se, respectivamente, $-2x^5z^2 - 4x^3y^2z + 2xy^3z^2 - 2x^3yz - 2xy^4$ e $2x^4yz^4 + 4x^2y^3z^3 - x^2y^2z^3 + 2y^5z^2 + x^4z^2 + y^4z^2 - x^2y^2z$, e não há cancelamento.

No próximo capítulo iremos apresentar uma implementação dos métodos construídos neste capítulo e nos capítulos 1 e 2.

4 O PACOTE *LEAPS3DIM*

Neste capítulo apresentamos uma implementação dos procedimentos discutidos nos capítulos 1, 2 e 3, a saber, para buscar integrais primeiras Liouvillianas de sistemas diferenciais polinomiais 3D (3CVP) em geral, 3CVP representando 1EDOs com funções elementares e 3CVP associados a 2EDOs racionais, respectivamente. Esses procedimentos são aprimoramentos e extensões de métodos desenvolvidos pelo nosso grupo:

- O procedimento geral que implementa os métodos desenvolvidos no capítulo 1 é uma extensão do trabalho que realizei em minha dissertação de mestrado (1) e esta implementação é uma extensão do pacote *Leaps3D*. As principais mudanças consistem em:
 - Uma extensão ao procedimento mais clássico (no estilo Darbouxiano) que praticamente ‘jogamos fora’ na minha dissertação por ser muito oneroso (computacionalmente falando). Neste trabalho, descobrimos que podemos usar de modo mais eficiente a estrutura do fator integrante:

$$R = \exp\left(\frac{A}{B}\right) \prod_i p_i^{n_i}.$$

Como vimos, o fator exponencial $\exp\left(\frac{A}{B}\right)$ é uma autofunção do 3CVP \mathfrak{X} , levando a que $\mathfrak{X}\left(\frac{A}{B}\right)$ é um polinômio. Pudemos usar este conhecimento para ‘dividir’ o cálculo de P e N nas equações (veja as equações (21) e (eq1pdeN2))

$$P\left(\mathfrak{X}(A) - A\frac{\mathfrak{X}(B)}{B} + B\sum_i n_i q_i\right) + B\left(F\partial_y\left(\frac{G}{F}\right)P + F\partial_y\left(\frac{H}{F}\right)N + \mathfrak{X}(P)\right) = 0,$$

$$N\left(\mathfrak{X}(A) - A\frac{\mathfrak{X}(B)}{B} + B\sum_i n_i q_i\right) + B\left(F\partial_z\left(\frac{G}{F}\right)P + F\partial_z\left(\frac{H}{F}\right)N + \mathfrak{X}(N)\right) = 0.$$

O termo $\mathfrak{X}(A) - A\frac{\mathfrak{X}(B)}{B} + B\sum_i n_i q_i$ que aparece é, a princípio, desconhecido. Contudo, usando o fato de que $\mathfrak{X}\left(\frac{A}{B}\right)$ é um polinômio, podemos determinar o polinômio A (de maneira linear – veja a seção 1.4.1 sobre o algoritmo P_{RO1}) antes de substituí-lo nas equações acima. Isso divide o algoritmo em duas partes, uma parte quadrática apenas nos n_i e uma linear, ao invés de uma quadrática nos coeficientes de A e nos n_i .

- Uma melhoria no algoritmo que usava o método da função-S (materializada no algoritmo P_{RO2}) que nos permite calcular os campos vetoriais associados \mathfrak{X}_i de uma maneira mais abrangente (usando o fato de termos três 2CVPs associados).

- O procedimento 3 que implementa os métodos desenvolvidos no capítulo 2 é uma extensão do trabalho realizado em (4) e esta implementação (algoritmo P_{RO3}) é um aperfeiçoamento do pacote *LeapS1ode* e uma implementação do capítulo 5 de (4). As principais mudanças consistem em:
 - Em relação ao primeiro aperfeiçoamento (algoritmo $P_{RO3.1}$) a novidade é a inclusão de funções algébricas.
 - Em relação à implementação do algoritmo $P_{RO3.2}$, a implementação é original e inclui a mesma vantagem do algoritmo $P_{RO3.1}$, ou seja, a inclusão das funções algébricas.
- O procedimento 4 que implementa os métodos desenvolvidos no capítulo 3 é uma extensão do trabalho realizado em (3) e esta implementação (algoritmo P_{RO4}) se constitui em dois aperfeiçoamentos do pacote *InSyDE*:
 - O primeiro aperfeiçoamento é baseado no teorema 3.1 e no corolário 3.1. Na verdade, a grande ‘revolução’ consiste no fato expresso pela observação 3.2: que **é possível suspender a obstrução** causada pelo cancelamento de um fator polinomial comum entre os termos N e $zP + Q$ **com uma transformação de variáveis simples**.
 - O segundo aperfeiçoamento é baseado no corolário 3.2 e, da mesma forma que no primeiro aperfeiçoamento, também aqui a melhoria reside no fato expresso pela observação 3.3: que **é possível suspender a obstrução** causada pelo cancelamento de termos monomiais de mesma assinatura e coeficientes com sinais contrários nos polinômios zP e Q **com uma transformação de variáveis simples**.

Esses algoritmos foram implementados em um pacote computacional escrito na linguagem Maple (*LeapS3Dim*), com algumas modificações para melhorar sua flexibilidade e praticidade. O pacote será apresentado da seguinte forma:

1. Na primeira seção, mostramos os comandos que compõe o pacote *LeapS3Dim*: fazemos uma descrição detalhada da funcionalidade e das chamadas de cada um dos comandos que compõe o pacote. Como o pacote em si, consiste em uma conjunção da extensão de pacotes semelhantes com especificidades bem distintas, resolvemos criar um chaveamento que permite ao pesquisador escolher o caminho a tomar, onde cada um deles contempla basicamente um dos capítulos (1, 2 ou 3).
2. Na segunda seção, apresentamos exemplos práticos do uso dos comandos do pacote aplicados a exemplos específicos.

4.1 Os comandos do pacote *LeapS3Dim*

Nesta seção apresentaremos uma descrição (detalhada) dos principais comandos do sub pacote *LeapS3Dim*.

Resumo dos comandos:

- **Dx** constrói o operador D_x (o operador derivada total sobre as soluções) associado com a 1EDO (32).
- **XX** - constrói o operador \mathfrak{X} (3CVP) associado com o sistema diferencial 3D (9), com a 1EDO (32) ou com a 2EDO (88).
- **Tr3Dsys** - aplica uma transformação (dada pelo usuário) no sistema (9) e retorna o sistema transformado e a transformação inversa.
- **Tr1ode** - aplica uma transformação (dada pelo usuário) na 1EDO (32) e retorna a 1EDO transformada e a transformação inversa.
- **Tr2ode** - aplica uma transformação (dada pelo usuário) na 2EDO (88) e retorna a 2EDO transformada e a transformação inversa.
- **Xi2D** - tenta determinar os 2CVP \mathfrak{X}_i (que determinam as funções S) associados com o campo vetorial \mathfrak{X} .
- **Darboux** - tenta determinar os polinômios de Darboux do 3CVP \mathfrak{X} .
- **FindA** - tenta determinar o polinômio A (numerador do fator exponencial $\exp(A/B)$).
- **InFa** - tenta determinar um fator integrante (usando o algoritmo P_{RO1}).
- **InFa2** - tenta determinar um fator integrante (usando o algoritmo P_{RO2}) usando as soluções gerais para as 1EDOs racionais associadas.
- **Invar** - tenta determinar a integral primeira do campo vetorial \mathfrak{X} .
- **Solde** - tenta determinar a integral primeira do campo vetorial \mathfrak{X} ou, equivalentemente, a solução geral da 1EDO (32).

4.1.1 Comando: Dx

Funcionalidade: Este comando retorna o operador D_x , que é o operador derivada total (sobre as soluções da 1EDO).

Sequência de chamada:

```
[> Dx(ode);
```

Parâmetros:

ode - a 1EDO (32).

Sinopse:

O comando Dx retorna o *operador derivada total* sobre as soluções da 1EDO (32), i.e., o operador $d_x = \partial_x + y' \partial_y$ restrito ao limite $I(x, y) = c$, que são as curvas de solução. Uma vez que sobre as soluções, $y' = \phi(x, y)$, isto implica que

$$D_x = (\partial_x + y' \partial_y)|_{y'=\phi} = \partial_x + \phi \partial_y, \quad \left(y' = \frac{dy}{dx} \right). \quad (95)$$

4.1.2 Comando: XX

Funcionalidade: Este comando retorna o campo vetorial \mathfrak{X} (3CVP) associado com o sistema diferencial 3D (9).

Linha de comando:

```
[> XX (s3d);
```

Parâmetros:

s3d - o sistema diferencial 3D (9) – uma lista com os três polinômios [F,G,H].

Parâmetros extras:

ALG = [1,2,3] - onde $ALG \in \{1, 2, 3\}$ diz ao comando se é um sistema (ALG=1), uma 1EDO com funções (ALG=2) ou uma 2EDO racional (ALG=3). O *default* é ALG=1.

Sinopse:

O comando XX retorna o *campo vetorial* \mathfrak{X} associado com o sistema (9), i.e., o comando retorna um operador $u \rightarrow F \partial_x(u) + G \partial_y(u) + H \partial_z(u)$. Se o parâmetro ALG estiver presente

então $ALG=2$ faz com que o programa escreva o 3CVP equivalente a uma 1EDO com uma função elementar e $ALG=3$ devolve o 3CVP equivalente a uma 2EDO racional

4.1.3 Comando: Tr3Dsys

Funcionalidade: Este comando aplica uma transformação ao sistema (9) e retorna o sistema transformado e a transformação inversa.

Linha de comando:

```
[> Tr3Dsys(s3d);
```

Parâmetros:

s3d - o sistema diferencial 3D (9) – uma lista com os três polinômios [F,G,H].

Parâmetros extras:

TR = [x=f(x,y,z), y=g(x,y,z), z=h(x,y,z)], onde [x=f(x,y,z), y=g(x,y,z), z=h(x,y,z)] é uma transformação inversível - O *default* é a transformação identidade.

Sinopse:

O comando Tr3Dsys aplica uma transformação T (fornecida pelo usuário) ao sistema (9) e retorna uma lista com duas entradas: o sistema transformado e a transformação inversa T^{-1} da transformação fornecida (pelo parâmetro TR). Se a transformação inversa não puder ser obtida, o comando retorna somente o sistema transformado. Se nenhuma transformação é fornecida, o *default* é a transformação identidade.

4.1.4 Comando: Tr1ode

Funcionalidade: Este comando aplica uma transformação à 1EDO (32) e retorna a 1EDO transformada e a transformação inversa.

Linha de comando:

```
[> Tr1ode(ode);
```

Parâmetros:

ode - a 1EDO (32).

Parâmetros extras:

$\text{TR} = [\mathbf{x}=\mathbf{F}(\mathbf{x}, \mathbf{y}), \mathbf{y}=\mathbf{G}(\mathbf{x}, \mathbf{y})]$, onde $[\mathbf{x}=\mathbf{F}(\mathbf{x}, \mathbf{y}), \mathbf{y}=\mathbf{G}(\mathbf{x}, \mathbf{y})]$ é uma transformação inversível - O *default* é a transformação identidade.

Sinopse:

O comando `Tr1ode` aplica uma transformação T (fornecida pelo usuário) à 1EDO (32) e retorna uma lista com duas entradas: a 1EDO transformada e a transformação inversa T^{-1} da transformação fornecida (pelo parâmetro `TR`). Se a transformação inversa não puder ser obtida, o comando retorna somente a 1EDO transformada. Se nenhuma transformação é fornecida, o *default* é a transformação identidade.

4.1.5 Comando: Tr2ode

Funcionalidade: Este comando aplica uma transformação à 2EDO (88) e retorna a 2EDO transformada e a transformação inversa.

Linha de comando:

```
[> Tr2ode(ode);
```

Parâmetros:

`ode` - a 2EDO (88).

Parâmetros extras:

$\text{TR} = [\mathbf{x}=\mathbf{F}(\mathbf{x}, \mathbf{y}), \mathbf{y}=\mathbf{G}(\mathbf{x}, \mathbf{y})]$, onde $[\mathbf{x}=\mathbf{F}(\mathbf{x}, \mathbf{y}), \mathbf{y}=\mathbf{G}(\mathbf{x}, \mathbf{y})]$ é uma transformação inversível - O *default* é a transformação identidade.

Sinopse:

O comando `Tr2ode` aplica uma transformação T (fornecida pelo usuário) à 2EDO (88) e retorna uma lista com duas entradas: a 2EDO transformada e a transformação inversa T^{-1} da transformação fornecida (pelo parâmetro `TR`). Se a transformação inversa não puder ser obtida, o comando retorna somente a 2EDO transformada. Se nenhuma transformação é fornecida, o *default* é a transformação identidade.

4.1.6 Comando: Xi2D

Funcionalidade: Este comando retorna os 2CVP \mathfrak{X}_i (que determinam as funções S) associados com o campo vetorial \mathfrak{X} .

Linha de comando:

```
[> Xi2D(s3d);
```

Parâmetros:

s3d - o sistema diferencial 3D (9) – uma lista com os três polinômios $[F,G,H]$.

Parâmetros extras:

DegQPN = $[degQ, degP, degN]$ - onde $[degQ, degP, degN]$ é uma lista de graus dos candidatos a Q , P e N , respectivamente. O *default* é $[3,3,3]$.

QPN = **QP** ou **PN** ou **QN** - diz ao comando **Xi2D** que par de polinômios utilizar, i.e., qual das equações (29) utilizar.

Sinopse:

O comando **Xi2D** tenta encontrar os 2CVPs \mathfrak{X}_i associados (através de uma integral primeira Liouvilliana I) ao sistema diferencial 3D (9). O comando computa, se possível, dois polinômios do conjunto $\{Q, P, N\}$ (lembrando que com qualquer par podemos achar o terceiro através da relação $QF + PG + NH = 0$). Os campos vetoriais associados formam a base para um dos caminhos para encontrar a integral primeira Liouvilliana I do campo vetorial \mathfrak{X} , associado ao sistema (9). Temos o parâmetro **DegQPN**, uma lista com três inteiros $[degQ, degP, degN]$ que diz ao comando o grau dos candidatos a Q , P e N , respectivamente.

4.1.7 Comando: Darboux

Funcionalidade: Este comando determina os polinômios de Darboux do 3CVP \mathfrak{X} .

Linha de comando:

```
[> Darboux(s3d);
```

Parâmetros:

s3d - o sistema diferencial 3D (9) – uma lista com os três polinômios $[F,G,H]$.

Parâmetros extras:

$\text{Degp} = \text{dp}$ - onde dp é um número inteiro positivo que informa o grau do candidato a polinômio de Darboux p_c . O *default* é 3.

$\text{Degq} = \text{dq}$ - onde dq é um número inteiro positivo que informa o grau do candidato a cofator q_c . O *default* é o grau máximo possível para o sistema (9), ou seja, $\text{dq} = \max\{\text{deg}_F, \text{deg}_G, \text{deg}_H\} - 1$.

Sinopse:

O comando `Darboux` tenta determinar os polinômios de Darboux do campo vetorial \mathfrak{X} até um certo grau determinado. O método empregado consiste em construirmos dois polinômios associados (através de uma integral primeira Liouvilliana I) ao sistema diferencial 3D (9). O comando constrói, dois polinômios candidatos p_c e q_c com coeficientes indeterminados e tentamos resolver o sistema resultante da equação $\mathfrak{X}(p_c) = q_c p_c$. O parâmetro `Degp` nos diz o grau máximo para construirmos o candidato a polinômio de Darboux p_c e o parâmetro `Degq` nos diz o grau máximo para construirmos o candidato a cofator q_c . O comando retorna, se tiver sucesso, uma lista de listas com os polinômios de Darboux e seus respectivos cofatores: `[[p1,q1], [p2,q2], ...]`.

4.1.8 Comando: FindA

Funcionalidade: Este comando determina o polinômio A que é o numerador do argumento do fator exponencial que compõe o fator integrante (para uma possível configuração do polinômio B que é o denominador do argumento do fator exponencial).

Linha de comando:

```
[> FindA(s3d,Bc);
```

Parâmetros:

`s3d` - o sistema diferencial 3D (9) – uma lista com os três polinômios $[F,G,H]$.

`Bc` - uma configuração possível do polinômio B .

Parâmetros extras:

$\text{Dega} = \text{da}$ - onde da é um número inteiro positivo que informa o grau do candidato a polinômio A . O *default* é 3.

$\text{Degb} = \text{db}$ - onde db é um número inteiro positivo que informa o grau do candidato B_c . O *default* é 3.

Sinopse:

O comando **FindA** tenta pré-determinar o polinômio A usando o fato que $\mathfrak{X}(A/B)$ é um polinômio. O parâmetro **Dega** nos diz o grau máximo para construirmos o candidato A_c . Este cômputo do polinômio A só vale para a configuração B_c do polinômio B .

4.1.9 Comando: InFa

Funcionalidade: Este comando determina um fator integrante para a 1-forma $Q dx + P dy + N dz$, baseado nas equações (21) e (22).

Linha de comando:

```
[> InFa(s3d);
```

Parâmetros:

s3d - o sistema diferencial 3D (9) – uma lista com os três polinômios [F,G,H].

Parâmetros extras:

$\text{Degp} = \text{dp}$ - onde dp é um número inteiro positivo que informa o grau do candidato a polinômio de Darboux p_c . O *default* é 3.

$\text{Degq} = \text{dq}$ - onde dq é um número inteiro positivo que informa o grau do candidato a cofator q_c . O *default* é o grau máximo possível para o sistema (9), ou seja, $\text{dq} = \max\{\text{deg}_F, \text{deg}_G, \text{deg}_H\} - 1$.

$\text{Dega} = \text{da}$ - onde da é um número inteiro positivo que informa o grau do candidato a polinômio A . O *default* é 3.

$\text{Degb} = \text{db}$ - onde db é um número inteiro positivo que informa o grau do candidato B_c . O *default* é 3.

$\text{PD} = \text{pd}$ - onde pd é uma lista de listas com os polinômios de Darboux e seus respectivos cofatores.

$\text{AB} = \text{ab}$ - onde ab é uma lista com os polinômios A e B .

Sinopse:

O comando **InFa** tenta determinar um fator integrante para o campo vetorial \mathfrak{X} (ou, equivalentemente, para a 1-forma $Q dx + P dy + N dz$) até um certo grau determinado para os polinômios de Darboux. O método empregado consiste, basicamente, em uma implementação do algoritmo P_{RO1} : calculamos os polinômios de Darboux até um certo grau e, de posse deles, podemos construir diversos candidatos para o polinômio B . Para cada candidato B_c possível de um determinado grau, tentamos determinar o A linearmente. A partir daí tentamos encontrar n_i que satisfaçam às equações (21) e (22). Uma solução encontrada nos leva a um fator integrante R .

4.1.10 Comando: InFa2

Funcionalidade: Este comando determina um fator integrante para a 1-forma $Q dx + P dy + N dz$, baseado no algoritmo P_{RO2} .

Linha de comando:

```
[> InFa2(s3d);
```

Parâmetros:

s3d - o sistema diferencial 3D (9) – uma lista com os três polinômios [F,G,H].

Parâmetros extras:

DegQPN = [degQ, degP, degN] - onde [degQ, degP, degN] é uma lista de graus dos candidatos a Q , P e N , respectivamente. O *default* é [3,3,3].

QPN = qpn - onde qpn é QP ou QN ou PN. O *default* é PN.

X2D = x2d - onde x2d é 1, 2 ou 3, correspondentes aos campos \mathfrak{X}_1 , \mathfrak{X}_2 e \mathfrak{X}_3 . O *default* é 1.

Sinopse:

O comando **InFa2** tenta determinar um fator integrante para o campo vetorial \mathfrak{X} (ou, equivalentemente, para a 1-forma $Q dx + P dy + N dz$) até um certo grau determinado para os polinômios Q , P e N . O método empregado consiste, basicamente, em uma implementação do algoritmo P_{RO2} : calculamos os polinômios Q , P e N e, de posse deles, podemos construir o fator integrante comum aos 2CVPs \mathfrak{X}_i e, portanto, ao 3CVP \mathfrak{X} .

4.1.11 Comando: Invar

Funcionalidade: Este comando tenta encontrar integral primeira Liouvilliana para o sistema (9), ou de forma equivalente, a integral primeira para o campo vetorial \mathfrak{X} .

Linha de comando:

```
[> Invar(s3d);
```

Parâmetros:

s3d - o sistema diferencial 3D (9) – uma lista com os três polinômios [F,G,H].

Parâmetros extras:

DegQPN = [degQ, degP, degN] - onde [degQ, degP, degN] é uma lista de graus dos candidatos a Q , P e N , respectivamente. O *default* é [3,3,3].

QPN = qpn - onde qpn é QP ou QN ou PN. O *default* é PN.

Degp = dp - onde dp é um número inteiro positivo que informa o grau do candidato a polinômio de Darboux p_c . O *default* é 3.

Degq = dq - onde dq é um número inteiro positivo que informa o grau do candidato a cofator q_c . O *default* é o grau máximo possível para o sistema (9), ou seja, $dq = \max\{deg_F, deg_G, deg_H\} - 1$.

Dega = da - onde da é um número inteiro positivo que informa o grau do candidato a polinômio A . O *default* é 3.

Degb = db - onde db é um número inteiro positivo que informa o grau do candidato B_c . O *default* é 3.

PD = pd - onde pd é uma lista de listas com os polinômios de Darboux e seus respectivos cofatores.

AB = ab - onde ab é uma lista com os polinômios A e B .

PRO = nk - onde nk é um dos números do conjunto {1, 2, 3, 4, 31, 32, 42}. Os números se referem aos procedimentos desenvolvidos nos capítulos 1, 2 e 3.

Sinopse:

O comando `Invar` tenta determinar a integral primeira Liouvilliana para o campo vetorial \mathfrak{X} ou, equivalentemente, para o sistema (9). Este comando é responsável pelo processo como um todo, isto é, ele devolve a parte final do processo (a integral primeira I). Por isso temos um grande número de parâmetros extras, onde o responsável pelo caminho seguido é o parâmetro `nk` que é um dos seguintes números: $\{1, 2, 3, 4, 31, 32, 42\}$:

- `nk =1` – Corresponde ao algoritmo P_{RO1} , isto é, à implementação do procedimento Darbouxiano ‘clássico’: calculamos os polinômios de Darboux e tentamos determinar um fator integrante usando as equações (21) e (eq1pdeN2) – veja a seção 1.4.1.
- `nk =2` – Corresponde ao algoritmo P_{RO2} , ou seja, à implementação do procedimento que determina os campos vetoriais polinomiais 2D associados \mathfrak{X}_i (usando as equações (29)) e os usa para obter (pelo método da função S ou qualquer outro) a integral primeira I do campo vetorial original \mathfrak{X} – veja a seção 1.4.2.
- `nk =3` – Corresponde ao algoritmo P_{RO3} que usa a correspondência entre 1EDOs com uma função elementar e um 3CVP \mathfrak{X} – veja o início do capítulo 2.
- `nk =31` – Corresponde ao algoritmo $P_{RO3,1}$ que usa o fato que para uma 1EDO em \mathcal{L}_S podemos conhecer de antemão um dos polinômios $\{Q, P, N\}$ – veja a seção 2.1.
- `nk =32` – Corresponde ao algoritmo $P_{RO3,2}$ que usa o fato que podemos, para uma 1EDO em \mathcal{L}_S , construir um algoritmo linear ($P_{RO3,2}$) para determinar os polinômios $\{Q, P, N\}$ – veja a seção 2.2.
- `nk =4` – Corresponde ao algoritmo P_{RO4} que usa a correspondência entre uma 2EDO racional e um 3CVP \mathfrak{X} – veja o início do capítulo 3.
- `nk =42` – Corresponde ao algoritmo P_{RO4} com o uso do fato que, se os polinômios zP e Q não possuem termos monomiais de mesma assinatura e coeficientes com sinais contrários, então todos os monômios de P estão em M_{0z} – veja o corolário 3.2 na parte final do capítulo 3.

4.1.12 Comando: Solde

Funcionalidade: Este comando tenta encontrar uma solução geral Liouvilliana para a 1EDO (32), ou de forma equivalente, a integral primeira do campo vetorial \mathfrak{X} . Ela retorna $I(x, y, \theta) = c$, onde θ é uma função elementar de (x, y) .

Linha de comando:

[> Solde(ode);

Parâmetros:

ode - a 1EDO (32).

Parâmetros extras: O mesmos acima e:

II - Se este parâmetro está presente na chamada de comando, então o comando retorna uma lista com a integral primeira Liouvilliana do campo vetorial \mathfrak{X} e a transformação $z \rightarrow \theta : [I(x, y, z), z = \theta(x, y)]$.

Sinopse:

O comando `Solde` é a parte final do processo (no caso de campos vetoriais representando 1EDOs com uma função elementar), i.e., ele retorna, se possível, a solução geral da 1EDO (32). Com o parâmetro II o usuário pode fazer o comando retornar a integral primeira do campo vetorial \mathfrak{X} , junto com a transformação $z \rightarrow \theta$.

Observação 4.1. *Os parâmetros extras não são mandatórios, alguns são até redundantes. Por exemplo, se nós fornecermos a função-S, não precisamos fornecer o grau do polinômio candidato P_c . Se estes parâmetros redundantes forem fornecidos simultaneamente, o pacote (esperamos) fará a melhor escolha.*

4.2 Exemplos da utilização dos comandos do pacote

Nesta seção vamos mostrar o uso dos comandos do pacote *LeapS3Dim* simulando seu uso em uma plataforma Maple para exemplificar (de maneira prática) como operá-lo.

4.2.1 Lidando com 3CVPs gerais

Considere o sistema diferencial abaixo

$$\begin{cases} \dot{x} = x^3y^2 - yx^3 + 2xz \\ \dot{y} = -2x^2y^3 + 2y^2z + 2y \\ \dot{z} = 2x^2y^2z + 2x^2y + 2z^2. \end{cases} \quad (96)$$

Depois de abrir uma sessão do Maple, carregaremos os seguintes pacotes:

```
[> with(DEtools): read('LeapS3Dim.txt'):
```

O sinal de dois pontos (:) no final de uma linha de comando evita a impressão (na tela) do resultado. O sinal [> simula o 'prompt' de uma seção de Maple. O pacote *DEtools* carrega vários comandos para lidar com EDOs. O comando `read('LeapS3Dim.txt')`: carrega nosso pacote. Vamos carregar o sistema (96) digitando

```
[> sys := [x^3*y^2-x^3*y+2*x*z, -2*x^2*y^3+2*y^2*z+2*y, 2*x^2*y^2*z+2*x^2*y
+2*z^2];
```

$$sys := [x^3y^2 - yx^3 + 2xz, -2x^2y^3 + 2y^2z + 2y, 2x^2y^2z + 2x^2y + 2z^2] \quad (97)$$

O campo vetorial associado \mathfrak{X} é

$$\mathfrak{X} = (x^3y^2 - yx^3 + 2xz)\partial_x + (-2x^2y^3 + 2y^2z + 2y)\partial_y + (2x^2y^2z + 2x^2y + 2z^2)\partial_z.$$

Podemos representá-lo como um operador usando o comando `XX`. Basta digitar

```
[> X3D := XX(sys):
```

Em seguida, se quisermos seguir pela abordagem Darbouxiana clássica, precisamos calcular os polinômios de Darboux associados com o campo vetorial \mathfrak{X} . Podemos fazer isso usando o comando `Darboux`:

```
[> darbs := Darboux(sys);
```

$$darbs := [[x, x^2y^2 - x^2y + 2z], [y, -2x^2y^2 + 2yz + 2], [x^2y - z, -2x^2y + 2z]] \quad (98)$$

Em seguida, podemos tentar usar esses polinômios de Darboux para encontrar um fator integrante. Nesse caso, como estamos usando o procedimento 1 (algoritmo P_{RO1}), i.e., as equações que envolvem o termo $\mathfrak{X}\left(\frac{A}{B}\right)$, podemos usar o comando `FindA` para calcular o polinômio A em primeiro lugar e tornar o processo viável. Para a configuração $B = x^2y - z$ (o terceiro polinômio de Darboux em `darbs`) podemos digitar:

```
[> AA := FindA(sys, darbs[3][1]);
```

$$AA := 1 \quad (99)$$

Usando essas informações podemos tentar determinar um fator integrante para o sistema:

```
[> RR := InFa(sys, PD=darb, AB=[AA, darbs[3][1]]);
```

$$RR := -\frac{e^{(x^2y-z)^{-1}}}{y^2(x^2y-z)^2} \quad (100)$$

Por fim, podemos usar esse fator integrante para encontrar a integral primeira I . Para isso usamos o comando `Invar`:

```
[> Inv := Invar(sys, InF=RR);
```

$$Inv := \frac{e^{(x^2y-z)^{-1}}}{y} + Ei\left(1, -(x^2y-z)^{-1}\right) \quad (101)$$

Também poderíamos ter chegado a este resultado usando o procedimento 2 (algoritmo P_{RO2}). Podemos calcular os campos vetoriais 2D associados \mathfrak{X}_i digitando:

```
[> X2D1 := Xi2D(sys, QPN=PN, DegQPN = [7, 8, 5]);
```

$$X2D1 := [y(x^2y^2 - yz - 1), x^4y^3 - x^4y^2 - x^2y^2z + 2x^2yz - x^2y - z^2] \quad (102)$$

A saída é uma lista com os coeficientes do campo vetorial 2D associado $\mathfrak{X}_1 = N \partial_y - P \partial_z$. Podemos usar esse campo vetorial 2D para encontrar um fator integrante para o campo vetorial \mathfrak{X}

```
[> RR2 := InFa2(sys, x2d=X2D1);
```

$$RR2 := -\frac{e^{(x^2y-z)^{-1}}}{y^2(x^2y-z)^2} \quad (103)$$

4.2.2 Lidando com 1EDOs com um função elementar

Considere a 1EDO (37):

$$\frac{dy}{dx} = \frac{e^x \left((x^2 + 2x)(e^x)^2 - (2x^2y + x^2 + 4xy)e^x + x^2y^2 + 2xy^2 + y \right)}{(e^x)^2 + (-x^2 - 2y)e^x + y^2 + y}. \quad (104)$$

Vamos carregar a 1EDO (104) digitando

```
[> _1ode := diff(y(x), x) = exp(x)*((x^2+2*x)*(exp(x))^2+(-2*x^2*y(x)-x^2-4*x*y(x))*exp(x)+x^2*y(x)^2+2*x*y(x)^2+y(x))/((exp(x))^2+(-x^2-2*y(x))*exp(x)+y(x)^2+y(x));
```

Primeiro, podemos encontrar o campo vetorial associado usando o comando **XX** (sem o parâmetro **FGH** para gerar o operador \mathfrak{X} , e com o parâmetro **FGH** para determinar os polinômios F , G e H – os coeficientes de \mathfrak{X}). Digitando

```
[> X1ode := XX(_1ode, ALG=2);
```

```
[> fgh := XX(_1ode, FGH, ALG=2);
```

obtemos (o comando [`> X1ode := XX(_1ode,ALG=2)`]: atribui $X1ode$ como o campo vetorial \mathfrak{X} e [`> fgh: = XX (_1ode, FGH)`]; atribui fgh como uma lista com seus coeficientes [F, G, H]):

$$fgh := [x^2z - y^2 + 2zy - z^2 - y, -z(x^2y^2 - 2x^2yz + z^2x^2 - x^2z + 2xy^2 - 4xyz + 2xz^2 + y)]$$

Podemos calcular os campos vetoriais 2D associados \mathfrak{X}_i digitando:

[`> X2D1 := Xi2D(sys,QPN=PN)`];

$$X2D1o := [-(x^2z - y^2 + 2zy - z^2 - y), x^4y^3 - x^4y^2 - x^2y^2z + 2x^2yz - x^2y - z^2] \quad (105)$$

Usando o campo 2D \mathfrak{X}_1 podemos (aplicando o método da função- S – veja (3)) encontrar uma integral primeira para o campo vetorial \mathfrak{X} e a solução geral da 1EDO:

[`> Sol := Solde(_1ode,x2d=X2D1o,II)`];

$$Sol := \left[\frac{-\ln(x^2z - y)z + \ln(x^2z - y)y - 1}{-z + y}, z = e^x \right] \quad (106)$$

[`> Inv := Sol[1]`];

$$Inv := \frac{-\ln(x^2z - y)z + \ln(x^2z - y)y - 1}{-z + y} \quad (107)$$

[`> solger := subs(Sol[2],Sol[1])=c`];

$$solger := e^{(e^x - y)^{-1}} (e^x x^2 - y) = c \quad (108)$$

Podemos confirmar que I é uma integral primeira do campo vetorial \mathfrak{X} digitando

[`> X1ode(Inv)`];

0

E podemos testar a solução encontrada com:

[`> DX := Dx(_1ode)`];

[`> DX(lhs(solger))`];

0

4.2.3 Lidando com 2EDOs racionais

Considere a seguinte 2EDO:

$$z' = \frac{x^5 z - x^4 z^2 - 3zx^4 + 4x^3 z^2 - xy + xz + yz - z^2 - y}{x^5 - y}. \quad (109)$$

Entramos com a 2EDO digitando:

```
[> _2ode := diff(y(x),x,x)=(x^5*(diff(y(x),x))-x^4*(diff(y(x),x))^2-
3*(diff(y(x),x))^2-x*y(x)+x*(diff(y(x),x))+y(x)*(diff(y(x),x))^2-x*y(x)+
x*(diff(y(x),x))+y(x)*(diff(y(x),x))^2-y(x))/(x^5-y(x)):
```

O campo vetorial associado \mathfrak{X} é

$$\mathfrak{X} = (x^5 - y)\partial_x + z(x^5 - y)\partial_y + (x^5 z - x^4 z^2 - 3zx^4 + 4x^3 z^2 - xy + xz + yz - z^2 - y)\partial_z.$$

Podemos representá-lo como um operador usando o comando **XX**. Basta digitar

```
[> X2ode := XX(_2ode,ALG=3):
```

Podemos calcular os campos vetoriais 2D associados \mathfrak{X}_i digitando:

```
[> X2D2o := Xi2D(_2ode);
```

$$X2D2o := [x - z, x^5 - y] \quad (110)$$

Podemos usar o método da função S para encontrar a integral primeira I . Para isso usamos o comando **Invar**:

```
[> Inv := Invar(_2ode,x2d=X2D2o);
```

$$Inv := \frac{(zx^4 - y)e^{-x}}{-z + x} \quad (111)$$

Podemos testar a integral primeira:

```
[> X2ode(Inv);
```

5 DESEMPENHO

Neste capítulo vamos discutir o desempenho dos procedimentos desenvolvidos nos capítulos 1, 2 e 3. Vamos estruturar este capítulo da seguinte maneira:

1. Na primeira seção, comparamos os procedimentos apresentados no capítulo 1 (para 3CVPs em geral): comparamos os tempos gastos pelos algoritmos P_{RO1} e P_{RO2} em uma implementação em Maple²⁰ (pacote *LeapS3Dim*).
2. Na segunda seção, apresentamos alguns exemplos de 1EDOs com funções elementares e sua solução com o algoritmo de ‘redução’ linear ($P_{RO3,2}$). Também mostramos os tempos gastos e a quantidade de memória utilizada em cada caso.
3. Na terceira subseção, comparamos o método apresentado no capítulo 3 com seu ‘predecessor’ (o método apresentado em (3)) para duas 2EDOs racionais. Também mostramos, como nas subseções anteriores, os tempos de CPU gastos e a quantidade de memória RAM utilizada.
4. Na última seção, discutimos um pouco sobre as vantagens e desvantagens, limitações e generalizabilidade dos algoritmos apresentados.

Observação 5.1. *Neste trabalho, todos os programas / informações computacionais – tempo de CPU gasto, consumo de memória etc – foram rodados / obtidas em um mesmo Laptop que apresenta a seguinte configuração: Intel(R) Core(TM) i5-8265U @ 1.8 GHz.*

5.1 Procedimento 1 (P_{RO1}) \times Procedimento 2 (P_{RO2})

Nesta seção apresentamos dois sistemas diferenciais que são muito complicados para a grande maioria dos métodos. Estes dois sistemas são uma pequena amostra de uma vasta classe cujos membros são muito refratários à grande maioria dos métodos. Assim, escolhemos apenas dois sistemas nos quais fica evidente que os algoritmos P_{RO1} e P_{RO2} atuam eficientemente em ‘áreas’ disjuntas.

²⁰ Maple é um sistema de computação algébrica de propósito geral (CAS) que é também uma linguagem de programação.

Considere os seguintes sistemas:

$$\begin{cases} \dot{x} = 2x(xz + 2y^2 - x)(xz - 2y^2) \\ \dot{y} = x^2yz^2 - 3y^5 + 2x^3z + 4x^2y^2 - x^2yz + 2xy^3 - x^3 - x^2z - y^2x \\ \dot{z} = -x^2z^3 + 3y^4z - 4x^2yz - 8xy^3 + 2x^2y + 2xyz + 2y^3. \end{cases} \quad (112)$$

$$\begin{cases} \dot{x} = x^3y^2 - yx^3 + 2xz \\ \dot{y} = -2x^2y^3 + 2y^2z + 2y \\ \dot{z} = 2x^2y^2z + 2x^2y + 2z^2. \end{cases} \quad (113)$$

Após abrirmos uma sessão de Maple, ‘entrarmos’ com os sistemas

```
[> sys[1] := [ 2*x*(x*z+ ...
[> sys[2] := [X^3*y^2- ...
```

e aplicarmos o comando `Invar` aos dois sistemas usando o parâmetro `PRO` que diz ao comando `Invar` qual o algoritmo a ser seguido (veja os detalhes do comando `Invar` na seção 4.1.11):

```
[> Inv[1][1] := Invar(sys[1],PRO=1,...);
[> Inv[1][2] := Invar(sys[1],PRO=2,...);
[> Inv[2][1] := Invar(sys[2],PRO=1,...);
[> Inv[2][2] := Invar(sys[2],PRO=2,...);
```

Ambos encontram para o sistema [1]:

$$Inv[1][1] := e^{\frac{x}{xz+y^2}} (-zy + x) + Ei \left(1, -\frac{x}{xz + y^2} \right) \quad (114)$$

e

$$Inv[1][2] := e^{\frac{x}{xz+y^2}} (-yz + x) + Ei \left(1, -\frac{x}{xz + y^2} \right) \quad (115)$$

Ambos encontram para o sistema [2]:

$$Inv[2][1] := - \left(-yEi \left(1, (-x^2y + z)^{-1} \right) - e^{-(x^2y+z)^{-1}} \right) y^{-1} \quad (116)$$

e

$$Inv[2][2] := \frac{e^{(x^2y-z)^{-1}}}{y} + Ei \left(1, - (x^2y - z)^{-1} \right) \quad (117)$$

O desempenho dos algoritmos P_{RO1} e P_{RO2} :

- Para o procedimento 1 (P_{RO1}):

Sistema	Tempo (seg.)	Memória (MB)
[1] (112)	2	42
[2] (113)	13	150

- Para o procedimento 2 (P_{RO2}):

Sistema	Tempo (seg.)	Memória (MB)
[1] (112)	20	260
[2] (113)	0,9	16

Observação 5.2. Alguns comentários:

- Quando iniciamos uma sessão de Maple, só o fato de carregarmos os pacotes básicos (que compõe a estrutura central) que são ativados com a abertura da plataforma padrão do programa já é responsável pelo consumo de mais de 4 MB de memória. Por isso, se após alguma operação, o número que o aparece (indicando o gasto de memória) é perto de 5 (MB), o que realmente está sendo ‘dito’ é: a operação, praticamente, não consome memória.
- Em relação à memória consumida e ao tempo gasto pelos algoritmos (materializados nos comandos do pacote *LeapS3Dim*), colocamos apenas os Algarismos que nos pareceram significativos (e não exatamente os números que apareceram na seção).
- Colocamos dois exemplos que ‘demonstram’ o que foi comentado nas observações finais do capítulo 2 (observações 1.23 e 1.24): no sistema [1] (112) o procedimento 1 (que usa a abordagem Darboxiana mais clássica, i.e., o cômputo dos polinômios de Darboux para determinar um fator integrante) foi mais eficiente que o procedimento 2 (que determina os campos vetoriais associados $2D \mathfrak{X}_i$), ao passo que no sistema [2] (113) o procedimento 2 é que foi mais eficiente.
- Um detalhe muito importante e que tem relação com as reticências colocadas nas entradas do comando `Invar` (`[> Inv[1][1] := Invar(sys[1], PRO=1, ...)`): Os números que indicam a memória consumida e o tempo gasto que estão sendo mostrados nas tabelas acima só dizem respeito à setagem final dos parâmetros, i.e., não está sendo levado em conta o tempo gasto nas configurações que não levam a nenhum resultado. Mais claramente, o tempo (e consumo de memória) das tentativas frustradas não está sendo computado.

- *Em relação ao comentário acima, esses tempos (das tentativas do comando que resultam FAIL) não foram computados porque não temos o conhecimento prévio das setagens que resultam em sucesso. Na verdade, isso ainda está incipiente tanto na teoria quanto no programa, isto é, não sabemos ainda (nem heurísticamente, uma vez que não fizemos testes suficientes) a configuração ‘ótima’ (para colocar como default nos valores dos parâmetros nas entradas dos comandos) que os parâmetros devem ter (na média estatística dos casos) para que a solução seja determinada em um tempo mínimo.*
- *Em relação ao ‘demonstram’ que aparece no terceiro comentário, parece bastante suspeito o fato de haver apenas dois exemplos. O que dois exemplos demonstram? Num sentido mais amplo, dois exemplos sequer podem ‘mostrar’ algo estatisticamente, quanto mais demonstrar. O ponto de termos colocado somente estes exemplos se deve a (principalmente) três razões:*
 1. *Não conseguimos ainda realizar uma grande bateria de testes para nos dar uma ‘visão estatística’ mais precisa do que acontece na maioria dos casos ‘em média’.*
 2. *Gerar uma base estatística de confiável também não é muito fácil. Não estamos nos referindo aqui apenas ao número de exemplos que é necessário, mas também à dificuldade (de resolução por parte do algoritmo) de cada um dos exemplos. Devido ao fato que os procedimentos P_{RO1} e P_{RO2} ‘atacam’ sistemas muito difíceis para outros métodos, não dispomos de uma ‘arena comum’ que seja interessante para nós. Explicando melhor: Vamos supor que tal arena comum existisse e vamos denominá-la AC. Poderia ser que, por exemplo, o algoritmo P_{RO1} fosse muito muito mais eficiente que o P_{RO2} em, digamos, 80% dessa arena comum AC (vasta e confiável). Contudo, vamos supor que quase todos os casos que correspondem a esses 80% fossem resolvidos com facilidade por outros métodos. Se isso acontecesse e o P_{RO2} fosse muito mais eficiente que o P_{RO1} nos 20% restantes dos casos (que nenhum dos outros métodos consegue resolver), a conclusão seria que, do ponto de vista prático, o algoritmo P_{RO2} é muito melhor que o algoritmo P_{RO1} , apesar de perder ‘feio’ nos sistemas ‘fáceis’ (pois nesses sistemas teríamos uma variedade de outros métodos para dar conta do problema).*
 3. *Ainda em relação à questão anterior (a dificuldade de estabelecer uma arena comum vasta e confiável) mas ligado a outra origem temos o seguinte problema: vamos ‘aumentando’ a dificuldade dos sistemas simplesmente aumentando o grau dos polinômios envolvidos. Contudo, como o método de resolver o sistema algébrico de coeficientes indeterminados (as equações quadráticas citadas na observação 1.23) é baseado na teoria das bases de Gröbner (veja (39)), não*

sabemos se a dificuldade (de resolução) de cada sistema que geramos está (pelo menos estatisticamente) diretamente ligada (i.e., proporcionalmente ligada) à dificuldade gerada pelo aumento no grau dos polinômios. Além disso, como os sistemas são quadráticos e o método das bases de Gröbner é construído para sistemas polinomiais gerais (de qualquer grau), pode haver outros métodos mais específicos (para solucionar sistemas quadráticos ‘do nosso tipo’) que não apenas melhorem a eficiência como também mudem o tempo de solução de maneira não proporcional.

5.2 O Algoritmo $P_{RO3.2}$ – 1EDOs com uma função elementar

Nesta seção também vamos ser ‘econômicos’ com o número de exemplos. Além de motivos análogos aos citados anteriormente, neste caso, estamos apenas testando uma implementação com o algoritmo linear $P_{RO3.2}$ em comparação com o algoritmo $P_{RO3.1}$ que já foi testado em (4). Assim, vamos apenas mostrar o ganho em termos de memória e tempo para dois exemplos e mostrar um exemplo de 1EDO com uma função algébrica.

Considere as seguintes 1EDOs:

$$\frac{dy}{dx} = \frac{(e^y y^3 + x^2)(e^y y^5 - x^2 y^2 + x)y}{x((e^y)^2 y^9 + 3(e^y)^2 y^8 - e^y x^2 y^6 - 2(e^y)^2 y^6 - 3e^y x^2 y^5 + 4e^y x^2 y^3 + x e^y y^4 + 3x e^y y^3 - 2x^4)} \quad (118)$$

$$\frac{dy}{dx} = -\frac{(4x^4 y^3 + 1)(x^4 y^3 + \ln(x) - y)}{x((\ln(x))^2 + (2x^4 y^3 + 3x^4 y^2)\ln(x) + x^8 y^6 + 3x^8 y^5 - 3x^4 y^3)} \quad (119)$$

Após abrirmos uma sessão de Maple, ‘entrarmos’ com os sistemas

```
[> _1ode[1] := ...
[> _1ode[2] := ...
```

e aplicarmos o comando `Invar` aos dois sistemas usando o parâmetro `PRO` que diz ao comando `Invar` qual o algoritmo a ser seguido (veja os detalhes do comando `Invar` na seção 4.1.11):

```
[> x2d[1][1] := Xi2D(_1ode[1], PRO=31, ...);
[> x2d[1][2] := Xi2D(_1ode[1], PRO=32, ...);
```

```
[> x2d[2][1] := Xi2D(_1ode[2],PRO=31,...);
[> x2d[2][2] := Xi2D(_1ode[2],PRO=32,...);
```

Ambos encontram para a $_1ode$ [1]:

$$X2[1][1] := [-3y^8z^2 + 3x^2y^5z + 2y^6z^2 - 4x^2y^3z - 3xzy^3 + 2x^4, (-y^5z + x^2y^2 - x)y^4] \quad (120)$$

e

$$X2[1][2] := [-3y^8z^2 + 3x^2y^5z + 2y^6z^2 - 4x^2y^3z - 3xzy^3 + 2x^4, (-y^5z + x^2y^2 - x)y^4] \quad (121)$$

Ambos encontram para a $_1ode$ [2]:

$$X2[2][1] := [-3y^2(x^4y^3 - x^4z + z), x^4y^3 - y^6 - x^4z + 2y^3z - z^2 + z] \quad (122)$$

e

$$X2[2][2] := [-3y^2(x^4y^3 - x^4z + z), x^4y^3 - y^6 - x^4z + 2y^3z - z^2 + z] \quad (123)$$

O desempenho dos algoritmos $P_{RO3.1}$ e $P_{RO3.2}$:

- Para o procedimento $P_{RO3.1}$:

1EDO	Tempo (seg.)	Memória (MB)
[1] (118)	2,8	42
[2] (119)	11	42

- Para o procedimento 2 ($P_{RO3.2}$):

1EDO	Tempo (seg.)	Memória (MB)
[1] (118)	0,2	17
[2] (119)	0,5	17

Considere agora a 1EDO

$$\frac{dy}{dx} = \frac{y(-xy^7 + 2\sqrt{xy}y^5 - x^2y^4 + 2x\sqrt{xy}y^2 - \sqrt{xy}y^3 - y^4 + \sqrt{xy}x - xy)}{x(-5xy^7 + 10\sqrt{xy}y^5 - x^2y^4 + 4xy^5 + 2x\sqrt{xy}y^2 - \sqrt{xy}y^3 - 4x^2y^2 - 5y^4 + \sqrt{xy}x - xy)}$$

(124)

O algoritmo encontra a solução em 0,2 segundos (e usando 17Mb de memória):

$$e^{\frac{x}{\sqrt{xy-xy^2}}} \sqrt{xy} (-y^3 + x)^{-1} = c \quad (125)$$

5.3 O Algoritmo $P_{RO4.2}$ – 2EDOs racionais

Nesta seção, assim como na anterior, em relação ao desempenho prático dos algoritmos, vamos apenas comentar os ganhos relativos de tempo e memória por conta dos aperfeiçoamentos realizados. Na seção seguinte faremos uma breve discussão dos benefícios teóricos e no que resultaram para as melhorias de desempenho.

Considere as seguintes 2EDOs:

$$z' = \frac{z(x^2z^5 - x^2z^4 + 2xyz^3 - xz^4 - 2xyz^2 - yz^3 + xz^2 + y^2z - 2yz^2 - y^2 - yz)}{(xz^2 - xz + yz + y)(xz^2 - y)}, \quad (126)$$

e

$$z' = (x^5z^7 + 3x^4yz^6 + 2x^5yz^4 + 6x^4y^2z^3 + x^5y^2z - 2x^4z^4 + x^3yz^4 + 3x^4y^3 + x^4yz^2 - 6x^3yz^3 - 2x^4yz + x^3y^2z - 6x^3y^2 - z^5 + x^3z - xz^3 + 3x^2y - yz^2)/(-2x^4yz^3 + x^2z^6 + x^4y^2 + 2x^2yz^3 + 2xz^4 - x^3y + x^2y^2 - 2xz^3 - xyz - 2xy + z + 1). \quad (127)$$

Após abrirmos uma sessão de Maple, ‘entrarmos’ com as 2EDOs

```
[> _2ode[1] := ...
```

```
[> _2ode[2] := ...
```

e aplicarmos o comando `Xi2D` à primeira 2EDO usando o parâmetro `PRO` (que diz ao comando qual o algoritmo a ser seguido):

```
[> x2d[1][1] := Xi2D(_2ode[1], PRO=4, ...);
```

```
[> x2d[1][2] := Xi2D(_2ode[1], PRO=42, ...);
```

O primeiro encontra para a `_2ode[1]`:

$$X2[1][1] := [-z(x^2z^4 + 2xyz^2 - xz^2 + xz + y^2 - yz - y), (xz^2 - xz + yz + y)(xz^2 - y)]$$

e o segundo não encontra nada:

$$X2[1][2] := \quad (128)$$

O que acontece é que o polinômio zP tem um termo monomial que cancela com outro de Q . Fazendo a transformação $T = \{x = x^2, y = y\}$ na 2EDO, obtemos

$$\begin{aligned} z' = & -z(2x^3z^4 - z^5x^2 + 16x^3yz^2 - 8x^2yz^3 + 32x^3y^2 - 8x^3z^2 - \\ & 16x^2y^2z + 2x^2z^3 + xz^4 + 8x^2yz + 16xyz^2 + 2yz^3 + 16xy^2 + \\ & 8zy^2)/(x(-z^2 + 4y)(2zx^2 - xz^2 - 4xy - 2yz)) \end{aligned} \quad (129)$$

e ambos encontram

$$\begin{aligned} X2[1][1] := & [-(xz^4 + 8xyz^2 + 8zx^2 + 16xy^2 - 4xz^2 - 16xy - 8yz)z, \\ & (-z^2 + 4y)(2zx^2 - xz^2 - 4xy - 2yz)] \end{aligned}$$

e

$$\begin{aligned} X2[1][2] := & [-(xz^4 + 8xyz^2 + 8zx^2 + 16xy^2 - 4xz^2 - 16xy - 8yz)z, \\ & (-z^2 + 4y)(2zx^2 - xz^2 - 4xy - 2yz)] \end{aligned}$$

Agora aplicando o comando Xi2D à segunda 2EDO:

```
[> x2d[2][1] := Xi2D(_2ode[1], PR0=4, ...);
[> x2d[2][2] := Xi2D(_2ode[1], PR0=42, ...);
```

Ambos encontram:

$$\begin{aligned} X2[2][1] := & [-x(x^4z^6 + 2x^4yz^3 + x^4y^2 - 2x^3z^3 + x^3yz - 2x^3y + x^2 - z^2), -2x^4yz^3 + \\ & x^2z^6 + x^4y^2 + 2x^2yz^3 + 2xz^4 - x^3y + x^2y^2 - 2xz^3 - xyz - 2xy + z + 1] \end{aligned}$$

e

$$\begin{aligned} X2[2][2] := & [-x(x^4z^6 + 2x^4yz^3 + x^4y^2 - 2x^3z^3 + x^3yz - 2x^3y + x^2 - z^2), -2x^4yz^3 + \\ & x^2z^6 + x^4y^2 + 2x^2yz^3 + 2xz^4 - x^3y + x^2y^2 - 2xz^3 - xyz - 2xy + z + 1] \end{aligned}$$

O desempenho dos algoritmos P_{RO4} e $P_{RO4.2}$:

- Para o procedimento P_{RO4} :

2EDO	Tempo (seg.)	Memória (MB)
[1t] (129)	3,8	42
[2] (127)	125	345

- Para o procedimento $P_{RO4.2}$:

1EDO	Tempo (seg.)	Memória (MB)
[1t] (118)	0,06	17
[2] (119)	0,08	17

As respectivas integrais primeiras são:

$$Inv[1t] = e^{\frac{2z}{x(z^2+4y)}} (x^2 - y) + Ei \left(1, -2 \frac{z}{x(z^2 + 4y)} \right) \quad (130)$$

e

$$Inv[2] = \frac{e^{\frac{z}{xz^3+xy-1}}}{x^3y - z} \quad (131)$$

5.4 Uma breve discussão sobre os algoritmos

Nesta seção vamos comentar um pouco sobre o desempenho dos métodos partindo do ponto de vista de sua construção. Em primeiro lugar vamos falar dos algoritmos mais genéricos, P_{RO1} e P_{RO2} e depois, sobre os mais específicos P_{RO3} e P_{RO4} .

Observação 5.3. *O dsolve, resolvidor do Maple, não consegue resolver um exemplo sequer dos mostrados aqui.*

5.5 Os procedimentos 1 e 2

O procedimento 1 (como havíamos dito) é um procedimento Darbouxiano clássico, e assim, sua obstrução principal costuma ser a dificuldade de computarmos polinômios de Darboux de grau elevado. Como a maioria dos métodos de cálculo de polinômios de Darboux (PDs) tendem a crescer muito em complexidade a medida que o grau dos polinômios aumenta, esta é (a dificuldade de computar os PDs) a principal obstrução aos métodos Darbouxianos clássicos. Além disso, essa vertente tinha sido descartada por nós há alguns anos porque, além dessa dificuldade, nos casos em que o sistema apresentava uma integral primeira Liouvillianiana não-elementar, o sistema algébrico dos coeficientes era cúbico em alguns deles e isso tornava o processo inviável. Com a melhoria que vislumbramos (usar o fato que $\mathfrak{X} \left(\frac{A}{B} \right) \in \mathbb{C}[x, y, z]$ – veja a observação 1.20) o procedimento 1 se tornou bastante

mais palatável, embora ainda dependa da determinação dos polinômios de Darboux. Outra grande vantagem (teórica / prática) que decorre desse detalhe com essa melhoria é que temos uma maneira semi-algorítmica (razoavelmente prática) de encontrar integrais primeiras Liouvillianas não-elementares de sistemas diferenciais em três dimensões.

O procedimento 2 é focado em evitar o problema de determinar os polinômios de Darboux: a ideia central é construir campos vetoriais polinomiais em duas dimensões (que neste trabalho, denotamos por \mathfrak{X}_i) que apresentam como integral primeira a mesma função que é uma integral primeira do 3CVP \mathfrak{X} . O processo de determinar esses campos vetoriais associados é também quadrático nos coeficientes indeterminados e, assim, o procedimento 2 não apresenta uma superioridade global sobre o procedimento 1. Isto é, ambos são processos que envolvem a resolução de sistemas algébricos quadráticos e, portanto, em certas situações o procedimento 1 é superior e, em outras, o procedimento 2 é o vencedor.

Observação 5.4. *Em geral, nos casos em que os polinômios de Darboux são mais fáceis de calcular o P_{RO1} é mais eficiente e nos casos em que um dos campos vetoriais associados é mais simples, o P_{RO2} leva vantagem (veja a seção 5.1).*

Observação 5.5. *Um grande avanço (do ponto de vista teórico) consiste no resultado apresentado pelo teorema 1.8: Para qualquer sistema diferencial 3D que apresente uma integral primeira Liouvilliana em L_S , temos que o campo vetorial associado \mathfrak{X} possui um fator integrante Darbouxiano (i.e., da forma $e^{(A/B)} \prod_i p_i^{n_i}$). Isso nos permite, por exemplo, usar o fato que $\frac{\mathfrak{X}(R)}{R} \in \mathbb{C}[x, y, z]$ e que torna viável o procedimento 1.*

Ainda falta talvez uma maneira de calcular os polinômios de Darboux de uma maneira mais eficiente, ou por outro lado, encontrar uma forma mais eficiente de calcular os 2CVP \mathfrak{X}_i associados. No caso geral não encontramos (ainda) uma maneira de computar apenas um polinômio do campo associado nem uma maneira muito eficiente de calcular um polinômio de Darboux em três variáveis.

5.6 Os procedimentos 3.1 e 3.2

No caso de um campo vetorial polinomial 3D \mathfrak{X} representando uma 1EDO com uma função elementar implementamos um algoritmo capaz de determinar os campos vetoriais 2D associados \mathfrak{X}_i linearmente. O algoritmo implementado tinha sido criado em (4). Neste trabalho pudemos tornar mais clara a definição da classe \mathcal{L}_S e assim, incluir nos algoritmos 3.1 e 3.2 as 1EDOs com funções algébricas. Também pudemos esclarecer a questão da existência de fatores polinomiais comuns aos polinômios P e $Q + zN$ e mostrar que essa obstrução é apenas aparente, i.e., com uma transformação simples de variáveis podemos suspender o problema. Por causa desse detalhe pudemos tornar o procedimento 3.2 um algoritmo completo. As consequências práticas desse algoritmo linear são bem claras

quando examinamos os tempos de CPU gastos e a quantidade de memória consumida: são praticamente nulos. Assim, nessa direção temos um caminho promissor para seguir.

5.7 Os procedimentos 4 e 4.2

Também no caso de um campo vetorial polinomial 3D \mathfrak{X} representando uma 2EDO racional pudemos fazer aprimoramentos pequenos mas com grandes consequências teóricas e práticas. A base para o que fizemos neste trabalho foi desenvolvida em (3), mas apenas para os casos em que o denominador da 2EDO coincidia com o polinômio N . Pudemos mostrar que:

- Os casos em que o denominador da 2EDO não é o polinômio N são devidos a cancelamentos de fatores comuns aos polinômios N e $Q + zP$. Esses cancelamentos podem ser evitados (como mostramos no capítulo 3) facilmente com uma transformação simples como $T = \{x = y, y = x\}$.
- O item anterior nos permite transformar (sem dificuldade) qualquer 2EDO racional (com uma integral primeira Liouvillian) em uma outra tal que o denominador é o polinômio N . Esse fato tem uma consequência teórica: **o algoritmo que obtêm os 2CVPs associados é um algoritmo completo**; e uma consequência prática: um dos polinômios que compõe \mathfrak{X}_i é previamente determinado e, portanto, o algoritmo é muito mais eficiente (uma vez que temos que determinar apenas um polinômio).
- Os casos de 2EDOs nas quais existem termos monomiais em P que não estão em M_{0z} (devido ao fato que termos de zP cancelam com termos de Q) podem ser tratados com facilidade (com uma transformação simples como $T = \{x = x^2, y = y\}$ – veja a seção 5.3). Além da abrangência, os casos em que não há cancelamentos desse tipo são muito mais eficientes (veja o exemplo da 2EDO (127)).

CONCLUSÃO

Neste trabalho desenvolvemos um novo método (na verdade aperfeiçoamos dois procedimentos antigos) para determinar integrais primeiras Liouvillianas de campos vetoriais polinomiais em três dimensões. Para campos vetoriais tais que a integral primeira I é um membro da classe L_S (veja a definição 1.6), pudemos mostrar que ela apresenta um fator integrante Darbouxiano. Este fato nos permite gerar dois procedimentos (semi-algoritmos P_{RO1} e P_{RO2}) para encontrar integrais primeiras Liouvillianas

Os métodos desenvolvidos se apresentaram de forma eficiente em ‘áreas’ de atuação diferentes o que é um ponto muito positivo. Contudo, para a grande maioria dos casos ‘complicados’ fica muito difícil avaliar a eficiência dos nossos algoritmos, uma vez que não existe uma ‘arena’ razoavelmente vasta e, além disso, na maioria dos casos complicados o nosso método é a única opção que consegue algum sucesso (veja os comentários na observação 5.2, no fim da seção 5.1).

Nos capítulos 2 e 3, desenvolvemos vários resultados teóricos que nos permitiram aprimorar (em muito) a eficiência dos procedimentos 1 e 2: no capítulo 2, graças a uma generalização do teorema 2.1 (veja o teorema 2.2) pudemos incluir as 1EDOs que apresentam funções algébricas e implementar um algoritmo linear para encontrar os campos vetoriais 2D \mathfrak{X}_i associados. E no capítulo 3 pudemos mostrar que o algoritmo 4 é um algoritmo completo e, além disso, conseguir computar os campos vetoriais 2D \mathfrak{X}_i associados com um algoritmo tão rápido que até parece linear (veja a 2EDO (127) e o tempo gasto pelo algoritmo para encontrar \mathfrak{X}_i).

Os métodos (que usam os algoritmos desenvolvidos nos capítulos 1, 2 e 3) foram implementados em um pacote computacional na plataforma de computação simbólica Maple. O pacote, além de encontrar as integrais primeiras Liouvillianas usando os métodos citados acima, apresenta comandos que materializam todas as etapas do processo de resolução. Os comandos também possuem vários parâmetros que ajudam tanto no processo de busca de integrais primeiras como também na pesquisa em física e matemática.

Também discutimos o desempenho do pacote (e dos algoritmos que desenvolvemos) brevemente, no capítulo 5.

Resultados que queremos buscar:

1. Pesquisar uma maneira (talvez na linha do método de A. Ferragut (31)) de tornar mais eficiente o cômputo dos polinômios de Darboux em três variáveis.
2. A determinação dos 2CVPs associados para sistemas diferenciais polinomiais gerais de uma maneira mais eficiente.

REFERÊNCIAS

- 1 EIRAS, J. P. C. *Sistemas 3D de 1EDOs: a Busca por invariantes Liouvilianos*. 2017. 98f. Dissertação (Mestrado) - Instituto de Física Armando Dias Tavares, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2017
- 2 DUARTE, L. G. S; MOTA, L. A. C. P. 3D polynomial dynamical systems with elementary first integrals. *J. Phys. A: Math. Theor.*, Bristol, v. 43, n. 6, p. 065204, Jan. 2010.
- 3 AVELLAR, J; CARDOSO, M. S; DUARTE, L. G. S; MOTA, L. A. C. P. Dealing with Rational Second Order Ordinary Differential Equations where both Darboux and Lie Find It Difficult: The S-function Method. *Computer Physics Communications*, [S.l.], v. 234, p. 302-314, 2019.
- 4 QUEIROZ, A. B. M. M. *Um Novo Método para Buscar Soluções Liouvilianas de Equações Diferenciais Ordinárias de Primeira Ordem que Apresentam Funções Elementares*. 2020. 94f. Tese (Doutorado) - Instituto de Física Armando Dias Tavares, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2020.
- 5 DARBOUX, G. Mémoire sur les équations différentielles algébriques du premier ordre et du premier degré (Mélanges). *Bull. Sci. Math.*, Paris, 2e.éd. v. 32, p. 60-96, 1878.
- 6 JOUANOLOU, J. P. *Equations de Pfaff algébriques*. New York: Springer, 1979, Lectures Notes in Mathematics, v. 708.
- 7 SCHLOMIUK, D. Algebraic particular integrals, integrability and the problem of the center. *Trans. Am. Math. Soc.*, [S.l.] v.338, n.2, p.799–841, Ago. 1993.
- 8 PRELLE, M; SINGER, M. Elementary first integral of differential equations. it *Trans. Amer. Math. Soc.*, [S.l.] v. 279, p. 215-229, Sep.1983.
- 9 SINGER, M. Liouvillian First Integrals, it *Trans. Amer. Math. Soc.*, [S.l.] v. 333, p.673-688, Oct.1992.
- 10 CHRISTOPHER, C. Liouvillian first integrals of second order polynomial differential equations. *Electron. J. Differential Equations*, [S.l.], v.1999, n. 49, p. 1-7, Dec.1999.
- 11 DAVENPORT, J. H; SIRET, Y; TOURNIER, E. *Computer Algebra: Systems and Algorithms for Algebraic Computation*. London: Academic Press, 1993
- 12 DUARTE, L. G. S; DUARTE, S. E. S; MOTA, L. A. C. P; SKEA, J. F. E. Extension of the Prelle-Singer Method and a MAPLE implementation. *Computer Physics Communications*, Holanda, v. 144, n. 1, p. 46-62, 2002.

- 13 DUARTE, L. G. S; DUARTE, S. E. S; MOTA, L. A. C. P. A method to tackle first order ordinary differential equations with Liouvillian functions in the solution. *J. Phys. A: Math. Gen.*, [S.l.], v. 35, p. 3899-3910, 2002.
- 14 DUARTE, L. G. S; DUARTE, S. E. S; MOTA, L. A. C. P. Analyzing the Structure of the Integrating Factors for First Order Ordinary Differential Equations with Liouvillian Functions in the Solution. *J. Phys. A: Math. Gen.*, [S.l.], v. 35, p. 1001-1006, 2002.
- 15 AVELLAR, J; DUARTE, L. G. S; DUARTE, S. E. S; MOTA, L. A. C. P. Integrating First-Order Differential Equations with Liouvillian Solutions via Quadratures: a Semi-Algorithmic Method. *Journal of Computational and Applied Mathematics.*, Philadelphia, v. 182, n. 2, p. 327-332, Oct. 2005
- 16 AVELLAR, J; DUARTE, L. G. S; DUARTE, S. E. S; MOTA, L. A. C. P. Determining Liouvillian first integrals for dynamical systems in the plane. *Computer Physics Communications.*, [S.l.], v. 177, p. 584-596, Oct. 2007.
- 17 CHÈZE, G; COMBOT, T. Symbolic Computations of First Integrals for Polynomial Vector Fields. it *Foundations of Computational Mathematics.*, [S.l.], v. 20, p. 681–752, 2019. <https://doi.org/10.1007/s10208-019-09437-9>
- 18 DARBOUX, G. De l'emploi des solutions particulières algébriques dans l'intégration des systèmes d'équations différentielles algébriques. *C. R. Math. Acad. Sci.*, Paris, v. 86, p. 1012–1014, 1878.
- 19 COUTINHO, S. C; SCHECHTER, L. M. Algebraic solutions of plane vector fields. *J. Pure Appl. Algebra.*, [S.l.], v. 213, p. 144–153, Jan. 2009. <https://doi.org/10.1016/j.jpaa.2008.06.003>
- 20 CHRISTOPHER, C; LLIBRE, J; PATANZI, C; ZHANH, X. Darboux integrability and invariant algebraic curves for planar polynomial systems. *J. Phys. A.*, [S.l.], v. 35, p. 2457–2476, 2002. <https://doi.org/10.1088/0305-4470/35/10/310>
- 21 CHAVARRIGA, J; GIACOMINI, H; GINÉ, J; LLIBRE, J. Darboux integrability and the inverse integrating factor. *J. Differential Equations.*, NL, v.194, p.116–139. Oct. 2003. [https://doi.org/10.1016/S0022-0396\(03\)00190-6](https://doi.org/10.1016/S0022-0396(03)00190-6)
- 22 CHRISTOPHER, C; LLIBRE, J; PATANZI, C; WALCHER, S. Inverse problems for multiple invariant curves. *Proc. Roy. Soc.*, Edinburgh UK, v. 137A, p. 1197 – 1226, Dec. 2007. <https://doi.org/10.1017/S0308210506000400>
- 23 CHRISTOPHER, C; LLIBRE, J; PEREIRA, J. V. Multiplicity of Invariant Algebraic Curves in Polynomial Vector Fields. *Pacific Journal of Mathematics.*, USA, v. 229, n. 1, p. 63-117, Jan. 2007. <https://doi.org/10.2140/pjm.2007.229.63>

- 24 CHRISTOPHER, C; LLIBRE, J; PATANZI, C; WALCHER, S. Inverse problems for invariant algebraic curves: Explicit computations. *Proc. Roy. Soc.*, Edinburgh v.139, p. 287-302, 2009 <https://doi.org/10.1017/S0308210507001175>
- 25 CHRISTOPHER, C; LLIBRE, J; PATANZI, C; WALCHER, S. Darboux integrating factors: inverse problems. *J. Differential Equations.*, NL, v. 250, p. 1-25, Jan. 2011. <https://doi.org/10.1016/j.jde.2010.10.013>
- 26 CHRISTOPHER, C; LLIBRE, J; PATANZI, C; WALCHER, S. Inverse Problems in Darboux' Theory of Integrability. *Acta Applicandae Mathematicae.*, [S.l.] v. 120, p. 101–126, Aug.2012 <https://doi.org/10.1007/s10440-012-9671-9>
- 27 ZHANG, X. Liouvillian integrability of polynomial differential systems. *Trans. Amer. Math. Soc.*, [S.l.] v. 368, p. 607-620, 2016. <https://doi.org/10.1090/S0002-9947-2014-06387-3>
- 28 FERRAGUT, A; GIACOMONI, H. A New Algorithm for Finding Rational First Integrals of Polynomial Vector Fields. *Qual. Theory Dyn. Syst.*, Spain, v. 9, p. 89–99, Aug. 2010 <https://doi.org/10.1007/s12346-010-0021-x>
- 29 CHÈZE, G. Computation of Darboux polynomials and rational first integrals with bounded degree in polynomial time. *Journal of Complexity.*, USA, v. 27, p. 246-262, Apr. 2011 <https://doi.org/10.1016/j.jco.2010.10.004>
- 30 BOSTAN, A; CHÈZE, G; CLUZEAU, T; WEIL, J.-A. Efficient algorithms for computing rational first integrals and Darboux polynomials of planar polynomial vector fields. *Mathematics of Computation* ., USA, v.85, n.299, p. 1393-1425, May.2016. <https://doi.org/10.1090/mcom/3007>
- 31 FERRAGUT, A; GASULL, A. Seeking Darboux Polynomials. *Acta Applicandae Mathematicae.*, [S.l.], v. 139, p. 167–186, 2015. <https://doi.org/10.1007/s10440-014-9974-0>
- 32 FERRAGUT, A; GALINDO, C; MONSERAT, F. On the computation of Darboux first integrals of a class of planar polynomial vector fields. *Journal of Mathematical Analysis and Applications.*, USA, v. 478, p. 743-763, Oct.2019. <https://doi.org/10.1016/j.jmaa.2019.05.052>
- 33 PEREIRA, J. V. Vector Fields, Invariant Varieties and Linear Systems. *Ann. Inst. Fourier.*, Grenoble, v. 51, n. 5, p. 1385-1405, 2001 <https://doi.org/10.5802/aif.1858>
- 34 LLIBRE, J; ZHANG, X. Darboux theory of integrability in C^n taking into account the multiplicity. *Journal of Differential Equations.*, Holanda, v. 246, p. 541–551, Jan. 2009.

- 35 LLIBRE, J; ZHANG, X. Darboux theory of integrability for polynomial vector fields in \mathbb{R}^n taking into account the multiplicity at infinity. *Bull. Sci. Math.*, [S.1], v. 133, p. 765–78, Oct. 2009.
- 36 BERRONE, L. R; GIACOMINI, H. Inverse Jacobi multipliers. *Rend. Circ. Mat. Palermo.*, Italy, v. 52, p.77–130, Feb.2003.
- 37 MAN, Y. -K. Computing Closed Form Solutions of First Order ODEs Using the Prelle-Singer Procedure. *Journal of Symbolic Computation.*, Holanda, v. 16, p. 423-443, Nov. 1993.
- 38 SHTOKHAMER, R; GLINOS, N; CAVINESS B. F. Computing Elementary First Integrals of Differential Equations. *Computers and Mathematics Conference Manuscript.*, Stanford, 1986.
- 39 BUCHBERGER, B. *An Algorithm for Finding the Basis Elements of the Residue Class Ring of a Zero Dimensional Polynomial Ideal*. Thesis (Ph.D) - University of Innsbruck, 1965. English translation by Michael Abramson in *Journal of Symbolic Computation.*, [S.1], v. 41, p. 471-511, 2006.
- 40 KAMKE, E. *Differentialgleichungen: Lösungsmethoden und Lösungen*. New York: Chelsea Publishing Co, 1959.