**Universidade do Estado do Rio de Janeiro**
Centro de Tecnologia e Ciências
Instituto de Física Armando Dias Tavares

Andrea Barroso Melo Monteiro de Queiroz

# Um novo método para buscar soluções liouvillianas de equações diferenciais ordinárias de primeira ordem que apresentam funções elementares

Rio de Janeiro
2020

Andrea Barroso Melo Monteiro de Queiroz

# Um novo método para buscar soluções liouvillianas de equações diferenciais ordinárias de primeira ordem que apresentam funções elementares

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

Orientador: Prof. Dr. Luis Antônio Campinho Pereira da Mota
Coorientador: Prof. Dr. Luiz Guilherme Silva Duarte

Rio de Janeiro
2020

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

Q3 Queiroz, Andrea Barroso Melo Monteiro de.
Um novo método para buscar soluções liouvillianas de equações diferencias ordinárias de primeira ordem que apresentam funções elementares / Andrea Barroso Melo Monteiro de Queiroz - 2020
94 f.

Orientador: Luis Antônio Campinho da Mota.
Coorientador: Luiz Guilherme Silva Duarte.
Tese (doutorado) - Universidade do estado do Rio de Janeiro, Instituto de Física Armando Dias Tavares.

1.Equações diferencias ordinárias - Teses. 2. Sistemas dinâmicos diferenciais - Teses. 3. Integrais (Matemática) - Teses. 4. Física matemática - Teses. I. Mota, Luis 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 517.91

Bibliotecária: Denise da Silva Gayer CRB7/5069



_____ _____

Assinatura Data

Andrea Barroso Melo Monteiro de Queiroz

**Um novo método para buscar soluções liouvillianas de equações diferenciais ordinárias de primeira ordem que apresentam funções elementares**

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

Aprovada em 16 de dezembro de 2020
Banca Examinadora:

---

Prof. Dr. Luis Antonio 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. James Ewan Faskin Skea
Instituto de Física Armando Dias Tavares - UERJ

---

Prof. Dr. Jair Koiller
Universidade Federal de Juiz de Fora

---

Prof. Dr. Alejandro Cabrera
Universidade Federal do Rio de Janeiro

---

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

---

Prof. Dr. Rafael de Sousa Dutra
Instituto Federal de Educação, Ciência e Tecnologia do Rio de Janeiro

Rio de Janeiro
2020

## DEDICATÓRIA

Aos meus pais, Wanda e Antônio, ao meu esposo, André Felipe, a nossa filha, Maria Fernanda e a minha irmã Rosana.

## AGRADECIMENTOS

Ao Inmetro pela oportunidade e incentivo.

À UERJ pela acolhida e materialização do desenvolvimento de meu trabalho.

Aos meus orientadores, grandes mestres, pela nobre missão de me guiar com segurança, sabedoria e muito entusiasmo.

Aos familiares e amigos de longa jornada por todo o apoio.

## RESUMO

QUEIROZ, A. B. M. M. *Um novo método para buscar soluções liouvillianas de equações diferenciais ordinárias de primeira ordem que apresentam funções elementares*. 2020. 94f. Tese (Doutorado em Física) - Instituto de Física Armando Dias Tavares, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2020.

Nesta tese apresentamos uma nova abordagem para lidar com equações diferenciais ordinárias de primeira ordem (1ODEs) que apresentam funções elementares. O método que desenvolvemos é uma alternativa para o procedimento que foi apresentado em (1, 2). Em (3), foi estabelecida uma base teórica para lidar com equações diferenciais ordinárias racionais de segunda ordem (2ODEs racionais) – o método da função $S$ – que apresentavam (pelo menos) uma integral primeira Liouvilliana. Nesta tese, combinamos a técnica utilizada em (3) com uma ideia análoga àquela usada em (1, 2) e produzimos um método que se provou muito eficiente e bastante abrangente em um grande número de casos nos quais os métodos generalistas tradicionais mais poderosos (os métodos de Lie e Darboux) apresentam dificuldades. O procedimento que apresentamos aqui para resolver 1ODEs está colocado em uma base matemática sólida e dois dos algoritmos principais foram implementados em um pacote computacional (*LeapS1ode*) em Maple. O pacote *LeapS1ode* inclui comandos que permitem a obtenção de todas as etapas intermediárias do processo de resolução de 1ODEs e possui um desenho que o torna muito útil para a pesquisa (tanto em física como em matemática) e especialmente eficaz na busca de regiões de integrabilidade para 1ODEs que apresentam parâmetros.



Palavras-chave: Equações diferenciais ordinárias de primeira ordem. Método da função $S$. Integrais primeiras liouvillianas. Sistemas dinâmicos.

## ABSTRACT

QUEIROZ, A. B. M. M. *A new method to search for Liouvillian solutions of first order ordinary differential equations that present elementary functions*. 2020. 94f. Tese (Doutorado em Física) - Instituto de Física Armando Dias Tavares, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2020.

In this thesis we present a new approach to deal with first order ordinary differential equations (1ODEs) presenting elementary functions. The method we developed is an alternative to the procedure that was presented in (1, 2). In (3), a theoretical basis was established to deal with second order rational ordinary differential equations (rational 2ODEs) - the $S$-function method - that presents (at least) a Liouvillian first integral. Here, we combine the technique used in (3) with an idea analogous to that used in (1, 2) in order to produce a method that proved to be very efficient in a large number of cases in which the most powerful traditional generalist methods (the Lie and Darboux methods) present difficulties. The procedure we present here for solving 1ODEs is placed on a solid mathematical basis and two of the main algorithms were implemented in a computational package (*LeapS1ode*) in Maple. The *LeapS1ode* package includes commands that allow obtaining all the intermediate steps of the 1ODEs resolution process and has a design that makes it very useful for research (both in physics and mathematics) and especially effective in finding regions of integrability for 1ODEs that presenting parameters.



Keywords: Ordinary differential equations of first order. $S$-function method. Liouvillian first integrals. Dynamical systems.

## SUMÁRIO

# INTRODUÇÃO

Como as equações diferenciais (EDs) se constituem nos modelos principais para sistemas físicos (e também para várias outras ciências como a química, a biologia, a engenharia, a economia etc), grande parte do estudo comportamental desses sistemas está diretamente ligado ao entendimento das EDs que o descrevem. Dessa forma, desde o advento das teorias científicas modernas, passou a ser de extrema importância a busca por soluções / invariantes / características / simetrias / etc dessas EDs. Os sistemas de EDs fomentaram, portanto, uma busca ferrenha por métodos / procedimentos / abordagens / algoritmos que pudessem fornecer as soluções / propriedades / etc de tais sistemas.

Os primeiros métodos eram de natureza mais 'classificatória', ou seja, cada novo procedimento para resolver um dado tipo de ED era catalogado e 'somado' aos métodos já existentes. Ao longo dos séculos XVIII e XIX foi reunido um grande conjunto de métodos, cada um dos quais especialmente desenhado para lidar com um tipo específico de ED.

Em fins do século XIX, as teorias de Lie e Darboux (4, 5) trouxeram uma 'sabor' mais generalista (no sentido de que eles não visam EDs de um tipo definido por seu método de solução) ao estudo/determinação de integrais primeiras e soluções de EDs. O método de Lie é baseado no conceito de *simetria* de uma ED[1] e a abordagem de Darboux-Prelle-Singer é centrada no conceito de *polinômio de Darboux* (PD)[2]. Devido à sua grande abrangência e potencial de extensão, essas duas abordagens têm sido extensivamente estudadas/desenvolvidas/estendidas ao longo do último século (em especial nas últimas décadas em virtude do desenvolvimento e popularização da computação simbólica). É neste contexto que se localiza a maior parte do trabalho desenvolvido pelo nosso grupo: Desenvolvimento (e implementação computacional) de algoritmos e métodos matemáticos (generalistas) para a busca de soluções / integrais primeiras de EDs[3].

Neste trabalho desenvolvemos um novo método para buscar integrais primeiras de sistemas dinâmicos (com funções) no plano ou, equivalentemente, para buscar soluções gerais de equações diferenciais ordinárias de primeira ordem (1ODEs) que apresentam funções. Nas seções seguintes desta introdução vou descrever (de forma breve) onde e como o trabalho aqui apresentado se encaixa no vasto campo mencionado acima.

---

[1] Uma simetria de uma ED é um grupo de transformações que mantém a forma da ED. Para um conhecimento mais profundo do método de Lie veja, por exemplo, (8, 46, 47, 48, 49).

[2] Um PD é um polinômio que é uma autofunção do operador de Darboux associado com a ED em questão. A ideia base pode ser entendida consultando (28).

[3] Nosso grupo de pesquisa tem se dedicado a estas duas vertentes (os métodos de Lie e Darboux) e seus desdobramentos.

**As abordagens de Lie e Darboux-Prelle-Singer e seu problemas principais**

Como já mencionado acima, os métodos generalistas analíticos mais poderosos e abrangentes para a busca de integrais primeiras (ou soluções) Liouvillianas de EDs eram (e ainda são) o método de simetrias de Lie e as abordagens Darbouxianas. Contudo, embora muito eficientes e amplamente aplicáveis, essas abordagens tinham seus 'calcanhares de Aquiles': o método de Lie pode ser aplicado a equações diferenciais ordinárias (EDOs) de qualquer ordem, a sistemas de EDOs, a equações diferenciais parciais (EDPs), a sistemas de EDPs etc, contudo ele não fornece um algoritmo para calcular as simetrias[4]. E, com frequência, podemos nos deparar com uma EDP (para as simetrias) que é muito mais complicada de resolver do que a EDO em questão. Por outro lado, os métodos Darbouxianos foram (a princípio) aplicados apenas a sistemas polinomiais de EDOs de primeira ordem (1EDOs). Além disso, quando o grau dos polinômios de Darboux (PDs) que fazem parte do fator integrante são relativamente altos[5] o método encontra dificuldades. Para superar esses problemas, várias abordagens foram desenvolvidas:

- Para lidar com EDOs usando métodos de simetria L.G.S. Duarte e L.A.C.P. da Mota (6, 7) criou uma série de heurísticas para encontrar simetrias de EDOs de primeira e segunda ordem; P. J. Olver introduziu o conceito de campo vetorial exponencial (ver (8), p. 185); B. Abraham-Shrauner, A. Guo, K.S. Govinder, P.G.L. Leach, F.M. Mahomed, A.A. Adam, M. L. Gandarias, M. S. Bruzón, M. Senthilvelan e outros trabalharam com o conceito de simetrias ocultas e não locais (9, 10, 11, 12, 13, 14, 15, 16). C. Muriel e J.L. Romero desenvolveram o conceito de $\lambda$ -simetria (17, 18) que inaugura uma rica área de pesquisa (ver (19, 20, 21, 22) e suas referências). E. Pucci e G. Saccomandi criaram o conceito de simetria telescópica (23). Outra grande abordagem foi trazida por M.C. Nucci que usou o conceito de último multiplicador de Jacobi (ver (24, 25)) de uma forma muito inteligente.

- No contexto Darbouxiano, Cairó e Llibre (26) apontaram que usando os fatores exponenciais, introduzidos por Cristopher (27), os métodos de Darboux poderiam ser generalizados para lidar com elementares (ao invés de apenas algébricos) primeiro integrais. Em 1983, Prelle e Singer (28) encontraram uma abordagem semialgorítmica para encontrar primeiras integrais elementares de campos vetoriais 2D (ou, equivalentemente, 1EDOs racionais). Por causa de suas características notáveis, a abor-

[4] Especialmente quando as EDOs não apresentam simetrias de ponto. Neste caso não podemos separar a equação determinante (a EDP que as simetrias devem obedecer) nas podtências das derivadas da variável dependente para obter um sistema sobredeterminado de EDPs.

[5] Para 1EDOs racionais o grau 4 já costuma ser inviável se usarmos o *método de coeficientes indeterminados* (MCI); para EDOs racionais de segunda ordem (2EDOs racionais) o grau 2 já é problemático.

dagem Darbouxiana gerou muitas extensões (1, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41). Em particular, em (32, 34, 35, 36, 38) o método foi estendido para lidar com 1EDOs racionais apresentando soluções Liouvillianas. Em (37, 39, 40), ele foi estendido para lidar com 2EDOs racionais que apresentam pelo menos uma primeira integral elementar. Em (41, 42) são tratados sistemas polinomiais de 1EDOs em mais de duas variáveis.

Contudo, as EDOs que apresentam fatores integrantes formados por PDs de grau relativamente alto (o que significa que, na prática, é difícil determiná-los com a memória de um computador pessoal padrão) e, além disso, não apresentam simetrias facilmente determináveis (ou seja, possuem simetrias muito complicadas ou não possuem simetrias de ponto) continuam sendo desafiadoras para os métodos de resolução.

**Nosso Grupo de Pesquisa**

O nosso grupo trabalha basicamente com a criação de métodos matemáticos e algoritmos para lidar com equações diferenciais (EDs) e sistemas de EDs (sistemas dinâmicos – SDs) tanto analítica quanto numericamente. Grande parte desse trabalho consiste na busca de soluções / integrais primeiras de 1EDOs e 2EDOs precisamente tentando contribuir para solucionar o tipo de EDO descrito no último parágrafo da seção anterior. Nosso viés para abordar o problema é generalista (não classificatório) e nos aventuramos na abordagem Darboux-Prelle-Singer (1, 2, 37, 34, 35, 36, 38, 39, 40, 41) e no método de Lie desenvolvendo métodos heurísticos em (6, 7). Também desenvolvemos um método que pode ser visto como uma abordagem mista (Lie & Darboux): O método da função-$S$ (3, 43). Uma outra característica de nosso grupo é, além de desenvolver a base teórica e os métodos / algoritmos, implementar tais processos em pacotes computacionais (1, 2, 3, 6, 7, 38, 43, 44).

Em sua maioria, lidamos com EDOs racionais, ou seja, na forma geral (para 1EDOs e 2EDOs):

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

onde $M$ e $N$ são polinômios. Fizemos, no passado, também algumas incursões em EDOs contendo funções elementares (1, 2), sendo estas extensões todas heurísticas. Atualmente, estamos no processo de produzir resultados para dar uma base teórica mais sólida para a busca de EDOs contendo funções elementares:

$$y' = \frac{dy}{dx} = \frac{M(x,y,\theta_1,\cdots,\theta_r)}{N(x,y,\theta_1,\cdots,\theta_r)}, \tag{2}$$

$$y'' = \frac{dy'}{dx} = \phi(x, y, y', \theta_1, \cdots, \theta_r) = \frac{M(x, y, y'y, \theta_1, \cdots, \theta_r)}{N(x, y, y', \theta_1, \cdots, \theta_r)}, \tag{3}$$

onde $M$ e $N$ são polinômios e $\theta_1, \cdots, \theta_r$ são geradores elementares[6].

Meu trabalho se insere neste contexto, no que concerne à resolução de EDOs que apresentam funções elementares (desta feita, embasados em resultados teóricos sólidos, ou seja, trocando procedimentos heurísticos por semi algoritmos). Nosso grupo iniciou esse caminho com o trabalho (43) que lida com 2EDOs dos tipos

$$y'' = \frac{M(y, y', \theta(x))}{N(y, y', \theta(x))}, \tag{4}$$

$$y'' = \frac{M(x, y', \theta(y))}{N(x, y', \theta(y))}, \tag{5}$$

onde $M$ e $N$ são polinômios e $\theta(x)$ e $\theta(y)$ são geradores elementares tais que a integral primeira $I_1$ da 2EDO (4) é tal que $I_1 \in \mathbb{C}(y, y', \theta(x))$ e a integral primeira $I_2$ da 2EDO (5) é tal que $I_2 \in \mathbb{C}(x, y', \theta(y))$[7].

Nesta tese vou tratar de 1EDOs que apresentam funções elementares. Na seção seguinte vou descrever de maneira mais formal e precisa (e, espero, mais clara) o tipo de problema que abordaremos aqui.

**Este Trabalho**

Considere uma 1EDO que pode ser escrita na seguinte forma:

$$y' = \frac{dy}{dx} = \phi(x, y, \theta) = \frac{M(x, y, \theta)}{N(x, y, \theta)} \tag{6}$$

onde $\phi$ é uma função racional de $(x, y, \theta)$, $\theta$ é uma função elementar de $(x, y)$ e $M$ e $N$ são polinômios coprimos em $(x, y, \theta)$. De maneira mais formal, $\phi \in K = \mathbb{C}(x, y, \theta)$, ou seja, $M$, $N \in \mathbb{C}[x, y, \theta]$, onde $\theta$ é um gerador elementar sobre $\mathbb{C}(x, y)$[8]. Considere yambém que

[6] Veja (45).

[7] $\mathbb{C}(y, y', \theta(x))$ é o corpo das funções racionais nas variáveis $(y, y', \theta(x))$ e $\mathbb{C}(x, y', \theta(y))$ é o corpo das funções racionais nas variáveis $(x, y', \theta(y))$. Veja (45).

[8] $K = \mathbb{C}(x, y, \theta)$ é o corpo diferencial das funções racionais nas variáveis $(x, y, \theta)$, $\mathbb{C}[x, y, \theta]$ é o anel diferencial das funções polinomiais nas variáveis $(x, y, \theta)$ e $\mathbb{C}(x, y)$ é o corpo diferencial das funções racionais nas variáveis $(x, y)$.

$I(x, y, \theta) = C$ (onde $C$ é uma constante) representa uma solução geral da 1ODE (6) e $I$ é uma função Liouvilliana de $(x, y, \theta)$[9]. Considere, além disso, que as derivadas parciais de $I$ em relação a $x$, $y$ e $\theta$ possam ser expressas como

$$I_x = R\,P_1 \tag{7}$$

$$I_y = R\,P_2 \tag{8}$$

$$I_\theta = R\,P_3 \tag{9}$$

onde $R$ é uma função elementar de $(x, y, \theta)$ e $P_1,\ P_2,\ P_3 \in \mathbb{C}[x, y, \theta]$[10].

Neste trabalho vou me propor a resolver a seguinte questão:

**Questão 0.1** *É possível determinar se uma 1EDO da forma (6) possui uma solução geral $I(x, y, \theta) = C$, onde $I$ pertence a uma extensão Liouvilliana $L$ de $\mathbb{C}(x, y, \theta)$? E, em caso positivo, é possível computarmos a função $I(x, y, \theta)$?*

O método que foi empregado para tentar responder à questão 0.1 tem como ideia central associar a 1EDO (6) com um campo vetorial polinomial em três variáveis que possui $I(x, y, z)$ como integral primeira. Essa ideia tem certas analogias com o procedimento que usamos em (1) e com o que foi empregado em (43). No nosso caso em particular, o ponto principal em fazermos essa associação é a possibilidade de usarmos teoremas já bem estabelecidos para campos vetoriais polinomiais (veja (41)) e também de trabalharmos em uma 'área mais conhecida'. Uma segunda vantagem é a possibilidade de associarmos o campo vetorial a uma 2EDO racional que também tenha $I(x, y, z)$ como integral primeira pois, deste modo, além das vantagens já mencionadas acima, temos a possibilidade de aplicar o método da função-$S$ (que desenhamos para encontrar integrais primeiras Liouvillianas de 2EDOs racionais) para o qual inclusive já dispomos de um pacote computacional em Maple desenvolvido pelo nosso grupo (veja (3)). A princípio, pode parecer um pouco estranho trocarmos a tentativa de resolver uma 1EDO pela busca de uma integral primeira de uma 2EDO, quero dizer, a primeira vista não parece uma troca muito vantajosa. A vantagem, como veremos no decorrer desta tese, é baseada no fato que o método da função-$S$ é especialmente eficiente em casos nos quais os métodos de Lie e Darboux encontram mais dificuldade[11] e, por outro lado, a desvantagem aparente não

[9] Isto é, $I \in L$, onde $L$ é uma extensão Liouvilliana de $\mathbb{C}(x, y, \theta)$.

[10] É um modo formal de dizer que $P_1$, $P_2$ e $P_3$ são polinômios em $(x, y, \theta)$.

[11] A saber: quando a 1EDO apresenta simetrias muito complicadas e, além disso, os polinômios de Darboux presentes no fator integrante possuem um grau que inviabiliza o procedimento MCI (método dos coeficientes indeterminados).

é tão expressiva uma vez que estamos trocando uma 1EDO que apresenta uma função elementar por uma 2EDO racional.

Esta tese foi organizada da seguinte forma:

- O capítulo 1 consiste nesta Introdução.
- No capítulo 2, apresentamos (de forma breve) os fundamentos da abordagem de Darboux-Prelle-Singer (O método de Prelle-Singer e seus desdobramentos) e do método da função-$S$ (que dispensa o uso dos PDs):
  1. Na primeira seção apresentamos os principais conceitos e procedimentos envolvidos na abordagem de Darboux-Prelle-Singer.
  2. Na segunda seção, apresentamos o método da função-$S$ para encontrar integrais primeiras Liouvillianas de 2EDOs racionais.
     - Na primeira subseção, apresentamos alguns resultados básicos e definimos *Função-S*.
     - Na segunda subseção, apresentamos o conceito de *1EDO Associada* e, em seguida, o método da função-$S$.
- No capítulo 3 tem início a descrição do trabalho original apresentado nesta tese, isto é, uma nova maneira de resolver 1EDOs com funções elementares que possuem solução geral Liouviliana:
  1. Na primeira seção apresentamos a ideia base que motivou este trabalho: Uma possível relação de equivalência entre 1EDOs com uma função elementar e campos vetoriais polinomiais em três variáveis.
  2. Na segunda seção, mostramos uma conexão entre campos vetoriais polinomiais em três variáveis e 2EDOs racionais. A ideia aqui é nos aproveitarmos do método da função-$S$ para encontrar a integral primeira sem a necessidade de calcularmos os PDs.
  3. Na terceira seção, construímos um método que nos permite responder, em parte, à questão 0.1. Aplicamos o método para três exemplos em um primeiro teste para determinar sua eficácia.
  4. Na quarta seção apresentamos uma possível melhoria do método baseados em dois resultados teóricos. Podemos definir, no caso do método aprimorado um grande conjunto de 1EDOs para os quais podemos responder à questão 0.1.
  5. Na quinta seção, apresentamos dois possíveis algoritmos $Lsolv$ e $FastL_s$ (baseados, respectivamente, nos métodos desenvolvidos nas seções anteriores) para calcular a integral primeira Liouvilliana do campo vetorial associado à 1EDO (6).

- No capítulo 4 apresentamos uma implementação computacional dos algoritmos desenvolvidos. Criamos o pacote *LeapS*1*ode – Liouvillian, Elementary, Algebraic, Polynomial Solutions of 1ODEs.*

    1. Na primeira seção, mostramos um resumo dos comandos que compõe o pacote *LeapS*1*ode* e sua funcionalidade e, logo em sequência, fazemos uma descrição detalhada de cada um dos comandos que compõe o pacote.
    2. Na segunda seção, apresentamos exemplos práticos do uso dos comandos do pacote aplicados a exemplos específicos.

- No capítulo 5, apresentamos uma análise do desempenho dos algoritmos *Lsolv* e $FastL_S$ bem como do desempenho do pacote *LeapS*1*ode*:

    1. Na primeira seção, começamos a análise do pacote *LeapS*1*ode* construíndo um pequeno conjunto amostral de 1EDOs que não são resolvidas usando os métodos tradicionais (bem como alguns não muito tradicionais) implementados no sistema Maple de computação algébrica (Maple CAS). Esse conjunto é apenas uma amostra de uma classe muito abrangente de 1EDOs que é bastante 'opaca' à maioria dos métodos.
    2. Na segunda seção, mostramos um uso mais avançado do pacote *LeapS*1*ode*:
        - Na primeira subseção, mostramos o uso dos comandos do pacote em casos mais complicados.
        - Na segunda subseção, mostramos como podemos usar uma característica especial do pacote para analisar a região de integrabilidade em 1EDOs que apresentam parâmetros.
    3. Na quarta seção, fazemos uma comparação do desempenho dos algoritmos *Lsolv* e $FastL_S$ implementados no pacote *LeapS*1*ode* e, além disso, uma comparação do desempenho do pacote (que usa uma adaptação da técnica desenvolvida em (3) – um método da função-$S$ modificado) com uma abordagem DPS padrão (isto é, que determina os polinômios de Darboux e os utiliza para construir um fator integrante).

- No capítulo 6, vamos apresentar um resultado que pode ser visto como um aperfeiçoamento do algoritmo $FastL_S$. Este novo procedimento está em um capítulo separado (ou seja, 'distante' da parte que fala dos métodos e algoritmos – capítulo 3) pois ele se 'materializou' após o fechamento estrutural desta tese. Além disso, a ideia que se encontra por trás da melhoria inaugura toda uma nova área de pesquisa que se estende além dos limites deste trabalho e, desse modo, achei mais conveniente apresentá-la em um capítulo à parte.

  - Na primeira seção, mostramos um exemplo no qual o cálculo da função-$S$ associada ao campo vetorial polinomial pode ser obtido de maneira linear.
  - Em seguida, mostramos que podemos estender este processo para uma classe muito abrangente de 1EDOs com funções: a classe $L_S$ de 1EDOs que podem ser resolvidas pelo algoritmo $FastL_S$.
  - Na terceira seção, fazemos uma breve análise desta melhoria.

- Por fim, no capítulo 7, apresentamos a nossa conclusão: fazemos algumas considerações sobre o trabalho desenvolvido, nossa avaliação sobre os resultados obtidos e possíveis direções a seguir.

# 1 A ABORDAGEM DE DARBOUX-PRELLE-SINGER E O MÉTODO DA FUNÇÃO-$S$

Neste capítulo definiremos a base matemática (ou seja, os conceitos e resultados) necessários para a construção do nosso método. Na primeira seção apresentaremos (brevemente) a abordagem de Darboux-Prelle-Singer e, em seguida, os fundamentos do método da função-$S$.

## 1.1 A abordagem de Darboux-Prelle-Singer

Se quisermos ser sucintos ao explicar a abordagem de Darboux-Prelle-Singer (DPS), na busca de integrais primeiras Liouvillianas para sistemas autônomos de 1ODEs polinomiais no plano, podemos resumir a ideia central em um parágrafo:

*Encontre os polinômios de Darboux (PDs) associados ao sistema de 1EDOs e use-os para determinar um fator integrante.*

Sem nenhum exagero, a determinação dos PDs é a parte mais importante e, podemos dizer, de longe a mais complexa de todo o processo. Assim, com algumas poucas definições e resultados, podemos descrever de maneira compreensível a abordagem DPS quando aplicada a campos vetoriais no plano. Para começar, vamos definir mais formalmente os conceitos aos quais nos referimos. Considere o seguinte sistema de equações diferenciais ordinárias de primeira ordem (sistema de 1EDOs) polinomiais no plano:

$$\begin{cases} \dot{x} = f(x,y) \\ \dot{y} = g(x,y), \end{cases} \tag{10}$$

onde $f$ e $g$ são polinômios coprimos em $(x,y)$ e o ponto sobre as variáveis significa derivada com relação a um parâmetro $t(\dot{u} \equiv du/dt)$.

**Definição 1.1** *A função $I(x,y)$ é chamada uma* **integral primeira** *do sistema (10) se $I$ é uma constante sobre as soluções de (10).*

**Definição 1.2** *O campo vetorial associado com o sistema (10), dado por*

$$X \equiv f(x,y)\frac{\partial}{\partial x} + g(x,y)\frac{\partial}{\partial y}, \tag{11}$$

*é chamado* **operador de Darboux** *associado a ao sistema (10).*

**Observação 1.1** *Se $I(x,y)$ é uma integral primeira do sistema (10), então $X(I) = 0$.*

**Definição 1.3** *O polinômio $p(x,y) \in \mathbb{C}[x,y]$ é chamado* ***polinômio de Darboux*** *do campo vetorial $X$ se $X(p) = q\,p$, onde $q$ é um polinômio denominado* ***cofator****.*

**Teorema 1.1** *(Prelle-Singer). Se o sistema (10) apresenta uma integral primeira elementar, então existe um fator integrante $R$ para o sistema (10) da forma $\prod_i p_i^{n_i}$, onde $p_i$ são os polinômios de Darboux do sistema (10) e $n_i$ são números racionais.*

*Prova.* Para a prova deste teorema veja (28).

Como $\dfrac{X(R)}{R} = -\nabla(X) = -(f_x + g_y)$, substituindo $R = \prod_i p_i^{n_i}$ obtemos (veja (1))

$$\sum_i n_i \frac{X(p_i)}{p_i} = \sum_i n_i q_i = -(f_x + g_y), \tag{12}$$

onde os $q_i$ são cofatores dos $p_i$. Portanto, um método possível para encontar uma integral primeira elementar é:

**Método Prelle-Singer:** (esboço)

- Determine os polinômios de Darboux $p_i$ (e seus respectivos cofatores $q_i$) associados ao sistema.
- Encontre os números $n_i$ que satisfazem à equação $\sum_i n_i q_i = -(f_x + g_y)$.
- Construa o fator integrante $R = \prod_i p_i^{n_i}$ e encontre uma integral primeira $I(x,y)$ por quadraturas.

**Observação 1.2** *Se o sistema (10) apresenta uma integral primeira Liouvilliana não elementar, o método Prelle-Singer (PS) necessita de algumas modificações (veja o método Christopher-Singer em (36, 38)). Contudo, a principal premissa se mantém a mesma: determinar os PDs do sistema.*

**Observação 1.3** *Para campos vetorias em $\mathbb{R}^3$, a situação fica um pouco mais complicada: em primeiro lugar, o fator integrante $R$ não é um multiplicador de Jacobi (como acontece em $\mathbb{R}^2$). Assim, $X(R)/R$ não é um polinômio conhecido e depende de uma função racional não determinada a priori. Em segundo, necessitamos determinar os polinômios de Draboux (PDs) em três variáveis, o que é uma tarefa muito difícil, uma vez que já não é fácil determiná-los em duas variáveis*[12].

[12] Aqui, as palavras *fácil* e *difícil* se referem ao fato que a partir do grau 4 é muito custoso computacionalmente (em geral, inviável) determinar PDs em duas variáveis. No caso de PDs em em três variáveis, o grau 2 já começa a ficar inviável computacionalmente.

## 1.2 O método da função-$S$

O conceito da função-$S$ surgiu quando tentamos adaptar a abordagem DPS para equações diferenciais ordinárias racionais de segunda ordem (2EDOs racionais). A ideia era "adicionar" uma 1-forma nula à 1-forma que representava a 2EDO (veja (37, 40)). No entanto, o problema de cálculo dos PDs era tão custoso, em termos computacionais, que desenvolvemos um procedimento misto que foi chamado de "método da função-$S$". Muito resumidamente, o método da função-$S$ não utiliza o cálculo de polinômios de Darboux mas, em troca, depende da resolução de duas 1EDOs. A boa notícia é que o método da função-$S$ se mostrou mais eficaz, justamente em problemas nos quais os PDs tinham um grau relativamente alto, sendo uma ótima alternativa a ser utilizada nestes casos. Nesta seção veremos em que consiste o método e como ele opera.

### 1.2.1 Algumas definições e resultados básicos

Considere a 2EDO racional dada por

$$z' = \frac{dz}{dx} = \phi(x,y,z) = \frac{M(x,y,z)}{N(x,y,z)}, (z \equiv y'), \tag{13}$$

onde $M$ e $N$ são polinômios coprimos em $\mathbb{C}[x,y,z]$.

**Definição 1.4** *Uma função I(x,y,z) é chamada* **integral primeira** *da 2EDO (13) se I é constante sobre as soluções de (13).*

**Observação 1.4** *Se $I(x,y,z)$ é uma integral primeira da 2EDO (13) então, sobre as curvas-solução de (13), a 1- forma exata $\omega = dI = I_x dx + I_y dy + I_z dz$ é nula.*

Sobre as curvas-solução de (13), temos duas 1-formas não nulas:

$$\alpha = \phi\, dx - dz, \tag{14}$$

$$\beta = z\, dx - dy. \tag{15}$$

Então, a 1-forma $\omega$ é um vetor no espaço determinado pelas 1-formas $\alpha$ e $\beta$, i.e., podemos escrever $\omega = r_1(x,y,z)\,\alpha + r_2(x,y,z)\,\beta$:

$$\begin{aligned} dI &= I_x\, dx + I_y\, dy + I_z\, dz = r_1\,(\phi dx - dz) + r_2\,(z dx - dy) \\ &= (r_1\,\phi + r_2\, z) + (-r_2\,) dy + (-r_1)\, dz. \end{aligned} \tag{16}$$

Definindo $R \equiv r_1$ e $S \equiv \frac{r_2}{r_1}$, podemos escrever

$$dI \equiv R[(\phi + z\,S)\,dx - S\,dy - dz]. \tag{17}$$

Portanto, temos que

$$\begin{aligned} I_x &= R(\phi + z\,S), \\ I_y &= -R\,S, \\ I_z &= -R. \end{aligned}$$

**Definição 1.5** *Seja $\gamma$ um 1-forma. Dizemos que $R$ é um* ***fator integrante*** *para a 1-forma $\gamma$ se $R\,\gamma$ é uma 1-forma exata.*

**Definição 1.6** *Seja $I$ uma integral primeira da 2EDO (13). A função definida por $S \equiv I_y/I_z$ é chamada uma* ***função-S*** *associada à 2EDO através da integral primeira $I$.*

**Observação 1.5** *Das definições acima e, tendo em vista a definição (17) podemos ver que $R$ é um fator integrante para a 1-forma $(\phi + z\,S)\,dx - S\,dy - dz$ e $S$ é a função-S associada à 2EDO (13) através de $I$.*

**Teorema 1.2** *Seja $I(x, y, z)$ uma integral primeira da 2EDO (13). Se $S$ e $R$ são como em (17), então podemos escrever*

$$\begin{aligned} D_x(R) + R(S + \phi_x) &= 0, & (18)\\ SD_x(R) + R(D_X(S) + \phi_x) &= 0, & (19)\\ -(R_z S + RS_z) + R_y &= 0, & (20) \end{aligned}$$

*onde $D_x \equiv \partial_x + z\partial_y + \phi(x, y, z)\partial_z$.*

Para prova veja (40).

**Corolário 1.1** *Seja $S$ uma função-S associada à 2EDO $z' = \phi(x, y, z)$. Então $S$ obedece a seguinte equação:*

$$D_x(S) = S^2 + \phi_z S - \phi_y. \tag{21}$$

Para prova veja (3).

### 1.2.2 As 1EDOs associadas

De (21) vemos que a função-$S$ associada com a 2EDO racional (13) satisfaz a uma equação diferencial parcial de ordem um (1EDP) quase linear nas variáveis $(x, y, z)$. Sobre

as soluções da 2EDO (13) temos que $y = y(x)$ e $z = z(x)$ e, portanto, o operador $D_x$ é, formalmente, $d_x$ [13]. Então, formalmente, sobre as soluções da 2EDO (13) podemos escrever a 1PDE (21) como uma 1EDO de Riccati:

$$\frac{ds}{dx} = s^2 + \phi_z(x)\, s - \phi_y(x). \tag{22}$$

É de conhecimento comum que a transformação

$$y(x) = -\frac{r'(x)}{f(x)r(x)} \tag{23}$$

transforma a equação de Riccati

$$y'(x) = f(x)\, y(x)^2 + g(x)\, y(x) + h(x) \tag{24}$$

em uma 2EDO linear

$$r''(x) = \frac{(f'(x) + g(x)\, f(x))\, r'(x)}{f(x)} - f(x)\, h(x)\, r(x). \tag{25}$$

Então, com a transformação

$$s(x) = -\frac{w'(x)}{w(x)}, \tag{26}$$

a 1EDO de Riccatti para $s(x)$ (22), sobre as soluções da 2EDO (13) transforma-se na 2EDO linear homogênea

$$w''(x) = \phi_z(x)\, w'(x) + \phi_y\, w(x). \tag{27}$$

Portanto, podemos usar a equivalência formal $D_x \sim d_x$ para produzir uma conexão entre a função-$S$ e as simetrias da 2EDO, escritas numa forma particular. Aplicando a transformação

$$S = -\frac{D_x(\nu)}{\nu} \tag{28}$$

na equação (21), obtemos

$$D_x^2(\nu) = \phi_z\, D_x(\nu) + \phi_y\, \nu. \tag{29}$$

---

[13] $d_x \equiv \dfrac{d}{dx}$.

A equação (29) é justamente a condição para que $\nu$ seja um infinitésimo que define o gerador de simetria na forma evolucionária. Então, temos

**Teorema 1.3** *Seja $\nu$ uma função de (x,y,z) tal que $[0, \nu]$ define uma simetria da 2EDO (13) na forma evolucionária, i.e., $X_e \equiv \nu\,\partial_y$ gera uma transformação de simetria para (13). Então a função definida por $S \equiv -D_x(\nu)/\nu$ é uma função-S associada à 2EDO (13).*

*Prova.* Veja em (3)

**Corolário 1.2** *Seja S uma função S associada a 2EDO (13). Então a função $\nu$ dada por*

$$\nu \equiv e^{\int_x(-S)}, \tag{30}$$

*onde $\int_x$ é o operador inverso de $D_x$, i.e., $\int_x D_x = D_x \int_x = 1$ define uma simetria da 2EDO (13) na forma evolucionária.*

*Prova.* Veja em (3).

O teorema (1.3) e o corolário (1.2) estabelecem uma conexão entre as funções-$S$ e as simetrias de Lie. Foi esta conexão que permitiu evitar o uso de polinômios de Darboux na busca pelas integrais primeiras para a 2EDO (13). O conceito principal é a 1EDO [14], que é uma 1EDO que tem sua solução geral definida por uma das integrais primeiras da 2EDO.

**Definição 1.7** *Seja $I$ uma integral primeira da 2EDO 13 e seja $S(x, y, z)$ a função-S associada com (13) através de $I$. A equação diferencial ordinária de primeira ordem definida por*

$$\frac{dz}{dy} = -S(x, y, z), \tag{31}$$

*onde $x$ é tomado como um parâmetro, é chamada **1EDO associada** com a 2EDO (13) através de $I$.*

**Teorema 1.4** *Seja $I$ uma integral primeira da 2EDO (13) e seja S(x,y,z) a função-S associada com (13) através de $I$. Então $I(x, y, z) = C$ é uma solução geral da 1EDO associada com (13) através de $I$.*

*Prova.* Veja em (3).

[14] Este conceito foi desenvolvido em (40)

**Observação 1.6** *A variável $x$, que é a variável independente da 2EDO (13), é apenas um parâmetro na 1EDO (31), então, desde que qualquer função de $x$ é um invariante para o operador $D_a \equiv \partial_y - S\partial_z$, i.e., $D_a(x) = 0$, a relação entre uma solução geral $H(x, y, z) = K$ de 1EDO (31) e a integral primeira $I(x, y, z)$ da 2EDO (13) é $I(x, y, z) = F(x, H)$, tal que*

$$D_x(I) = \frac{\partial F}{\partial x} + (H_x + z\, H_y + \phi_z)\frac{\partial F}{\partial H} = 0. \tag{32}$$

Resumidamente, o método da função-$S$ consiste em:

- Determinar a função-$S$.
- Resolver a 1EDO associada (31), encontrando a função $H$ (veja a observação 1.6).
- Resolver a 1EDO que representa o sistema característico da equação diferencial parcial para $F$ (a 1EDP (32)).

**Observação 1.7** *Alguns comentários:*

- *Uma vez que a função-S é definida como a divisão de $I_y$ por $I_z$, podemos pensar se as divisões $I_x/I_z$ e $I_x/I_y$ teriam algum significado ou utilidade. A resposta é sim. Podemos definir funções-S $\{S_1, S_2, S_3\}$ com significados similares (veja (3)).*
- *A principal vantagem do método da função-S é que, em geral, é mais eficiente determinar a função-S do que calcular os polinômios de Darboux, precisamente nos casos onde tais polinômios têm graus relativamente altos.*
- *A grande vantagem de podermos contar com três funções-S é que a dificuldade associada com a determinação de cada uma delas pode ser muito diferente, i.e., pode ser muito mais fácil (algoritmicamente) calcular uma delas do que outras duas.*
- *Para mais detalhes e exemplos da aplicação do método, recomendamos que o leitor veja a referência (3)*

# 2 UM NOVO MÉTODO PARA RESOLVER 1EDOS QUE APRESENTAM FUNÇÕES

Neste capítulo apresentaremos um método para encontar a solução geral de uma 1EDO que apresente funções [15]. O capítulo está estruturado da seguinte maneira:

- Na primeira seção, mostraremos uma equivalência entre uma 1EDO de um certo tipo e campos vetorias polinomiais em três variáveis.
- Na seção seguinte, mostraremos como adaptar a técnica desenvolvida em (3) para campos vetoriais polinomiais em três variáveis.
- Na terceira seção, construímos um método para encontrar a solução geral da 1EDO e o aplicamos a alguns exemplos.
- Na quarta seção apresentamos uma possível melhoria do método baseados em dois resultados teóricos. Em seguida, definimos uma classe de 1EDOs para os quais podemos responder à questão 0.1 aplicando o método aperfeiçoado.
- Na quinta seção, apresentamos dois possíveis algoritmos $Lsolv$ e $FastL_S$ (baseados, respectivamente, nos métodos apresentados nas seções 2.3 e 2.4) que tentam determinar uma integral primeira Liouvilliana para o campo vetorial associado à 1EDO (6).

## 2.1 1EDO com funções × campos vetoriais polinomiais

O método que desenvolvemos é aplicável a 1EDOs que podem ser escritas na forma:

$$y' = \frac{dy}{dx} = \phi(x, y, \theta) = \frac{M(x, y, \theta)}{N(x, y, \theta)}, \tag{33}$$

onde $\phi$ é uma função racional de $(x, y, \theta)$, i.e., $\phi \in K = \mathbb{C}(x, y, \theta)$ [16], $\theta$ é um gerador elementar sobre $\mathbb{C}(x, y)$ [17] e $M$ e $N$ são polinômios coprimos em $(x, y, \theta)$, i. e., $M, N \in \mathbb{C}[x, y, \theta]$ [18]. Além disto, se $I(x, y, \theta) = c$ representa uma solução geral da 1EDO (33) e $I$

[15] Esta versão do método está limitada às funções elementares (veja (45)).

[16] $K = \mathbb{C}(x, y, \theta)$ é o campo diferencial das funções racionais nas variáveis $(x, y, \theta)$.

[17] Esta é uma maneira formal de dizer que $\theta$ é uma função elementar de $(x, y)$ (veja (45)).

[18] $\mathbb{C}[x, y, \theta]$ é um anel diferencial das funções polinomiais nas variáveis $(x, y, \theta)$.

é uma função Liouvilliana de $(x, y, \theta)$, então para que o método funcione é suficiente que as derivadas parciais de $I$ em relação a $x, y$ e $\theta$ possam ser expressas da seguinte forma:

$$I_x = RP_1 \tag{34}$$
$$I_y = RP_2 \tag{35}$$
$$I_\theta = RP_3 \tag{36}$$

onde $R$ é uma função elementar de $(x, y, \theta)$ e $P_1, P_2, P_3 \in \mathbb{C}[x, y, \theta]$ [19]. A ideia principal é realizar a transformação $z = \theta(x, y)$ que permite a associação da 1EDO (33) com um campo vetorial polinomial que tenha $I(x, y, z)$ como uma integral primeira Liouvilliana. Vamos ver como isto acontece:

- Em primeiro lugar, uma vez que $D(I) = 0$, onde $D \equiv \partial_x + \phi\,\partial_y$, temos que:

$$D(I) = \partial_x(I) + \phi\,\partial_y(I) = \frac{\partial I}{\partial x} + \frac{\partial I}{\partial \theta}\frac{\partial \theta}{\partial x} + \phi\left(\frac{\partial I}{\partial y} + \frac{\partial I}{\partial \theta}\frac{\partial \theta}{\partial y}\right). \tag{37}$$

- Em seguida, uma vez que $\theta$ é um gerador elementar sobre $\mathbb{C}(x, y)$, i.e., $\theta$ está em uma extensão elementar de $\mathbb{C}(x, y)$, então as derivadas $\frac{\partial \theta}{\partial x}$, $\frac{\partial \theta}{\partial y} \in K = \mathbb{C}(x, y, \theta)$.

- Tendo isto em vista, podemos escrever $D(I)$ como

$$\begin{aligned} D(I) &= \frac{\partial I}{\partial x} + \phi\frac{\partial I}{\partial y}\left(\frac{\partial \theta}{\partial x} + \phi\frac{\partial \theta}{\partial y}\right)\frac{\partial I}{\partial \theta} = \\ &= \frac{\partial I}{\partial x} + \phi\,\frac{\partial I}{\partial y} + (\theta_x + \theta_y\,\phi)\,\frac{\partial I}{\partial \theta} \end{aligned} \tag{38}$$

  onde $\theta_x \equiv \frac{\partial \theta}{\partial x}$ e $\theta_y \equiv \frac{\partial \theta}{\partial y}$ são funções racionais de $(x, y, \theta)$.

- Em função do fato que $\phi$, $\theta_x$ e $\theta_y$ são funções racionais de $(x, y, \theta)$ podemos multiplicar $D(I)$ pelo **mmc** dos denominadores de $\phi$, $\theta_x$ e $\theta_y\,\phi$, obtendo

$$\text{mmc}(\text{den}(\phi), \text{den}(\theta_x), \text{den}(\theta_y\,\phi))D(I) = (f\partial_x + g\partial_y + h\partial_\theta)(I(x, y, \theta)) = 0 \tag{39}$$

  onde $f$, $g$ e $h$ são polinômios em $(x, y, \theta)$, $\text{den}(u) \equiv$ denominador de $u$ e

$$f = \text{mmc}(\text{den}(\phi), \text{den}(\theta_x), \text{den}(\theta_y\,\phi)), \tag{40}$$
$$g = \text{mmc}(\text{den}(\phi), \text{den}(\theta_x), \text{den}(\theta_y\,\phi))\,\phi \tag{41}$$
$$h = \text{mmc}(\text{den}(\phi), \text{den}(\theta_x), \text{den}(\theta_y\,\phi))(\theta_x + \theta_y\,\phi). \tag{42}$$

[19] Esta é uma maneira formal de dizer que $P_1, P_2$ e $P_3$ são polinômios em $(x, y, \theta)$.

- Se fizermos a substituição $\theta \to z$, temos que

$$(f(x,y,z)\,\partial_x + g(x,y,z)\,\partial_y + h(x,y,z)\,\partial_z)(I(x,y,z)) = 0. \tag{43}$$

Então, seguindo estes passos podemos associar o campo vetorial polinomial

$$\chi \equiv f(x,y,z)\,\partial_x + g(x,y,z)\,\partial_y + h(x,y,z)\,\partial_z \tag{44}$$

com a 1EDO (33). Além disto, podemos associar a integral primeira $I(x,y,z)$ de $\chi$ (veja a equação (43)) com a solução geral da 1EDO (33), i.e. $I(x,y,\theta) = c$. Então, se pudermos encontrar a integral primeira Liouvilliana $I(x,y,z)$ para o campo vertorial $\chi$, poderemos encontrar a solução geral da 1EDO (33). Observe que temos apenas que aplicar a substituição $z \to \theta(x,y)$ em $I(x,y,z) = c$.

## 2.2 Método da função-$S$ adaptado para campos vetoriais polinomiais em três variáveis

Nesta seção mostraremos como podemos adaptar o método descrito em (3) para campos vetoriais polinomiais em três variáveis. Considere que $I(x,y,z)$ é uma integral primera Liouvilliana do campo vetorial $\chi \equiv f\,\partial_x + g\,\partial_y + h\,\partial_z$, tal que as derivadas $I_x$, $I_y$ e $I_z$ são dadas, respectivamente, por $R\,P_1, R\,P_2$ e $R\,P_3$ (veja a seção 2.1). Agora suponha que $I(x,y,z)$ é uma integral primeira Liouvilliana de uma hipotética 1EDO racional, tal que $x$ é a variável independente, $y$ é a variável dependente e $z$ é $dy/dx$. Esta 2EDO hipotética seria dada por

$$z' = \Phi(x,y,z) = -\frac{I_x + z\,I_y}{I_z} = -\frac{P1 + z\,P_2}{P_3} = \frac{M_0(x,y,z)}{N_0(x,y,z)}, \tag{45}$$

onde $M_0$ e $N_0$ são polinômios coprimos em $(x,y,z)$. Como dissemos na seção 1.2, as funções $S_1$, $S_2$ e $S_3$ associadas com a 2EDO (45) através da integral primeira $I$ são dadas pelas seguintes funções racionais:

$$S_1 = \frac{I_y}{I_z} = \frac{P_2}{P_3}, \quad S_2 = \frac{I_x}{I_z} = \frac{P_1}{P_3}, \quad S_3 = \frac{I_x}{I_y} = \frac{P_1}{P_2}, \tag{46}$$

onde $S_1$, $S_2$ e $S_3$ obedecem às equações

$$D_x(S_1) = {S_1}^2 + \Phi_z\, S_1 - \Phi_y, \tag{47}$$

$$D_x(S_2) = -\frac{{S_2}^2}{z} + \left(\Phi_z - \frac{\Phi}{z}\right) S_2 - \Phi_x, \tag{48}$$

$$D_x(S_3) = -\frac{\Phi_y}{\Phi}\, {S_3}^2 + \frac{\Phi_x - z\Phi_y}{\Phi}\, S_3 + z\,\Phi_x, \tag{49}$$

onde $D_x \equiv \partial_x + z\,\partial_y + \Phi\,\partial_z$. Uma vez que não temos a função $\Phi$, não podemos usar as equações (47), (48) e (49) para encontrar as funções $S_1$, $S_2$ e $S_3$ associadas com a 2EDO hipotética (45). Então, a fim de usar a técnica desenvolvida em (3), necessitaremos de alguma relação entre a 2EDO hipotética (45) e o campo vetorial (44). Iremos estabelecer esta relação usando o fato de que a 2EDO (45) e o campo vetorial $\chi$ tem em comum a integral primeira $I$. Este fato nos permite estabelecer um resultado que nos fornecerá uma forma de aplicar o método usado em (3) para o nosso problema. Vamos começar com a seguinte definição:

**Definição 2.1** *Seja $f\partial_x + g\partial_y + h\partial_z$ um campo vetorial polinomial em três variáveis, que apresenta uma integral primeira $I(x,y,z)$ e seja $z' = \Phi(x,y,z)$ uma 2EDO racional tal que ela também tem $I(x,y,z)$ como integral primeira. Nós dizemos que a 2EDO e o campo vetorial são* ***associados através da integral primeira*** *$I$.*

**Observação 2.1** *Note que, uma vez que as situações das variáveis (x,y,z) no campo vetorial $f\partial_x + g\partial_y + h\partial_z$ são as mesmas, existem, em princípio, seis 2EDO distintas associadas a:*

$$\Phi_1 = -\frac{I_x + zI_y}{I_z}, \tag{50}$$

$$\Phi_2 = -\frac{I_y + zI_x}{I_z}, \tag{51}$$

$$\Phi_3 = -\frac{I_x + yI_z}{I_y}, \tag{52}$$

$$\Phi_4 = -\frac{I_z + yI_x}{I_y}, \tag{53}$$

$$\Phi_5 = -\frac{I_y + xI_z}{I_x}, \tag{54}$$

$$\Phi_6 = -\frac{I_z + xI_y}{I_x}, \tag{55}$$

*Isto será útil no processo futuro de buscar a integral primeira $I$ através do método da função-S (veja os capítulos 4 e 5).*

**Teorema 2.1** *Seja $\chi$, definido pela (44), um campo vetorial polinomial que apresenta uma integral priemira Liouvilliana $I$ e, seja a 2EDO racional definida por (45), associada com o campo vetorial $\chi$ através de $I$. Se suas derivadas são dadas por:*

$$I_x = R\,P_1, \tag{56}$$
$$I_y = R\,P_2, \tag{57}$$
$$I_z = R\,P_3, \tag{58}$$

*onde $R$ é uma função Darboux de $(x, y, z)$ e $P_1$, $P_2$ e $P_3 \in \mathbb{C}[x, y, z]$, então a função $S_1$ associada a 2EDO racional (45) é dada por*

$$S_1 = \frac{M_0\,f - N_0\,h}{N_0\,(g - z\,f)}, \tag{59}$$

*onde $g - z\,f \neq 0$.*

**Prova**: Das hipóteses do teorema temos que

$$D_0(I) = N_0\,\partial_x(I) + z\,N_0\,\partial_y(I) + M_0\,\partial_z(I) = 0, \tag{60}$$
$$\chi(I) = f\,\partial_x(I) + g\,\partial_y(I) + h\,\partial_z(I) = 0, \tag{61}$$

onde $M_0$ e $N_0$ são, respectivamente, o numerador e o denominador da 2EDO (45). Definindo $D_1 \equiv f\,D_0 - N_0\,\chi$, e aplicando-o em $I$ obtemos:

$$D_1(I) = (z\,f - g)\,N_0\,\partial_y(I) + (M_0\,f - N_0\,h)\,\partial_z(I) = (f\,D_0 - N_0\,\chi)(I) = 0. \tag{62}$$

Da equação (62) temos (veja (3)) que $I_y/I_z = -(M_0 f - N_0 h)/(N_0(zf - g)) = S_1$. □

**Observação 2.2** *Notem que a função $S_1$ como apresentada na equação (59) 'define', em certo sentido, a relação entre o campo vetorial $\chi$ e a 2EDO (45). Então, podemos usá-la para produzir um semi algoritmo para encontar a 2EDO (45) (ou seja, encontrar $M_0$ e $N_0$) e, tendo em vista a equação (59), encontrar também a função-S.*

## 2.3 Um método possível

Nesta seção, veremos como usar a equação (59) no processo de construção de um algoritmo para encontrar a solução geral da 1EDO (33). Antes de escrevermos com mais detalhes os passos de um possível algoritmo, vamos delinear as etapas principais do processo e discutir sua aplicação em alguns exemplos, a fim de materilizar e esclarecer o caminho que estamos tentando seguir para atingir nosso objetivo.

**O Procedimento:** (esboço)

- Se a 1EDO pode ser colocada na forma da (33) então construa o operador $D$ (38).
- Faça a substituição $\theta \to z$ no operador $D$ e encontre o operador polinomial $\chi$ (44).
- Construa os polinômios candidatos $M_c$ e $N_c$, com coeficientes indeterminados $\{m_i\}$ e $\{n_j\}$, de algum grau escolhido, nas variáveis $(x, y, z)$.
- Substitua

  $$S_1 = \frac{M_C f - N_C h}{N_C(g - zf)}$$

  na equação para a função-$S$:

  $$D_X(S_1) = {S_1}^2 + \frac{\partial \Phi_C}{\partial z} S_1 - \frac{\partial \Phi_c}{\partial y},$$

  onde $D_X \equiv \partial_x + z\,\partial_y + \Phi_c\,\partial_z$ e $\Phi_c \equiv M_c/N_c$.
- Resolva a equação polinomial para os coeficientes indeterminados $\{m_i\}$ e $\{n_j\}$, obtendo $S_1$.
- Use o método da função-$S$ para encontrar a integral primeira Liouvilliana da 1EDO $z' = \Phi(x, y, z)$.
- Faça a substituição $z \to \theta$ na integral primeira $I$ e obtenha a solução geral desejada da 1EDO.

Antes de apresentarmos um algoritmo de uma maneira mais formal, vamos discutir o uso do procedimento acima em alguns exemplos:

**Exemplo 2.1**

Considere a 1EDO dada por

$$\frac{dy}{dx} = \frac{e^x x^3 y^2 + e^x x^2 y^2 + 2e^x x^2 y + e^x xy + e^x x + y^2 + e^x}{x^2 y^2 + e^x x^2 + xy + 1}. \tag{63}$$

Vamos aplicar o procedimento esboçado acima à 1EDO (63) e tentar obter sua solução geral:

- Escolhendo $\theta = e^x$, temos que $\theta_x \equiv \frac{\partial \theta}{\partial x} = e^x = \theta$ e $\theta_y \equiv \frac{\partial \theta}{\partial y} = 0$. Então o operador $D$ será:

$$D = \partial_x + \frac{\theta\, x^3y^2 + \theta\, x^2y^2 + 2\,\theta\, x^2y + \theta\, xy + \theta\, x + y^2 + \theta}{x^2y^2 + \theta\, x^2 + xy + 1}\partial_y + \theta\, \partial_\theta. \tag{64}$$

- Os polinômios $f$, $g$ e $h$ serão:

$$\begin{aligned} f &= x^2y^2 + zx^2 + xy + 1, && (65)\\ g &= zx^3y^2 + zx^2y^2 + 2zx^2y + zxy + zx + y^2 + z, && (66)\\ h &= z(x^2y^2 + zx^2 + xy + 1). && (67) \end{aligned}$$

O operador $\chi$ será:

$$\chi = f\partial_x + g\partial_y + h\partial_z. \tag{68}$$

- Para os graus $deg_M = 4$ e $deg_N = 5$ nós obtemos:

$$\begin{aligned} M_0 &= -(xz - y)(xz + y), && (69)\\ N_0 &= -x(xy + 1)^2, && (70)\\ S_1 &= -\frac{x^2y^2 + x^2z + xy + 1}{x(xy + 1)^2}. && (71) \end{aligned}$$

- Usando $S_1$ no método da função-$S$, obtemos a seguinte integral primeira:

$$I = e^{\frac{1}{xy+1}}(zx - y). \tag{72}$$

- Fazendo a substituição $z = e^x$, finalmente chegamos à solução:

$$e^{\frac{1}{xy+1}}(xe^x - y) = C \tag{73}$$

Agora vamos observar mais atentamente as derivadas da integral primeira $I$ (do campo

vetorial $\chi$):

$$I_x = \frac{e^{\frac{1}{xy+1}}}{(x+1)^2}(zx^2y^2+zxy+y^2+z), \tag{74}$$

$$I_y = -\frac{e^{\frac{1}{xy+1}}}{(x+1)^2}(x^2y^2+zx^2+xy+1), \tag{75}$$

$$I_z = -\frac{e^{\frac{1}{xy+1}}}{(x+1)^2}x(xy+1)^2. \tag{76}$$

A partir destas derivadas podemos podemos determinar diretamente o fator integrante $R$, a 2EDO associada ($\Phi$), os polinômios, $P_1$, $P_2$ e $P_3$ e as funções-$S$ $S_1$, $S_2$ e $S_3$:

$$\Phi = \frac{x^2z^-y^2}{(xy+1)^2x}, \tag{77}$$

$$R = \frac{e^{\frac{1}{xy+1}}}{(xy+1)^2}, \tag{78}$$

$$P_1 = x^2y^2z+xyz+y^2+z0, \tag{79}$$

$$P_2 = -(x^2y^2+x^2z+xy+1), \tag{80}$$

$$P_3 = x(xy+1)^2, \tag{81}$$

$$S_1 = -\frac{x^2y^2+zx^2+xy+1}{(xy+1)^2x}, \tag{82}$$

$$S_2 = \frac{zx^2y^2+zxy+y^2+z}{(xy+1)^2x}, \tag{83}$$

$$S_3 = -\frac{zx^2y^2+zxy+y^2+z}{x^2y^2+zx^2+xy+1}. \tag{84}$$

As componentes $(f,g,h)$ do campo vetorial $\chi$ são

$$f = x^2y^2+zx^2+xy+1, \tag{85}$$

$$g = zx^3y^2+zx^2y^2+2zx^2y+zxy+zx+y^2+z, \tag{86}$$

$$h = z(x^2y^2+zx^2+xy+1). \tag{87}$$

**Observação 2.3** *Comparando $(f,g,h)$ com os polinômios $P_1$, $P_2$ e $P_3$ e com os numera-*

*dores e denominadores das funções-S, podemos ver que:*

$$N_0 = P_3, \quad f = -P_2, \quad h = -zP_2. \tag{88}$$

*A primeira destas equações pode não surpreender, uma vez que* $\Phi = -(P_1 + zP_2)/P_3 = M_0/N_0$. *Contudo as duas equações seguintes são, à primeira vista, intrigantes.*

**Exemplo 2.2**

Considere agora a 1EDO dada por:

$$\frac{dy}{dx} = \frac{-e^y(e^{2y}x^2y - 2e^yxy^2 - xe^yy + y^3 - 1)}{e^{3y}x^3y + e^{3y}x^3 - 2e^{2y}x^2y^2 - 3e^{2y}x^2y + e^yxy^3 + e^yxy^2 + xe^yy - xe^y + 1} \tag{89}$$

Aplicando o procedimento na íntegra obtemos,

$$f = x^3yz^3 + z^3x^3 - 2x^2y^2z^2 - 3z^2x^2y + xy^3z + zxy^2 + xzy - zx + 1, \tag{90}$$

$$g = -z(z^2x^2y - 2zxy^2 - xzy + y^3 - 1), \tag{91}$$

$$h = -z^2(z^2x^2y - 2zxy^2 - xyz + y^3 - 1). \tag{92}$$

A função $S_1$ é

$$S_1 = \frac{z^3x^3 - 2z^2x^2y + zxy^2 + xyz + 1}{x(z^2x^2y - 2zxy^2 - xzy + y^3 - 1)}. \tag{93}$$

De $S_1$ podemos determinar a integral primeira para o vetor $\chi$ e, fazendo a substituição $z = e^y$, determinamos a solução geral da 1EDO (89):

$$I = e^{\frac{1}{zx-y}}(xyz + 1), \quad \stackrel{z\to\theta}{\longrightarrow} \quad e^{\frac{1}{e^yx-y}}(xe^y + 1) = C. \tag{94}$$

A partir das derivadas da integral primeira $I$ podemos obter $\Phi$, os polinômios $P_1$, $P_2$ e $P_3$

e as funções-$S$ $S_1$, $S_2$ e $S_3$:

$$\Phi = \frac{(zx-y)^2(zx+y)z}{x(z^2x^2y-2zxy^2-xzy+y^3-1)}, \tag{95}$$

$$P_1 = z(z^2x^2y-2zxy^2-xzy+y^3-1), \tag{96}$$

$$P_2 = z^3x^3-2z^2x^2y+zxy^2+xzy+1, \tag{97}$$

$$P_3 = x(z^2x^2y-2zxy^2-xzy+y^3-1), \tag{98}$$

$$S_1 = \frac{z^3x^-2z^2x^2y+zxy^2+xzy+1}{x(z^2x^2y-2zxy^2-xzy+y^3-1)}, \tag{99}$$

$$S_2 = \frac{z}{x}, \tag{100}$$

$$S_3 = \frac{z(z^2x^2y-2zxy^2-xzy+y^3-1)}{z^3x^3-2z^2x^2y+zxy^2+xzy+1}. \tag{101}$$

**Observação 2.4** *Novamente, se compararmos $(f,g,h)$ com os polinômios $P_1$, $P_2$ e $P_3$ e com as funções-S, veremos que:*

$$N_0 = P_3, \quad g = -P_1, \quad h = -z\,P_1. \tag{102}$$

*Além disto, como os papéis de $x$ e $z$ podem ser trocados na integral primeira $I$, i.e., as transformações $x \to z$, $z \to x$ são transformações de simetria para $I$, a função-S $S_2$ tem um formato simples.*

### Exemplo 2.3

Vamos ver agora um exemplo em que temos uma função de $x$ e $y$. Considere a 1EDO

$$\frac{dy}{dx} = \frac{xy\ln\left(\frac{x}{y}\right)+(xy^5-2x^2y^3+x^3y+y^4+x^2y-2xy^2+x^2)y}{2xy^2\ln\left(\frac{x}{y}\right)+x(-xy^5+2x^2y^3-x^3y+2xy^3+y^4-2xy^2+x^2)}. \tag{103}$$

Temos:

$$f = x(xy^5-2x^2y^3+x^3y-2xy^3-y^4+2xy^2-2zy^2-x^2), \tag{104}$$

$$g = -(xy^5-2x^2y^3+x^3y+y^4+x^2y-2xy^2+x^2+zx)y, \tag{105}$$

$$h = 2xy^5-4x^2y^3+2x^3y-2xy^3+x^2y-2zy^2+zx. \tag{106}$$

A função $S_1$ é

$$S_1 = \frac{xy^4 - 2x^2y^2 + x^3 - 2xy^2 - 2yz}{y^4 - 2xy^2 + x^2}. \tag{107}$$

A intergral primeira e a solução geral da 1EDO são:

$$I = e^{\frac{1}{y^2-x}}(xy + z), \quad e^{\frac{1}{y^2-x}}\left(xy + \ln\left(\frac{x}{y}\right)\right) = C. \tag{108}$$

Os polinômios $P_1$, $P_2$, $P_3$ e $\Phi$ são:

$$P_1 = y^5 - 2xy^3 + x^2y + xy + z, \tag{109}$$

$$P_2 = xy^4 - 2x^2y^2 + x^3 - 2xy^2 - 2yz, \tag{110}$$

$$P_3 = (-y^2 + x)^2 \tag{111}$$

$$\Phi = \frac{xy^4z - 2x^2y^2z + y^5 + x^3z - 2xy^3 - 2xy^2z + x^2y - 2yz^2 + xy + z}{(-y^2 + x)^2}. \tag{112}$$

**Observação 2.5** *Alguns comentários:*

1. *Desta vez, olhando para os polinômios $P_1$, $P_2$ e $P_3$ no exemplo 3, vemos que nenhum deles é um membro de $\{cf, cg, ch\}$ onde $c$ é uma constante.*

2. *A diferença entre os exemplos 1, 2 e o exemplo 3 está no fato de que, nos exemplos 1 e 2 a função elementar presente na 1EDO era uma função de somente uma variável (ou $x$, ou $y$), enquanto que no exemplo 3 a função $\theta$ era uma função de duas variáveis $(x, y)$, i.e., $\theta = \theta(x, y)$.*

3. *Em geral é muito mais simples, computacionalmente falando, calcular a função-S se já conhecemos seu numerador ou denominador. Nesses casos, o cálculo é feito de modo muito mais eficiente, i.e., mais rápido e com menor consumo de memória (veja o capítulo 4).*

4. *O que aconteceu nos exemplos 1 e 2 não foi por mero acaso. Podemos mostrar que, em quase todos os casos, se $\theta$ é uma função elementar de somente uma variável, então um dos coeficientes do campo vetorial $\chi$ divdidirá um dos polinômios $(P_1, P_2, P_3)$.*

5. *A melhor notícia é que, se $\theta$ é uma função elementar, em geral podemos realizar uma transformação de variáveis que leva $\theta(x, y)$ em $\bar{\theta}(x)$ ou em $\bar{\theta}(y)$.*

## 2.4 Um aperfeiçoamento do método

O que foi apresentado na observação 2.5 (comentário 4) pode ser demonstrado e, em conjunto com o fato apresentado na observação 2.5 (comentário 5), pode nos ajudar muito a melhorar a eficiência do algoritmo numa grande variedade de casos. Antes de apresentar este resultado, vamos definir um pouco melhor o tipo de 1EDO com que esta melhoria pode lidar.

**Definição 2.2** *Considere que a 1EDO descrita pela equação (33) tem as seguintes características:*

1. *$I(x, y, \theta(x, y)) = c$, onde $c$ é constante, representa uma solução geral da 1EDO (33) e $I$ é uma função Liouvilliana de $(x, y, \theta)$.*

2. *A função $\theta(x, y)$ está em uma extensão elementar $E$ do corpo diferencial (de característica zero) $\mathbb{C}(x, y)$ tal que $E = \mathbb{C}(x, y, exp(r))$ ou $E = \mathbb{C}(x, y, \ln(r))$, onde $r \in \mathbb{C}(x, y)$.*

3. *A 1EDO (33) é associada ao campo vetorial $\chi \equiv f\,\partial_x + g\,\partial_y + h\,\partial_z$, tal que $f, g, h \in \mathbb{C}[x, y, z]$ e*

$$\begin{aligned} f &= \text{mmc}(\text{den}(\phi), \text{den}(\theta_x), \text{den}(\theta_y\,\phi))|_{\theta\to z}, \\ g &= \text{mmc}(\text{den}(\phi), \text{den}(\theta_x), \text{den}(\theta_y\,\phi))\,\phi|_{\theta\to z}, \\ h &= \text{mmc}(\text{den}(\phi), \text{den}(\theta_x), \text{den}(\theta_y\,\phi))(\theta_x + \phi\,\theta_y)|_{\theta\to z}. \end{aligned}$$

4. *As derivadas da integral primeira $I$ (do campo vetorial $\chi$) são da forma:*

$$\partial_k(I) = exp(A/B)\prod_i {p_i}^{n_i} P_k,\ k \in \{1, 2, 3\}, \tag{113}$$

   *onde $A, B, p_i, P_k$ são polinômios em $\mathbb{C}[x, y, z]$ e $n_i$ são inteiros.*

5. *Seja $T$ a transformação $T = \{u = r(x, y), v = y\}$ (ou $T = \{u = x, v = r(x, y)\}$), onde $r \in \mathbb{C}(x, y)$ é o argumento de $\theta$. Se $T^{-1} = \{x = r_1(u, v), y = v\}$ (ou $T^{-1} = \{x = u, y = r_1(u, v)\}$) denota sua inversa, vamos assumir que $r_1$ é uma função racional de $(u, v)$, i.e., $r_1 \in \mathbb{C}(u, v)$.*

*Vamos indicar o conjunto das 1EDOs que têm estas carcteríticas como $L_S$.*

Vamos agora apresentar dois resultados que, para uma 1EDO $\in L_S$, podem ser usados para construir uma melhoria no método esboçado acima. O primeiro resultado pode ser enunciado na forma do seguinte teorema:

**Teorema 2.2** *Seja $\chi \equiv f\partial_x + g\partial_y + h\partial_z$ um campo vetorial polinomial associado à 1EDO (33) ($\phi = M/N$, $M, N \in \mathbb{C}[x,y,\theta]$) tal que a integral primeira $I(x,y,z)$ de $\chi$ define a solução geral da 1EDO (33) ($I(x,y,\theta) = C$). Alem disso, as derivadas de $I$ são da forma: $\partial_k(I) = exp(A/B)\prod_i {p_i}^{n_i} P_k$, $k \in \{1,2,3\}$, onde $A, B, p_i, P_k$ são polinômios em $\mathbb{C}[x,y,z]$ e $n_i$ são números racionais. Podemos afirmar que*
*i) Se $\theta = \mathrm{e}^x$, então $f|P_2$, $h = z\,f$ e $g|(P_1 + z\,P_3)$.*
*ii) Se $\theta = \ln(x)$ então $h|P_2$, $f = x\,h$ e $g|(x\,P_1 + P_3)$.*
*iii) Se $\theta = \mathrm{e}^y$, $g|P_1$, $h = z\,g$ e $f|(P_2 + z\,P_3)$.*
*iv) Se $\theta = \ln(y)$ então $h|P_1$ $g = y\,h$ e $f|(y\,P_2 + P_3)$.*

*Prova.* Vimos que podemos escrever o operador $D \equiv \partial_x + \phi\,\partial_y$, associado à 1EDO (33), na forma $D = \partial_x + \phi\,\partial_y + (\theta_x + \phi\,\theta_y)\partial_\theta$, onde $\theta_x \equiv \frac{\partial\theta}{\partial x}$ e $\theta_y = \frac{\partial\theta}{\partial y}$ são funções racionais de $(x,y,\theta)$. A fim de obter o campo vetorial $\chi$ temos que multiplicar $D$ pelo mmc dos denominadores de $\phi$, $\theta_x$ e $\phi\,\theta_y$ e fazer a substituição $\theta \to z$, obtendo $\chi = f\,\partial_x + g\,\partial_y + h\,\partial_\theta$, onde $f$, $g$ e $h$ são polinômios dados por

$$\begin{aligned} f &= \mathrm{mmc}(\mathrm{den}(\phi), \mathrm{den}(\theta_x), \mathrm{den}(\theta_y\,\phi))|_{\theta\to z}, \\ g &= \mathrm{mmc}(\mathrm{den}(\phi), \mathrm{den}(\theta_x), \mathrm{den}(\theta_y\,\phi))\,\phi|_{\theta\to z}, \\ h &= \mathrm{mmc}(\mathrm{den}(\phi), \mathrm{den}(\theta_x), \mathrm{den}(\theta_y\,\phi))(\theta_x + \phi\,\theta_y)|_{\theta\to z}. \end{aligned}$$

Agora, podemos provar as afirmações:

($i$): Se $\theta = e^x$, então $\theta_x = z$ e $\theta_y = 0$ e, então

$$f = N(x,y,z),\ g = M(x,y,z),\ h = zN(x,y,z). \tag{114}$$

Temos que, por hipótese, $\chi(I) = fI_x + gI_y + hI_z = fRP_1 + gRP_2 + hRP_3 = R(fP_1 + gP_2 + hP_3) = 0$. Substituindo (114) e notando que $R$ não é nulo, podemos escrever $NP_1 + MP_2 + zNP_3 = 0$, implicando que $N(P_1 + zP_3) = -MP_2$. Portanto, podemos escrever

$$P_1 + zP_3 = -\frac{MP_2}{N} \quad \text{e} \quad P_2 = -\frac{N(P_1 + zP_3)}{M}. \tag{115}$$

Uma vez que $M$ e $N$ são coprimos, temos que $N|P_2 \Rightarrow f|P_2$ e $M|(P_1+zP_3) \Rightarrow g|(P_1+zP_3)$.

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

$$f = xN(x,y,z),\ g = xM(x,y,z),\ h = N(x,y,z), \tag{116}$$

se $x$ não é um fator de $N$ ou

$$f = x^n P_N(x,y,z),\ g = M(x,y,z),\ h = x^{n-1}P_N(x,y,z), \tag{117}$$

onde $P_N = N/x^n$ é um polinômio, $n$ é um inteiro positivo e $x$ e $P$ são coprimos.
Primeiro caso ($x$ e $N$ são coprimos): $N(xP_1 + P_3) = -xMP_2$. Portanto, $N|P_2 \Rightarrow h|P_2$ e $xM = g|(xP_1 + P_3)$.
Segundo caso($x|N$): $x^{n-1}P_N(x\,P_1 + P_3) = -M\,P_2$. Uma vez que $M$ e $N$ são coprimos, temos que $x^{n-1}P_N(= h)$ e $M$ são coprimos. Portanto $h|P_2$ e $g|(x\,P_1 + P_3)$.

$(iii)$: Se $\theta = e^y$, então $\theta_x = 0$ e $\theta_y = z$ e, então

$$f = N(x, y, z),\ g = M(x, y, z),\ h = z\,M(x, y, z). \tag{118}$$

Podemos escrever $N\,P_1 + M\,P_2 + z\,M\,P_3 = 0$, implicando que $N\,P_1 = -M\,(P_2 + z\,P_3)$. Portanto, podemos escrever

$$P_2 + z\,P_3 = -\frac{N\,P_1}{M} \text{ and } P_1 = -\frac{M\,(P_2 + z\,P_3)}{N}. \tag{119}$$

Uma vez que $M$ e $N$ são coprimos, temos que $N|(P_2+z\,P_3) \Rightarrow f|(P_2+z\,P_3)$ e $M|P_1 \Rightarrow g|P_1$.

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

$$f = y\,N(x, y, z),\ g = y\,M(x, y, z),\ h = M(x, y, z), \tag{120}$$

se $y$ não é um fator de $M$ ou

$$f = N(x, y, z),\ g = y^n P_M(x, y, z),\ h = y^{n-1}P_M(x, y, z), \tag{121}$$

onde $P_M = M/y^n$ é um polinômio, $n$ é um inteiro positivo e $y$ e $P_M$ são coprimos.
Primeiro caso ($y$ e $M$ são coprimos): $N\,y\,P_1 = -M\,(y\,P_2+P_3)$. Portanto, $M|P_1 \Rightarrow h|P_1$ e $y\,N|(y\,P_2 + P_3) \Rightarrow f|(y\,P_2 + P_3)$.
Segundo caso ($y|M$): $y^{n-1}P_M\,(y\,P_2 + P_3) = -\,N\,P_1$. Como $M$ e $N$ são coprimos, temos que $y^{n-1}P_M\,(= h)$ e $N$ são coprimos. Portanto, $h|P_1$ e $f|(y\,P_2 + P_3)$. □

**Corolário 2.1** *Se a 1EDO (33) $\in L_S$, então existe uma transformação racional $T$ (com uma inversa racional $T^{-1}$) tal que a 1EDO$_{tr}$ transformada tem um campo vetorial associado $\chi_{tr}$ no qual $f_i|P_j$ para algum $f_i \in \{f, g, h\}$ e para algum $P_j \in \{P_1, P_2, P_3\}$.*

*Prova.* Para qualquer 1EDO $\in L_S$ podemos realizar uma transformação de variáveis que leve $\theta(x, y)$ a $\exp(x), \ln(x), \exp(y)$ ou $\ln(y)$. Assim, qualquer 1EDO $\in L_S$ pode ser transformada numa 1EDO que atende totalmente às premissas do teorema 2.2. Portanto, a conclusão é uma consequência direta do teorema 2.2. □

O segundo resultado vem do fato que podemos expressar as condições de compatibilidade para as derivadas cruzadas da integral primeira $I$ do campo vetorial $\chi$ em termos

de $f$, $g$ e $h$. Isto nos permite escrever a 1EDP para a função-$S$ $S_1$ em termos de $f$, $g$ e $h$. Podemos escrever esse resultado na forma do seguinte teorema:

**Teorema 2.3** *Seja $I$ uma integral primeira do campo vetorial $\chi \equiv f\,\partial_x + g\,\partial_y + h\,\partial_z$ associado com a 1EDO* (6)*. Então a função-S associada com $\chi$ através de $I$ obedece à 1EDP dada por*

$$\chi(S) = S^2\left(\frac{fg_z - gf_z}{f}\right) + S\left(\frac{gf_y - fg_y + fh_z - hf_z}{f}\right) - \left(\frac{fh_y - hf_y}{f}\right). \tag{122}$$

*Prova.* Veja a demonstração do teorema 5.2 em (51). □

Uma vez que $S_1 = I_y/I_z = P_2/P_3$ e que o 'status' das variáveis $(x, y, z)$ no campo vetorial $f\,\partial_x + g\,\partial_y + h\,\partial_z$ é o mesmo, podemos realizar as seguintes transformações de variáveis:

- Em primeiro lugar, podemos aplicar a transformação $T_1$ tal que sua inversa é dada por ${T_1}^{-1} = \{y_1 = x, x_1 = r(x, y)\}$ (onde $r(x, y)$ é o argumento de $\theta$) para obter uma $1\text{EDO}_{[1]}$ tal que $\theta_{[1]} = \exp(x_1)$ or $\theta_{[1]} = \ln(x_1)$.

- Se $\theta_{[1]} = \exp(x_1)$ podemos aplicar a transformação $T_2 = \{x_1 = \ln(x_2), y_1 = y_2\}$ e obter a $1\text{EDO}_{[2]}$ tal que $\theta_{[2]} = \ln(x_2)$.

- Para a $1\text{EDO}_{[2]}$, o campo vetorial associado $\chi_2$ é dado por $f_2\,\partial_{x_2} + g_2\,\partial_{y_2} + h_2\,\partial_{z_2}$, onde (para tornar a notação mais 'leve', omitimos o índice $_{[2]}$ nas equações seguintes):

  $$f = x\,N(x, y, z),\ g = x\,M(x, y, z),\ h = N(x, y, z), \tag{123}$$

  se $x$ não é um fator de $N$ ou

  $$f = x^n\,P_N(x, y, z),\ g = M(x, y, z),\ h = x^{n-1}P_N(x, y, z), \tag{124}$$

  onde $P_N = N/x^n$ é um polinômio, $n$ é um inteiro positivo e $x$ e $P_N$ são coprimos (veja a afirmação $(ii)$ do teorema 2.2).

- Uma vez que o status das variáveis $(x_2, y_2, z_2)$ no campo vetorial $\chi_2$ é o mesmo, podemos usar a transformação $T_3 = \{x_2 = y_3, y_2 = z_3, z_2 = x_3\}$ que fará $P_2$ tomar o lugar de $P_3$ (que é o denominador da função-$S$ transformada). O campo vetorial transformado $\chi_3$ será dado por $f_3\,\partial_{x_3} + g_3\,\partial_{y_3} + h_3\,\partial_{z_3}$, onde

  $$\begin{aligned} f_3 &= T_3(h_2), \\ g_3 &= T_3(f_2), \\ h_3 &= T_3(g_2). \end{aligned}$$

Neste ponto do processo, podemos usar a equação (122) para determinar o numerador da função-$S$ transformada ($S_{1[3]}$). Aplicando a transformação ${T_3}^{-1}$ (a inversa de $T_3$), obtemos a função-$S$ associada ao campo vetorial $\chi_2$ ($S_{1[2]}$) e a utilizamos para resolver a 1EDO$_{[2]}$ obtendo $I_{[2]} = C_{[2]}$. Se aplicarmos a transformação ${T_2}^{-1}$ seguida de ${T_1}^{-1}$ a $I_{[2]}$ obteremos $I = {T_1}^{-1}({T_2}^{-1}(I_{[2]}))$, onde $I = C$ representa a solução geral da 1EDO original. Vamos tornar esses passos mais claros em um exemplo:

**Exemplo 2.4**

Considere a 1EDO dada por

$$\frac{dy}{dx} = \frac{2\,(\mathrm{e}^{xy})^2\,x^3y^2 - \mathrm{e}^{xy}x^3y^2 + \mathrm{e}^{xy}xy^3 - 5\,\mathrm{e}^{xy}x^2y + \mathrm{e}^{xy}y^2 + 2\,x}{(\mathrm{e}^{xy})^2\,x^2y^2 + \mathrm{e}^{xy}x^4y - \mathrm{e}^{xy}x^2y^2 + \mathrm{e}^{xy}x^3 - 3\,xy\mathrm{e}^{xy} + 1}. \tag{125}$$

- Em primeiro lugar, vamos fazer a transformação $T_1 = \{x = x/y,\, y = y\}$ que nos leva à 1EDO$_{[1]}$:

$$\frac{dy}{dx} = \frac{y\,(2\,\mathrm{e}^{2\,x}x^3 + \mathrm{e}^{x}xy^3 - \mathrm{e}^{x}x^3 + \mathrm{e}^{x}y^3 - 5\,\mathrm{e}^{x}x^2 + 2\,x)}{\mathrm{e}^{2\,x}x^2y^3 + 2\,\mathrm{e}^{2\,x}x^4 - 2\,\mathrm{e}^{x}xy^3 - 4\,\mathrm{e}^{x}x^3 + y^3 + 2\,x^2}. \tag{126}$$

- Em seguida, vamos aplicar a transformação $T_2 = \{x = \ln(x),\, y = y\}$ obtendo a 1EDO$_{[2]}$:

$$\frac{dy}{dx} = \frac{y(2\,x^2\ln(x)^3 + x\ln(x)y^3 - x\ln(x)^3 + xy^3 - 5\,x\ln(x)^2 + 2\,\ln(x))}{(x^2\ln(x)^2y^3 + 2\,x^2\ln(x)^4 - 2\,x\ln(x)y^3 - 4\,x\ln(x)^3 + y^3 + 2\,\ln(x)^2)x}. \tag{127}$$

- O campo vetorial $\chi_{[2]}$ associado com a 1EDO$_{[2]}$ é definido por:

$$f_2 = \left(y^3 + 2\,z^2\right)(xz - 1)^2\,x, \tag{128}$$

$$g_2 = y\left(2\,x^2z^3 + zxy^3 + xy^3 - xz^3 - 5\,z^2x + 2\,z\right), \tag{129}$$

$$h_2 = \left(y^3 + 2\,z^2\right)(xz - 1)^2. \tag{130}$$

- Aplicando a transformação $T_3 = \{x = y, y = z, z = x\}$ vamos obter o campo vetorial $\chi_{[3]}$ definido por:

$$f_3 = \left(z^3 + 2\,x^2\right)(xy - 1)^2, \tag{131}$$

$$g_3 = \left(z^3 + 2\,x^2\right)(xy - 1)^2\,y, \tag{132}$$

$$h_3 = z\left(2\,x^3y^2 + xyz^3 - x^3y + yz^3 - 5\,x^2y + 2\,x\right). \tag{133}$$

- Construindo um polinômio $P_c$ em $(x, y, z)$ com coeficientes arbitrários e substituindo na equação (122), obtemos (para o grau 5):

$$P = x\,z\left(-z^3 + x^2\right) \quad \Rightarrow \quad S_{1\,[3]} = \frac{x\,z\left(-z^3 + x^2\right)}{\left(z^3 + 2\,x^2\right)\left(xy - 1\right)^2}. \tag{134}$$

- Usando o método da função-$S$, podemos encontrar uma integral primeira $I_3$ para o campo vetorial $\chi_{[3]}$:

$$I_{[3]} = -\ln\left(z^3 - x^2\right) + 2\,\ln\left(z\right) - \frac{1}{xy - 1}. \tag{135}$$

- Aplicando a transformação ${T_3}^{-1}$ à $I_{[3]}$ obtemos a integral primeira $I_{[2]}$ (para o campo vetorial $\chi_{[2]}$):

$$I_{[2]} = -\ln\left(y^3 - z^2\right) + 2\,\ln\left(y\right) - \frac{1}{xz - 1}. \tag{136}$$

- De $I_{[2]}$ podemos escrever a solução geral da $1\text{EDO}_{[2]}$ simplesmente fazendo a substituição $z \to \ln(x)$:

$$-\ln\left(y^3 - \left(\ln\left(x\right)\right)^2\right) + 2\,\ln\left(y\right) - \frac{1}{x\ln\left(x\right) - 1} = C_2. \tag{137}$$

- Aplicando a transformação ${T_2}^{-1} = \{x = \mathrm{e}^x,\, y = y\}$ à solução geral da $1\text{EDO}_{[2]}$ (137), obtemos a solução geral da $1\text{EDO}_{[1]}$:

$$-\ln\left(y^3 - x^2\right) + 2\,\ln\left(y\right) - \frac{1}{x\mathrm{e}^x - 1} = C_1. \tag{138}$$

- Finalmente, aplicando a transformação ${T_1}^{-1} = \{x = x\,y,\, y = y\}$ à solução geral da $1\text{EDO}_{[1]}$ (138) temos a solução geral da 1EDO original (125):

$$-\ln\left(-x^2y^2 + y^3\right) + 2\,\ln\left(y\right) - \frac{1}{xy\mathrm{e}^{xy} - 1} = C. \tag{139}$$

**Observação 2.6** *Alguns comentários em relação ao aperfeiçoamento (proposto nesta seção) para o método descrito na seção 2.3:*

- *Embora a aplicação de uma sequência de transformações seguida pela sequência de transformações inversas (aplicada em ordem reversa) seja um pouco cansativa de acompanhar, na prática (isto é, computacionalmente) essas etapas não têm nenhum 'peso algorítmico', ou seja, praticamente não consomem memória ou tempo de CPU.*

- *O ponto principal desta variante do método é que essas transformações levam a um campo vetorial ($\chi_{[3]}$) cuja função-S ($S_{1\,[3]}$) associada (a ser determinada) depende da determinação de apenas um polinômio (em vez de 2). Isso torna esta variante do algoritmo muito mais eficiente.*

- *A parte mais complicada nesta variante do método continua sendo, portanto, determinar a função-S. A vantagem é que (como mencionamos) quando sabemos o denominador (ou o numerador) da função $S_1$, o método da função-S é muito eficiente precisamente nos casos em que outros métodos poderosos (como o método de simetrias de Lie ou a abordagem DPS) costumam apresentar dificuldades.*

## 2.5 **Dois algoritmos:** *Lsolv* **e** $FastL_S$

Nesta seção vamos apresentar dois algoritmos para resolver uma 1EDO da forma (33). O primeiro é mais genérico, ainda que bem menos eficiente.

### 2.5.1 O algoritmo *Lsolv*

**Algoritmo 2.1 (Lsolv)**
*Este algoritmo é baseado no método descrito na seção 2.3.*
***Passos:***

1. *Construa o campo vetorial $\chi$, i.e., determine $\{f, g, h\}$, usando a definição 2.2.*
2. *Escolha $DegMNP$, uma lista de três inteiros positivos $[d_M, d_N, d_P]$ para os graus dos candidatos $M_c$, $N_c$ e $P_c$, ou seja, os candidatos a polinômios $M_0$, $N_0$ e $P_3$.*
3. *Construa três polinômios $M_c$, $N_c$ e $P_c$ de graus $d_M$, $d_N$ e $d_P$, respectivamente, com coeficientes inderteminados.*
4. *Substitua-os na equação $E_1 : M_c f - N_c h + (zf - g)P_c = 0$.*
5. *Colete a equação $E_1$ nas variáveis $(x, y, z)$, obtendo um conjunto de equações lineares $S_{E_1}$ para os coeficientes dos polinômios $M_c$, $N_c$ e $P_c$.*
6. *Resolva $S_{E_1}$ para coeficientes indeterminados.*
7. *Substitua as soluções de $S_{E_1}$ na equação $E_2 : D_x(S_c) - {S_c}^2 - \partial_z(\Phi_c)S_c + \partial_y(\Phi_c) = 0$, onde $D \equiv \partial_x + z\partial_y + \Phi_c\partial_z$, $\Phi_c = M_c/N_c$ e $S_c = P_c/N_c$.*
8. *Colete o numerador de $E_2$ nas variáveis $(x, y, z)$ obtendo um conjunto de equações $S_{E_2}$ para os coeficientes indeterminados restantes dos polinômios $M_c$, $N_c$ e $P_c$.*

9. *Resolva $S_{E_2}$ para os coeficientes indeterminados restantes. Se nenhuma solução foi encontrada, então O ALGORITMO FALHOU.*

10. *Substitua a solução $S_{E_2}$ em $M_c$ e $N_c$ para obter $M_0$, $N_0$ e $P_c/N_c$ para obter $S_1$.*

11. *Use o método da função-S para encontrar a integral priemira Liouvilliana da 2EDO $z' = \Phi(x, y, z)$.*

12. *Faça a substituição $z \to \theta$ na integral primeira $I(x, y, z)$ e obtenha a solução geral desejada $I(x, y, \theta)$ da 1EDO original. SUCESSO.*

**Observação 2.7** *Alguns comentários:*

- *Uma vez que não existem limitantes para os graus $\{d_M, d_N, d_P\}$ o procedimento pode nunca chegar a uma conclusão, i.e., de maneira mais formal podemos dizer que* Lsolv *é um semi algoritmo.*

- *Alguns dos passos descritos acima são, obviamente, muito mais complicados do que outros. Alguns involvem algoritmos muito complexos em si mesmos, como por exemplo, o algoritmo que resolve sistemas polinomiais não lineares. Além disto, uma vez que estamos adaptando o método da função-S para resolver a parte final, o penúltimo item é, por si só, um algoritmo muito complicado.*

### 2.5.2 O algoritmo $FastL_S$

**Algoritmo 2.2 (FastLs)**
*Este algoritmo é um aperfeiçoamento do algoritmo 2.1 e é baseado no resultado do teorema 2.2 e do corolário 2.1. Ele é restrito a 1EDOs $\in L_S$ (veja a definição 2.2).*
***Passos:***

1. *Aplique a transformação de variáveis $T_1 = \{x = r(x, y), y = y\}$, onde $r(x, y)$ é o argumento de $\theta$, e obtenha uma 1EDO$_{[1]}$ tal que $\theta_{[1]} = \exp(x)$ ou $\theta_{[1]} = \ln(x)$.*

2. *Se $\theta_{[1]} = \exp(x)$ então faça uma transformação $T_2 = \{x = \ln(x), y = y\}$ e obtenha uma 1EDO$_{[2]}$ tal que $\theta_{[2]} = \ln(x)$.*

3. *Construa o campo vetorial $\chi_2 = f_2\,\partial_x + g_2\,\partial_y + h_2\,\partial_z$ associado com a 1EDO$_{[2]}$, onde*

$$f_2 = x\,N(x, y, z),\ g_2 = x\,M(x, y, z),\ h_2 = N(x, y, z), \tag{140}$$

*se $x$ não é fator de $N$ ou*

$$f_2 = N(x,y,z),\ g_2 = M(x,y,z),\ h_2 = \frac{N(x,y,z)}{x}, \tag{141}$$

*se $x|N$.*

4. *Aplique a transformação $T_3 = \{x = y, y = z, z = x\}$ ao campo vetorial $\chi_{[2]}$, obtendo $\chi_3 = f_3\,\partial_x + g_3\,\partial_y + h_3\,\partial_z$, onde*

$$\begin{aligned} f_3 &= T_3(h_2), \\ g_3 &= T_3(f_2), \\ h_3 &= T_3(g_2). \end{aligned}$$

5. *Escolha um inteiro positivo $d_P$ para o grau de $P_c$ (candidato para o polinômio $P_2$ – numerador da função-S).*

6. *Construa um polinômio $P_c$ de grau $d_P$ nas variáveis $(x,y,z)$ com coeficientes arbitrários e substitua-o na equação $E_1$: (122).*

7. *Colete a equação $E_1$ nas variáveis $(x,y,z)$ obtendo um conjunto de equações quadráticas $S_{E_1}$ para os coeficientes do polinômio $P_c$.*

8. *Resolva $S_{E_1}$ para os coeficientes indeterminados. Se nenhuma solução for encontrada então O ALGORITMO FALHOU.*

9. *Substitua a solução de $S_{E_1}$ em $P_c/f$ e obtenha $S_{1\,[3]}$.*

10. *Use o método da função-S para determinar a integral primeira $I_{[3]}$ do campo vetorial $\chi_{[3]}$.*

11. *Aplique a transformação ${T_3}^{-1}$ (inversa de $T_3$) e obtenha a integral primeira $I_{[2]}$ do campo vetorial $\chi_{[2]}$ associado à 1EDO$_{[2]}$. De $I_{[2]}$ obtemos (substituindo $z$ por $\ln(x)$) a solução geral da 1EDO$_{[2]}$.*

12. *Aplique a transformação ${T_2}^{-1}$ (inversa de $T_2$) à solução geral da 1EDO$_{[2]}$ e obtenha a solução geral da 1EDO$_{[1]}$.*

13. *Aplique a transformação ${T_1}^{-1}$ (inversa de $T_1$) à solução geral da 1EDO$_{[1]}$ e obtenha a solução geral da 1EDO original. SUCESSO.*

**Observação 2.8** *O algoritmo $FastLs$ usa a equação (122) para determinar o polinômio $P$ (o numerador da função-S). Como a equação (122) tem um termo da forma $S^2$, podemos escrevê-la como (veja o exemplo 4):*

$$\chi\left(\frac{P}{f}\right) = \left(\frac{P}{f}\right)^2 \frac{f g_z - g f_z}{f} + \left(\frac{P}{f}\right) \frac{g f_y - f g_y + f h_z - h f_z}{f} - \frac{f h_y - h f_y}{f}. \tag{142}$$

*Podemos isolar a equação (142) para $P^2$, obtendo $P^2 =$ polinomial. Visto que os monômios de maior grau de $P$ ao quadrado não podem se cancelar (com eles mesmos), o grau máximo do polinômio $P$ é limitado. Nesse caso, portanto, a parte do algoritmo $FastLs$ que determina a função-S é um algoritmo completo.*

**Observação 2.9** *O fato que $h|P_2$, $f = x\,h$ e $g|(x\,P_1+P_3)$ (veja o teorema 2.2, afirmação (ii)) não implica que $f = c\,P_2$ e $g = -c\,(P_1 + z\,P_3)$ ($c$ constante). Se os polinômios $P_2$ e $(x\,P_1 + P_3)$ possuírem um fator polinomial em comum, o algoritmo $FastL_S$ pode perder esse caso.*

# 3 O PACOTE *LEAPS1ODE*

Neste capítulo apresentaremos uma implementação dos algoritmos 2.1 e 2.2 em um pacote computacional escrito na linguagem Maple, com algumas modificações para melhorar sua flexibilidade e praticidade.

1. Na primeira seção, mostramos os comandos que compõe o pacote *LeapS1ode*: fazemos uma descrição detalhada da funcionalidade e das chamadas de cada um dos comandos que compõe o pacote.

2. Na segunda seção, apresentamos exemplos práticos do uso dos comandos do pacote aplicados a exemplos específicos.

## 3.1 Os comandos do pacote *LeapS1ode*

Nesta seção apresentaremos uma descrição (detalhada) dos principais comandos do pacote *LeapS1ode*.

**Resumo dos comandos:**

- `Dx` constrói o operador $D_x$ (o operador derivada total sobre as soluções) associado com a 1EDO (6).

- `Xi` - constrói o operador $\chi$ associado com a 1EDO (6).

- `Tr1ode` - aplica uma transformação (dada pelo usuário) na 1EDO (6) e retorna a 1EDO transformada e a transformação inversa.

- `Ode2` - tenta determinar a 2EDO associada ao campo vetorial $\chi$.

- `Sfunction` - tenta determinar a função-$S$ associada com o campo vetorial $\chi$

- `Ode1a` - tenta determinar a 1EDO racional associada com a 2EDO.

- `Hfunction` - tenta encontrar a solução geral para 1EDO racional associada.

- `PDEassol` - constrói e tenta resolver a 1EDP que relaciona a integral primeira $I$ com as funções-$H$.

- `Solde` - tenta determinar a integral primeira do campo vetorial $\chi$ ou, equivalentemente, a solução geral da 1EDO (6).

### 3.1.1 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 (6).

*Sinopse:*

O comando `Dx` retorna o *operador derivada total* sobre as soluções da 1EDO (6), 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

$$D_x = (\partial_x + y'\,\partial_y)|_{y'=\phi} = \partial_x + \phi\,\partial_y, \quad \left(y' = \frac{dy}{dx}\right). \tag{143}$$

### 3.1.2 Comando: `Xi`

*Funcionalidade:* Este comando retorna o campo vetorial $\chi$ associado com a 1EDO (6).

*Linha de comando*:

[> `Xi (ode);`

*Parâmetros:*

`ode` - a 1EDO (6).

*Parâmetros extras:*

`FGH` - Se este parâmetro extiver presente na chamada do comando, então o comando retorna uma lista : $[f, g, h]$.

*Sinopse:*

O comando `Xi` retorna o *campo vetorial $\chi$ associado com a 1EDO* (6), i.e., o comando retorna um operador $u \to f\partial_x(u) + g\partial_y(u) + h\partial_z(u)$, onde $f$, $g$ e $h$ são polinômios dados por (114,114,114). Se o parâmetro extra `FGH` estiver presente na entrada do comando então o comando retorna uma lista com os polinômios $f$, $g$ e $h$: $[f, g, h]$.

### 3.1.3 Comando: `Tr1ode`

*Funcionalidade:* Este comando aplica uma transformação à 1EDO (6) e retorna a 1EDO transformada e a transformação inversa.

*Linha de comando*:

[> `Tr1ode(ode)`;

*Parâmetros:*

`ode` - a 1EDO (6).

*Parâmetros extras:*

`TR = [x=F(x,y), y=G(x,y)]`, onde `[x=F(x,y), y=G(x,y)]` é uma transformação inversível - O *default* é a transfornação identidade.

*Sinopse:*

O comando `Tr1ode` aplica uma transformação $T$ (fornecida pelo usuário) à 1EDO (6) 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 nehuma transformação é fornecida, o *default* é a transformação identidade.

### 3.1.4 Comando: `Ode2`

*Funcionalidade:* Este comando retorna a 2EDO associada à 1EDO (6).

*Linha de comando*:

[> `Ode2(ode)`;

*Parâmetros:*

`ode` - a 1EDO (6).

*Parâmetros extras:*

`PS` - Se o parâmetro estiver presente na sequência de chamada do comando, então o comando retorna os polinômios $P_1$, $P_2$ e $P_3$ : $[P_1, P_2, P_3]$.

*Sinopse:*
O comando `Ode2` retorna a função $\Phi(x, y, z) = M_0(x, y, z)/N_0(x, y, z)$ da 2EDO associada a 1EDO (6). Esta 2EDO é uma das seis possíveis 2EDOs (veja a observação 2.1 na seção 2.2), nesse caso, aquela que considera $x$ como variável independente, $y$ como variável independente e $z$ como a derivada primeira ($dy/dx$). Se o parâmetro extra `PS` estiver presente na entrada do comando, então ele retorna uma lista com os polinômios $P_1$, $P_2$ e $P_3$ : $[P_1, P_2, P_3]$.

### 3.1.5 Comando: `Sfunction`

*Funcionalidade:* Este comando tenta encontrar a função-$S$ $S_1$.

*Linha de comando*:

[> `Sfunction(ode)`;

*Parâmetros:*

`ode` - a 1EDO (6).

*Parâmetros extras:*

`Deg = n` - onde `n` é um inteiro positivo denotando o grau do polinômio $P$ (o numerador e o denominador de $S_1$). O *default* é 3.

`DegMNP = [degM,degN,degP]` - onde `[degM,degN,degP]` é uma lista de graus dos candidatos a $M_0$, $N_0$ e $P_0$, respectivamente.

`Den = deno` - onde `deno` é o denominador presumido da função-$S$.

*Sinopse:*
O comando `Sfunction` tenta encontar a função-$S$ associada (através de uma integral primeira Liouvilliana $I$) à 2EDO racional ($\Phi = M_0/N_0$). O comando computa, se possível, um polinômio $P$ que é o denominador (ou o numerador) de $S_1$. A função-$S$ é a base para encontrar a integral primeira Liouvilliana do campo vetorial $\chi$, associado à 1EDO (6) – veja a seção 2.1. No caso em que o numerador ou o denominador de $S_1$ é um dos polinômios $\{f, g, h\}$, podemos usar o parâmetro `Deg` para informar ao programa o grau que usaremos para o candidato $P_c$. Associado a este caso, temos temos também o parâmetro `Den` que pode acelerar o processo, indicando o numerador ou o denominador de $S_1$. No caso em que nenhum dos polinômios $\{f, g, h\}$ é o numerador ou o denominador de $S_1$, usamos o parâmetro `DegMNP`, que é, na realidade, uma lista com três inteiros `[degM,degN,degP]`

que diz ao comando o grau dos candidatos $M_0$, $N_0$ e $P_0$, respectivamente. De maneira mais simples: basicamente, o parâmetro `DegMNP` implementa uma versão do algoritmo $Lsolv$, ao passo que o parâmetro `Deg` implementa uma versão do algoritmo $FastL_S$.

### 3.1.6 Comando: `Ode1a`

*Funcionalidade:* Este comando determina as 1EDOs associadas.

*Linha de comando*:

[> `Ode1a(ode);`

*Parâmetros:*

`ode` - a 1EDO (6).

*Parâmetros extras:*

`SF = S1` - onde `S1` é a função-$S$ $S_1$.

*Sinopse:*

O comando `Ode1a` usa a função-$S$ para construir a 1EDO associada. Os primeiros parâmetros extras são os mesmos do comando `Sfunction` e os últimos parâmetros extras (`SF = S1`) permitem ao usuário passar a função-$S$ $S_1$ para o comando `Ode1a`.

### 3.1.7 Comando: `Hfunction`

*Funcionalidade:* Este comando tenta encontrar a solução da 1EDO associada.

*Sequência de chamada, parâmetros e parâmetros extras*: os mesmos que acima.

*Sinopse:*

O comando `Hfunction` tenta encontar a função-$H$, isto é, a solução da 1EDO racional associada (veja a definição 1.7 na seção 1.2.2) à 2EDO racional ($\Phi = M_0/N_0$). O comando tenta resolver a 1EDO racional associada usando o comando `dsolve` (o comando resolvedor de equações diferenciais padrão do Maple).

### 3.1.8 Comando: `PDEassol`

*Funcionalidade:* Este comando constrói e tenta resolver a 1EDP que relaciona a função-$H$ à integral primeira $I$.

*Sequência de chamada, parâmetros e parâmetros extras*: os mesmos que acima.

*Sinopse:*
O comando `PDEassol` constrói a 1EDP associada (veja (3)) e tenta resolvê-la gerando (e tentando resolver) a 1EDO que representa a equação característica da 1EDP associada. O comando tenta resolver a 1EDO característica usando o comando `dsolve` (o comando resolvedor de equações diferenciais padrão do Maple).

### 3.1.9 Comando: `Solde`

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

*Linha de comando*:

```
[> Solde(ode);
```

*Parâmetros:*

`ode` - a 1EDO (6).

*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 $\chi$ e a transformação $z \to \theta : [I(x, y, z),\ z = \theta(x, y)]$.

*Sinopse:*
O comando `Solde` é a parte final do processo, i.e., ele retorna, se possível, a solução geral da 1EDO. Com o parâmetro `II` o usuário pode fazer o comando retornar a integral primeira do campo vetorial $\chi$, junto com a transformação $z \to \theta$.

**Observação 3.1** *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.*

## 3.2 Exemplos da utilização dos comandos do pacote

Nesta seção vamos mostrar os comandos do pacote *LeapS1ode* simulando seu uso em uma plataforma Maple para exemplificar (de maneira prática) seu 'modus operandi'.

Considere a 1ODE (63) apresentada na seção 2.3 (exemplo 2.1):

$$\frac{dy}{dx} = \frac{\mathrm{e}^x x^3 y^2 + \mathrm{e}^x x^2 y^2 + 2\,\mathrm{e}^x x^2 y + \mathrm{e}^x xy + \mathrm{e}^x x + y^2 + \mathrm{e}^x}{x^2 y^2 + \mathrm{e}^x x^2 + xy + 1}. \tag{144}$$

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

```
[> with(DEtools):  read(‘LeapS1ode.txt‘):
```

O sinal de dois pontos ( : ) no final de uma linha de comando evita a impressão (na tela) do resultado. O pacote *DEtools* carrega vários comandos para lidar com EDOs. O comando `read (‘LeapS1ode.txt‘):` carrega nosso pacote. Vamos carregar a 1ODE (144) digitando

```
[> _1ode := diff(y(x),x) = (exp(x)*x^3*y(x)^2+exp(x)*x^2*y(x)^2
+2*exp(x)*x^2*y(x)+exp(x)*x*y(x)+exp(x)*x+y(x)^2+exp(x))/(x^2*y
(x)^2+exp(x)*x^2+x*y(x)+1):
```

Primeiro, podemos encontrar o campo vetorial associado usando o comando `Xi` (sem o parâmetro `PS` para gerar o operador $\chi$, e com o parâmetro `PS` para determinar os polinômios $f$, $g$ e $h$ – os coeficientes de $\chi$). Digitando

```
[> X := Xi(_1ode):
[> fgh := Xi(_1ode,FGH);
```

obtemos (o comando `[> X := Xi(_1ode):` atribui $X$ como o campo vetorial $\chi$ e `[> fgh := Xi (_1ode, FGH);` atribui $fgh$ como uma lista com seus coeficientes $[f, g, h]$):

$$fgh := [x^2y^2 + zx^2 + xy + 1, zx^3y^2 + zx^2y^2 + 2\,zx^2y + zxy + zx + y^2 + z, z\left(x^2y^2 + zx^2 + xy + 1\right)]$$

We can find the $S$-function $S_1$ by typing

```
[> S1 := op(Sfunction(_1ode,Deg=5));
```

$$S1 := -\frac{x^2y^2 + x^2z + xy + 1}{x\left(xy+1\right)^2} \tag{145}$$

Podemos usar o comando `Ode2` para determinar a 2EDO associada, isto é, a 2EDO racional que tem a integral primeira $I$ em comum com o campo vetorial $\chi$:

```
[> MON0 := Ode2(_1ode,SF=S1);
```

$$M0N0 := [(zx - y)\left(zx + y\right), x\left(xy+1\right)^2] \tag{146}$$

```
[> Phi := MON0[1]/MON0[2];
```

$$\Phi := \frac{(zx - y)\left(zx + y\right)}{x\left(xy+1\right)^2} \tag{147}$$

Ou seja, a 2EDO dada por

$$y'' = z' = \frac{(zx - y)\left(zx + y\right)}{x\left(xy+1\right)^2} \tag{148}$$

é construída de tal forma que uma de suas integrais primeiras (digamos $I$) é também uma integral primeira do campo vetorial polinomial $\chi$ definido por seu coeficientes $f$, $g$ e $h$ (determinados pelo comando `Xi` acima).

Podemos obter os polinômios $P_1$, $P_2$ e $P_3$ adicionando o parâmetro `PS` na chamada do comando `Ode2`:

```
[> PPP := Ode2(_1ode,SF=S1,PS);
```

$$PPP := [zx^2y^2 + zxy + y^2 + z, -x^2y^2 - zx^2 - xy - 1, x\left(xy+1\right)^2] \tag{149}$$

Usando a função $S_1$ podemos (aplicando o método da função-$S$ – veja (3)) encontrar uma integral primeira para o campo vetorial $\chi$ e também para a 2EDO associada (usando o parâmetro `I` na chamada do comando `Solde`):

```
[>  Inv := Solde(_1ode,SF=S1,II);
```

$$Inv := [\mathrm{e}^{\left(xy+1\right)^{-1}}\left(zx - y\right), z = \mathrm{e}^{x}] \tag{150}$$

Podemos confirmar que $I$ é uma integral primeira do campo vetorial $\chi$ e também da 2EDO associada (construindo o operador $D_x \equiv \partial_x + z\,\partial_y + \Phi\,\partial_z$) digitando

```
[> DX2 := u->simplify(diff(u,x)+z*diff(u,y)+Phi*diff(u,z),symbolic):
[> X(Inv[1]);
[> DX2(Inv[1]);
```

$$0$$

$$0$$

Finalmente, podemos (fazendo a substituição $z \to \theta$ ou aplicando diretamente o comando `Solde`) obter a solução geral da 1EDO (144):

```
[> sol1ode := Solde(_1ode,SF=S1);
```

$$sol1ode := \mathrm{e}^{(xy+1)^{-1}} \left(\mathrm{e}^{x} x - y\right) = \_C1 \tag{151}$$

E agora podemos testar a solução encontrada com:

```
[> DX := Dx(_1ode):
[> DX(lhs(sol1ode));
```

$$0$$

# 4 DESEMPENHO

Neste capítulo vamos discutir o desempenho dos métodos desenvolvidos neste trabalho. Com a expressão **métodos desenvolvidos** estamos nos referindo tanto aos métodos teóricos e algoritmos desenvolvidos quanto às implementações materializadas no pacote *LeapS*1*ode*. Fazemos essa discussão da seguinte maneira: dividimos a análise apresentada neste capítulo em três partes:

- Na primeira seção, mostramos como o pacote *LeapS*1*ode* lida com (e resolve com relativa facilidade) um conjunto de 1EDOs que não puderam ser resolvidas usando métodos tradicionais (bem como não tradicionais) implementados no sistema Maple de computação algébrica (Maple CAS). Mostramos também que, embora o conjunto não seja numeroso (10 1EDOs), ele (o conjunto) é apenas uma pequena amostra de uma vasta classe de 1EDOs para a qual a maioria dos métodos encontra muita dificuldade em tratar.

- Na segunda seção, mostramos algumas características especiais do pacote *LeapS*1*ode* que permitem o seu uso para resolver 1EDOs bem mais complexas e, além disso para pesquisar regiões de integrabilidade.

  - Na primeira subseção, vamos exemplificar o uso dos comandos do pacote quando nos deparamos com 1EDOs mais problemáticas do que as apresentadas na seção 4.1

  - Na segunda subseção, destacamos o uso do parâmetro `Par` que permite que o pacote (o comando `Sfunction`) aplique (em função da característica algébrica do algoritmo) uma análise da região de integrabilidade nos casos em que a 1EDO em questão apresente parâmetros que possam assumir diversos (ou infinitos – discretos ou contínuos) valores.

- Na terceira seção, fazemos uma comparação do desempenho dos algoritmos *Lsolv* e $FastL_S$ materializados (no pacote *LeapS*1*ode*) no comando `Sfunction` através dos parâmetros `DegMNP` e `Deg`, respectivamente[20]. Em seguida, fazemos uma comparação do desempenho do pacote *LeapS*1*ode* em relação à técnica que empregamos para encontrar a integral primeira do campo vetorial $\chi$ associado com a 1EDO (ou seja, o método da função-$S$ adaptado) e o procedimento baseado na abordagem DPS original (ou seja, usando o cálculo dos PDs para construir um fator integrante).

[20] Os algoritmos implementados no pacote *LeapS*1*ode* não são exatamente os algoritmos *Lsolv* e $FastL_S$ como definidos na seção 2.5. Contudo, as pequenas alterações visam apenas atender a finalidades técnicas e de flexibilidade e praticidade do pacote, não alterando em nada a essência dos algoritmos.

**Observação 4.1** *Aplicamos (no processo de testes do programa) o pacote LeapS1ode a todas as 1EDOs do livro de E. Kamke (veja (52)) que se apresentavam dentro do escopo do método e o resultado foi que o método se saiu bem (isto é, resolveu a 1EDO) em todos os casos. Como essas 1EDOs também foram resolvidas pelo resolvedor padrão do Maple – o comando* `dsolve` *– decidimos não incluir uma listagem delas nesta análise de desempenho.*

**Observação 4.2** *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`.

## 4.1 Um conjunto de 1EDOs 'difíceis'

Nesta seção apresentamos um pequeno conjunto de 1EDOs que os métodos implementados no resolvedor de equações diferenciais do Maple (`dsolve`) não conseguem resolver. Este conjunto representa apenas uma pequena amostra de uma vasta classe de 1EDOs que são muito refratárias à grande maioria dos métodos. Apresentamos a aplicação do comando `Solde` a cada uma das 1EDOs como foi feito na sessão de Maple e calculamos o tempo de CPU que o pacote *LeapS1ode* consome para encontrar a solução geral das 1EDOs do conjunto e a memória consumida em cada resolução.

Considere as seguintes 1EDOs:

1. As cinco 1EDOs seguintes possuem solução Liouvilliana elementar (vamos nos referir a elas como 1EDOs 1, 2,..., 5)

$$\frac{dy}{dx} = \frac{x\left(2\,(\ln(x))^2 x^4y^2 + 2\,\ln(x)\,x^2y + 2\,\ln(x)\,y^2 - yx^2 + y^2 + 2\right)}{(\ln(x))^2 x^4y^2 + \ln(x)\,x^4 + \ln(x)\,x^2y + 1} \tag{152}$$

$$\frac{dy}{dx} = \frac{3\,(\mathrm{e}^x)^2x^4y^4 + (y^4 - y^3x^4 - x^3y^3 + 6\,x^3y^2 + xy^4)\,\mathrm{e}^x + 3\,x^2}{(\mathrm{e}^x)^2\,x^2y^4 + (x^4y^2 - xy^3 + 2\,xy^2)\,\mathrm{e}^x - x^3 + y + 1} \tag{153}$$

$$\frac{dy}{dx} = -\frac{(2\,x^3y^4 - 4\,x^2y^2 - x^2 + \ln(y) + 2\,x)\,y}{-2\,x^4y^2 - x^2y^4 + 2\,\ln(y)\,x^2y^2 + 2\,xy^2 - 1} \tag{154}$$

$$\frac{dy}{dx} = -\frac{3\,(\mathrm{e}^y)^2\,x^4y^2 - 7\,\mathrm{e}^yx^3y + 3\,x^2 - y}{x\left((\mathrm{e}^y)^2\,x^4y^2 - 3\,\mathrm{e}^yx^3y - x^3\mathrm{e}^y + x^2 - y - 1\right)} \tag{155}$$

$$\frac{dy}{dx} = -\frac{4\,x^3y^6 + 8\,x^4y^3 + 4\,x^5 - x^4 + \cos(y)}{(y^6 + 2\,xy^3 + x^2)\sin(y) - 3\,x^4y^2 + 3\,\cos(y)\,y^2} \tag{156}$$

2. As cinco 1EDOs seguintes possuem solução Liouvilliana não elementar. Vamos nos referir a elas como 1EDOs 6, 7,..., 10.

$$\frac{dy}{dx} = \frac{(2\,x^2 - y)\,(\ln(x)\,y^2 - x^2y - 1)\,y}{x\left((\ln(x))^2\,y^3 - \ln(x)\,x^2y^2 - (\ln(x))^2\,y^2 + 2\,\ln(x)\,x^2y - x^4 - \ln(x)\,y\right)} \tag{157}$$

$$\frac{dy}{dx} = -\frac{(\ln(x))^2\,xy^2 - 2\,\ln(x)\,x^2y - \ln(x)\,xy + \ln(x)\,y^2 + x^3}{x\,(\ln(x))^2\,y} \tag{158}$$

$$\frac{dy}{dx} = \frac{(-\mathrm{e}^{2\,x}x^6y^2 - 2\,\mathrm{e}^{2\,x}x^5y^2 + \mathrm{e}^{2\,x}x^4y^2 + \mathrm{e}^{x}x^4y + 3\,\mathrm{e}^{x}x^3y + 1)\,\mathrm{e}^{-x}}{x^3\,(\mathrm{e}^{x}x^3y - x - 1)} \tag{159}$$

$$\frac{dy}{dx} = -\frac{(\ln(y))^2\,xy^4 - (\ln(y))^2\,y^4 - x^2\ln(y)\,y^2 + \ln(y)\,xy^2 + x^3}{xy\,(2\,\ln(y) + 1)\,(\ln(y)\,y^2 - x^2 - x)} \tag{160}$$

$$\frac{dy}{dx} = \frac{y^3\,(-xy^3 + \mathrm{e}^{y} + 1)}{\mathrm{e}^{y}x^2y^6 + 3\,x^2y^5 - 2\,\mathrm{e}^{y}xy^3 - 3\,\mathrm{e}^{y}xy^2 - 3\,xy^2 + \mathrm{e}^{y}} \tag{161}$$

Após abrirmos uma sessão de Maple, 'entrarmos' com as 1EDOs

```
[> _1ode[1] := diff(y(x),x)= ...
[> _1ode[2] := diff(y(x),x)= ...
```

⋮

```
[> _1ode[10] := diff(y(x),x)= ...
```

e aplicarmos o comando `Solde` a cada uma das 10 1EDOs da maneira a seguir, vamos obter:

```
[> t0 := time(): solde[1] := Solde(_1ode[1],Deg=7); time()-t0;
```

$$solde[1] := -\frac{\ln\left(x^2-y\right)\ln\left(x\right)x^2y+\ln\left(x^2-y\right)+1}{\ln\left(x\right)x^2y+1} = \_C1$$

0.563

```
[> t0 := time(): solde[2] := Solde(_1ode[2],Deg=7); time()-t0;
```

$$solde[2] := -\frac{xy^2\mathrm{e}^x+1}{\mathrm{e}^x\ln\left(x^3-y\right)xy^2+\ln\left(x^3-y\right)+y} = \_C1$$

1.078

```
[> t0 := time(): solde[3] := Solde(_1ode[3],Deg=5); time()-t0;
```

$$solde[3] := \left(-x^2+\ln\left(y\right)\right)\mathrm{e}^{\frac{x}{xy^2-1}} = \_C1$$

0.209

```
[> t0 := time(): solde[4] := Solde(_1ode[4],Deg=5); time()-t0;
```

$$solde[4] := -\frac{\mathrm{e}^y xy\ln\left(x^3\mathrm{e}^y+1\right)-\ln\left(x^3\mathrm{e}^y+1\right)+1}{xy\mathrm{e}^y-1} = \_C1$$

0.625

```
[> t0 := time(): solde[5] := Solde(_1ode[5],Deg=7); time()-t0;
```

$$solde[5] := \ln\left(-2\,\mathrm{e}^{iy}x^4+\mathrm{e}^{2\,iy}+1\right)+\frac{-iy^4-iyx+1}{y^3+x} = \_C1$$

3.687

```
[> t0 := time(): solde[6] := Solde(_1ode[6],Deg=5); time()-t0;
```

$$solde[6] := -\frac{Ei\left(1,-\left(\ln\left(x\right)y-x^2\right)^{-1}\right)y+\mathrm{e}^{\left(\ln(x)y-x^2\right)^{-1}}}{y} = \_C1$$

0.203

```
[> t0 := time(): solde[7] := Solde(_1ode[7],Deg=7); time()-t0;
```

$$solde[7] := \left(Ei\left(1,\left(\ln\left(x\right)y-x\right)^{-1}\right)\mathrm{e}^{\left(\ln(x)y-x\right)^{-1}}+x\right)\mathrm{e}^{-\left(\ln(x)y-x\right)^{-1}} = \_C1$$

0.218

```
[> t0 := time(): solde[8] := Solde(_1ode[8],Deg=9); time()-t0;
```

$$solde[8] := \left(Ei\left(1, -\left(\mathrm{e}^{x}x^{2}y-1\right)^{-1}\right)x + \mathrm{e}^{\left(\mathrm{e}^{x}x^{2}y-1\right)^{-1}}\right)x^{-1} = \_C1$$

0.687

```
[> t0 := time(): solde[9] := Solde(_1ode[9],Deg=6); time()-t0;
```

$$solde[9] := x\,\mathrm{e}^{\frac{x}{\ln(y)\,y^{2}-x}} + Ei\left(1, -\frac{x}{\ln(y)\,y^{2}-x}\right) = \_C1$$

0.437

```
[> t0 := time(): solde[10] := Solde(_1ode[10],Deg=6); time()-t0;
```

$$\left(\mathrm{e}^{-\left(xy^{3}-1\right)^{-1}}Ei\left(1, -\left(xy^{3}-1\right)^{-1}\right) + \mathrm{e}^{y}\right)\mathrm{e}^{\left(xy^{3}-1\right)^{-1}} = \_C1$$

0.297

Tabela 1 - Tempos de processamento e memória gastos

| 1EDO | Tempo (seg.) | Memória (MB) |
|---|---|---|
| 1 (152) | 0,672 | 21,07 |
| 2 (153) | 1,078 | 35,60 |
| 3 (154) | 0,209 | 4,18 |
| 4 (155) | 0,625 | 33,18 |
| 5 (156) | 3,687 | 28,47 |
| 6 (157) | 0,329 | 17,41 |
| 7 (158) | 0,218 | 16,18 |
| 8 (159) | 0,687 | 21,26 |
| 9 (160) | 0,437 | 29,86 |
| 10 (161) | 0,297 | 18,42 |

Fonte: A autora, 2020.

**Observação 4.3** *Alguns comentários:*

- *Quando iniciamos uma sessão de Maple, o simples carregamento dos pacotes básicos que são ativados com a abertura da plataforma padrão do programa já usam 4,18 MB. Assim, quando esse número aparece (na tabela acima), o significado é, basicamente, 'quase nenhum gasto' de memória.*

- *Para a medição de tempo de CPU consumido, acontece um problema parecido: a medida de tempo fornecida é 'variável de sessão para sessão', por algum motivo que desconheço. O ponto principal é que o número que representa o tempo gasto é, sempre, razoavelmente próximo e, assim, creio que podemos 'confiar' no resultado. Nos dados colocados na tabela acima evitamos fazer médias rodando muitas sessões (o que quer dizer que os dois últimos algarismos não são significativos) pois estávamos focados em um resultado mais qualitativo.*

- *Outra informação (bastante relevante para quem não 'roda muito' a plataforma Maple para resolver 1EDOs deste 'calibre' usando o comando* `dsolve`*) é que a memória consumida nas 10 1EDOs é, relativamente, muito baixa.*

- *Outro detalhe interessante é que a maior parte do tempo consumido (mesmo nos casos em que o tempo consumido é muito curto) é geralmente usado na determinação da função-S, ou seja, quase nenhum tempo de CPU é gasto com a ideia central deste trabalho. Portanto, além de validar o nosso método, o resultado anterior é também uma reafirmação do método da função-S.*

- *Comentamos que essas 1EDOs causavam dificuldades para a grande maioria dos métodos. A princípio, essa afirmação pode parecer muito exagerada, mas podemos 'verificar' que ela é bastante acurada se atentarmos para as simetrias de Lie ou para os fatores integrantes das 1EDOs do conjunto. Observando a tabela a seguir vemos que apenas as 1EDOs 3, 5 e 10 possuem fatores integrantes nos quais a função $\theta$ está ausente (e, mesmo assim, não são simples).*

Tabela 2 - Simetrias apresentadas e fatores integrantes correspondentes

| 1EDO | Simetria ($\eta$) | Fator integrante |
|---|---|---|
| 1 (152) | $-\frac{(x^2y\ln(x)+1)^2}{\mathrm{e}^{(x^2y\ln(x)+1)^{-1}}\left((\ln(x))^2x^4y^2+\ln(x)x^4+x^2y\ln(x)+1\right)}$ | $\frac{\mathrm{e}^{(x^2y\ln(x)+1)^{-1}}}{(x^2y\ln(x)+1)^2}$ |
| 2 (153) | $-\frac{(xy^2\mathrm{e}^x+1)^2\,\mathrm{e}^{\frac{-y}{xy^2\mathrm{e}^x+1}}}{(\mathrm{e}^x)^2x^2y^4+\mathrm{e}^xx^4y^2-\mathrm{e}^xxy^3+2\,xy^2\mathrm{e}^x-x^3+y+1}$ | $\frac{\mathrm{e}^{\frac{y}{xy^2\mathrm{e}^x+1}}}{(xy^2\mathrm{e}^x+1)^2}$ |
| 3 (154) | $\frac{(xy^2-1)^2y\,\mathrm{e}^{\frac{-x}{xy^2-1}}}{-2\,x^4y^2-x^2y^4+2\,\ln(y)x^2y^2+2\,xy^2-1}$ | $-\frac{\mathrm{e}^{\frac{x}{xy^2-1}}}{y\,(xy^2-1)^2}$ |
| 4 (155) | $\frac{(xy\mathrm{e}^y-1)^2\,\mathrm{e}^{(1-xy\mathrm{e}^y)^{-1}}}{\mathrm{e}^yx\left((\mathrm{e}^y)^2x^4y^2-3\,\mathrm{e}^yx^3y-x^3\mathrm{e}^y+x^2-y-1\right)}$ | $\frac{-\mathrm{e}^{(xy\mathrm{e}^y-1)^{-1}}\mathrm{e}^y}{(xy\mathrm{e}^y-1)^2}$ |
| 5 (156) | $\frac{(y^3+x)^2\,\mathrm{e}^{(-y^3-x)^{-1}}}{(\sin(y)y^6-3\,x^4y^2+2\,\sin(y)xy^3+3\,\cos(y)y^2+\sin(y)x^2)}$ | $\frac{-\mathrm{e}^{(y^3+x)^{-1}}}{(y^3+x)^2}$ |
| 6 (157) | $\frac{(\ln(x)y-x^2)^2y^2\,\mathrm{e}^{(x^2-\ln(x)y)^{-1}}}{(\ln(x))^2y^3-\ln(x)x^2y^2-(\ln(x))^2y^2+2\,\ln(x)x^2y-x^4-\ln(x)y}$ | $\frac{-\mathrm{e}^{(\ln(x)y-x^2)^{-1}}}{(\ln(x)y-x^2)^2xy^2}$ |
| 7 (158) | $\frac{(\ln(x)y-x)^2}{\mathrm{e}^{-(\ln(x)y-x)^{-1}}(\ln(x))^2\,y}$ | $\frac{-\mathrm{e}^{-(\ln(x)y-x)^{-1}}}{(\ln(x)y-x)^2\,x}$ |
| 8 (159) | $\frac{\mathrm{e}^{-\frac{\mathrm{e}^xx^3y-x+1}{\mathrm{e}^xx^2y-1}}(\mathrm{e}^xx^2y-1)^2}{x\,(\mathrm{e}^xx^3y-x-1)}$ | $\frac{-\mathrm{e}^{\frac{\mathrm{e}^xx^3y-x+1}{\mathrm{e}^xx^2y-1}}}{x^2(\mathrm{e}^xx^2y-1)^2}$ |
| 9 (160) | $\frac{\mathrm{e}^{-\frac{x}{\ln(y)y^2-x}}(\ln(y)y^2-x)^2}{y\,(2\,\ln(y)+1)\,(\ln(y)y^2-x^2-x)}$ | $\frac{-\mathrm{e}^{\frac{x}{\ln(y)y^2-x}}}{(\ln(y)y^2-x)^2\,x}$ |
| 10 (161) | $\frac{\mathrm{e}^{-(xy^3-1)^{-1}}(xy^3-1)^2}{\mathrm{e}^yx^2y^6+3\,x^2y^5-2\,\mathrm{e}^yxy^3-3\,\mathrm{e}^yxy^2-3\,xy^2+\mathrm{e}^y}$ | $\frac{-\mathrm{e}^{(xy^3-1)^{-1}}}{(xy^3-1)^2}$ |

Fonte: A autora, 2020.

## 4.2 **Uso mais avançado do pacote** *LeapS1ode*

Nesta seção, vamos mostrar alguns comandos e parâmetros que dotam o pacote *LeapS1ode* de certas propriedades um tanto incomuns. Essas características especiais do pacote facilitam a resolução de 1EDOs bem mais complexas e, além disso, o tornam muito valioso na pesquisa em física e matemática.

- Na primeira subseção, mostramos como usar o pacote para resolver algumas 1EDOs bastante complexas.
- Na segunda subseção, em especial, destacamos o uso do parâmetro `Par` que permite que o pacote faça uma análise da região de integrabilidade da 1EDO.

### 4.2.1 Uso de comandos para driblar os problemas mais difíceis

Nas dez 1EDOs mostradas na seção 4.1, $\theta$ é uma função de apenas uma variável ($x$ ou $y$). Nestes casos, como mostrado na seção 2.4 (teoremas 2.2 e 2.3), podemos usar uma melhoria que nos permite calcular apenas um polinômio em vez de dois (algoritmo $FastL_S$). No entanto, devido a alguns detalhes técnicos, decidimos deixar a primeira parte do algoritmo $FastL_S$ nas mãos do usuário do pacote. Nesta seção, mostraremos como usar os comandos do pacote para obter a solução em casos mais difíceis.

**Exemplo 4.1**

Considere a 1EDO dada por:

$$\frac{dy}{dx} = \frac{\left(3\,x^5y^2 - 6\,\ln\left(\frac{x}{y}\right)x^4y + 3\,\left(\ln\left(\frac{x}{y}\right)\right)^2 x^3 - x^4y + x^3 + xy^2 - y\right)y}{x\left(x^4y + x^2y^3 - 2\,\ln\left(\frac{x}{y}\right)xy^2 + \left(\ln\left(\frac{x}{y}\right)\right)^2 y + x^3 - xy^2 - y\right)}. \tag{162}$$

Depois de carregar o pacote *LeapS1ode* e a 1EDO (162), podemos tentar (primeiro) encontrar a função-$S$ usando o comando `Sfunction`. No entanto, se digitarmos

```
[> t0 := time(): S1 := Sfunction(_1ode,Deg=7); time()-t0;
```

obtemos

$$S1 := \frac{1}{y} \tag{163}$$

11.812

após $\approx$ 12 segundos (e gastando 215 MB de memória). No entanto, se digitarmos

```
[> sol := Solde(_1ode,SF=S1);
```

A saida é

$$sol := \tag{164}$$

o que é intrigante porque, como a função-$S$ é simples, o método deveria ter encontrado a solução (se é que ela existe). O que está realmente acontecendo é que o seguinte invariante é encontrado para o campo vetorial associado $\chi$: $I = -\ln(x) + \ln(y) + z$. Agora, se atentarmos para o fato que a função $\theta$ presente na 1EDO é $\theta = \ln(x/y)$, fazendo a substituição $z \to \theta$ obteremos $I = -\ln(x) + \ln(y) + \ln(x/y) = 0$ que representa a identidade $\theta = \ln(x/y)$ (e, portanto, como não é uma resposta para o que procuramos, o programa a descarta). Podemos tentar usar o parâmetro `DegMNP` que é, na verdade, uma lista com três inteiros `[degM, degN, degP]` que dá ao comando `Sfunction` os graus dos polinômios que são candidatos para $M_0$, $N_0$ e $P_2$. Isso, no entanto, resulta em um grande consumo de recursos e em um 'estouro' da memória.

No entanto, ainda podemos ter sucesso usando o comando `Tr1ode` para transformar a 1EDO (162) em um 1EDO tal que $\theta$ seja uma função de apenas uma variável. Podemos fazer isso digitando (por exemplo)

```
[> _1odetr := Tr1ode(_1ode,TR=[x=x*y,y=y]);
```

e obter uma 1EDO transformada dada por:

$$\frac{dy}{dx} = \frac{-y\left(3\,x^5y^6 - 6\,x^4\ln\left(x\right)y^4 - x^4y^4 + 3\,x^3\ln\left(x\right)^2y^2 + x^3y^2 + xy^2 - 1\right)}{3x^6y^6 - 6x^5\ln(x)y^4 - 2x^5y^4 + 3x^4\ln(x)^2y^2 - x^3y^4 + 2\ln(x)x^2y^2 + 2x^2y^2 - x\ln(x)^2} \tag{165}$$

Portanto, na 1EDO transformada, temos que $\theta = \ln(x)$. Agora, podemos aplicar novamente o comando `Sfunction` (e, neste caso, usando o algoritmo $FastLs$)

```
[> t0:=time(): s1tr := Sfunction(_1odetr[1],Deg=11); time()-t0;
```

que nos retorna

$$s1tr := \frac{3x^5y^6 - 6x^4zy^4 - 2x^4y^4 + 3x^3y^2z^2 - x^2y^4 + 2zxy^2 + 2xy^2 - z^2}{y\left(x^3y^2 - 1\right)} \tag{166}$$

2.375

ou seja, o algoritmo encontra a função-$S$ em menos de 3 segundos (e consumindo apenas 30 MB). Neste ponto, só temos que digitar

```
[> soltr := Solde(_1odetr[1],ST=s1tr);
```

para obter a solução geral da 1EDO transformada (165) em $0,3$ segundos:

$$soltr := \frac{-xy^2\ln(x^3y^2-1) - xy^2\ln(y) + \ln(x)\ln(x^3y^2-1) + \ln(x)\ln(y) - 1}{xy^2 - \ln\left(x\right)} = \_C1 \tag{167}$$

e

```
[> Itr := Solde(_1odetr[1],ST=s1tr,II);
```

para obter a integral primeira do campo vetorial transformado:

$$Itr := \left[\frac{xy^2\ln(x^3y^2-1) + xy^2\ln(y) - z\ln(x^3y^2-1) - z\ln(y) + 1}{-xy^2 + z}, z = \ln(x)\right] \tag{168}$$

O comando `Tr1ode` retorna (como segunda saída) a transformação inversa. Então, digitando

```
[> itr := _1odetr[2]:
[> Ior := subs(itr,Itr[1]);
```

obtemos a integral primeira do campo vetorial associado à 1EDO (162):

$$Ior := \frac{xy\ln\left(\frac{x^3}{y} - 1\right) + xy\ln\left(y\right) - z\ln\left(\frac{x^3}{y} - 1\right) - z\ln\left(y\right) + 1}{z - yx} \tag{169}$$

Podemos checar esses resultados com

```
[> X := Xi(_1ode):
[> X(Ior);
```

$$0$$

Finalmente, fazendo a substituição $z \to \ln(x/y)$, encontramos a solução geral da 1EDO original (162):

```
[> DX := Dx(_1ode):
[> sol := subs(z=ln(x/y),Ior=_C1);
[> DX(lhs(sol));
```

$$sol := \frac{xy\ln\left(\frac{x^3}{y}-1\right)+xy\ln(y)-\ln(\frac{x}{y})\ln\left(\frac{x^3}{y}-1\right)-\ln(\frac{x}{y})\ln(y)+1}{\ln(\frac{x}{y})-yx} = \_C1 \tag{170}$$

$$0$$

**Exemplo 4.2**

Considere a 1EDO dada por:

$$\frac{dy}{dx} = \frac{-2\,\ln(y)\,x\,(\ln(y)\,x^2-\ln(y)-y)\,y}{(\ln(y))^2\,x^4+\ln(y)\,x^4-3\,\ln(y)\,x^2y-\ln(y)\,x^2-x^2y+\ln(y)\,y+2\,y^2}. \tag{171}$$

Carregando o pacote *LeapS1ode* e a 1EDO (171), se digitarmos

```
[> sol := Solde(_1ode,Deg=6);
```

vamos obter a seguinte saida:

$$sol := \tag{172}$$

Também acontece se aumentarmos o grau indicado pelo parâmetro `Deg`. E também acontece o mesmo se usarmos o parâmetro `DegMNP`. Contudo, se usarmos o comando `Sfunction` a função-$S$ aparece quase instantaneamente:

```
[> t0 := time(): s1 := Sfunction(_1ode,Deg=3); time()-t0;
```

$$s1 := -\frac{zx^2-y-z}{z^2x^4+zx^4-2\,zx^2y-x^2y-zx^2+y^2} \tag{173}$$

$$0.031$$

O que pode ter acontecido? Poderia ser um 'bug' no programa, ou a 1EDO não possui solução Liouvilliana, ou o comando do Maple (`dsolve`) não conseguiu resolver a 1EDO associada (ou não conseguiu resolver a 1EDO que representa o sistema característico da 1EDP associada). As dúvidas se dissipam rapidamente se digitarmos

```
[> _1edoa := Ode1a(_1ode,Deg=3);
```

$$\_1edoa := \frac{d}{dy}z\left(y\right) = \frac{z\left(y\right)x^2 - y - z\left(y\right)}{\left(z\left(y\right)\right)^2 x^4 + z\left(y\right)x^4 - 2\,z\left(y\right)x^2y - x^2y - z\left(y\right)x^2 + y^2} \tag{174}$$

seguido de

```
[> dsolve(_1edoa);
```

que resulta em uma saída vazia. Pode ser então que a 1EDO tenha uma solução Liouvilliana e (apenas) o `dsolve` não conseguiu resolver a 1EDO associada. Ou pode ser que a solução não seja Liouvilliana. Como saberemos? Bom, não temos como saber de maneira geral qual das alternativas é a correta. Contudo, usando a informação fornecida pelo programa (a função-$S$ associada) podemos (neste caso) verificar que a alternativa correta é a primeira. Como fazer isso? Como a integral primeira da 2EDO racional associada é solução da 1EDO associada `_1edoa` (174), então o operador de Darboux $D_A$ para a 1EDO (174) ($D_A \equiv (z^2x^4 + z\,x^4 - 2\,z\,x^2y - x^2y - z\,x^2 + y^2)\,\partial_y + (z\,x^2 - y - z)\,\partial_z$) é tal que $D_A(I) = 0$. E, portanto, os polinômios de Darboux presentes no fator integrante, também são polinômios de Darboux do operador $D_A$ (veja (51)). Assim, ao usarmos o MCI (método dos coeficientes indeterminados) para tentar determinar os PDs nos damos conta que o problema anteriormente muito complicado que seria determinar os PDs do operador $D_0 \equiv N_0\,\partial_x + z\,N_0\,\partial_y + M_0\,\partial_z$ (ou, equivalentemente, do operador $\chi \equiv f\,\partial_x + g\,\partial_y + h\,\partial_z$) que são PDs em três variáveis e de graus possivelmente altos, passa ao problema de determinarmos os PDs em duas variáveis e de graus possivelmente menores. Vamos ver como isso se dá na prática:

Em primeiro lugar digitamos

```
[> Na := denom(-s1):
[> Ma := numer(-s1):
[> Da := u->simplify(Na*diff(u,y)+Ma*diff(u,z),symbolic):
```

para definir o operador $D_A$. O grau máximo possível para os cofatores é $degQ_{MAX} = \max\{\deg(M_a) - 1, \deg(N_a) - 1\}$ e, assim, já começamos a perceber as vantagens: Os graus de $f$, $g$ e $h$ em $\{x, y, z\}$ são, respectivamente, dados por

```
[> fgh := Xi(_1edo,SF=s1,FGH):
[> f := fgh[1]; g := fgh[2]; h := fgh[3];
[> degree(f,{x,y,z}); degree(g,{x,y,z}); degree(h,{x,y,z});
```

$$f := x^4z^2 + x^4z - 3\,x^2yz - x^2y - zx^2 + 2\,y^2 + yz$$

$$g := -2\,zx\left(zx^2 - y - z\right)y$$

$$h := -2\,zx\left(zx^2 - y - z\right)$$

$$6$$

$$6$$

$$5$$

ao passo que os graus de $M_a$ e $N_a$ em $\{y, z\}$ são

```
[> Na; Ma;
[> degree(Na,{y,z}); degree(Ma,{y,z});
```

$$x^4z^2 + x^4z - 2\,x^2yz - x^2y - zx^2 + y^2$$

$$zx^2 - y - z$$

$$2$$

$$1$$

Por causa disso, os cofatores em relação ao operador $D_A$ têm grau máximo igual a 1. Este fato facilita muito as contas: Podemos começar buscando por polinômios de Darboux de grau 1. Digitando

```
[> P := Polg([y,z],1,p_,'pall');
[> Q := Polg([y,z],1,q_,'qall');
[> eqpq := {coeffs(collect(Da(P)-Q*P,[y,z],distributed),[y,z])};
[> solpq := [solve(eqpq,{op(pall),op(qall)})];
```

$$P := p_1\,y + p_2\,z + p_0$$

$$Q := q_1\,y + q_2\,z + q_0$$

$$eqpq := \left\{-q_0\,p_0, -q_1\,p_1 + p_1, p_1\,x^4 - q_2\,p_2, -2\,p_1\,x^2 - q_2\,p_1 - q_1\,p_2, -p_1\,x^2 - q_1\,p_0 - q_0\,p_1 - \right.$$

$$\left. p_2, p_1\,x^4 - p_1\,x^2 + p_2\,x^2 - q_2\,p_0 - q_0\,p_2 - p_2\right\}$$

$$solpq := [\{p_0 = 0, p_1 = 0, p_2 = 0, q_0 = q_0, q_1 = q_1, q_2 = q_2\},$$

$$\left\{p_0 = 0, p_1 = p_1, p_2 = -p_1\,x^2, q_0 = 0, q_1 = 1, q_2 = -x^2\right\},$$

$$\{p_0 = p_0, p_1 = 0, p_2 = 0, q_0 = 0, q_1 = 0, q_2 = 0\}]$$

Substituindo as soluções em $P$ e $Q$, vamos obter:

```
[> for i to nops([solpq]) do traperror(subs(solpq[i],[P,Q])) end do;
```

$$[0, q_1\,y + q_2\,z + q_0]$$

$$[-p_1\,\left(zx^2 - y\right), -zx^2 + y]$$

$$[p_0, 0]$$

A primeira e a terceira soluções são triviais, mas a segunda nos mostra um PD de grau 3 em três variáveis, mas que em relação ao operador $D_A$ é um PD de grau 1 em duas variáveis (motivo pelo qual o MCI o encontra quase instantaneamente). De posse desse PD podemos encontrar um fator integrante para o campo vetorial polinomial $\chi$:

$$\frac{2\,z\,x\mathrm{e}^{\left(zx^2-y\right)^{-1}}}{\left(zx^2-y\right)^2}. \tag{175}$$

Depois, podemos usar o fator integrante para encontrar a integral primeira[21] de $\chi$:

$$\mathrm{e}^{\left(zx^2-y\right)^{-1}} z + Ei\left(1, -\left(zx^2-y\right)^{-1}\right). \tag{176}$$

A partir da integral primeira, a solução geral da 1EDO (171) segue diretamente ($z \to \theta$):

$$\mathrm{e}^{\left(\ln(y)x^2-y\right)^{-1}} \ln(y) + Ei\left(1, -\left(\ln(y)\,x^2-y\right)^{-1}\right) = C. \tag{177}$$

**Observação 4.4** *Algumas considerações:*

- *Estes dois exemplos nos dão uma pequena amostra de como usar de forma mais avançada os comandos que fornecem os passos intermediários do procedimento como um todo:*

  *No primeiro exemplo (exemplo 4.1) vimos como usar o comando* `Tr1ode` *para mudar uma 1EDO que possuia uma função $\theta$ de duas variáveis em uma com $\theta$ mais palatável ($\theta = \ln(x)$). Essa mudança possibilitou a resolução do problema uma vez que o algoritmo que poderia resolver a 1EDO na forma original (materializado no uso do parâmetro* `DegMNP`*) 'estoura' a memória quando é acionado.*

  *No segundo (exemplo 4.2), o problema era de outra natureza: a função-S foi encontrada e era realmente a função-S que proporcionaria a solução da 1EDO. Contudo, o comando do Maple responsável pela resolução de equações diferenciais (*`dsolve`*) não foi capaz de resolver a 1EDO associada $dz/dy = -S_1(x, y, z)$ (etapa essencial no método da função-S). Porém, como essa parte interna está 'visível', isto é, materializada na saída dos comandos que extraem do programa os passos intermediários do processo, foi possível usar a função-S para resolver o problema aplicando uma abordagem DPS à 1EDO associada. É importante frizar que essa*

[21] O cálculo do fator integrante e da integral primeira foi realizado usando a abordagem Darbouxiana descrita em (36, 38, 50).

*abordagem DPS, se fosse aplicada ao problema original (como foi feito em (1)) não obteria sucesso (prático) pois estaríamos buscando polinômios de Darboux de grau 3 em três variáveis e a memória do computador estouraria após um longo tempo de espera.*

- *Outra reflexão importante, de certa forma também relacionada com a consideração anterior sobre o exemplo 4.2, é a de que* **não veríamos** *a possibilidade de resolver a 1EDO (171) se os comandos* `Sfunction` *e* `Ode1a` *não tivessem produzido a 1EDO associada (174) em relação à qual a variável $x$ é apenas uma constante e, portanto, nos permite usar o MCI (método dos coeficientes indeterminados) para determinar os polinômios de Darboux de forma prática (uma vez que o operador $D_A$ é um operador polinomial em apenas duas variáveis e, além disso, em relação a ele, o polinômio de Darboux relevante para a construção do fator integrante tem grau 1 – isto sem mencionar o grau máximo dos cofatores que passa de 5 para 1).*

- *Ainda em relação ao tema da observação anterior, podemos notar que a parte da resolução da 1EDO (171) que usa o operador de Darboux $D_A$ da 1EDO associada (174) para calcular os polinômios de Darboux do campo vetorial associado, é uma maneira nova de realizar o processo de resolver 1EDOs deste tipo (do tipo que nos propomos a resolver neste trabalho). Isto porque não é a abordagem DPS padrão e nem é o método da função-S adaptado que construímos. É uma espécie de método híbrido que pode ser desenvolvido futuramente.*

- *Finalmente, ainda em relação às considerações anteriores mas, mais especificamente, em relação à anterior, um fenômeno parecido ocorreu em um de nossos testes do método / programa: percebemos que, em todas as 1EDOs $\in L_S$ (veja a definição 2.2), podemos nos aproveitar do fato que, em relação ao campo vetorial $\chi$, as variáveis $(x, y, z)$ têm o mesmo status. Esse detalhe nos permitiu construir um procedimento linear para calcular a função-S associada. Para descrever este procedimento escrevemos um novo capítulo (o próximo).*

### 4.2.2 Uma análise de regiões de integrabilidade

Além dos recursos mostrados acima, apresentamos nesta seção uma maneira muito útil de aplicar o pacote *LeapS1ode*: o parâmetro `Par`. Quando estamos estudando uma 1EDO que apresenta parâmetros (representando grandezas físicas, por exemplo), podemos usar o parâmetro `Par` nas entradas do comando `Sfunction`. Assim, se o comando `Sfunction` não conseguiu encontrar uma função-$S$ para valores genéricos dos parâmetros, a entrada `Par =` $\{\alpha,\ \beta,\ \ldots\ \}$ diz ao comando para tentar encontrar uma função-$S$

para alguma relação funcional (algébrica) entre eles, ou seja, o método subjacente empregado[22] (ao fazer uso de `Par`) nos traz a possibilidade de uma análise da região de integrabilidade dos parâmetros da 1EDO.

**Exemplo 4.3**

Para ver um caso real de uso do pacote na pesquisa de regiões de integrabilidade em termos de funções Liouvillianas (soluções em forma fechada), vamos tomar como exemplo uma 1EDO obtida a partir do estudo de 2EDOs que modelam fenômenos de oscilações não lineares. No nosso caso a 1EDO que vamos apresentar aparece na redução de uma equação de Levinson-Smith[23] (53)

$$\ddot{x} + F(x, \dot{x})\,\dot{x} + G(x) = 0, \tag{178}$$

uma 2EDO que é uma generalização da equação de Liénard[24] (55)

$$\ddot{x} + F(x)\,\dot{x} + G(x) = 0. \tag{179}$$

Como a 2EDO (178) não depende explicitamente do tempo, podemos fazer a seguinte transformação de coordenadas para reduzir sua ordem:

$$\begin{cases} x = x, \\ \dot{x} = y, \\ \ddot{x} = \frac{dy}{dx}\, y. \end{cases} \tag{180}$$

A aplicação desta tranformação à 2EDO (178) leva à 1EDO dada por

$$\frac{dy}{dx} = -\frac{F(x, y)\, y + G(x)}{y}. \tag{181}$$

Podemos agora, a partir de funções $F$ e $G$ bem genéricas, tentar obter casos integráveis em termos de funções Liouvilianas (i.e., soluções em forma fechada). Vamos analisar o seguinte caso:

$$\frac{dy}{dx} = \frac{-fx^2y\mathrm{e}^x - ex^2\mathrm{e}^x - x\mathrm{e}^x yb - y^2c - dx - ya}{y} \tag{182}$$

[22] Estamos nos referindo aqui à resolução de sistemas algébricos de coeficientes a determinar.

[23] A equação de Levinson-Smith possui aplicações em astrofísica (54).

[24] Existe uma vasta literatura sobre o uso desta equação para modelar diversos fenômenos: variando de circuitos elétricos, estudo de batimentos cardíacos, atividade de neurônios, cinética química a fenômenos de turbulência (56, 57, 58, 59).

Usando `Sfunction` com o uso do parâmetro `Par` como assignado abaixo

```
[> _1ode := diff(y(x),x)= \cdots:
[> ss1 := [Sfunction(_1ode,Par={b,d,e},DegMNP=[6,4,5])]:
[> for i from 3 by 3 to nops(ss1) do
   for j to nops(ss1[i]) do
   print([i,j],____________________,traperror(factor(subs(ss1[i][j],
   ss1[i-2]))),subs(ss1[i][j],[‘b=‘,b,‘c=‘,c,‘d=‘,d,‘e=‘,e,‘f=‘,f]));
   trys[i,j]:=traperror(factor(subs(ss1[i][j],ss1[i-1][1]/ss1[i-1][2])));
   print(trys[i,j]);
   end do;
   end do;
```

vamos obter[25]

$$\vdots$$

$$[3,4],\ \_\_\_\_\_,\ -\frac{cz}{cy+a}\ [‘b=‘,b,‘c=‘,c,‘d=‘,0,‘e=‘,0,‘f=‘,f]$$

$$\frac{\left(cfx^2z+bcxz+c^2y+ac+cy+cz+a\right)z}{cy+a}$$

$$\vdots$$

Usando a região determinada na 1EDO (182) obtemos finalmente

```
[> new1ode := subs(ss1[3][4],_1ode);
```

$$\frac{d}{dx}y\left(x\right)=\frac{-fx^2y\left(x\right)\mathrm{e}^x-x\mathrm{e}^xy\left(x\right)b-\left(y\left(x\right)\right)^2c-y\left(x\right)a}{y\left(x\right)} \tag{183}$$

que pode ser resolvida com

```
[> s1 := ss1[1];
[> new1ode := subs(ss1[3][4],_1ode);
[> Solde(new1ode,SF=s1):
```

$$-\frac{cz}{cy+a}$$

$$\frac{d}{dx}y\left(x\right)=\frac{-fx^2y\left(x\right)\mathrm{e}^x-x\mathrm{e}^xy\left(x\right)b-\left(y\left(x\right)\right)^2c-y\left(x\right)a}{y\left(x\right)}$$

Arrumando a solução obtemos

$$y=\left(-\frac{f\,\mathrm{e}^x\left(\left(c+1\right)^2x^2-2\left(c+1\right)x+2\right)}{\left(c+1\right)^3}-\frac{b\,\mathrm{e}^x\left(\left(c+1\right)x-1\right)}{\left(c+1\right)^2}-\frac{a}{c}+\_C1\right)$$

[25] Omitimos a saída completa pois ela é muito longa.

Substituindo $y$ por $z$ e $x$ por $y$, vamos obter a integral primeira da equação de Levinson-Smith (para as regiões de parâmetros que encontramos):

$$I = z + \left( \frac{f\,\mathrm{e}^{y}\left((c+1)^{2}\,y^{2} - 2\,(c+1)\,y + 2\right)}{(c+1)^{3}} + \frac{b\,\mathrm{e}^{y}\left((c+1)\,y - 1\right)}{(c+1)^{2}} + \frac{a}{c} \right)$$

## 4.3 $Lsolv \times FastL_S \times$ Abordagem DPS

Nesta seção, fazemos uma análise comparativa entre os dois algoritmos $Lsolv$ e $FastL_S$ e também entre o pacote $LeapS1ode$ e uma abordagem DPS padrão[26]:

- Na primeira subseção, comparamos o método com ele mesmo, uma vez que estamos comparando o desempenho dos algoritmos $Lsolv$ e $FastL_S$. Mais especificamente, o que fazemos é uma análise do funcionamento e escopo dos algoritmos.
- Na segunda, fazemos uma comparação do desempenho do pacote $LeapS1ode$ e um procedimento que usa o MCI e a abordagem DPS raiz (ou seja, usando o cálculo dos PDs para construir um fator integrante).

### 4.3.1 Analisando os algoritmos $Lsolv$ e $FastL_S$

Nesta seção, comentamos sobre as diferenças de atuação e abrangência dos algoritmos $Lsolv$ e $FastL_S$ quando os aplicamos (na prática) a 1EDOs. Comparamos o tempo gasto e a memória consumida pelos dois algoritmos e salientamos as vantagens e desvantagens de cada um.

**Exemplo 4.4**

Considere a seguinte 1EDO (A 1EDO do exemplo 2.1):

$$\frac{dy}{dx} = \frac{e^{x}x^{3}y^{2} + e^{x}x^{2}y^{2} + 2e^{x}x^{2}y + e^{x}xy + e^{x}x + y^{2} + e^{x}}{x^{2}y^{2} + e^{x}x^{2} + xy + 1}. \tag{184}$$

Para usar $Lsolv$, vamos digitar

```
[> _1ode := diff(y(x),x)= ... :
[>  t0 := time(): Lsol := Solde(_1ode,DegMNP=[4,5,5]); time()-t0;
```

[26] Com *abordagem padrão* estamos nos referindo ao uso do MCI para o cálculo dos PDs.

que nos retorna a saída

$$Lsol := \mathrm{e}^{(xy+1)^{-1}} (\mathrm{e}^{x} x - y) = \_C1 \tag{185}$$

0.627

Para usar $FastL_S$, digitamos

```
[>  t0 := time(): Fsol := Solde(_1ode,Deg=5); time()-t0;
```

que resulta em

$$Fsol := \mathrm{e}^{(xy+1)^{-1}} (\mathrm{e}^{x} x - y) = \_C1 \tag{186}$$

0.109

Uma das partes dos algoritmos que é responsável por uma boa parcela do tempo de CPU gasto com a resolução está implementada na rotina `Sfunction`. Podemos testar os gastos comparativos digitando

```
[> t0 := time(): sl := Sfunction(_1ode,DegMNP=[4,5,5]); time()-t0;
[> t0 := time(): sf := Sfunction(_1ode,Deg=5); time()-t0;
```

que resulta na saída

$$sl := -\frac{x^2y^2 + zx^2 + xy + 1}{x\,(x^2y^2 + 2\,xy + 1)} \tag{187}$$

0.344

$$sf := -\frac{x^2y^2 + zx^2 + xy + 1}{x\,(xy + 1)^2} \tag{188}$$

0.046

Aqui podemos ver a diferença de tempos entre os algoritmos como aproximadamente um fator 6. Vamos examinar se a mesma relação se mantém em uma 1EDO mais complexa.

**Exemplo 4.5**

Considere agora a 1EDO (156):

$$\frac{dy}{dx} = -\frac{4\,x^3y^6 + 8\,x^4y^3 + 4\,x^5 - x^4 + \cos(y)}{(y^6 + 2\,xy^3 + x^2)\sin(y) - 3\,x^4y^2 + 3\,\cos(y)\,y^2}. \tag{189}$$

Começando com $Lsolv$, vamos digitar

```
[> _1ode := diff(y(x),x)= ... :
[> t0 := time(): Lsol := Solde(_1ode,DegMNP=[10,8,8]); time()-t0;
```

Decorridos 90 segungos (e com um consumo de 370 MB) interrompemos o comando. Usando $FastL_S$ temos

```
[> t0 := time(): Fsol := Solde(_1ode,Deg=7); time()-t0;
```

que resulta em

$$Fsol := \ln\left(-2\,\mathrm{e}^{iy}x^4 + \mathrm{e}^{2\,iy} + 1\right) - \frac{i\,(y^4 + xy + i)}{y^3 + x} = \_C1 \tag{190}$$

3.578

Podemos ver neste segundo exemplo que, a medida que o grau dos polinômios que compõe a função-$S$ aumenta, a diferença parece aumentar de maneira não linear. Essa variação pode ter diferentes causas e, devido ao fato que o comando `solve` do Maple é o responsável pela resolução do sistema algébrico de equações que resulta quando aplicamos o MCI, não podemos determinar com precisão como aumenta a complexidade do algoritmo.

**Exemplo 4.6**

Considere agora a 1EDO:

$$\frac{dy}{dx} = -\frac{2\left(\mathrm{e}^{x^2y}\right)^3 xy + (4\,x^2y - 2\,xy)\left(\mathrm{e}^{x^2y}\right)^2 + (2\,x^3y + 2\,xy^3 - 1)\,\mathrm{e}^{x^2y} + y^2}{\left(\mathrm{e}^{x^2y}\right)^3 x^2 + (2\,x^3 - x^2 - 2\,y)\left(\mathrm{e}^{x^2y}\right)^2 + (x^4 + x^2y^2 - 4\,xy)\,\mathrm{e}^{x^2y} - 2\,x^2y} \tag{191}$$

Começando com $Lsolv$ e digitando

```
[> _1ode := diff(y(x),x)= ... :
[> t0 := time(): Lsol := Solde(_1ode,DegMNP=[4,2,4]); time()-t0;
```

obtemos

$$Lsol := -\mathrm{e}^{\left(\mathrm{e}^{x^2y+x}\right)^{-1}}\left(-y^2 + \mathrm{e}^{x^2y}\right) = \_C1 \tag{192}$$

0.094

Para usar $FastL_S$ temos que, em primeiro lugar, transformar a 1EDO em uma outra tal que $\theta$ seja função de uma variável apenas. Podemos fazer isso digitantdo

```
[> otr := Tr1ode(_1ode,TR=[x=x,y=y/x^2]):_1odetr := otr[1]; tri := otr[2];
```

que resulta em

$$\_1odetr := \frac{d}{dx}y\,(x) = \tag{193}$$

$$-\frac{-\mathrm{e}^{y(x)}x^5+4\,\mathrm{e}^{2\,y(x)}y\,(x)^2+8\,\mathrm{e}^{y(x)}xy\,(x)^2+4\,x^2y\,(x)^2+x\,y\,(x)^2}{x\left(\mathrm{e}^{3y(x)}x^4+2\mathrm{e}^{2y(x)}x^5+\mathrm{e}^{y(x)}x^6-\mathrm{e}^{2\,y(x)}x^4-2\mathrm{e}^{2\,y(x)}y(x)-4\mathrm{e}^{y(x)}xy(x)+\mathrm{e}^{y(x)}y(x)^2-2x^2y(x)\right)}$$

$$tri := [x = x, y = x^2y]$$

Em seguida, digitando

```
[> t0 := time(): Fsoltr := Solde(_1odetr,Deg=4); time()-t0;
[> Fsol := simplify(subs(tri,Fsoltr));
```

vamos obter a solução da 1EDO transformada ($Fsoltr$) e, após aplicarmos a tranformação inversa, a solução geral da 1EDO original (191) ($Fsol$):

$$Fsoltr := -\frac{\mathrm{e}^{\left(\mathrm{e}^y+x\right)^{-1}}\left(x^4\mathrm{e}^y-y^2\right)}{x^4} = \_C1 \tag{194}$$

0.391

$$Fsol := -\mathrm{e}^{\left(\mathrm{e}^{x^2y}+x\right)^{-1}}\left(-y^2+\mathrm{e}^{x^2y}\right) = \_C1$$

Neste terceiro exemplo, vemos que o algoritmo $Lsolv$ foi mais eficiente que o $FastL_S$. Provavelmente, porque a 1EDO (191) possuia uma função $\theta$ de duas variáveis o que inviabiliza a aplicação direta do algoritmo $FastL_S$. Contudo, mesmo com a diferença de tempo de CPU, o tempo gasto por ambos foi muito pequeno (relativamente).

**Exemplo 4.7**

Considere agora a 1EDO:

$$\frac{dy}{dx} = -\frac{2\,x^2y^4-2\,\ln(x)\,xy^2-xy^2+(\ln(x))^2}{2\,x^3y^3} \tag{195}$$

Este exemplo, $Lsolv$ não consegue resolver: se digitarmos

```
[> _1ode := diff(y(x),x)= ... :
[> t0 := time(): Lsol := Solde(_1ode,DegMNP=[7,6,6]); time()-t0;
```

o resultado é

$$Lsol :=$$

3.312

E, a princípio, $FastL_S$ também não consegue:

```
[> _1ode := diff(y(x),x)= \cdots:
[> t0 := time(): Fsol := Solde(_1ode,Deg=6); time()-t0;
```

$$Fsol :=$$

$$0.828$$

Contudo, se fizermos a transformação

```
[> otr := Tr1ode(_1ode,TR=[x=exp(x),y=y]):_1odetr := otr[1]; tri := otr[2];
```

que resulta na 1EDO

$$\_1odetr := \frac{d}{dx}y\left(x\right) = -\frac{\left(2\,\mathrm{e}^{2\,x}\left(y\left(x\right)\right)^{4} - 2\,x\mathrm{e}^{x}\left(y\left(x\right)\right)^{2} - \left(y\left(x\right)\right)^{2}\mathrm{e}^{x} + x^{2}\right)\mathrm{e}^{-2\,x}}{2\,y\left(x\right)^{3}} \tag{196}$$

$$tri := [x = \ln\left(x\right), y = y]$$

E, depois,se tentarmos novamente, digitando

```
[> t0 := time(): Fsoltr := Solde(_1odetr,Deg=5); time()-t0;
```

obtemos

$$Fsoltr := \left(Ei\left(1, \left(y^{2}\mathrm{e}^{x} - x\right)^{-1}\right)\mathrm{e}^{\left(y^{2}\mathrm{e}^{x} - x\right)^{-1}} + x\right)\mathrm{e}^{-\left(y^{2}\mathrm{e}^{x} - x\right)^{-1}} = \_C1 \tag{197}$$

$$0.110$$

que é a solução da 1EDO transformada. Neste ponto só temos que aplicar a transformação inversa em $Fsoltr$

```
[> Fsol := simplify(subs(tri,Fsoltr));
```

$$Fsol := \left(Ei\left(1, -\left(\ln\left(x\right) - y^{2}x\right)^{-1}\right)\mathrm{e}^{-\left(\ln(x) - y^{2}x\right)^{-1}} + \ln\left(x\right)\right)\mathrm{e}^{\left(\ln(x) - y^{2}x\right)^{-1}} = \_C1$$

Neste quarto exemplo, vemos que, curiosamente, após aplicarmos a transformação $T = \{x = \mathrm{e}^{x}, y = y\}$, o algoritmo $FastL_S$ conseguiu resolver a 1EDO transformada e, dessa maneira, resolver a 1EDO (195) que já possuia uma função $\theta$ que era função de uma variável apenas. Também, inesperadamente, o algoritmo *Lsolv* não encontrou a resposta.

**Observação 4.5** *Alguns comentários:*

- *Se tomarmos como base apenas os quatro exemplos mostrados nesta seção, pode parecer que o algoritmo Lsolv não é muito útil uma vez que dispomos do algoritmo $FastL_S$. Apenas em um dos quatro exemplos Lsolv resolveu a 1EDO em um tempo inferior ao gasto pelo algoritmo $FastL_S$ e em dois deles não conseguiu achar a resposta. Isto não é apenas porque os exemplos são poucos: se ampliarmos muito o conjunto de exemplos, o padrão observado vai se manter.*

- *Porém, em virtude do algoritmo Lsolv calcular $M_0$, $N_0$ e $P_2$, ele nos permite usar o parâmetro* `Par` *que é muito importante na pesquisa de regiões de integrabilidade.*

- *Alguns detalhes ainda nos escapam como, por exemplo, a 1EDO*

$$\frac{dy}{dx} = -\frac{\ln(x)^3 + (-2\,y^2 - 1)\ln(x)^2 + (y^4 + 2\,y^2 - x + 1)\ln(x) - y^4 + xy^2}{2\,yx\,((x-1)\ln(x) - xy^2)}. \qquad (198)$$

*Essa 1EDO ilude os dois algoritmos (não sabemos ainda a razão). $FastL_S$ (com* `Deg=6`*) desiste em 0,2 segundos e, se tentarmos uma estratégia como a do exemplo 4.7, o algoritmo desiste, aproximadamente, no mesmo tempo. Com o algoritmo Lsolv se dá o mesmo (também em aproximadamente 0,2 segundos). Contudo, ela é facilmente resolvida pelo algoritmo linear apresentado no capítulo 5.*

### 4.3.2 *LeapS1ode* comparado ao DPS padrão

Nesta seção, fazemos uma comparação entre o pacote *LeapS1ode* e uma abordagem Darbouxiana padrão. Antes de começar, é importante ressaltar que o que estamos comparando aqui é, basicamente, a ideia original com a ideia atualizada. Explicando melhor: ao surgir a ideia de substituir a função $\theta$ por uma nova variável ($\theta \to z$), implicando na troca de uma 1EDO com uma função elementar ($y' = \phi(x, y, \theta(x, y))$, $\phi \in \mathbb{C}(x, y, \theta)$) por um campo vetorial polinomial em três variáveis ($\chi \equiv f\,\partial_x + g\,\partial_y + h\,\partial_z,\ f, g, h \in \mathbb{C}[x, y, z]$), o que estava em pauta era, em primeiro lugar, obter uma vantagem teórica em relação ao que fizemos em (1) (pois o procedimento que desenvolvemos em (1) não estava posto em uma base teórica sólida). Logo depois, surgiu a ideia de adaptar o método da função-$S$ para tratar esses casos, fugindo assim, da tarefa inglória de determinar polinômios de Darboux em três variáveis. O que eu estou tentando deixar claro é que a ideia principal por trás do trabalho desta tese começou justamente pela possibilidade de adaptar a técnica (e também os programas) desenvolvida em (3) (o método da função-$S$) com a ideia original descrita acima. Isto porque, na prática, o cômputo de PDs em três variáveis é muito custoso (computacionalmente) se usarmos uma abordagem DPS padrão (isto é, se usarmos o MCI). Dessa maneira, essa 'comparação' é mais qualitativa, uma vez que, para a grande maioria das 1EDOs com funções que apresentamos neste trabalho, a abordagem DPS padrão não consegue achar os PDs relevantes para a construção de um fator integrante.

Para ilustrar o que dissemos vamos a dois pequenos exemplos:

**Exemplo 4.8**

Considere a seguinte 1EDO (154) (a terceira 1EDO da tabela apresentada na seção 4.1 e

a mais simples das dez):

$$\frac{dy}{dx} = -\frac{(2\,x^3y^4 - 4\,x^2y^2 - x^2 + \ln(y) + 2\,x)\,y}{-2\,x^4y^2 - x^2y^4 + 2\,\ln(y)\,x^2y^2 + 2\,xy^2 - 1} \tag{199}$$

**Usando** *LeapS*1*ode*:

Recordando, o pacote encontra a solução em 0,2 segundos (e consumindo menos de 4 MB de memória).

**Usando a abordagem DPS**: Digitando

```
[> _1ode := diff(y(x),x)= ... :
[> X := Xi(_1ode):
[> PP := Polg([x,y,z],3,a_,'aall'):
[> QQ := Polg([x,y,z],6,b_,'ball'):
[> t1 := time(): eqs := {coeffs(collect(X(PP)-QQ*PP,
   [x,y,z],distributed),[x,y,z])}:
[> solpq := [solve(eqs,{op(aall),op(ball)})]: time()-t1;
[> for i to nops(solpq) do traperror(subs(solpq[i],[PP,QQ])) end do;
```

que nos retorna a saída (eliminamos as triviais):

$$8.094$$

$$[a\_4\ \left(x^2 - z\right), 4\,y^2x^3 + 1]$$

O método DPS gasta $\approx 8$ segundos (e consome 40 MB de memória) para determinar o polinômio de Darboux que será usado para construir um fator integrante.

**Exemplo 4.9**

Considere a seguinte 1EDO (155) (a quarta 1EDO da tabela apresentada na seção 4.1):

$$\frac{dy}{dx} = -\frac{3\,\left(\mathrm{e}^y\right)^2x^4y^2 - 7\,\mathrm{e}^yx^3y + 3\,x^2 - y}{x\left(\left(\mathrm{e}^y\right)^2x^4y^2 - 3\,\mathrm{e}^yx^3y - x^3\mathrm{e}^y + x^2 - y - 1\right)} \tag{200}$$

**Usando** *LeapS*1*ode*:

O pacote encontra a solução em 0,6 segundos (e consumindo 33 MB de memória).

**Usando a abordagem DPS**: Digitando

```
[> _1ode := diff(y(x),x)= ... :
[> X := Xi(_1ode):
[> PP := Polg([x,y,z],4,a_,'aall'):
[> QQ := Polg([x,y,z],8,b_,'ball'):
```

```
[> t1 := time(): eqs := {coeffs(collect(X(PP)-QQ*PP,
   [x,y,z],distributed),[x,y,z])}:
[> solpq := [solve(eqs,{op(aall),op(ball)})]: time()-t1;
[> for i to nops(solpq) do traperror(subs(solpq[i],[PP,QQ])) end do;
```

Desligamos o comando após 20 minutos (com um consumo de 275 MB de memória).

**Observação 4.6** *Como já mencionamos, a comparação não é muito 'justa' uma vez que o 'método da função-S adaptado' foi elaborado justamente para lidar com casos de 1EDOs nas quais os PDs presentes no fator integrante possuiam graus muito altos e / ou as simetrias de Lie eram muito complicadas. Desse modo, não é tão surpreendente assim o 'excelente resultado' obtido. De toda maneira, o algoritmo* $FastL_S$ *nos surpreendeu bastante positivamente (por causa da possibilidade de computar apenas um polinômio ao invés de dois).*

No próximo capítulo apresentamos um resultado realmente surpreendente: um **algoritmo linear** para computar a função-$S$.

# 5 UM ALGORITMO LINEAR

Neste capítulo, apresentamos um resultado que nos permite ampliar, consideravelmente, a eficiência do algoritmo $FastL_S$. Para a classe de 1EDOs $L_S$ (veja a definição 2.2 no capítulo 2) para o qual o algoritmo $FastL_S$ é aplicável, desenvolvemos um algoritmo linear para a obtenção da função-$S$ associada ao campo vetorial polinomial $\chi$ (e à 2EDO 45). Este capítulo terá a seguinte estrutura:

- Na primeira seção, mostramos uma 1EDO na qual, no processo do cálculo da função-$S$ associada ao campo vetorial polinomial $\chi$, notamos que (por um feliz acaso) o coeficiente do termo não linear (o coeficiente do termo $S^2$) na equação (122)

$$\chi(S) = S^2\left(\frac{fg_z - gf_z}{f}\right) + S\left(\frac{gf_y - fg_y + fh_z - hf_z}{f}\right) - \left(\frac{fh_y - hf_y}{f}\right).$$

  era zero.

- Na segunda seção, mostramos que o 'feliz acaso' ocorrido no exemplo mostrado na seção anterior poderia ser 'induzido' a acontecer no processo de solução de todas as 1EDOs que fazem parte da classe $L_S$: apresentamos um resultado (teorema 5.1) que nos permite construir um novo algoritmo (**linear**) para determinar a função-$S$ associada ao campo vetorial $\chi$.

- Na terceira seção fazemos uma primeira análise da eficiência do algoritmo desenvolvido.

## 5.1 Um exemplo 'piloto'

Os resultados que vamos apresentar nesta seção surgiram no estudo da eficiência do algoritmo $FastL_S$. O exemplo que vamos apresentar não possui a complicação (PDs com graus muito altos) de um caso no qual precisaríamos aplicar o método linear para resolver o problema, contudo, a ideia central fica preservada.

Considere a seguinte 1EDO:

$$\frac{dy}{dx} = \frac{4\ln(x)^3x^4y^6+(x^4y^6-8\,x^5y^3)\ln(x)^2+(4\,x^6+x^5-2\,x^5y^3-x^4y^3)\ln(x)+x^6+y^5-xy^2}{yx\left((2\,y^6+3\,x^4y)\left(\ln\left(x\right)\right)^2+(-4\,xy^3-3\,y^3)\ln\left(x\right)+2\,x^2\right)} \tag{201}$$

Após abrir uma seção de maple, carregar o pacote *LeapS1ode* e entrar com a 1EDO (201), usamos o comando `Sfunction` para calcular a função-$S$:

```
[> t0 := time(): s1 := Sfunction(_1ode,Deg=12); time()-t0;
```

$$s1 := -\frac{y\left(2\,y^6z^2+3\,x^4yz^2-4\,xy^3z-3\,y^3z+2\,x^2\right)}{x^4y^6z^2-2\,x^5y^3z-x^4y^3z+x^6+y^5} \tag{202}$$

$$4.594$$

Obtivemos a resposta acima em aproximadamente 5 segundos com um gasto de aproximadamente 34MB de memória. Calculamos $f$, $g$ e $h$, os polinômios $P_1$, $P_2$ e $P_3$, e o $\Phi$ da 2EDO associada:

```
[> fgh := Xi(_1ode,FGH):
[> ppp := Ode2(_1ode,SF=s1,PS):
[> mn := Ode2(_1ode,SF=s1):
[> f := fgh[1]; g := fgh[2]; h := fgh[3];
[> P1 := ppp[1]; P2 := ppp[2]; P3 := ppp[3];
[> Phi := factor(mn[1]/mn[2]);
```

$$f := xy\left(2\,y^6z^2+3\,x^4yz^2-4\,xy^3z-3\,y^3z+2\,x^2\right)$$

$$g := 4\,z^3x^4y^6+x^4y^6z^2-8\,z^2x^5y^3-2\,x^5y^3z-x^4y^3z+4\,zx^6+x^6+x^5z+y^5-xy^2$$

$$h := y\left(2\,y^6z^2+3\,x^4yz^2-4\,xy^3z-3\,y^3z+2\,x^2\right)$$

$$P1 := 4\,x^3y^6z^3-8\,x^4y^3z^2+4\,x^5z+x^4z-y^2$$

$$P2 := -y\left(2\,y^6z^2+3\,x^4yz^2-4\,xy^3z-3\,y^3z+2\,x^2\right)$$

$$P3 := x^4y^6z^2-2\,x^5y^3z-x^4y^3z+x^6+y^5$$

$$\Phi := -\frac{4x^3y^6z^3-2y^7z^3-8x^4y^3z^2-3x^4y^2z^3+4xy^4z^2+4x^5z+3\,y^4z^2+x^4z-2x^2yz-y^2}{x^4y^6z^2-2\,x^5y^3z-x^4y^3z+x^6+y^5}$$

Podemos observar que, como já predito no caso $(ii)$ do teorema 2.2, $h|P_2$. Como a função $S_1 = P_2/P_3$, foi necessário determinar o denominador da função $S_1$. O que estávamos querendo neste ponto do processo era tirar proveito do fato que as variáveis $\{x, y, z\}$ têm o mesmo status em relação ao campo vetorial $\chi$. Assim, pensamos em melhorar o algoritmo 'trocando' as variáveis para que o $N_0$ (que é igual à $P_3$) fosse o polinômio conhecido. Explicando melhor:

- A equação usada para determinar a função-$S$ ($S_1$) é:

$$\chi(S_1) = {S_1}^2\left(\frac{fg_z-gf_z}{f}\right)+S_1\left(\frac{gf_y-fg_y+fh_z-hf_z}{f}\right)-\left(\frac{fh_y-hf_y}{f}\right).$$

- $S_1 = \frac{P_2}{P_3}$ e, além disso, $h|P_2$ ($h = -P_2$) e $P_3 = N_0$. Assim, conhecemos $P_2$ ($=-h$) e não $P_3$ ($=N_0$).

- Como as variáveis têm o mesmo status, podemos fazer uma transformação ($\tilde{T}$) de variáveis para que o 'novo' polinômio $\tilde{P}_3$ seja o polinômio conhecido. Como $P_2$ é o polinômio que conhecemos, devemos exigir que $\tilde{P}_3 = \tilde{T}(P_2)$. Mas $P_2$ é o polinômio associado à derivada da integral primeira $I$ em relação à $y$ ($I_y = R\,P_2$) e, dessa maneira devemos exigir que $y = \tilde{z}$ implicando que

$$\tilde{T}(\partial_y(I)) = \partial_{\tilde{z}}(\tilde{T}(I)) = \tilde{I}_{\tilde{z}} = \tilde{T}(\underbrace{R\,P_2}_{\partial_y(I)}) = \tilde{R}\,\tilde{T}(P_2) = \tilde{R}\,\tilde{P}_3$$

- Temos duas tranformações de variáveis que satisfazem à exigência do item anterior:

$$\tilde{T_1} = \begin{cases} x = \tilde{y} \\ y = \tilde{z} \\ z = \tilde{x} \end{cases}$$

ou

$$\tilde{T_2} = \begin{cases} x = \tilde{x} \\ y = \tilde{z} \\ z = \tilde{y} \end{cases}$$

- Escolhemos $\tilde{T} = \tilde{T_1}$ e, portanto, como $z = \tilde{x}$, temos que

$$\tilde{T}(h\,\partial_z) = \tilde{T}(h)\,\partial_{\tilde{x}} = \tilde{f}\,\partial_{\tilde{x}} \;\;\Rightarrow\;\; \tilde{T}(h) = \tilde{f}$$

Portanto, $\tilde{T}$ 'transforma' $h$ ($=-P_2$) em $\tilde{f}$, isto é, o polinômio $h$ (que multiplica $\partial_z$) se transforma no polinômio $\tilde{f}$ (que multiplica $\partial_{\tilde{x}}$) pela ação da transformação $\tilde{T}$.

- Dessa forma (voltando ao exemplo), ficamos com

```
[> tr := [x=y,y=z,z=x]:
[> f_2 := subs(tr,h);
[> g_2 := subs(tr,f);
[> h_2 := subs(tr,g);
[> P1_2 := subs(tr,P3);
[> P2_2 := subs(tr,P1);
[> P3_2 := subs(tr,P2);
```

$$f_2 := z\left(2\,x^2z^6 + 3\,x^2y^4z - 4\,xyz^3 - 3\,xz^3 + 2\,y^2\right)$$

$$g_2 := yz\left(2\,x^2z^6 + 3\,x^2y^4z - 4\,xyz^3 - 3\,xz^3 + 2\,y^2\right)$$

$$h_2 := 4\,x^3y^4z^6 + x^2y^4z^6 - 8\,x^2y^5z^3 - 2\,xy^5z^3 - xy^4z^3 + 4\,xy^6 + xy^5 + y^6 + z^5 - yz^2$$

$$P1_2 := x^2y^4z^6 - 2\,xy^5z^3 - xy^4z^3 + y^6 + z^5$$

$$P2_2 := 4\,x^3y^3z^6 - 8\,x^2y^4z^3 + 4\,xy^5 + xy^4 - z^2$$

$$P3_2 := -z\left(2\,x^2z^6 + 3\,x^2y^4z - 4\,xyz^3 - 3\,xz^3 + 2\,y^2\right)$$

- Neste ponto devemos substituir os polinômios acima na equação para determinar o $\tilde{S}_1$

$$\tilde{\chi}(\tilde{S}_1) = \tilde{S_1}^{\,2}\left(\frac{\tilde{f}\tilde{g}_{\tilde{z}} - \tilde{g}\tilde{f}_{\tilde{z}}}{\tilde{f}}\right) + \tilde{S}_1\left(\frac{\tilde{g}\tilde{f}_{\tilde{y}} - \tilde{f}\tilde{g}_{\tilde{y}} + \tilde{f}\tilde{h}_{\tilde{z}} - \tilde{h}\tilde{f}_{\tilde{z}}}{\tilde{f}}\right) - \left(\frac{\tilde{f}\tilde{h}_{\tilde{y}} - \tilde{h}\tilde{f}_{\tilde{y}}}{\tilde{f}}\right),$$

  onde

$$\tilde{S}_1 = \frac{\tilde{P}_2}{\tilde{P}_3} = \frac{\tilde{P}_2}{\tilde{f}},$$

  ou seja, ficaremos com uma equação cuja única incognita é o polinômio $\tilde{P}_2$.

- Aparentemente, tínhamos conseguido; contudo, ao substituirmos efetivamente os polinômios percebemos que a ideia tinha ido além de nossas espectativas, pois o termo que multiplica $\tilde{S_1}^{\,2}$, dado por

$$\frac{\tilde{f}\tilde{g}_{\tilde{z}} - \tilde{g}\tilde{f}_{\tilde{z}}}{\tilde{f}},$$

  tinha como valor

```
[> (f_2*diff(g_2,z)-g_2*diff(f_2,z))/f_2;
```

$$0$$

- Em outras palavras, a equação tinha sido reduzida a

$$\tilde{\chi}(\tilde{S}_1) = \tilde{S}_1\left(\frac{\tilde{g}\tilde{f}_{\tilde{y}} - \tilde{f}\tilde{g}_{\tilde{y}} + \tilde{f}\tilde{h}_{\tilde{z}} - \tilde{h}\tilde{f}_{\tilde{z}}}{\tilde{f}}\right) - \left(\frac{\tilde{f}\tilde{h}_{\tilde{y}} - \tilde{h}\tilde{f}_{\tilde{y}}}{\tilde{f}}\right),$$

  o que significava que tínhamos uma equação **linear** em $\tilde{S}_1$ e, portanto, em $\tilde{P}_2$.

- Usando o método dos coeficientes indeterminados (MCI) temos uma equação linear para os coeficientes:

```
[> PP := Polg([x,y,z],12,a_,'aall');
[> A1 := (-f_2*diff(g_2,y)+g_2*diff(f_2,y) + f_2*diff(h_2,z)-h_2*diff
   (f_2,z))/f_2:
[> A0 := - (f_2*diff(h_2,y)-h_2*diff(f_2,y))/f_2:
[> EE := -PP*chi(f_2)+f_2*chi(PP)-f_2*PP*A1-f_2^2*A0:
[> t1 := time(): eqs := {coeffs(collect(EE,[x,y,z],distributed),[x,y,
   z])}: solp := solve(eqs,aall); time()-t1;
[> factor(subs(op(solp),PP)/f);
```

$$PP := a_0 + a_1 x + a_2 y + a_3 z + a_4 x^2 + \cdots + a_{454}\, z^{12}$$

$$solp := [[a_0 = 0, a_1 = 0, a_2 = 0, a_3 = 0, a_4 = 0, \cdots, a_{453} = 0, a_{454} = 0]]$$

$$0.375$$

$$-\frac{4\,x^3y^3z^6 - 8\,x^2y^4z^3 + 4\,xy^5 + xy^4 - z^2}{z\,(2\,x^2z^6 + 3\,x^2y^4z - 4\,xyz^3 - 3\,xz^3 + 2\,y^2)}$$

**Observação 5.1**
*Em relação a todo o processo descrito nos itens acima, o ponto mais importante a ser observado é o fato que toda 1EDO pertencente ao conjunto $L_S$ pode ser transformada em uma 1EDO tal que $\theta = \ln(x)$. Em vista disso, teremos (veja o teorema 2.2) que $h|P_2$ e, usando o resultado expresso no teorema 5.1, a estratégia mostrada sempre pode ser aplicada.*

## 5.2 Determinando a função-$S$ linearmente

Nesta seção vamos apresentar uma formalização das ideias que foram mostradas na seção acima. Para este fim, vamos começar com um resultado que generaliza o tipo de procedimento construído na seção anterior:

**Teorema 5.1** *Seja $\mathcal{E} \equiv y' = \phi(x, y, \theta)$ uma 1EDO tal que $\mathcal{E} \in L_S$ (veja a definição 2.2). Então, se $\chi \equiv f\,\partial_x(I) + g\,\partial_y(I) + h\,\partial_z$ é o campo vetorial polinomial associado com $\mathcal{E}$, sua função-S associada $S_1$ pode ser determinada por um algoritmo MCI linear em relação aos coeficientes.*

*Prova.* Das hipóteses do teorema temos que: se $\mathcal{E} \in L_S$ então existe uma tranformação de variáveis $T$ tal que a 1EDO transformada , $\widehat{\mathcal{E}}$ é da forma

$$y' = \frac{dy}{dx} = \varphi(x, y, \theta) = \frac{M(x, y, \theta)}{N(x, y, \theta)}, \tag{203}$$

onde $M$, $N \in \mathbb{C}[x, y, \theta]$ são coprimos e $\theta = \ln(x)$. Usando o resultado do teorema 2.2 podemos afirmar que $h \mid P_2$ e, além disso, $f = x\,h$. Para essa 1EDO, temos que as funções-

$S$ obedecem às equações (veja (51))

$$\chi(S_1) = {S_1}^2 f\left(\frac{g}{f}\right)_z + S_1\, f\left(\left(\frac{h}{f}\right)_z - \left(\frac{g}{f}\right)_y\right) - f\left(\frac{h}{f}\right)_y, \tag{204}$$

$$\chi(S_2) = {S_2}^2 g\left(\frac{f}{g}\right)_z + S_2\, g\left(\left(\frac{h}{g}\right)_z - \left(\frac{f}{g}\right)_x\right) - g\left(\frac{h}{g}\right)_x, \tag{205}$$

$$\chi(S_3) = {S_3}^2 h\left(\frac{f}{h}\right)_y + S_3\, h\left(\left(\frac{g}{h}\right)_y - \left(\frac{f}{h}\right)_x\right) - h\left(\frac{g}{h}\right)_x, \tag{206}$$

onde

$$S_1 = \frac{I_y}{I_z} = \frac{P_2}{P_3}, \qquad S_2 = \frac{I_x}{I_z} = \frac{P_1}{P_3}, \qquad S_3 = \frac{I_x}{I_y} = \frac{P_1}{P_2}.$$

Substituindo as funções $S_1$, $S_2$ e $S_3$ em (204), (205) e (206) e levando em conta que $h \,|\, P_2 \Rightarrow P_2 = p\,h$ e que $f = x\,h$, vamos obter

$$\chi\left(\frac{p\,h}{P_3}\right) = \left(\frac{p\,h}{P_3}\right)^2 h\left(\frac{g}{h}\right)_z - \left(\frac{p\,h}{P_3}\right) h\left(\frac{g}{h}\right)_y, \tag{207}$$

$$\chi\left(\frac{P_1}{P_3}\right) = \left(\frac{P_1}{P_3}\right)^2 g\left(\frac{x\,h}{g}\right)_z + \left(\frac{P_1}{P_3}\right) g\left(\left(\frac{h}{g}\right)_z - \left(\frac{x\,h}{g}\right)_x\right) - g\left(\frac{h}{g}\right)_x, \tag{208}$$

$$\chi\left(\frac{P_1}{p\,h}\right) = \frac{P_1}{p}\left(\left(\frac{g}{h}\right)_y - 1\right) - h\left(\frac{g}{h}\right)_x. \tag{209}$$

Multiplicando a equação (207) por $\dfrac{P_3}{p\,h}$, a equação (208) por $\dfrac{P_3}{P_1}$ e a equação (209) por $\dfrac{p\,h}{P_1}$, temos

$$\frac{\chi(p)}{p} + \frac{\chi(h)}{h} - \frac{\chi(P_3)}{P_3} = \frac{p\,h^2}{P_3}\left(\frac{g}{h}\right)_z - h\left(\frac{g}{h}\right)_y, \tag{210}$$

$$\frac{\chi(P_1)}{P_1} - \frac{\chi(P_3)}{P_3} = \frac{P_1}{P_3} g\left(\frac{x\,h}{g}\right)_z + g\left(\left(\frac{h}{g}\right)_z - \left(\frac{x\,h}{g}\right)_x\right) - \frac{P_3}{P_1} g\left(\frac{h}{g}\right)_x, \tag{211}$$

$$-\frac{\chi(p)}{p} - \frac{\chi(h)}{h} + \frac{\chi(P_1)}{P_1} = h\left(\left(\frac{g}{h}\right)_y - 1\right) - \frac{p\,h^2}{P_1}\left(\frac{g}{h}\right)_x. \tag{212}$$

Das equações (210) e (212) podemos deduzir que $\dfrac{\chi(p)}{p}$ é um polinômio (isto é, $p$ é um PD do operador $\chi$), pois $p$ não pode ser um fator de $P_1$ e $P_3$ simultaneamente (uma vez que já é fator de $P_2$). Assim, dois casos possíveis:

- O fator $p$ é constante.
- O fator $p$ é um polinômio.

O primeiro item é trivial pois, nesse caso, a equação (210) multiplicada por $P_3$ se torna

$$P_3\frac{\chi(h)}{h} - \chi(P_3) = p\,h^2\left(\frac{g}{h}\right)_z - h\,P_3\left(\frac{g}{h}\right)_y, \tag{213}$$

que é linear em $P_3$ e, portanto, em seus coeficientes.
No segundo caso, o problema é que sendo $P_2 = p\,h$, como $\chi(I) = 0$ temos que $f\,P_1 + g\,P_2 + h\,P_3 = x\,h\,P_1 + g\,p\,h + h\,P_3 = h\,(x\,P_1 + g\,p + P_3) = 0$ implicando que, por coincidência, em $x\,P_1 + P_3 = g\,p$ (isto é, $p$ é um fator de $x\,P_1 + P_3$). Nessa situação, tudo que precisamos fazer é uma transformação de variáveis na 1EDO que tinha $\theta = \ln(x)$ (por exemplo, $\{x = \mathrm{e}^x, y = y\}$) e a infeliz combinação que implica em $p\,|\,(x\,P_1 + P_3)$ será eliminada. □

**Algoritmo 5.1** *($(L_S)^2$ –Esboço)*

**Passos:**

1. Faça uma transformação de variáveis tal que $\theta$ seja $\ln(x)$. Nesse caso, usando o resultado do teorema 2.2 podemos afirmar que $h = k\,P_2$ e, além disso, $f = x\,h$ (caso, por infelicidade, $h\,p = P_2$, fazemos uma tranformação de variáveis como a prescrita na demonstração do teorema 5.1 acima).

2. Faça a tranformação de variáveis $T$ dada por:

$$T_1 = \begin{cases} x = y_1 \\ y = z_1 \\ z = x_1 \end{cases}$$

   Como $z = x_1$, temos que

$$T_1(h\,\partial_z) = T_1(h)\,\partial_{x_1} = f_1\,\partial_{x_1} \;\;\Rightarrow\;\; T_1(h) = f_1$$

   e, portanto:

$$f_1 = T_1(h),\; g_1 = T_1(f) = T_1(x\,h) = y_1\,f_1,\; h_1 = T_1(g),$$

   e

$$P1_1 = T_1(P3),\; P2_1 = T_1(P1),\; P3_1 = T_1(P2).$$

3. Assim, o termo que multiplica $S1_1^2$ será zero ($\partial_{z_1}(g_1/f_1) = \partial_{z_1}(y_1\,f_1/f_1) = 0$.

4. Construa um polinômio $PP$ com coeficientes indeterminados e substitua $S1_1 = P2_1/P3_1 = T_1(P1)/T_1(P2) = PP/(k\,f_1)$ na equação para a função $S1_1$.

5. Resolva o sistema linear nos coeficientes de $PP$ e aplique a inversa $T_1^{-1}$ para obter o polinômio $P1$. A partir de $P1$ temos $P3$ (uma vez que $x\,P1 + P3 = k\,g$ e, portanto a função $S1 = P2/P3 = k\,h/P3$.

## 5.3 Desempenho do algoritmo $(L_S)^2$

Nesta seção, vamos mostrar brevemente o desempenho do algoritmo no cálculo da função $S$.

Considere a seguinte 1EDO

$$\frac{dy}{dx} = -\frac{4\,(\ln(x))^3 + (-8\,y^3 - 1)\,(\ln(x))^2 + (4\,y^6 - x^4 + 2\,y^3 + 1)\ln(x) + x^4y^3 - y^6}{3\,y^2x\,((x^4 - 1)\ln(x) - x^4y^3)} \tag{214}$$

**Usando** *LeapS1ode*:

O pacote não consegue encontrar a solução.

**Usando o algoritmo** $(L_S)^2$:

O algoritmo encontra a solução em 0,19 segundos (com um consumo de memória de 17 MB).

Considere agora a seguinte 1EDO:

$$\frac{dy}{dx} = -\frac{(4\,x^4y^3 + 1)\,(x^4y^3 + \ln(x) - y)}{x\left((\ln(x))^2 + (2\,x^4y^3 + 3\,x^4y^2)\ln(x) + x^8y^6 + 3\,x^8y^5 - 3\,x^4y^3\right)} \tag{215}$$

**Usando** *LeapS1ode*:

O pacote encontra a solução em 11 segundos (com um consumo de memória de 42 MB).

**Usando o algoritmo** $(L_S)^2$:

O algoritmo encontra a solução em 0,44 segundos (com um consumo de memória de 17 MB).

**Observação 5.2** *Alguns comentários:*

- *O desempenho do algoritmo $(L_S)^2$ é muito promissor uma vez que é linear nos coeficientes. Por isso, mesmo sem uma implementação pensada (um pacote computacional estruturado), isto é, apenas com uma sequência simples de comandos, ele consegue excelentes resultados.*

- *Como a principal dificuldade no método da função-S é a própria determinação da função, este algoritmo praticamente põe fim às limitações (para 1EDOs da classe $L_S$) decorrentes da existência de polinômios de Darboux de graus elevados ou da existência de simetrias muito complicadas.*

- *Esse resultado nos faz pensar em como aplicá-lo na resolução de 1EDOs racionais (com soluções gerais Liouvillianas) tal que os PDs presentes no fator integrante tenham graus muito elevados ou que suas simetrias sejam muito complexas.*

- *Poderíamos usar algo semelhante para o cálculo da função-S em 2EDOs racionais que possuem integrais primeiras Liouvillianas mas que não advêm de um processo de associação com uma 1EDO que apresente uma função elementar (isto é, onde não temos conhecimento dos polinômios que compõe o campo vetorial $\chi$)?*

# CONCLUSÃO

Neste trabalho desenvolvemos um novo método para resolver 1EDOs que apresentam funções elementares. O nosso procedimento é aplicável a 1EDOs que possuam solução geral Liouvilliana e, além disso, ele é restrito a 1EDOs que possam ser expressas na seguinte forma:

$$y' = \frac{dy}{dx} = \phi(x, y, \theta) = \frac{M(x, y, \theta)}{N(x, y, \theta)} \tag{216}$$

onde $\phi$ é uma função racional de $(x, y, \theta)$ e $\theta$ é uma função elementar de $(x, y)$ ($M$ e $N$ são polinômios coprimos em $(x, y, \theta)$). A ideia central consiste em estabelecermos uma conexão entre uma 1EDO (do tipo mostrado na equação (216)) e um campo vetorial polinomial em três variáveis de maneira que a solução geral da 1EDO esteja associada à integral primeira do campo vetorial. A ideia tem uma dupla função: em primeiro lugar, podemos aplicar técnicas já conhecidas para campos vetoriais polinomiais como, por exemplo, as abordagens DPS; em segundo lugar, podemos adaptar uma técnica já desenvolvida por nós em (3) para lidar com 2EDOs racionais (o método da função $S$) e que, além de bastante eficiente, já contava com um pacote computacional para buscar a função $S$ e, a partir dela, encontrar a integral primeira.

Podemos ser levados a pensar (a princípio) que o problema de encontrar uma integral primeira Liouvilliana para um campo vetorial polinomial (ou, equivalentemente, para uma 2EDO racional) seja (talvez) bem mais complicado que encontrar a solução geral de uma 1EDO. Porém, há dois fatos que fazem com que essa troca de problemas leve a um resultado positivo: a 1EDO não é racional mas a 2EDO é e, além disso, o método da função $S$ é muito eficaz precisamente nos casos em que a busca pelas integrais primeiras de 2ODEs racionais (por procedimentos como o método das simetrias de Lie e abordagens DPS) são particularmente problemáticas.

O método que desenvolvemos (algoritmo $Lsolv$) se apresentou (em uma primeira avaliação) razoavelmente eficiente se comparado com a alternativa mais canônica que seria tentar uma abordagem DPS ou de simetrias. Contudo, o surgimento de dois resultados inesperados, engrandeceu o nosso trabalho excedendo em muito as nossas espectativas. Esses dois resultados (expressos nos teoremas 2.2, 2.3 e 5.1) permitiram a criação de dois algoritmos ($FastL_S$ e $(L_S)^2$) que, para uma classe muito abrangente de 1EDOs (a classe $L_S$ – veja a definição 2.2), apresentaram um desempenho fantástico.

Os métodos (que usam os algoritmos $Lsolv$ e $FastL_S$ foram implementados em um pacote computacional na plataforma de computação simbólica Maple. O pacote, além de resolver 1EDOs 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 soluções como também na pesquisa

em física e matemática.

Também discutimos o desempenho do pacote e dos algoritmos que desenvolvemos, mostrando como ele se sai comparativamente a outros métodos de ataque ao problema. Mostramos também um exemplo prático de como usar o pacote na pesquisa em física e matemática (veja o exemplo 4.2 e a seção 4.2.2). Por fim, em relação à função do pacote como uma ferramenta muito útil na pesquisa, podemos apresentar o resultado que responde por todo o capítulo 5 deste trabalho. Nele apresentamos um procedimento linear para o cálculo da função $S$ associada à 2EDO racional cuja integral primeira é equivalente à solução geral da 1EDO. Como a determinação da função $S$ é um passo fundamental (e o mais difícil) de todo o processo, esse resultado configura um novo patamar algorítmico para este tipo de problema. Além disso, ele abre toda uma linha de pesquisa sobre possíveis métodos análogos para serem aplicados em problemas não lineares semelhantes (veja os itens 3 e 4 abaixo).

Algumas considerações sobre a pesquisa futura que pode acontecer como consequência direta (ou indireta) do trabalho que realizamos aqui:

1. Pesquisar o método (também híbrido) que foi descoberto ao resolvermos o exemplo 4.2. Em resumo, após o cômputo da função $S$, usar os operadores de Darboux das 1EDOs racionais associadas (veja (3)) $D_A$ (etc) para computar os polinômios de Darboux (em duas e não em três variáveis). Pesquisar também se temos alguma vantagem pelo fato de contarmos com mais de um operador que possui um mesmo conjunto de PDs.

2. Busca de regiões de integrabilidade para qualquer problema em física (ou outra área) que seja modelado por EDOs com parâmetros.

3. A determinação da função $S$ para 2EDOs racionais que não tenham sido construídas por associação com uma 1EDO pertencente à classe $L_S$ (mesmo que ela possa ser associada com uma). Nesse caso, pode se pensar em resolver a seguinte questão: podemos decidir se uma 2EDO racional pode ser associada a alguma 1EDO $\in L_S$? Ou, em outras palavras, podemos decidir se uma dada 2EDO racional apresenta uma integral primeira Liouvilliana $I$ tal que suas derivadas sejam da forma 35?

4. Busca de algoritmos lineares para a solução de problemas não lineares que tragam alguma analogia com a equação da função $S$ para sistemas 3D.

5. Etc.

# REFERÊNCIAS

1 DUARTE, L.G.S. et al. Extension of the Prelle-Singer method and a Maple implementation. *Computer Physics Communications*, Holanda, v. 144, n. 1, p. 46-62, 2002.

2 AVELLAR, J.; DUARTE, L.G.S.; MOTA, L.A.C.P. PSsolver: a Maple implementation to solve first order ordinary differential equations with liouvillian solutions. *Computer Physics Communications*, [S.l.], v. 183, n. 10, p. 2313-2316, Oct. 2012.

3 AVELLAR, J.et al. Dealing with rational second order ordinary differential equations where both Darboux and Lie find it dicult: the S-function method. *Computer Physics Communications*, [S.l.], v. 234, p. 302-314, 2019.

4 LIE, S. *Theorie der Transformationsgruppen*. Chelsea: American Mathematical Soc, 1970.

5 DARBOUX, G. Mémoire sur les équations différentielles algébriques du premier ordre et du premier degré (Mélanges), *Bull. Sci. Math.*, [S. l.], v. 2, n. 2, p. 60-96, 1878.

6 CHEB-TERRAB, E.S.; DUARTE, L.G.S.; MOTA, L.A.C.P. Computer algebra solving of first order ODEs using symmetry methods. *Computer Physics Communications*, [S.l.], v. 101, p. 254-262, 1997.

7 CHEB-TERRAB, E.S.; DUARTE, L.G.S.; MOTA, L.A.C.P. Computer algebra solving of second order ODEs using symmetry methods. *Computer Physics Communications*, [S.l.], p. 108-116, 1998

8 OLVER, P.J. *Applications of Lie groups to differential equations*. New York: Springer-Verlag, 1986.

9 ABRAHAM-SHRAUNER, B.; GUO, A. Hidden symmetries sssociated with the projective group of nonlinear first-order ordinary differential equations. *J. Phys. A: Math.Gen.*, [S.l.], v. 25, p. 5597-5608, 1992.

10 ABRAHAM-SHRAUNER, B.; GUO, A. Hidden and Nonlocal Symmetries of nonlinear differential equations. *Modern Group Analysis: Advanced Analytical and Computational Methods in Mathematical Physics, Hidden Symmetries of Differential Equations*, [S.l.], p. 1-5, 1993.

11 ABRAHAM-SHRAUNER, B.; GOVINDER, K.s.; LEACH, P.G.L. Integration of second order ordinary differential equations not possessing Lie point symmetries. *Phys. Lett. A*, [S.l.], v. 203, p. 169-174, 1995.

12 ABRAHAM-SHRAUNER, B. Hidden symmetries and nonlocal group generators for ordinary differential equations. *IMA Journal of Applied Mathematics*, [S.l.], v. 56, p. 235-252, 1996.

13 GOVINDER, K.S.; LEACH, P.G.L. A group theoretic approach to a class of second-order ordinary differential equations not possessing Lie point symmetries. *J. Phys. A: Math. Gen.*, [S.l.], v. 30, p. 2055-2068, 1997.

14 ADAM, A.A.; MAHOMED, F.M. Non-local symmetries of first-order equations. *IMA Journal of Applied Mathematics*, [S.l.], v. 60, p. 187-198, 1998.

15 GANDARIAS, M.L.; BRUZÓN, M.S. Reductions for some ordinary differential equations through nonlocal symmetries. *Journal of Nonlinear Mathematical Physics*, [S.l.], v. 18, p. 123-133, 2011.

16 BRUZÓN, M.S.; GANDARIAS, M.L.; SENTHILVELAN, M. Nonlocal symmetries of Riccati and Abel chains and their similarity reductions. *Journal of Mathematical Physics*, [S.l.], v. 53, p. 023512-023518, 2012.

17 MURIEL, C.; ROMERO, J.L. New methods of reduction for ordinary differential equations. *IMA Journal Applied Mathematics*, [S.l.], v. 66, p. 111-125, 2001.

18 MURIEL, C.; ROMERO, J.L. $C^\infty$-Symmetries and reduction of equations without Lie point symmetries. *J. Lie Theory*, [S.l.], v. 13, p. 167-188, 2003.

19 MURIEL, C.; ROMERO, J.L. The $\lambda$-symmetry reduction method and Jacobi last multipliers. *Communications in Nonlinear Science and Numerical Simulation*, [S.l.], v. 19, p. 807-820, 2014.

20 MURIEL, C.; ROMERO, J.L. Nonlocal symmetries, telescopic vector fields and $\lambda$-symmetries of ordinary differential equations. *Symmetry, Integrability and Geometry: Methods and Applications*, [S.l.], v. 8, p. 106-126, 2012.

21 CICOGNA, G.; S.WALCHER. Dynamical systems and $\sigma$-symmetries. *Journal of Physics A: Mathematical and Theoretical*, [S.l.], v. 46, n. 23, p. 235204-235212, 2013.

22 CICOGNA, G.; GAETA, G.; MORANDO, P. On the relation between standard and $\mu$-symmetries for PDEs. *Journal Of Physics A: Mathematical and General*, [S.l.], v. 37, n. 40, p. 9467-9473, 2004.

23 PUCCI, E.; SACCOMANDI, G. On the reduction methods for ordinary differential equations. *Journal of Physics A: Mathematical and General*, [S.l.], v. 35, p. 6145-6155, 2002.

24 NUCCI, M.C. Jacobi last multiplier and Lie symmetries: a novel application of an old relationship. *Journal of Nonlinear Mathematical Physics*, [S.l.], v. 12, p. 284-304, 2005.

25 NUCCI, M.C. Lie symmetries of a Painlevé-type equation without Lie symmetries. *Journal of Nonlinear Mathematical Physics*, [S.l.], v. 15, p. 205-211, 2008.

26 CAIRÓ, L.; LLIBRE, J.. Darboux integrability for 3D Lotka-Volterra systems. *J. Phys. A: Math. Gen.*, [S.l.], v. 33, p. 2395-2406, 2000.

27 C. Christopher, CHRISTOPHER, C. Invariant algebraic curves and conditions for a center. *Proc. R. Soc. Edin. A*, [S.l.], v. 124, p. 1209-1217, 1994.

28 PRELLE, M.; SINGER, M. Elementary first integral of differential equations. *Transactions of the American Mathematical Society*, [S.l.], v. 279, p. 215-229, 1983.

29 SHTOKHAMER, R. *Solving first order differential equations using the Prelle-Singer algorithm*. Dover: Center For Mathematical Computation, University of Delaware, 1988. 2 p

30 COLLINS, C. B. Algebraic classification of homogeneous polynomial vector fields in the plane. *Japan Journal of Industrial And Applied Mathematics*, [S.l.], v. 13, n. 1, p. 63-91, 1996.

31 SINGER, M. Liouvillian first integrals. *Trans. Amer. Math. Soc.*, [S.l.], v. 333, p. 673-688, 1992.

32 CHRISTOPHER, C.; LLIBRE, J. Integrability via invariant algebraic curves for planar polynomial differential systems. *Annual Differential Equations*, [S.l.], v. 16, n. 1, p. 5-19, 2000.

33 LLIBRE, J. Integrability of polynomial differential systems. In: CAÑADA, A.; DRÁBEK, P.; FONDA, A. *Handbook of Differential equations, Ordinary Differential Equations*. Holanda: Elsevier B.V., 2004. Cap. 5. p. 437-531.

34 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.

35 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.

36 AVELLAR, J. et al. Integrating first- order differential equations with Liouvillian solutions via quadratures: a semi- algorithmic method. *Journal of Computational And Applied Mathematics*, [S.l.], v. 182, p. 327-332, 2005.

37 DUARTE, L.G.S. et al. Solving second order ordinary differential equations by extending the Prelle-Singer method. *J. Phys. A: Math.Gen.*, [S.l.], v. 34, p. 3015-3024, 2001.

38 AVELLAR, J. et al. Determining Liouvillian first integrals for dynamical systems in the plane. *Computer Physics Communications*, [S.l.], v. 177, p. 584-596, 2007.

39 AVELLAR, J. et al. A semi-algorithm to find elementary first order invariants of rational second order ordinary differential equations. *Appl. Math. Comp.*, [S.l.], v. 184, p. 2-11, 2007.

40 DUARTE, L. G. S.; MOTA, L. A. C. P. Finding elementary first integrals for rational second order ordinary differential equations. *Journal of Mathematical Physics*, [S.l.], v. 50, n. 1, p. 013514-013520, 2009.

41 DUARTE, L.G.S.; MOTA, L.A.C.P. Finding elementary first integrals for rational second order ordinary differential equations. *J. Phys. A: Math. Theor.*, [S.l.], v. 43, n. 6, p. 065204-065210, 2010.

42 LLIBRE, J.; ZHANG, X. Darboux theory of integrability for polynomial vector fields in $R^n$ taking into account the multiplicity at infinity. *Bull. Sci. Math.*, [S.l.], v. 133, p. 765-778, 2009.

43 BRAZ, A. et al. A generalization of the S-function method applied to a Duffing–Van der Pol forced oscillator. *Computer Physics Communications*, [S.l.], v. 254, p. 107306-107312, 2020

44 AVELLAR, J.; DUARTE, L.G.S.; MOTA, L.A.C.P. A Maple package to find first order differential invariants of 2ODEs via a Darboux approach. *Computer Physics Communications*, [S.l.], v. 185, n. 1, p. 307-316, jan. 2014.

45 DAVENPORT, J.H.; SIRET, Y.; TOURNIER, E. *Computer Algebra: systems and algorithms for algebraic computation*. London: Academic Press, 1993.

46 BLUMAN, G.W.; ANCO, S.C. *Symmetries and integration methods for differential equations*. New York: Springer-Verlag, 2002. 154v. (Applied Mathematical Series).

47 IBRAGIMOV, N.H. *Elementary Lie group analysis and ordinary*. Wiley: Chichester, 1999.

48 SCHWARZ, F.*Theory for solving ordinary differential equations*. New York: Chapman Hall / CRC - Taylor and Francis Group, 2008.

49 STEEB, W.H. *Continuous symmetries, Lie algebras, differential equations and computer algebra*. New Jersey: World Scientic Publishing Co. Pte. Ltd., 2007

50 AVELLAR, J. Determinação de Integrais Primeiras Liouvillianas em Equações Diferenciais Ordinárias Racionais de Segunda Ordem. 2013. 102 f. Tese (Doutorado) - Curso de Física, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2013.

51 EIRAS, J. P. C. Sistemas 3D de 1EDOs: a busca por invariantes liouvillianos. 2017. 98 f. Dissertação (Mestrado) - Curso de Física, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2017.

52 KAMKE, E. *Differentialgleichungen: Lösungsmethoden und Lösungen*. New York: Chelsea Publishing Co, 1959.

53 LEVINSON, N.; SMITH, O. A general equation for relaxation oscillations.*Duke Mathematical Journal*, [S.l.], v. 9, p. 382-403, 1942.

54 RAN, Z. One exactly soluble model in isotropic turbulence. *Advances And Applications In Fluid Mechanics*, [S.l.], v. 5, p. 41-47, 2009.

55 LIÉNARD, A. Systemes de Lienard et decomposition potentielle-Hamiltonienne. *Revue Générale de L'électricité*, [S.l.], v. 23, p. 901-912, 1928.

56 POL, B. van Der. On relaxtion-oscillations. *Edinburgh and Dublin Philosophical Magazine and Journal Of Science*, [S.l.], v. 2, p. 978-92, 1927.

57 POL, B. van Der; MARK, J. van Der. The heart beat considered as a relaxation oscillations and an electrical model of the heart. *Edinburgh and Dublin Philosophical Magazine And Journal Of Science*, [S.l.], v. 6, p. 763-775, 1928.

58 FITZHUGH, F. Impulses and physiological states in theoretical models of nerve membranes. *Biophysics Journal*, [S.l.], v. 1, p. 445-466, 1961.

59 STROGATZ, S. H. *Nonlinear Dynamics and Chaos*. Cambridge: Addison-Wesley, 1994.