

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA APLICADA

SOLUÇÃO DE SISTEMAS DE EQUAÇÕES
ALGÉBRICO-DIFERENCIAIS
ORDINÁRIAS DE ÍNDICE SUPERIOR

por

Márcia Lourenço

Dissertação submetida como requisito parcial
para obtenção do grau de
Mestre em Matemática Aplicada

Prof. Dr. Argimiro Resende Secchi
Orientador

Porto Alegre, 25 fevereiro de 2002.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Lourenço, Márcia

Solução de Equações Algébrico-Diferenciais Ordinárias de Índice Superior / Márcia Lourenço. – Porto Alegre: PPGMAp da UFRGS, 2002.

90p.: il.

Dissertação (mestrado) – Universidade do Federal do Rio Grande do Sul, Programa de Pós-Graduação em Matemática Aplicada, Porto Alegre, 2002.
Orientador: Secchi, Argimiro Resende

Dissertação: Matemática Aplicada
Solução de Equações Algébrico-Diferenciais Ordinárias de Índice Superior, Dissertação

SOLUÇÃO DE EQUAÇÕES ALGÉBRICO-DIFERENCIAIS ORDINÁRIAS DE ÍNDICE SUPERIOR

por

MÁRCIA LOURENÇO

Dissertação submetida ao programa de Pós-Graduação em Matemática Aplicada do Instituto de Matemática da Universidade Federal do Rio Grande do Sul, como requisito parcial para a obtenção do grau de

Mestre em Matemática Aplicada

Linha de Pesquisa: Algoritmos Numéricos e Algébricos

Orientador: Prof. Dr. Argimiro Resende Secchi

Banca Examinadora:

Prof. Dr. Vilmar Trevisan

PPGMAp/UFRGS

Prof. Dr. Paulo Ricardo de Avila Zingano

PPGM/UFRGS

Prof. Dr. Rogério Ricardo Steffenon

UNISINOS/RS

Prof. Dr. Vilmar Trevisan

Coordenador do PPGMAp

Agradecimentos:

Aos amigos da UCS, pela amizade e por tornarem agradável o ambiente de estudo.

Ao professor Argimiro Resende Secchi, pela ajuda e auxílio na elaboração deste trabalho.

A minha mãe Rosa Lourenço e aos meus irmãos Adão Luís e Eva Regina pela força para que eu continuasse.

Ao meu namorado Frederico Sporket, pela ajuda, compreensão e carinho.

ÍNDICE

ÍNDICE DE FIGURAS	VI
ÍNDICE DE TABELAS	VII
LISTA DE SÍMBOLOS	VIII
RESUMO	IX
ABSTRACT	X
1. Introdução	1
2. Conceitos básicos	3
2.1. EAD x EDO	3
2.2. Índice	4
2.3. Caracterização Estrutural de um Sistema de EADs	8
2.4. Resolubilidade	12
2.5. Inicialização	15
3. Redução de Índice	19
3.1. Teoria de Grafos	20
3.2. Outras Técnicas de Redução do Índice	39
4. Solução do Sistema Reduzido	46
4.1. Efeito de Não Considerar as Restrições Escondidas	49
4.2. Proposição de Técnicas Para Levar em Conta as Restrições Escondidas	51
i. Sistema Reduzido	52
ii. Sistema Estendido com Variáveis de Restrições	53
iii. Sistema Estendido com as Derivadas das Variáveis de Restrição..	54
5. Exemplos Resolvidos	55
5.1. Exemplo 1 – índice 2	55
5.2. Exemplo 2 – índice 2	66
5.3. Exemplo 3 – índice 3	71
5.4. Exemplo 4 – índice 3	76
6. Conclusões e Sugestões	85
Referências Bibliográficas	87

ÍNDICE DE FIGURAS

Figura 3.1	Grafo dirigido.....	25
Figura 3.2	Grafo bipartido.....	26
Figura 3.3	Grafo bipartido do exemplo 1.....	29
Figura 3.4	Grafo resultante da diferenciação da f_5	31
Figura 3.5	Grafo resultante do sistema estendido.....	33
Figura 3.6	Grafo bipartido $\{f\}--\{x\}$ do exemplo 2	35
Figura 3.7	Grafo bipartido intermediário	36
Figura 3.8	Grafo bipartido solução final.....	36
Figura 3.9	Grafo bipartido $\{f\}--\{y\}$ do exemplo 3	37
Figura 3.10	Grafo bipartido intermediário	38
Figura 3.11	Grafo bipartido da solução final	38
Figura 5.1	Variável z_{xt} - exemplo 1	57
Figura 5.2	Variável y_{xt} - exemplo 1	57
Figura 5.3	Variável x_{xt} - exemplo 1	58
Figura 5.4	Análise do erro (e) - exemplo 1	63
Figura 5.5	Análise da integral do erro (e) - exemplo 1	63
Figura 5.6	Variável x_{1xt} -exemplo2	69
Figura 5.7	Variável x_{2xt} - exemplo 2	69
Figura 5.8	Variável y_{xt} - exemplo 2	70
Figura 5.9-	Análise do erro (e) - exemplo 2	70
Figura 5.10	Análise da integral do erro(e) -exemplo2	70
Figura 5.11-	Variável y_{xt} - exemplo3	73
Figura 5.12-	Variável x_{1xt} - exemplo3	73
Figura 5.13-	Variável x_{2xt} - exemplo3	74
Figura 5.14-	Variável z_{xt} - exemplo3	74
Figura 5.15-	Análise do erro (e) - exemplo3	75
Figura 5.16-	Análise da integral do erro (e) - exemplo3	75

Figura 5.17-	. Análise do erro (\mathbf{e}) - exemplo3	76
Figura 5.18	Análise da integral do erro (e) - exemplo3	76
Figura 5.19-	Coluna de Destilação em batelada – exemplo 4	77
Figura 5.20-	Variável x_0 - exemplo 4	81
Figura 5.21-	Variável T_0 - exemplo 4	81
Figura 5.22-	Variável R - exemplo 4	82
Figura 5.23-	Variável H_0 - exemplo 4	82
Figura 5.24	Análise do erro (\mathbf{e}) - exemplo 4	83
Figura 5.25	Análise da integral do erro (e) - exemplo 4	83
Figura 5.26	Análise do erro (\mathbf{e}) - exemplo 4	84
Figura 5.27	Análise da integral do erro (e) - exemplo 4	84

ÍNDICE DE TABELAS

Tabela 5.1	Representação numérica da variável z e sua derivada – exemplo 1	59
Tabela 5.2	Tabela comparativa das técnicas de inicialização – exemplo 1	59
Tabela 5.3	Tabela comparativa da variável z para valores de \mathbf{b} – exemplo 1	62
Tabela 5.4	Desempenho do algoritmo de integração para vários valores de \mathbf{b} -exemplo1	62
Tabela 5.5	Representação da variável de restrição - exemplo 1	65
Tabela 5.6	Tabela comparativa da variável y - exemplo 1	65
Tabela 5.7	Tabela comparativa da variável z - exemplo 1	66
Tabela 5.8	Tabela comparativa da variável x - exemplo 1	66
Tabela 5.9	Variável restrição – exemplo 2	68
Tabela 5.10	Tabela comparativa dos métodos de integração – exemplo 2	71
Tabela 5.11	Tabela comparativa dos métodos de integração – exemplo 3	76
Tabela 5.12	Tabela comparativa dos métodos de integração – exemplo 4	84

LISTA DE SÍMBOLOS

A0	Algoritmo de Duff (1981)
A1	Algoritmo de Pantelides (1988)
Ar1	Algoritmo de Gear e Petzold (1984)
Ar2	Algoritmo de Gear e Petzold (1984)
Ar3	Algoritmo de Bachamann (1990)
Ar4	Algoritmo de
e	Erro
<i>e</i>	Integral do erro
<i>J</i>	Matriz Jacobiana
<i>N</i>	Matriz de Nilpotência
a	Parâmetro da Matriz Pencil
b	Fator de Escala Para a Variável ϵ .
<i>n</i>	Dimensão do sistema de EADs
<i>nd</i>	Número de variáveis diferenciais
<i>na</i>	Número de variáveis algébricas
<i>l</i>	Parâmetro complexo
v_d	Índice diferencial
v_p	Índice de perturbação

RESUMO

A modelagem matemática de problemas importantes e significativos da engenharia, física e ciências sociais pode ser formulada por um conjunto misto de equações diferenciais e algébricas (EADs). Este conjunto misto de equações deve ser previamente caracterizado quanto a resolubilidade, índice diferencial e condições iniciais, para que seja possível utilizar um código computacional para resolvê-lo numericamente.

Sabendo-se que o índice diferencial é o parâmetro mais importante para caracterizar um sistema de EADs, neste trabalho aplica-se a redução de índice através da teoria de grafos, proposta por Pantelides (1988). Este processo de redução de índice é realizado numericamente através do algoritmo DAGRAFO, que transforma um sistema de índice superior para um sistema reduzido de índice 0 ou 1. Após esta etapa é necessário fornecer um conjunto de condições iniciais consistentes para iniciar o código numérico de integração, DASSLC.

No presente trabalho discute-se três técnicas para a inicialização consistente e integração numérica de sistemas de EADs de índice superior. A primeira técnica trabalha exclusivamente com o sistema reduzido, a segunda com o sistema reduzido e as restrições adicionais que surgem após a redução do índice introduzindo variáveis de restrição, e a terceira técnica trabalha com o sistema reduzido e as derivadas das variáveis de restrição.

Após vários testes, conclui-se que a primeira e terceira técnica podem gerar um conjunto solução mesmo quando recebem condições iniciais inconsistentes. Para a primeira técnica, esta característica decorre do fato que no sistema reduzido algumas restrições, muitas vezes com significado físico importante, podem ser perdidas quando as equações algébricas são diferenciadas. Trabalhando com o sistema reduzido e as derivadas das variáveis de restrição, o erro da inicialização é absorvido pelas variáveis de restrição, mascarando a precisão do código numérico. A segunda técnica adotada não tem como absorver os erros da inicialização pelas variáveis de restrição, desta forma, quando as restrições adicionais não são satisfeitas, não é gerada solução alguma.

Entretanto, ao aplicar condições iniciais consistentes para todas as técnicas, conclui-se que o sistema reduzido com as derivadas das variáveis restrição é o método mais conveniente, pois apresenta melhor desempenho computacional, inclusive quando a matriz jacobiana do sistema apresenta problema de mau condicionamento, e garante que todas as restrições que compõem o sistema original estejam presentes no sistema reduzido.

ABSTRACT

The mathematical modelling of important and significant problems of engineering, physics, and social sciences can be formulated by a mixed set of differential and algebraic equations (DAEs). This mixed set of equations should be previously characterized by its solvability, differential index, and initialization conditions, in order to use a computational code to solve it numerically.

Knowing that the differential index is the most important parameter to characterize a system of DAEs, in this work the index reduction is done through the graph theory, proposed by Pantelides (1988). The index reduction procedure is accomplished numerically by the DAGRAFO algorithm, that transforms a high-index system into a reduced system of index 0 or 1. After this stage it is necessary to provide a set of consistent initial conditions to start the code for numerical integration, DASSLC.

In the present work, three techniques for consistent initialization and numerical integration of DAEs are discussed. The first technique works exclusively with the reduced system, the second one works with the reduced system and the additional constraints that appear after the index reduction introducing constraint variables, and the third technique works with the reduced system and the derivatives of the constraint variables.

After several tests, it is concluded that the first and the third techniques can generate a solution set even when they receive inconsistent initial conditions. For the first technique this characteristic is due to the fact that in the reduced system some constraints, many times with important physical meaning, can be lost when the algebraic equations are differentiated. When working with the reduced system and the derivatives of the constraint variables, the error of the initialization is absorbed by the constraint variables, masking the accuracy of the numerical code. The second adopted technique is not able to absorb the errors of the initialization through the constraint variables, and then when the additional constraints are not satisfied no solution is generated.

However, when applying consistent initial conditions for all the techniques, it is concluded that the reduced system with the derivatives of the constraint variables is the most convenient method, because it presents better computational performance, even when the jacobian matrix of the system presents a problem of bad conditioning, and it guarantees that all the constraints that compose the original system are present in the reduced system.

1. INTRODUÇÃO

Problemas importantes de diferentes áreas como mecânica, robótica e engenharia química, por exemplo, sempre que puderem ser escritos através de conjuntos de equações diferenciais e algébricas, são chamados de sistemas de Equações Algébrico-Diferenciais (EADs).

Dependendo da área de aplicação, estes sistemas mistos, também podem ser conhecidos por sistemas de equações singulares, descritivas, não canônica, não casuais, degenerativas, restritivas, não padrão ou de ordem reduzida (Brenan et al., 1989).

O estudo da teoria analítica de sistemas de EADs iniciou-se por volta de 1960. Entretanto, a primeira aplicação de um método prático numérico surgiu em 1971, desenvolvido por Gear et al., 1971. Atualmente, sistemas de EADs são de interesse por causa da importância destas equações para uma grande classe de processos dinâmicos e, também, por causa das dificuldades numéricas encontradas pelos códigos computacionais.

Os modelos matemáticos utilizados por estes processos transientes geralmente representam balanços de massa, energia e quantidade de movimento, através das equações diferenciais, e relações constitutivas necessárias, como equações de estado, através das equações algébricas. Desta forma, torna-se conveniente trabalhar com modelagem na forma de sistemas de EADs onde as variáveis têm significado físico e é preservada a generalidade do sistema.

Os primeiros modelos na forma de EADs apresentados na literatura diferenciavam as equações algébricas, ou as resolviam separadamente, e as substituíam nas equações diferenciais, até transformar o sistema original em um sistema de equações diferenciais ordinárias (EDOs). Este processo, quando possível, além de consumir tempo e esforço computacional pode apresentar uma solução numérica diferente da solução do sistema original, além de perder a generalidade do sistema original. Devido a estas dificuldades tornou-se necessário utilizar outras técnicas para trabalhar diretamente com o sistema de EADs original.

As técnicas numéricas para trabalhar diretamente com o sistema original foram muito exploradas na década de 80 e, até hoje, não se conhece algum método numérico que o resolva incondicionalmente. Diante destas dificuldades tornou-se necessário determinar parâmetros para classificar sistemas de EADs, dentre estes, o mais utilizado é o índice diferencial.

Sistemas de EADs podem ser classificados como índice 0, 1 ou superior, onde aqueles que apresentam índice até 1, comportam-se de forma semelhante a sistemas de EDOs e podem ser resolvidos por métodos aplicados a esta classe. Entretanto, sistemas que apresentam índice

superior comportam-se de forma diferente das EDOs e apresentam complicações nos códigos numéricos.

Atualmente existem diversos códigos computacionais que permitem a integração direta de sistemas de índice menor que 2 como o DASSL de Petzold (1989), o RADAU de Hairer e Wanner (1991) e DASSLC de Secchi (1992). Entretanto, ainda não é possível integrar o sistema de EADs de forma direta com índice superior, com exceção de casos específicos como os sistemas lineares.

Além das dificuldades de integração, os sistemas de EADs, também apresentam dificuldade para determinar condições iniciais consistentes, sendo estas necessárias para dar partida ao código de integração. Determinar condições iniciais consistentes não é uma tarefa trivial, porque este conjunto de valores, além de satisfazer o sistema original das EADs, também deve satisfazer as derivadas de algumas de suas equações.

Neste trabalho são propostas e discutidas técnicas de inicialização consistente e integração numérica de EADs de índice qualquer, que levam em consideração as restrições adicionais do sistema estendido, resultante do processo de redução de índice. O código computacional utilizado foi o DASSLC (Secchi e Pereira, 1997), escrito em linguagem C, que realiza a integração direta de sistemas genéricos de EADs.

Para a compreensão das técnicas propostas, apresenta-se no capítulo 2 conceitos fundamentais da teoria de EADs. Neste capítulo são fornecidos conceitos de índice, caracterização e resolubilidade de EADs, assim como, a inicialização e sua importância para a resolução numérica.

No capítulo 3, apresenta-se tópicos sobre a teoria de grafos que são utilizados para ilustrar o algoritmo proposto por Pantelides (1988), que determina o conjunto mínimo de equações que precisa ser diferenciado para a inicialização consistente. No capítulo 4, as técnicas de levar em consideração as restrições adicionais são apresentadas, enfatizando-se esta necessidade para a integração do sistema através de código numérico.

Os resultados obtidos com a aplicação do código DASSLC, considerando as diferentes técnicas que levam em consideração as restrições adicionais do sistema estendido, são apresentados no capítulo 5. Para análise desses resultados utilizou-se exemplos de sistemas algébrico-diferenciais descritos na literatura.

O capítulo 6 finaliza o trabalho com as conclusões finais a respeito das técnicas propostas, que levam em consideração as restrições adicionais do sistema estendido, e sugestões para trabalhos futuros.

2. CONCEITOS BÁSICOS

Este capítulo apresenta conceitos e definições importantes para a análise e caracterização de EADs. São apresentadas algumas diferenças existentes entre EADs e EDOs, a fim de salientar que não é possível resolver todos os tipos de EADs através de métodos numéricos aplicados a sistemas de EDOs. É definido índice diferencial, sendo este o mais adotado na caracterização das EADs, embora existam outros tipos de índice; também definidos neste capítulo. São também apresentados os tipos e características de sistemas de EADs, resolubilidade e consistência da inicialização.

2.1. EAD x EDO

Sistemas composto por equações algébricas acopladas a equações diferenciais são denominados sistema de equações algébrico-diferenciais (EADs) que pode ser equacionado por:

$$F(x(t), \dot{x}(t), t) = 0, \quad (2.1)$$

onde $t \in \mathbb{R}$ é a variável independente, $x \in \mathbb{R}^n$ (vetor formado pelas variáveis diferenciais e algébricas) e $F \in \mathbb{R}$. Quando a matriz $F_{\dot{x}}$ de derivadas parciais de F com relação a \dot{x} é singular, tem-se um sistema de equações algébrico-diferenciais (EAD).

Equações diferenciais ordinárias e equações diferenciais parciais também são exemplos de uma das classes de EADs.

A equação (2.1), que está na forma implícita, quando puder ser rescrita na forma:

$$\dot{x}(t) = f(x(t), t), \quad (2.2)$$

passará a ser um sistema de Equações Diferenciais Ordinárias (EDOs) explícito.

Segundo Petzold (1982) alguns tipos de sistemas de EADs podem ser resolvidos numericamente por técnicas aplicadas a sistemas de EDOs, sendo estas bastante conhecidas. Porém, podem surgir algumas dificuldades numéricas geradas pelo código, devido à classe de

sistema de EADs também incluir problemas com determinadas propriedades, as quais são muito diferentes da forma padrão de sistema de EDOs.

Alguns desses sistemas de EADs não podem ser resolvidos por técnicas de integração que utilizam, em cada passo, as fórmulas de BDF (*Backward-Differential-formula*), muito usadas para resolver sistemas de EDOs, podendo modificar de alguma forma o conjunto de equações originais e gerar diferença na ordem da estimativa do erro de integração.

A resolução de um sistema de equações algébrico-diferenciais por eliminação das equações algébricas, transforma o sistema original em um sistema de equações diferenciais ordinárias. Este novo sistema poderá ser resolvido por métodos utilizados em sistemas de EDOs, se o sistema original não apresentar um índice elevado.

A transformação de um sistema de equações algébrico-diferenciais em um sistema de equações diferenciais ordinárias se faz necessária devido à dificuldade de se trabalhar com códigos numéricos, quando o sistema apresenta índice elevado.

Estas são algumas diferenças relacionadas a uma das classes pertencentes ao sistema de EADs; entretanto, a diferença entre EADs e EDOs está relacionada com a matriz de derivadas parciais de F com relação a \dot{x} . Na equação (2.1), quando esta é singular, o sistema é dito EAD, caso contrário o sistema será EDO. Detalhes mais precisos referentes a diferenças entre EADs e EDOs encontram-se em Brenan et al., (1989).

2.2. ÍNDICE

Se a solução de sistemas de EADs apresentar dificuldades de inicialização consistente e/ou propagação do erro de integração, este sistema possui problemas de índice, sendo este, o parâmetro mais utilizado para a caracterização de sistemas de EADs. Na literatura encontram-se quatro diferentes tipos de índices, que são descritos a seguir.

O índice de nilpotência está relacionado com a nilpotência da forma canônica de sistema de EADs lineares com coeficientes constantes, isto é:

$$A\dot{x} + Bx = f(t), \quad (2.3)$$

onde A e B são matrizes $n \times n$ reais ou complexas. Se λ é um parâmetro complexo, então:

$$IA + B, \quad (2.4)$$

é chamada de matriz pencil.

Se o determinante da matriz pencil $IA + B$ não é identicamente nulo, então $IA + B$ é dita regular e o sistema é solúvel. A regularidade da matriz pencil conduz ao seguinte teorema (Brenan et al., 1989).

Teorema 2.1 (Nilpotência) - Sendo $IA + B$ uma matriz pencil, então existe duas matrizes P e Q invertíveis, tais que:

$$PAQ = \begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix} \text{ e } PBQ = \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}. \quad (2.5)$$

Onde N é a matriz de nilpotência, I é a matriz identidade e C uma matriz na forma canônica de Jordan. A matriz N tem nilpotência k se $N^k = 0$ e $N^{k-1} \neq 0$, sendo k é grau de nilpotência de N . No caso especial em que A é não-singular, tem-se $PAQ = I$, $PBQ = C$ e define-se $k = 0$. Se o $\det(IA + B) = cte$, então $PAQ = N$ e $PBQ = I$.

A solução do sistema (2.3) é determinada pela forma canônica da matriz pencil:

$$IPAQ + PBQ, \quad (2.6)$$

onde P e Q são matrizes $n \times n$ não-singulares que através de mudanças de coordenadas transformam o sistema (2.3) para a forma canônica:

$$PAQ\dot{y} + PBQy = Pf, \quad (2.7)$$

onde $x = Qy$.

Em outras palavras, o índice das EADs lineares com coeficientes constantes pode ser definido como o número de iterações necessárias, para concluir o seguinte procedimento que, a cada iteração reduz o índice em uma unidade (Brenam, 1983):

1. Obter restrições algébricas explícitas através de reformulação de EADs por mudança de coordenadas;

2. Diferenciar estas restrições até reduzir o sistema a EDOs explícitas.

O mesmo procedimento pode ser usado para definir o índice de EADs lineares com coeficientes variáveis, desde que o posto de $A(t)$ seja constante.

Definição 2.1 (Índice de Nilpotência) - O índice de nilpotência da forma canônica de um sistema de EADs lineares de coeficientes constantes é o grau de nilpotência da matriz desta forma canônica.

Índice diferencial é uma extensão do índice de nilpotência para sistemas de EADs na forma implícita.

Considere o caso geral de EADs:

$$F(\dot{x}, x, t) = 0, \quad (2.8)$$

onde $\dot{x} = \frac{dx}{dt} \in \mathbb{R}^n$, $x \in \mathbb{R}^n$, $F \in \mathbb{R}^{2n+1}$, $t \in I$, e I é um subintervalo de \mathbb{R} .

Definição 2.2 (Índice diferencial) - Índice diferencial (v_d) é o número mínimo de vezes que um sistema de EADs (ou equações derivadas dele) precisa ser diferenciado em relação a t até ser transformado em um sistema de EDOs explícito do tipo $\dot{x} = F^*(t, x)$.

Após esta definição, sistemas compostos apenas por EDOs apresentam índice zero. Todo o sistema de EADs do tipo (2.8) é considerado de índice superior (ou elevado) se $v_d \geq 2$.

Índice de perturbação, definido por Hairer et al. (1989), é a medida da sensibilidade das soluções frente a perturbações do sistema de equações.

Definição 2.3 (Índice de perturbação) - O sistema de EADs $F(\dot{x}, x, t) = 0$ tem índice de perturbação (v_p) ao longo da solução $x(t)$ para t no intervalo I , se v_p é o menor inteiro tal que a diferença entre a solução $x(t)$ e a solução do sistema perturbado

$$F(\hat{x}, \dot{\hat{x}}, t) = \mathbf{d}(t), \quad (2.9)$$

exista no intervalo de interesse e seja limitada por

$$\max \|\hat{x}(t) - x(t)\| \leq C \left\{ \|\hat{x}(0) - x(0)\| + \max \|\mathbf{d}(t)\| + \dots + \max \|\mathbf{d}^{(v_p-1)}(t)\| \right\}. \quad (2.10)$$

O índice de perturbação é de grande importância para a determinação do impacto do erro de arredondamento na solução numérica de EADs. Uma relação entre o índice de perturbação (v_p) e o índice diferencial (v_d) dado por (Gear, 1990)

$$v_d \leq v_p \leq v_d + 1. \quad (2.11)$$

O índice de esforço numérico ou índice singular está relacionado com o esforço para obter a solução numérica de um sistema de EADs (Chung e Westerberg, 1991).

Definição 2.4 (Índice singular) – O índice singular de um sistema de EADs na forma geral $F(x, \dot{x}, y, t) = 0$, onde F é um vetor de funções suficientemente diferenciáveis, é definido como 1 (um) mais o número de diferenciações de EADs necessárias para tornar a matriz Jacobiana com relação as variáveis \dot{x} e y não singular.

Caso o índice singular do sistema seja maior do que 1, o sistema tem problema de índice, apresentando dificuldade com a consistência da inicialização e/ou com a propagação estável do erro.

A definição de índice também apresenta relação com o conceito de rigidez de sistema de EDOs. Esta rigidez é caracterizada pelo número de condicionamento k da matriz Jacobiana do sistema, ou seja, pela relação entre o maior e o menor valor característico em módulo da matriz Jacobiana. Um sistema com $k \sim 20$ é dito não rígido, $k \sim 10^3$ é dito rígido e $k \sim 10^6$ é dito muito rígido. Portanto, um sistema de EADs pode ser interpretado como um sistema de EDOs infinitamente rígido (Chung e Westerberg, 1991).

Pelo sistema da forma (2.1), um sistema de índice maior que zero apresenta singularidade da matriz de derivadas parciais $\partial F / \partial \dot{x}$, o que significa pelo menos um valor característico nulo,

logo, um outro valor característico diferente de zero, leva $k = \infty$, isto significa que o sistema é muito rígido.

Entretanto, Asher e Petzold (1998) definem rigidez em função do desempenho de um método explícito para a sua solução. Um problema de valor inicial é dito rígido se o passo de integração necessário para tornar consistente a estabilidade do método de Euler explícito é muito menor que o tamanho do passo requerido para representar a solução de forma acurada.

Sob o ponto de vista numérico, quanto menor o índice, mais simples será resolver o sistema. Todavia, para a solução numérica de índices superiores, deve-se usar uma das seguintes estratégias:

- (i) utilização de um código adequado para sistemas de índice superior;
- (ii) redução do índice do sistema para 1 ou 0.

Se o índice for até 1 (um), existem códigos computacionais robustos e consistentes para sua solução; porém, apenas os sistemas de índice menores ou iguais a 1 (um) e as EADs lineares com coeficientes constantes de índice qualquer, podem ser resolvidos por métodos de EDOs. Para índices maiores que 1 (um) ainda não foi desenvolvido um código genérico, sendo que os existentes podem levar a solução numérica a não convergência ou convergindo para uma solução incorreta. Para as EADs lineares gerais e as não lineares de índice superior, não existem métodos numéricos que as resolvam incondicionalmente.

Diante destas dificuldades, a estratégia mais conveniente e em geral a mais simples é a redução do índice do sistema para 1 (um) ou 0 (zero). O maior inconveniente para o sistema resultante, após a redução de índice, é o surgimento de mais soluções que no sistema original, o que pode afetar a convergência do método numérico, (Rascol et al., 1998). Outro problema resultante com a diferenciação das equações algébricas é a perda das constantes, o que pode ter implicações graves na qualidade da solução numérica.

Além disto, ao longo do processo dessas diferenciações podem surgir outras equações algébricas que estavam implícitas no sistema de equações originais. O conjunto destas equações, formado pelo sistema original (equações algébricas e diferenciais) e as equações resultantes das diferenciações, compõem o chamado sistema estendido.

2.3. CARACTERIZAÇÃO ESTRUTURAL DE UM SISTEMA DE EADs

Um sistema de EADs implícito da forma $F(x(t), \dot{x}(t), t) = 0$, onde F e x possuem a mesma dimensão, pode ser classificado segundo propriedades estruturais de suas matrizes

jacobianas $\partial F/\partial x$ e $\partial F/\partial \dot{x}$, como posto e estrutura de esparsidade. Uma propriedade de uma matriz é dita estrutural se todos os seus elementos não nulos podem ser substituídos por valores aleatórios e ainda assim a propriedade persistir.

Estas informações são de grande importância para o estudo das propriedades analíticas do sistema, análise da convergência e estabilidade dos métodos numéricos. A seguir são apresentados os diferentes tipos de sistemas expostos na literatura.

Sistema implícito

O sistema de EADs representado abaixo corresponde ao sistema implícito do tipo não linear:

$$F(x(t), \dot{x}(t), t) = 0, \quad (2.12)$$

onde $t \in \mathbb{R}$, x e $F \in \mathbb{R}^n$ são funções suficientemente diferenciáveis, onde a matriz das derivadas parciais $\partial F/\partial \dot{x}$ é singular.

O sistema de EADs representado abaixo correspondem, respectivamente, ao sistema implícito do tipo linear com coeficientes variáveis e linear com coeficientes constantes:

$$A(t)\dot{x}(t) + B(t)x(t) = f(t), \quad (2.13)$$

$$A\dot{x}(t) + Bx(t) = f(t), \quad (2.14)$$

onde A e B são matrizes quadradas de números reais ou complexos, t é uma variável real e A é singular para todo o t real (Brenan et al., 1989).

Sistema linearmente implícito

O sistema é dito linearmente implícito quando é do tipo não linear, porém linear na derivada, representado por:

$$A(t, x)\dot{x} + f(t, x) = 0. \quad (2.15)$$

Sistema particionado

O sistema particionado é representado pela união de equações diferenciais (2.16) e equações algébricas (2.17). As variáveis também podem ser divididas em variáveis cujas derivadas aparecem ou não no sistema de EADs, definidas como variáveis diferenciais (x) e variáveis algébricas (y). Assim, tem-se:

$$f(x(t), \dot{x}(t), y(t)) = 0, \quad (2.16)$$

$$g(x(t), y(t)) = 0. \quad (2.17)$$

Sistema semi-implícito

O sistema semi-implícito, que consiste de um sistema de EADs particionado composto por um sistema de EDOs explícito sujeito a restrições algébricas, apresenta propriedades particulares onde o número nd de equações diferenciais (f) é igual ao número nx de variáveis diferenciais (x) e o número na de equações algébricas (g) é igual ao número ny de variáveis algébricas (y), representado na forma geral por :

$$\dot{x}(t) = f(t, x, y), \quad (2.18)$$

$$g(t, x, y) = 0. \quad (2.19)$$

Sistema de Hessenberg

O sistema de Hessenberg é uma estrutura particular de EADs semi-implícita com propriedades conhecidas. Muitos dos sistemas de índice superior que surgem de aplicações, estão sob a chamada forma de Hessenberg, representado pelo sistemas (2.20) de tamanho r :

$$\begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ \cdot & \cdot & I & \cdot & \cdot \\ \cdot & \cdot & \cdot & I & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 \end{bmatrix} \begin{bmatrix} x_1' \\ \cdot \\ \cdot \\ \cdot \\ x_r' \end{bmatrix} + \begin{bmatrix} B_{11} & * & * & B_{1,r-1} & B_{1r} \\ B_{21} & * & * & B_{2,r-1} & 0 \\ 0 & * & * & * & \cdot \\ \cdot & \cdot & * & * & \cdot \\ 0 & \cdot & 0 & B_{r,r-1} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_r \end{bmatrix} = \begin{bmatrix} f_1 \\ \cdot \\ \cdot \\ \cdot \\ f_r \end{bmatrix}, \quad (2.20)$$

onde a matriz resultante do produto $B_{r,r-1}B_{r-1,r-2}\dots B_{1r}$ é quadrada e regular.

As formas de Hessenberg para sistemas lineares para $r=2$ e $r=3$ são mostradas abaixo:

$$\begin{aligned} \dot{x}_1 + B_{11}x_1 + B_{12}x_2 &= f_1, \\ B_{21}x_1 &= f, \end{aligned} \quad (2.21)$$

onde a matriz $B_{21}B_{12}$ é quadrada e regular e,

$$\begin{aligned} \dot{x}_1 + B_{11}x_1 + B_{12}x_2 + B_{13}x_3 &= f_1, \\ \dot{x}_2 + B_{21}x_1 + B_{22}x_2 &= f_2, \\ B_{32}x_2 &= f_3, \end{aligned} \quad (2.22)$$

onde a matriz $B_{32}B_{21}B_{13}$ também é quadrada e regular.

A forma de Hessenberg para sistemas não lineares de tamanho r é mostrada abaixo:

$$\begin{aligned} \dot{x}_1 &= F_1(x_1, x_2, \dots, x_r, t), \\ \dot{x}_2 &= F_2(x_1, x_2, \dots, x_{r-1}, t), \\ &\dots \\ \dot{x}_i &= F_i(x_{i-1}, x_i, \dots, x_{r-1}, t), \\ &\dots \\ 0 &= F_r(x_{r-1}, t), \end{aligned} \quad (2.23)$$

onde $(\partial F_r / \partial x_{r-1}) \cdot (\partial F_{r-1} / \partial x_{r-2}) \dots (\partial F_2 / \partial x_1) \cdot (\partial F_1 / \partial x_r)$ é não singular.

As formas de Hessenberg para sistemas não lineares para $r=2$ e $r=3$ são mostradas abaixo:

$$\begin{aligned}\dot{x}_1 &= F_1(x_1, x_2, t), \\ 0 &= F_2(x_1, t),\end{aligned}\tag{2.24}$$

onde $(\partial F_2 / \partial x_1)(\partial F_1 / \partial x_2)$ é regular.

$$\begin{aligned}\dot{x}_1 &= F_1(x_1, x_2, x_3, t), \\ \dot{x}_2 &= F_2(x_1, x_2, t), \\ 0 &= F_3(x_2, t),\end{aligned}\tag{2.25}$$

onde $(\partial F_3 / \partial x_2)(\partial F_2 / \partial x_1)(\partial F_1 / \partial x_3)$ é regular.

Sistema reduzido

Sistemas de EADs de índice 1 (um) ou zero resultante do processo de redução de índice.

Sistema estendido correspondente

O sistema estendido correspondente é o sistema formado pelas equações originais e por todas as equações resultantes, após as diferenciações necessárias do sistema original, para transformá-lo em um sistema de índice 1 ou zero. Isto é, é a união do sistema original com o sistema reduzido e as demais equações intermediárias geradas no processo de redução de índice.

2.4. RESOLUBILIDADE

A resolubilidade de um sistema de EADs garante que a solução exista e seja única, no intervalo de interesse, e revela a dependência entre tal existência e o valor inicial da variável dependente.

EADs lineares com coeficientes constantes, representada por:

$$A\dot{x}(t) + Bx(t) = f(t),\tag{2.26}$$

onde A e B são matrizes $n \times n$, são solúveis se, e somente se, $IA + B$ é uma matriz pencil regular, isto é, o determinante do feixe de matrizes não é identicamente nulo como uma função do parâmetro complexo I . A regularidade da matriz pencil implica na existência de outras duas matrizes não singulares, P e Q , tal que:

$$PAQ = \begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix} \text{ e } PBQ = \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}, \quad (2.27)$$

onde N é a matriz de nilpotência, I é a matriz identidade e C uma matriz na forma canônica de Jordan.

Aplicando mudanças de coordenadas $Qy = x$ em (2.26) e pré-multiplicando-o por P , obtemos as seguintes equações:

$$\dot{y}_1 + Cy_1 = P_1 f_1, \quad (2.28)$$

$$Ny_2 + y_2 = P_2 f_2, \quad (2.29)$$

os sistemas (2.26) e (2.28) - (2.29) são chamados analiticamente equivalentes.

A equação (2.28) é um sistema de EDOs cuja solução existe para um valor inicial do vetor x_1 e funções f_1 contínuas. Para a equação (2.29) há somente uma solução e esta deve satisfazer a parte algébrica para que seja consistente.

EADs lineares com coeficientes variáveis, representados por:

$$A(t)\dot{x}(t) + B(t)x(t) = f(t), \quad (2.30)$$

diferem das EADs lineares com coeficientes constantes (2.26) por não apresentarem equivalência entre a resolubilidade e regularidade do feixe de matrizes, que são conceitos independentes.

Um sistema de EADs lineares com coeficientes variáveis, como a equação (2.30), é analiticamente solucionável no intervalo $I = [0, T]$, se para $f(t)$ suficientemente suave as soluções existem e são únicas para tais valores, para qualquer $t_0 \in I$. Quando o sistema é

analiticamente solúvel pode-se usar mudanças de coordenadas $x = Q(t)y$ e multiplicar a esquerda o sistema (2.30) por $P(t)$, obtendo:

$$P(t)A(t)\dot{Q}(t)yQ(t)\dot{y}(t) + [P(t)B(t)Q(t) + P(t)A(t)\dot{Q}(t)]y(t) = P(t)f(t), \quad (2.31)$$

onde $P(t)$ e $Q(t)$ são não singulares, no intervalo de interesse, podendo ser colocadas na forma Canônica Padrão:

$$\dot{y}_1 + C(t)y_1 = P_1f_1, \quad (2.32)$$

$$N(t)\dot{y}_2 + y_2 = P_2f_2, \quad (2.33)$$

onde $N(t)$ é nilpotente e triangular inferior (ou superior). Caso N seja uma matriz constante, esta é dita Forma Canônica Padrão Forte.

O sistema dado pela equação (2.30) é solúvel, se ele for analiticamente equivalente ao sistema na Forma Canônica Padrão, equações (2.32) e (2.33). Porém, nem todo o problema solucionável pode ser colocado na Forma Canônica Padrão.

Para EADs não lineares representadas por:

$$F(x(t), \dot{x}(t), t) = 0, \quad (2.34)$$

a regularidade do feixe de matriz $IF_{\dot{x}} + F_x$ não é uma condição necessária nem suficiente para a resolubilidade do sistema.

A verificação prática da resolubilidade das EADs não lineares foi analisada por Pantelides e Barton (1992), baseada na estrutura da matriz $IF_{\dot{x}} + F_x$. Esta matriz é usada nos algoritmos de integração numérica durante a solução do conjunto de equações não lineares para cada passo de integração, na qual I é inversamente proporcional ao tamanho do passo de integração. Caso a matriz seja singular, os métodos numéricos possivelmente falham e as EADs não serão resolvidas. Esta singularidade pode ser evidenciada a partir do seu padrão (pat(M)) – representação de uma matriz M , obtida pela substituição de seus elementos não nulos por *

Unger et al. (1995), independente da fórmula simbólica ou dos elementos da matriz. A regularidade não garante que ela não seja identicamente ou localmente singular.

Todo o sistema não linear pode ser organizado em alguma forma conhecida, isto é, pode por exemplo estar na forma Hessenberg ou Forma Canônica Padrão. Os sistemas de Hessenberg de dimensão r são solúveis, admitindo-se que as funções F_i (equações (2.23)) são suficientemente diferenciáveis.

2.5. INICIALIZAÇÃO

Certamente a etapa mais difícil da solução de um sistema de EADs é a determinação das condições iniciais. Estas devem ser consistentes e fisicamente corretas para que o trabalho de estudo tenha êxito no processo final. Os valores que satisfazem as equações iniciais e o sistema estendido são chamados de consistentes.

Considere o sistema EADs (2.8), as condições iniciais são os valores de \dot{x} e x para algum tempo t_0 , denotados por \dot{x}_0 e x_0 . Para que as condições iniciais sejam consistentes com o sistema (2.8) elas devem satisfazer:

$$F(\dot{x}_0, x_0, t_0) = 0 . \quad (2.35)$$

Definição 2.5. (Condições iniciais) – Os vetores \dot{x}_0 e x_0 são chamados de condições iniciais consistentes para o sistema $F(\dot{x}, x, t) = 0$ de EADs para t_0 se eles satisfazem o correspondente sistema estendido para t_0 .

A definição 2.5 implica que o estado inicial deve ser solução não apenas do sistema de EADs original, mas também de todas as equações intermediárias geradas quando o sistema é convertido em um sistema de EDOs (ou EADs de índice 1), em que $t = t_0$.

O número de variáveis que podem ser fixadas arbitrariamente como valores iniciais, ao longo da inicialização consistente, de um sistema de EADs é dito grau de liberdade. O grau de liberdade m é encontrado através da relação entre as ne equações do sistema estendido e as ni incógnitas deste sistema, isto é, $m = ni - ne$. O grau de liberdade m é inferior ou igual ao

número de variáveis do sistema. Poderia-se pensar, erroneamente, que o número de graus de liberdade do sistema é igual ao número de variáveis diferenciais. Entretanto, este valor é apenas um limite superior (Costa Jr., 2001), sendo sempre igual para sistemas de EDOs.

Considere o seguinte sistema de índice diferencial 2 (Martinson et al., 2000):

$$\dot{u}_1 - u_2 = 0, \quad (2.36)$$

$$u_1 = f(t). \quad (2.37)$$

Existe uma constante implícita na equação (2.37). Derivando-se a equação (2.37), tal constante é perdida (se a equação (2.37) não for considerada):

$$\dot{u}_1 = \dot{f}(t). \quad (2.38)$$

Substituindo-se a equação (2.38) na equação (2.36), encontra-se uma nova equação:

$$u_2 = \dot{f}(t). \quad (2.39)$$

Por fim, derivando-se a equação (2.39) obtém-se a equação (com mais uma provável perda de constante se a equação (2.39) ou (2.36) não for considerada):

$$\dot{u}_2 = \ddot{f}(t). \quad (2.40)$$

Para que o sistema de EADs, descrito acima, apresente condições iniciais consistentes, deve satisfazer não somente o sistema original (2.36) e (2.37), mas também o sistema (2.38) e (2.40), formando assim, o sistema estendido. O sistema apresenta índice 2, pois foram necessárias 2 derivações para transformá-lo em um sistema de EDOs, e existem 4 equações - (2.36), (2.37), (2.38) e (2.40), e 4 incógnitas: $\dot{u}_1(t), u_1(t), \dot{u}_2(t)$ e $u_2(t)$. Desta forma, o conjunto de equações do sistema não apresenta grau de liberdade algum. Caso o processo de diferenciação parasse na sistema de EADs de índice 1, teria-se 3 equações (2.36), (2.37) e (2.38) e 3 incógnitas: $\dot{u}_1(t), u_1(t)$ e $u_2(t)$, ou seja, nenhum grau de liberdade.

Observa-se que se o sistema formados pelas equações (2.38) e (2.40) fosse resolvido para condições iniciais arbitrárias (sistema de EDOs, resultante da redução de índice), a solução

obtida poderia não satisfazer as equações (2.36) e (2.37) , devido a perda das constantes no processo de diferenciação.

Sistema de índice superior, como o descrito acima, sempre geram problemas de inicialização. Estes problemas aparecem devido à exigência da consistência para o sistema estendido, e não apenas para o sistema original. Até mesmo os sistemas de índice 1 podem apresentar equações algébricas implícitas.

Nem sempre apenas manipular o sistema é suficiente. Caso o sistema apresente mais variáveis diferenciais que equações diferenciais, é impossível determinar $\dot{x}(t_0)$ apenas por manipulações algébricas do sistema, tornando-se necessário diferenciar algumas ou todas as equações algébricas. Desta forma, procura-se extrair do sistema informações fundamentais que estão nele contidas, mas que não são explícitas.

Códigos numéricos apresentados para a solução das EADs freqüentemente falham ou tornam-se ineficientes, quando os valores das condições iniciais são inconsistentes com o sistema de EADs. Alguns pacotes de caracterização são encontrados na literatura, porém não há código computacional ou método numérico algum que se aplique a todo e qualquer sistema algébrico diferencial.

Petzold (1982), em seu código DASSL, aplicado a EADs gerais de índice 1, conduz a retro-diferenciação de 1ª ordem (Euler) e leva a iterações de Newton para determinar as condições iniciais consistentes. Esta proposta foi lançada inicialmente por Gear (1971). Campbell (1986) sugere aproximações baseadas na expansão em série de Taylor, para determinar condições iniciais consistentes para problemas lineares com coeficientes variáveis. Entretanto, estes métodos são limitados por exigirem um grande conhecimento das derivadas das matrizes do sistema (Leimkuhler et al., 1991). Campbell (1988) implementou seu método para problemas não lineares, mas não garante a solução para problemas complexos devido a matriz Jacobiana não ser conhecida, ou ser difícil de calcular.

Pantelides (1988) em seu algoritmo estrutural, baseado na teoria de grafos, determina o menor subconjunto de equações que, estruturalmente diferenciadas, fornece informações da ordem da solução para garantir a consistência dos valores iniciais.

Outras ferramentas de caracterização de sistemas de EADs são os programas INDEX1.0 de Murata (1996), baseado em análise simbólica, e os pacotes ALGO e PALG de Unger et al. (1995), de caráter estrutural.

Atualmente o código mais utilizado para solução numérica de EADs é o código DASSL de Petzold (1989) devido a sua robustez e a aplicabilidade a sistemas implícitos. O código

DASSL, desenvolvido em linguagem Fortran, emprega fórmulas do tipo BDF de ordem variável de 1 a 5, e se destina a sistemas implícitos de índice 1 ou sistemas semi-implícitos de índice 2.

Neste trabalho é usado o código DASSLC v2.0 de Secchi e Pereira (1997), desenvolvido em linguagem C, que é uma extensão do código DASSL para sistemas esparsos em geral.

3. REDUÇÃO DE ÍNDICE

Segundo Gear e Petzold (1984), sistemas de índice zero ou 1 (um) e EADs lineares com coeficientes constantes podem ser resolvidas por métodos numéricos para EDOs, sem alterar seu índice. Entretanto, a redução de índice se faz necessária para solução numérica de sistemas de índices superiores. Segundo Williams et al. (2000), problemas de índice superior ocorrem quando são introduzidas equações algébricas que interferem no valor das variáveis diferenciais.

O inconveniente do processo de redução de índice é o surgimento de novas variáveis, ou a perda de constantes existentes no sistema original, assim como o aumento do número de equações do sistema seguinte, Rascol et al. (1998).

O simples fato das equações algébricas do sistema de EADs tornarem-se equações implícitas no novo sistema de EDOs, conduz a instabilidade dos códigos numéricos. Devido a esta instabilidade, vários códigos numéricos que utilizam a redução de índice usam as chamadas técnicas de estabilização das restrições, necessitando analisar e caracterizar o sistema de EADs para obter condições iniciais eficientes e consistentes com a solução numérica. Contudo, nem todo o código de redução tem etapas de diferenciação ou apresenta sistema de maior dimensão.

Encontram-se na literatura algumas técnicas de redução de índice. Brenan et al. (1989), sugere a substituição de variáveis cujas derivadas não são explícitas, por novas variáveis diferenciais para reduzir seu índice. No exemplo a seguir, aplica-se a técnica descrita acima para um sistema de EADs lineares com coeficientes constantes de índice diferencial 3:

$$\begin{aligned}\dot{x}_2 &= x_1 + f_1(t), \\ \dot{x}_3 &= x_2 + f_2(t), \\ 0 &= x_3 + f_3(t).\end{aligned}\tag{3.1}$$

Fazendo a troca de variáveis $\dot{z}_1 = x_1$ o sistema passa a ter índice diferencial 2:

$$\begin{aligned}\dot{x}_2 - \dot{z}_1 &= f_1(t), \\ \dot{x}_3 &= x_2 + f_2(t), \\ 0 &= x_3 + f_3(t).\end{aligned}\tag{3.2}$$

O novo sistema (3.2) tem mais soluções que o sistema original (3.1).

Gear (1988), propõe a redução de índice pelo uso de operações estruturais. No seu algoritmo, as matrizes são representadas pelos seus padrões ($pad(M)$), onde os elementos são zero (0) ou um (1). Desta forma, o padrão da matriz jacobiana de um sistema, $pad(J)$, indica apenas se uma determinada variável está ou não presente em uma determinada equação, deixando de informar as expressões exatas.

Gear (1989), descrito por Bauchmann et al. (1990), utiliza a diferenciação das restrições algébricas com relação ao tempo e a linearidade das novas equações, introduzidas pela diferenciação das derivadas de índice superior, para eliminar os termos nos quais estas derivadas de ordem superior aparecem. As novas equações estão em função das variáveis de estado e das variáveis algébricas e são incorporadas ao sistema no algoritmo.

Pantelides (1988), propõe um algoritmo baseado na teoria de grafos através da estrutura das equações do sistema. O algoritmo determina o menor subconjunto de equações que, estruturalmente diferenciadas, fornece a consistência do sistema. Unger et al. (1995), implementa uma versão do algoritmo de Pantelides, baseado somente na análise estrutural para a esparsidade padrão do sistema jacobiano $\partial F/\partial x$ e $\partial F/\partial \dot{x}$.

O algoritmo apresentado por Unger et al. (1995), conhecido por PALG, determina apenas se o feixe estrutural ($pad(\partial F/\partial \dot{x})+pad(\partial F/\partial x)$) é estruturalmente não singular e indica o número de diferenciações que deve ser aplicada a cada equação do sistema, até que seja possível a inicialização consistente. Se um modelo for rejeitado neste teste, ele provavelmente não poderá ser resolvido numericamente, devido à singularidade da matriz de iteração.

3.1 TEORIA DE GRAFOS

A inicialização de um sistema de EADs na forma geral:

$$F(x, \dot{x}, y, t) = 0, \quad (3.3)$$

onde $x, \dot{x} \in \mathbb{R}^{nd}$ são as variáveis diferenciais e suas derivadas e $y \in \mathbb{R}^{na}$ são as variáveis algébricas com $n = na + nd$, é considerada consistente, se as condições iniciais (x_0, \dot{x}_0, y_0)

satisfizerem não somente o sistema (3.3), mas também as equações intermediárias provenientes da geração do sistema estendido. O sistema original adicionado ao sistema resultante obtido da transformação das EADs em EDOs (ou EADs de índice 1), compõe o sistema estendido.

O sistema estendido aumenta o número de equações resultantes em relação ao sistema original. Este aumento de equações determina o grau de liberdade do sistema inicial, informando o número de variáveis que podem ser escolhidas arbitrariamente na inicialização. O exemplo do condensador com volume constante e massa de líquido desprezível, apresentado por Pantelides et al. (1988), exemplifica o sistema estendido. Seja o sistema:

$$\dot{M}_v = F - L, \quad (3.4)$$

$$M_v \cdot C_p \cdot \dot{T} = F \cdot C_p \cdot (T_0 - T) + I \cdot L - U \cdot S \cdot (T - T_c), \quad (3.5)$$

$$p \cdot V = M_v \cdot R \cdot T, \quad (3.6)$$

$$p = A \cdot \exp(-B/T), \quad (3.7)$$

onde:

- M_v número de moles de vapor no condensador;
- T, p temperatura e pressão no condensador;
- L taxa de saída do líquido;
- u variáveis de projeto (conhecidas em qualquer t), que representam respectivamente a vazão do vapor que entra no condensador, a temperatura de entrada e a temperatura na camisa de resfriamento; $u \equiv [F \ T_0 \ T_c]^T$;
- U, S coeficiente de transferência de calor e a área de troca térmica (conhecidos);
- V volume do condensador;
- R, A e B constantes dadas,
- C_p calor específico
- I calor latente de vaporização

O sistema do condensador apresenta índice igual a 2, pois são necessárias duas derivações para transformá-lo em um sistema de EDOs. A primeira diferenciação, através das equações (3.6) e (3.7), fornece, respectivamente:

$$\dot{p}.V = \dot{M}_v.R.T + M.R.\dot{T}, \quad (3.8)$$

$$\dot{p} = \frac{AB}{T^2} \cdot \exp\left(\frac{-B}{T}\right) \dot{T}. \quad (3.9)$$

Substituindo-se na equação (3.8) as expressões \dot{M}_v, \dot{T} e \dot{p} , obtidas das equações (3.4), (3.5) e (3.9) respectivamente, obtém-se a equação:

$$L = \Theta(M_v, T, t). \quad (3.10)$$

onde a função $\Theta(M_v, T, t)$ é o resultado desta manipulação algébrica.

A segunda diferenciação é aplicada a equação algébrica (3.10), para obtenção de (3.11):

$$\dot{L} = \frac{\partial \Theta}{\partial M_v} \dot{M}_v + \frac{\partial \Theta}{\partial T} \dot{T} + \dot{\Theta}. \quad (3.11)$$

Durante o processo de transformação das EADs em EDOs, as equações algébricas, as EDOs explícitas e as equações algébricas originais, formam o chamado sistema estendido. O sistema estendido é formado pelas equações (3.4), (3.5), (3.6), (3.7), (3.8), (3.10) e (3.11) e possui oito variáveis: $M_v, T, p, L, \dot{M}_v, \dot{T}, \dot{L}$ e \dot{p} , e portanto apresenta apenas 1 grau de liberdade.

Como os métodos numéricos consistentes resolvem sistemas de EADs de índice 1, o algoritmo de Pantelides reduz o sistema original para um sistema de índice 1, não havendo a necessidade de mais uma diferenciação. No exemplo acima, o sistema estendido, gerado pelo algoritmo, é formado pelas equações (3.4), (3.5), (3.6), (3.7), (3.8) e (3.10) e possui sete variáveis: $M_v, T, p, L, \dot{M}_v, \dot{T}$ e \dot{p} , apresentando apenas um grau de liberdade.

A obtenção do sistema estendido através de processo manual é uma tarefa bastante cansativa, na qual, muitas vezes, tal procedimento executa muitas diferenciações desnecessárias para o estabelecimento das condições iniciais. O algoritmo de Pantelides (1988), baseado na

teoria de grafos, determina quais equações devem ser diferenciadas para formar o sistema estendido, determinando o menor subconjunto de equações que estruturalmente diferenciadas, fornece um sistema consistente.

Ao diferenciar um subconjunto de equações do sistema original, o número de variáveis novas é igual ao número de variáveis existentes no subconjunto diferenciado, cujas derivadas não estão presentes no sistema original, isto é, subconjuntos de \dot{x} e y . Caso, após a diferenciação do subconjunto de equações, o sistema apresentar um número de variáveis menor que o número de equações do subconjunto, o sistema apresentará mais equações que variáveis novas, apresentando certas restrições que devem ser determinadas de modo a resolver o sistema.

Definindo-se o vetor $z = (\dot{x}, y)^T$, o sistema (3.3), pode ser rescrito, como:

$$\bar{f}(x, z, t) = 0. \quad (3.12)$$

Considerando-se um subconjunto genérico com k equações do sistema (3.12) definido por:

$$\bar{f}(\bar{x}, \bar{z}, t) = 0, \quad (3.13)$$

onde $\bar{f} \in \mathbb{R}^k$ e $\bar{x} \in \mathbb{R}^q$ e $\bar{z} \in \mathbb{R}^l$ são vetores derivados de x e z , respectivamente, tem-se que $q \leq nd$ e $l \leq n$.

O subconjunto genérico (3.13) deve ser analisado a fim de obter equações escondidas. Considerando-se que a matriz $(\bar{f}_{\bar{x}} | \bar{f}_{\bar{z}})$ possui posto completo em linhas e diferenciando-se as equações que compõe o sistema (3.13), tem-se:

$$\bar{f}_{\bar{x}} \dot{\bar{x}} + \bar{f}_{\bar{z}} \dot{\bar{z}} + \bar{f}_t = 0. \quad (3.14)$$

Sendo o posto de $\bar{f}_{\bar{z}}$ igual a r , tem-se:

$$r \leq \min(k, l). \quad (3.15)$$

Após operações algébricas apropriadas o sistema (3.14) pode ser reduzido para a forma:

$$A\dot{\bar{x}} + B\dot{\bar{z}} = a, \quad (3.16)$$

$$C\dot{\bar{x}} = b, \quad (3.17)$$

onde $A \in \mathbb{R}^{r \times q}$, $B \in \mathbb{R}^{r \times l}$, $C \in \mathbb{R}^{k-r \times q}$ são matrizes e $a \in \mathbb{R}^r$, $b \in \mathbb{R}^{k-r}$ são vetores, todos funções de (\bar{x}, \bar{z}, t) .

A diferenciação introduziu novas variáveis $\dot{\bar{z}}$ e novas equações que não apareciam no sistema original (3.12). Devido a equação (3.15), pode-se afirmar que $r \leq l$ e desta forma, o vetor $\dot{\bar{z}}$ pode ser particionado em $\dot{\bar{z}}_1 \in \mathbb{R}^r$ e $\dot{\bar{z}}_2 \in \mathbb{R}^{l-r}$. A matriz B particionada gera $B_1 \in \mathbb{R}^{r \times r}$ e $B_2 \in \mathbb{R}^{r \times l-r}$, tal que B_1 é não singular. Observando estes procedimentos o vetor $\dot{\bar{z}}_1$ pode ser explicitado pela resolução do sistema (3.16):

$$\dot{\bar{z}}_1 = B_1^{-1}a - B_1^{-1}(A\dot{\bar{x}} + B_2\dot{\bar{z}}_2). \quad (3.18)$$

Portanto, para algum conjunto de valores das variáveis $(\bar{x}, \dot{\bar{x}}, \bar{y})$, pode-se determinar os valores das novas variáveis de $\dot{\bar{z}}$, tal que a equação (3.16) seja satisfeita, não fornecendo nenhuma restrição adicional às variáveis do sistema original.

Contudo, pela consideração de que $(\bar{f}_{\bar{x}} | \bar{f}_{\bar{z}})$ possui posto k , a matriz C na equação (3.17) tem posto completo $k-r$, constituindo $(-)$ novas equações, que as variáveis originais devem satisfazer juntamente com a equação (3.12).

Desta forma, pode-se concluir que as equações “escondidas”, necessárias para a inicialização consistente, podem ser determinadas pela localização de todos os subconjuntos de k equações, tais que:

$$\text{Posto} \left(\frac{\partial \bar{f}}{\partial \bar{z}} \right) < k. \quad (3.19)$$

O algoritmo, usado para analisar as propriedades das EADs e gerar os subconjuntos com a propriedade (3.19), é baseado na observação de que uma condição suficiente para que um conjunto tenha tal propriedade é que:

$$l < k . \tag{3.20}$$

ou seja, que ele seja estruturalmente singular com relação ao subconjunto de variáveis $\{\bar{z}\}$.

O subconjunto de equações que satisfaz a equação (3.20) é chamado estruturalmente singular com respeito ao vetor z . Se além dessa condição, nenhum dos subconjuntos de equações desse subconjunto for estruturalmente singular, o subconjunto é chamado estruturalmente minimamente singular (SEM).

Antes da descrição do algoritmo de Pantelides faz-se necessário algumas definições:

Definição 3.1- Grafo É uma dupla ordenada $G=(V,A)$, onde :

$V=\{1,\dots,N\}$ é um conjunto de vértices;

$A=\{(u,v)/ u,v \in V, u \neq v\}$ é um conjunto de arestas.

Definição 3.2- Grafo Dirigido(ou dígrafo): Um grafo é dito direcionado quando existe uma direção associada a cada aresta, desta forma, as arestas são formadas por pares ordenados de (u,v) , sendo u a cauda e v a cabeça da aresta. Assim pode-se dizer que a aresta (u,v) sai do vértice u e entrar no vértice v .

Na Figura 3.1, podemos considerar $G=(V,A)$, onde $V=\{\alpha, \beta, \gamma, \delta\}$ e $A=\{(\alpha, \beta), (\alpha, \gamma), (\beta, \delta), (\gamma, \alpha), (\gamma, \delta), (\gamma, \delta)\}$.

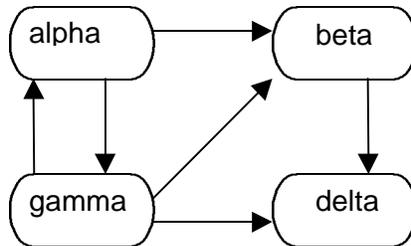


Figura 3.1- Grafo dirigido

Observa-se na Figura 3.1 que os vértices α e γ são vértices adjacentes pelo fato de seus extremos pertencerem a uma mesma aresta. Neste mesmo exemplo pode-se extrair a definição de caminho.

Definição 3.3- Caminho: um caminho de um vértice n_0 a um vértice n_k é uma seqüência $n_0, a_0, n_1, a_1, \dots, n_{k-1}, a_{k-1}, n_k$ de vértices e arestas onde, para cada i , os extremos das arestas a_i são os vértices n_i e n_{i+1} .

Definição 3.4 – Associação: Conjunto de arestas $(i - j)$ cujos nós i ou j aparecem em apenas uma das arestas, chamadas arestas combinadas

Definição 3.5 – Nó exposto: Nó que não aparece em nenhuma aresta combinada.

Definição 3.6 – Caminho estendido: é aquele caminho que possui nós expostos nas duas extremidades e arestas combinadas e não combinadas entre eles.

Definição 3.7 – Associação completa: é aquela associação que não deixa nós expostos.

Definição 3.8 - Associação parcial: é aquela associação que deixa algum nó exposto.

Definição 3.9- Grafo Bipartido: é o grafo $G=(\{E,V\},A)$, onde:

E – nós – conjunto de nós de um grafo referentes às equações $\{F\}$;

V – nós – conjunto de nós de um gráfico referente às variáveis $\{\dot{x}, y\}$;

$A=\{(f,x)/f \in E \text{ e } x \in V\}$ é um conjunto de arestas.

Para este exemplo $E = \{f_1, f_2, f_3, f_4, f_5\}$ e $V = \{\dot{x}, \dot{y}, \dot{w}, \dot{z}, T\}$. O grafo bipartido é aquele mostrada na figura 3.2:

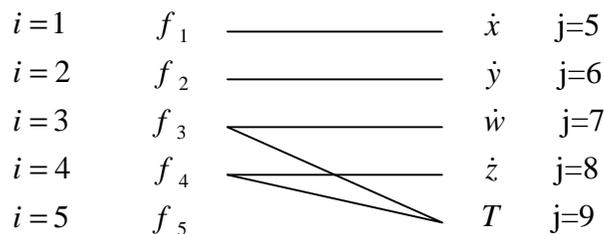


Figura 3.2 –Grafo bipartido

O algoritmo descrito por Pantelides (1988) utiliza recursivamente o algoritmo de busca em primeira profundidade, Duff et al. (1981), denominado AUGMENTPATH. O processo de associação AUGMENTPATH(i , PATHFOUND) retorna VERDADEIRO caso exista um caminho estendido que começa em i . Neste caso nenhuma diferenciação é necessária. Caso AUGMENTPATH(i , PATHFOUND) retorne FALSO o caminho não existe, neste momento os E

– nós e os V – nós são chamados coloridos ou não coloridos, conforme o processo. O subconjunto colorido é estruturalmente minimamente singular em relação a z e deve ser diferenciado. O vetor *ASSIGN*, no procedimento abaixo, contém as associações variável-equação, ou seja, *ASSIGN(j)* contém a equação à qual a variável j está associada. Se o valor de *ASSIGN(j)* é zero, então o correspondente nó de V está exposto.

Algoritmo A0 (Duff, 1981)

Procedimento AUGMENTPATH (i,PATHFOUND)

(1) Colorir E-nó i

(2) Se existir um V -nó tal que a aresta (i-j) exista e *ASSIGN(j)* = 0 então

(2a) PATHFOUND = Verdadeiro

(2b) *ASSIGN(j)* = i

(2c) Retornar

(3) Para cada j tal que a aresta (i-j) exista e j não está colorido

(3a) Colorir V -nó j

(3b) $k = \text{ASSIGN}(j)$

(3c) AUGMENTPATH (k,PATHFOUND)

(3d) Se PATHFOUND = Verdadeiro então

(3d.1) *ASSIGN(j)* = i

(3d.2) Retornar

(4) Retornar

O algoritmo A_1 , a seguir, detecta os subconjuntos de equações que são necessárias diferenciar. Para inicializar este processo é necessário o grafo bipartido $E-V$, os valores de N (sistema de equações), os valores de M (conjunto de variáveis) e a lista de diferenciações das variáveis.

A lista de diferenciações das variáveis é a relação entre as variáveis e suas derivadas com relação ao tempo, definida como:

$$A(j) = \begin{cases} k & \text{se } x_k = \frac{dx_j}{dt}, 1 \leq j, k \leq 2nd + na \\ 0, & \text{caso contrário} \end{cases} \quad (3.21)$$

Utilizando o exemplo do pêndulo descrito por Pantelides 1988, composto pelas equações f_1, f_2, f_3, f_4, f_5 :

$$(f_1) \quad \dot{x} = w, \quad (3.22)$$

$$(f_2) \quad \dot{y} = z, \quad (3.23)$$

$$(f_3) \quad \dot{w} = T \cdot x, \quad (3.24)$$

$$(f_4) \quad \dot{z} = T \cdot y - g, \quad (3.25)$$

$$(f_5) \quad 0 = x^2 + y^2 - L^2, \quad (3.26)$$

onde:

x, y posições vertical e horizontal do pêndulo;

w, z componentes da velocidade nestas direções;

T tensão na haste,

L comprimento da haste,

g gravidade

o sistema apresenta um número inicial de equações $N = nd + na = 5$, sendo $nd = 4$ equações diferenciais e $na = 1$ equação algébrica e $M = 2nd + na = 9$ incógnitas.

O vetor X , que contém todas as variáveis desconhecidas é dado por:

$$X = (x, y, w, z, \dot{x}, \dot{y}, \dot{w}, \dot{z}, T).$$

A Figura 3.2 é a representação do grafo $\{f\} - \{x\}$, onde uma aresta $(i - j)$ existe se a equação i contém a variável j .

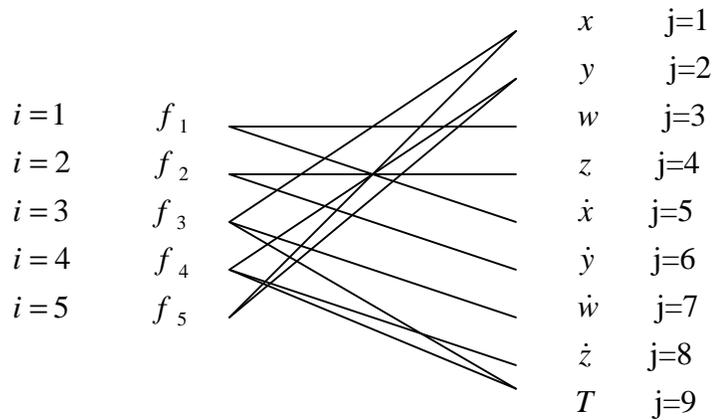


Figura 3.3 – Grafo bipartido do exemplo 1.

A lista de diferenciações A , para este problema é dada por:

$$A = (5, 6, 7, 8, 0, 0, 0, 0, 0).$$

Como o problema apresenta 9 variáveis, esta lista de diferenciações deverá ter 9 elementos. Para interpretá-la deve-se lembrar do vetor $X = (x, y, w, z, \dot{x}, \dot{y}, \dot{w}, \dot{z}, T)$, onde $A(1) = 5$ indica que a variável $X(5)$ é a derivada de $X(1)$, $A(2) = 6$ indica que a variável $X(6)$ é a derivada de $X(2)$, $A(8) = 0$ indica que não existe a derivada da variável $X(8)$, e assim sucessivamente.

De forma semelhante, a lista de diferenciações das equações, é definida como:

$$B(i) = \begin{cases} k & \text{se a } k\text{-ésima equação for diferenciada a partir da equação } i, \\ 0, & \text{caso contrário.} \end{cases}$$

Algoritmo A_1 (Pantelides, 1988)

(1) Inicialização:

$$\text{ASSIGN}(j) = 0, \quad j = 1 \text{ (1) } M$$

$$B(i) = 0, \quad i = 1 \text{ (1) } N$$

(2) Fazer $N' = N$

(3) Para $k = 1$ (1) N'

(3a) PATHFOUND = Falso

(3b) $i = k$

(3c) Enquanto PATHFOUND = Falso

(3c.1) Eliminar todos os V-nós com $A(.) \neq 0$ e todas as suas arestas no grafo

(3c.2) Designar todos os nós como sem cor

(3c.3) AUGMENTPATH(i,PATHFOUND)

(3c.4) Se PATHFOUND = Falso então

(i) Para cada um dos j V-nós coloridos, fazer

(i.1) $M = M + 1$

(i.2) Criar um novo V-nó M

(i.3) $A(j) = M$

(ii) Para cada um dos p E-nós coloridos, fazer

(ii.1) $N = N + 1$

(ii.2) Criar um novo E-nó N

(ii.3) Criar arestas do E-nó N para todos os V-nós j e $A(j)$ se a aresta $(p-j)$ existir

(ii.4) $B(p) = N$

(iii) Para cada um dos j V-nós coloridos, fazer ASSIGN($A(j)$) = B(ASSIGN(j))

(iv) $i = B(i)$

(4) Fim

A rotina AUGMENTPATH(i,PATHFOUND) em (3c.3) retorna PATHFOUND= Verdadeiro se existir um caminho estendido que começa em i , e se o caminho não existir, retorna PATHFOUND = Falso. Os V-nós e E-nós são chamados sem cor ou coloridos conforme tendo ou não sido verificados por esta rotina.

Analisando o algoritmo A1 para o exemplo do pêndulo tem-se:

(3c) Enquanto PATHFOUND = Falso

(3c.1) Eliminar todos os V-nós com $A(.) \neq 0$ e todas as suas arestas no grafo, isto é, (x, y, w, z) foram retiradas do grafo, conforme Figura 3.2, apresentada anteriormente..

A subrotina AUGMENTPATH(i,PATHFOUND) para $k = 1, 2, 3, 4$ retorna PATHFOUND = Verdadeiro. O vetor ASSIGN = $(0, 0, 0, 0, 1, 2, 3, 4, 0)$, isto é, a equação f_1 está relacionada com a variável \dot{x} , a equação f_2 está relacionada com a variável \dot{y} , e assim para as demais componentes, segundo a tabela abaixo:

k	PATHFOUND	CAMINHO ESTENDIDO
1	V	f_1, \dot{x}
2	V	f_2, \dot{y}
3	V	f_3, \dot{w}
4	V	f_4, \dot{z}

(3c.3) AUGMENTPATH(i ,PATHFOUND) retorna Falso para $k = 5$

f_5 é diferenciada, surgindo um novo E – nodo

(3c.4) Se PATHFOUND = Falso então

(ii.1) $N = N + 1 = 6$

(ii.2) Criar um novo E-nó f_6 , correspondente a derivação de f_5

(iii.3) Criar arestas de E-nó f_6 para os V-nós 5 e 6, Figura 3.4, existentes na equação.

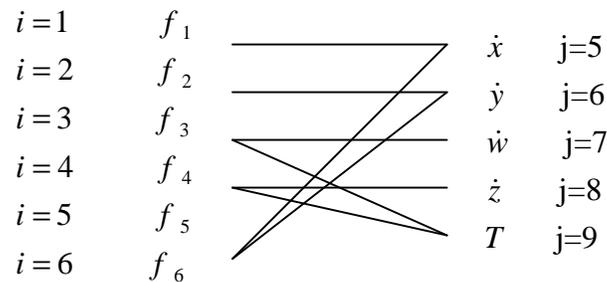


Figura 3.4 – Grafo resultante da diferenciação da f_5 .

(ii.4) $B(p) = N$

$B = (0, 0, 0, 0, 6, 0)$, a equação 6 resulta da diferenciação da equação 5.

(iv) $i = B(i) = 6$

A lista de associação de equações B indica que a equação f_6 surgiu a partir da diferenciação da equação f_5 . Neste momento, o sistema estendido apresenta $N = 6$ equações e $M = 9$ variáveis. Para que o sistema seja consistente ele deve ser estruturalmente não singular,

caso contrário, o sistema de EADs será estruturalmente inconsistente. Esta propriedade pode ser verificada formando o grafo correspondente e aplicando o AUGMENTPATH, a cada um dos E – nós. Caso nenhum caminho estendido seja encontrado para um E – nó, o sistema será consistente, e um subconjunto de $(M - N)$ variáveis deve ser selecionado para fornecer as condições iniciais.

Seguindo com o algoritmo tem-se:

(3c.2) Designar todos os nós como sem cor

(3c.3) AUGMENTPATH(6, PATHFOUND) retorna Falso.

Neste momento um subconjunto de 3 equações (f_1, f_2, f_6) são diferenciadas.

(3c.4) Se PATHFOUND = Falso então

(i) Para cada um dos j V-nós coloridos (nós 5 e 6), fazer

(i.1) $M = M + 1 = 11$

(i.2) criar um novo V-nó M (nós 10 e 11)

(i.3) $A(j) = M$ ($A(5) = 10$ e $A(6) = 11$)

(ii) Para cada um dos E-nós coloridos (nós 1, 2 e 6), fazer

(ii.1) $N = N + 1 = 9$

(ii.2) Criar um novo E nó (nós 7, 8 e 9)

(ii.3) Criar arestas do E-nó (nós 7,8 e 9) para os V-nós j e $A(j)$, se a aresta existir, conforme Figura 3.5.

(ii.4) $B(j) = N$ ($B(1) = 7$, $B(2) = 8$ e $B(6) = 9$)

(iii) Para cada um dos j V-nós coloridos fazer $ASSIGN(A(j)) = B(ASSIGN(j))$

(iv) $i = B(i) = 9$ ($i = 6$)

Retornando:

$$A = (5, 6, 7, 8, 10, 11, 0, 0, 0, 0, 0),$$

$$B = (7, 8, 0, 0, 6, 9, 0, 0, 0),$$

$$ASSIGN = (0, 0, 0, 0, 1, 2, 3, 4, 0, 7, 8).$$

O grafo representando esta etapa está exposto na Figura 3.5, onde já foram eliminados todos os nós $A(.) \neq 0$.

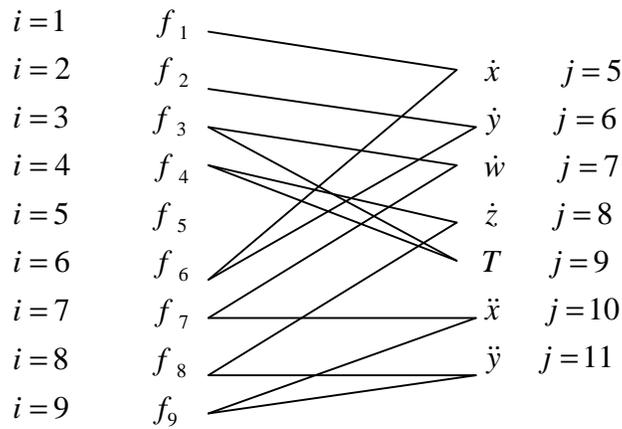


Figura 3.5 – Grafo resultante do sistema estendido

Aplicando o processo AUGMENTPATH para E – nó f_9 , encontra-se o caminho estendido:

$$(f_9 -- \ddot{x} -- f_7 -- \dot{w} -- f_3 -- T)$$

não sendo necessário novas diferenciações.

A nova associação resultante da última aplicação do algoritmo é $ASSIGN = (0, 0, 0, 0, 1, 2, 7, 4, 3, 9, 8)$. O sistema estendido correspondente é constituído por 11 equações ($M = 11$) e 9 variáveis ($N = 9$). Sendo assim, o sistema final possui grau de liberdade 2, ou seja, apenas duas das variáveis do conjunto $X = (x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, w, \dot{w}, z, \dot{z}, T)$ podem ser escolhidas arbitrariamente no processo de inicialização. As condições iniciais consistentes podem ser resolvidas para as demais variáveis, desde de que o sistema seja não singular.

De acordo com Gear et al. (1984), o índice diferencial de um sistema de EADs é o número máximo de vezes que ele precisa ser diferenciado em relação a t até ser transformado em um sistema de EDOs explícito. Analisando o sistema estendido apresentado ao final do algoritmo, e comparando com o conceito de índice, pode-se dizer que o algoritmo proposto por Pantelides, além de fornecer o sistema estendido, também fornece o índice diferencial do sistema original (limite superior).

A seguir é diferenciado manualmente o problema do pêndulo, a fim de identificar seu índice diferencial.

Relembrando as equações iniciais do sistema:

$$\dot{x} = w , \quad (3.27)$$

$$\dot{y} = z , \quad (3.28)$$

$$\dot{w} = T.x , \quad (3.29)$$

$$\dot{z} = T.y - g , \quad (3.30)$$

$$0 = x^2 + y^2 - L^2 . \quad (3.31)$$

Derivando-se a equação (3.31) e substituindo nela a equação (3.27) e (3.28), encontra-se:

$$xw + yz = 0 . \quad (3.32)$$

Derivando-se a equação (3.32) e substituindo nela as equações (3.27) a (3.31), encontra-se:

$$w^2 + z^2 + T.L^2 - y.g = 0 . \quad (3.33)$$

Derivando-se a equação (3.33) e substituindo nela a equação (3.28), (3.29), (3.30) e (3.32), encontra-se:

$$\dot{T} = \frac{3zg}{L^2} . \quad (3.34)$$

Observando-se que foram necessárias três derivações para transformar o sistema de EADs em EDOs e, partindo da definição de índice diferencial, conclui-se que o sistema possui índice 3.

Se analisarmos o vetor de saída do algoritmo $A_1 : B = (7,8,0,0,6,9,0,0,0)$, observa-se que a equação f_6 foi diferenciada a partir da equação f_5 , e a equação f_9 foi diferenciada a partir da equação f_6 . Como o algoritmo considera somente o menor subconjunto de equações necessárias para transformar o sistema original em um sistema de EADs de índice 1, deve-se somar 1 (um) ao número de diferenciações. Portanto, pode-se dizer que o índice

diferencial é no máximo $3 = 2 + 1$. Este índice vai ao encontro do índice encontrado a partir das diferenciações do sistema original. Assim sendo, o algoritmo A_1 também pode ser usado para dar idéia aproximada do índice diferencial do sistema.

Outros exemplos são demonstrados a seguir.

Exemplo 2

Considere o sistema de EADs:

$$\dot{y} + x - 1 = 0, \quad (3.35)$$

$$\dot{z} + y = 0, \quad (3.36)$$

$$z + \frac{y^2}{2} = 0. \quad (3.37)$$

O sistema apresenta $N = nd + na = 3$, sendo $nd = 2$ equações diferenciais e $na = 1$ equação algébrica e $M = 2nd + na = 5$ variáveis. A Figura 3.6 apresenta o grafo bipartido inicial.

O vetor X que contém todas as variáveis desconhecidas é dado por:

$$X = (y, z, \dot{y}, \dot{z}, x).$$

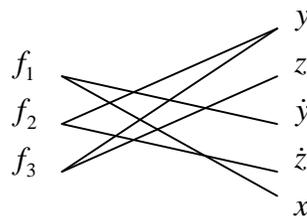


Figura 3.6 - Grafo bipartido $\{f\} - \{x\}$ do exemplo 2.

Inicializando o algoritmo A_1 , encontra-se o vetor A :

$$A = (3, 4, 0, 0, 0)$$

Todos os V - nós com $A(.) \neq 0$ são retirados do grafo, conforme a Figura 3.7:

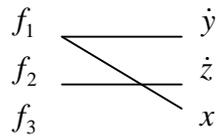


Figura 3.7 - Grafo bipartido intermediário

A subrotina AUGMENTPATH(i,PATHFOUND) para $k=1,2$ retorna PATHFOUND= Verdadeiro. Desta forma, o vetor ASSIGN torna-se:

$$ASSIGN = (0, 0, 1, 2, 0)$$

Para $k=3$ a subrotina retorna PATHFOUND = Falso, isto é, não há variável para ser associada a equação 3. Neste momento, o E-nó f_3 e os V-nós referentes as variáveis 1 e 2 são eliminados, apresentando a nova equação f_4 , Figura 3.8:

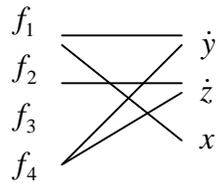


Figura 3.8 - Grafo bipartido solução final

Novamente a subrotina AUGMENTPATH(i,PATHFOUND) é chamada e retorna PATHFOUND = Verdadeiro Neste momento não há necessidade de outra diferenciação, finalizando o algoritmo e retornando os vetores:

$$A = (3, 4, 0, 0, 0),$$

$$B = (0, 0, 4, 0),$$

$$ASSIGN = (0, 0, 4, 2, 1).$$

O sistema final tem dimensão $M = 4$ equações e $N = 5$ variáveis, implicando que o sistema final possui 1 grau de liberdade.

Analisando o vetor B observa-se que a equação f_4 foi originada a partir da diferenciação da equação f_3 , assim pode-se dizer que o índice diferencial é no máximo 2.

Para ter certeza do valor do índice, transforma-se o sistema de EADs do exemplo 2 em um sistema de EDOs:

Derivando-se a equação (3.37) e substituindo-se nela a equação (3.36), chega-se:

$$\dot{y} = 1. \quad (3.38)$$

Substituindo a equação (3.38) na e equação (3.35) e derivando esta nova equação, obtém-se:

$$\dot{x} = 0. \quad (3.39)$$

Portanto, pode-se afirmar que o índice diferencial é 2.

Exemplo 3

Considere o sistema de EADs:

$$-2\dot{y}_2 + y_1 = 0, \quad (3.40)$$

$$4t^2 \dot{y}_2 + \dot{y}_3 + y_2 = 0, \quad (3.41)$$

$$y_3 = 0. \quad (3.42)$$

O sistema apresenta $N = nd + na = 3$, sendo $nd = 2$ equações diferenciais e $na = 1$ equação algébrica e $M = 2nd + na = 5$ variáveis. A Figura 3.9 apresenta o grafo bipartido inicial.

O vetor das variáveis desconhecidas X e a lista de associação das variáveis são dados por:

$$X = (y_2, y_3, \dot{y}_2, \dot{y}_3, y_1),$$

$$A = (3, 4, 0, 0, 0).$$

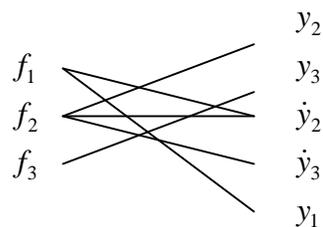


Figura 3. 9 - Grafo bipartido $\{f\} - \{y\}$ do exemplo 3.

Todos os V – nós com $A(.) \neq 0$ são retirados do grafo, resultando a Figura 3.10:

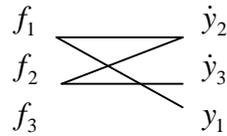


Figura 3. 10 - Grafo bipartido intermediário

Nas duas primeiras chamadas a subrotina AUGMENTPATH(i , PATHFOUND) retorna PATHFOUND= Verdadeiro. Desta forma, o vetor ASSIGN torna-se:

$$ASSIGN = (0, 0, 1, 2, 0).$$

Para a terceira chamada PATHFOUND retorna Falso, isto é, não há variável para ser associada a equação 3. Neste momento o E –nó f_3 é colorido e o V – nó referente a variável número 2 é removido, apresentando a nova equação f_4 , Figura 3.11:

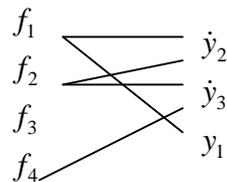


Figura 3.11 - Grafo bipartido da solução final

Novamente a subrotina AUGMENTPATH(i ,PATHFOUND) é chamada e retorna PATHFOUND= Verdadeiro. Logo, não há necessidade de outra diferenciação, finalizando o algoritmo e retornando os vetores:

$$A = (3, 4, 0, 0, 0),$$

$$B = (0, 0, 4, 0),$$

$$ASSIGN = (0, 0, 2, 4, 1).$$

O sistema final tem dimensão $M = 4$ equações e $N = 5$ variáveis, implicando que o sistema final possui 1 grau de liberdade.

Analisando o vetor B , observa-se que a equação f_4 foi originada a partir da diferenciação da equação f_3 . Assim, pode-se dizer que o índice diferencial é no máximo 2.

Para ter certeza do valor do índice, transforma-se o sistema de EADs do exemplo 3, em um sistema de ED0s:

Derivando-se a equação (3.42) e substituindo esta derivação na equação (3.41), chega-se a:

$$\dot{y}_2 = -\frac{y_2}{4t^2}. \quad (3.43)$$

Substituindo a equação (3.43) na equação (3.40), e derivando a equação resultante chega-se a:

$$\dot{y}_1 = \frac{\dot{y}_2}{2t^2} + \frac{y_2}{t^3}. \quad (3.44)$$

Portanto, pode-se afirmar que o índice diferencial é 2.

3.2 OUTRAS TÉCNICAS DE REDUÇÃO DE ÍNDICE

Existem na literatura diversos trabalhos que descrevem técnicas para reduzir o índice diferencial de EADs. Todos eles sugerem que, de alguma forma, as restrições algébricas sejam introduzidas novamente ao sistema, Vieira (1998).

Os algoritmos propostos por Gear e Petzold (1984), Ar1 e Ar2 a seguir, propõem a diferenciação das restrições algébricas com relação ao tempo, e utilizam o fato de que os termos com derivadas de ordem superiores introduzidas pela diferenciação aparecem linearmente nas novas equações. A técnica de redução do algoritmo Ar1 é válida para EADs lineares a coeficientes variáveis ou lineares na derivada e o algoritmo Ar2 é estendido para EADs lineares em geral.

O algoritmo Ar1 sugere que a diferenciação das restrições algébricas reduza o índice do sistema sem gerar alterações em sua solução. O algoritmo Ar2, após cada iteração, reduz seu sistema gerando matrizes lineares dependentes do tempo, que reduzem o sistema a forma semi-canônica.

Descrição do algoritmo Ar1:

1. Se o sistema $A(t)\dot{x}(t) + B(t)x(t) = f(t)$, equação (2.13), for regular, terminar.

2. Caso contrário, multiplicar a esquerda por uma matriz $\mathbf{P}(t)$ não singular, objetivando tornar nulas o maior número de linhas da matriz A . Deslocar estas linhas para a base para obter:

$$\begin{bmatrix} A_{11} \\ 0 \end{bmatrix} \dot{x} + \begin{bmatrix} B_{11} \\ B_{12} \end{bmatrix} x = P(t) f(t).$$

3. Diferenciar a parte inferior do sistema obtido em (2) para obter o novo sistema:

$$\begin{bmatrix} A_{11} \\ B_{12} \end{bmatrix} \dot{x} + \begin{bmatrix} B_{11} \\ \dot{B}_{12} \end{bmatrix} x = \begin{bmatrix} \dot{f}_1(t) \\ \dot{f}_2(t) \end{bmatrix}.$$

4. Voltar a 1.

Descrição do algoritmo Ar2:

1. Se o sistema $A(t)\dot{x}(t) + B(t)x(t) = f(t)$, equação (2.13), for regular, terminar.
2. Caso contrário, multiplicar a esquerda por uma matriz $\mathbf{P}(t)$ não singular, objetivando tornar nulas o maior número de linhas da matriz A . Deslocar estas linhas para a base para obter:

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & 0 \end{bmatrix} \dot{x} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} x = P(t) f(t).$$

3. Trocar as colunas A e B para que a matriz B_{22} seja regular.
4. Resolver para $\dot{x}_2 = B_{22}^{-1}(f_2 - B_{21}x_1)$. Diferenciar. Resolver para \dot{x}_2 em termos de x_1 e \dot{x}_1 .
5. Substituir as expressões para \dot{x}_2 e x_2 no topo do sistema. Rescrever o sistema na forma (2.13) para x_1 .
6. Voltar a 1.

Bachamann et al. (1990) propõem um algoritmo, Ar3, para EADs lineares com coeficientes variáveis, onde as equações algébricas que estavam "escondidas" tornam-se explícitas e substitui as equações diferenciais por restrições algébricas. A proposta do algoritmo Ar3 é semelhante ao algoritmo desenvolvido por Gear (1989), porém, o algoritmo de Gear implementa variáveis desnecessárias, ocasionando um aumento no número de equações. O algoritmo Ar3 indica quais equações algébricas devem ser diferenciadas e que os termos de derivadas superiores introduzidos com a diferenciação aparecem linearmente nas novas equações.

Este algoritmo é aplicado somente para sistemas da forma:

$$\dot{x} + B_{11}(t).x + B_{12}(t).y = g_1(t), \quad (3.45)$$

$$B_{21}(t).x + B_{22}(t).y = g_2(t), \quad (3.46)$$

onde $B_{11} \in M_{nd \times nd}(\square)$, $B_{12} \in M_{nd \times na}(\square)$, $B_{21} \in M_{na \times nd}(\square)$, $B_{22} \in M_{na \times na}(\square)$.

Descrição do algoritmo Ar3:

1. Se B_{22} for regular, parar.
2. Multiplicar B_{22} por uma matriz regular $P_{na \times na}$, tal que:

$$P.B_{22} = (D \quad 0)^T,$$

onde a matriz $D_{(na_1 \times na_1)}$ tem posto linha máximo na_1 . Para um problema com resolubilidade a matriz $C_1_{(na_2 \times na)}$ onde $na_2 = na - na_1$ tal que:

$$P.B_{21} = (C_1 \quad C_2)^T,$$

tem posto máximo na_2 , tal que os componentes x_1, \dots, x_{na_2} do vetor x são determinados em função dos demais componentes de x . A seleção dos na_2 componentes de x_1, \dots, x_{na_2} não é única, e pode ser conduzida apenas localmente, embora seja possível mantê-la por vários passos de integração.

3. No sistema resultante:

$$\dot{x} + B_{11}.x + B_{12}.y = g_1(t), \quad (3.47)$$

$$C_1.x + D.y = g_{21}(t), \quad (3.48)$$

$$C_2.x = g_{22}(t), \quad (3.49)$$

as equações (3.47) que contém x_1, \dots, x_{na_2} são substituídas por equações algébricas (3.48) usando as equações (3.49) diferenciadas. As derivadas de x são então eliminadas usando as equações (3.49).

4. Com o novo sistema, rescrito na forma das equações (3.45) e (3.46) voltar a etapa 1.

O exemplo do pêndulo descrito por Pantelides (1988), é aplicado para descrever o algoritmo Ar3.

Seja o sistema composto pelas equações (3.22), (3.23), (3.24), (3.25) e (3.26). O algoritmo Ar3 propõe reduzir o sistema de índice 3 para índice 1. Diferenciando a equação (3.26), obtém-se:

$$0 = T \cdot (x^2 + y^2) + w^2 + z^2 - y \cdot g . \quad (3.50)$$

Após aplicar o algoritmo para este exemplo, tem-se o possível conjunto solução para $y \neq 0$:

$$\dot{x} = w , \quad (3.51)$$

$$\dot{w} = T \cdot x , \quad (3.52)$$

$$0 = x^2 + y^2 - L^2 , \quad (3.53)$$

$$0 = x \cdot w + y \cdot z , \quad (3.54)$$

$$0 = T \cdot (x^2 + y^2) + w^2 + z^2 - y \cdot g . \quad (3.55)$$

Se y tende a zero, o sistema torna-se singular porque y e \dot{y} não são mais determinadas pelas equações (3.53) e (3.54). Neste caso o erro das equações diferenciais do sistema inicial é corrigido, executando novamente o passo 3, fornecendo uma outra solução:

$$\dot{y} = z , \quad (3.56)$$

$$\dot{z} = T \cdot y - g , \quad (3.57)$$

$$0 = x^2 + y^2 - L^2 , \quad (3.58)$$

$$0 = x \cdot w + y \cdot z , \quad (3.59)$$

$$0 = T.(x^2 + y^2) + w^2 + z^2 - y.g . \quad (3.60)$$

O algoritmo proposto por Matsson et al. (1993), Ar4, conserva todas as equações originais e suas derivadas sucessivas. Desta forma, gera-se um aumento no tamanho do sistema original e reduz-se o índice diferencial para 1, formando um sistema superdeterminado. Ao executar o algoritmo é gerada uma variável dependente (pseudo derivada), após cada equação resultante da diferenciação, que é uma variável algébrica e não sujeita a discretização. Desta forma, o sistema pode ser resolvido numericamente. Esta técnica é aplicada em larga escala e pode ser facilmente implementada usando diferenciação simbólica ou diferenciação automática.

Descrição do algoritmo Ar4:

1. Se o problema for estruturalmente singular, parar e retornar mensagem de erro.
2. Aplicar o algoritmo de Pantelides (teoria de grafos), estabelecendo o menor número de vezes que cada equação deve ser diferenciada e proceder a diferenciação.
3. Particionar o problema, de modo a obter um bloco triangular inferior. Separando o sistema em problema não diferenciado e problema diferenciado.
4. Reduzir o índice, selecionando as derivadas que serão substituídas pelas pseudo derivadas. Caso não seja possível selecionar uma pseudo derivada para cada equação diferenciada manipular as equações originais usando as informações dadas pelas equações diferenciais. Voltar ao passo 1.

Aplicando o algoritmo Ar4 ao problema do pêndulo de massa m e comprimento L , dados pelas equações:

$$x^2 + y^2 - L^2 = 0, \quad (3.61)$$

$$m\ddot{x} + (I/L)x = 0, \quad (3.62)$$

$$m\ddot{y} + (I/L)y + mg = 0, \quad (3.63)$$

onde g é a constante gravitacional e $I = -T.L.m$ ou $T = -\frac{I}{L.m}$ é a tração do fio.

Aplicando o algoritmo de Pantelides, diferencia-se a equação (3.61) 2 vezes obtendo o seguinte conjunto de equações, resultante da etapa 2 do algoritmo.

$$2x\ddot{x} + 2\dot{x}^2 + 2y\ddot{y} + 2\dot{y}^2 = 0, \quad (3.64)$$

$$m\ddot{x} + (1/L)x = 0, \quad (3.65)$$

$$m\ddot{y} + (1/L)y + mg = 0. \quad (3.66)$$

A etapa 3 resulta em um único bloco composto pela matriz jacobiana com respeito a maior ordem das derivadas $(\ddot{x} \ \ddot{y} \ \mathbf{I})^T$:

$$J = \begin{pmatrix} 2x & 2y & 0 \\ m & 0 & x/L \\ 0 & m & y/L \end{pmatrix}.$$

Sendo o $\det(J) = -2m(x^2 + y^2)L = -2mL$, o sistema passa a ser de índice 1 se, e somente se, $m \neq 0$ e $L \neq 0$. Neste caso, o processo de redução elimina sucessivamente as linhas e colunas da matriz jacobiana acima e durante este processo as pseudo derivadas são apropriadamente escolhidas. A matriz resultante das derivadas parciais é apresentada a seguir:

$$H_1^{[1]} = \begin{pmatrix} \ddot{x} & \ddot{y} & \mathbf{I} \\ a \end{pmatrix} \begin{pmatrix} 2x & 2y & 0 \end{pmatrix}.$$

Existe duas possibilidades de obtenção de uma matriz quadrada não singular $M_1^{[1]}$ a partir de $H_1^{[1]}$. No primeiro caso $|x| > |y|$, onde $M_1^{[1]} = (2x)$ as derivadas \ddot{x} e \dot{x} devem ser substituídas por pseudo derivadas, obtédo-se:

$$x^2 + y^2 - L^2 = 0, \quad (3.67)$$

$$2x\dot{x} + 2y\dot{y} = 0, \quad (3.68)$$

$$2xx'' + 2x'^2 + 2y\ddot{y} + 2\dot{y}' = 0, \quad (3.69)$$

$$mx'' + (\mathbf{I}/L)x = 0, \quad (3.70)$$

$$m\ddot{y} + (\mathbf{I}/L)y + mg = 0. \quad (3.71)$$

A segunda possibilidade tem-se $|x| \leq |y|$, onde $M_1^{[1]} = (2y)$, e:

$$x^2 + y^2 - L^2 = 0, \quad (3.72)$$

$$2x\dot{x} + 2y\dot{y}' = 0, \quad (3.73)$$

$$2x\ddot{x} + 2\dot{x}^2 + 2y\ddot{y}' + 2\dot{y}' = 0, \quad (3.74)$$

$$m\ddot{x} + (\mathbf{I}/L)x = 0, \quad (3.75)$$

$$m\ddot{y}' + (\mathbf{I}/L)y + mg = 0. \quad (3.76)$$

O sistema final tem 5 equações e 5 incógnitas, $x, y, \mathbf{I}, \dot{y}$ e \ddot{y} , para o primeiro caso e $x, y, \mathbf{I}, \dot{x}$ e \ddot{x} para o segundo caso. Ambos os casos têm apenas uma variável diferencial que aparece diferenciada duas vezes.

Atualmente está em desenvolvimento no PEQ/COPPE/UFRJ e DEQUI/UFRGS um código computacional, implementado com o algoritmo de Pantelides (1988), que gera automaticamente o sistema estendido do sistema de EADs fornecido pelo usuário. O usuário deve fornecer o sistema de EADs de acordo com os padrões adotados pelo código DASSLC (Secchi, 1992). Este código aplica o algoritmo de Pantelides, constrói o sistema estendido através de diferenciação exata das equações e informa o número r de graus de liberdade do sistema ao usuário. Após o usuário fornecer um número r de variáveis no tempo inicial, o algoritmo executará um teste de consistência estrutural neste conjunto. Informações mais detalhadas encontra-se em Costa Jr. (2001).

4. SOLUÇÃO DO SISTEMA ESTENDIDO

Considerando o exposto neste trabalho, admite-se que o número de variáveis a serem arbitradas na inicialização de sistemas de EADs será sempre inferior a dimensão do sistema, devido a presença das restrições algébricas. Desta forma, o número de graus de liberdade de um sistema de EADs será sempre inferior ao número de variáveis diferenciais.

O simples fato da dimensão do sistema de EADs não ser equivalente ao seu grau de liberdade ocasiona problemas de inicialização nos códigos numéricos. Deste modo, alguns métodos numéricos utilizam procedimentos de diferenciações sucessivas, como é o caso do algoritmo proposto por Pantelides (1988).

O algoritmo A_1 , discutido no capítulo 3, proposto por Pantelides, determina o conjunto mínimo de equações que deve ser diferenciado para obter a inicialização consistente. Entretanto, se faz necessário transformar o sistema de EADs de índice superior para um sistema de EADs de índice menor ou igual a 1. Devido a esta transformação, todas as equações que estavam “escondidas” no sistema original estarão presentes no sistema estendido.

Para o conjunto mínimo de equações a ser diferenciado em todos os modelos matemáticos do presente trabalho, que compõem o sistema estendido, utilizou o código numérico DAGRAFO, por Costa Jr. et al. (2000), que é uma implementação do algoritmo A_1 .

Para a execução deste código, o usuário deve fornecer, em um arquivo de entrada, a matriz de dependência do sistema original. Essa matriz indicará quais as variáveis e suas derivadas aparecem em cada equação. Para um sistema de equações na forma da equação (2.1) o elemento (i, j) da matriz de dependência é dada por:

$$\begin{aligned} &0 \text{ se a } i\text{-ésima equação não envolve } x_j \text{ e } \dot{x}_j, \\ &1 \text{ se a } i\text{-ésima equação envolve } x_j \text{ mas não } \dot{x}_j, \\ &2 \text{ se a } i\text{-ésima equação envolve } \dot{x}_j \text{ mas não } x_j, \\ &3 \text{ se a } i\text{-ésima equação envolve } x_j \text{ e } \dot{x}_j. \end{aligned} \tag{4.1}$$

Após vários testes executados pelo código descrito acima, obteve-se resultados satisfatórios em quase todos os casos. Observou-se que DAGRAFO não detecta subconjuntos de equações que sejam singulares, mas estruturalmente não singular. Um exemplo típico desta falha é o seguinte sistema de EADs linearmente implícito:

$$x_1 \dot{x}_1 - y + 1 = 0, \quad (4.2)$$

$$x_2 \dot{x}_2 - x_1 + 2 = 0, \quad (4.3)$$

$$x_1 \dot{x}_1 - y + x_2 + 3 = 0. \quad (4.4)$$

O sistema acima apresenta problema de inicialização devido o fato da sua matriz jacobiana em relação a \dot{x} e y ser singular:

$$\det(J) = \begin{vmatrix} \dot{x}_1 & \dot{x}_2 & y \\ x_1 & 0 & -1 \\ 0 & x_2 & 0 \\ x_1 & 0 & -1 \end{vmatrix} = 0. \quad (4.5)$$

O código DAGRAFO para este exemplo informou que não houve necessidade de diferenciação alguma, induzindo que a matriz jacobiana seja estruturalmente não singular. Após a execução do algoritmo, retorna a seguinte relação:

$$\begin{aligned} X &= (x_1, x_2, \dot{x}_1, \dot{x}_2, y), \\ A &= (3, 4, 0, 0, 0), \\ B &= (0, 0, 0), \\ ASSIGN &= (0, 0, 1, 2, 3). \end{aligned}$$

onde A é a lista de associação de variáveis (que expressa a relação entre as variáveis e suas derivadas em relação ao tempo), B é a lista de associação das equações (que indica qual equação foi criada a partir da diferenciação da i -ésima equação) e $ASSIGN$ contém a equação à qual cada variável j está associada.

Analisando somente o arquivo de saída do algoritmo DAGRAFO, conclui-se erroneamente através do vetor B , que o sistema já possui índice inferior, não havendo necessidade alguma de diferenciação. Na realidade, o sistema descrito pelas equações (4.2) a (4.4) possui índice 3.

Após encontrar o conjunto de equações que compõem o sistema estendido, faz-se necessário obter as condições iniciais que devem satisfazer o sistema original, assim como o

sistema estendido para serem consistentes. Nesta etapa do trabalho, utilizou-se o código computacional DASSLC, Secchi (1992).

Para executar este código é necessário fornecer as equações que compõem o sistema reduzido a índice 1 (um), juntamente com valores iniciais para as variáveis do sistema. A execução de DASSLC gera os valores iniciais das derivadas e os valores das variáveis e das suas derivadas para todo o intervalo de integração.

O exemplo 2, citado no capítulo 3, é utilizado a seguir para ilustrar as equações que compõem o sistema estendido e as condições iniciais. O sistema original de EADs é formado pelas equações:

$$\dot{y} + x - 1 = 0, \quad (4.6)$$

$$\dot{z} + y = 0, \quad (4.7)$$

$$z + \frac{y^2}{2} = 0. \quad (4.8)$$

Após aplicação do algoritmo DAGRAFO, no sistema citado, encontra-se o seguinte sistema reduzido (índice 1):

$$\dot{y} + x - 1 = 0, \quad (4.9)$$

$$\dot{z} + y = 0, \quad (4.10)$$

$$\dot{z} + y\dot{y} = 0, \quad (4.11)$$

representado pelo grafo bipartido da Figura 3.8 e constituído a partir das informações dos vetores:

$$X = (y, z, \dot{y}, \dot{z}, x),$$

$$A = (3, 4, 0, 0, 0),$$

$$B = (0, 0, 4, 0),$$

$$ASSIGN = (0, 0, 4, 2, 1).$$

de onde conclui-se que a equação (4.8) deve ser diferenciada para gerar a equação (4.11), $B(3) = 4$, e as equações que devem fazer parte do sistema reduzido são as equações (4.11), (4.7) e (4.6) cujos índices são 4, 2 e 1, respectivamente, dadas pelo vetor *ASSIGN*.

A equação (4.8), “rejeitada” pelo processo de redução de índice, deve ser satisfeita pelas condições iniciais, mas não é necessária para a solução do sistema (apesar de também ter que ser satisfeita em todos o intervalo). Esta equação é denominada de restrição “escondida” ou “rejeitada”.

As equações (4.8), (4.9), (4.10), e (4.11), que compõem o sistema estendido, devem ser satisfeitas pelo conjunto de condições iniciais arbitrado, como por exemplo:

$$\begin{aligned} y(0) &= 1, \quad \dot{y}(0) = 1, \\ z(0) &= -\frac{1}{2}, \quad \dot{z}(0) = -1, \\ x(0) &= 0. \end{aligned} \tag{4.12}$$

Observa-se que as condições iniciais dadas pela equação (4.12) são consistentes.

4.1 EFEITO DE NÃO CONSIDERAR AS RESTRIÇÕES ESCONDIDAS

Sistemas de EADs da forma (2.1) sempre que puderem ser transformados para a forma explícita (2.2) poderão ser resolvidas por métodos aplicados a sistemas de EDOs. Entretanto, como já foi discutido no capítulo 3, nem sempre é possível realizar este procedimento e, quando possível, não há garantia que a solução numérica do novo sistema seja compatível com a solução exata do sistema original.

Técnicas de inicialização de sistemas de EADs estão sendo aprimoradas para que problemas, como os citados acima, possam apresentar resultados confiáveis. Segundo Vieira (1998) a resolução de sistema de EADs por métodos diretos mostra-se mais convenientes devido a vários motivos:

- (i) o sistema de equações efetivamente integrado é mais simples e as variáveis têm significado físico;
- (ii) são evitadas manipulações algébricas e informações fundamentais que poderiam ser perdidas na diferenciação das equações são preservadas;

- (iii) equações algébricas fundamentais, relacionadas a descrições de fenômenos básicos são facilmente incluídas e/ou modificadas sempre que necessário.

Entretanto, para alguns sistemas de EADs, a resolução direta nem sempre é possível, devido a singularidade na matriz jacobiana.

Desta forma, a maneira mais coerente de resolver sistemas de EADs que apresente singularidade na matriz jacobiana e/ou índice elevado, é reduzir o índice diferencial do sistema, isto é, transformar o sistema original de EADs para um sistema reduzido na forma de EDOs, ou EADs de índice 1.

Os passos para encontrar o sistema reduzido foram detalhados no capítulo 3 e novamente citados no capítulo 4. Portanto, as equações contidas no vetor de associação, ASSIGN, fazem parte do sistema reduzido. As demais equações que compõem o sistema e que não fazem parte desta associação, equações que não estão associadas as variáveis, representam das equações “rejeitadas”.

Embora as equações “rejeitadas” não estejam associadas a variável alguma, são de extrema importância para a inicialização e, devem ser satisfeitas não somente na inicialização, como também em todos os tempos de integração. Geralmente as equações não associadas são as equações algébricas do sistema original de EADs que foram diferenciadas na construção do sistema reduzido.

Caso, após o processo de redução de índice, o sistema resultante apresentar mais equações que variáveis novas, o novo sistema apresentará restrições entre as variáveis do sistema que não estavam implícitas nas equações originais. Algumas destas restrições serão equações “rejeitadas” na construção do sistema reduzido, mas que podem conter informações importantes, como balanços globais de massa.

Do exposto acima, conclui-se que é necessário trabalhar com o sistema reduzido para resolver sistemas de EADs de índice elevado. Porém, deve-se ficar alerta à perda de algumas restrições algébricas que estavam presentes no sistema original, restrições que podem ser importantes para a viabilidade do método utilizado. Ou seja, para a obtenção das condições iniciais consistentes deve-se resolver o sistema estendido e para a integração do sistema de EADs deve-se, no mínimo, resolver o sistema reduzido.

O exemplo abaixo ilustra as restrições impostas a partir da transformação do sistema original no sistema reduzido:

$$\dot{x}_1 - x_1 - x_2 + y = 0, \tag{4.13}$$

$$\dot{x}_2 - x_2 - 2y = 0, \quad (4.14)$$

$$x_1 + 3x_2 = 0. \quad (4.15)$$

O sistema composto pelas equações (4.13), (4.14) e (4.15) de índice diferencial 2, não pode ser integrado diretamente, fazendo-se necessário reduzir seu índice. Diferenciando-se a equação (4.15) uma vez, obtém-se:

$$\dot{x}_1 + 3\dot{x}_2 = 0. \quad (4.16)$$

O novo sistema composto pelas equações (4.13), (4.14) e (4.16) resulta no sistema reduzido de índice 1 e pode ser resolvido por técnicas de integração direta. Entretanto, a equação (4.15) foi perdida ao longo do sistema.

A perda da equação (4.15) é equivalente a perda de uma constante de integração que pode ter uma interpretação física importante para a solução do problema original.

Como estas restrições são de grande importância para a confiabilidade do método numérico aplicado, optou-se por desenvolver uma técnica que leve-as em conta na integração do sistema. O desenvolvimento desta técnica gera equações a mais no sistema reduzido e, por consequência, a dimensão do sistema deve ser aumentada, com um número de variáveis novas igual ao de equações não associadas. Estas variáveis novas são denominadas de “variáveis de restrição”. Deste modo, resolve-se o sistema estendido ao invés do sistema reduzido.

As equações que até então não estavam associadas e passaram a compor o novo sistema têm o seu valor subtraído de uma dessas variáveis de restrições, que são inicializadas com valor nulo. Desta forma, o código passa a controlar a violação das restrições escondidas e o usuário tem informação direta de quanto cada restrição escondida é violada.

4.2. PROPOSIÇÃO DE TÉCNICAS PARA LEVAR EM CONTA AS RESTRIÇÕES ESCONDIDAS

Ao aplicar um algoritmo que seja estável para resolver problemas de EADs encontrar-se-á uma solução computada próxima da exata. Levando-se em consideração uma propriedade característica do problema, chamada condicionamento, esta também contribuirá para a exatidão do resultado computado.

O condicionamento mostra-nos como a solução do problema muda, se os dados de entrada contêm alguma perturbação em relação aos verdadeiros. Esta preocupação surge porque na prática, muitos destes dados, provêm de observações experimentais, nas quais as medições estão sujeitas a perturbações.

Muitos códigos tendem a falhar devido a duas situações: a natureza do problema e/ou o algoritmo utilizado. O algoritmo DASSLC apresentou-se eficiente quando aplicado a sistemas genéricos, porém, a solução de sistemas de EADs, nem sempre é confiável devida a inicialização do mesmo. Por este motivo optou-se por levar em consideração as restrições escondidas no sistema reduzido, para se ter um melhor controle da solução.

No presente trabalho executou-se o código DASSLC, para todos os exemplos estudados, considerando três situações:

- (i) sistema reduzido;
- (ii) sistema estendido com variáveis de restrições;
- (iii) sistema estendido com as derivadas das variáveis de restrições.

(i) SISTEMA REDUZIDO

Seja o sistema de EADs composto pelas seguintes equações:

$$\dot{y} + x - 1 = 0, \quad (4.17)$$

$$\dot{z} + y = 0, \quad (4.18)$$

$$z + \frac{y^2}{2} = 0. \quad (4.19)$$

Após aplicar a algoritmo DAGRAFO, recebe-se a informação que a equação (4.19) deve ser diferenciada uma vez, assim passa-se a ter uma nova equação:

$$\dot{z} + y\dot{y} = 0. \quad (4.20)$$

O novo sistema composto pelas equações (4.17), (4.18) e (4.20) forma o sistema reduzido, representado pela forma implícita:

$$F(\dot{x}, x, y, t) = 0, \quad (4.21)$$

onde x e $\dot{x} \in \mathbb{R}^{nd}$ são as variáveis diferenciais e suas derivadas, t é a variável independente, $y \in \mathbb{R}^{na}$ são as variáveis algébricas e $F \in \mathbb{R}^{nd+na}$ é suficientemente diferenciável.

Após a diferenciação realizada acima observa-se a perda da equação (4.19). O fato desta não fazer parte da lista de associação apresentada em ASSIGN, faz com que ela seja “rejeitada” do conjunto de equações que compõem o sistema reduzido.

Entretanto, a necessidade de considerar esta equação para obter a solução numérica já foi argumentada no item anterior. A partir dessa diferenciação o sistema reduzido está pronto para ser resolvido pelo algoritmo de integração DASSLC.

(ii) SISTEMA ESTENDIDO COM VARIÁVEIS DE RESTRIÇÃO

Neste caso o sistema de EADs a ser resolvido é composto pelo sistema reduzido e pelas equações “rejeitadas” ou restrições escondidas com a introdução das variáveis de restrição. Estas restrições são controladas pelo integrador e o usuário passa a monitorar o quanto elas interferem na solução aproximada do sistema.

Ao conjunto de equações (4.17), (4.18) e (4.20), que compõem o sistema reduzido, é adicionado a equação abaixo, para formar o sistema estendido com a variável de restrição e :

$$z + \frac{y^2}{2} = e. \quad (4.22)$$

O novo sistema, composto pelas equações descritas acima, é representado pela forma implícita:

$$\begin{cases} F(\dot{x}, x, y, t) = 0 \\ G(\dot{x}, x, y, t) = e \end{cases}, \quad (4.23)$$

onde x e $\dot{x} \in \mathbb{R}^{nd}$ são as variáveis diferenciais e suas derivadas, t é a variável independente, $y \in \mathbb{R}^{na}$ são as variáveis algébricas, $e \in \mathbb{R}^r$ são as variáveis de restrição, $F \in \mathbb{R}^{nd+na}$, $G \in \mathbb{R}^r$ e $0 < r$ é o número de equações “rejeitadas”.

O novo sistema passa a contar com as variáveis de restrição, e , que, ao executar o código numérico DASSLC fornece além do erro da restrição, e , no instante t a derivada do erro, \dot{e} .

No momento em que as variáveis de restrição são introduzidas no sistema, passa-se a monitorar o quanto cada restrição escondida é violada. Desta forma, tem-se uma solução numérica mais confiável.

(iii) SISTEMA ESTENDIDO COM AS DERIVADAS DAS VARIÁVEIS DE RESTRIÇÕES

Neste caso, o sistema de EADs a ser resolvido é composto pelo sistema reduzido e pelas equações rejeitadas, com a introdução das derivadas das variáveis de restrição.

Para o exemplo dado, o novo sistema é composto pelas equações (4.17), (4.18) e (4.20) que compõem o sistema reduzido adicionado a equação abaixo com a derivada da variável de restrição \dot{e} (ou mais precisamente, a variável restrição na forma de derivada):

$$z + \frac{y^2}{2} = \dot{e}. \quad (4.24)$$

Assim, o novo sistema, composto pelas equações (4.17), (4.18), (4.20) e (4.24), é representado pela forma implícita:

$$\begin{cases} F(\dot{x}, x, y, t) = 0 \\ G(\dot{x}, x, y, t) = \dot{e} \end{cases} \quad (4.25)$$

onde x e $\dot{x} \in \mathbb{R}^{nd}$ são as variáveis diferenciais e suas derivadas, t é a variável independente, $y \in \mathbb{R}^{na}$ são as variáveis algébricas, $\dot{e} \in \mathbb{R}^r$ são as derivadas das variáveis das restrições, $F \in \mathbb{R}^{na+nd}$, $G \in \mathbb{R}^r$ e r é o número de equações “rejeitadas”.

A integração do novo sistema pelo código numérico DASSLC, fornece a integral do erro (e) e o erro da restrição (\dot{e}) no instante t .

5. EXEMPLOS RESOLVIDOS

Neste capítulo são exemplificados os resultados das aplicações das técnicas de inicialização consistente e integração numérica de sistemas de EADs no programa DASSLC. A análise dos dados é apresentada a partir de problemas descritos na literatura, onde são enfatizadas as técnicas de solução do sistema reduzido, descritas no capítulo 4, e discutidos seus pontos fortes e suas limitações.

A análise da eficácia de cada solução leva em consideração as observações dos dados obtidos a partir das técnicas empregadas e a estrutura da matriz jacobiana do sistema discretizado, também chamada de matriz de iteração do sistema.

Para todos os exemplos descritos neste capítulo aplicou-se as diferentes técnicas de inicialização e integração para a solução numérica:

$$(i) \quad F(\dot{x}, x, y, t) = 0, \quad (5.1)$$

$$(ii) \quad \begin{cases} F(\dot{x}, x, y, t) = 0 \\ G(\dot{x}, x, y, t) = \mathbf{e} \end{cases}, \quad (5.2)$$

$$(iii) \quad \begin{cases} F(\dot{x}, x, y, t) = 0 \\ G(\dot{x}, x, y, t) = \dot{e} \end{cases}. \quad (5.3)$$

e adotou-se tolerâncias relativa de 10^{-5} e absoluta de 10^{-7} .

5.1 EXEMPLO 1. PROPOSTO POR CAMPBELL (1983) – ÍNDICE 2

$$\dot{y} + x - 1 = 0, \quad (5.4)$$

$$\dot{z} + y = 0, \quad (5.5)$$

$$z + \frac{y^2}{2} = 0. \quad (5.6)$$

O conjunto de EADs descrito acima apresenta solução analítica do tipo:

$$\begin{aligned}
 y(t) &= (\sqrt{-2z_0} + t), \\
 \dot{y}(t) &= 1, \\
 z(t) &= \left(\sqrt{-2z_0} - \frac{t}{2} \right)^2, \\
 \dot{z}(t) &= -(\sqrt{-2z_0} + t), \\
 x(t) &= 0.
 \end{aligned} \tag{5.7}$$

A solução analítica do sistema é apresentada através de gráficos, Figuras 5.1 a 5.3, juntamente com a solução numérica. A análise gráfica dos dois tipos de resolução do sistema de EADs tornam-se importantes para comparar o quanto a solução numérica aproxima-se da solução exata. Esta análise é realizada em função das variáveis y , z e x com condições iniciais consistentes e condições iniciais inconsistentes, para todas as técnicas de inicialização e integração, a fim de observar o quanto o código numérico adotado torna-se confiável.

O conjunto de condições iniciais consistentes para o exemplo 1 deve levar em consideração a restrição existente na terceira equação do sistema, equação (5.6), devido a dependência entre as variáveis z e y . Assim, uma solução consistente para a inicialização deste sistema de EADs é dado por:

$$\begin{aligned}
 y(0) &= (\sqrt{-2z_0}), & \dot{y}(0) &= 1, \\
 z(0) &= z_0, & \dot{z}(0) &= -\sqrt{-2z_0}, \quad z_0 \leq 0, \\
 x(0) &= 0.
 \end{aligned} \tag{5.8}$$

Este conjunto de condições consistentes, já apresentado na equação (4.12), juntamente com o sistema reduzido, descrito pelo conjunto de equações (4.21), foi submetido ao código computacional DASSLC para as três diferentes formulações, dadas pelas equações (5.1), (5.2) e (5.3). Os resultados gerados por estas simulações, mostraram-se menos precisos quando aplicados ao sistema (5.2). A evidência desta menor precisão é observada através das Tabelas 5.1 e 5.2.

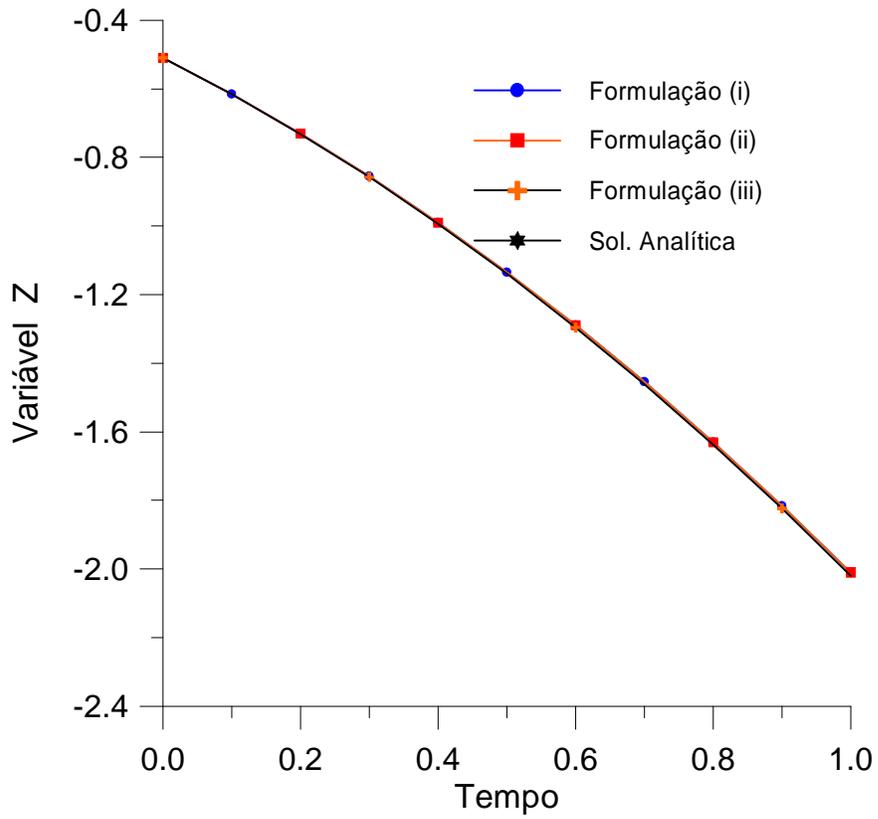


Figura 5.1- Variável $z \times t$ - exemplo 1

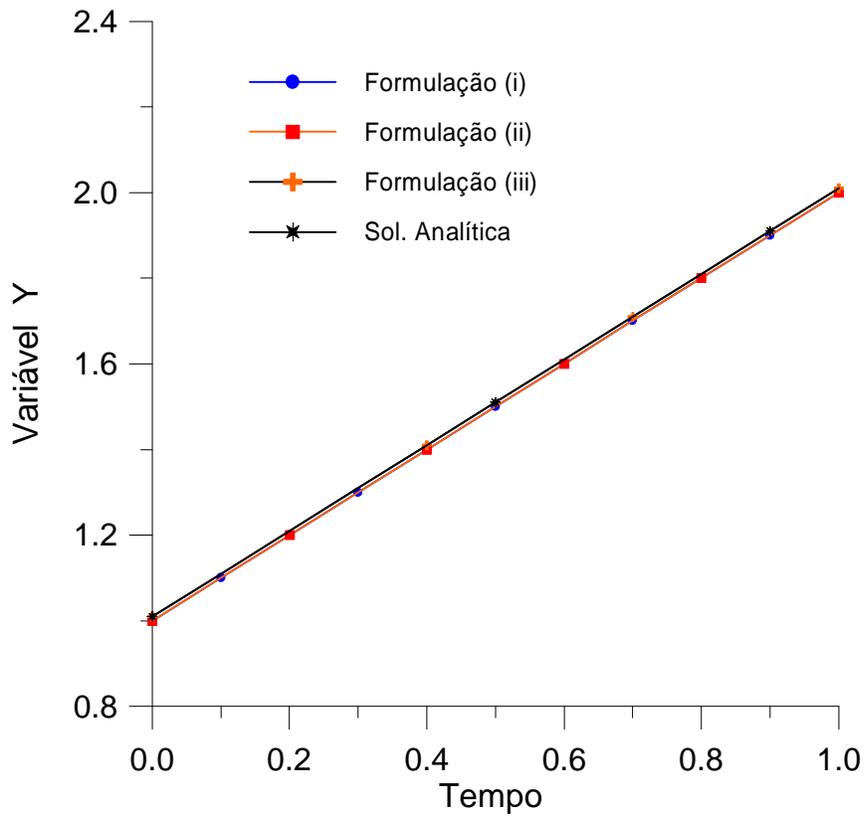


Figura 5.2- Variável $y \times t$ - exemplo 1

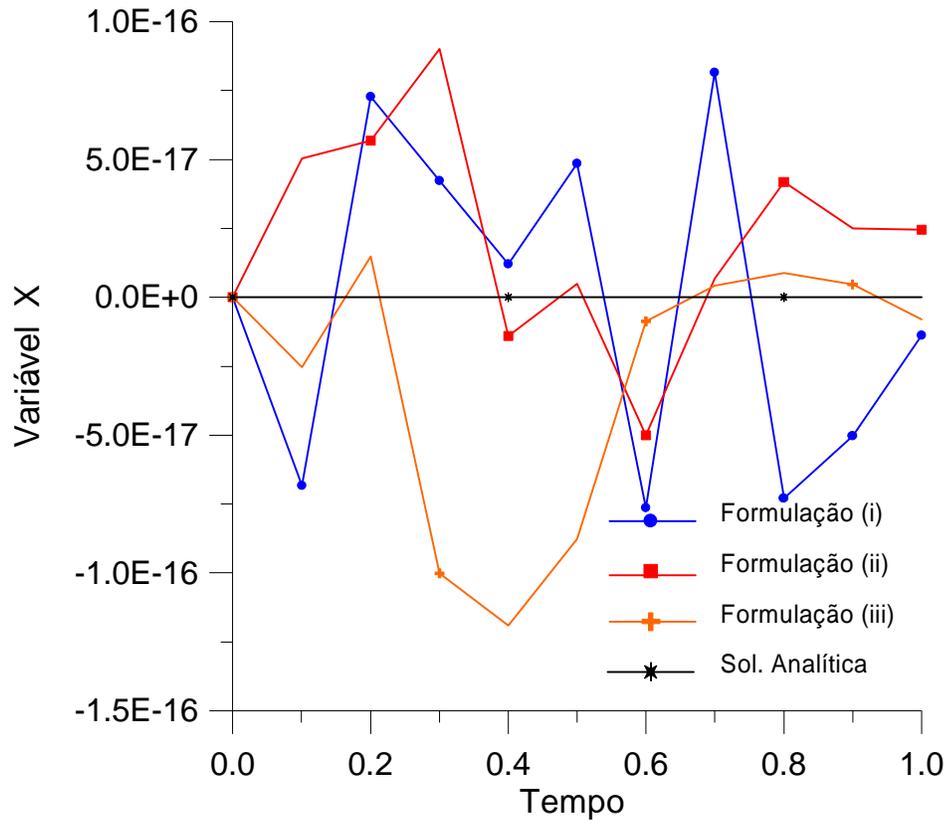


Figura 5.3- Variável $x \times t$ - exemplo 1

A Tabela 5.1, representada pela variável z , apresentou maior erro de integração para a simulação do sistema (5.2), devido a dependência entre as variáveis y e z , já mencionada anteriormente, e devido a rigidez do sistema.

A Tabela 5.2 apresenta informações estatísticas importantes da integração para análise do código numérico em questão. Observa-se que o sistema (5.2), com exceção ao número total de falhas do teste de convergência mostrou-se inferior, quando comparado aos sistemas (5.1) e (5.3)

Tabela 5.1- Representação numérica da variável z e sua derivada – exemplo 1

Variável z	Sol. exata	Sistema (i)		Sistema (ii)		Sistema (iii)	
	z	z	\dot{z}	z	\dot{z}	z	\dot{z}
t=0	-0,500	-0,500	-1,0	-0,50000	-1,0	-0,500	-1,0
t=0,1	-0,605	-0,605	-1,1	-0,605035	-1,101	-0,605	-1,1
t=0,2	-0,720	-0,720	-1,2	-0,72007	-1,20	-0,720	-1,2
t=0,3	-0,845	-0,845	-1,3	-0,845105	-1,301	-0,845	-1,3
t=0,4	-0,980	-0,980	-1,4	-0,98014	-1,400	-0,980	-1,4
t=0,5	-1,125	-1,125	-1,5	-1,12517	-1,501	-1,125	-1,5
t=0,6	-1,280	-1,280	-1,6	-1,28021	-1,601	-1,280	-1,6
t=0,7	-1,445	-1,445	-1,7	-1,44525	-1,701	-1,445	-1,7
t=0,8	-1,620	-1,620	-1,8	-1,62028	-1,800	-1,620	-1,8
t=0,9	-1,805	-1,805	-1,9	-1,80532	-1,901	-1,805	-1,9
t=1	-2,000	-2,000	-2,0	-2,00035	-2,001	-2,000	-2,0

Tabela 5.2- Tabela comparativa das técnicas de inicialização – exemplo 1

Total do número de	Sistema (i)	Sistema (ii)	Sistema (iii)
Pontos	24	1430	24
avaliação do resíduo	20	5699	20
avaliação da matriz jacobiana	18	2850	18
iterações de Newton	20	5699	20
falhas do teste do erro	0	1421	0
falhas do teste da convergência	0	0	0
pontos rejeitados	0	1421	0
iterações de Newton-Raphson rejeitada	0	1421	0
Tempo total (segundos)	0.03	0.36	0.03

A inferioridade da solução numérica para o sistema (5.2) é devido ao mau condicionamento do sistema de EADs, que pode ser analisado através da matriz de iteração do sistema.

No código DASSLC a integração numérica é realizada por método tipo predição-correção, onde a etapa de correção é realizada através do método de Newton-Raphson, ou seja:

$$x_{n+1}^{(k+1)} = x_{n+1}^{(k)} - \left[J_{n+1}^{(k)} \right]^{-1} F \left(x_{n+1}^{(k)} \right), \quad (5.9)$$

onde $J_{n+1}^{(k)}$ representa a matriz de iteração, que pode ser escrita pela soma das seguintes parcelas:

$$J_{n+1}^{(k)} = \mathbf{g}A + B, \quad (5.10)$$

onde $A = \left(\left. \frac{\partial F}{\partial \dot{y}} \right|_{t_{n+1}} \right)^{(k)}$, $B = \left(\left. \frac{\partial F}{\partial y} \right|_{t_{n+1}} \right)^{(k)}$ e $\mathbf{g} \propto \frac{1}{h}$, h representa o tamanho do passo de integração.

A matriz de iteração do sistema (5.2) é dada por:

$$J(\mathbf{a}, y, \dot{y}) = \underbrace{\begin{pmatrix} \dot{y} & \dot{z} & \dot{x} & \varepsilon \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ y & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_A + \mathbf{a} \underbrace{\begin{pmatrix} y & z & x & \varepsilon \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ \dot{y} & 0 & 0 & 0 \\ y & 1 & 0 & -1 \cdot \mathbf{b} \end{pmatrix}}_B = \begin{pmatrix} 1 & 0 & \mathbf{a} & 0 \\ \mathbf{a} & 1 & 0 & 0 \\ y + \dot{y} \cdot \mathbf{a} & 1 & 0 & 0 \\ \mathbf{a} \cdot y & \mathbf{a} & 0 & -\mathbf{a} \cdot \mathbf{b} \end{pmatrix}, \quad (5.11)$$

onde $\alpha = \gamma^1$ e \mathbf{b} representa um fator de escala para a variável ε .

Adotando os seguintes valores para as variáveis y , \dot{y} , \mathbf{a} e \mathbf{b} :

$$\begin{aligned} y &= 1, & \dot{y} &= 1 \\ \mathbf{a} &= 10^{-5}, & \mathbf{b} &= 1 \end{aligned}, \quad (5.12)$$

a equação (5.13) fornece a solução dos autovalores $(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4)^T$ e os autovetores da matriz J .

$$\text{autovalores: } \begin{pmatrix} -1 \cdot 10^{-5} \\ 1 \\ -1 \cdot 10^{-5} \\ 1 \end{pmatrix} \text{ autovetores: } \begin{pmatrix} 0 & -0,5 & 7,071 \cdot 10^{-6} & 0,5 \\ 0 & 0,809 & -7,071 \cdot 10^{-11} & 0,309 \\ 0 & 0,309 & -0,7071 & 0,809 \\ 1 & 3,09 \cdot 10^{-6} & 0,7071 & 8,09 \cdot 10^{-6} \end{pmatrix}. \quad (5.13)$$

Analisando a relação entre os autovalores, observa-se o elevado mau condicionamento da matriz de iteração. Este mau condicionamento foi numericamente evidenciado no sistema (5.2) e está diretamente ligado ao autovalor \mathbf{a}_3 que mantém as variáveis x e ε fortemente acopladas, de

acordo com o respectivo autovetor. Observa-se que este efeito não está presente em \mathbf{a}_1 porque este autovalor está relacionado somente com a variável ε , como mostra o seu autovetor associado.

O processo descrito acima foi realizado com diferentes valores para o escalar \mathbf{b} e obteve-se melhoras significativas na integração do código numérico. Observou-se que quando $\mathbf{b} \rightarrow 1$ o sistema fornece uma solução numérica não confiável e, quando \mathbf{b} se afasta de 1, a solução numérica fornecida aproxima-se da solução exata. Este melhor desempenho é descrito pelas equações (5.14) e (5.15), onde:

$\mathbf{b} = 10$.

$$\text{autovalores: } \begin{pmatrix} -2 \cdot 10^{-4} \\ 1 \\ -1 \cdot 10^{-5} \\ 1 \end{pmatrix} \text{ autovetores: } \begin{pmatrix} 0 & -0,5 & -1 \cdot 10^{-5} & 0,5 \\ 0 & 0,809 & 1 \cdot 10^{-10} & 0,309 \\ 0 & 0,309 & 1 & 0,809 \\ 1 & 3,09 \cdot 10^{-6} & 1,111 \cdot 10^{-6} & 8,09 \cdot 10^{-6} \end{pmatrix}, \quad (5.14)$$

$\mathbf{b} = 100$

$$\text{autovalores: } \begin{pmatrix} -1 \cdot 10^{-3} \\ 1 \\ -1 \cdot 10^{-5} \\ 1 \end{pmatrix} \text{ autovetores: } \begin{pmatrix} 0 & -0,5 & -1 \cdot 10^{-5} & 0,5 \\ 0 & 0,809 & 1 \cdot 10^{-10} & 0,309 \\ 0 & 0,309 & 1 & 0,809 \\ 1 & 3,09 \cdot 10^{-6} & 1,01 \cdot 10^{-7} & 8,08 \cdot 10^{-6} \end{pmatrix}. \quad (5.15)$$

O desempenho do código para os diferentes valores do escalar \mathbf{b} pode ser observado através das Tabelas 5.3 e 5.4. Valores elevados de \mathbf{b} aumentam o desacoplamento entre as variáveis x e ε ao integrar o sistema.

Tabela 5.3 - Tabela comparativa da variável z para vários valores de β - exemplo 1

Variável z	Sistema (ii)			
	$b = 1$	$b = 10$	$b = 100$	sol. exata
t=0	-0,50000	-0,50000	-0,50000	-0,50000
t=0,1	-0,605035	-0,605013	-0,60500	-0,60500
t=0,2	-0,72007	-0,720013	-0,72000	-0,72000
t=0,3	-0,845105	-0,845013	-0,84500	-0,84500
t=0,4	-0,98014	-0,980013	-0,98000	-0,98000
t=0,5	-1,12517	-1,12501	-1,12500	-1,12500
t=0,6	-1,28021	-1,28001	-1,28000	-1,28000
t=0,7	-1,44525	-1,44501	-1,44500	-1,44500
t=0,8	-1,62028	-1,62001	-1,62000	-1,62000
t=0,9	-1,80532	-2,80501	-1,80500	-1,80500
t=1,0	-2,00035	-2,00001	-2,00000	-2,00000

Tabela 5.4- Desempenho do algoritmo de integração para vários valores de β - exemplo1

Total do número de:	Sistema (ii)		
	$b = 1$	$b = 10$	$b = 100$
Pontos	1430	32	24
avaliação do resíduo	5699	47	20
avaliação da matriz jacobiana	2850	18	18
iterações de Newton	5699	47	20
falhas do teste do erro	1421	0	0
falhas do teste da convergência	0	0	0
pontos rejeitados	1421	0	0
iterações de Newton-Raphson rejeitada	1421	0	0
Tempo total (segundos)	0,36	0,03	0,03

Sabendo-se que as técnicas de inicialização e integração (5.2) e (5.3) incorporam ao sistema reduzido as equações “rejeitadas”, torna-se possível fornecer o erro de integração e a integral do erro, a partir da variável de restrição. As relações entre as variáveis de restrição das duas formulações são dadas por:

$$\begin{cases} e = \int_{t_0}^t \mathbf{e} dt & (\text{integral do erro}) \\ \dot{e} = \mathbf{e} & (\text{erro}) \end{cases} \quad (5.16)$$

Desta forma, é possível monitorar o quanto cada restrição é violada a cada passo de integração do sistema. E, em um próximo estágio de desenvolvimento do algoritmo de integração numérica, pode-se incorporar estas variáveis de restrição no controle do passo de integração (controle do erro local).

A seguir são representados o erro de integração e a integral do erro através das Figuras 5.4 e 5.5, respectivamente. Observa-se um crescimento significativo (em valor absoluto) do erro de restrição para a formulação (5.2).

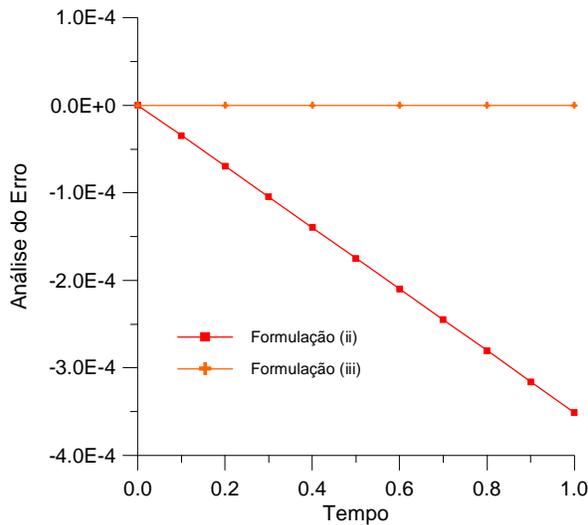


Figura 5.4- Análise do erro (\mathbf{e}) - exemplo1

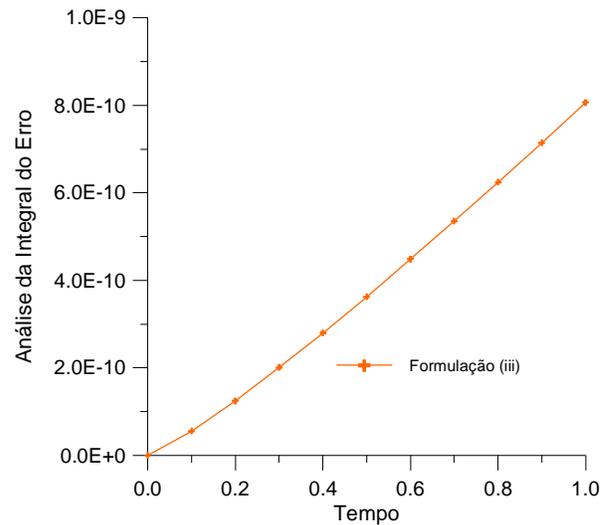


Figura 5.5- Análise da integral erro (e) - exemplo1

Para quantificar o efeito de não levar em conta as restrições rejeitadas, o exemplo 5.1 é resolvido novamente, por todos os métodos de inicialização descritos anteriormente, porém arbitrando um conjunto de condições iniciais inconsistentes, da forma:

$$\begin{aligned} y(0) &= 1, \quad \dot{y}(0) = 1, \\ z(0) &= -0,51; \quad \dot{z}(0) = -1, \\ x &= 0. \end{aligned} \quad (5.17)$$

A condição inicial aplicada a variável z torna o conjunto das condições iniciais inconsistente.

Ao aplicar estas condições iniciais no sistema reduzido (5.1), o código numérico não fez restrição alguma quanto a validade das condições iniciais e integrou o sistema para todo o intervalo desejado, sem detectar o erro. Este procedimento é possível porque o sistema em estudo rejeita as equações que não estejam contidas no vetor de associação ASSIGN. Para o exemplo em estudo a equação “rejeitada” foi justamente a equação (5.6) que contém a variável z .

Apesar de gerar um conjunto solução, o sistema (5.1) não é recomendado porque não satisfaz o sistema original, podendo violar a solução de alguma variável que tenha significado físico importante para a solução do sistema.

Aplicando o mesmo conjunto de condições iniciais para o sistema (5.2) o código detectou o erro de inicialização e não gerou solução alguma. Porém, quando aplicado ao sistema (5.3), o código DASSLC foi executado sem problemas, onde a propagação do erro de inicialização pode ser observado na variável $e(t)$.

Este procedimento tornou-se possível porque o sistema (5.3), além de incorporar as equações que haviam sido “rejeitadas” pelo sistema estendido, trabalha com a derivada da variável de restrição. Desta forma, o erro da inicialização é detectado e transferido para a variável de restrição que, por sua vez, pode ser monitorada. Assim, é possível saber o quanto o erro gerado pela inicialização inconsistente prejudica o sistema, tornando-o não confiável. Este procedimento não é possível para o sistema (5.2) porque a inconsistência da condição inicial afeta diretamente o valor inicial da variável de restrição.

Através da Tabela 5.5 é possível verificar que o erro de inicialização da variável z foi transferido para a derivada da restrição, tornando o sistema consistente para a inicialização.

Tabela 5.5- Representação da variável de restrição – exemplo 1

Variável k	Sistema (iii)	
	Erro (e)	Integral do Erro (e)
t=0	-0,01	0,000
t=0,1	-0,01	-0,001
t=0,2	-0,01	-0,002
t=0,3	-0,01	-0,003
t=0,4	-0,01	-0,004
t=0,5	-0,01	-0,005
t=0,6	-0,01	-0,006
t=0,7	-0,01	-0,007
t=0,8	-0,01	-0,008
t=0,9	-0,01	-0,009
t=1,0	-0,01	-0,010

As Tabelas 5.6 a 5.8 apresentam a comparação entre a solução analítica e as soluções numéricas das variáveis y , z e x para as formulações (5.1) e (5.3). A solução analítica foi calculada somente com o novo valor de z_0 .

Verifica-se pelos resultados que o monitoramento do erro na variável de restrição é um bom indicador da qualidade da solução.

Tabela 5.6 - Tabela comparativa da variável y - exemplo 1

Variável	Sol. Analítica	Sistema (i)	Sistema (iii)
y			
t=0	1,0099505	1	1
t=0,1	1,1099505	1,1	1,1
t=0,2	1,2099505	1,2	1,2
t=0,3	1,3099505	1,3	1,3
t=0,4	1,4099505	1,4	1,4
t=0,5	1,5099505	1,5	1,5
t=0,6	1,6099505	1,6	1,6
t=0,7	1,7099505	1,7	1,7
t=0,8	1,8099505	1,8	1,8
t=0,9	1,9099505	1,9	1,9
t=1,0	2,0099505	2	2

Tabela 5.7- Tabela comparativa da variável z - exemplo 1

Variável z	Sol. Analítica	Sistema (i)	Sistema (iii)
t=0	0,510	-0,510	-0,510
t=0,1	-0,615995	-0,615	-0,615
t=0,2	-0,73199	-0,730	-0,730
t=0,3	-0,8579851	-0,855	-0,855
t=0,4	-0,993988	-0,990	-0,990
t=0,5	-1,139975	-1,135	-1,135
t=0,6	-1,29597	-1,290	-1,290
t=0,7	-1,461965	-1,455	-1,455
t=0,8	-1,63796	-1,630	-1,630
t=0,9	-1,823955	-1,815	-1,815
t=1,0	-2,01995	-2,010	-2,010

Tabela 5.8 - Tabela comparativa da variável x - exemplo 1

Variável x	Sol. Analítica	Sistema (i)	Sistema (iii)
t=0	0	0	0
t=0,1	0	-1,31501E-17	-5,83067E-17
t=0,2	0	3,59778E-17	2,83783E-17
t=0,3	0	8,28186E-18	7,08084E-17
t=0,4	0	-6,18535E-19	7,96388E-17
t=0,5	0	-9,18283E-18	6,07887E-17
t=0,6	0	-1,77411E-17	3,77165E-17
t=0,7	0	-2,62999E-17	3,21155E-17
t=0,8	0	-3,47827E-17	2,77254E-17
t=0,9	0	-4,31893E-17	2,4546E-17
t=1,0	0	-5,15198E-17	2,25773E-17

5.2. EXEMPLO 2. - PROPOSTO POR PANTELIDES ET AL. (1988) – ÍNDICE 2

$$\dot{x}_1 - x_1 - x_2 - y = 0, \quad (5.18)$$

$$\dot{x}_2 - x_1 + x_2 + y = 0, \quad (5.19)$$

$$x_1 + 2x_2 = 0. \quad (5.20)$$

O sistema de EADs acima apresenta solução analítica da forma:

$$\begin{aligned} x_1(t) &= ke^{4t}, & \dot{x}_1(t) &= 4ke^{4t}, \\ x_2(t) &= -\frac{k}{2}e^{4t}, & \dot{x}_2(t) &= -2ke^{4t}, \\ y(t) &= \frac{7}{2}ke^{4t}. \end{aligned} \quad (5.21)$$

Após aplicar o algoritmo DAGRAFO no exemplo 2, foi gerado o sistema reduzido (5.22):

$$\begin{aligned} \dot{x}_1 - x_1 - x_2 - y &= 0, \\ \dot{x}_2 - x_1 + x_2 + y &= 0, \\ \dot{x}_1 + 2\dot{x}_2 &= 0. \end{aligned} \quad (5.22)$$

a partir das informações dos vetores abaixo:

$$\begin{aligned} X &= (x_1, x_2, \dot{x}_1, \dot{x}_2, y), \\ A &= (3, 4, 0, 0, 0), \\ B &= (0, 0, 4, 0), \\ ASSIGN &= (0, 0, 4, 2, 1). \end{aligned}$$

Com o sistema reduzido determinado é necessário escolher um conjunto de condições iniciais consistentes para dar partida ao código numérico DASSLC, entretanto, como mencionado no capítulo 2, sistemas de índice superior sempre apresentam problemas de inicialização, e o exemplo 2 mostra claramente esta dificuldade.

A dificuldade de inicialização do problema em questão, devido a seu índice elevado, foi estudada por Vieira (1998). Segundo a autora, um conjunto de condições iniciais consistentes deve ser do tipo:

$$\begin{aligned}
 x_1(0) &= \frac{2y(0)}{7}, & \dot{x}_1(0) &= \frac{8y(0)}{7}, \\
 x_2(0) &= -\frac{y(0)}{7}, & \dot{x}_2(0) &= -\frac{4y(0)}{7}.
 \end{aligned}
 \tag{5.23}$$

Arbitrando-se $y(0) = \frac{7}{4}$ obtém-se o seguinte conjunto de condições iniciais para o sistema (5.22):

$$\begin{aligned}
 x_1(0) &= \frac{1}{2}, & \dot{x}_1(0) &= 2, \\
 x_2(0) &= -\frac{1}{4}, & \dot{x}_2(0) &= -1, \\
 y(0) &= \frac{7}{4}.
 \end{aligned}
 \tag{5.24}$$

Após esta etapa, o exemplo 2 foi submetido às três técnicas de inicialização e integração sugeridas anteriormente. Todas as técnicas aproximaram-se da solução analítica, porém a formulação (5.3) apresentou resultados mais precisos que a (5.2). Este fato pode ser analisado através da Tabela 5.9, que representa o desempenho da variável de restrição.

Tabela 5.9 – Variável restrição – exemplo 2

	Sistema (ii)	Sistema (iii)
t=0	0	0
t=0,1	-3,62337e-20	-1,493e-21
t=0,2	-1,48318e-19	-3,020e-21
t=0,3	-3,36269e-19	-4,548e-21
t=0,4	-6,00087e-19	-6,075e-21
t=0,5	3,46940e-16	3,454e-16
t=0,6	-4,59963e-15	2,373e-16
t=0,7	6,92656e-14	-5,107e-16
t=0,8	-8,78597e-13	7,813e-16
t=0,9	4,76801e-12	-1,458e-15
t=1,0	-2,40001e-11	1,078e-15

A solução das variáveis x_1 , x_2 e y , assim como, o erro (\mathbf{e}) e a integral do erro (e), podem ser analisadas nas Figuras 5.6 a 5.10, respectivamente.

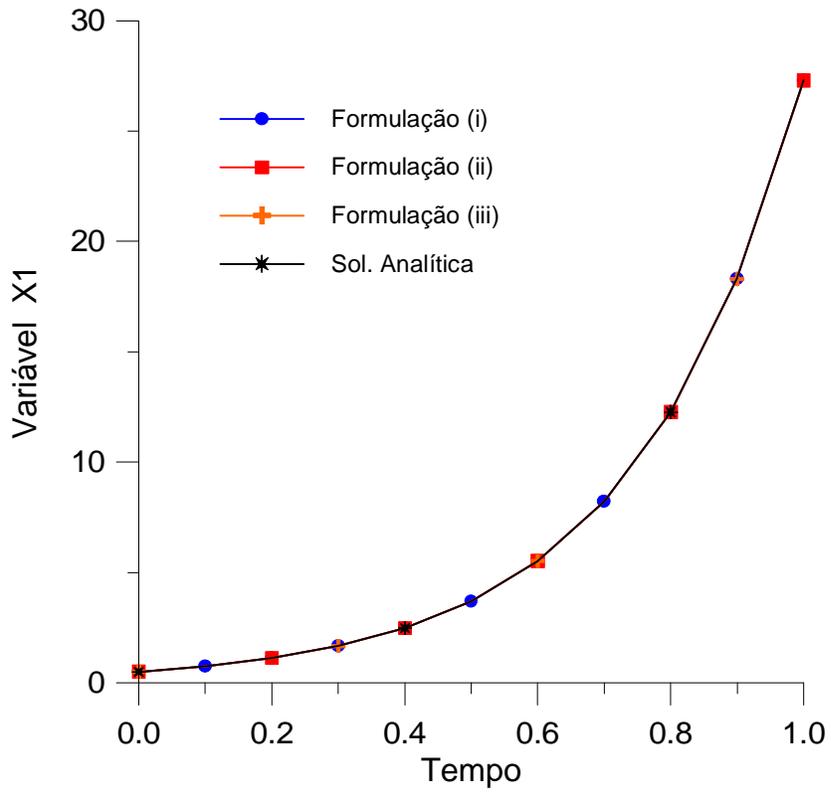


Figura 5.6 - Variável x_1 x t - exemplo 2

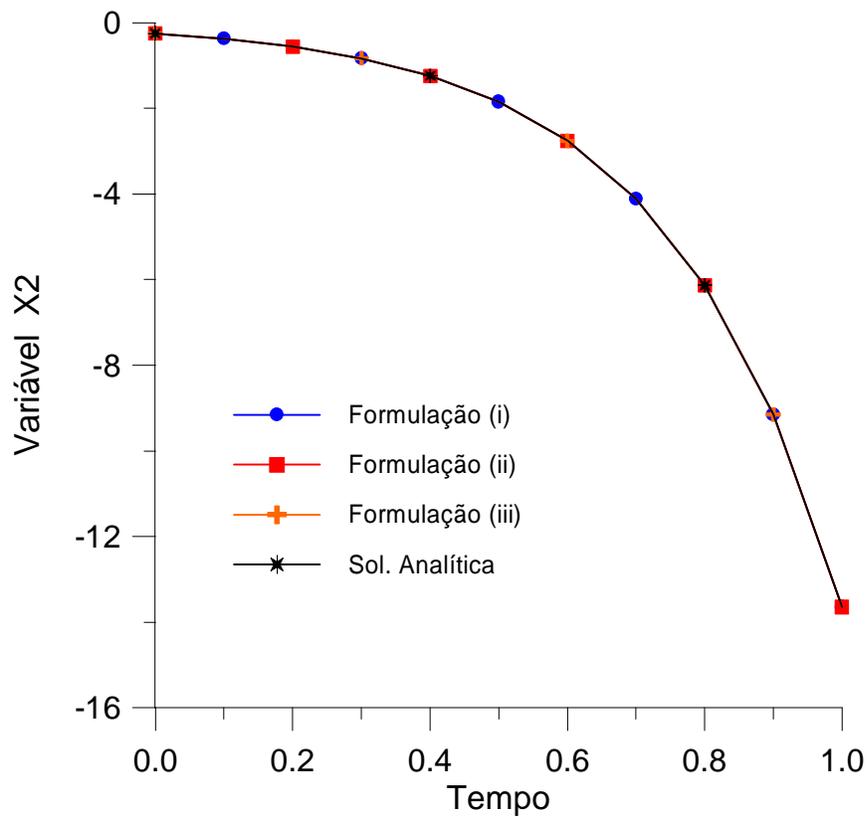


Figura 5.7- Variável x_2 x t - exemplo 2

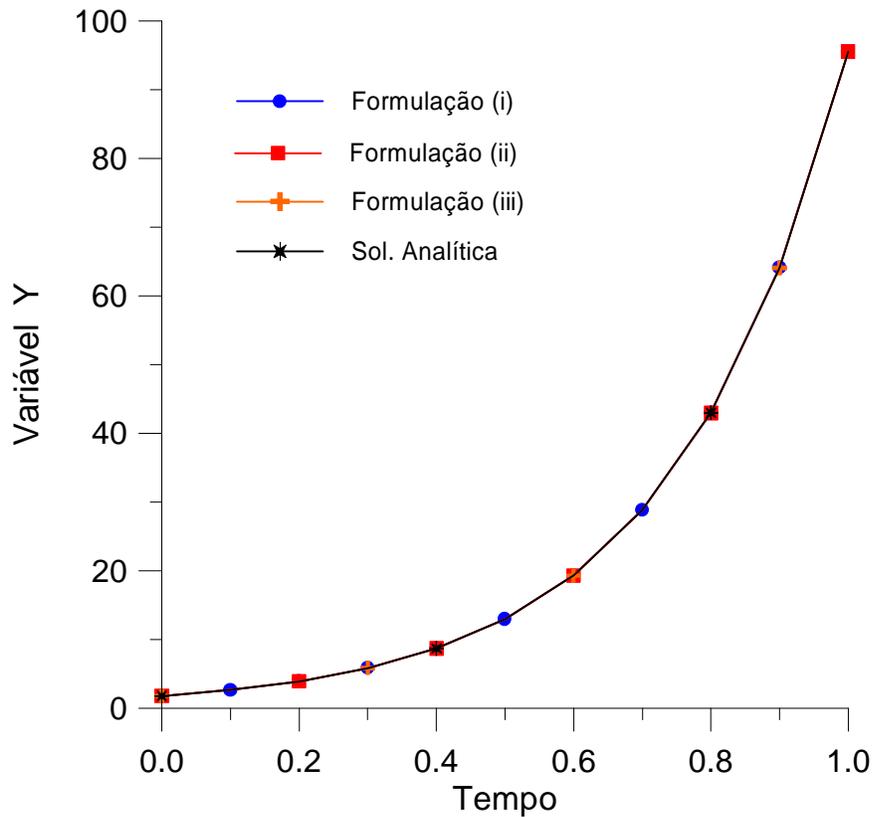


Figura 5.8- Variável $y \times t$ - exemplo 2

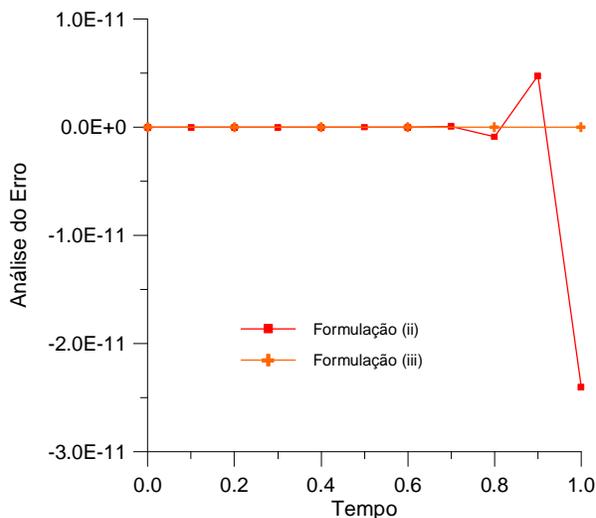


Figura 5.9- Análise do erro (e) - exemplo 2

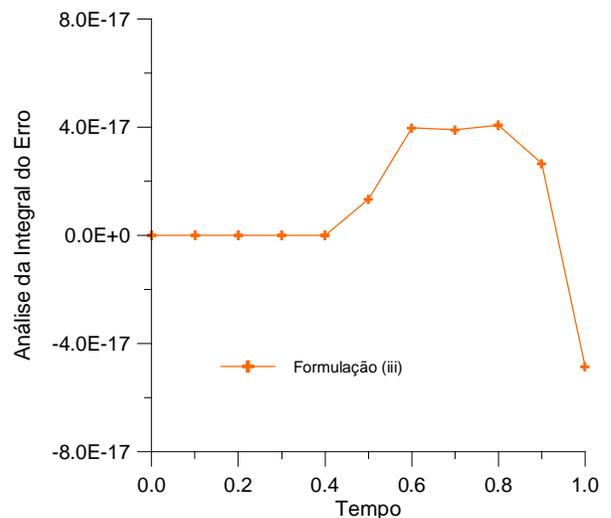


Figura 5.10- Análise da integral do erro (e) - exemplo 2

A Tabela 5.10 compara os resultados das 3 técnicas após aplicar o código numérico em estudo. Observa-se que os resultados são equivalentes para todos métodos. Entretanto, por motivos já mencionados no capítulo 4 não é conveniente utilizar o sistema (5.1). Pela análise do

erro, Figura 5.9, o sistema (5.3) apresenta menor erro de integração que o sistema (5.2), tornando-o mais adequado.

Tabela 5.10 – Tabela comparativa dos métodos de integração – exemplo 2

Total do número de	Sistema (i)	Sistema (ii)	Sistema (iii)
pontos	51	51	51
avaliações do resíduo	62	64	64
avaliações da matriz jacobiana	15	14	14
iterações de Newton	62	64	64
falhas do teste do erro	0	0	0
falhas do teste da convergência	0	0	0
pontos rejeitados	0	0	0
iterações de Newton-Raphson rejeitada	0	0	0
Tempo total (segundos)	0,030	0,04	0,04

5.3. EXEMPLO 3. - PROPOSTO POR CHUNG E WESTERBERG (1990) – ÍNDICE 3

$$\dot{x}_1 - x_2 - 2t = 0, \quad (5.25)$$

$$\dot{x}_2 - y - 5 = 0, \quad (5.26)$$

$$x_1 - 4t = 0. \quad (5.27)$$

O exemplo 3, por apresentar índice superior, deve ser reduzido a um sistema de EADs de índice 1 ou 0. Esta redução é realizada pela aplicação do algoritmo DAGRAFO que gera as seguintes informações:

$$X = (x_1, x_2, \dot{x}_1, \dot{x}_2, z),$$

$$A = (3, 4, 6, 0, 0, 0),$$

$$B = (5, 0, 4, 6, 0, 0),$$

$$ASSIGN = (0, 0, 1, 5, 2, 6).$$

que resulta no seguinte sistema reduzido:

$$\begin{aligned}
\dot{x}_1 - x_2 - 2t &= 0, \\
\dot{x}_2 - y - 5 &= 0, \\
z - \dot{x}_2 - 2 &= 0, \\
z &= 0.
\end{aligned}
\tag{5.28}$$

O conjunto de equações (5.28) deve ser transformado nos sistemas (5.1), (5.2) e (5.3) a fim de analisar a eficácia de cada técnica. Para esta análise é necessário utilizar um conjunto consistente de condições iniciais, como o apresentado abaixo:

$$\begin{aligned}
x_1(0) &= 0, & \dot{x}_1(0) &= 4, \\
x_2(0) &= 4, & \dot{x}_2(0) &= -2, \\
y(0) &= -7, & \dot{y}(0) &= 0, \\
z(0) &= 0, & \dot{z}(0) &= 0.
\end{aligned}
\tag{5.29}$$

A solução analítica do sistema original é dada por:

$$\begin{aligned}
x_1(t) &= 4t, & x_2(t) &= 4 - 2t, \\
y(t) &= -7, \\
\dot{x}_1(t) &= 4, & \dot{x}_2(t) &= -2.
\end{aligned}
\tag{5.30}$$

Após a execução do código DASSLC obteve-se a solução das variáveis x_1 , x_2 , y e z que podem ser analisadas através das Figuras 5.16 a 5.19, respectivamente.

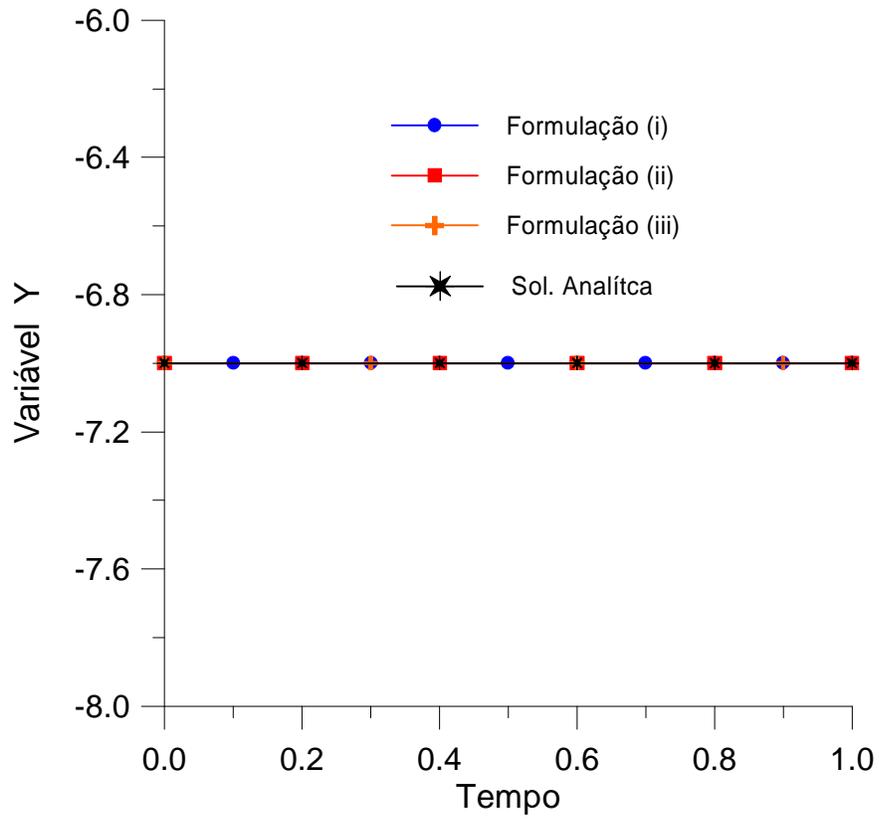


Figura 5.11- Variável y x t - exemplo 3

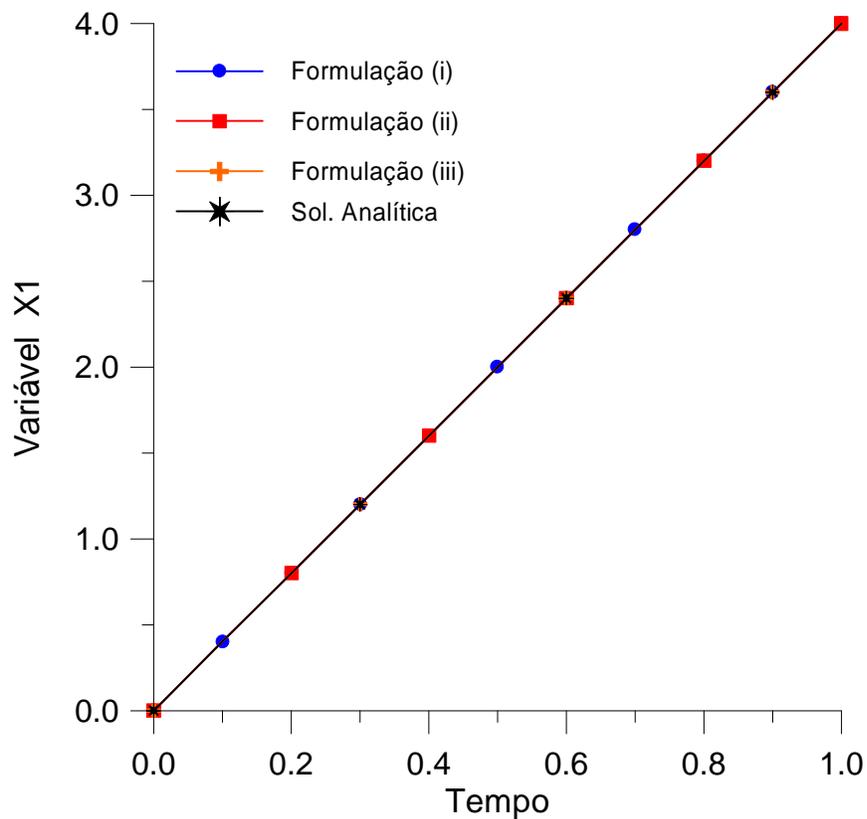


Figura 5.12 - Variável x_1 x t - exemplo 3

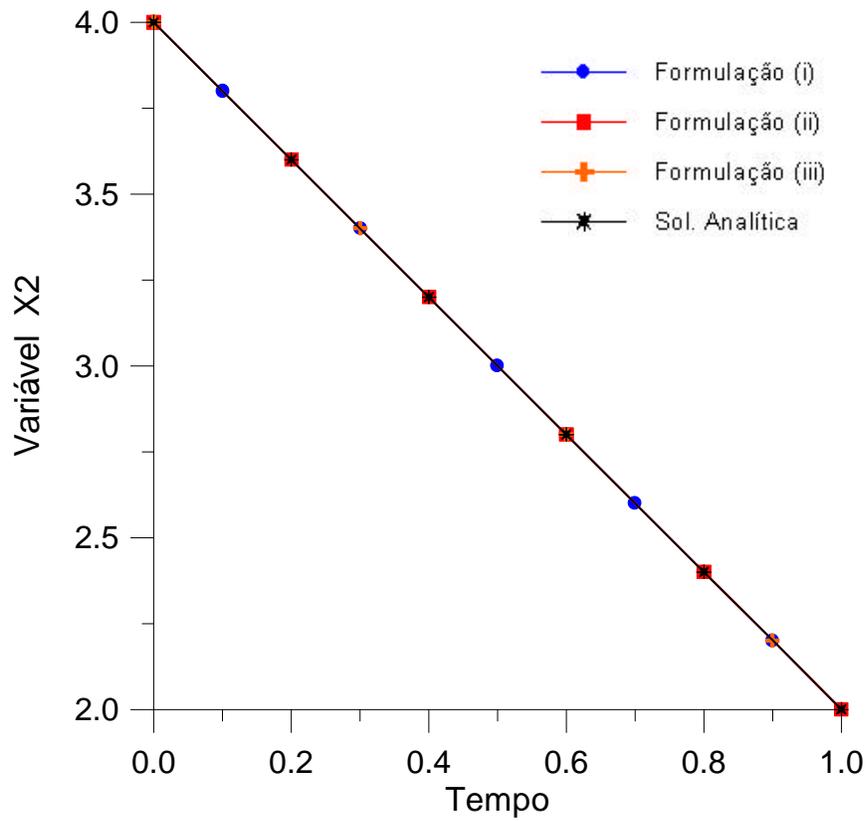


Figura 5.13 - Variável $x_2 \times t$ - exemplo 3

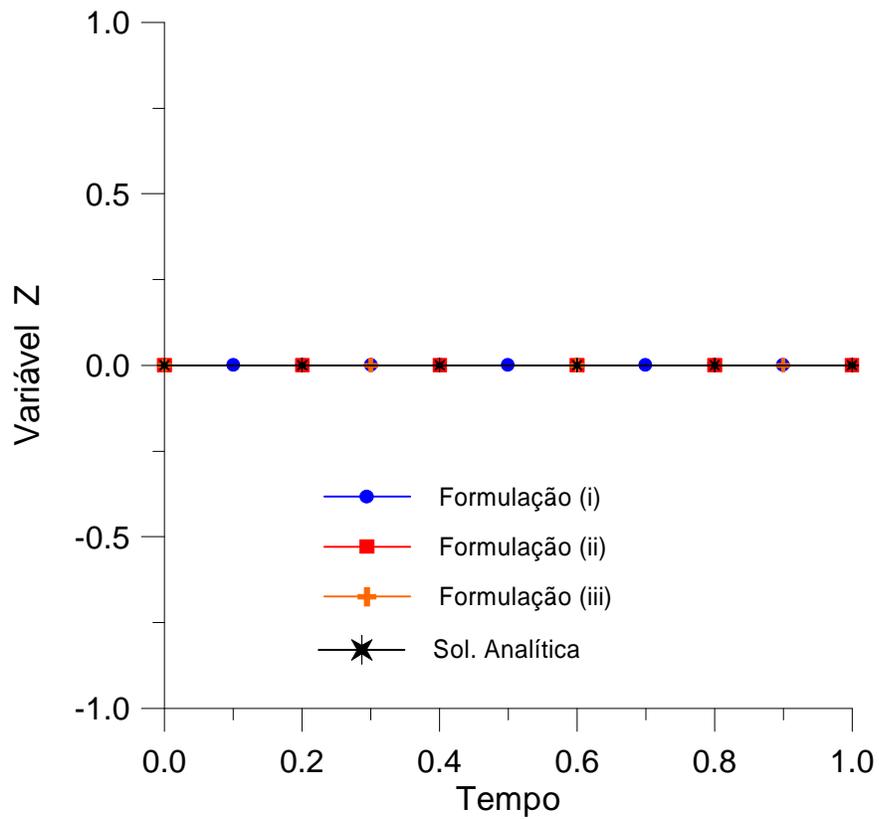


Figura 5.14 - Variável $z \times t$ - exemplo 3

Analisando as respostas das variáveis x_1 , x_2 , y e z , através das Figuras 5.11 a 5.14, observa-se que as três técnicas utilizadas mostraram-se equivalentes.

No processo de redução de índice do sistema original, as seguintes restrições foram rejeitadas do sistema reduzido (equações 3 e 4 do vetor ASSIGN):

$$x_1 - 4t = 0, \quad (5.31)$$

$$\dot{x}_1 - 4 = 0, \quad (5.32)$$

conseqüentemente, são necessárias duas variáveis de restrição para cada uma das formulações (5.2) e (5.3).

As Figuras 5.15 a 5.18 apresentam o erro e a integral do erro, onde verifica-se que os sistemas (5.2) e (5.3) mostraram-se satisfatórios e com desempenho equivalente. Isto também pode ser observado na Tabela 5.11, com as informações estatísticas das soluções numéricas.

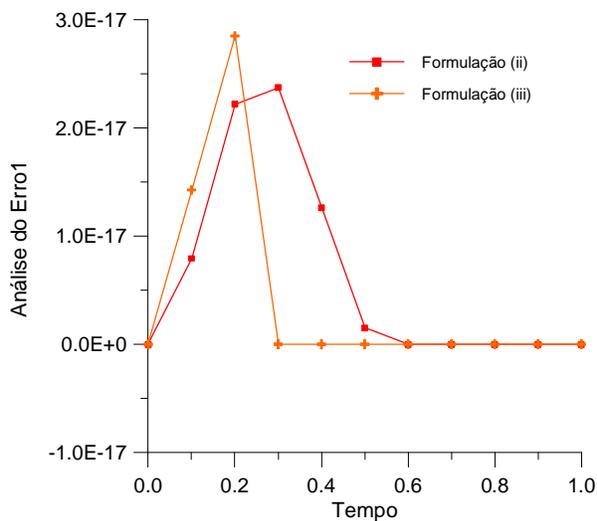


Figura 5.15- Análise do erro (e) - exemplo 3

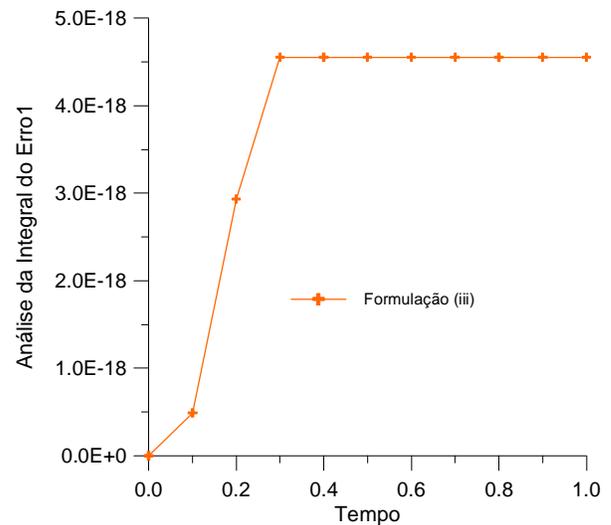


Figura 5.16- Análise da integral do erro (e) - exemplo 3

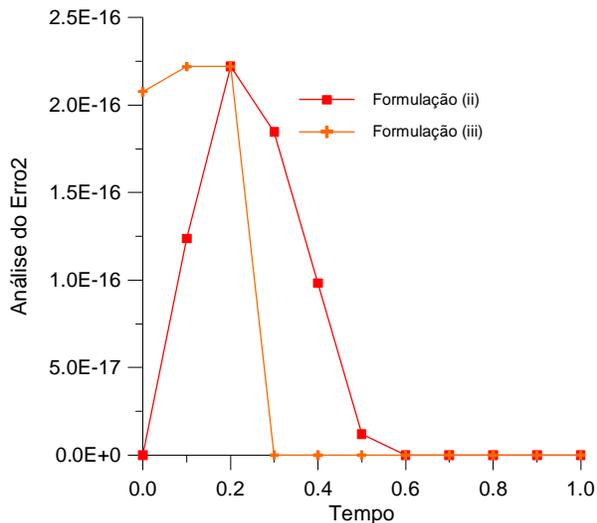


Figura 5.17- Análise do erro (e) - exemplo 3

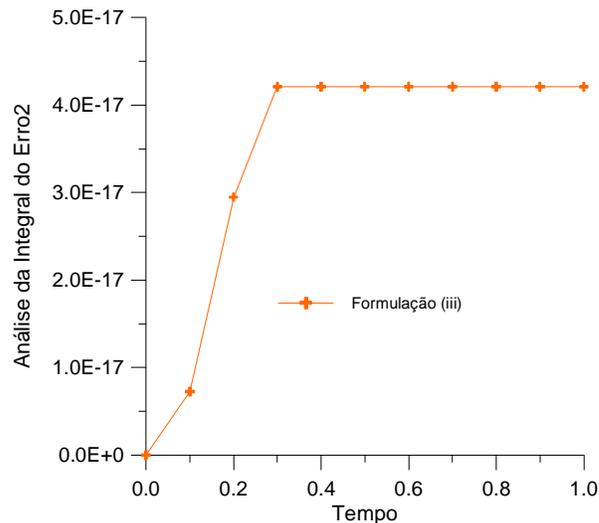


Figura 5.18- Análise da integral do erro (e) - exemplo 3

Tabela 5.11 – Tabela comparativa dos métodos de integração – exemplo 3

Total do número de	Sistema (i)	Sistema (ii)	Sistema (iii)
pontos	31	31	31
avaliações do resíduo	27	26	26
avaliações da matriz jacobiana	26	25	25
iterações de Newton	27	26	26
falhas do teste do erro	0	0	0
falhas do teste da convergência	0	0	0
pontos rejeitados	0	0	0
iterações de Newton-Raphson rejeitada	0	0	0
Tempo total	0.04	0.03	0.03

EXEMPLO 4. – PROPOSTO POR LOGSDON E BIEGLER (1993) – ÍNDICE 3

O exemplo descrito a seguir representa uma, entre as diversas, aplicabilidade de sistemas de EADs dentro da engenharia química. Este refere-se ao controle de pureza do produto obtido na fase de produção de uma coluna de destilação em batelada. A descrição acima pode ser ilustrada através da Figura 5.19:

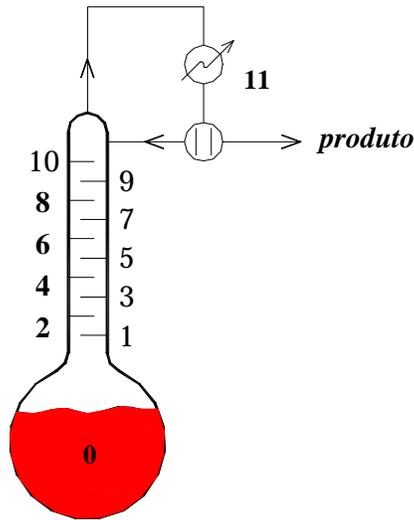


Figura 5.19 – Coluna de destilação em batelada – exemplo 4

O modelo matemático proposto por Logsdon e Biegler (1993) é descrito pelo seguinte sistema de EADs:

$$\dot{x}_0^j = \frac{V}{H_0} \left(x_0^j - y_0^j + \frac{R}{R+1} (x_1^j - x_0^j) \right), \quad (5.33)$$

$$\dot{x}_i^j = \frac{V}{H_i} \left(y_{i-1}^j - y_i^j + \frac{R}{R+1} (x_{i+1}^j - x_i^j) \right) \quad i = 1, \dots, np, \quad (5.34)$$

$$\sum_{j=1}^{nc} y_i^j = 1 \quad i = 0, \dots, np+1, \quad (5.35)$$

$$\dot{x}_{np+1}^j = \frac{V}{H_{np+1}} (y_{np}^j - x_{np+1}^j), \quad (5.36)$$

$$\dot{H}_0 = -\frac{V}{R+1}, \quad (5.37)$$

$$x_{np+1}^j = 0,998, \quad (5.38)$$

onde:

V vazão molar de vapor em todos os estágios;

H_i número de moles em cada estágio da coluna;

R razão do refluxo;

i é o número de estágio, sendo igual a 0 para o refeedor e $np+1$ para o condensador;

j é o número do componente, variando de 1 a $nc-1$ (número de componentes);

x fração molar na fase líquida;

y fração molar na fase vapor em função de x e da temperatura T do estágio, sendo

calculado pela relação de equilíbrio $y_i^j = K(T_i, x_i^j) x_i^j$, $i = 0, \dots, np+1$

x_i^{nc} fração molar na fase líquida do componente nc no estágio i , representada por

$$x_i^{nc} = 1 - \sum_{j=1}^{nc-1} x_i^j, \quad i = 0, \dots, np+1$$

Os autores sugerem $nc = 2$ componentes e $np = 10$ pratos, desta forma o sistema de equações algébrico-diferencial original é composto por 26 equações, as quais 13 representam as equações diferenciais e as demais representam as algébricas e, 26 variáveis as quais x_i^j ($j = 1, \dots, nc-1; i = 0, \dots, np+1$) e H_0 representam as variáveis diferenciais e T_i ($i = 0, \dots, np+1$) e R , representam as variáveis algébricas. O processo final deve apresentar 99,8% de pureza para o elemento que está sendo retirado do sistema, equação (5.38).

De forma genérica, é possível estabelecer a relação existente entre o número de equações e o número de variáveis. Sabe-se que para qualquer número de componentes e pratos, o sistema contém $(2nc-2)(np+2)+2$ equações, sendo $(nc-1)(np+2)+1$ equações diferenciais e as demais equações algébricas e, $(nc-1)(np+2)+1$ variáveis diferenciais e $(nc-1)(np+2)+1$ variáveis algébricas. O fato da variável R não estar presente em equação algébrica alguma torna o sistema de índice superior.

Analisando as condições descrita acima percebe-se que aumentando o número de componentes e/ou o número de pratos, o sistema passará a fornecer um número elevado de equações e variáveis, tornando-se difícil estabelecer o conjunto de condições iniciais consistentes sem ter conhecimento físico do problema em estudo.

De forma análoga aos exemplos anteriores, às condições iniciais devem satisfazer, também, o sistema estendido. Assim, torna-se necessário determinar quais equações devem ser

diferenciadas para reduzir o índice diferencial do sistema em estudo. Nesta etapa a matriz de iteração do sistema original deve ser fornecida ao código DAGRAFO. Após a execução do código obteve-se o seguinte arquivo de saída:

$$\begin{aligned}
 X &= \begin{pmatrix} x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, H_0, \\ \dot{x}_0, \dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4, \dot{x}_5, \dot{x}_6, \dot{x}_7, \dot{x}_8, \dot{x}_9, \dot{x}_{10}, \dot{x}_{11}, \dot{H}_0, \\ T_0, T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}, T_{11}, R \end{pmatrix}, \\
 A(j) &= \begin{pmatrix} 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 0, 0, 0, 0, 0, \\ 0, 0, 0, 0, 0, 0, 40, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 41, 0, 0, 0, 0 \end{pmatrix}, \quad (5.39) \\
 B(j) &= (0, 28, 29, 0, 0, 27, 30, 0, 0, 0), \\
 ASSIGN(j) &= \begin{pmatrix} 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 29, \\ 23, 24, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 21, 30, 28 \end{pmatrix}.
 \end{aligned}$$

Analisando o vetor $ASSIGN$ e $B(j)$, percebe-se que o novo sistema é resultante de uma diferenciação das equações (5.35) do estágio np , (5.36) e duas diferenciações da equação (5.38), caracterizando-se como um sistema de índice 3. Após estas diferenciações o novo sistema passa a apresentar quatro novas equações e duas novas variáveis: \ddot{x}_{11} e \dot{T}_{10} . O sistema estendido apresenta 30 equações e 41 variáveis, resultando em 11 graus de liberdade.

Com o sistema estendido já determinado a próxima etapa refere-se ao uso do código DASSLC. Este necessita de um conjunto consistente de condições iniciais, como o dado abaixo, para dar início a integração do sistema:

$$\begin{aligned}
x_0 &: 0.55, & x_1 &: 0.59865556, & x_2 &: 0.64731111, \\
x_3 &: 0.69596667, & x_4 &: 0.74462222, & x_5 &: 0.79327778, \\
x_6 &: 0.84193333, & x_7 &: 0.89058889, & x_8 &: 0.9879, \\
x_9 &: 0.9879, & x_{10} &: 0.99492775, & x_{11} &: 0.998, \\
T_0 &: 91.34141035, & T_1 &: 90.02305247, & T_2 &: 88.751938268, \\
T_3 &: 87.525391, & T_4 &: 86.34092796, & T_5 &: 85.19623785, \\
T_6 &: 84.08917402, & T_7 &: 83.01773809, & T_8 &: 80.97443124, \\
T_9 &: 80.97443124, & T_{10} &: 80.8317286, & T_{11} &: 80.76954276. \\
H_0 &: 100, & R &: 10.06032282, \\
\dot{x}_1 &: 0, & \dot{T}_{10} &: 0.
\end{aligned} \tag{5.40}$$

Após a execução do código em estudo, deseja-se analisar o comportamento do sistema de EADs com e sem restrição, após a redução do sistema. O exemplo em questão foi submetido às técnicas de integração descritas no capítulo 4, cujos resultados podem ser visualizadas nas Figuras 5.20 a 5.23, através das variáveis x_0 , T_0 , R e H_0 , respectivamente:

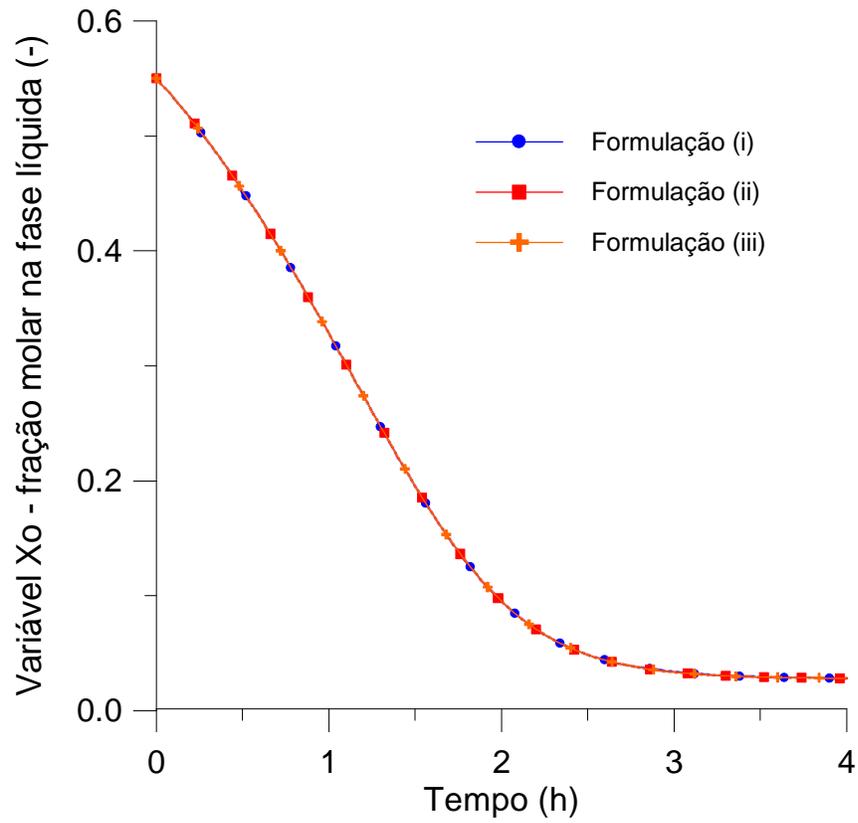


Figura 5.20 – Variável x_0 – exemplo 4

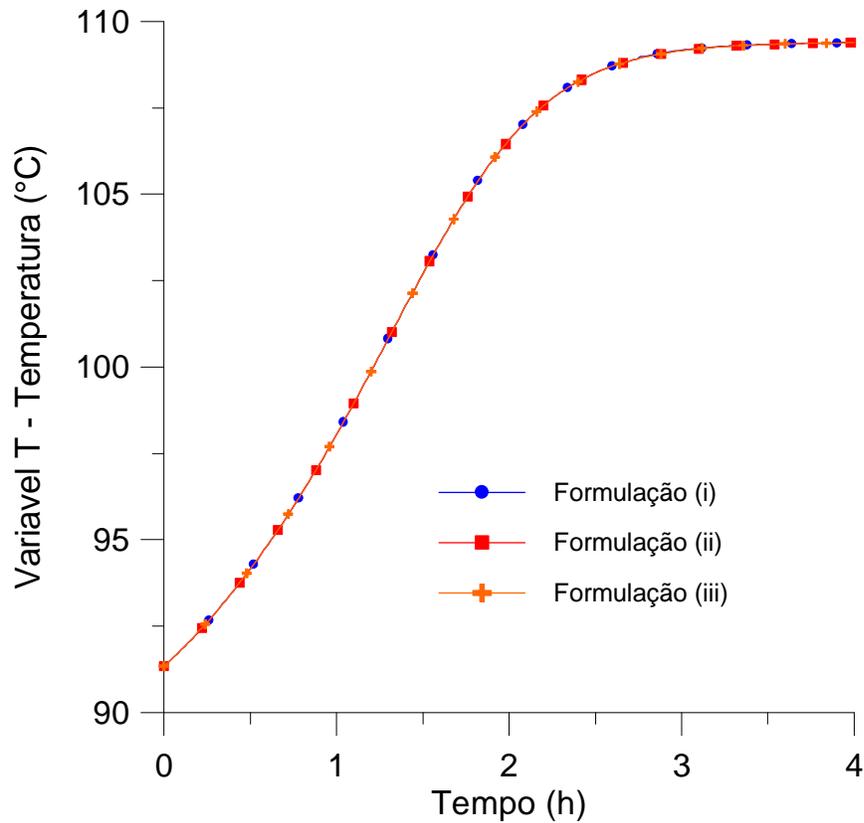


Figura 5.21 – Temperatura (T_0) – exemplo 4

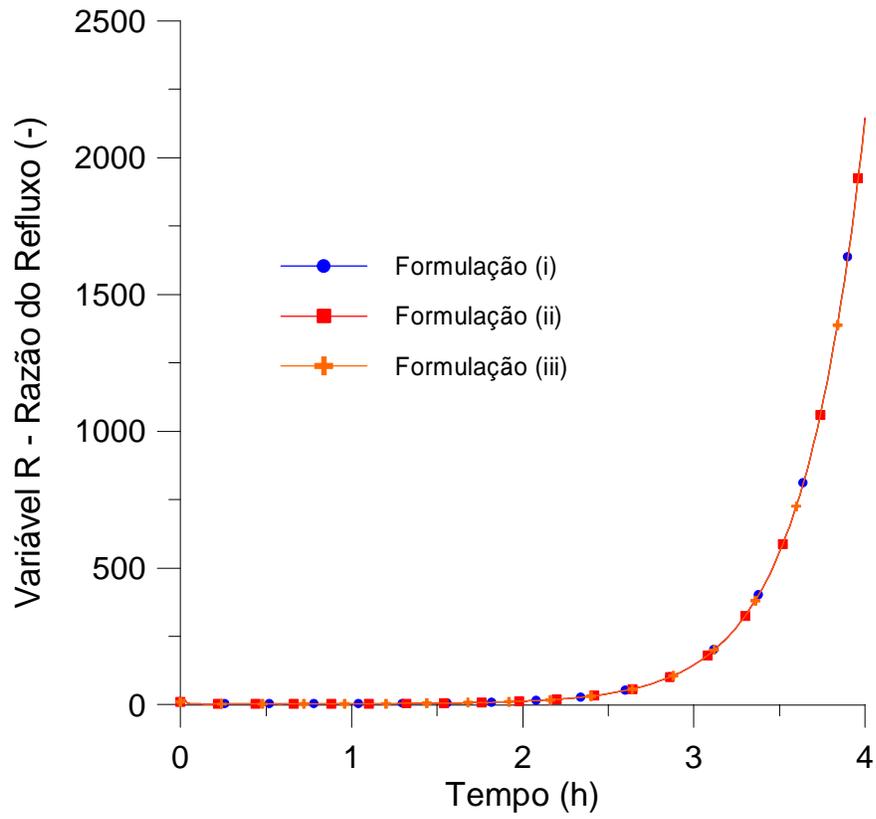


Figura 5.22 – Razão do refluxo (R) – exemplo 4

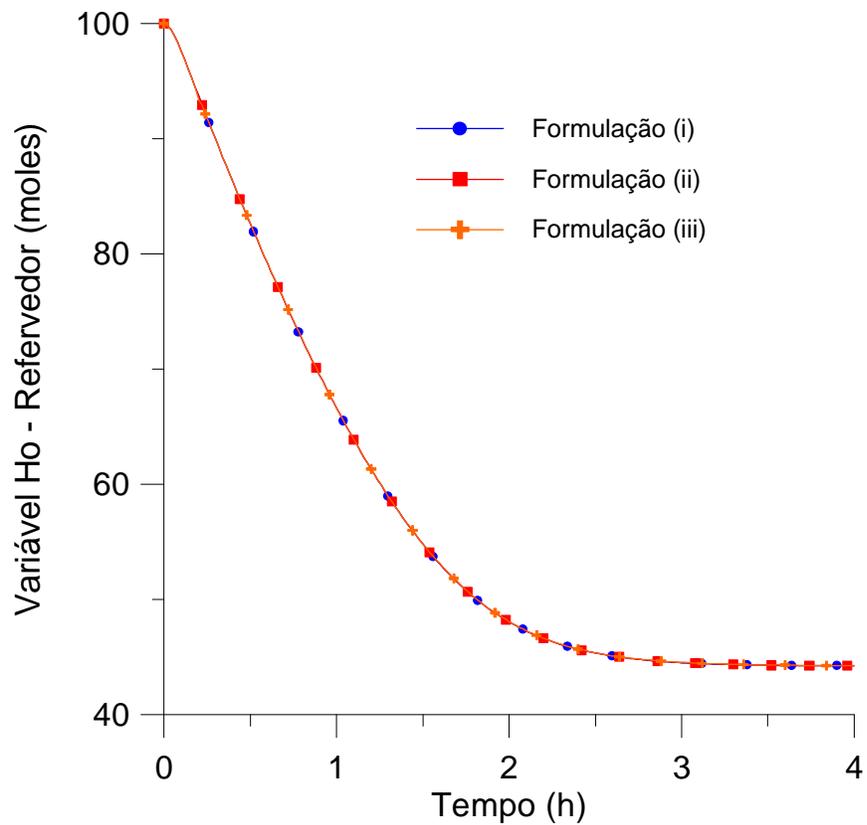


Figura 5.23 – Refluxo (H_0) – exemplo 4

Analisando os resultados obtidos, conforme Figuras 5.20 a 5.23, conclui-se que o código numérico em estudo apresentou desempenho satisfatório, pois demonstrou comportamento semelhante ao descrito por Costa Jr et. al., 2001. A análise física do exemplo 5 não é objetivo deste trabalho, entretanto, se faz necessário alguns comentários.

A variável x_0 , Figura 5.20, mostra o processo de destilação do componente mais leve, que é o produto desejado. A medida que sua concentração dentro da coluna vai reduzindo, é necessário um correspondente acréscimo na temperatura, Figura 5.21, para garantir a separação dos componentes.

Observa-se que a medida que a composição do componente desejado se aproxima de zero no fundo da coluna (Figura 5.20 para $t > 2,5$ h), é necessário elevar a razão de refluxo no topo da coluna, Figura 5.22, para garantir a especificação do produto ($x_{11} = 0,998$). Nesta altura a produção está em níveis baixos, como pode-se observar na pequena redução da quantidade de material na fundo da coluna, Figura 5.23.

O objetivo em estudo é a eficácia das técnicas de inicialização e integração. Desta forma, através da análise gráfica, afirma-se que os comportamentos das três técnicas de inicialização e integração são similares, não fornecendo visualmente a melhor técnica de integração. Entretanto a análise do erro (e), Figura 5.24 e 5.26 e a integral do erro (e), Figuras 5.27 e 5.27, juntamente com os dados estatísticos dos método de integração, Tabela 5.12, fornecem uma melhor análise das técnica de integração.

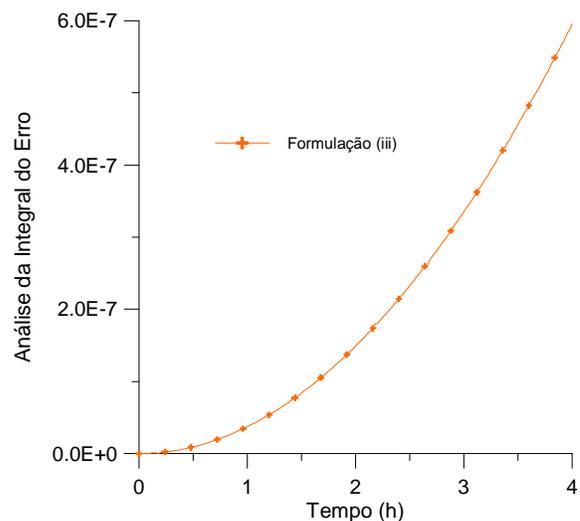
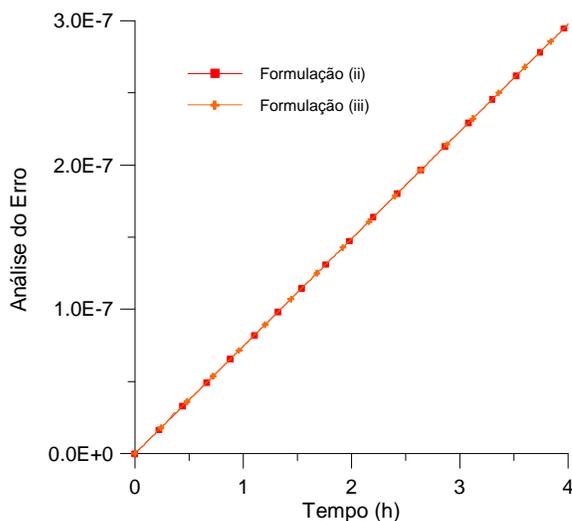


Figura 5.24 – Análise do erro (e) - exemplo 4

Figura 5.25 – Análise da integral do erro (e) - exemplo 4

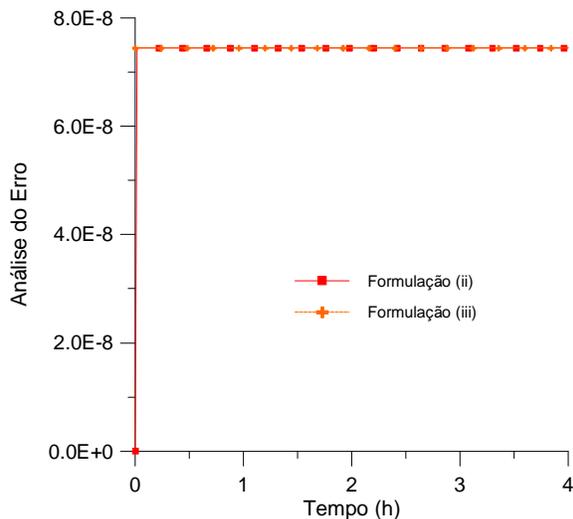


Figura 5.26 – Análise do erro (e) - exemplo 4

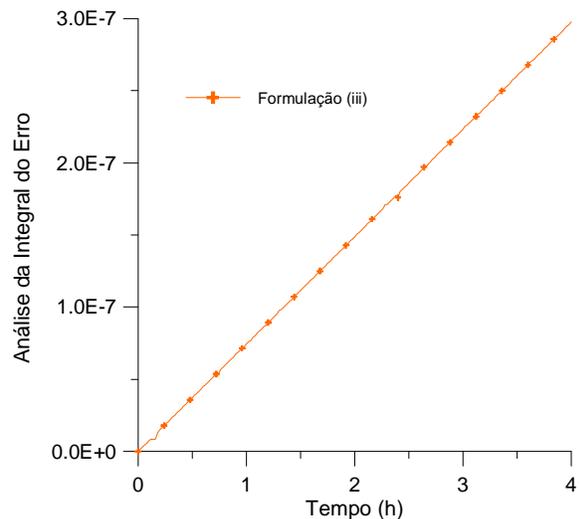


Figura 5.27 – Análise da integral do erro (e) - exemplo 4

Tabela 5.12 – Tabela comparativa dos métodos de integração – exemplo 4

Total do número de	Sistema (i)	Sistema (ii)	Sistema (iii)
Pontos	344	340	358
avaliações do resíduo	484	445	498
avaliações da matriz jacobiana	51	50	59
iterações de Newton	484	445	498
falhas do teste do erro	15	22	19
falhas do teste da convergência	0	0	0
pontos rejeitados	15	22	19
iterações de Newton-Raphson rejeitada	89	87	75
Tempo total (segundos)	0,34	0,36	0,39

A análise do erro (e) e a integral do erro (e) mostraram-se equivalentes para as técnicas (5.2) e (5.3). A tabela comparativa, Tabela 5.12, apresentou resultados diferentes, porém não suficientes para afirmar que um método seja extremamente melhor que outro. Entretanto por motivos já mencionados ao longo desta análise, recomenda-se utilizar o sistema estendido que apresenta a derivada da variável de restrição, sendo este um elemento importante para verificar o quanto o método em estudo torna-se confiável. Outro elemento que deve ser levado em consideração quando o sistema de EADs é composto por um número elevado de equações e variáveis é o tempo de execução do código numérico. Para o exemplo em estudo, este também manteve-se muito próximo para todas as três técnicas de integração.

6. CONCLUSÕES E SUGESTÕES

Ao longo do texto, vários itens foram abordados, visando concluir o processo de implementação de técnicas para inicialização consistente e integração de sistemas de EADs ordinárias de índice elevado. Buscou-se, em primeiro lugar, o aprimoramento conceitual, indispensável para o desenvolvimento prático deste processo.

Este estudo envolveu modelagem matemática, para exemplos puramente numéricos, e conceitos físicos para exemplos aplicados no mundo real, partida de uma coluna de destilação em batelada, por exemplo. Todos os modelos matemáticos aqui apresentados foram analisados a partir da solução do sistema estendido, obtido pelo processo de redução de índice.

A análise dos resultados apresentados permite concluir que a eficácia de cada solução leva em consideração as observações dos dados obtidos a partir das técnicas de inicialização e a estrutura da matriz jacobiana do sistema discretizado. As três técnicas de integração utilizadas apresentaram diferenças significativas quando o sistema de EADs em estudo apresenta problema de condicionamento da matriz jacobiana.

A técnica de integração que utiliza o sistema estendido com as derivadas das variáveis de restrição é a técnica mais conveniente. Esta apresentou melhor desempenho computacional, inclusive quando a matriz jacobiana do sistema apresentou problema de mau condicionamento, além de garantir a presença de todas as variáveis que compõem o sistema original. Este fato não ocorre para a formulação somente com o sistema reduzido, pois, além de “rejeitar” determinadas equações que podem conter variáveis com significado físico importante e/ou representar restrições implícitas no sistema, pode gerar uma solução não exata.

Com base na comparação dos resultados numéricos a técnica de integração do sistema estendido com as derivadas das variáveis de restrição apresentou-se mais eficiente que o sistema estendido com as variáveis de restrição. Um dos motivos que permite esta afirmação é o fato do sistema estendido com as variáveis de restrição apresentar-se bastante falho quando submetido a sistemas mal condicionados, além de não possibilitar a solução partindo de condições iniciais inconsistentes.

Outro fato que favorece a técnica do sistema estendido com as derivadas das variáveis de restrição é a facilidade de tratar as reinicializações do sistema. Caso o sistema trabalhe com reinicialização ao longo do processo de integração, este necessita sempre de um novo cálculo das condições iniciais consistentes. Este problema é facilmente resolvido pela formulação (iii), capítulo 5, pois esta atribui na inicialização valor zero às variáveis de restrição, e distribui o erro

de inicialização para as derivadas das variáveis de restrição. Esta facilidade não ocorre para a formulação (ii), capítulo, pois esta não inicializa o sistema se as condições iniciais não forem consistentes. A maior preocupação com este tipo de situação é saber aceitar ou rejeitar o conjunto solução fornecido após o processo de integração numérica, pois o erro de integração, devido as condições iniciais inconsistentes, estará presente no conjunto solução.

Como sugestão para trabalhos futuros, pode-se desenvolver técnicas para o controle do erro local de integração utilizando as variáveis de restrição. Poderia-se desenvolver critérios para o ajuste do passo e da ordem da integração de modo a evitar a propagação do erro devido ao uso de condições iniciais inconsistentes, ou devido aos erros de truncamento e arredondamento. Sugere-se para isto o uso da técnica do sistema estendido com a derivada das variáveis de restrição, por conter informações sobre o erro local e o erro acumulado das restrições rejeitadas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ASCHER, U.M., PETZOLD, L., 1998, “**Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations**”, Classics in Applied Mathematics Series, SIAM.
- [2] BACHMANN, R., BRÜLL, L., MRZIGLOD, T., and PALLASKE, 1990, “**On Methods for Reducing the index of Differential Algebraic Equations**”, Comp. Che. Engng., 14(11), 1271-1273.
- [3] BRENAN, K. E., 1983, “**Stability and Convergence of Difference Approximations for Higher Index Differential-Algebraic Systems with Applications in Trajectory Control**”, Department of Mathematics, UCLA, Los Angeles, CA.
- [4] BRENAN, K.E., CAMPBELL, S.L., PETZOLD, L.R., 1989, “**Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations**”, New York, 1ª edição, Elsevier Sc Publishing Co Inc.
- [5] CAMPBELL, S. L. 1986, “**Consistent Initial Conditions for linear time varying Singular Systems, in Frequency Domain and State Space Methods for Linear Systems**”, Elsevier Science (North-Holland) Amsterdam.
- [6] CAMPBELL, S.L., 1988, “**A Computation Method for General Higher-Index Nonlinear Singular Systems of Differential Equations**”, IMACS Conference.
- [7] CAMPBELL, S.L., 1983, “**Consistent Initial Conditions For Singular Nonlinear Systems**”, Circuits Systems Signal Process, Vol. 2 nº. 1 , 44-55.
- [8] CHUNG, Y., WESTERBERG, A.W., 1991, “**Solving Stiff DAE Systems as “Near” Index Problems**”, Technical Report, Engineering Research Design Center, Carnegie Mellon University.

- [9] CHUNG, Y., WESTERBERG, A.W., 1990, “**A proposed Numerical Algorithm for Solving Nonlinear Index Problems**”, Ind. Eng. Chem. Res. ,29, 1234-1239.
- [10] COSTA Jr., E. F., 2001, “**Caracterização Estrutural Automática de Sistemas Algébrico-Diferenciais**”, Proposta de tese de doutorado, COPPE/UFRJ, Rio de Janeiro.
- [11] COSTA Jr.,E. F., SECCHI, A. R., BISCAIA Jr, E. C., 2000, “**Automatic Structural Characterization of DAE Systems**”, In: Proc. ESCAPE 11, Kolding, Denmark, S 109-114.
- [12] DUFF, I. S., 1981 “**On algorithms for Obtaining a Maximum Transversal**”, ACM Trans. Math. Software, 7, p. 315-330.
- [13] DOTTO, O. J., 1998, “**Métodos Matriciais e Computacionais**” Notas de aula, convênio UCS/UFRGS.
- [14] GEAR, C.W., 1990, “**Differential Algebraic Equations Index Transformation**”, SIAM Journal of Analysis, 27(6), 1527-1534.
- [15] GEAR, C.W., 1971, “**The Simultaneous Numerical Solutions of Differential-Algebraic Equations**”, IEEE Trans. Circuit Theory, CT-18, pp. 89-95.
- [16] GEAR, C.W., PETZOLD, L.R., 1984, “**ODE Methods for the Solution of Differential-Algebraic Systems**”, SIAM Journal of Numerical Analysis, v21, n.4 (Ago), pp. 716-728.
- [17] GEAR, C.W., 1988, “**Differential-Algebraic Equations Index Transformations**”, SIAM Journal of Scientific and Statistical Computing, v.9, n.1 (Jan), pp.39-47.
- [18] HAIRER, E., WANNER, G., 1991, “**Solving Ordinary Differential Equations II – Stiff and Differential-Algebraic Problems**”, Springer-Verlag Berlin Heidelberg, USA.
- [19] HAIRER, E., LUBICH, C., LOUCHE, M.,1989, “**The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods**”, Springer-Verlag, Berlin, Lectures Notes in Mathematics, 1409.

- [20] LEIMKEIHLER, B., PETZOLD, L.R., GEAR, C.W., 1991, “**Approximations Methods for the Consistent Initialization of Differential-Algebraic Equations**”, SIAM Journal of Analysis, v.28 (1), 205-226.
- [21] LOGSDON, J.S., BIEGLER, T., 1993, “**Accurate Determination of Optimal Reflux Polices for the Maximum Distillate Problem in Batch Distillation**”, Ind. Eng. Chem. Res., Vol. 32 n°4, 692-700.
- [22] MARTINSON. W. S., BARTON, P., 2000, “**A Differentiation Index for Partial Differential-Algebraic Equations**”, SIAM J. Society for Industrial and Applied Mathematics, v. 21(6), pp. 2295-2315.
- [23] MATSSON, S. E., SÖDERLIND, G., 1993, “**Index Reduction in Differential-Algebraic Equations Using Dummy Derivatives**”, SIAM J. Sci.. Stat. Comp., v. 14(3) pp. 1-16.
- [24] MURATA, V.V, 1996, “**Caracterização Simbólica de Equações Algébrico-Diferenciais por um Sistema de Álgebra Computacional com Aplicações na Engenharia Química**”, Tese de Doutorado, PEQ/COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- [25] PANTELIDES, C.C, 1988, “**The Consistent Initialization of Differential-Algebraic Systems**”, SIAM Journal of Sci. Stat. Comput., , v.9 pp. 213-232.
- [26] PANTELIDES, C.C, BARTON, P.I., 1992, “**Equation – Oriented Dynamic Simulation: Current Status and Future Perspectives**”, In: Proc. Escape – 2, Toulouse, France.
- [27] PETZOLD, L., 1982, “**Differential/Algebraic Equations are not ODEs**”, SIAM Journal of Sci. Stat. Comput., , v.3 (3) pp. 367-384.
- [28] PETZOLD, L., 1982, “**A description of DASSL: a Differential/Algebraic System Solve**”, Proceedings of IMACS Word Congress, Montreal, Canada.

- [29] PETZOLD, L. R., 1989, **“DASSL Code, Version 1989”**, Computing and Mathematics Research Division, Lawrence Livermore National Laboratory, L316, PO Box 808, Livermore, CA 94559.
- [30] RASCOL, E., MEYER, M., LANN, J. M. LE, PRESVOST, M., 1998, **“Numerical Problems Encountered in the Simulation of Reactive Absorption: DAE Index Reduction and Consistent Initialization”**, Computers Chem. Engng. v. 22 pp. S929-S932.
- [31] SECCHI, A. R., 1992, **“DASSLC User’s Manual Version 1.0”**, Universidade Federal do Rio Grande do Sul, DEQUI, Porto Alegre, RS, Brasil.
- [32] SECCHI, A. R., PEREIRA, F., A., 1997, **“DASSLC User’s Manual version 2.0”**, Universidade Federal do rio Grande do Sul, DEQUI, Porto Alegre, RS, Brasil.
- [33] UNGER, J., KRÖNER, A., MARQUARDT, W., 1995, **“Structural Analysis of Differential-Algebraic Equations Systems – Theory and Applications”**, Computers Chemical Engineering, v19(8) pp. 867-882.
- [34] VIEIRA, R. C., 1998, **“Métodos Diretos para Inicialização de Sistemas Algébrico-Diferenciais”**, Dissertação de Mestrado, PEQ-COPPE-UFRJ, Rio de Janeiro, RJ, Brasil.
- [35] VIERIA, R. C., 2001, **“Técnicas de Inicialização de Sistemas Algébrico-Diferenciais”**, Tese de Doutorado, PEQ-COPPE-UFRJ, Rio de Janeiro, RJ, Brasil.
- [36] WILLIAMS, R., CAMERON, I., BURRAGE, K., 2000, **“A New Index 2 Runge-Kutta method for the Simulation of Batch and Discontinuous Processes”**, Computers and Chemical Engineering, v. 24 pp. 625-630.