**Análise e Implementação de uma Ferramenta
para a Solução Numérica de Equações
Algébrico-Diferenciais com
Singularidades**

Renato César da Silva

Orientador: Prof. Dr. Antonio Castelo Filho

Dissertação de Mestrado apresentada ao Instituto de Ciências Matemáticas de São Carlos, da
Universidade de São Paulo, como parte dos requisitos para a obtenção do título de Mestre em Ciências -
Área : "*Ciências de Computação e Matemática Computacional*".

São Carlos - Junho de 1996

A Adriana

À minha família

Agradecimentos

Ao Professor, **Antonio Castelo Filho**, pela orientação, dedicação e incentivo indispensáveis para a realização deste trabalho.

Aos professores do Instituto de Ciências Matemáticas de São Carlos, da Universidade de São Paulo, que muito me ajudaram.

Às secretárias da pós-graduação, Beth e Laura, pela simpatia, atenção e eficiência com que sempre me atenderam.

Aos amigos que direta e indiretamente contribuíram com seu apoio durante este período de estudos em São Carlos.

Ao **Cnpq** pelo apoio financeiro fornecido durante todo o desenvolvimento deste trabalho.

Análise e Implementação de uma Ferramenta para a Solução Numérica de Equações Algébrico Diferenciais com Singularidades

Renato César da Silva

Orientador: Prof.Dr.Antonio Castelo Filho

RESUMO

Este trabalho diz respeito à solução numérica de equação algébrico-diferencial (E.A.D.) com singularidades. Inicialmente realizamos um estudo sobre a teoria geral de E.A.D. e dos métodos numéricos a ela aplicado. Uma nova classe de métodos para E.A.D. com singularidades, desenvolvida em [Castelo-Tavares-94], é introduzida. Na implementação dessa nova classe de métodos, utilizamos o método BDF com ordem e amplitude do passo variáveis. A utilização do método BDF, bem como as técnicas de seleção de ordem e amplitude do passo foram baseadas nas utilizadas no código DASSL [Brenan-89]. Finalizamos o trabalho com alguns exemplos numéricos que ilustram o funcionamento dos métodos considerados.

Analysis and Implementation of a Appliance for a Numerical Solution of Differential-Algebraic Equations with Singularities

Renato César da Silva

Adviser: Prof.Dr.Antonio Castelo Filho

ABSTRACT

This work is concerned with the numerical solution of differential-algebraic equation (D.A.E.) with singularities. Initially we made a study about the general theory of D.A.E. and numericals methods applicaded itself. A new class of methods for D.A.E. with singularities, developed in [Castelo-Tavares-94], was introduced. In the implementation this new class of methods, we used Backward Differentiation Formulas (BDF) methods with variable order and stepsize. The BDF methods, well as the techniques of order and stepsize selection are based on the DASSL code [Brenan-89]. We end the work with some numericals examples which illustrate the methods considered.

CONTEÚDO

INTRODUÇÃO	1
1 CONCEITOS BÁSICOS	3
2 TEORIA DAS E.A.D.	8
2.1 Introdução	8
2.2 Solubilidade e Índice	8
2.3 E.A.D. Lineares com Coeficientes Constantes	9
2.4 E.A.D. Lineares com Coeficientes Variando com o Tempo	12
2.4.1 Solubilidade e Índice	12
2.4.2 Formas Estruturais	12
2.5 E.A.D. Não Lineares	15
2.5.1 Solubilidade e Índice	15
2.5.2 Formas Estruturais	16
3 MÉTODOS NUMÉRICOS PARA E.A.D.	18
3.1 Introdução	18
3.2 Métodos One-Leg e BDF	18
3.3 Convergência do Método BDF	20
3.3.1 Sistemas Lineares com Coeficientes Constantes	20
3.3.2 Sistemas Semi Explícitos	20
3.3.3 Sistemas Completamente Implícitos	21

4 E.A.D. com SINGULARIDADES	23
4.1 Introdução	23
4.2 Conceitos Básicos	23
4.3 Teoremas de Existência e Unicidade	29
4.4 Singularidades	30
4.5 Equações com posto menor do que m	30
5 MÉTODO NUMÉRICO para E.A.D. com SINGULARIDADE	36
5.1 Introdução	36
5.2 Descrição do Método	37
5.3 Tentativas de Redução da Ordem do Sistema a Ser Resolvido pelo Método de Newton	42
5.4 Conclusões	46
6 ALGORÍTMOS e ESTRATÉGIAS no GSDAE	48
6.1 Introdução	48
6.2 Fórmulas Básicas	48
6.2.1 Coeficientes	49
6.2.2 Polinômio Preditor	51
6.2.3 Iteração Corretora	51
6.2.3.1 Fórmula Corretora	51
6.2.3.2 Solução do Sistema Não Linear	52
6.2.3.3 Algoritmo para a Iteração Corretora	55
6.3 Estratégias para Seleção de Ordem e Amplitude do Passo	58
6.3.1 Teste para a Aceitação do Passo	58
6.3.2 Critério para Determinar a Ordem do Próximo Passo	60
Algoritmo para a aceitação do passo e para determinar a ordem a ser utilizada no próximo passo	61
6.3.3 Critério para Determinar a Amplitude para o Próximo Passo	62
Algoritmo para determinar a amplitude do passo	63
6.4 Alguns Comentários de Implementação	64
6.4.1 Norma Peso	64

6.4.2 UP DATE das Diferenças Divididas	65
6.5 Processo de Inicialização	66
7 RESULTADOS NUMÉRICOS	68
CONCLUSÃO	89
BIBLIOGRAFIA	92

INTRODUÇÃO

Uma Equação Algébrico-Diferencial (E.A.D.) é um sistema que envolve equações algébricas e/ou diferenciais. A E.A.D. surge frequentemente em problemas de diversas áreas de estudo, dentre as quais destacamos

- Equações do Movimento Descrito por Pêndulos ;
- Simulação de Trabalhos Elétricos ;
- Dinâmica de Fluídos ;
- Problemas em Robótica ;
- Engenharia de Controle ;
- Sistemas Mecânicos Restritos e outros.

O interesse de cientistas e engenheiros de diversas áreas em trabalhar diretamente com E.A.D. tem se mostrado crescente nos últimos 25 anos. Dependendo da área a E.A.D. é denominada singular, implícita, algébrico-diferencial, não canônica, não casual, degenerada, não natural, modelo de ordem reduzida, sistemas fora do padrão.

Nos anos 60 e 70 havia um estudo analítico da teoria de sistemas lineares com coeficientes constantes e alguns sistemas não lineares [Luenberger-77] e [Silverman-69].

O primeiro método numérico prático para certas classes de E.A.D. foi dado pelas fórmulas de diferenciação para trás (Backward Differentiation Formulas) BDF de [Gear-71]. No fim dos anos 70, houve um ressurgimento do interesse em E.A.D. motivado pelos vários problemas cujo modelo matemático resultam em E.A.D. . Novos e mais

robustos códigos, tais como DASSL [Petzold-82], LSOLDI [Hindmarsh-80] e RADAU [Hairer-91] foram desenvolvidos para a solução de E.A.D. O entendimento teórico de quando esperar o bom funcionamento de métodos numéricos para E.A.D. ou o que fazer em casos particulares, tem sido aperfeiçoado.

Faremos agora uma breve descrição deste nosso trabalho:

- No capítulo 1, introduzimos algumas definições e resultados que julgamos necessários para o bom entendimento deste.
- No capítulo 2 apresentamos um resumo da teoria geral sobre Equações Algébrico Diferenciais ;
- No capítulo 3, há uma breve discussão dos métodos numéricos para E.A.D. de índices 0,1 e 2 estes métodos fornecem uma aproximação para a solução clássica da E.A.D. (As definições de solução clássica e índice de uma E.A.D. encontram-se no capítulo 2);
- No capítulo 4, apresentamos um resumo da teoria das Equações Algébrico-Diferenciais com Singularidades, este estudo se faz necessário pois os métodos apresentados no capítulo 3 não apresentam um bom comportamento em presença de singularidades. E este fato nos serve também como motivação para apresentar no capítulo 5 uma nova abordagem numérica para a solução de uma E.A.D. de índice 0,1 e algumas de índice 2, com base na teoria apresentada em [Castelo-94], onde nesta nova abordagem obtemos uma aproximação numérica não para a solução clássica mas sim para uma solução geral da E.A.D. (A definição de solução geral encontra-se no capítulo 4);
- No capítulo 6; descrevemos as estratégias de implementação do método apresentado no capítulo 5. Procedimentos para a predição, bem como da correção da solução são discutidos em detalhes além do critério para aceitação, aumento/diminuição da ordem e seleção da amplitude de um passo são apresentados. Todas estas técnicas foram baseadas no código DASSL [Brenan-89].

CAPÍTULO 1

CONCEITOS BÁSICOS

1.1 Introdução

Neste capítulo listamos algumas definições e resultados que serão úteis para o bom entendimento deste trabalho.

Definição 1.1: Um Homeomorfismo entre X e Y é uma aplicação $H: X \rightarrow Y$ contínua, inversível com inversa contínua. Se H e H^{-1} são de classe C^k , dizemos que H é um Homeomorfismo de classe C^k entre X e Y . Um Homeomorfismo de classe C^k , com $k \geq 1$ é dito um Difeomorfismo de classe C^k , onde C^j é o espaço das funções j -vezes diferenciáveis com a j -ésima derivada contínua.

Definição 1.2: $M \subset R^k$ é uma Variedade de classe C^r de dimensão n se para todo $x \in M$, existem vizinhanças abertas $U \subset R^k$ de x , $V \subset R^n$ e um Homeomorfismo de classe C^r , $H: V \rightarrow U \cap M$.

Definição 1.3 O método linear de passo múltiplo para uma equação diferencial ordinária é definido como:

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \sum_{j=0}^k \beta_j f_{n+j}, \text{ onde } \alpha_j \text{ e } \beta_j \text{ são constantes tal que } \alpha_k \neq 0,$$

α_0 e β_0 não são nulos simultaneamente [Lambert-73].

Definição 1.4: Dado o método de passo múltiplo para uma E.D.O., denominamos como 1º polinômio característico:

$$\rho(\xi) = \sum_{j=0}^k \alpha_j \xi^j$$

e como 2º polinômio característico:

$$\sigma(\xi) = \sum_{j=0}^k \beta_j \xi^j.$$

Definições 1.5 :

a) Polinômio de estabilidade é :

$$\pi(r, \bar{h}) = \rho(r) - \bar{h} \sigma(r) \quad \text{onde } \bar{h} = h\lambda$$

b) O Método de Passo Múltiplo é dito absolutamente estável se as raízes do polinômio de estabilidade possuírem valor em módulo estritamente menor que a unidade.

c) Seja $r_j(\bar{h})$ as raízes de $\pi(r, \bar{h}) = \rho(r) - \bar{h} \sigma(r)$.

$$\text{O conjunto } W = \left\{ \bar{h} \in \mathbb{C}: \begin{array}{l} \text{todas as raízes de } \pi(r, \bar{h}) \text{ satisfazem} \\ |r_j(\bar{h})| \leq 1, \text{ raízes múltiplas satisfazem } |r_j(\bar{h})| < 1 \end{array} \right\} \text{ é}$$

chamado região de estabilidade absoluta do método de passo múltiplo.

Definição 1.6: O método de passo múltiplo é dito estável, se as raízes do 1º polinômio característico não tem módulo maior do que 1 e as de módulo 1 são simples.

Observação: Com a finalidade de distinguir este conceito de estabilidade dos outros, é às vezes denominado zero-estável ou em homenagem à Dahlquist, D-estabilidade.

Teorema 1.7: O método BDF de k passos é estável para $k \leq 6$ e instável para $k \geq 7$ [Hairer-93].

Definição 1.8: $f(x, h)$ é $O(h^p)$ se existe $K > 0$ tal que $\|f(x, h)\| \leq Kh^p$ para h suficientemente pequeno.

Definição 1.9: Seja A uma matriz $N \times (N+1)$ com posto máximo. Então a inversa de Moore-Penrose de A é definida por $A^+ = A^t (AA^t)^{-1}$.

Lema 1.10: Seja A uma matriz $N \times (N+1)$ com posto máximo então

- i) A^+A é a projeção ortogonal de $t(A)^\perp$ sobre R^{N+1} onde $t(A)$ é o núcleo de A^t isto é, $A^+A = Id - t(A)(t(A))^t$;*
- ii) $A^+A = Id$.*

Lema 1.11: Sejam B, E matrizes $N \times (N+1)$ tais que B tenha posto máximo e $\|B^+\| \|E\| \leq 1$.

Então $B-E$ tem posto máximo, e

$$\|(B-E)^+\| \leq \frac{\|B^+\|}{1 - \|B^+\| \|E\|} , \quad [Allgower-90]$$

Definição 1.12: Seja $y' = Ay + b(t)$. Este sistema é Stiff se :

- i) $\text{Re}(\lambda_t) < 0$, $t = 1(1)m$;*
- ii) $\max |\text{Re}(\lambda_t)| \gg \min |\text{Re}(\lambda_t)|$.*

Denominamos raio de Stifficidade como sendo o número:

$$\frac{\max |\operatorname{Re}(\lambda_t)|}{\min |\operatorname{Re}(\lambda_t)|}, \quad \text{onde } \lambda_t, \quad t=1(1)m \text{ são os auto valores da matriz } A.$$

Definições 1.13 [Lambert-73]:

a) O método é A -estável se a região de estabilidade absoluta $W \supset \{z \in \mathbb{C}; \operatorname{Re}(z) < 0\}$.

b) O método é $A(\alpha)$ -estável se $W \supset W_\alpha$ onde $W_\alpha = \{z \in \mathbb{C}; -\alpha < \pi - \arg z < \alpha\}$.

c) O método é Stiff estável se:

i) $\mathbb{W} \supset R_1, \mathbb{W} \supset R_2$ onde $R_1 = \{z: \operatorname{Re}(z) < -a\}$,

$$R_2 = \{z: -a \leq \operatorname{Re}(z) \leq b; -c \leq \operatorname{Im}(z) \leq c\}$$

ii) O método é preciso para todo $h \in R_2$ quando aplicado para a equação teste $y' = \lambda y$, λ uma constante complexa com $\operatorname{Re}(\lambda) < 0$.

CAPÍTULO 2

TEORIA DAS E.A.D.

2.1 Introdução

Neste capítulo apresentaremos uma teoria geral sobre as equações algébrico-diferenciais (E.A.D.); onde veremos:

- Algumas classes de E.A.D. incluindo o caso de E.A.D. lineares com coeficientes constantes, E.A.D. lineares com coeficientes variando com o tempo e E.A.D. não lineares;
- As definições de solubilidade e índice em cada um dos casos.

Esta teoria encontra-se em [BRENAM-89].

2.2 Solubilidade e Índice

Definição 2.1 A função $y(t)$ é uma solução (ou solução clássica) da E.A.D $F(t, y, y') = 0$ (2.1)
em um intervalo I se y é continuamente diferenciável em I e satisfaz (2.1) para todo $t \in I$.

Definição 2.2 Seja I um intervalo aberto de R , Ω um conjunto conexo aberto de R^{2m+1} e F uma função diferenciável de Ω em R^m . Então a E.A.D. (2.1) é solúvel em I sobre Ω se existe uma família r -dimensional de soluções $\Phi(t, c)$ definidas em um conjunto conexo aberto $I \times \Gamma, \Gamma \subset R^r$ tal que :

1. $\Phi(t, c)$ está definida em I para cada $c \in \Gamma$;
2. $(t, \Phi(t, c), \Phi'(t, c)) \in \Omega$ para $(t, c) \in I \times \Gamma$;
3. Se $\Psi(t, c)$ é qualquer outra solução e $(t, \Psi(t, c), \Psi'(t, c)) \in \Omega$, então $\Psi(t, c) = \Phi(t, c)$ para algum $c \in \Gamma$;

4.O gráfico de Φ como uma função de (t,c) é uma variedade $(r+1)$ -dimensional [Def-1.1/1.2].

Definição 2.3 O índice de uma E.A.D. é denotado por v , onde v é o menor número de vezes que a equação (2.1), ou parte dela, deve ser diferenciada em relação a t para determinar y' como uma função contínua de y e t . Este índice também é chamado de índice global.

Exemplo 2.1 Como exemplo considere a equação de um pêndulo em coordenadas cartesianas. O sistema é escrito como:

$$\begin{cases} x'' = -\lambda x & \text{(i)} \\ y'' = -\lambda y - g & \text{(ii)} \\ 0 = x^2 + y^2 - L^2 & \text{(iii)} \end{cases} \quad \text{onde } g \text{ é}$$

a força da gravidade, L é o tamanho da barra do pêndulo e λ é a tensão da barra. Assumindo $g = 1$, $L = 1$, vamos obter o índice deste sistema. Derivando (iii), temos: $xx' + yy' = 0$ (iv). Derivando (iv) obtemos: $xx'' + yy'' + x'^2 + y'^2 = 0 \Rightarrow$ por (i) e (ii) temos: $-\lambda - y + x'^2 + y'^2 = 0$ (v). Derivando (v) e novamente utilizando as relações (i) e (ii) temos: $\lambda' + 3y' = 0$ (vi).

Portanto a E.A.D. acima tem índice três.

2.3 E.A.D. Lineares com Coeficientes Constantes

Consideremos a E.A.D. linear com coeficientes constantes:

$$Ax' + Bx = f, \quad (2.2)$$

onde A e B são matrizes $m \times m$.

Definição 2.4 Se λ é um parâmetro complexo, então $(\lambda A + B)$ é a matriz pencil.

Definição 2.5 Se $\det(\lambda A+B)$ não é identicamente nulo como uma função de λ , então a matriz pencil (ou simplesmente pencil) é dita regular.

Quando o pencil é regular dizemos que a E.A.D.(2.2) é regular.

Definição 2.6 Uma matriz N tem índice de nilpotência k se $N^k = 0$ e $N^{k-1} \neq 0$.

Definição 2.7 O pencil $(\lambda A+B)$ é um pencil nilpotente quando $\det(\lambda A+B)$ é constante.

Teorema 2.8 Suponhamos que $(\lambda A+B)$ é um pencil regular. Então existem matrizes não singulares P e Q tais que :

$$PAQ = \begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix}, \quad PBQ = \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} \quad (2.3)$$

onde N é uma matriz nilpotente com índice k e I é a matriz identidade.

No caso especial em que A é não singular, $PAQ = I$ e $PBQ = C$. Se $\det(\lambda A + B)$ é constante (nilpotente), então (2.3) reduz a $PAQ = N$ e $PBQ = I$.

Aplicando-se as mudanças de coordenadas P e Q proporcionadas por este teorema à E.A.D. (2.2) e considerando $x = Qy$, temos que (2.2) torna-se:

$$AQy' + BQy = f.$$

Multiplicando à esquerda por P , obtemos :

$$PAQy' + PBQy = Pf \quad (2.4)$$

logo pelo teorema 2.8, obtemos, com $Pf = [f_1, f_2]'$,

$$\begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} y_1' \\ y_2' \end{bmatrix} + \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

ou seja,

$$\begin{cases} y_1' + Cy_1 = f_1 \end{cases} \quad (2.5a)$$

$$\begin{cases} Ny_2' + y_2 = f_2 \end{cases} \quad (2.5b)$$

A equação (2.5a) é uma E.D.O. cuja solução existe para qualquer valor inicial de y_1 e qualquer função contínua f_1 .

Quanto a (2.5b), esta tem apenas uma solução pois ;

$$Ny_2' + y_2 = (ND + I)y_2 = f_2,$$

e assim,

$$y_2 = (ND + I)^{-1} f_2, \text{ onde } D = d/dt.$$

Como

$$(ND + I)^{-1} = (I - (-ND))^{-1} = \sum_{i=0}^{\infty} (-ND)^i = \sum_{i=0}^{k-1} (-ND)^i = \sum_{i=0}^{k-1} (-1)^i N^i D^i,$$

onde k é o índice de nilpotência de N e assim $(ND)^i = 0$ para $i \geq k$.

Logo,

$$y_2 = (ND + I)^{-1} f_2 = \sum_{i=0}^{k-1} (-1)^i N^i f_2^i \quad (2.6)$$

onde

$$f_2^i = \frac{d^i f_2}{dt^i}, \text{ para } f_2 \text{ de classe } C^{(k-1)}.$$

Teorema 2.9 A E.A.D. linear com coeficientes constantes (2.2) é solúvel se, e somente se $(\lambda A + B)$ é um pencil regular.

Teorema 2.10 Suponhamos que a E.A.D. linear com coeficientes constantes $Ax' + Bx = f$ seja solúvel e de índice $v \geq 1$. Então, cada iteração da resolução para a restrição explícita e diferenciação destas restrições reduz o índice por 1, tal que após v diferenciações a E.A.D. será reduzida a uma E.D.O..

2.4 E.A.D. Lineares com Coeficientes Variando com o Tempo

A equação geral de uma E.A.D. linear com coeficientes variando com o tempo têm a seguinte forma:

$$A(t)x'(t) + B(t)x(t) = f(t) \quad (2.7)$$

e está definida no intervalo I .

2.4.1 Solubilidade e Índice

Definição 2.11 O sistema (2.7) com matrizes $A(t)$ e $B(t)$, $m \times m$ é solúvel no intervalo I se para toda f m -vezes diferenciável, existe pelo menos uma solução para (2.7) continuamente diferenciável.

E além disso, as soluções estão definidas em todo o intervalo I e são determinadas unicamente por seus valores iniciais para qualquer $t \in I$.

Definição 2.12 Suponhamos que (2.7) é solúvel em I . O índice de (2.7) é o número mínimo de diferenciações necessárias para determinar unicamente x' como uma função de x e t .

Definição 2.13 Se (2.7) é uma E.A.D. regular, então o índice local para t , denotado por $\nu_1(t)$, é o índice do pencil $(\lambda A(t) + B(t))$.

2.4.2 Formas Estruturais

Considerando a mudança de coordenada em (2.7) dada por $x = Q(t)y$ e também multiplicando por $P(t)$, obtemos:

$$P(t)A(t)Q(t)y' + (P(t)A(t)Q'(t) + P(t)B(t)Q(t))y = P(t)f \quad (2.8)$$

onde $P(t)$ e $Q(t)$ são não singulares no intervalo de interesse e pelo menos tão suave quanto os coeficientes da E.A.D..

Teorema 2.14 *Suponhamos que (2.7) é uma E.A.D. solúvel em I . Então as seguintes afirmações são equivalentes :*

1. (2.7) é uma E.D.O. implícita ;
2. $v=0$;
3. $v_l \equiv 0$
4. $v_l(t_0)=0$ para algum $t_0 \in I$, onde v é o índice e v_l é o índice local de (2.7).

Teorema 2.15 *Suponhamos que (2.7) é solúvel, então $v=1$ se, e somente se $v_l \equiv 1$.*

Observação: Estes resultados mostram que v e v_l são os mesmos para sistemas solúveis de índice zero ou um. Isto não ocorre em sistemas de índice maior ou igual a dois [CAMPBELL-82].

Teorema 2.16 *Suponhamos que (2.7) é solúvel em I . As seguintes afirmações são preservadas quando transformamos (2.7) num sistema analiticamente equivalente a (2.8):*

1. $v_l \equiv 0$;
2. $v_l \equiv 1$;
3. $v_l(t) \geq 2$ para todo $t \in I$.

Dessa forma , podemos considerar três classes de E.A.D. solúveis:

- E.D.O implícitas;
- índice um;
- índice ≥ 2 .

Definição 2.17 *O sistema (2.7) é semi explícito se estiver na forma :*

$$\begin{cases} \dot{x}_1 + B_{11}(t)x_1 + B_{12}(t)x_2 = f_1 \\ B_{21}(t)x_1 + B_{22}(t)x_2 = f_2 \end{cases} \quad (2.9)$$

Observação: A E.A.D. semi explícita (2.9) tem índice um se, e somente se B_{22} é não singular para todo t .

Definição 2.18 A E.A.D. (2.7) está na forma de Hessenberg com dimensão r se pode ser escrita como :

$$\begin{bmatrix} I & 0 & . & . & 0 \\ 0 & I & . & . & . \\ . & . & I & . & . \\ . & . & . & . & . \\ 0 & . & . & . & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ . \\ . \\ \dot{x}_r \end{bmatrix} + \begin{bmatrix} B_{11} & . & . & B_{1r-1} & B_{1r} \\ B_{21} & . & . & B_{2r-1} & 0 \\ 0 & . & . & . & . \\ . & . & . & . & . \\ 0 & . & . & B_{r-1} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_r \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ . \\ . \\ f_r \end{bmatrix}$$

onde x_i são vetores, B_{ij} são matrizes e o produto $B_{r-1}B_{r-2}\dots B_{21}B_{1r}$ é não singular.

Observação: Toda E.A.D. na forma de Hessenberg com dimensão r é solúvel e tem índice r .

Definição 2.19 Dizemos que o sistema (2.7) está na forma canônica padrão se pode ser escrito como:

$$\begin{bmatrix} I & 0 \\ 0 & N(t) \end{bmatrix} \dot{x} + \begin{bmatrix} C(t) & 0 \\ 0 & I \end{bmatrix} x = f(t), \quad (2.10)$$

onde N é estritamente triangular inferior (ou superior).

A definição acima não exige que a matriz N tenha posto ou índice constante.

Definição 2.20 Uma propriedade de uma matriz é uma propriedade estrutural se todo elemento não nulo da matriz pode ser trocado por variáveis independentes e a propriedade ainda vale em um conjunto aberto de valores destas variáveis.

Teorema 2.21 Suponhamos que (2.7) seja solúvel em I , então (2.7) é analiticamente equivalente a :

$$\begin{bmatrix} I & G \\ 0 & N \end{bmatrix} Z' + \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} Z = \begin{bmatrix} g \\ h \end{bmatrix} \quad (2.11)$$

onde $Nz'_2 + z_2 = h$ tem solução única para cada função h .

Além disso, existe uma família contável de intervalos abertos disjuntos I_i tal que $\bigcup I_i$ é denso em I . O sistema $Nz'_2 + z_2 = h$ é analiticamente equivalente a algum sistema na forma canônica padrão $Mw' + w = f$, onde M é estruturalmente nilpotente.

Definição 2.22 A E.A.D. é uma cadeia triangular de r -ésima ordem se tem a forma

$$\begin{bmatrix} A_{11} & 0 & . & . & 0 \\ A_{21} & A_{22} & . & . & . \\ A_{31} & A_{32} & A_{33} & . & . \\ . & . & . & . & . \\ A_{r1} & . & . & . & A_{rr} \end{bmatrix} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ . \\ x'_r \end{bmatrix} + \begin{bmatrix} B_{11} & 0 & . & . & 0 \\ B_{21} & B_{22} & . & . & . \\ B_{31} & . & . & . & . \\ . & . & . & . & . \\ B_{r1} & . & . & . & B_{rr} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ x_r \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ . \\ f_r \end{bmatrix}$$

e cada sub sistema $A_{ii}x'_i + B_{ii}x_i = f_i - \sum_{j=1}^{i-1} (A_{ij}x'_j + B_{ij}x_j)$ é de índice zero, ou de índice um, ou tem a forma de Hessenberg ou está na forma canônica padrão.

Qualquer sistema que é analiticamente equivalente a uma cadeia triangular é solúvel.

2.5 E.A.D. não Lineares

Muitas das E.A.D. importantes são não lineares. A não linearidade terá apenas resultados locais. Também será mais difícil estabelecermos condições necessárias e suficientes, usualmente, teremos que nos satisfazer apenas com condições suficientes.

2.5.1 Solubilidade e Índice

Consideraremos a E.A.D. não linear (2.1) e o seguinte sistema de equações:

$$\left\{ \begin{array}{l} F(t, y, y') = 0 \\ \frac{d}{dt} F(t, y, y') = 0 \\ \vdots \\ \frac{d^{j-i}}{dt^{j-i}} F(t, y, y') = 0 \end{array} \right. \quad (2.12)$$

o qual pode ser escrito como:

$$\left\{ \begin{array}{l} F_{[0]}(t, y, y') = 0 \\ F_{[1]}(t, y, y', y'') = 0 \\ \vdots \\ F_{[j-1]}(t, y, y', y'', \dots, y^{(j)}) = 0 \end{array} \right. \quad (2.13)$$

onde, por exemplo, $F_{[1]} = F_{y'} y'' + F_y y' + F_t$. Podemos também, escrever (2.13), na forma:

$$F_j(t, y, y_j) = 0, \quad (2.14)$$

onde $y_j = [y', \dots, y^{(j)}]^t$.

Definição 2.23 O índice v de (2.1) é o menor número v tal que F_{v+1} determina apenas a variável y' como uma função contínua de y e t .

Definição 2.24 O índice local v_l de (2.1) para $(\tilde{t}, \tilde{y}, \tilde{y}')$ é o índice do pencil $\lambda F_{y'}(\tilde{t}, \tilde{y}, \tilde{y}') + F_y(\tilde{t}, \tilde{y}, \tilde{y}')$.

Proposição 2.25 Suponhamos que $F_{y'}$ tenha posto constante. Então $v=1$ se, e somente se $v_l \equiv 1$.

2.5.2 Formas Estruturais

Definição 2.26 A E.A.D. (2.12) possui a forma de Hessenberg de dimensão r se pode ser escrita como:

$$\begin{cases} x_1' = F_1(x_1, x_2, \dots, x_r, t) \\ x_2' = F_2(x_1, x_2, \dots, x_{r-1}, t) \\ \vdots \\ x_i' = F_i(x_{i-1}, x_i, \dots, x_{r-1}, t) \\ \vdots \\ 0 = F_r(x_{r-i}, t) \end{cases}$$

, $3 \leq i \leq r-1$ e $(\partial F_r / \partial x_{r-1})(\partial F_{r-1} / \partial x_{r-2}) \dots (\partial F_2 / \partial x_1)(\partial F_1 / \partial x_r)$ é não singular.

Assumindo que os F_i são suficientemente diferenciáveis, a forma de Hessenberg de dimensão r tem índice e índice local r .

Definição 2.27 A E.A.D. (2.1) está na forma canônica padrão de dimensão r se pode ser escrita como:

$$\begin{cases} x_1' = F_1(x_1, t) \\ N(x_1, x_2, t)x_2' = x_2 + h(x_1, t) \end{cases}$$

onde N é uma matriz bloco rxr estritamente triangular inferior.

Definição 2.28 A E.A.D. (2.1) é uma cadeia triangular inferior de ordem r se tem a forma :

$$\begin{cases} F_1(x_1', x_1, t) = 0 \\ F_2(x_1', x_2', x_1, x_2, t) = 0 \\ \vdots \\ F_r(x_1', \dots, x_r', x_1, \dots, x_r, t) = 0 \end{cases}$$

e o sistema definido pelas funções F_i é uma E.D.O. implícita, ou tem índice um, ou está na forma Hessenberg, ou na forma canônica padrão em relação a variável x_i .

Uma cadeia triangular é sempre solúvel.

CAPÍTULO 3

MÉTODOS NUMÉRICOS PARA E.A.D.

3.1 Introdução :

Neste capítulo apresentaremos alguns métodos clássicos para resolução numérica de uma E.A.D.. Em particular estudaremos as propriedades de convergência e ordem do método BDF aplicado em sistemas de Equações Algébrico-Diferenciais na forma

$$F(t, y, y') = 0 \quad (3.1)$$

onde F e y tem a mesma dimensão.

Assumiremos que F é continuamente diferenciável .

3.2 Métodos One-Leg e BDF

Trataremos em linhas gerais sobre as aplicações de métodos one-leg e BDF para sistemas de E.A.D. de índice um.

Para a E.D.O. não linear:

$$y' = g(t, y) \quad (3.2)$$

o método de passo múltiplo linear tem a forma:

$$\sum_{j=0}^k \alpha_j y_{n-j} - h \sum_{j=0}^k \beta_j g(t_{n-j}, y_{n-j}) = 0 \quad (3.3)$$

O método one-leg para (3.2) envolve apenas uma avaliação de g e tem a forma:

$$\frac{1}{h} \sum_{j=0}^k \alpha_j y_{n-j} - g\left(\sum_{j=0}^k \beta_j t_{n-j}, \sum_{j=0}^k \beta_j y_{n-j}\right) = 0 \quad (3.4)$$

O método BDF pode ser considerado como sendo um método de passo múltiplo linear ou um método one-leg, basta tomar nas equações (3.3) e (3.4), $\beta_0 = 1$ e $\beta_j = 0$, $j=1(1)k$.

A primeira técnica geral bem sucedida para a solução numérica de E.A.D., foi proposta por [Gear-71a/b]. Esta técnica, que utilizava os métodos BDF, foi inicialmente aplicada a sistemas de equações diferenciais acopladas a equações algébricas:

$$\begin{cases} y' = f(t, x, y) \\ 0 = g(t, x, y) \end{cases}, \quad (3.5)$$

onde y e g tem a mesma dimensão.

O método BDF trata as variáveis algébricas e diferenciais do problema (3.5) de maneira análoga. Assim este método pode ser estendido para a generalização de qualquer sistema completamente implícito (3.1).

O método BDF mais simples é o método de Euler implícito, que consiste em substituir a derivada em (3.1) por uma diferença para trás

$$F(t_n, y_n, \frac{y_n - y_{n-1}}{h}) = 0 \quad (3.6)$$

onde $h = t_i - t_{i-1}$, $i=1(1)n$. O sistema de equações não lineares resultante para y_n em cada passo é usualmente resolvido pelo método de Newton.

O método BDF de k passos consiste em substituir y' pela derivada do polinômio que interpola a solução sobre os pontos $t_n, t_{n-1}, \dots, t_{n-k}$ avaliados em t_n . Isto é,

$$F(t_n, y_n, \frac{\rho y_n}{h}) = 0, \quad (3.7)$$

onde $\rho y_n = \sum_{i=0}^k \alpha_i y_{n-i}$ e α_i , $i=0(1)k$ são os coeficientes do método BDF.

Observação: Da teoria de EDO, sabemos que um método BDF de k passos é estável para $k \leq 6$ e instável para $k \geq 7$, [Ver capítulo 1 definições 1.3 - 1.6 / teorema 1.1].

3.3 Convergência do Método BDF

Apresentaremos alguns resultados sobre a convergência do método BDF aplicado em sistemas de E.A.D. de índice um.

3.3.1 Sistemas Lineares com Coeficientes Constantes

Teorema 3.1 *O método BDF de k passos, com $k < 7$ e h fixo, aplicado ao sistema de E.A.D linear com coeficientes constantes e índice v é convergente de ordem $O(h^k)$ [Ver capítulo 1 definição 1.7] depois de $(v-1)k+1$ passos.*

O resultado acima nos dá a convergência do método BDF aplicado ao sistema de E.A.D. linear com coeficientes constantes cuja demonstração encontra-se em [Brenan-89].

3.3.2 Sistemas Semi Explícitos

Sistemas semi explícitos de índice um são subconjuntos importantes do sistema de índice um, que surge frequentemente em problemas de Ciências e Engenharia. Estes sistemas são escritos na forma:

$$\begin{cases} y' = f(t, x, y) \\ 0 = g(t, x, y) \end{cases}, \quad (3.8)$$

onde $g_y(t, x, y)$ é inversível na vizinhança da solução de (3.8).

Definição 3.2 *Um método de passo múltiplo linear aplicado a (3.8) é dado por:*

$$\sum_{j=0}^k \alpha_j y_{n-j} = h \sum_{j=0}^k \beta_j f(t_{n-j}, x_{n-j}, y_{n-j}) \quad (3.9a)$$

$$0 = g(t_n, x_n, y_n) \quad (3.9b)$$

Os métodos de passos múltiplos aplicados às E.A.D semi explícitas de índice um são estáveis e convergente para a E.A.D. de mesma ordem. Este resultado segue do teorema da função implícita. Especificamente, g pode ser resolvida para y_n em termos de x_n , isto é, $y_n = G(t_n, x_n)$, então y_n pode ser inserida na fórmula de passo múltiplo (3.9a) para produzir a mesma equação de diferença da obtida quando aplicamos o método linear de passo múltiplo na E.D.O.

$$y' = f(t, x, G(x, t)) \quad (3.10)$$

Em princípio a E.A.D. semi explícita pode ser resolvida por qualquer método de passo múltiplo linear que seja apropriado para a E.D.O. (3.10).

3.3.3 Sistemas Completamente Implícitos

Sistemas completamente implícitos de índice um, têm a forma:

$$\begin{cases} F(t, y, y') = 0 \\ y(t_0) = y_0 \end{cases} \quad (3.11)$$

Assumiremos que F é uma função suficientemente suave e que, existe uma solução suave $y(t)$ satisfazendo os valores iniciais dados.

Definição 3.3 A E.A.D. não linear (3.11) é de índice uniforme um se o índice do sistema de coeficientes constantes $Aw'(t) + Bw(t) = g(t)$, onde $A = F_{y'}(\tilde{t}, \tilde{y}, \tilde{y}')$ e $B = F_y(\tilde{t}, \tilde{y}, \tilde{y}')$, é um para todo $(\tilde{t}, \tilde{y}, \tilde{y}')$ na vizinhança do gráfico da solução e, além disso, se:

1. as derivadas parciais de A com respeito a t, y, y' existem e são limitadas na vizinhança da solução;

2. o posto de A é constante na vizinhança da solução.

Teorema 3.4 *Considere (3.11) sendo uma E.A.D. uniforme de índice um no intervalo $I=[t_0, t_0 + T]$. Então a solução numérica de (3.11) dada pelo método BDF de k passos com $k < 7$ e h fixo, converge com $O(h^k)$ de exatidão se todos valores iniciais são $O(h^k)$ e se a iteração de Newton em cada passo é resolvida com $O(h^{k+1})$ de exatidão.*

A demonstração deste teorema encontra-se em [Brenan-89] e mais detalhada em [Pires-94].

CAPÍTULO 4

E.A.D. COM SINGULARIDADES

4.1 Introdução

Neste capítulo trataremos de E.A.D. com singularidades. Convém observarmos que os métodos apresentados no capítulo 3, quando aplicados a uma E.A.D. com singularidade não apresentam um bom comportamento. Desta forma, este capítulo nos serve como motivação para a introdução de uma nova abordagem para a resolução de E.A.D., que em particular, resolve E.A.D. com singularidade satisfatoriamente.

Os resultados aqui apresentados foram obtidos em [Castelo-94].

4.2 Conceitos Básicos

Consideremos a Equação Algébrica Diferencial de ordem q , definida por:

$$F(X, Y(X), Y^{(1)}(X), \dots, Y^{(q)}(X)) = 0 \quad (4.1)$$

onde $F : \Omega \rightarrow \mathbb{R}^m$ é uma função suficientemente diferenciável do conjunto aberto

$$\Omega \subset \mathbb{R}^{(q+1)m+1}.$$

Introduziremos agora o conceito de índice para a E.A.D. (4.1), para tal seja nd : o número mínimo de diferenciações que parte ou todas equações do sistema (4.1) devem ser diferenciadas com relação a X de modo que $Y^{(q)}$ pode ser escrita como uma função contínua de $X, Y, Y', Y'', \dots, Y^{(q-1)}$, istoé, $Y^{(q)} = G(X, Y, Y', Y'', \dots, Y^{(q-1)})$. Assim temos:

Definição 4.1 Dada a E.A.D. (4.1)

Se $q > 0$. O seu índice v é dado por: $v = nd - q + 1$;

Se $q = 0$, ou seja é uma equação puramente algébrica, seu índice é: $v = nd$.

Definição 4.2 O conjunto de restrições diferenciais de (4.1) é o domínio de G , que denotaremos por RD .

Definição 4.3 O conjunto de restrições algébricas, denotado por RA , é o conjunto que satisfaz todas as equações algébricas que aparecem no processo de obtenção do índice de (4.1).

Definição 4.4 Um ponto $(X, Y, Y^{(1)}, Y^{(2)}, \dots, Y^{(q-1)})$ é dito ponto crítico de (4.1) se: $(X, Y, Y^{(1)}, Y^{(2)}, \dots, Y^{(q-1)}) \in RD \cap RA$, $G(X, Y, Y^{(1)}, Y^{(2)}, \dots, Y^{(q-1)}) = 0$ e existe z com $(X, Y, Y^{(1)}, Y^{(2)}, \dots, Y^{(q-1)}, z) \in \Omega$ com $F(X, Y, Y^{(1)}, Y^{(2)}, \dots, Y^{(q-1)}, z) = 0$.

Definição 4.5 Um ponto $(X, Y, Y^{(1)}, Y^{(2)}, \dots, Y^{(q-1)})$ é dito uma singularidade de (4.1) se: $(X, Y, Y^{(1)}, Y^{(2)}, \dots, Y^{(q-1)}) \in \partial RD \cap RA$ e existe z com $(X, Y, Y^{(1)}, Y^{(2)}, \dots, Y^{(q-1)}, z) \in \Omega$ e $F(X, Y, Y^{(1)}, Y^{(2)}, \dots, Y^{(q-1)}, z) = 0$.

Definição 4.6 A imagem de uma função $c: J \rightarrow \mathbb{R}^{(q+1)m+1}$ é uma solução geral da E.A.D. (4.1) se $c \in C^1(J)$ e satisfaz

$$\left\{ \begin{array}{l} F(c(s)) = 0 \\ y_1(s)x'(s) - y_0'(s) = 0 \\ y_2(s)x'(s) - y_1'(s) = 0 \\ \vdots \\ y_q(s)x'(s) - y_{q-1}'(s) = 0 \end{array} \right. \quad (4.2)$$

para todo $s \in J$ onde $c(s) = (x(s), y_0(s), y_1(s), \dots, y_q(s))$ com $x(s) \in \mathbb{R}$, $y_i(s) \in \mathbb{R}^m$, $i = 0(1)q$.

A seguir dois exemplos para elucidar as definições acima.

Exemplo 4.1

Seja a E.A.D. dada por:

$$(*) \quad \begin{cases} y_2^2 + y_1'^2 = 1 & (a) \\ t^2 + y_1^2 = \frac{1}{4} & (b) \end{cases}$$

Derivando (b), temos

$$2t + 2y_1 y_1' = 0 \Rightarrow y_1' = -\frac{t}{y_1} \quad (c).$$

$$RD1 = \{(t, y_1, y_2); y_1 \neq 0\}$$

Substituindo (c) em (a),

$$y_2^2 + \frac{t^2}{y_1^2} = 1 \quad (d)$$

Derivando (d),

$$2y_2 y_2' + \frac{2ty_1^2 - 2t^2 y_1 y_1'}{y_1^4} = 0$$

ou

$$y_2 y_2' + \frac{t y_1 - t^2 y_1'}{y_1^3} = 0 \quad (e)$$

Substituindo (c) em (e)

$$y_2 y_2' + \frac{t}{y_1^2} + \frac{t^3}{y_1^4} = 0 \quad (f)$$

logo,

$$y_2' = -\frac{t}{y_1^2 y_2} \left(1 + \frac{t^2}{y_1^2} \right) \quad (g)$$

$$RD2 = \{(t, y_1, y_2); y_1 \cdot y_2 \neq 0\}$$

Logo o índice ν da equação (*) é $nd - q + 1 = 2 - 1 + 1 = 2$ e $G: RD \rightarrow R^2$ definida por :

$$G_1(t, y_1, y_2) = -\frac{t}{y_1}$$

$$G_2(t, y_1, y_2) = \frac{-t}{y_1^2 y_2} \left(1 + \frac{t^2}{y_1^2} \right)$$

onde $RD = RD1 \cap RD2$.

E o conjunto das restrições algébricas de (*) é definido pelas equações (b) e (d):

$$RA = \left\{ (t, y_1, y_2); t^2 + y_1^2 = \frac{1}{4}, y_2^2 + \frac{t^2}{y_1^2} = 1 \right\}.$$

Exemplo 4.2

Seja a E.A.D. puramente algébrica; $F(t, y) = 0$, onde $F: \Omega \rightarrow R$ é dada por:

$$F(t, y) = y^2 + t^2 - 1 \text{ e } \Omega \subset R^2$$

Seu índice ν é 1, pois diferenciando F com relação a t , apenas uma vez, temos:

$$2yy' + 2t = 0 \quad \text{ou seja,} \quad y' = \frac{-t}{y}.$$

Assim a função $G(t,y)$ da definição 4.1 é dada por:

$$G: RD \rightarrow R$$

$$G(t,y) = \frac{-t}{y}$$

$$\text{onde, } RD = \{(t,y) \in R^2 : y \neq 0\}$$

Os pontos $P_1 = (0,1), P_2 = (0,-1)$ são pontos críticos, $P_3 = (1,0)$ e $P_4 = (-1,0)$ são pontos singulares da E.A.D. $y^2 + t^2 - 1 = 0$, pois $G(P_1) = G(P_2) = 0$ e P_3, P_4 estão no bordo de RD (e satisfazem $y^2 + t^2 - 1 = 0$).

Agora seja $c: J \rightarrow R^2$ definida por:

$$c(s) = (\sin(s), \cos(s)).$$

Temos que $c \in C^1(J)$, onde $J \subset R$. Agora, $F(c(s)) = 0$.

Logo a imagem da função c ($c(J)$) é uma solução geral da E.A.D. $y^2 + t^2 - 1 = 0$.

Teorema 4.7 *Considere uma solução geral de (4.1) dada por $c: J \rightarrow R^{(q+1)m+1}$.*

Se $x'(s) \neq 0$ em um intervalo $I \subset J$, então

$y_0(s) = Y_0(x(s)), y_1(s) = \frac{d}{dx} Y_0(x(s)), y_2(s) = \frac{d^2}{dx^2} Y_0(x(s)), \dots, y_q(s) = \frac{d^q}{dx^q} Y_0(x(s))$ em I e y_0 definida em $T(I)$ é uma solução clássica da E.A.D. (2.1).

O problema de valor inicial para a E.A.D. (4.1) é definido como:

$$\begin{cases} F(X, Y(X), Y'(X), \dots, Y^{(q)}(X)) = 0 \\ Y^{(i)}(x_0) = a_i, i=0(1)q \end{cases} \quad (4.3)$$

onde $F(x_0, a_0, a_1, \dots, a_q) = 0$.

Definição 4.8 Uma w -estrutura para a equação (4.1) é uma função $w: \Omega \rightarrow \mathbb{R}^{q \times ((q+1)m+1)}$, definida por:

$$w(x, y_0, y_1, \dots, y_q) = \begin{pmatrix} w_q(x, y_0, y_1, \dots, y_q) \\ w_{q-1}(x, y_0, y_1, \dots, y_q) \\ \vdots \\ w_2(x, y_0, y_1, \dots, y_q) \\ w_1(x, y_0, y_1, \dots, y_q) \end{pmatrix} =$$

$$= \begin{pmatrix} y_q & 0 & 0 & \dots & 0 & -I_m & 0 \\ y_{q-1} & 0 & 0 & \dots & -I_m & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_2 & 0 & -I_m & \dots & 0 & 0 & 0 \\ y_1 & -I_m & 0 & \dots & 0 & 0 & 0 \end{pmatrix} \text{ com } \begin{cases} y_i \in \mathbb{R}^{m \times 1}, i = 1(1)q \\ I_m \in \mathbb{R}^{m \times m} \\ 0 \in \mathbb{R}^{m \times m} \end{cases}$$

Agora podemos reescrever (4.2) como :

$$\begin{cases} F(c(s)) = 0 \\ w(c(s))c'(s) = 0 \end{cases} \quad (4.4)$$

O sistema (4.4) é chamado: sistema diferencial associado à E.A.D. (4.1).

O problema de valor inicial para (4.4) é definido como:

$$\begin{cases} F(c(s)) = 0 \\ w(c(s))c'(s) = 0 \\ c(0) = c_0 = (x_0, a_0, a_1, \dots, a_q) \end{cases} \quad (4.5)$$

com $F(c_0)=0$.

Diferenciando a primeira equação de (4.4) com respeito a s e definindo:

$$A(c(s)) = \begin{pmatrix} DF(c(s)) \\ w(c(s)) \end{pmatrix},$$

temos um novo problema de valor inicial:

$$\begin{cases} A(c(s))c'(s) = 0 \\ c(0) = c_0 = (t_0, a_0, a_1, \dots, a_q) \end{cases} \quad (4.6)$$

Note que se (4.1) é do tipo $F(x(s), y(s)) = 0$ então $A(c(s)) = DF(c(s))$.

4.3 Teoremas de Existência e Unicidade

Teorema 4.9 *Se DF é Lipschitz em uma vizinhança aberta de c_0 e $A(c_0)$ tem posto máximo, então existe $r>0$ tal que o problema de valor inicial (4.4) tem solução única em $B = \{u: \|u - c_0\| \leq r\}$. A solução é única no sentido que o traço é único, isto é, a imagem de c é invariante.*

Corolário 4.10 Se DF é Lipschitz em uma vizinhança aberta de $c_0 = (x_0, a_0, \dots, a_q)$ e $A(x_0, a_0, \dots, a_q)$ tem posto máximo, então existe $r > 0$ tal que o problema de valor inicial (4.2), tem uma única solução geral no conjunto:

$$B = \{(X, Y_0, \dots, Y_q) : \|(X, Y_0, \dots, Y_q) - (x_0, a_0, \dots, a_q)\| \leq r\}.$$

4.4 Singularidades

Proposição 4.11 Seja $c: I \rightarrow \mathbb{R}^{m(q+1)+1}, c(s) = (x(s), y_0(s), \dots, y_q(s))$ com $c'(s_0) \neq 0$, uma solução geral de (4.1), seja DF Lipschitz em uma vizinhança de $c(s_0)$.

Então:

i) Se $x'(s_0) = 0$ então $y'_0(s_0) = 0, \dots, y'_{q-1}(s_0) = 0, y'_q(s_0) \neq 0$ e $F_{y_q}(c(s_0))$ é singular.

ii) Se $A(c(s_0))$ tem posto máximo e $F_{y_q}(c(s_0))$ é singular então $x'(s_0) = 0, y'_0(s_0) = 0, \dots, y'_{q-1}(s_0) = 0, y'_q(s_0) \neq 0$ e $F_{y_q}(c(s_0))$ tem posto $m-1$.

4.5 Equações com posto menor que m

A hipótese de posto máximo no corolário 4.10 é essencial.

Se $F_{y_q}(c(s_0))$ é não singular (isto é, a E.A.D. tem índice 0) então $A(c(s_0))$ tem posto máximo, pois:

$$A(c(s_0)) = \begin{pmatrix} F_x(c(s_0)) & F_{y_0}(c(s_0)) & F_{y_1}(c(s_0)) & \cdots & F_{y_{q-1}}(c(s_0)) & F_{y_q}(c(s_0)) \\ y_q(s_0) & 0 & 0 & \cdots & -I_m & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_2(s_0) & 0 & -I_m & \cdots & 0 & 0 \\ y_1(s_0) & -I_m & 0 & \cdots & 0 & 0 \end{pmatrix}$$

Agora se $F_{y_q}(c(s_0))$ tem posto $(m-1)$, $A(c(s_0))$ ainda pode ter posto máximo

pois o vetor $\begin{pmatrix} F_x(c(s_0)) \\ y_q(s_0) \\ \vdots \\ y_1(s_0) \end{pmatrix}$ pode recuperar o posto de $A(c(s_0))$.

Proposição 4.12 Se $F_{y_q}(c(s_0))$ tem posto menor ou igual a $(m-1)$ em uma vizinhança de $c_0=c(s_0)$, então existe trocas de variáveis no domínio e na ordenação de F , onde podemos escrever da seguinte forma:

$$\begin{cases} f(X, W(X), Z(X), \dots, W^{(q-1)}(X), Z^{(q-1)}(X), W^{(q)}(X)) = 0 \\ g(X, W(X), Z(X), \dots, W^{(q-1)}(X), Z^{(q-1)}(X)) = 0 \end{cases} \quad (4.7)$$

onde $f: \Omega \rightarrow \mathbb{R}^n$ e $g: \Omega \rightarrow \mathbb{R}^p$ são funções suficientemente diferenciáveis definidas no conjunto aberto conexo $\Omega \subset \mathbb{R}^{(q+1)n+qp+1}$ e $n+p=m$, com $f_{W^{(q)}}$ não singular.

Para a E.A.D. na forma (4.7), temos definições de solução análogas às anteriores:

Definição 4.13 Uma função (W, Z) com $W: I \rightarrow \mathbb{R}^m$, $Z: I \rightarrow \mathbb{R}^p$ é dita uma solução clássica de (4.7) se $W \in C^q$ e $Z \in C^{q-1}$ em $I \subset \mathbb{R}$ e satisfaz (4.7).

Definição 4.14 A imagem de uma função $c: J \rightarrow \mathbb{R}^{(q+1)n+qp+1}$ é uma solução geral da E.A.D (4.7) se $c \in C^1$ em $J \subset \mathbb{R}$ e satisfaz :

$$\begin{cases} f(c(s)) = 0 \\ g(c(s)) = 0 \\ w_1(s)x'(s) - w'_0(s) = 0 \\ z_1(s)x'(s) - z'_0(s) = 0 \\ \vdots \\ w_{q-1}(s)x'(s) - w'_{q-2}(s) = 0 \\ z_{q-1}(s)x'(s) - z'_{q-2}(s) = 0 \\ w_q(s)x'(s) - w'_{q-1}(s) = 0 \end{cases} \quad (4.8)$$

para todo $s \in J$, onde $c(s) = (x(s), w_0(s), z_0(s), \dots, w_{q-1}(s), z_{q-1}(s), w_q(s))$.

A w -estrutura agora é definida pela função $w: \Omega \rightarrow \mathbb{R}^N$ e $A: \Omega \rightarrow \mathbb{R}^M$, com $N = (qm + (q-1)p)((q+1)m + qp + 1)$ e $M = ((q+1)m + qp)((q+1)m + qp + 1)$, onde $w(x, w_0, z_0, \dots, w_{q-1}, z_{q-1}, w_q) =$

$$= \begin{pmatrix} w_q & 0 & 0 & 0 & 0 & \dots & 0 & 0 & -I_m & 0 & 0 \\ z_{q-1} & 0 & 0 & 0 & 0 & \dots & 0 & -I_p & 0 & 0 & 0 \\ w_{q-1} & 0 & 0 & 0 & 0 & \dots & -I_m & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ z_2 & 0 & 0 & 0 & -I_p & \dots & 0 & 0 & 0 & 0 & 0 \\ w_2 & 0 & 0 & -I_m & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ z_1 & 0 & -I_p & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ w_1 & -I_m & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

e

$$A(c(s)) = \begin{pmatrix} Df(c(s)) \\ Dg(c(s)) \\ w(c(s)) \end{pmatrix}.$$

Então, os seguintes problemas de valor inicial:

$$\begin{cases} f(X, W(X), Z(X), \dots, W^{q-1}(X), Z^{q-1}(X), W^q(X)) = 0 \\ g(X, W(X), Z(X), \dots, W^{q-1}(X), Z^{q-1}(X)) = 0 \\ W^i(x_0) = a_i, i = 0(1)q \\ Z^i(x_0) = b_i, i = 0(1)q-1 \end{cases} \quad (4.9)$$

$$\begin{cases} f(c(s)) = 0 \\ g(c(s)) = 0 \\ w(c(s))c'(s) = 0 \\ c(0) = c_0 = (x_0, a_0, b_0, \dots, a_{q-1}, b_{q-1}, a_q) \end{cases} \quad (4.10)$$

e

$$\begin{cases} A(c(s))c'(s) = 0 \\ c(0) = c_0 = (x_0, a_0, b_0, \dots, a_{q-1}, b_{q-1}, a_q) \end{cases} \quad (4.11)$$

com $f(c_0) = 0$ e $g(c_0) = 0$, são respectivamente equivalentes a (4.3), (4.5) e (4.6).

Lema 4.15 *Considere uma solução geral de (4.7) dada por $c: J \rightarrow \mathbb{R}^{(q+1)n+qp+1}$.*

Se $x'(s) \neq 0$ em um intervalo $I \subset J$, então

$$w_0(s) = W_0(x(s)), w_1(s) = \frac{d}{dx} W_0(x(s)), \dots, w_{q-1}(s) = \frac{d^{q-1}}{dx^{q-1}} W_0(x(s)), w_q(s) = \frac{d^q}{dx^q} W_0(x(s)),$$

$z_0(s) = Z_0(x(s)), z_1(s) = \frac{d}{dx} Z_0(x(s)), \dots, z_{q-1}(s) = \frac{d^{q-1}}{dx^{q-1}} Z_0(x(s))$ em I e (W_0, Z_0) definida em $x(I)$ é uma solução clássica da E.A.D. (4.7).

Teorema 4.16 *Se DF e Dg são Lipschitz em uma vizinhança aberta de c_0 e $A(c_0)$ tem posto máximo, então existe $r > 0$ tal que o problema de valor inicial (4.10) tem solução única em $B = \{v: \|v - c_0\| \leq r\}$. (solução única no sentido de traço único)*

Corolário 4.17 *Se DF e Dg são Lipschitz em uma vizinhança aberta de $(x_0, a_0, b_0, \dots, a_{q-1}, b_{q-1}, a_q)$ e $A(x_0, a_0, b_0, \dots, a_{q-1}, b_{q-1}, a_q)$ tem posto máximo, então existe $r > 0$ tal que o problema de valor inicial (4.10) tem uma única solução geral no conjunto:*

$$B = \left\{ (X, W_0, Z_0, \dots, W_{q-1}, Z_{q-1}, W_q); \left\| (X, W_0, Z_0, \dots, W_{q-1}, Z_{q-1}, W_q) - (x_0, a_0, b_0, \dots, a_{q-1}, b_{q-1}, a_q) \right\| \leq r \right\}$$

Proposição 4.18 *Seja $c: J \rightarrow \mathbb{R}^{(q+1)n+qp+1}$, $c(s) = (x(s), w_0(s), z_0(s), \dots, w_{q-1}(s), z_{q-1}(s), w_q(s))$ com $c'(s_0) \neq 0$ uma solução geral da equação (4.7) e Df e Dg Lipschitz em uma vizinhança de $c(s_0)$.*

i) *Se $x'(s_0) = 0$ então $w'_0(s_0) = 0, z'_0(s_0) = 0, \dots, w'_{q-1}(s_0) = 0, z'_{q-1}(s_0) \neq 0$, e $g_{Z_{q-1}}(c(s_0))$ é singular.*

ii) Se $A(c(s_0))$ tem posto máximo e $g_{Z_{q-1}}(c(s_0))$ é singular então $x'(s_0) = 0$, $w'_0(s_0) = 0, z'_0(s_0) = 0, \dots, w'_{q-1}(s_0) = 0, z'_{q-1}(s_0) \neq 0$, e $g_{Z_{q-1}}(c(s_0))$ tem posto $p-1$

Exemplo 4.3

$$\text{Seja } \begin{cases} (Y'(X))^2 = Y(X) \\ Y(x_0) = y_0 \\ Y'(x_0) = z_0 \end{cases} \quad \text{onde } x_0 > 0, y_0 > 0, z_0 > 0 \text{ e } z_0^2 = y_0.$$

Uma das soluções clássicas desta equação é : $Y(X) = ((X - x_0) / 2 + y_0^{1/2})^2$ para $X \in R$ e o conjunto de singularidades é $\{(X, 0) ; X \in R\}$. O w -sistema associado é

$$\begin{cases} y_1^2(s) - y_0(s) = 0 \\ y_1(s)x'(s) - y'_0(s) = 0 \\ x(0) = x_0 \\ y_0(0) = y_0 \\ y_1(0) = z_0 \end{cases}$$

e uma das soluções gerais é dada por $c(s) = (s + x_0, (s / 2 + y_0^{1/2})^2, s / 2 + y_0^{1/2})$ para $s \in R$. Como $x'(s) = 1 \neq 0$ temos que $c(s) = (X, Y(X), Y'(X))$. (veja fig.4.1)

fig 4.1

Note que a hipótese de posto máximo em $A(c_0)$ é essencial como pode ser visto neste exemplo.

Aqui, $A(c(s)) = \begin{pmatrix} 0 & -1 & 2y_1(s) \\ y_1(s) & -1 & 0 \end{pmatrix}$ não possui posto máximo

quando $y_1(s) = 0$ e se $c_0 = (x_0, 0, 0)$ então o exemplo tem muitas soluções gerais, duas delas são dadas por $c(s) = (s + x_0, 0, 0)$ e $c(s) = (s + x_0, s^2/4, s/2)$.

CAPÍTULO 5

MÉTODO NUMÉRICO PARA E.A.D. COM SINGULARIDADE

5.1 Introdução

Neste capítulo apresentaremos uma nova abordagem para a resolução numérica de uma E.A.D. , com base na teoria apresentada no capítulo 4.

Lembramos que, se desejamos resolver numericamente um sistema do tipo

$$F(x)=0$$

em vez de utilizarmos o método de Newton ,

$$x^{k+1} = x^k - (DF(x^k))^{-1} F(x^k)$$

podemos usar o método de Newton modificado,

$$x^{k+1} = x^k - \alpha (DF(x^0))^{-1} F(x^k)$$

sem perda de convergência, onde α é o fator de aceleração para o método.

Observação: α é um coeficiente relacionado com o raio de convergência da matriz de iteração, mais detalhes serão dados no capítulo 6.

Dada a E.A.D. (2.1) os métodos numéricos usualmente utilizados (como o DASSL, por exemplo) encontram uma aproximação para a solução clássica $y(t)$, o que faremos através desta nova abordagem é encontrar uma aproximação para a solução geral $c(s)$ da E.A.D. (4.1), utilizando um método BDF.

5.2 Descrição do Método

Consideremos então o sistema diferencial (4.4) associado à E.A.D. (4.1) e o p.v.i. (4.5). Pelo teorema 4.9 este p.v.i. possui localmente uma única solução.

Por diferenciação obteve-se de (4.4) outro p.v.i. equivalente, dado por (4.6). Na demonstração do teorema 4.9 [Pires-94], foi verificado que o p.v.i. (4.6) é equivalente ao p.v.i. para uma equação diferencial ordinária da forma :

$$\begin{cases} c'(s) = \tau(A(c(s))) \\ c(0) = c_0 \end{cases} \quad (5.1)$$

onde $A(u) \in \mathbb{R}^{(q+1)m \times (q+1)m+1}$, $\tau(A(u)) \in \mathbb{R}^{(q+1)m+1}$ com $\|\tau(A(u))\| = 1$,

$$\det \begin{pmatrix} A(u) \\ \tau(A(u))^t \end{pmatrix} > 0 \text{ e } A(u) \tau(A(u)) = 0.$$

Resolveremos então o p.v.i. (5.1). Primeiramente, encontraremos o vetor $\tau(A)$, que pode ser obtido da seguinte maneira:

Suponhamos que a decomposição QR de A^t ($A^t = Q \begin{pmatrix} R \\ 0^t \end{pmatrix}$) seja dada onde Q

é uma matriz $((q+1)m+1) \times ((q+1)m+1)$ ortogonal, isto é, $Q^t Q = Q Q^t = I$ e R é uma matriz triangular superior $(q+1)m+1 \times (q+1)m$, não singular, isto é, $R_{ij} = 0$ para $i > j$ e $R_{ii} \neq 0$.

Assim se z denota a última coluna de Q então

$$Az = \begin{pmatrix} R^t & 0 \end{pmatrix} Q^t z = \begin{pmatrix} R^t & 0 \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} = 0,$$

pois as colunas de Q são ortonormais. Ainda nos resta encontrar o sinal de z tal que

$$\det \begin{pmatrix} A \\ z^t \end{pmatrix} > 0.$$

Note que: $\begin{pmatrix} A^t & z \end{pmatrix} = Q \begin{pmatrix} R & 0 \\ 0^t & 1 \end{pmatrix},$

então,

$$\det \begin{pmatrix} A \\ z^t \end{pmatrix} = \det \begin{pmatrix} A^t & z \end{pmatrix} = \det Q \cdot \det R.$$

Logo, $\tau(A) = \pm z$, concordando com o sinal de $\det Q \cdot \det R$, se é positivo ou negativo.

Observação: $\det R$ é o produto dos elementos diagonais de R , e seu sinal é facilmente determinado, bem como o sinal do $\det Q$, pois Q é ortogonal, então $\det Q = \pm 1$.

Por exemplo: se rotações de Givens são usadas para obter a decomposição QR de A^t então $\det Q = 1$, se usarmos reflexões de Householder, cada reflexão muda o sinal, e então o sinal do $\det Q$ é dado por: $(-1)^p$, onde p é o número de reflexões que são envolvidas na fatoração de A^t pelo método de Householder. Em ambos os casos não existe qualquer custo computacional adicional.

O custo computacional para determinar $\tau(A)$ é aproximadamente $\frac{2}{3}(m(q+1))^3$, para cada passo. No entanto, este cálculo pode ser minimizado, pois, olhando para a expressão de w , e escrevendo

$$\begin{aligned} y_1(s)\tau_x - \tau_{y_0} &= 0 \Rightarrow \tau_{y_0} = y_1(s)\tau_x \\ y_2(s)\tau_x - \tau_{y_1} &= 0 \Rightarrow \tau_{y_1} = y_2(s)\tau_x \\ &\vdots \\ y_q(s)\tau_x - \tau_{y_{q-1}} &= 0 \Rightarrow \tau_{y_{q-1}} = y_q(s)\tau_x \end{aligned} \tag{5.2}$$

onde $\tau = (\tau_x, \tau_{y_0}, \dots, \tau_{y_q})$, substituindo em $DF(c(s))\tau = 0$, temos

$$B(c(s))\psi = \begin{pmatrix} F_x + F_{y_0}y_1 + \dots + F_{y_{q-1}}y_q & F_{y_q} \end{pmatrix} \begin{pmatrix} \tau_x \\ \tau_{y_q} \end{pmatrix} = 0.$$

Obtemos $\psi = \begin{pmatrix} \tau_x \\ \tau_{y_q} \end{pmatrix}$, usando uma decomposição QR para B^t com

aproximadamente um custo computacional da ordem de $\frac{2}{3}(m+1)^3$. Portanto podemos utilizar a relação (5.2) para obtermos τ .

O método de EULER explícito para o sistema (5.1) é dado por:

$$c_n = c_{n-1} - h\tau(A(c_{n-1})) = 0 \quad (5.3)$$

O esquema numérico (5.3) será utilizado como um método *inicializador*, pois c'_0 não é dado de entrada.

Para o passo corretor um método *BDF* pode ser usado no sistema (4.4):

$$\begin{cases} F(c_n) = 0 \\ w(c_n) \frac{\rho c_n}{h} = 0 \end{cases} \quad (5.4)$$

onde $\rho c_n = \sum_{i=0}^k \gamma_i c_{n-i}$, com $\gamma_0 = 1$. Para obtermos c_n , um método de Newton modificado para sistemas indeterminados é utilizado, isto é,

$$c_n^{i+1} = c_n^i - DG^+(c_n^{(0)})G(c_n^i), \text{ onde } DG^+ \text{ é a inversa de Moore-Penrose (ver}$$

capítulo 1) avaliada no ponto predito. Observemos que a dimensão de DG^+ é: $(q+1)m \times (q+1)m+1$, logo o custo computacional para decompô-la na forma QR é de $\frac{2}{3}(m(q+1)+1)^3$.

Neste caso a parametrização de c está implícita. Uma forma de explicitarmos esta parametrização é considerarmos uma parametrização por comprimento de arco, adicionando a equação $\|c'(s)\|^2 = 1$, no sistema (4.4) que torna-se:

$$\left\{ \begin{array}{l} F(x, y_0, y_1, \dots, y_q) = 0 \\ y_1 x' - y'_0 = 0 \\ y_2 x' - y'_1 = 0 \\ \vdots \\ y_q x' - y'_{q-1} = 0 \\ \|x'\|^2 + \|y'_0\|^2 + \|y'_1\|^2 + \dots + \|y'_q\|^2 - 1 = 0 \end{array} \right. \quad (5.5)$$

Analogamente, para o passo corretor um método BDF pode ser utilizado no sistema (5.5) :

$$G(c_n) = \left\{ \begin{array}{l} F(c_n) = 0 \\ w(c_n) \frac{\rho c_n}{h} = 0 \\ \left\| \frac{\rho c_n}{h} \right\|^2 - 1 = 0 \end{array} \right. \quad (5.6)$$

com $G: \mathbb{R}^{(q+1)m+1} \rightarrow \mathbb{R}^{(q+1)m+1}$. Logo $DG^+ = (DG)^{-1}$, onde DG é quadrada com ordem $(q+1)m+1$ e dada por:

$$\begin{pmatrix} F_x & F_{y_0} & F_{y_1} & F_{y_2} & \dots & F_{y_{q-1}} & F_{y_q} \\ (y_{1,n})^p & -I & (x'_n)^p I & 0 & \dots & 0 & 0 \\ (y_{2,n})^p & 0 & -I & (x'_n)^p I & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ (y_{q-1,n})^p & 0 & 0 & \dots & \dots & (x'_n)^p I & 0 \\ (y_{q,n})^p & 0 & 0 & \dots & \dots & -I & (x'_n)^p I \\ 2(x'_n)^p & 2(y'_{0,n})^p & 2(y'_{1,n})^p & 2(y'_{2,n})^p & \dots & 2(y'_{q-1,n})^p & 2(y'_{q,n})^p \end{pmatrix}$$

Um método de Newton modificado para o sistema (5.6) é dado por:

$$c_n^{i+1} = c_n^i - \alpha DG^{-1}(c_n^0) G(c_n^i)$$

onde

$$G(c_n) = \begin{pmatrix} F(c_n) \\ w(c_n) \frac{\rho c_n}{h} \\ \left\| \frac{\rho c_n}{h} \right\|^2 - 1 \end{pmatrix}, \quad DG^{-1} \text{ é a inversa de } G \text{ avaliada no ponto predito e } \alpha \text{ é o}$$

fator de aceleração para o método de Newton.

Para calcularmos $v = DG^{-1}(c_n^0)G(c_n^i)$, tomamos a decomposição QR de DG: $DG = QR$. Daí, $v = \alpha DG^{-1}(c_n^0)G(c_n^i) \Rightarrow \frac{1}{\alpha} DG(c_n^0)v = G(c_n^i) \Rightarrow Rv = \gamma$ onde γ é dado por: $Q^t G(c_n^i)$.

O custo computacional para o método de Newton Modificado é aproximadamente $\frac{2}{3}(m(q+1)+1)^3 + j(m(q+1)+1)^2$, onde j é o número de iterações.

Podemos melhorar o desempenho computacional do método, realizando simples permutações de linhas e colunas na matriz DG, tal que ela adquira a seguinte forma:

$$DG = \begin{pmatrix} F_{y_q} & F_{y_{q-1}} & F_{y_{q-2}} & \cdots & F_{y_1} & F_{y_0} & F_x \\ (x')^p I & -I & 0 & \cdots & 0 & 0 & y_q \\ 0 & (x')^p I & -I & \cdots & 0 & 0 & y_{q-1} \\ 0 & 0 & (x')^p I & \ddots & 0 & 0 & y_{q-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & (x')^p I & -I & y_1 \\ 2(y_q')^p & 2(y_{q-1}')^p & 2(y_{q-2}')^p & \cdots & 2(y_1')^p & 2(y_0')^p & 2(x')^p \end{pmatrix}.$$

Com esta simples mudança na construção de DG obtemos um significativo ganho computacional ao calcularmos a decomposição QR de DG, pois os blocos nulos abaixo da diagonal principal não serão alterados, preservando assim a estrutura de esparsidade da matriz, bem como reduzindo o número de operações envolvidas no processo QR, que agora passa a ser aproximadamente da ordem de $\frac{2}{3}m^3(q+1)$. É conveniente lembrar que o custo do método esta diretamente relacionado com o custo da decomposição QR.

5.3 Tentativas de Redução da Ordem do Sistema a Ser Resolvido pelo Método de Newton

Tentativas de redução da ordem do sistema (5.6) foram colocadas em prática, mas a instabilidade gerada por estas reduções tornaram-nas ineficientes. Brevemente vamos citar duas dessas tentativas:

$$1. \text{ Ao considerarmos o sistema } G(c_n) = \begin{cases} F(c_n) = 0 \\ w(c_n) \frac{\rho c_n}{h} = 0 \text{ e } \rho c_n = c_n + C_n \\ \left\| \frac{\rho c_n}{h} \right\|^2 - 1 = 0 \end{cases}$$

com $c_n = (x_n, y_{0n}, \dots, y_{qn})$ e $C_n = (C_{x_n}, C_{y_{0n}}, \dots, C_{y_{qn}})$, reduziremos a ordem do sistema a $(m+1)$ equações e consequentemente a ordem da matriz de iteração a $(m+1) \times (m+1)$.

Olhando para a expressão de w , desde que $w(c_n) \frac{\rho c_n}{h} = 0$, temos:

$$\begin{aligned} y_{1,n}(x_n + C_{x_n}) - (y_{0,n} + C_{y_{0,n}}) &= 0 \Rightarrow y_{0,n} = y_{1,n}(x_n + C_{x_n}) - C_{y_{0,n}} \\ y_{2,n}(x_n + C_{x_n}) - (y_{1,n} + C_{y_{1,n}}) &= 0 \Rightarrow y_{1,n} = y_{2,n}(x_n + C_{x_n}) - C_{y_{1,n}} \\ &\vdots \\ y_{q,n}(x_n + C_{x_n}) - (y_{q-1,n} + C_{y_{q-1,n}}) &= 0 \Rightarrow y_{q-1,n} = y_{q,n}(x_n + C_{x_n}) - C_{y_{q-1,n}} \end{aligned}$$

ou

$$y_{i,n} = y_{q,n}(x_n + C_{x_n})^{q-i} - \sum_{j=0}^{q-1-i} C_{y_{i+j,n}}(x_n + C_{x_n})^j, \quad i=0(1)q-1. \quad (5.7)$$

Portanto, para as primeiras m equações em (5.6), temos:

Para calcular c_n resolvemos $H(x_n, y_{q,n}) = 0$, com um custo computacional de aproximadamente $\frac{2}{3}(m+1)^3 + j(m+1)^2$, onde j é o número de iterações no método de Newton Modificado. (c_n é obtido da relação (5.7))

2. O código DASSL trabalha com a matriz das derivadas da seguinte maneira:

$$J = \alpha \frac{\partial F}{\partial y'} + \frac{\partial F}{\partial y}, \text{ onde } \alpha \text{ é um coeficiente relacionado com o raio de}$$

convergência da matriz de iteração, mais detalhes serão dados no capítulo 6.

O GSDAE também trabalha com uma estrutura semelhante, isto é:

$$DG = \alpha \frac{\partial G}{\partial c'} + \frac{\partial G}{\partial c}, \text{ onde}$$

$$\frac{\partial G}{\partial c} = \begin{pmatrix} F_x & F_{y_0} & F_{y_1} & F_{y_2} & \cdots & F_{y_{q-1}} & F_{y_q} \\ 0 & 0 & x'I & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & x'I & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & x'I & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & x'I \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \quad e$$

$$\alpha \frac{\partial G}{\partial c'} = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \alpha y_1 & -\alpha I & 0 & 0 & \cdots & 0 & 0 \\ \alpha y_2 & 0 & -\alpha I & 0 & \cdots & 0 & 0 \\ \alpha y_3 & 0 & 0 & -\alpha I & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha y_q & 0 & 0 & 0 & \cdots & -\alpha I & 0 \\ 2\alpha x' & 2\alpha y_0'^t & 2\alpha y_1'^t & 2\alpha y_2'^t & \cdots & 2\alpha y_{q-1}'^t & 2\alpha y_q'^t \end{pmatrix}.$$

Observação: GSDAE é a denominação do código gerado apartir do método numérico aqui introduzido.

Considerando DG como acima e a iteração de Newton como: $DG v = G$, onde $v = c_{n+1}^{k+1} - c_{n+1}^k$, ou seja :

$$\left\{ \begin{array}{l} F_x v_x + F_{y_0} v_0 + \dots + F_{y_q} v_q = F(x^k, y_0^k, \dots, y_q^k) \\ \alpha y_1^p v_x - \alpha v_0 + x'^p v_1 = y_1^k x'^k - y_0'^k \\ \alpha y_2^p v_x - \alpha v_1 + x'^p v_2 = y_2^k x'^k - y_1'^k \\ \vdots \\ \alpha y_{q-1}^p v_x - \alpha v_{q-2} + x'^p v_{q-1} = y_{q-1}^k x'^k - y_{q-2}'^k \\ \alpha y_q^p v_x - \alpha v_{q-1} + x'^p v_q = y_q^k x'^k - y_{q-1}'^k \\ 2\alpha \left(x'^p v_x + \sum_{i=0}^q (y_i'^p)^t v_i \right) = |x'|^2 + \sum_{i=0}^q \|y_i'^k\|^2 - 1 \end{array} \right. \quad (5.10)$$

Novamente nosso objetivo é reduzir a ordem do sistema (5.10) a $m+1$ equações. Para alcançá-lo vamos expressar $v_i, i=0(1)q-1$ em função de (v_x, v_q) , que após um certo algebrismo obtemos:

$$v_i = \frac{1}{\alpha} (y_i'^k - y_{i+1}^k x'^k + x'^p v_{i+1}) + y_{i+1}^p v_x \quad (5.11)$$

Para efeito de simplificação consideremos as seguintes relações:

a) $w = \frac{1}{\alpha};$

b) $\delta_{i+1} = y_i'^k - y_{i+1}^k x'^k, i=(q-1)(1)0;$

c) $\gamma_i = y_i^p, i=q(1)0;$

d) $\beta = x'^p;$

e) $\psi_i = (y_i'^p)^t, i=0(1)q;$

f) $NORK = \|c'^k\|^2 - 1;$ e assim temos:

$v_i = w(\delta_{i+1} + \beta v_{i+1}) + \gamma_{i+1} v_x, i=(q-1)(1)0.$ As expressões de H e DH são dadas pelas seguintes relações:

$$h_p = F^k - \sum_{i=0}^{q-1} F_{y_i} \left(\sum_{j=i+1}^q w^{(j-i)} \beta^{(j-i-1)} \delta_j \right), p=1(1)m;$$

$$h_{m+1} = \frac{w}{2} \text{NORK} - \sum_{i=0}^{q-1} \psi_i \left(\sum_{j=i+1}^q w^{(j-i)} \beta^{(j-i-1)} \delta_j \right);$$

$$dh_{pl} = F_x + \sum_{i=0}^{q-1} F_{y_i} \left(\sum_{j=i+1}^q (w\beta)^{(j-i-1)} \gamma_j \right), p=1(1)m;$$

$$dh_{m+1l} = \beta + \sum_{i=0}^{q-1} \psi_i \left(\sum_{j=i+1}^q (w\beta)^{(j-i-1)} \gamma_j \right);$$

$$dh_{pt} = F_{y_q} + \sum_{i=0}^{q-1} F_{y_i} (w\beta)^{q-i}, p=1(1)m, t=2(1)m+1;$$

$$dh_{m+lt} = \sum_{i=0}^q \psi_i (w\beta)^{q-i}, t=2(1)m+1.$$

5.4 Conclusões

Ao utilizarmos a formulação (5.5)/(5.6) temos uma certa vantagem sobre a anterior, pois, podemos utilizar qualquer método de resolução de sistemas, como L.U., QR, dentre outros, para resolvermos o sistema: $DG(c_n^{i+1} - c_n^i) = -G(c_n^i)$.

Quando introduzimos a restrição $\|c'(s)\|^2 = 1$ no sistema, não aumentamos o custo computacional. Como vimos a maior parcela do custo esta relacionada com a decomposição QR que neste caso permaneceu inalterada, tendo em vista que antes tínhamos que decompor $(DG^+)^t_{(q+1)m+1 \times (q+1)m}$ na forma QR, ou seja,

$$\left(DG^+ \right)_{(q+1)m+1x(q+1)m}^t = Q_{(q+1)m+1x(q+1)m+1} R_{(q+1)m+1x(q+1)m}$$

e agora

$$(DG)_{(q+1)m+1x(q+1)m+1} = Q_{(q+1)m+1x(q+1)m+1} R_{(q+1)m+1x(q+1)m+1}$$

CAPÍTULO 6

Algoritmos e Estratégias no GSDAE

6.1 Introdução

Neste capítulo, discutiremos:

- Estratégias para a predição e correção da solução (resolução do sistema não linear), onde apresentamos as fórmulas básicas, os coeficientes utilizados na construção do método BDF e descrição do processo de correção;
- Critério para aceitação do passo, técnicas para determinar a ordem do método BDF e a amplitude do passo ;
- Além de alguns comentários sobre a implementação do método.

Devemos observar que todas as estratégias utilizadas são baseadas nas do código DASSL [Brenan-89].

6.2 Fórmulas Básicas

Partiremos do princípio que possuímos aproximações c_{n-i} para as soluções verdadeiras de $c(s_{n-i})$ para $i=0(1)k$, onde k é a ordem do método BDF que estamos utilizando. Gostaríamos de encontrar uma aproximação para a solução em s_{n+1} . Os valores preditos para c e c' em s_{n+1} são obtidos pela avaliação do polinômio preditor, onde: $c_{n+1}^{(0)} = \chi_{n+1}^p(s_{n+1})$ e $c'_{n+1}^{(0)} = \chi'_{n+1}^p(s_{n+1})$. O polinômio preditor é tal que $\chi_{n+1}^p(s_{n-i}) = c_{n-i}$, $i = 0(1)k$.

A solução que finalmente é aceita pelo GSDAE é obtida pelo método BDF de ordem k com coeficientes principais fixos ("fórmula corretora"). A implementação com os coeficientes principais fixos é uma maneira de estender o método BDF com passo fixo para uma forma de passo variável. Há três principais aproximações para estender um método de passo múltiplo com passo fixo para um de passo variável. Estas formulações são denominadas de "coeficientes fixos", "coeficientes variáveis" e "coeficientes principais fixos". Os métodos com coeficientes fixos caracterizam-se por serem muito eficientes em problemas suaves mas tornam-se ineficientes ou possivelmente instáveis quando aplicados em problemas que requerem frequentes ajustes na amplitude do passo. Os métodos com coeficientes variáveis são as implementações mais estáveis, mas possuem a desvantagem de requererem um maior número de avaliações da matriz Jacobiana em um intervalo quando a amplitude do passo é ajustada e portanto são usualmente considerados menos eficientes do que uma implementação com coeficientes fixos para a maioria dos problemas. A formulação com coeficientes principais fixos está entre as fórmulas com coeficientes fixos e a com coeficientes variáveis, oferecendo as vezes menos estabilidade juntamente com poucas avaliações da matriz Jacobiana, do que a formulação com coeficientes variáveis. Ainda é uma questão em aberto para E.D.O. Stiff e para E.A.D. qual formulação é a melhor para um código geral. [Brenan-89] e com mais detalhes em [Jackson-Sacks-Davis-80].

6.2.1 Coeficientes

Os valores da predição $(c_{n+1}^{(0)}, c_{n+1}'^{(0)})$ e da correção (c_{n+1}) em s_{n+1} são definidos em termos de polinômios que interpolam a solução nos passos anteriores.

Seguindo as idéias de Krogh [Krogh-72], Shampine e Gordon [Shampine-75], estes polinômios são representados no GSDAE em termos de diferenças divididas modificadas de c . As quantidades que serão atualizadas de passo a passo são dadas por:

$$\psi_i(n+1) = h_{n+1} + h_n + \dots + h_{n+2-i} = s_{n+1} - s_{n+1-i}, \quad i \geq 1$$

$$\alpha_i(n+1) = h_{n+1}/\psi_i(n+1), \quad i \geq 1$$

$$\beta_1(n+1) = 1$$

$$\beta_i(n+1) = \frac{\psi_1(n+1)\psi_2(n+1)\cdots\psi_{i-1}(n+1)}{\psi_1(n)\psi_2(n)\cdots\psi_{i-1}(n)}, \quad i > 1$$

$$\phi_1(n) = c_n$$

$$\phi_i(n) = \psi_1(n)\psi_2(n)\cdots\psi_{i-1}(n)[c_n, c_{n-1}, \dots, c_{n-i+1}], \quad i > 1$$

$$\phi_i^*(n) = \beta_i(n+1)\phi_i(n), \quad i \geq 1$$

$$\sigma_1(n+1) = 1$$

$$\sigma_i(n+1) = \frac{h_{n+1}^i(i-1)!}{\psi_1(n+1)\psi_2(n+1)\cdots\psi_i(n+1)}, \quad i > 1$$

$$\gamma_1(n+1) = 0$$

$$\gamma_i(n+1) = \gamma_{i-1}(n+1) + \alpha_{i-1}(n+1)/h_{n+1}, \quad i > 1$$

$$\alpha_s = -\sum_{j=1}^k \frac{1}{j}$$

$$\alpha^0(n+1) = -\sum_{j=1}^k \alpha_j(n+1)$$

(6.1)

As diferenças divididas são definidas pela recorrência

$$[c_n] = c_n$$

$$[c_n, c_{n-1}, \dots, c_{n-k}] = \frac{[c_n, c_{n-1}, \dots, c_{n-k+1}] - [c_{n-1}, c_{n-2}, \dots, c_{n-k}]}{s_n - s_{n-k}}.$$

6.2.2 Polinômio Predictor

O polinômio predictor é dado em termos das diferenças divididas:

$$\chi_{n+1}^p(s) = c_n + (s - s_n)[c_n, c_{n-1}] + (s - s_n)(s - s_{n-1})[c_n, c_{n-1}, c_{n-2}] + \dots \\ \dots + (s - s_n)(s - s_{n-1}) \dots (s - s_{n-k+1})[c_n, c_{n-1}, \dots, c_k] \quad (6.2)$$

Avaliando χ_{n+1}^p em s_{n+1} , e reescrevendo (6.2) em termos da notação em (6.1), obtemos a fórmula preditora :

$$c_{n+1}^{(0)} = \sum_{i=1}^{k+1} \phi_i^* .$$

Diferenciando (6.2) e avaliando $\chi_{n+1}^{p'}$ em s_{n+1} , obtemos após alguns cálculos:

$$c_{n+1}'^{(0)} = \sum_{i=1}^{k+1} \gamma_i (n+1) \phi_i^*(n) .$$

6.2.3 Iteração Corretora

6.2.3.1 Fórmula Corretora

A solução da "fórmula corretora" é representada por $c_{n+1} = (x_{n+1}, y_{n+1,j})$, $x_{n+1} \in \mathbb{R}$ e $y_{n+1,j} \in \mathbb{R}^m$, $j=0(1)q$; tal que o polinômio corretor $\chi_{n+1}^c(s)$ e sua derivada satisfazem a E.A.D. (5.5) em s_{n+1} e além disso o polinômio corretor interpola o polinômio predictor em k pontos igualmente espaçados anteriores à s_{n+1} (utilização da formulação com coeficientes principais fixos) , isto é ,

$$\left\{ \begin{array}{l} \chi_{n+1}^c(s_{n+1}) = c_{n+1} \\ \chi_{n+1}^c(s_{n+1} - ih_{n+1}) = \chi_{n+1}^p(s_{n+1} - ih_{n+1}), 1 \leq i \leq k, \\ G = \begin{cases} F(\chi_{n+1}^c(s_{n+1})) = 0 \\ w(\chi_{n+1}^c(s_{n+1}))\chi_{n+1}'^c(s_{n+1}) = 0 \\ \|\chi_{n+1}'^c(s_{n+1})\|^2 - 1 = 0 \end{cases} \end{array} \right. \quad (6.3)$$

Para encontrarmos a fórmula corretora, notemos que (6.3) implica que:

$$\chi_{n+1}^c(s_{n+1}) - \chi_{n+1}^p(s_{n+1}) = b(s)(c_{n+1} - c_{n+1}^{(0)}) \quad , \quad (6.4)$$

onde $b(s_{n+1} - ih_{n+1}) = 0$ e $b(s_{n+1}) = 1$, $i = 1(1)k$. Diferenciando (6.4) e

$$\text{avaliando em } s_{n+1}, \text{ resulta: } \alpha_s(c_{n+1} - c_{n+1}^{(0)}) + h_{n+1}(c_{n+1}' - c_{n+1}'^{(0)}) = 0 \quad (6.5)$$

onde o coeficiente principal fixo α_s está definido em (6.1).

Resolvendo (6.5) para c_{n+1}' , obtemos que a iteração corretora deve resolver:

$$G\left(s_{n+1}, c_{n+1}, c_{n+1}'^{(0)} - \frac{\alpha_s}{h_{n+1}}(c_{n+1} - c_{n+1}^{(0)})\right) = \begin{cases} F(c_{n+1}) = 0 \\ w(c_{n+1})\left(c_{n+1}'^{(0)} - \frac{\alpha_s}{h_{n+1}}(c_{n+1} - c_{n+1}^{(0)})\right) = 0 \\ \left\|\left(c_{n+1}'^{(0)} - \frac{\alpha_s}{h_{n+1}}(c_{n+1} - c_{n+1}^{(0)})\right)\right\|^2 - 1 = 0 \end{cases} \quad (6.6)$$

A solução c_{n+1} , definida implicitamente pelas condições (6.3), é determinada pela resolução de um sistema de equações não lineares através do método de Newton Modificado.

6.2.3.2 Solução do Sistema Não Linear

No capítulo 5 obtivemos a seguinte iteração de Newton :

$$c_{n+1}^{(m+1)} = c_{n+1}^{(m)} - ac DG^{-1}(c_{n+1}^{(0)}) G(c_{n+1}^{(m)}) \quad , \text{ onde } G(c_{n+1}) = \begin{pmatrix} F(c_{n+1}) \\ w(c_{n+1}) \frac{\rho c_{n+1}}{h} \\ \left\| \frac{\rho c_{n+1}}{h} \right\|^2 - 1 \end{pmatrix} \text{ com}$$

ρc_{n+1} dado agora por $\left(c_{n+1}'^{(0)} - \frac{\alpha_s}{h_{n+1}} (c_{n+1} - c_{n+1}^{(0)}) \right)$, DG^{-1} é a inversa de DG e

ac é o fator de aceleração para o método de Newton.

O custo computacional na iteração de Newton está relacionado com o número de iterações m e com o cálculo do vetor $v = DG^{-1}(c_{n+1}^0) G(c_{n+1}^m)$. Observe que a cada passo teríamos que avaliar DG no ponto predito e realizar uma decomposição QR para podermos resolver o sistema. Se a matriz de iteração e a constante $\alpha = -\alpha_s / h_{n+1}$ não tem mudado muito desde que a última matriz de iteração foi calculada, então é desejável utilizar a matriz de iteração anterior ao invés de reavaliá-la, e como consequência não teremos que realizar uma nova decomposição QR da matriz em todo passo.

Então a iteração que realmente utilizamos é:

$$(c_{n+1}^{m+1}) = (c_{n+1}^{m+1}) - ac (D\hat{G})^{-1} G(c_{n+1}^m), \quad (6.7) \text{ onde}$$

$(D\hat{G})^{-1}$ é a matriz de iteração calculada num passo anterior.

Nossa principal preocupação agora é quando deveremos atualizar a matriz de iteração DG .

A constante ac é escolhida para acelerar a convergência da iteração quando $\alpha \neq \hat{\alpha}$, onde $\hat{\alpha}$ é calculada no mesmo passo que $(D\hat{G})^{-1}$. Para encontrar o valor ótimo de ac , notemos que a iteração (6.7) converge, para uma boa condição inicial, quando o raio espectral de $B = (I - ac(D\hat{G})^{-1} DG)$ for menor do que um. Quando (6.7) é utilizado para resolver a EDO linear, $y' - Ay = 0$, encontramos que se λ é um auto valor de A , então $\bar{\lambda} = 1 - ac(\alpha - \lambda) / (\hat{\alpha} - \lambda)$ é um auto valor de B . Para uma E.A.D. geral onde os auto valores de DG são desconhecidos, tomamos ac tal que

minimize o máximo dos λ no lado esquerdo do plano complexo. Esta minimização dá:

$$ac = \frac{2\hat{\alpha}}{\alpha + \hat{\alpha}}.$$

Quando ac é escolhido desta maneira, o correspondente $\bar{\lambda}$ é dado por :

$$\bar{\lambda} = \frac{\hat{\alpha} - \alpha}{\alpha + \hat{\alpha}} \quad (6.8).$$

A expressão (6.8) é útil na decisão de quando deveremos avaliar a nova matriz de iteração. Assumindo que $\bar{\lambda}$ é uma aproximação para o raio de convergência de (6.7) quando $\alpha \neq \hat{\alpha}$, e que estamos dispostos a tolerar um raio de convergência de $\mu = 0.25$, então uma nova matriz de iteração é calculada antes de tentar a iteração sempre que a amplitude do passo e/ou a ordem mudam tal que $|\bar{\lambda}| > \mu$.

Critério de Parada

Uma decisão que é importante para a confiança e eficiência de um código como o GSDAE é quando terminar a iteração corretora (6.7). É bem conhecido que

$$\|c^* - c^{(i+1)}\| \leq \frac{\rho}{1-\rho} \|c^{(i+1)} - c^{(i)}\|$$

onde c^* é a solução da equação corretora e ρ é o raio de convergência da iteração. Seguindo Shampine [Shampine-75], a iteração é continuada até que

$$\frac{\rho}{1-\rho} \|c^{(i+1)} - c^{(i)}\| < 0.33 \quad (6.9)$$

tal que o erro da iteração $\|c^* - c^{(i+1)}\|$ seja suficientemente pequeno. Embora a condição (6.9) parece ser em um primeiro momento independente de qualquer tolerância para erro requerida pelo usuário, esta informação está escondida na norma. Em particular a norma utilizada no GSDAE é uma ‘norma-peso’, onde os pesos dependem de uma combinação dos erros relativo e absoluto e dos valores de c no início do passo. A constante 0.33 foi escolhida tal que o erro devido ao término da iteração corretora não afetaria de maneira prejudicial a estimativa do erro de integração. Mais precisamente, se

os pesos são todos iguais e proporcionais a uma desejável tolerância ε do erro de integração, então a condição (6.9) pode ser interpretada como uma restrição ao erro da iteração corretora levando-o a ser menor que 0.33ε . Um máximo de quatro iterações é permitido. Sempre que $\rho > 0.9$, a iteração é considerada falha, e a matriz de iteração, se não for corrente, é reavaliada.

Uma estimativa para o raio de convergência ρ é definido por:

$$\rho = \left(\frac{\|c^{(i+1)} - c^{(i)}\|}{\|c^{(1)} - c^{(0)}\|} \right)^{1/4}, \text{ e é estimado sempre que duas ou mais iterações no método de}$$

Newton tenham sido efetuadas.

Se a diferença entre o preditor e a primeira correção é muito pequena (erro relativo em c), a iteração é terminada, neste caso ρ não é computado, e assim é tomado como sendo o último raio calculado.

Se a iteração corretora falhar na convergência, a matriz de iteração é reavaliada e a iteração é reiniciada. Com a nova matriz de iteração, se o corretor voltar a falhar, a amplitude do passo é reduzida por um fator de um quarto. Após dez falhas consecutivas, ou se a amplitude do passo torna-se tão pequena que $s+h$ é igual a s na aritmética do computador, o GSDAE retorna ao programa principal um "flag" de erro.

6.2.3.3 Algoritmo para a Iteração Corretora

sentinela: variável que indica se houve a convergência ou não;

nint: contador para a iteração corretora $\Rightarrow (1 \leq nint \leq 10)$;

ncor: contador para a iteração de Newton $\Rightarrow (1 \leq ncor \leq 4)$;

pnrm: variável real que armazena o valor da norma do ponto predito;

d: variável real que armazena o valor da norma do vetor diferença $c_{n+1}^{(m)} - c_{n+1}^{(m-1)}$;

início

sentinela = 0

```

# Atualiza o vetor peso no ponto  $c_n$ 

# Calcula a norma do ponto predito  $c_{n+1}^0$  armazenado-a em pnm

# Teste para avaliar nova Jacobiana DG

# Calcula o fator de aceleração

/* Correção */
faça
    ncor = 0
    enquanto ((ncor<4) e (sentinela = 0) )
        ncor = ncor + 1;
        NEWTON ( parâmetros ) ; /* função que resolve o sistema não linear
através do método de Newton */
        atualiza c, c'
         $d = \|c_{n+1}^{(m)} - c_{n+1}^{(m-1)}\|$ 
        se (d < tolerância*100*pnm)
            sentinela = 1 /* convergiu */
            nint = 11
        senão se ( ncor = 1)
            d0 = d
            se (100 * d0 < 0.33)
                sentinela = 1 /* convergiu */
                nint = 11
        senão
            Avalia a função F no ponto corrigido  $c_{n+1} = (cx, cy_{n+1,i})$ ,  $i = 0(1)q$ 
            repete a iteração novamente
        senão
            /* estima o raio de convergência para a iteração */
            ro = d / d0
             $ro = ncor - \sqrt[n]{ro}$ 
            se (ro > 0.9)
                se (Jacobiana é corrente)

```

```

ncor = 4 /* necessário reduzir a amplitude do passo */
senão /* necessário avaliar a Jacobiana DG e tentar o mét. de Newton
novamente */

    avalia DG no ponto predito  $c_{n+1}^0$ 
    sentinela = 0
    ncor = 0
    senão se  $((ro / (1 - ro)) * d < 0.33)$ 
        sentinela = 1 /* convergiu */
        nint = 11
        senão
            Avalia a função F no ponto corrigido  $c_{n+1} = (cx, cy_{n+1,i})$ ,
             $i = 0(1)q$ 
            repete a iteração novamente
        /* fim enquanto */
    /* testes de convergência */
    se (sentinela = 1)
        terminou a correção
        retorna sucesso

/* Newton não convergiu --> definir causa --> tomar a decisão adequada */
se (jacobiana não é corrente)
    avalia a jacobiana
    nint = nint + 1
    tentar a correção novamente
senão
    reduzir a amplitude do passo
    atualizar dados
    realizar nova predição com a nova amplitude
    Avalia a função F no ponto predito;
    # Atualiza o vetor peso no ponto  $c_n$ 
    # Calcula a norma do ponto predito  $c_{n+1}^0$ 

```

```

# Teste para avaliar nova Jacobiana DG
# Calcula o fator de aceleração
nint = nint + 1

```

```

enquanto (nint < 10 )
/* fim faça - enquanto */
se ( (nint > 10 ) e (sentinela = 0 ))
    retorna erro na convergência do método de Newton
fim da correção

```

6.3 Estratégias para Seleção de Ordem e Amplitude do Passo

Nesta subseção descreveremos os critérios utilizados para decidir quando aceitar ou não um passo, e as estratégias para seleção da amplitude bem como da ordem para o próximo passo.

6.3.1 Teste para Aceitação do Passo

Uma questão importante a respeito de um código para uma E.A.D. é como decidir a aceitar ou não um passo.

Estaremos interessados em dois tipos de erro, um é o erro de truncamento local do método $\left(\left\| F(x_n, y_n^0, y_n^1, \dots, y_n^q) \right\| \right)$. O coeficiente principal fixo da fórmula BDF aproxima $h c'$ com a fórmula:

$$h_{n+1} c'_{n+1} = \sum_{i=2}^{k+1} \alpha_{i-1} (n+1) \phi_i (n+1) - (\alpha_s - \alpha^0 (n+1)) \phi_{k+2} (n+1).$$

A solução verdadeira $c(s)$ satisfaz:

$$h_{n+1}c'(s_{n+1}) = \sum_{i=2}^{k+1} \alpha_{i-1}(n+1)\tilde{\phi}_i(n+1) - (\alpha_s - \alpha^0(n+1))\tilde{\phi}_{k+2}(n+1) + \tau_{n+1}$$

, onde τ_{n+1} é o erro de truncamento local, dado por:

$$\tau_{n+1} = (\alpha_{k+1}(n+1) + \alpha_s - \alpha^0(n+1))\tilde{\phi}_{k+2}(n+1) + \sum_{i=k+3}^{\infty} \alpha_{i-1}(n+1)\tilde{\phi}_i(n+1) \quad (6.10)$$

e $\tilde{\phi}_i(n+1)$ são as diferenças divididas definidas em (6.1), avaliadas na solução verdadeira $c(s)$.

O GSDAE estima o termo principal do erro de truncamento local (primeiro termo de (6.10)). A diferença dividida $\tilde{\phi}_{k+2}(n+1)$ em (6.10) é aproximada por $\phi_{k+2}(n+1)$, e este é dado pela diferença entre o preditor e o corretor. Para ver isto, seja $\chi_{n+1}^*(s)$ o único polinômio que interpola c_{n+1-i} , $i=0(1)k+1$. Então lembrando que o preditor interpola c_{n-i} , $i=0(1)k$, segue que, $\chi_{n+1}^*(s) - \chi_{n+1}^p(s) = \phi_{k+2}(n+1)$ em $s = s_{n+1}$, como $\chi_{n+1}^*(s_{n+1}) = c_{n+1}$ e $\chi_{n+1}^p(s_{n+1}) = c_{n+1}^0$ por definição, temos $c_{n+1} - c_{n+1}^0 = \phi_{k+2}(n+1)$.

Portanto, o termo principal do erro de truncamento local é estimado por:

$$(\alpha_{k+1}(n+1) + \alpha_s - \alpha^0(n+1)) \|c_{n+1} - c_{n+1}^0\| \quad (6.11).$$

O outro tipo de erro que é importante é o erro cometido na interpolação para encontrar a solução em s^* tal que $s_n < s^* < s_{n+1}$. No GSDAE a solução em s^* é dada pela avaliação do polinômio de grau k em s^* , (onde k é a ordem do método que foi utilizado para encontrar c_{n+1}) o qual interpola c nos últimos $k+1$ pontos $c_{n+1}, c_n, \dots, c_{n-k+1}$. O erro deste valor interpolado é composto de duas partes. A primeira é o "erro de interpolação", devido a interpolação da solução verdadeira $c(s)$ em $s_{n+1}, s_n, \dots, s_{n-k+1}$ por um polinômio. O termo principal do erro de interpolação é limitado por

$$\alpha_{k+1}(n+1) \|\phi_{k+2}(n+1)\|. \quad (6.12)$$

A segunda parte do erro é devido ao fato que estamos interpolando a solução computada ao invés da solução verdadeira. Este erro é proporcional ao erro da solução na malha de pontos, que está representado na relação (6.11).

Tomando (6.11) e (6.12) juntos, em todo passo exigimos que:

$$\text{ERR} = M \|c_{n+1} - c_{n+1}^0\| \leq 1.0, \quad (6.13)$$

onde

$$M = \max\left(\alpha_{k+1}(n+1), |\alpha_{k+1}(n+1) + \alpha_s - \alpha^0(n+1)|\right).$$

Se esta condição não é satisfeita, então o passo é rejeitado.

6.3.2 Critério para Determinar a Ordem do Próximo Passo

Se um passo é ou não rejeitado, o código deve decidir qual ordem para o método será utilizada no próximo passo. Desde que o preditor e o corretor são de mesma ordem, a ordem dos termos principais em sua correspondente expansão em série de Taylor são idênticos exceto pelas constantes do erro. O GSDAE estima os termos remanescentes da série de Taylor de ordens $k-2$, $k-1$ e k . Se os últimos $k+1$ passos têm sido tomados com ordem e amplitude constantes, então o termo de ordem $k+1$ também é estimado. Estas estimativas são:

$$\begin{aligned} \text{TERKM2} &= \|(k-1)\sigma_{k-1}(n+1)\phi_k(n+1)\| \approx \|h^{k-1}c^{(k-1)}\| \\ \text{TERKM1} &= \|k\sigma_k(n+1)\phi_{k+1}(n+1)\| \approx \|h^k c^{(k)}\| \\ \text{TERK} &= \|(k+1)\sigma_{k+1}(n+1)\phi_{k+2}(n+1)\| \approx \|h^{k+1}c^{(k+1)}\| \\ \text{TERKP1} &= \|(k+2)\sigma_{k+2}(n+1)\phi_{k+3}(n+1)\| \approx \|h^{k+2}c^{(k+2)}\| \end{aligned} \quad (6.14)$$

Desde que a estimativa de maior ordem TERKP1 é formada somente após $k+1$ passos com amplitude constante, $\sigma_{k+2}(n+1)$ é dado por: $\sigma_{k+2}(n+1) = 1/(k+2)$.

A estratégia para aumento ou diminuição da ordem é baseada em Shampine e Gordon [Shampine-75]. Há uma breve fase inicial onde a ordem e a amplitude são aumentados em todo passo, e depois a ordem é aumentada ou diminuída dependendo se TERKM2, TERKM1, TERK e TERKP1 formarem uma sequência decrescente ou não. A filosofia por trás desta estratégia é que a expansão em série de Taylor comporta-se como o esperado para ordens maiores somente se a magnitude dos sucessivos termos de ordens maiores formam uma sequência decrescente. Caso contrário, é mais seguro utilizar um método com ordem menor.

Algoritmo para aceitação do passo e para determinar a ordem a ser utilizada no próximo passo

início

knew = k

calcula a norma da diferença entre o ponto corrigido e o ponto predito

$$\text{Enorm} = \|c_{n+1} - c_{n+1}^0\|$$

/* cálculo dos termos TERK, TERKM1, TERKM2 */

termos (parametros)

/* cálculo de ERR */

se k = 1 então ERR = M * Enorm

senão

se (k > 2) então

se (max(TERKM1, TERKM2) < TERK) então knew = k - 1

EST = TERKM1 / k

senão se (TERKM1 < 0.5 * TERK) então knew = k - 1

EST = TERKM1 / k

ERR = M * Enorm

/* teste para aceitação do passo */

se (ERR > 1.0) então /* o passo não foi aceito */

```

incrementa o contador de falhas consecutivas
restaura os valores de c e de  $\phi$  para a nova ordem
k = knew
senão /* o passo foi aceito */
EST = TERK / k+1
se (knew = k - 1) então k = k - 1
EST = TERKM1 / k
senão
testa se k pode ser aumentado
caso afirmativo
constroe TERKP1
/* testes para aumentar ou diminuir a ordem */
se (k = 1) então f
se (TERKP1 <= 0.5 * TERK) então EST = TERKP1 / k+2
k = k+1
senão se (TERKM1 < min(TERK, TERKP1)) então EST = TERKM1 / k
k = k-1
senão se (TERKP1 <= (0.5 * TERK)) então EST = TERKP1 / k+2
k = k+1
/* realiza o up date das diferenças divididas */
update (parametros)
fim .

```

6.3.3 Critério para Determinar a Amplitude para o Próximo Passo

Após o passo ter sido aceito ou não e a ordem do método a ser utilizada no próximo passo ter sido determinada, o GSDAE decide qual será sua amplitude. A estratégia mais eficiente é a utilizada por Shampine e Gordon [Shampine-75].

O GSDAE escolhe a nova amplitude rh_{n+1} tal que o erro é aproximadamente metade da tolerância para o erro de integração. Portanto r é dado por:

$$r = (2.0EST)^{-1/(k+1)} \quad (6.15).$$

Onde EST é a estimativa do erro para o método de ordem k que foi selecionado para o próximo passo e é obtido pela estimativa dos termos em (6.14), mas escalados agora pela constante de erro $1/(k+1)$. Após um passo com sucesso, a amplitude é aumentada somente se pode ser dobrada. Se um decréscimo é necessário após um passo com sucesso, a amplitude é diminuída por no mínimo um fator $r = 0.9$, e no máximo por $r = 0.5$. Após um passo rejeitado, se for a primeira falha desde o último passo com sucesso, então r é determinado pela relação (6.15) e multiplicado por 0.9, e a amplitude do passo é reduzida por no mínimo $r = 0.9$ e no máximo por $r = 0.25$. Na segunda falha consecutiva, a amplitude é diminuída por um fator de um quarto. Após um grande decréscimo da amplitude do passo, a fórmula BDF com coeficientes principais fixos é mais precisa com ordem 1, assim após três falhas consecutivas, a ordem é reduzida a um, e a amplitude é reduzida por um fator de um quarto em toda falha dali em diante. Quando a amplitude é tão pequena que $s+h \approx s$, um retorno de erro é realizado.

Algoritmo para determinar a amplitude do passo (h_{n+1})

início

calcula r

se (passo foi aceito) **então** f

se ($r \geq 2.0$) **então** $h = 2.0 * h$ /* dobra o passo */

senão se ($r \leq 1.0$) **então** $r = \max(0.5, \min(0.9, r))$

$h = h * r$ /* passo é reduzido

por um fator de r */

senão /* o passo não foi aceito */

restaura ψ e s

se (contador de falhas consecutivas = 1) **então**

$$r = r * 0.9$$

$$r = \max(0.25, \min(0.9, r))$$

$$h = h * r$$

senão se (contador de falhas consecutivas = 2) então

$$h = h * 0.25$$

senão

$$k = 1$$

$$h = h * 0.25$$

fim.

6.4 Alguns Comentários sobre a Implementação

6.4.1 Norma Peso

A norma que o GSDAE utiliza é a ‘norma peso’.

$$\|v\| = \sqrt{\left(\frac{1}{NEQ}\right) \sum_{i=1}^{NEQ} \left(v_i / WT_i\right)^2}$$

onde

$WT_i = RTOL_i |u_i| + ATOL_i$, $RTOL_i$: é a tolerância para o erro relativo, $ATOL_i$: é a tolerância para o erro absoluto, u é o ponto anterior (c_n) e $NEQ = (q+1)m+1$.

Devemos lembrar que $c = (cx, cy_{i,j})$, $i=0(1)q$, $j=1(1)m$, logo, ao calcularmos a norma de c , devemos considerar c como um vetor de ordem NEQ e o vetor de pesos com uma componente correspondente a cx e outra a $cy_{i,j}$, isto é, WT_x e $WT_{y_{i,j}}$.

$$\text{Daí, temos que: } \|c\| = \left[\left(\frac{1}{NEQ}\right) \left(\left(\frac{cx}{WT_x}\right)^2 + \left(\sum_{i=0}^q \sum_{j=1}^m \left(\frac{cy_{i,j}}{WT_{y_{i,j}}}\right)^2 \right) \right) \right]^{\frac{1}{2}}$$

6.4.2 "UP DATE" das Diferenças Divididas

A cada passo o GSDAE atualiza as diferenças divididas, representadas por ϕ_i , cuja expressão é dada em (6.1). A relação que determina o "up date" é dada em Krogh [Krogh-72] por:

$$\phi_{i+1}(n+1) = \phi_i(n+1) - \beta_i \phi_i(n) \Rightarrow \phi_i(n+1) = \phi_{i+1}(n+1) + \phi_i^*(n) \quad (6.16)$$

Convém observarmos que ϕ_i^* é armazenado sobre ϕ_i .

Novamente, lembrando que $c = (c_x, c_{y_{i,j}})$, $i=0(1)q$, $j=1(1)n$, temos que $\phi_p = (\phi_{x_p}, \phi_{y_{p,i,j}})$, $p=1(1)k+3$, $i=0(1)q$, $j=1(1)n$, onde k é a ordem do método BDF utilizado.

Na subseção 6.3.1, vimos que $c_{n+1} - c_{n+1}^0 = \phi_{k+2}(n+1)$. No GSDAE, em cada passo a diferença $c_{n+1} - c_{n+1}^0$ é armazenada em $E = (E_x, E_{y_{i,j}})$, $i=0(1)q$, $j=1(1)n$, logo: $\phi_{x_{k+2}}(n+1) = E_x$, $\phi_{y_{k+2,i,j}}(n+1) = E_{y_{i,j}}$.

Daí, temos o seguinte algoritmo para o "up date" das diferenças divididas:
sejam:

kold: ordem anterior do método BDF;

kmax; ordem máxima permitida para o método BDF, que em nosso caso é $kmax = 5$.

início

se (kold < kmax) **então**

$$\phi_{x_{k+2}}(n+1) = E_x$$

para $i=0$ até q

para $j=1$ até n

$$\phi_{y_{k+2,i,j}}(n+1) = E_{y_{i,j}}$$

$$\phi_{x_{k+1}}(n+1) = \phi_{x_{k+1}}(n) + E_x$$

para $i=0$ até q

para $j=1$ até n

$$\phi_{y_{k+1,i,j}}(n+1) = \phi_{y_{k+1,i,j}}(n) + E_{y_{i,j}}$$

para $z=2$ até $k+1$

$$p = (k+1)-z+1$$

$$\phi_{x_p}(n+1) = \phi_{x_p}(n) + \phi_{x_{p+1}}(n+1)$$

para $i=0$ até q

para $j=1$ até n

$$\phi_{y_{p,i,j}}(n+1) = \phi_{y_{p,i,j}}(n) + \phi_{y_{p+1,i,j}}(n+1)$$

fim.

6.5 Processo de Inicialização

$$\text{Ao considerarmos o sistema} \quad \left\{ \begin{array}{l} F(c(s))=0 \\ w(c(s))c'(s)=0 \\ \|c'(s)\| - 1 = 0 \\ c(0)=c_0=(x_0, a_0, a_1, \dots, a_q) \end{array} \right. \quad (\text{ver cap.5, seção 2})$$

2), esbarramos no fato de não possuímos inicialmente uma aproximação para $c'(s)$.

Este problema é contornado ao resolvermos o p.v.i. para uma equação diferencial

$$\text{ordinária dada por: } \left\{ \begin{array}{l} c'(s) = \tau(A(c(s))) \\ c(0) = c_0 \end{array} \right. \quad (\text{cap.5, seção 2}).$$

O processo de inicialização resume-se no cálculo de $\tau(A)$.

cálculo de $\tau(A)$

O vetor $\tau(A)$ é obtido através da relação (5.2) e pela solução do sistema

$$B(c(s))\psi = \begin{pmatrix} F_x + F_{y_0}y_1 + \dots + F_{y_{q-1}}y_q & F_{y_q} \end{pmatrix} \begin{pmatrix} \tau_x \\ \tau_{y_q} \end{pmatrix} = 0, \text{ com um custo computacional}$$

da ordem de $\frac{2}{3}(m+1)^3$ (cap.5, seção 2).

Obtido $\tau(A)$, o GSDAE inicializa ϕ_2 como sendo $\tau(A)$ e assim pode iniciar o processo de integração com o polinômio preditor com $k = 1$, ou seja,

$$c_1^0 = \phi_1^* + \phi_2^* = c_0 + h_1 \tau(A(c_0)) \quad (\text{Euler Explícito}).$$

CAPÍTULO 7

Resultados Numéricos

7.1 Introdução

Apresentaremos neste capítulo alguns testes realizados com o GSDAE e com o DASSL, sendo que em alguns o teste também foi realizado utilizando o sistema (5.5) implementado no DASSL.

Denotaremos por **DASSL1** o código DASSL implementado na sua forma original e **DASSL2** à forma utilizando o sistema (5.5).

O código GSDAE permite integrar em x ou em s (comprimento de arco), em alguns exemplos integraremos em x e em outros em s , pois assim podemos fazer uso de um visualizador de curvas para visualizar a curva $c(s)$ gerada pelo GSDAE.

7.2 Resultados Numéricos

Teste-1 Equação da Esfera

O objetivo deste teste é ressaltar a eficiência do código GSDAE quando aplicado em um problema com singularidade.

Seja a E.A.D. não linear implícita :

$$x^2 + y^2 + y'^2 = 1 \quad (7.1)$$

com valores iniciais dados por: $\begin{cases} y(0) = 0 \\ y'(0) = 1 \end{cases}, (m=1, q=1)$

Resolveremos esta equação no intervalo $[0, 1]$ utilizando o GSDAE, DASSL1 e DASSL2 sendo que os resultados encontram-se respectivamente nas tabelas 7.1a, 7.1b e 7.1c.

No caso do GSDAE, primeiramente integraremos em x e constatamos que o método interrompeu a execução em $s_{\text{final}} = 1.575716049330584$, devido ao reconhecimento da singularidade, em seguida integraremos em s com o intuito de observarmos a curva $c(s)$ gerada pelo GSDAE.

x	y ₀₁	y ₁₁	RTOL	ATOL
0.797497923743	0.603318374352	0.002000198688	1.0×10^{-12}	1.0×10^{-15}

Tabela 7.1a Solução Equação da Esfera usando o GSDAE

Cuja estatística é:

402 → total de passos computados

1 → passos rejeitados

1 → falhas no Método de Newton

41 → avaliações da Jacobiana

731 → avaliações da Função

Para $t_0 = 0.0$ e $t_{\text{final}} = 1.00$ temos:

	y ₀₁	y ₁₁	RTOL	ATOL
x = 0.8	0.7173560909	0.6967067094	1.0×10^{-12}	1.0×10^{-15}
x=1.0	0.8414709848	0.5403023060	1.0×10^{-12}	1.0×10^{-15}

Tabela 7.1b Solução da Equação da Esfera usando o DASSL1

Cuja estatística é:

175 → total de passos computados

0 → passos rejeitados

0 → falhas no Método de Newton

47 → avaliações da Jacobiana

235 → avaliações da Função

Para os parâmetros $s_0 = 0.0$ e $s_{\text{final}} = 2.0$, temos:

	x	y ₀₁	y ₁₁	RTOL	ATOL
s = 1.574	0.7974987664	0.6033183759	0.00162939547	1.0×10^{-12}	1.0×10^{-15}
s = 1.576	0.7975003434	0.6033183777	-0.3706033772	1.0×10^{-12}	1.0×10^{-15}
s = 2.0	0.6624336561	0.6403460119	-0.3887655287	1.0×10^{-12}	1.0×10^{-15}

Tabela 7.1c Solução Equação da Esfera usando o DASSL2

Cuja estatística é:

818 → total de passos computados

5 → passos rejeitados

0 → falhas no Método de Newton

63 → avaliações da Jacobiana

1220 → avaliações da Função

Como podemos observar na tabela 7.1b o código DASSL1 não reconheceu a singularidade e assim gerou valores inconsistentes, já o GSDAE reconheceu a singularidade e como estávamos integrando em x houve a interrupção da execução do programa.

As figuras 7.1a, 7.1b, 7.1c e 7.1d correspondem às projeções de $c(s)$ nos planos xy, xy' , yy' e no plano xy com uma rotação de 10 graus.

Plano xy

figura 7.1a

Plano xy'

figura 7.1b

Plano yy'

figura 7.1c

Plano xy com rotação de 10 graus

figura 7.1d

Teste-2 Equação Exponencial

Seja a E.A.D. linear de índice zero

$$\begin{cases} y' = \lambda y \\ y(0) = 1 \\ y'(0) = \lambda y(0) \end{cases} \quad (7.2)$$

O objetivo deste teste é verificar a estabilidade do método GSDAE, para tal consideraremos o intervalo $[0, 0.5]$ com $\lambda = 1$ e $\lambda = 50$.

Resolveremos esta equação utilizando o GSDAE, DASSL1 e DASSL2 sendo que os resultados encontram-se respectivamente nas tabelas 7.2a e 7.2b.

$\lambda = 1$	GSDAE	DASSL1	DASSL2
Erro Relativo	4.003×10^{-10}	0.5×10^{-9}	-0.56×10^{-7}
tpas	84	59	101
rpas	0	2	0
fnewton	0	0	0
ajac	35	25	44
aF	145	86	125

Tabela 7.2a

$\lambda = 50$	GSDAE	DASSL1	DASSL2
Erro Relativo	0.254×10^{-6}	0.19×10^{-6}	0.254×10^{-6}
tpas	1637	833	1680
rpas	21	2	57
fnewton	20	0	0
ajac	162	25	146
aF	3426	1116	3014

Tabela 7.2b

Teste-3 E.D.O. de ordem 3

Seja a E.D.O. de ordem 3, com os seguintes valores iniciais:

$$\begin{cases} y''' + y' = 0 \\ y(0) = 0 \\ y'(0) = 1 \\ y''(0) = 2 \\ y'''(0) = -1 \end{cases} \quad (7.3)$$

Cuja solução é $y(x) = 2 - 2\cos(x) + \sin(x)$. Neste caso $m = 1$, $q = 3$.

Resolveremos esta equação utilizando o GSDAE e DASSL1 sendo que os resultados encontram-se respectivamente nas tabelas 7.3a e 7.3b. Devemos observar que para utilizarmos o DASSL1, devemos transformar esta equação em um sistema de equações diferenciais de primeira ordem (7.3b), ou seja; considerando $x_1 = y$, $x_2 = y'$ e $x_3 = y''$ temos $x'_1 = x_2$, $x'_2 = x_3$ e $x'_3 = -x_2$ daí

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \text{ com } \begin{cases} x_1(0) = 0 \\ x_2(0) = 1 \\ x_3(0) = 2 \end{cases} \text{ e } \begin{cases} x'_1(0) = 1 \\ x'_2(0) = 2 \\ x'_3(0) = -1 \end{cases} \quad (7.3b)$$

Agora temos $m = 3$, $q = 1$.

Para nossa análise consideraremos o seguinte intervalo: $[0, 1]$

$x = 1.00$
$y_{01} = 1.7608663744440427$
Erro Relativo: 0.0
$RTOL = 1.0 \times 10^{-9}$
$ATOL = 1.0 \times 10^{-15}$

Tabela 7.3a Solução Equação E.D.O. de ordem 3
usando o GSDAE

Cuja estatística é:

100 → total de passos computados

0 → passos rejeitados

0 → falhas no Método de Newton

36 → avaliações da Jacobiana

182 → avaliações da Função

$x = 1.0$
$y_{01} = 1.760866374$
Erro Relativo: -5.2247×10^{-10}
$RTOL = 1.0 \times 10^{-9}$
$ATOL = 1.0 \times 10^{-15}$

Tabela 7.3b Solução E.D.O. (7.3b) usando
DASSL1

Cuja estatística é:

106 → total de passos computados

0 → passos rejeitados

0 → falhas no Método de Newton

44 → avaliações da Jacobiana

151 → avaliações da Função

As figuras 7.2a e 7.2b correspondem às projeções de $c(s)$ nos planos xy e xy''' .

Plano xy

figura 7.2a

Plano xy'''

figura 7.2b

Teste-4 Equação de Van Der Pol

Seja a E.A.D. não linear semi explícita de índice zero:

$$\begin{cases} y_1' = y_2 \\ y_2' = \mu(1 - y_1^2)y_2 - y_1 \end{cases} \quad (7.4).$$

Neste caso $m = 2$, $q = 1$. A equação de Van Der Pol é uma das mais simples equações não lineares que descrevem um circuito elétrico. Esta equação é facilmente integrada para valores moderados de μ . Mas se, por exemplo considerarmos $\mu = 500$ o problema torna-se mais difícil [Hairer-91]. Logo temos um problema STIFF.

Consideraremos $\mu = 500$ e $\mu = 5000$, assim os valores iniciais são dados por:

$$\begin{cases} y_1(0) = 2 \\ y_2(0) = 0 \\ y_1'(0) = 0 \\ y_2'(0) = -2 \end{cases}$$

Resolveremos esta equação utilizando o GSDAE e DASSL1 sendo que os resultados encontram-se respectivamente nas tabelas 7.4a e 7.4b

	$\mu=500$	$\mu=5000$
x	1.0	1.0
y ₀₁	1.9986668139189576	1.9998666657909661
y ₀₂	-0.0013348155038424	-0.0001333481491012
RTOL:	1.0×10^{-5}	1.0×10^{-5}
ATOL:	1.0×10^{-15}	1.0×10^{-15}

Tabela 7.4a Solução Equação de Van Der Pol usando o GSDAE

Cujas estatísticas são:

$$\mu = 500$$

728 → total de passos computados

7 → passos rejeitados

166 → falhas no Método de Newton

546 → avaliações da Jacobiana

1789 → avaliações da Função

$$\mu = 5000$$

361 → total de passos computados

19 → passos rejeitados

10 → falhas no Método de Newton

112 → avaliações da Jacobiana

731 → avaliações da Função

	$\mu=500$	$\mu=5000$
x	1.0	1.0
y ₀₁	1.998666503	1.999866665
y ₀₂	-0.001334815979	-0.0001333481494
RTOL:	1.0×10^{-5}	1.0×10^{-5}
ATOL:	1.0×10^{-15}	1.0×10^{-15}

Tabela 7.4b Solução Equação de Van Der Pol usando o DASSL1

Cujas estatísticas são:

$$\mu = 500$$

116 → total de passos computados

0 → passos rejeitados

0 → falhas no Método de Newton

49 → avaliações da Jacobiana

154 → avaliações da Função

$$\mu = 5000$$

110 → total de passos computados

4 → passos rejeitados

0 → falhas no Método de Newton

51 → avaliações da Jacobiana

153 → avaliações da Função

Consideremos agora a equação de Van Der Pol como uma E.D.O. não linear de ordem 2, ou seja, $y'' = \mu(1 - y^2)y' - y$ ($m = 1$, $q = 2$) e utilizaremos o GSDAE para resolvê-la, sendo que os resultados encontram-se na tabela 7.4c. Novamente em nossos testes consideraremos $\mu = 500$ e $\mu = 5000$, assim os valores iniciais são dados por:

$$\begin{cases} y_1(0) = 2 \\ y_1'(0) = 0 \\ y_1''(0) = -2 \end{cases}$$

	$\mu=500$	$\mu=5000$
x	1.0	1.0
y ₀₁	1.9986668138402011	1.9998666651677752
y ₀₂	-0.0013348155039419	-0.0001333481492679
RTOL:	1.0×10^{-5}	1.0×10^{-5}
ATOL:	1.0×10^{-15}	1.0×10^{-15}

Tabela 7.4c Solução Equação de Van Der Pol ($m=1$, $q=2$) usando o GSDAE

Cujas estatísticas são:

$$\mu = 500$$

325 → total de passos computados

18 → passos rejeitados

13 → falhas no Método de Newton

90 → avaliações da Jacobiana

690 → avaliações da Função

$\mu = 5000$

339 → total de passos computados

26 → passos rejeitados

23 → falhas no Método de Newton

128 → avaliações da Jacobiana

729 → avaliações da Função

Teste-5 E.D.O. implícita

Seja a e.d.o. implícita com os seguintes valores iniciais:

$$\begin{cases} y \cos(t) + 2te^y + (\sin(t) + t^2 e^y + 2)y' = 0 \\ y(0) = 1 \\ y'(0) = -0.5 \end{cases} \quad (7.5)$$

Cuja solução exata é: $y \sin t + t^2 e^y + 2y = 2$.

Resolveremos esta equação no intervalo $[0, 1]$, utilizando o GSDAE e DASSL1 sendo que os resultados encontram-se respectivamente nas tabelas 7.5a e 7.5b.

$x = 1.00$
$y_{01} = 0.2513587857985173$
Erro Relativo: 0.8×10^{-8}
$RTOL = 1.0 \times 10^{-7}$
$ATOL = 1.0 \times 10^{-15}$

Tabela 7.5a E.D.O. implícita usando o GSDAE

Cuja estatística é:

138 → total de passos computados

0 → passos rejeitados

2 → falhas no Método de Newton

40 → avaliações da Jacobiana

289 → avaliações da Função

$x = 1.0$
$y_{01} = 0.2513585831$
Erro Relativo: -0.411×10^{-6}
$RTOL = 1.0 \times 10^{-7}$
$ATOL = 1.0 \times 10^{-15}$

Tabela 7.5b E.D.O. implícita usando o DASSL1

Cuja estatística é:

81 → total de passos computados

0 → passos rejeitados

0 → falhas no Método de Newton

17 → avaliações da Jacobiana

108 → avaliações da Função

Teste-6 E.A.D. algébrica

Seja a E.A.D. puramente algébrica :

$$\begin{cases} x^2 + y^2 = 1 \\ y(0) = 1 \end{cases} \quad (7.6)$$

Neste exemplo, gostaríamos de ressaltar que o DASSL em sua formulação original não resolveria esta equação, por esta razão vamos considerar apenas a resolução desta por meio do GSDAE no intervalo $[0, 2\pi]$, a curva $c(s)$ é dada na figura 3.

figura 3

Teste-7 Equação com posto menor do que m

Seja a E.A.D. na forma semi-implícita:

$$\begin{cases} x^2 + y_1'^2 = 1 \\ y_1^2 + y_2^2 = 0.25 \end{cases} \quad (7.7)$$

Condições iniciais:

$$\begin{cases} x_0 = 0 \\ y_1 = 0 \\ y_2 = 0.5 \\ y_1' = -1 \end{cases}$$

Ao integrarmos em x o GSDAE abortou a execução em $x = 0.525266497$ devido ao reconhecimento de uma singularidade, os resultados encontram-se na tabela 7.6

x	0.525266497
y ₀₁	-0.4999987300677280
y ₀₂	0.0011269106304962
y ₁₁	-0.8509377808681180
RTOL:	1.0×10^{-7}
ATOL:	1.0×10^{-15}

Tabela 7.6 Equação com posto menor do que m

Cuja estatística é:

108 → total de passos computados

3 → passos rejeitados

0 → falhas no Método de Newton

44 → avaliações da Jacobiana

205 → avaliações da Função

Ao integrarmos em s obtemos a curva $c(s)$ representada nas figuras 4a e 4b que correspondem as projeções de $c(s)$ nos planos xy_1' e y_1y_2 .

figura 4a

Plano xy_1

Plano y_1y_2

figura 4b

CONCLUSÃO

O código DASSL foi concebido para resolver E.A.D. de índices zero e um, sendo que próximo a singularidades pode gerar soluções errôneas. Em contra partida o GSDAE foi elaborado a partir da teoria apresentada no capítulo 4, sendo que a utilização do método BDF, bem como o critério para aceitação do passo e as técnicas para determinar a ordem do método e a amplitude do passo são baseadas nas utilizadas no DASSL. Portanto podemos considerar o GSDAE como uma extensão do DASSL para problemas com singularidades.

Nos testes apresentados no capítulo 7, não tínhamos a pretensão de comparar o GSDAE com o DASSL, uma vez que o DASSL é reconhecidamente eficiente quando aplicado aos problemas a que ele se destina, mas sim constatar esta eficiência e observar o comportamento do GSDAE nestes problemas e em exemplos que o DASSL não se aplica ou que gera soluções errôneas.

Podemos observar que o GSDAE é mais “caro”, quando aplicado a problemas com ausência de singularidades, apesar de utilizar os mesmos critérios de aceitação de

passo e técnicas de controle de ordem e amplitude do passo do DASSL. Este maior custo deve-se principalmente à parametrização por comprimento de arco utilizada no GSDAE, além é claro, do sistema considerado que como vimos possui dimensão $(q+1)m+1$, enquanto o DASSL trabalha com um sistema de dimensão m . Devemos considerar também a decomposição QR utilizada no método de Newton enquanto que o DASSL faz uso da decomposição LU.

No teste-1, utilizamos uma E.A.D. com singularidade, observamos que o GSDAE reconheceu a singularidade, enquanto o DASSL utilizando sua formulação original não a reconheceu e assim os valores obtidos daí em diante são inconsistentes, já ao utilizarmos o DASSL com o sistema (5.5), podemos observar que ele passou pela singularidade sem maiores problemas, ou seja, sem gerar instabilidades que afetassem a solução final. Devemos observar que no DASSL não há um controle para o reconhecimento da singularidade, por isso ao utilizá-lo com o sistema (5.5), ele passou pela singularidade.

No teste-6 utilizamos a equação da circunferência, que é uma equação puramente algébrica, onde constatamos que o GSDAE não teve problemas em resolvê-la já o DASSL em sua formulação original não se aplica a este tipo de problema ($q = 0$).

TRABALHOS FUTUROS

Ao considerarmos o sistema (5.5), o índice da E.A.D. é aumentado em pelo menos um, para ver isto considere $q = 1$ e assim para a E.A.D. original temos:

$$F(x, y, y') = 0. \quad (1)$$

Se $F_{y'}$ é não singular então seu índice é zero.

Seja $c = (x, y, y')$ então para o sistema (5.5) teremos:

$$G(c, c') = \begin{cases} \|c'\| - 1 = 0 \\ w(c)c' = 0 \\ F(c) = 0 \end{cases} = \begin{cases} h(c, c') = 0 \\ F(c) = 0 \end{cases} \quad (2)$$

$$\text{Logo, } \frac{\partial G}{\partial c'} = \begin{pmatrix} \frac{\partial h}{\partial c'} \\ 0 \end{pmatrix}, \text{ ou seja, } \frac{\partial G}{\partial c'} \text{ é singular.}$$

Na atual formulação o GSDAE limita-se a sistemas de índice 0. No capítulo 4, seção 4.5, vimos que é possível resolver equações com posto menor que m . Ou seja, podemos obter permutações de linhas e colunas tal que:

$$P \frac{\partial G}{\partial c'} U = QR \quad (3)$$

com $R = \begin{pmatrix} R_1 & R_2 \\ 0 & 0 \end{pmatrix}$ onde R_1 é triangular superior com dimensão $p \leq m$.

Caso $p < m$, no ponto inicial (2) pode ser escrito como $\begin{cases} f(x, z, w, z', w') = 0 \\ g(x, z, w) = 0 \end{cases}$

(4) onde $y = (z, w)$ e $c = (x, z, w, z', w')$.

Na atual formulação o GSDAE resolve sistemas de índice um na forma :

$$\begin{cases} f(x, z, w, z', w') = 0 \\ g(x, z, w) = 0 \end{cases}$$

Numa implementação futura pretendemos fazer uso desses resultados, a fim de que o usuário não precise saber nenhuma informação a respeito do posto da equação, ou seja, bastará fornecer a equação na forma implícita (1) e o código obterá o posto da equação e assim a integrará na forma (4).

BIBLIOGRAFIA

- [Aitken-94] Aitken,P., Jones,B. *Guia do Programador C*, Berkley,1994.
- [Allgower-90] Allgower,E.L., Georg,K. *Numerical Continuation Methods - An Introduction*, Springer-Verlag, 1990.
- [Brenan-89] Brenan,K.E., Campbell,S.L., Petzold,L.R. *Numerical Solution of Initial Value Problems in Differential-Algebraic Equations*, North-Holland, 1989.
- [Campbell-82] Campbell,S.L. *Singular Systems of Differential Equations II*, Pitman, 1982.
- [Castelo-94] Castelo,A.F., Tavares,G. *Differential-Algebraic Equations with Singularities*, 1994, Relatório Técnico Departamento de Matemática - PUC-RIO, MAT 24/95.
- [Gear-71a] Gear,C.W., *The Simultaneous Numerical Solution of Differential-Algebraic Equations*, IEEE Trans. Circuit Theory, ct-18 (1971), 89-95.
- [Gear-71b] Gear,C.W., *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, 1971.
- [Hairer-91] Hairer, E. & Wanner, G. *Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems*, Springer-Verlag, 1991.
- [Hairer-93] Hairer,E., Norsett,S.P., Wanner, G. *Solving Ordinary Differential Equations I - Nonstiff Problems*, Springer-Verlag, 1993.

[Hindmarsh-80] Hindmarsh,A.C., *LSODE and LSODI, Two New Initial Value Ordinary Differential Equations Solvers*, ACM-SIGNUM Newsletters, 15 (1980), 10-11.

[Hindmarsh-83] Hindmarsh,A.C., *ODEPACK , A Systematized Collection of O.D.E. Solvers in Scientific Computing*, eds. R.S. Stepleman et. al., North-Holland, Amsterdam, 1983, 55-64.

[Jackson-Sacks-Davis] Jackson, K.R., Sacks-Davis,R., *An Alternative Implementation of Variable Step-Size Multistep Formulas for Stiff ODEs*, ACM Transactions on Mathematical Software, vol 6, número 3, september 1980, pages 295-318.

[Krogh-72] Krogh,Fred,T. ,*Changing Stepsize in the Integration of Differential Equations Using Modified Divided Differences*, Proceedings of the Conference on the Numerical Solution of Differential Equations, 19,20,(October-1972), The University of Texas at Austin, p 22-71.

[Lambert-73] Lambert,J.D., *Computational Methods in Ordinary Differential Equations*, John Wiley & Sons, 1973.

[Luenberger-77] Luenberger,D.G., *Dynamic Equations in Descriptor Form*, IEEE Trans. Aut. Control, AC-22 (1970), 312-321 .

[Petzold-82] Petzold,L.R. *Differential / Algebraic Equations are not O.D.Es*, SIAM J. Sci. Stat. Comput. 3 (1982), 367-384.

[Pires-94] Pires,A.A. *Métodos Numéricos para a Solução de Equações Algébrico-Diferenciais*, Tese de Mestrado apresentada junto ao Instituto de Ciências Matemáticas de São Carlos, da Universidade de São Paulo, São Carlos, 1994.

[Romeiro-94] Romeiro, N.M.L. *Análise Comparativa de Métodos Numéricos para Equações Algébrico-Diferenciais*, Tese de Mestrado apresentada junto ao Instituto de Ciências Matemáticas de São Carlos, da Universidade de São Paulo, São Carlos, 1994.

[Shampine-75] Shampine, L.F., Gordon, M.K. *Computer Solution of Ordinary Differential Equations - The Initial Value Problem*, W.H. Freeman and Company, 1975.

[Silverman-69] Silverman, L.M. *Inversion of Multivariable Systems*, IEEE Trans. Aut. Control, AC-14 (1969), 270-276.

[Skelboe-77] Skelboe, S. *The Control of Order and Steplenght for Backward Differentiation Methods*, BIT, 17, 1977, 91-107.