UNIVERSIDADE FEDERAL DA PARAIBA

CENTRO DE CIÊNCIAS E TECNOLOGIA

COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Rosana Marques da Silva

UM ESTUDO DE SOLUÇÃO NUMÉRICA DE PROBLEMAS DE VALOR DE CONTORNO PARA EQUAÇÕES DIFERENCIAIS ORDINÁRIAS

Mário Toyotaro Hattori
Orientador

Campina Grande
maio - 1992

Rosana Marques da Silva

UM ESTUDO DE SOLUÇÃO NUMÉRICA DE PROBLEMAS DE VALOR DE CONTORNO PARA EQUAÇÕES DIFERENCIAIS ORDINÁRIAS

Dissertação apresentada ao curso de mestrado em Informática da Universidade Federal da Paraiba, em cumprimento às exigências para obtenção do grau de mestre.

Mário Toyotaro Hattori
Orientador

Campina Grande
maio - 1992

S586e Silva, Rosana Marques da.
Um estudo de solução numérica de problemas de valor de contorno para equações diferenciais ordinárias / Rosana Marques da Silva. - Campina Grande, 1992.
98 f.

Dissertação (Mestrado em Informática) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, 1992.
"Orientação : Prof. Mário Toyotaro Hattori".
Referências.

1. Programa de Computador. 2. Solução Numérica - Estudo. 3. Equações Diferenciais Ordinárias. 4. Dissertação - Informática. I. Hattori, Mário Toyotaro. II. Universidade Federal da Paraíba - Campina Grande (PB). III. Título

CDU 004.42:517.91(043)

UM ESTUDO DE SOLUÇÃO NUMÉRICA DE PROBLEMAS DE VALOR DE CONTORNO PARA EQUAÇÕES DIFERENCIAIS ORDINÁRIAS

Rosana Marques da Silva

Dissertação aprovada em 12/05/1992

Mário Toyotaro Hattori
Orientador

Aparecido Jesuino de Souza
Componente da banca

Bruno Correia da Nóbrega Queiroz
Componente da banca

Campina Grande
maio - 1992

Para duas pessoas maravilhosas,
meu marido: Francisco Antonio e
meu filho: Ricardo.

AGRADECIMENTOS

Ao Prof. Mário Toyotaro Hattori pela sua segura orientação.

A Francisco Antonio Morais de Souza pelo seu apoio, compreensão e incentivo em todos os momentos.

Aos colegas do Departamento de Matemática e Estatística pela oportunidade de qualificação. Em especial àqueles que colaboraram, esclarecendo dúvidas e/ou dando sugestões, durante a realização deste trabalho.

A Ana Lúcia Guimarães pelo seu apoio.

Aos membros da banca examinadora pela participação e interesse.

RESUMO

A primeira finalidade deste trabalho é apresentar um estudo de métodos para solução numérica de problemas de valor de contorno para equações diferenciais ordinárias (PVC/EDO). Foram estudados os métodos do *shooting* simples e múltiplo, implícito de Runge-Kutta, da colocação e dos elementos finitos.

A segunda finalidade é apresentar uma breve avaliação do desempenho desses métodos quando implementados em programas de computador. Para este fim, os métodos foram codificados e os resultados numéricos obtidos por esses códigos comparados com aqueles obtidos pelos pacotes MUS, COLSYS e COLNEW, que são implementações dos métodos do *shooting* múltiplo, da colocação com B-splines e com bases monomiais, respectivamante.

Os pacotes MUS e COLNEW foram os melhores dentre os disponíveis. O pacote COLSYS falhou na resolução de problemas não lineares.

ABSTRACT

The first purpose of this work is to present a review of methods for numerical solution of boundary-value problems in ordinary differential equations (BVP/ODE). The shooting and multiple shooting, implicit Runge-Kutta, collocation, and finite elements methods are reviewed.

The second purpose is to briefly present an assessment of performance of the methods when implemented as computer programs. For this end, the methods have been coded and numerical results delivered by codes are compared with those obtained by the packages MUS, COLSYS, and COLNEW, which implemented multiple shooting method, collocation method with B-splines, and collocation method with monomial bases, respectively.

The codes MUS and COLNEW are the best available. The COLSYS package failed to solve nonlinear problems.

CONTEÚDO

1 - INTRODUÇÃO

Muitos problemas em engenharia, quando formulados matematicamente, requerem a determinação de uma função que satisfaça a um problema de valor de contorno para equações diferenciais ordinárias (PVC/EDO). Por exemplo: problemas de transferência de calor como o ocorrido na injeção de fluidos através de um dos lados de um longo tubo vertical ou da reentrada de veículos espaciais, onde uma função do tempo deve ser escolhida para minimizar o aquecimento que o veículo experimenta durante a reentrada na atmosfera; problemas da deformação eletrostática sofrida por uma cápsula esférica, delgada, sujeita a uma carga; problemas relacionados com abalos sísmicos; problemas relacionados com alguns modelos epidemiológicos. Muitos desses problemas têm uma formulação original como uma equação diferencial parcial, que, com a aplicação de várias técnicas, são transformados em EDOs.

Do ponto de vista de um pesquisador ou engenheiro a solução de um PVC/EDO é apenas parte da solução de um problema maior, e um software constitui uma ferramenta de seu trabalho. Como ferramenta, é preciso saber quando e como utilizá-lo, os cuidados necessários para utilizá-lo e conhecer suas limitações. É uma das finalidades deste trabalho encontrar respostas a essas questões através de um estudo de métodos numéricos e do software existente.

1.1 - A Classe de Problemas

Seja o sistema de ordem mista de d equações diferenciais ordinárias de ordem

$$1 \leq m_1 \leq \cdots \leq m_d$$

$$\text{(1.1)} \qquad \begin{cases} u^{(m_1)} = F_1(x, \mathbf{y}(\mathbf{u}(x))), \\ u^{(m_2)} = F_2(x, \mathbf{y}(\mathbf{u}(x))), \qquad a < x < b, \\ \quad \cdots \\ u^{(m_d)} = F_d(x, \mathbf{y}(\mathbf{u}(x))), \end{cases}$$

onde $n = \sum_{i=1}^{d} m_i$ é o numero de variáveis,

$F_i : \mathbb{R}^{n+1} \longrightarrow \mathbb{R}$,

$\mathbf{u}(x) = [u_1(x) \ \ldots \ u_d(x)]^t$ é uma solução do sistema,

$\mathbf{y}(\mathbf{u}(x)) = [u_1(x) \ u_1'(x) \ \ldots \ u_1^{(m_1-1)}(x) \ u_2(x) \ \ldots \ u_d^{(m_d-1)}(x)]^t \in \mathbb{R}^n$.

Se as funções F_i dependerem de $\mathbf{u}$ e de suas derivadas, a EDO será não linear, caso contrário será linear.

A conversão do sistema (1.1) em um sistema de n equações de primeira ordem com n variáveis, resulta em

$$\mathbf{y}' = \mathbf{f}(x; \mathbf{y}), \qquad a < x < b, \tag{1.2}$$

onde $\mathbf{y} = \mathbf{y}(\mathbf{u}(x))$,

$\mathbf{f}(x, \mathbf{y}) = [u_1'(x) \ \ldots \ F_1(x, \mathbf{y}(\mathbf{u}(x))) \ u_2'(x) \ \ldots \ F_d(x, \mathbf{y}(\mathbf{u}(x)))]^t \in \mathbb{R}^n$,

ou na sua forma matricial, no caso de $\mathbf{f}$ ser linear

$$\mathbf{y}' = A(x)\mathbf{y} + \mathbf{q}(x), \qquad a < x < b, \tag{1.3}$$

onde $A \in \mathbb{R}^{n \times n}$ e $\mathbf{q} \in \mathbb{R}^n$.

As condições de contorno para PVC/EDO de primeira ordem, de uma forma geral são dadas por

$$\mathbf{g}(\mathbf{y}(a), \mathbf{y}(b)) = 0, \tag{1.4}$$

onde $\mathbf{g} = [g_1 \ \ldots \ g_n]^t$ e $g_i : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$.

Se a função $\mathbf{g}$ for linear, as condições de contorno podem ser colocadas na forma matricial

$$B_a\mathbf{y}(a) + B_b\mathbf{y}(b) = \mathbf{d}, \tag{1.5}$$

onde B_a e $B_b \in \mathbb{R}^{n \times n}$ e $\mathbf{d} \in \mathbb{R}^n$.

Se algumas das condições de contorno fornecerem informações a apenas um dos pontos, são denominadas parcialmente separáveis ou separáveis, no caso linear, podem ser representadas respectivamente por

$$B_{a1}\mathbf{y}(a) = \mathbf{d}_1, \qquad \text{e} \qquad B_{a2}\mathbf{y}(a) + B_{b2}\mathbf{y}(b) = \mathbf{d}_2,$$

ou

$$B_{a1}\mathbf{y}(a) = \mathbf{d}_1 \qquad \text{e} \qquad B_{b2}\mathbf{y}(b) = \mathbf{d}_2,$$

onde $B_{a1} \in \mathbb{R}^{(n-v)\times n}$, $\mathbf{d}_1 \in \mathbb{R}^{n-v}$, B_{a2} e $B_{b2} \in \mathbb{R}^{v\times v}$ e $\mathbf{d}_2 \in \mathbb{R}^v$, para $0 < v < n$. Analogamente para o caso não linear.

EXEMPLO 1.1 - Considere o sistema de PVC/EDO linear de primeira ordem

$$\begin{cases} y_1' = (1 - 19\cos 2x)y_1 + (1 + 19\,\mathrm{sen}\,2x)y_3 + e^x(-1 + 19\cos 2x - 19\,\mathrm{sen}\,2x) \\ y_2' = 19\,y_2 - e^x \\ y_3' = (-1 + 19\,\mathrm{sen}\,2x)y_1 + (1 + 19\cos 2x)y_3 + e^x(1 - 19\cos 2x - 19\,\mathrm{sen}\,2x), \end{cases}$$

para $0 < x < b$ e com as condições de contorno

$$\begin{aligned} y_1(0) + y_1(b) &= 1 + e^b \\ y_2(0) + y_2(b) &= 1 + e^b \\ y_3(0) + y_3(b) &= 1 + e^b \end{aligned}$$

que podem ser representadas por

$$\mathbf{g}(\mathbf{y}(0), \mathbf{y}(b)) = \begin{bmatrix} y_1(0) + y_1(b) \\ y_2(0) + y_2(b) \\ y_3(0) + y_3(b) \end{bmatrix} = \begin{bmatrix} 1 + e^b \\ 1 + e^b \\ 1 + e^b \end{bmatrix}.$$

Na forma matricial tem-se

$$\mathbf{y}'(x) = A(x)\,\mathbf{y}(x) + q(x), \quad 0 < x < b,$$

onde

$$A(x) = \begin{bmatrix} 1 - 19\cos 2x & 0 & 1 + 19\,\mathrm{sen}\,2x \\ 0 & 18 & 0 \\ -1 + 19\,\mathrm{sen}\,2x & 0 & 1 + 19\cos 2x \end{bmatrix}, \; q(x) = \begin{bmatrix} (-1 + 19\cos 2x - 19\mathrm{sen}\,2x)\,\mathrm{e}^x \\ -18\mathrm{e}^x \\ (1 - 19\cos 2x - 19\,\mathrm{sen}\,2x)\,\mathrm{e}^x \end{bmatrix},$$

e as condições de contorno $B_a\,\mathbf{y}(0) + B_b\,\mathbf{y}(b) = \mathbf{d}$, onde

$$B_a = B_b = I \qquad e \qquad \mathbf{d} = \begin{bmatrix} 1 + \mathrm{e}^b \\ 1 + \mathrm{e}^b \\ 1 + \mathrm{e}^b \end{bmatrix},$$

Neste exemplo, o número de equações é $d = 3$, a ordem de cada uma das equações é $m_1 = m_2 = m_3 = 1$, o sistema possui $n = \sum_{i=1}^{d} m_i = 3$ equações e $n = 3$ variáveis, a solução é representada pelo vetor $\mathbf{y}(x) = [y_1(x)\ y_2(x)\ y_3(x)]^t$. $\triangle$

EXEMPLO 1.2 - Considere agora o PVC não linear

$$\begin{cases} u'' + e^u = 0, & 0 < x < 1, \\ u(0) = 0, \\ u(1) = 0, \end{cases}$$

onde o número de equações é $d = 1$, a ordem da equação $m_1 = 2$, $n = m_1 = 2$, o vetor $\mathbf{y}(u(x)) = [u(x)\ u'(x)]^t$, a função $\mathbf{f}(x, \mathbf{y})$ de (1.2) é dada por $\mathbf{f}(x, \mathbf{y}) = [u'(x)\ F_1(x, \mathbf{y})]^t$, onde $F_1 = -e^u$. As condições de contorno são dadas por $g_1(\mathbf{y}(u(0)) = y_1(0)$ e $g_2(\mathbf{y}(u(1)) = y_2(1)$. $\triangle$

OBSERVAÇÃO. : *As condições de contorno podem se referir a mais de dois pontos do domínio do problema. Neste caso o problema será denominado PVC multi-ponto e as condições de contorno, em J pontos, terão a forma geral*

$$\sum_{j=1}^{J} B_j \mathbf{y}(\xi_j) = \mathbf{d}, \qquad a = \xi_1 \leq \xi_2 \leq \cdots \leq \xi_J = b,$$

se forem lineares e

$$\mathbf{g}_j(\mathbf{y}(\mathbf{u}(\xi_j))) = 0_j, \qquad 1 \leq j \leq J,$$

se forem não lineares e separáveis.

Um PVC multi-pontos pode ser convertido em um PVC 2 pontos pela transformação dos subintervalos $[\xi_i, \xi_{i+1}]$ *em* $[0, 1]$, *através da mudança de variável,* $t = \frac{x - \xi_j}{\xi_{j+1} - \xi_j}$, *para cada* j, *conforme Ascher et al.[1988].*

1.2 - SOLUÇÃO DE PVI e PVC

Se à EDO (1.1) for atribuída uma condição inicial $\mathbf{u}(a) = \mathbf{u}_0$ tem-se um problema de valor inicial, PVI, de ordem superior. O sistema de primeira ordem equivalente tem a forma

$$\begin{cases} \mathbf{y}' = \mathbf{f}(x, \mathbf{y}), & x > a, \\ \mathbf{y}(a) = \mathbf{y}_o, \end{cases}$$

para $\mathbf{y}$ definido em (1.1).

Os problemas de valor inicial podem ser considerados uma subclasse especial e mais simples dos problemas de valor de contorno (PVC). Para PVI/EDO existem resultados

teóricos garantindo a existência e unicidade de solução (o que é fundamental para desenvolver métodos numéricos) assim como códigos de finalidade geral que resolvem eficientemente muitos PVIs que aparecem na prática.

Em geral é extremamente difícil estabelecer explicitamente a existência e a unicidade da solução de PVC/EDO. Entretanto, é possível extrair alguns resultados expressando o PVC em termos de PVIs associados, aproveitando com isso, a teoria já desenvolvida.

1.5 - Organização

O estudo de métodos númericos para PVC/EDO exige um razoável conhecimento de equações diferenciais ordinárias, álgebra linear e análise numérica. Nos capítulos 1 e 2 será introduzido o material necessário para o desenvolvimento e entendimento dos métodos apresentados neste trabalho.

O capítulo 2 - Fundamentos Matemáticos - apresenta resultados teóricos sobre a existência, unicidade e estabilidade da solução de PVC/EDO de primeira ordem. Esse estudo é suficientemente abrangente uma vez que toda EDO de ordem superior (1.1) pode ser transformada em um sistema de equações de primeira ordem (1.2), assim como todo PVC multi-ponto pode ser transformado em um PVC de 2 pontos com condições de contorno separáveis.

O capítulo 3 - Solução Numérica de PVC/EDO - apresenta um estudo dos seguintes métodos: métodos do Valor Inicial, *shootings* simples e múltiplo, para problemas de primeira ordem; método das diferenças finitas de passo simples, usando as fórmulas implícitas de Runge-Kutta para sistemas de primeira ordem; o método da colocação, como um método das diferenças finitas para PVC de primeira ordem; O método da colocação para PVC de ordem superior usando bases monomiais, e B-splines ; e o método dos elementos finitos, usando funções lineares e splines cúbicos, para problemas de segunda ordem.

O capítulo 4 - Implementações - apresenta detalhes das implementações dos seguintes pacotes: MUS - software para resolução de sistemas de PVC de primeira ordem usando o método de superposição com *shooting* múltiplo usando reortogonalização com técnica de marcha e compactação; COLSYS e COLNEW - dois pacotes para resolução de problemas de ordem superior usando o método da colocação com pontos Gaussianos, B-splines como bases e bases Monomiais, respectivamente.

Capítulo 5 - Exemplos numéricos e Conclusões - apresenta problemas lineares e não lineares e os resultados obtidos usando as implementações descritas no capítulo 4.

2 - FUNDAMENTOS MATEMÁTICOS

2.1 - Existência de Solução para PVC/EDO

Os resultados sobre existência de solução de PVC, serão dados em termos de solução fundamental de PVIs associados.

Seja o PVC de primeira ordem

$$\text{(2.1a)} \qquad \begin{cases} \mathbf{y}' = \mathbf{f}(x, \mathbf{y}), & a < x < b, \\ \mathbf{g}(\mathbf{y}(a), \mathbf{y}(b)) = 0. \end{cases}$$

Dado $\mathbf{s} \in I\!R^n$, considere o PVI associado

$$\text{(2.1b)} \qquad \begin{cases} \mathbf{w}' = \mathbf{f}(x, \mathbf{w}), & a < x < b, \\ \mathbf{w}(a; \mathbf{s}) = \mathbf{s}. \end{cases}$$

Para cada $\mathbf{s}$, se a função $\mathbf{f}$ for lipschitziana em um domínio $D = \{(x, \mathbf{w}) : a \leq x \leq b, ||\mathbf{w} - \mathbf{s}|| \leq \rho\}$, para algum $\rho > 0$ e M é tal que $||\mathbf{f}(x, \mathbf{w})|| \leq M$ para todo $(x, \mathbf{w}) \in D$. Então o problema (2.1b) possui uma única solução, do tipo

$$\mathbf{w}(x; \mathbf{s}) = [w_1(x; \mathbf{s}) \; w_2(x; \mathbf{s}) \; \ldots \; w_n(x; \mathbf{s})]^t,$$

em $a < x < a + c$, onde $c = \min \{b - a, \rho/M\}$.

Se for possivel escolher $\mathbf{s}$ de forma que $\mathbf{g}(\mathbf{s}, \mathbf{w}(b; \mathbf{s})) = 0$, então $\mathbf{y}(x) = \mathbf{w}(x; \mathbf{s})$ é solução de (2.1a). O problema (2.1a) poderá ter muitas soluções dadas pelas raízes distintas $\mathbf{s}^*$ de $\mathbf{g}(\mathbf{s}^*, \mathbf{w}(b; \mathbf{s}^*)) = 0$. Neste caso, as soluções do PVC serão do tipo

$$\mathbf{y}(\cdot) = \mathbf{w}(\cdot, \mathbf{s}^*).$$

Considerando o PVC com $\mathbf{f}$ e $\mathbf{g}$ não lineares

$$\text{(2.2a)} \qquad \begin{cases} \mathbf{y}' = \mathbf{f}(x, \mathbf{y}), & \text{para} \quad a < x < b, \\ \mathbf{g}(\mathbf{y}(a), \mathbf{y}(b)) = 0, \end{cases}$$

geralmente, não se conhece quantas soluções este PVC possui, logo, uma propriedade importante na solução $\mathbf{y}(\cdot)$ de (2.2a) é a unicidade local. Para estabelecer esta unicidade, considera-se o problema linearizado[1] associado

$$\text{(2.2b)} \qquad \begin{cases} \mathbf{z}' = A(x)\mathbf{z}, & \text{para} \quad a < x < b, \\ B_a\mathbf{z}(a) + B_b\mathbf{z}(b) = 0, \end{cases}$$

onde

$$A = \frac{\partial \mathbf{f}(x, \mathbf{y}(x))}{\partial \mathbf{y}} \text{ com elementos } a_{ij} = \frac{\partial f_i(x, \mathbf{y}(x))}{\partial y_j},$$

$$B_a = \frac{\partial \mathbf{g}(\mathbf{y}(a), \mathbf{y}(b))}{\partial \mathbf{y}(a)} \text{ com elementos } b_{aij} = \frac{\partial g_i(\mathbf{y}(a), \mathbf{y}(b))}{\partial y_j(a)},$$

$$B_b = \frac{\partial \mathbf{g}(\mathbf{y}(a), \mathbf{y}(b))}{\partial \mathbf{y}(b)} \text{ com elementos } b_{bij} = \frac{\partial g_i(\mathbf{y}(a), \mathbf{y}(b))}{\partial y_j(b)},$$

$$1 \leq i, j \leq n \quad \text{e}$$

$\mathbf{z}(x) = \mathbf{y}(x) - \widehat{\mathbf{y}}(x)$, $\widehat{\mathbf{y}}(x)$ é uma solução aproximada de (2.2a) com $\widehat{\mathbf{y}}(a) \approx \mathbf{y}(a)$.

Uma solução $\mathbf{y}$ de um PVC é dita isolada ou localmente única se existir uma região na qual ela seja única, isto é, se existir $\rho > 0$ tal que para todo $\widehat{\mathbf{y}}(x)$ no espaço de soluções satisfazendo $\sup_{a \leq x \leq b} ||\widehat{\mathbf{y}}(x) - \mathbf{y}(x)||_\infty \leq \rho$, $\mathbf{y} = \widehat{\mathbf{y}}$, ou seja, $\mathbf{y}$ é a única solução do PVC nessa região. A solução $\mathbf{y}$ do PVC (2.2a) será isolada se o problema linearizado possuir uma única solução ($\mathbf{z} \equiv 0$). Em termos do PVI associado, significa que $\mathbf{s}^*$ é uma raiz simples de $\mathbf{g}(\mathbf{s}, \mathbf{w}(b; \mathbf{s})) = 0$ e $\mathbf{y}(\cdot) = \mathbf{w}(\cdot; \mathbf{s}^*)$ é uma solução isolada conforme Ascher et al.[1988].

Considerando agora o PVC/EDO linear

$$\text{(2.3a)} \qquad \begin{cases} \mathbf{y}' = A(x)\mathbf{y} + \mathbf{q}(x), & \text{se} \quad a < x < b, \\ B_a\mathbf{y}(a) + B_b\mathbf{y}(b) = \mathbf{d}, \end{cases}$$

O PVI associado (2.1b) possui uma única solução se A e $\mathbf{q}$ forem contínuas e será do tipo

$$\mathbf{w}(x; \mathbf{s}) = W(x; \mathbf{s})W^{-1}(a; \mathbf{s})\mathbf{s} + \int_a^x W(x; \mathbf{s})W^{-1}(t; \mathbf{s})\mathbf{q}(t)\, dt$$

[1]Expandindo $\mathbf{f}(x, \widehat{\mathbf{y}}(x))$ em série de Taylor em torno da solução exata $\mathbf{y}(x)$, obtem-se $\mathbf{f}(x, \widehat{\mathbf{y}}(x)) = \mathbf{f}(x, \mathbf{y}(x)) + \frac{\partial \mathbf{f}(x, \mathbf{y}(x))}{\partial \mathbf{y}}(\widehat{\mathbf{y}}(x) - \mathbf{y}(x)) + \mathbf{r}(x, \mathbf{y}, \widehat{\mathbf{y}})$, com $\mathbf{r}(x, \mathbf{y}, \widehat{\mathbf{y}}) = O(||\mathbf{z}(x)||^2)$. Ignorando $\mathbf{r}$, encontra-se a equação linearizada $\mathbf{z}' = A(x)\,\mathbf{z}$.

onde W é uma solução fundamental de $\mathbf{w}' = A(x)\,\mathbf{w}^2$, ou seja, W é solução do problema homogêneo

$$W' = A(x)W,$$

conforme Ascher et al.[1988], Coddington e Levinson [1955], Soutomayor [1979].

Tomando $Y(x) = W(x;\mathbf{s})$, tem-se

$$\mathbf{y}(x) = Y(x)Y^{-1}(a)\mathbf{s} + \int_a^x Y(x)Y^{-1}(t)\mathbf{q}(t)\,dt \tag{2.3b}$$

que é a única solução do PVC (2.3a) se e somente se a matriz $Q = B_aY(a) + B_bY(b)$ for não singular para $a \leq x \leq b$. (Significando a existência de um único $\mathbf{s}^*$ raiz de $\mathbf{g}(\mathbf{s}, \mathbf{w}(b,\mathbf{s})) = 0$) conforme Ascher et al. [1988].

Considerando a solução fundamantal $Y(x)$, tal que $Y(a) = I$, ou seja, $Y(x)$ é solução do PVI

$$\begin{cases} Y' = A(x)Y, \\ Y(a) = I, \end{cases}$$

a solução (2.3b) pode ser reescrita na forma

$$\mathbf{y}(x) = Y(x)\mathbf{s} + \mathbf{v}_p$$

onde

$$\mathbf{s} = Q^{-1}(\mathbf{d} - B_bY(b)\int_a^b Y^{-1}(t)\mathbf{q}(t)\,dt) \qquad \text{e}$$

$\mathbf{v}_p = Y(x)\int_a^x Y^{-1}(t)\mathbf{q}(t)\,dt$ é a solução do PVI não homogêneo $\begin{cases} \mathbf{v}' = A(x)\mathbf{v} + \mathbf{q}(x), \\ \mathbf{v}(a) = 0. \end{cases}$

EXEMPLO 2.1 - Seja o PVC/EDO linear

$$\begin{cases} u''(x) + u(x) = 0, & 0 < x < 1, \\ u(0) = 0, u(1) = \beta_2, \end{cases}$$

fazendo $y_1 = u$ e $y_2 = u'$ tem-se o sistema de equações de primeira ordem equivalente

$$\begin{cases} y_1' = y_2, & 0 < x < 1, \\ y_2' = -y_1, \end{cases}$$

[2]Uma matriz $W(x)$ de ordem $n \times n$, cujas colunas formam uma base do espaço de soluções da EDO homogênea de (2.3a), chama-se matriz solução fundamental ou simplesmente solução fundamental.

$$y_1(0) = 0,\ y_1(1) = \beta_2.$$

Reescrevendo na forma matricial

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}' = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_1(b) \\ y_2(b) \end{bmatrix} = \begin{bmatrix} 0 \\ \beta_2 \end{bmatrix}$$

a matriz solução fundamental, que satisfaz $Y(0) = I$, é

$$Y(x) = \begin{bmatrix} \cos x & \operatorname{sen} x \\ -\operatorname{sen} x & \cos x \end{bmatrix},$$

e

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} I + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos x & \operatorname{sen} x \\ -\operatorname{sen} x & \cos x \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \operatorname{sen} b & \operatorname{sen} b \end{bmatrix}$$

que é singular somente se $b = n\pi$ para qualquer inteiro n. $\triangle$

2.2 - Estabilidade da Solução do PVC/EDO Linear

O efeito que pequenas perturbações nos dados tem na solução de um problema é fundamental quando se está procurando soluções aproximadas numericamente. Para obter bons resultados, o problema, assim como o método numérico, devem ser estáveis, ou seja, pequenas perturbações nos dados não devem causar grandes mudanças na solução.

Considerando PVC/EDO do tipo (2.3a) e assumindo que $A(x)$ e $\mathbf{q}(x)$ são contínuas em $[a, b]$, que $\max(||B_a||_\infty, ||B_b||_\infty)^* = 1$ e que (2.3a) possui uma solução da forma (2.3b), esta solução pode ser reescrita na forma

$$\mathbf{y}(x) = \Phi\mathbf{d} + \int_a^b G(x,t)\mathbf{q}(t)\,dt, \tag{2.4}$$

onde

$$\Phi(x) = Y(x)Q^{-1},$$

$$G(x,t) = \begin{cases} \Phi(x)B_a\Phi(a)\Phi^{-1}(t) & \text{se} \quad x > t, \\ -\Phi(x)B_b\Phi(b)\Phi^{-1}(t) & \text{se} \quad x < t. \end{cases}$$

a função G é a função de Green associada ao problema.

*$||A|| = ||A||_\infty = \max_i \sum_j |a_{ij}|$, e $||A||_1 = \max_j \sum_i |a_{ij}|$.

De (2.4) tem-se que

$$||\mathbf{y}||_\infty \leq k_1||\mathbf{d}||_\infty + k_2||\mathbf{q}||_\infty,$$

onde

$$k_1 = ||YQ^{-1}||_\infty \qquad \text{e}$$

$$k_2 = \sup_{a\leq x,t\leq b}\{\int_a^b ||G(x,t)||_1\, dt\}.$$

A solução y, na forma (2.4), está em função dos dados do problema (2.3a), possibilitando, com isso, estabelecer a estabilidade (ou condicionamento) do PVC.

A constante $k = \max\{k_1, k_2\}$ é denominada constante de condicionamento do problema. Um PVC é bem condicionado se a constante k tiver valor moderado.[3] Pela definição de k_1 e k_2 pode-se observar que k_2 pode ser tomado como a constante de condicionamento.

Na prática, a verificação do bom condicionamento de um PVC pela determinação de k_2, geralmente, é muito difícil, uma vez que requer a determinação da função de Green (G(x,t)). Uma outra forma de estabelecer o condicionamento de um PVC é através do conceito de *dicotomia* da solução fundamental do PVI associado. Diz-se que uma solução fundamental possui uma dicotomia quando existe um subespaço de dimensão p, de soluções crescentes, para algum $p \leq n$, e um subespaço de dimensão $n-p$, de soluções decrescentes, onde n é a dimensão do espaço de solução. Os subespaços de soluções crescentes e decrescentes serão denominados modos crescente e decrescente de soluções, respectivamente.

Se existe uma dicotomia, de acordo com Ascher et al.[1988], a constante k_2 pode ser expressa em função de k_1, da seguinte forma:

$$k_2 = C(2k_1 + 1), \qquad \text{C constante real.}$$

Se a matriz $A(x)$ de (2.3a) for constante, o subspaço de soluções crescentes está relacionado com os autovalores de A com parte real positiva e o de soluções decrescentes com autovalores de A com parte real negativa. Se $A(x)$ não for constante, os autovalores de A não fornecem nenhuma informação sobre os modos de crescimento da função. Mas se a matriz $A(x)$ for estritamente diagonal dominante para todo $x \in [a,b]$, ($|a_{ii}| > \sum_{j\neq i}|a_{ij}|$ para todo i), e existirem constantes λ , $\mu \geq 0$ e $0 \leq p \leq n$ tal que

$$R_e(a_{ii}) > \mu, \quad i = 1, \ldots, m-p,$$

$$R_e(a_{ii}) < -\lambda, \quad i = m-p+1, \ldots, n,$$

[3] o valor de k fornece uma estimativa do número de digitos corretos na solução, ou seja, a constante de condicionamento nos permite avaliar a qualidade da solução obtida (veja constante de condicionamento em Ascher et al.[1988] e em Johnston [1982]).

então a solução fundamental possui uma dicotomia conforme Ascher et al.[1988]. A condição de diagonal dominância pode ser relaxada, bastando que exista uma matriz de transformação S tal que $S^{-1}AS$, seja diagonal dominante.

2.3 - Solução Numérica

O sucesso de processos numéricos depende de dois fatores: bom condicionamento dos problemas e algoritmos estáveis para resolvê-los. A construção de algoritmos estáveis para resolução de PVC é o assunto do próximo capítulo.

Viu-se que, teoricamente, existe uma relação muito próxima entre PVI e PVC, isto é, a existência e representação da solução de um PVC foram dadas em função da solução fundamental, que são soluções de PVIs associados ao PVC. Uma forma simples de construir métodos numéricos para um PVC é considerar os PVIs associados e resolver este último numericamente. Existem outros métodos para resolver numericamente um PVC que não usam explicitamente a resolução de PVIs, mas se forem olhados com mais cuidado é possivel encontrar uma relação destes métodos com aqueles que usam PVIs diretamente.

3 - SOLUÇÃO NUMÉRICA DE PVC/EDO

Neste capítulo serão estudados alguns métodos numéricos para a determinação de soluções aproximadas de PVC/EDO. Esses métodos dividem-se em duas classes: A primeira, constituída de métodos de valor inicial - métodos do *shooting* simples e do *shooting* múltiplo - usa PVIs associados para encontrar uma solução aproximada, conforme mostrado no capítulo anterior; a segunda, constituída de métodos conceitualmente diferentes da classe anterior, porque nenhum problema de valor inicial será integrado explicitamente, procurando, em vez disso, uma solução aproximada sobre todo o intervalo. Nessa classe serão estudados os métodos implícito de Runge-Kutta (método das diferenças finitas de passo simples), da colocação e o método de Ritz (método dos elementos finitos). Também será apresentado o método de Newton para sistemas não lineares.

3.1 - Métodos do Valor Inicial: Problemas Lineares

Nestes métodos um valor inicial é atribuído à solução procurada de tal forma que a solução da EDO com esse valor inicial satisfaça as condições de contorno do problema.

Seja o problema linear

$$\begin{cases} \mathbf{y}' = A(x)\mathbf{y} + \mathbf{q}(x), & \text{se} \quad a < x < b, \\ B_a\mathbf{y}(a) + B_b\mathbf{y}(b) = \mathbf{d}, \end{cases}$$

cuja solução é dada por

$$\mathbf{y}(x) = Y(x)\mathbf{s} + \mathbf{v}(x), \tag{3.1a}$$

onde $Y(x)$ é uma solução fundamental de

$$\begin{cases} Y'(x) = A(x)Y(x), \\ Y(a) = I, \end{cases} \tag{3.1b}$$

e $\mathbf{v}(x)$ uma solução particular , satisfazendo o PVI

$$\begin{cases} \mathbf{v}'(x) = A(x)\mathbf{v}(x) + \mathbf{q}(x), \\ \mathbf{v}(a) = 0, \end{cases} \tag{3.1c}$$

e $\mathbf{s}$ um vetor de parâmetros.

3.1.1 - Método do *Shooting* Simples

Aqui é atribuído um valor à solução no ponto inicial do intervalo, isto é $\mathbf{y}(a) = \mathbf{y}_0$ e a solução desse PVI deve satisfazer o PVC dado, veja figura 3.1.

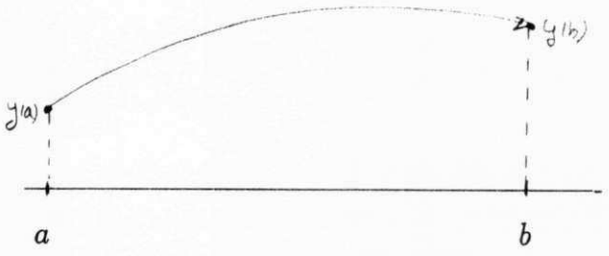


Figura 3.1 - *Shooting* Simples

Para determinar a solução do PVC dado em (3.1a) deve-se encontrar as n colunas da solução fundamental $Y(x)$ e a solução particular $\mathbf{v}(x)$ em (3.1b,c), para isso, são resolvidos $n+1$ PVIs. O vetor $\mathbf{s}$ é a solução do sistema linear

$$Q\mathbf{s} = \widehat{\mathbf{d}}, \tag{3.2a}$$

onde

$$Q = B_a Y(a) + B_b Y(b), \tag{3.2b}$$

$$\widehat{\mathbf{d}} = \mathbf{d} - B_b \mathbf{v}(b), \tag{3.2c}$$

obtendo, assim, $\mathbf{y}(a) = \mathbf{s}$. Resolvendo o PVI

$$\begin{cases} \mathbf{y}'(x) = A(x)\mathbf{y}(x) + \mathbf{q}(x), \\ \mathbf{y}(a) = \mathbf{s}, \end{cases} \tag{3.2d}$$

obtém-se a solução nos pontos desejados.

O método descrito é denominado de *superposição com shooting simples* .

EXEMPLO 3.1 - Seja o PVC/EDO linear

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}' = \begin{bmatrix} 0 & 1 \\ \lambda^2 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + (1-\lambda^2) \begin{bmatrix} 0 \\ e^x \end{bmatrix}, \qquad 0 < x < b,$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_1(b) \\ y_2(b) \end{bmatrix} = \begin{bmatrix} 1 \\ e^b \end{bmatrix}$$

com λ e $b > 0$, cuja matriz solução fundamental é

$$Y(x) = \begin{bmatrix} \cosh \lambda x & \lambda^{-1} \operatorname{senh} \lambda x \\ \lambda \operatorname{senh} \lambda x & \cosh \lambda x \end{bmatrix},$$

a solução particular do PVI não homogêneo é

$$\mathbf{v}(x) = \begin{bmatrix} e^x - \cosh \lambda x - \lambda^{-1} \operatorname{senh} \lambda x \\ e^x - \lambda \operatorname{senh} \lambda x - \cosh \lambda x \end{bmatrix},$$

o vetor de parametros é

$$\mathbf{s} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

e a solução é

$$\mathbf{y}(x) = \begin{bmatrix} e^x \\ e^x \end{bmatrix}. \qquad \triangle$$

A qualidade da solução obtida depende do erro de arredondamento e do erro de discretização do problema. Nas implementações desses métodos as rotinas de integração dos PVIs fazem o controle dos erros de arredondamento relativos e absolutos na determinação de Y e $\mathbf{v}$ de acordo com uma tolerância dada (Tol). Considerando a malha $a \leq x_1 < \cdots < x_{n+1} \leq b$, usada na discretização dos PVIs, o erro de discretização de $\mathbf{y}$ é estimado em $||\mathbf{y}(x_i) - \mathbf{y}_i|| \leq K\, Tol\, k$, onde k é a constante de condicionamento do PVC, e a constante K terá valor moderado se k o for. Quanto ao erro de arredondamento, é esperado que seu crescimento seja proporcional a $\varepsilon_M ||Y(x)||$, onde ε_M é o menor número da máquina tal que $1 + \varepsilon_M > 1$, denominado *epsilon da máquina*. As dificuldades quanto à precisão e estabilidade numérica aparecem se a solução fundamental $Y(x)$ apresentar modos de crescimento rápido (se existir modos de crescimento rápido na solução fundamental o PVI associado poderá ser mal condicionado). Se a matriz $Y(b)$ tiver elementos muito grandes, o acúmulo do erro de arredondamento será grande. Uma forma de tratar esse problema está em requerer que $||Y(x)|| < \frac{Tol}{K k \varepsilon_M}$, e que a matriz Q do sistema $Q\mathbf{s} = \widehat{\mathbf{d}}$ seja bem condicionada, isto é, $\text{cond}Q = ||Q||||Q^{-1}||$ tenha valor moderado conforme Ascher et al.[1988].

Se o PVC (2.3a) possuir condições de contorno separáveis ou parcialmente separáveis,

$$\begin{aligned} & B_{a1}\mathbf{y}(a) = \mathbf{d}_1 \quad \text{e} \quad B_{b2}\mathbf{y}(b) = \mathbf{d}_2, \\ \text{ou} \quad & B_{a1}\mathbf{y}(a) = \mathbf{d}_1 \quad \text{e} \quad B_{a2}\mathbf{y}(a) + B_{b2}\mathbf{y}(b) = \mathbf{d}_2, \end{aligned}$$

onde $B_{a1} \in \mathbb{R}^{(n-v)\times n}$, $\mathbf{d}_1 \in \mathbb{R}^{n-v}$, B_{a2} e $B_{b2} \in \mathbb{R}^{v\times v}$ e $\mathbf{d}_2 \in \mathbb{R}^v$, para $0 < v < n$, pode-se encontrar a solução $\mathbf{y}(x)$ resolvendo somente $v+1$ PVIs, e a solução será dada por

(3.3a) $$\mathbf{y}(x) = \widetilde{Y}(x)\widetilde{\mathbf{s}} + \mathbf{v}(x),$$

onde $\widetilde{Y}(x) \in I\!R^{n\times v}$ é uma solução fundamental de

$$\text{(3.3b)} \qquad \begin{cases} \widetilde{Y}'(x) = A(x)\widetilde{Y}(x), \\ B_{a1}\widetilde{Y}(a) = 0, \end{cases}$$

$\mathbf{v}(x)$, uma solução particular do PVI

$$\text{(3.3c)} \qquad \begin{cases} \mathbf{v}'(x) = A(x)\mathbf{v}(x) + \mathbf{q}(x), \\ \mathbf{v}(a) = \widehat{Y}(a)R^{-1}B_{a1}, \end{cases}$$

e o vetor $\widetilde{\mathbf{s}}$ é a solução do sistema linear

$$\widetilde{Q}\widetilde{\mathbf{s}} = \widetilde{\mathbf{d}},$$

onde

$$\widetilde{Q} = B_{a2}\widetilde{Y}(a) + B_{b2}\widetilde{Y}(b),$$
$$\widetilde{\mathbf{d}} = \mathbf{d}_2 - B_{a2}\mathbf{v}(a) - B_{b2}\mathbf{v}(b).$$

As matrizes $\widetilde{Y}(a), \widehat{Y}(a)$ e R, resultam da decomposição QR da matriz B_{a1}^t em uma matriz H e R ou seja,

$$\text{(3.3d)} \qquad B_{a1}^t = H^t \begin{bmatrix} R^t \\ 0 \end{bmatrix};$$

a matriz $\widetilde{Y}(a)$ é formada pelas v últimas colunas de H^t e $\widehat{Y}(a)$ pelas $n - v$ primeiras,

$$\text{(3.3e)} \qquad H^t = \begin{bmatrix} \widehat{Y}(a) & \widetilde{Y}(a) \end{bmatrix}$$

O método do *shooting* simples com as modificações mostradas acima é denominado de *superposição reduzida.*

A vantagem que esse método oferece com relação ao anterior, está no número de PVIs resolvidos e no tamanho da matriz $\widetilde{Q}$ envolvida, mas apresenta a desvantagem da maior complexidade na determinação dos valores iniciais.

EXEMPLO 3.2 - Seja o PVC/EDO linear do exemplo 3.1.

No exemplo 3.1 foi aplicado o método da superposição, onde foi necessário resolver $n + 1$ PVIs. Mas pode-se observar que esse problema apresenta condições de contorno separáveis, onde

$$B_{a1} = B_{b2} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \text{e} \quad v = 1.$$

Por (3.3d)

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{e} \quad R = [1],$$

e por (3.3e) obtêm-se os dados iniciais

$$\widetilde{Y}(a) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{e} \quad \mathbf{v}(a) = \begin{bmatrix} e^b \\ 1 \end{bmatrix} \qquad \triangle$$

Uma maneira de melhorar a qualidade da solução é diminuir o intervalo de integração de cada PVI, ou seja, subdividir o intervalo $[a, b]$ em uma *malha*

$$\Pi : a = x_1 < x_2 < \cdots < x_N < x_{N+1} = b, \tag{3.4}$$

e o problema é resolvido em cada subintervalo $[x_i, x_{i+1}], i = 1, \ldots, N$.

3.1.2 - Método do *Shooting* Múltiplo

Neste método procede-se da mesma forma que no método do *shooting* simples em cada subintervalo da malha Π dada em (3.4), ou seja, em cada subintervalo $[x_i, x_{i+1}]$ da malha Π é atribuído um valor inicial a $\mathbf{y}(x_i)$ e procura-se obter a solução do PVC em x_{i+1}, $\mathbf{y}(x_{i+1})$, não deixando de observar a continuidade da solução, $\mathbf{y}(x_{i+1}) = \mathbf{y}_{i+1}(x_{i+1})$, no intervalo $[a, b]$, veja figura 3.2.

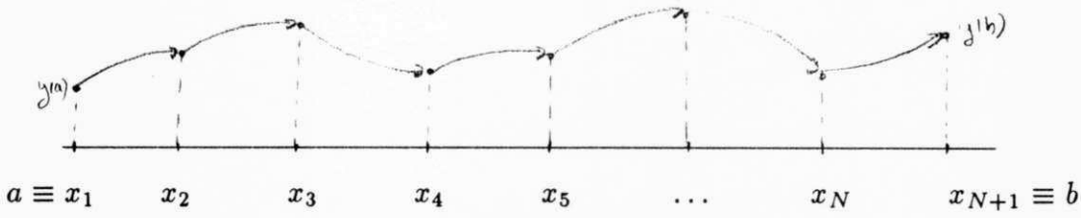


Figura 3.2 - *Shooting* Múltiplo

Para descrever o método considere o problema (2.3a) e a malha Π. Em cada subintervalo de Π, $[x_i, x_{i+1}], i = 1, \ldots, N$, uma solução da forma

$$\mathbf{y}_i(x) = Y_i(x)\widetilde{\mathbf{s}}_i + \mathbf{v}_i(x), \quad \text{para} \quad \widetilde{\mathbf{s}}_i = F_i^{-1}\mathbf{s}_i, \tag{3.5a}$$

é determinada, onde para cada i são definidos os PVIs

$$\begin{cases} Y_i'(x) &= A(x)Y_i(x), \\ Y_i(x_i) &= F_i, \end{cases} \quad \text{e} \quad \begin{cases} \mathbf{v}_i'(x) &= A(x)\mathbf{v}_i(x) + \mathbf{q}(x), \\ \mathbf{v}_i(x_i) &= 0. \end{cases} \tag{3.5b}$$

Os nN parâmetros de $\widetilde{\mathbf{s}} = [\widetilde{\mathbf{s}}_1\ \widetilde{\mathbf{s}}_2\ \ldots\ \widetilde{\mathbf{s}}_N]^t$, $\widetilde{\mathbf{s}}_i \in I\!R^n$, são determinados considerando a continuidade da solução (3.5a) nos extremos dos subintervalos e as condições de contorno, ou seja:

$$\mathbf{y}_i(x_{i+1}) = \mathbf{y}_{i+1}(x_{i+1}) \qquad \text{e} \qquad B_a\mathbf{y}_1(a) + B_b\mathbf{y}_N(b) = \mathbf{d},$$

ou

$$Y_i(x_{i+1})\widetilde{\mathbf{s}}_i + \mathbf{v}_i(x_{i+1}) = Y_{i+1}(x_{i+1})\widetilde{\mathbf{s}}_{i+1},$$

ou ainda,

$$-Y_i(x_{i+1})\widetilde{\mathbf{s}}_i + F_{i+1}\widetilde{\mathbf{s}}_{i+1} = \mathbf{v}_i(x_{i+1}), \tag{3.5c}$$

e

$$B_aF_1\widetilde{\mathbf{s}}_1 + B_bY_N(b)\widetilde{\mathbf{s}}_N = \mathbf{d} - B_b\mathbf{v}_N(b). \tag{3.5d}$$

Portanto, tem-se um sistema de nN equações com nN incógnitas

$$A\widetilde{\mathbf{s}} = \widehat{\mathbf{d}} \tag{3.5e}$$

onde

$$A = \begin{bmatrix} -Y_1(x_2) & F_2 & 0 & \ldots & 0 \\ 0 & -Y_2(x_3) & F_3 & \ldots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \ldots & & -Y_{N-1}(x_N) & F_N \\ B_aF_1 & \ldots & & 0 & B_bY_N(b) \end{bmatrix}, \tag{3.5f}$$

e

$$\widehat{\mathbf{d}} = \begin{bmatrix} \mathbf{v}_i(x_2) \\ \vdots \\ \mathbf{v}_{N-1}(x_N) \\ \mathbf{d} - B_b\mathbf{v}_N(b) \end{bmatrix}. \tag{3.5g}$$

A solução do problema nos pontos da malha será:

$$\mathbf{y}_i = F_i\widetilde{\mathbf{s}}_i = \mathbf{s}_i, \qquad i = 1, \ldots, N+1. \tag{3.5h}$$

Escolha da Malha e a Estabilidade do Método: O sucesso do método está relacionado com uma adequada escolha da malha, ou seja, o número de pontos de *shooting* deve

ser tal que a solução aproximada tenha um erro aceitável. Existe muita discussão sobre a ótima escolha dos pontos de *shooting* conforme Childs et al.[1979] pg.159. Um dos critérios para essa escolha está relacionado com a estabilidade do método. Se o PVC (2.3a) for bem condicionado e a malha, Π, for escolhida de forma que $M = \max_{1 \leq i \leq N} ||\Gamma_i||$ tenha valor moderado (onde $\Gamma_i = Y_i(x_{i+1})$ se $Y_i(x_i) = I$, caso contrário $\Gamma_i = Y_i(x_{i+1})F_i^{-1}$), o método será estável, no sentido de que a amplificação do erro de arredondamento é tolerável e estimado em $\approx \text{cond}(A)\widehat{N}\epsilon_M$, onde $\widehat{N}$ é o número de passos usados para resolver os PVIs entre dois pontos de *shooting* consecutivos.

De acordo com Ascher et al.[1988] a malha para um PVC bem condicionado com constante de condicionamento k deve ser de $N = \frac{Tol}{\widehat{N}kM\epsilon_M}$ subintervalos, onde Tol é a tolerância dada. Mas como se pode notar, a tarefa de escolher os pontos de *shooting* dessa forma não é fácil, uma vez que M depende de N e vice-versa.

Basicamente existem dois critérios para a escolha da malha: o primeiro fixa os pontos da malha, dividindo o intervalo $[a, b]$ uniformemente, adicionando, posteriormente, pontos de descontinuidade, se existirem, e/ou pontos predeterminados onde se deseja a solução, e cada ponto da malha é um ponto de *shooting*; o segundo, denominado *técnica de marcha*, na qual a escolha da malha é idêntica ao do primeiro critério, mas os pontos de *shooting* são definidos durante o processamento de acordo com um controle feito sobre o modo de crescimento da solução fundamental; quando $||\Gamma_i||$ ultrapassa um valor M prefixado, por exemplo se $M \leq \frac{Tol}{K\,k\,N\,\widehat{N}\,\epsilon_M}$, um novo ponto de *shooting* é considerado, resultando em uma *equidistribuição* global dos pontos de *shooting*.

O Método do *shooting* Múltiplo e suas Variações: Para encontrar uma solução aproximada $\mathbf{y}_i(x)$ em (3.5a) deve-se determinar as n colunas de $Y_i(x_{i+1})$ e a solução particular $v_i(x_{i+1})$ e, para isso deve-se atribuir valores a F_i em (3.5b), montando o sistema (3.5c) para obter o vetor de parâmetros $\mathbf{s}$. As variações desse método, descritas a seguir, dependem da escolha dos dados iniciais, F_i, em cada ponto de *shooting* e da forma de determinar o vetor $\mathbf{s}$.

O método do *shooting múltiplo Padrão* é caracterizado pela escolha comum das condições iniciais em todos os pontos de *shooting*, $F_i = I$, onde I é a matriz identidade. Neste caso, a matriz $\Gamma_i = Y_i(x_{i+1})$ e

(3.6a)
$$A = \begin{bmatrix} -\Gamma_1 & I & 0 & \dots & 0 \\ 0 & -\Gamma_2 & I & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & & -\Gamma_{N-1} & I \\ B_a & \dots & & 0 & B_b\Gamma_N \end{bmatrix},$$

a solução nos pontos de *shooting* é dada por

(3.6b) $$\mathbf{y}_i = \mathbf{y}(x_i) = \mathbf{s}_i.$$

Outra forma de obter os valores iniciais nos pontos de *shooting* é usando a *reortogonalização* da matriz $Y_i(x_{i+1})$, obtendo a matriz F_i consecutivamente da seguinte forma: Fatora-se a matriz $Y_i(x_{i+1})$ obtendo o produto $F_{i+1}\widehat{\Gamma}_i$, onde F_{i+1} é ortogonal e $\widehat{\Gamma}_i$ é triangular superior. Se existir uma dicotomia na solução fundamental e a matriz F_1 tiver sido escolhida convenientemente, ou seja, a matriz F_1 for tal que de alguma forma estabeleça uma ordem nos modos da solução fundamental, haverá um *desacoplamento* dos modos crescentes e decrescentes da solução fundamental. Para atingir esse objetivo é suficiente exigir que a matriz $\widehat{\Gamma}_1$ tenha diagonal ordenada de forma decrescente, isto é, os elementos da diagonal devem aparecer na ordem decrescente ($a_{ii} \geq a_{jj}$, $i < j$). conforme Mattheij e Staarink [1984b].

Quanto a obtenção do vetor $\widetilde{\mathbf{s}}$, pode-se resolver o sistema $A\widetilde{\mathbf{s}} = \widehat{\mathbf{d}}$, usando eliminação de Gauss (explorando, ou não, o fato de A ser esparsa) ou usando a relação de recorrência

$$F_{i+1}\widetilde{\mathbf{s}}_{i+1} = Y_i(x_{i+1})\widetilde{\mathbf{s}}_i + \mathbf{v}_i(x_{i+1})$$

ou

(3.7a) $$\widetilde{\mathbf{s}}_{i+1} = F_{i+1}^{-1}Y_i(x_{i+1})\widetilde{\mathbf{s}}_i + F_{i+1}^{-1}\mathbf{v}_i(x_{i+1}).$$

Para determinar, $\widetilde{\mathbf{s}}_i$, usando (3.7a), constrói-se uma sequência de soluções fundamantais $\{\Phi_i\}_{i=1}^{N+1}$ e de soluções particulares $\{\mathbf{r}_i\}_{i=1}^{N+1}$ por

(3.7b) $$\begin{cases} \Phi_{i+1} = \Gamma_i\Phi_i, \\ \Phi_1 = I \end{cases}$$

e

(3.7c) $$\begin{cases} \mathbf{r}_{i+1} = \Gamma_i\mathbf{r}_i + \mathbf{v}_i(x_{i+1}), \\ \mathbf{r}_1 = 0, \end{cases}$$

onde $\widetilde{\mathbf{s}}_i$ pode ser escrito em função de $\widetilde{\mathbf{s}}_1$, ou seja,

(3.7d) $$\widetilde{\mathbf{s}}_i = \Phi_i\widetilde{\mathbf{s}}_1 + \mathbf{r}_i,$$

e $\widetilde{\mathbf{s}}_1$ depende das condições de contorno,

(3.7e) $$(B_aF_1 + B_bF_{N+1}\Phi_{N+1})\widetilde{\mathbf{s}}_1 = \mathbf{d} - B_bF_{N+1}\mathbf{r}_{N+1}.$$

A solução procurada tem a forma

$$\mathbf{y}(x_i) = F_i\Phi_i\widetilde{\mathbf{s}}_1 + F_i\mathbf{r}_i = \mathbf{s}_i, \quad \text{para} \quad i = 1, \ldots, N.$$

O método, que usa a relação de recorrência (3.7a), como mostrado acima é denominado de *Compactação.*

Relações de recorrência, de uma forma geral, são numericamente instáveis, e, neste caso, retorna-se aos níveis de instabilidade do método do *shooting* simples, que pode ser observado a seguir.

Considerando o método do múltiplo *Shooting* padrão com compactação, $F_i = I$ para todo i, tem-se

$$\Gamma_i = Y_i(x_{i+1}),$$

e, neste caso,

$$\Phi_{N+1} = Y_N(b)Y_{N-1}(x_N)\ldots Y_1(x_2) = Y(b),$$

onde a matriz $Y(b)$ é a matriz do método do *shooting* simples, assim como a matriz $[B_a + B_b\Phi_{N+1}]$ é idêntica à matriz Q de $Q\mathbf{s} = \widehat{\mathbf{d}}$. Portanto o método do *shooting* múltiplo padrão com compactação tem os mesmos problemas de estabilidade encontrados no método do *shooting* simples.

EXEMPLO 3.3 - Considere, novamante o exemplo 3.1, e uma malha $\Pi : 0 \leq x_i < x_2 < \cdots < x_N < x_{N+1} \leq b$. Tomando $Y_i(x_i) = I$ e $\mathbf{v}_i(x_i) = 0$ para todo i.

Em cada subintervalo da malha determina-se a solução $\mathbf{y}_i(x) = Y_i(x)\widetilde{\mathbf{s}}_i + \mathbf{v}_i(x)$ onde a matriz solução fundamantal tem a forma

$$Y_i(x) = \begin{bmatrix} \cosh\lambda(x - x_i) & \lambda^{-1}\operatorname{senh}\lambda(x - x_i) \\ \lambda\operatorname{senh}\lambda(x - x_i) & \cosh\lambda(x - x_i) \end{bmatrix},$$

a solução particular do PVI não homogêneo é

$$\mathbf{v}_i(x) = \begin{bmatrix} e^x - e^{x_i}(\cosh\lambda(x - x_i) - \lambda^{-1}\operatorname{senh}\lambda(x - x_i)) \\ e^x - e^{x_i}(\lambda\operatorname{senh}\lambda(x - x_i) - \cosh\lambda(x - x_i)) \end{bmatrix},$$

o vetor de parâmetros

$$\widetilde{\mathbf{s}} = [\widetilde{\mathbf{s}}_1 \ \ldots \ \widetilde{\mathbf{s}}_N]^t$$

é determinado pela resolução do sistema (3.5e), onde A é dada em (3.6a). A solução procurada nos pontos da malha é dada por

$$\mathbf{y}_i(x_i) = \widetilde{\mathbf{s}}_i, \qquad 1 \leq i \leq N.$$

No método com compactação, as soluções fundamantais Φ_i e as soluções particulares $\mathbf{r}_i$ são determinadas através da forma recursiva de $\widetilde{\mathbf{s}}_i$ (3.7a), obtendo

$$\Phi_1 = I,$$

$$\Phi_2 = Y_1(x_2)\, I = \begin{bmatrix} \cosh\lambda(x_2 - x_1) & \lambda^{-1}\operatorname{senh}\lambda(x_2 - x_1) \\ \lambda\operatorname{senh}\lambda(x_2 - x_1) & \cosh\lambda(x_2 - x_1) \end{bmatrix},$$

$$\Phi_3 = Y_2(x_3)\,\Phi_2 = \begin{bmatrix} \cosh\lambda(x_2 - x_1) & \lambda^{-1}\operatorname{senh}\lambda(x_2 - x_1) \\ \lambda\operatorname{senh}\lambda(x_2 - x_1) & \cosh\lambda(x_2 - x_1) \end{bmatrix},$$

$$\ldots$$

$$\Phi_{N+1} = Y_N(b)\,\Phi_N = \begin{bmatrix} \cosh\lambda(b - x_1) & \lambda^{-1}\operatorname{senh}\lambda(b - x_1) \\ \lambda\operatorname{senh}\lambda(b - x_1) & \cosh\lambda(b - x_1) \end{bmatrix}$$

$$= \begin{bmatrix} \cosh\lambda b & \lambda^{-1}\operatorname{senh}\lambda b \\ \lambda\operatorname{senh}\lambda b & \cosh\lambda b \end{bmatrix}.$$

as soluções particulares são

$$\mathbf{r}_1 = 0,$$

$$\mathbf{r}_2 = \begin{bmatrix} e^{x_2} - e^{x_1}(\cosh\lambda(x_2 - x_1) - \lambda^{-1}\operatorname{senh}\lambda(x_2 - x_1)) \\ e^{x_2} - e^{x_1}(\lambda\operatorname{senh}\lambda(x_2 - x_1) - \cosh\lambda(x_2 - x_1)) \end{bmatrix},$$

$$\mathbf{r}_3 = \begin{bmatrix} e^{x_3} - e^{x_1}(\cosh\lambda(x_3 - x_1) - \lambda^{-1}\operatorname{senh}\lambda(x_3 - x_1)) \\ e^{x_3} - e^{x_1}(\lambda\operatorname{senh}\lambda(x_3 - x_1) - \cosh\lambda(x_3 - x_1)) \end{bmatrix},$$

$$\ldots$$

$$\mathbf{r}_{N+1} = \begin{bmatrix} e^{b} - e^{x_1}(\cosh\lambda(b - x_1) - \lambda^{-1}\operatorname{senh}\lambda(b - x_1)) \\ e^{b} - e^{x_1}(\lambda\operatorname{senh}\lambda(b - x_1) - \cosh\lambda(b - x_1)) \end{bmatrix}$$

$$= \begin{bmatrix} e^{b} - (\cosh\lambda b - \lambda^{-1}\operatorname{senh}\lambda b) \\ e^{b} - (\lambda\operatorname{senh}\lambda b - \cosh\lambda b) \end{bmatrix}.$$

e (3.7e) tem a forma

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \cosh\lambda b & \lambda^{-1}\operatorname{senh}\lambda b \\ \lambda\operatorname{senh}\lambda b & \cosh\lambda b \end{bmatrix}\widetilde{\mathbf{s}}_1 =$$

$$\begin{bmatrix} 1 \\ e^{b} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} e^{b} - (\cosh\lambda b - \lambda^{-1}\operatorname{senh}\lambda b) \\ e^{b} - (\lambda\operatorname{senh}\lambda b - \cosh\lambda b) \end{bmatrix}.$$

e o vetor

$$\widetilde{\mathbf{s}}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Por (3.7d) tem-se

$$\widetilde{\mathbf{s}}_2 = \begin{bmatrix} \cosh\lambda(x_2-x_1) & \lambda^{-1}\operatorname{senh}\lambda(x_2-x_1) \\ \lambda\operatorname{senh}\lambda(x_2-x_1) & \cosh\lambda(x_2-x_1) \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} +$$

$$\begin{bmatrix} e^{x_2} - (\cosh\lambda(x_2-x_1) - \lambda^{-1}\operatorname{senh}\lambda(x_2-x_1)) \\ e^{x_2} - (\lambda\operatorname{senh}\lambda(x_2-x_1) - \cosh\lambda(x_2-x_1)) \end{bmatrix} = \begin{bmatrix} e^{x_2} \\ e^{x_2} \end{bmatrix},$$

$$\widetilde{\mathbf{s}}_3 = \begin{bmatrix} e^{x_3} \\ e^{x_3} \end{bmatrix},$$

$$\ldots$$

$$\widetilde{\mathbf{s}}_{N+1} = \begin{bmatrix} e^{b} \\ e^{b} \end{bmatrix}. \qquad \triangle$$

Uma forma de conseguir um algoritmo mais estável, usando compactação, é usar o desacoplamento da solução fundamental Y_i pela reortogonalização de $Y_i(x_{i+1}) = F_{i+1}\widetilde{\Gamma}_i$, com uma escolha conveniente de F_1. Assumindo que o espaço de solução possui uma dicotomia, a expressão

(3.8a) $$\widetilde{\mathbf{s}}_{i+1} = \widetilde{\Gamma}_i\widetilde{\mathbf{s}}_i + \widehat{\mathbf{d}}_i,$$

onde $\widehat{\mathbf{d}}_i = F_{i+1}\mathbf{v}_i(x_{i+1})$, é instável quando calculada na ordem progressiva , $i = 1, \ldots, N$. O desacoplamento dado em $\widetilde{\Gamma}_i$

(3.8b) $$\widetilde{\Gamma}_i = \begin{bmatrix} D_i & C_i \\ 0 & E_i \end{bmatrix},$$

onde $D_i \in \mathbb{R}^{k\times k}$ representa o modo crescente da solução e $E_i \in \mathbb{R}^{(n-k)\times(n-k)}$ representa o modo decrescente, permite que a expressão (3.8a) seja reescrita na forma

(3.8c) $$D_i\widetilde{\mathbf{s}}_i^1 = \widetilde{\mathbf{s}}_{i+1}^1 - C_i\widetilde{\mathbf{s}}_i^2 - \widehat{\mathbf{d}}^1,$$

(3.8d) $$\widetilde{\mathbf{s}}_{i+1}^2 = E_i\widetilde{\mathbf{s}}_i^2 + \widehat{\mathbf{d}}^2.$$

De acordo com Mattheij e Staarink [1984b] se o particionamento for escolhido corretamente e existindo uma dicotomia no espaço das soluções, então $\|\prod_{j=1}^{i} E_j\|$ e $\|(\prod_{j+l}^{1} D_j)^{-1}\|$ são de ordem 1, portanto a contaminação da solução final pelo erro é pequena, ou seja, as expressões acima podem ser avaliadas de forma estável.

Para determinar as soluções fundamentais Φ_i, usa-se a forma homogênea das expressões (3.8c,d), com as condições iniciais $\Phi_1^2 = [\,0 \quad I_{n-k}\,]$ e $\Phi_{N+1}^1 = [\,I_k \quad 0\,]$ respectivamente.

As soluções particulares $\mathbf{r}_i, i = 1, \ldots, N$, são determinadas a partir das condições iniciais $\mathbf{r}_1^2 = 0$ e $\mathbf{r}_{N+1}^1 = 0$ e as expressões (3.8c,d).

3.2 - Métodos do Valor Inicial: Problemas não Lineares

Considerando o PVC (2.1a) com $\mathbf{f}$ e $\mathbf{g}$ não lineares

$$\text{(3.9a)} \qquad \begin{cases} \mathbf{y}' = \mathbf{f}(x, \mathbf{y}), & \text{se} \quad a < x < b, \\ \mathbf{g}(\mathbf{y}(a), \mathbf{y}(b)) = 0, \end{cases}$$

quando a discretização for feita em (3.9a), resulta um sistema de equações não lineares que é resolvido usando algum método iterativo como o método de Newton.

3.2.1 - O Método de Newton

Sejam $\mathbf{F}(\mathbf{s}) = 0$ um sistema não linear e uma aproximação inicial $\mathbf{s}^o$, onde $\mathbf{s} = [s_1 \ \ldots \ s_J]^t$ e $\mathbf{F}(\mathbf{s}) = [F_1(\mathbf{s}) \ \ldots \ F_J(\mathbf{s})]^t$. Determinam-se, por iteração, os valores $\mathbf{s}^1, \mathbf{s}^2, \ldots, \mathbf{s}^m, \cdots$ pela expressão $\mathbf{s}^{m+1} = \mathbf{s}^m + \mathbf{r}$, onde $\mathbf{r}$ é a solução do sistema

$$F'(\mathbf{s}^m)\mathbf{r} = -\mathbf{F}(\mathbf{s}^m),$$

onde $F'(\mathbf{s})$, com elementos $f_{ij} = \frac{\partial F_i(\mathbf{s})}{\partial s_j}$, é a matriz jacobiana de $\mathbf{F}(\mathbf{s})$. Sob certas condições os vetors $\mathbf{s}^m$ convergem para $\mathbf{s}$ conforme Ascher et al.[1988].

3.2.2 - Método do *Shooting* Simples

Seja $\mathbf{y}(x; \mathbf{s})$ uma solução do PVI associado ao problema (3.9a)

$$\text{(3.11a)} \qquad \begin{cases} \mathbf{y}'(x; \mathbf{s}) = \mathbf{f}(x, \mathbf{y}(x; \mathbf{s})), & \text{se} \quad a < x < b, \\ \mathbf{y}(a; \mathbf{s}) = \mathbf{s}. \end{cases}$$

O objetivo, agora, é determinar $\mathbf{s} = \mathbf{s}^*$ tal que as condições de contorno sejam satisfeitas, isto é,

$$\text{(3.11b)} \qquad g(\mathbf{s}^*, \mathbf{y}(b; \mathbf{s}^*)) = 0.$$

A existência de uma solução isolada de (3.9a), corresponde a uma raiz simples de (3.11b). Para determinar $\mathbf{s}^*$ define-se $\mathbf{F}(\mathbf{s}) = \mathbf{g}(\mathbf{s}, \mathbf{y}(b; \mathbf{s}))$ e usando o método de Newton, tem-se o sistema

$$F'(\mathbf{s})\mathbf{r} = -\mathbf{F}(\mathbf{s}),$$

ou seja

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad \mathbf{f}(x, \mathbf{y}) = \begin{bmatrix} y_2 \\ -e^{y_1} \end{bmatrix},$$

e as condições de contorno

$$\mathbf{g}(\mathbf{y}(0), \mathbf{y}(1)) = \begin{bmatrix} y_1(0) \\ y_1(1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

O sistema não linear

$$\mathbf{F}(\mathbf{s}) = \begin{bmatrix} y_1(0) \\ y_1(1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

e as matrizes B_a e B_b usadas para determinar a matriz jacobiana de $\mathbf{F}(s)$ tem a forma

$$B_a = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad B_b = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

e $Y(b)$ é determinada por (3.11d) onde A tem a forma

$$A = \begin{bmatrix} 0 & 1 \\ -e^{(y_1;\mathbf{s})} & 0 \end{bmatrix}. \qquad \triangle$$

3.2.3 - Método do *shooting* Múltiplo

Seja Π uma malha de N subintervalos. Para cada $\mathbf{s}_i$, $\mathbf{y}_i(x; \mathbf{s}_i)$ é solução do PVI

(3.12a)
$$\begin{cases} \mathbf{y}_i'(x; \mathbf{s}_i) = \mathbf{f}(x, \mathbf{y}(x; \mathbf{s}_i)), & \text{se} \quad x > x_i, \\ \mathbf{y}_i(x_i; \mathbf{s}_i) = \mathbf{s}_i. \end{cases}$$

Para determinar uma solução de (3.9a), deve-se encontrar um vetor $\mathbf{s} \in \mathbb{R}^{nN}$, $\mathbf{s} = [\mathbf{s}_1 \ \ldots \ \mathbf{s}_N]^t$, onde $\mathbf{s}_i \in \mathbb{R}^n$, que deverá ser determinado de forma contínua no intervalo $[a, b]$ e satisfazer as condições de contorno $\mathbf{g}(\mathbf{y}(a), \mathbf{y}(b)) = 0$.

A solução de (3.9a) é dada por:

(3.12b)
$$\mathbf{y}(x) = \mathbf{y}_i(x; \mathbf{s}_i), \qquad x_i \leq x \leq x_{i+1}, \quad i = 1, \ldots, N,$$

onde os $\mathbf{s}_i$ são determinados da seguinte forma:

$$\mathbf{y}_i(x_{i+1}; \mathbf{s}_i) = \mathbf{s}_{i+1}, \qquad i = 1, \ldots, N-1$$

e $$\mathbf{g}(\mathbf{s}_1, \mathbf{y}_N(b; \mathbf{s}_N)) = 0.$$

Define-se a equação $\mathbf{F}(\mathbf{s}) = 0$ por

$$\mathbf{F}(\mathbf{s}) = \begin{bmatrix} \mathbf{s}_2 - \mathbf{y}_1(x_2; \mathbf{s}_1) \\ \mathbf{s}_3 - \mathbf{y}_2(x_3; \mathbf{s}_2) \\ \vdots \\ \mathbf{s}_N - \mathbf{y}_{N-1}(x_N; \mathbf{s}_{N-1}) \\ \mathbf{g}(\mathbf{s}_1, \mathbf{y}_N(b; \mathbf{s}_N)) \end{bmatrix} \tag{3.12c}$$

e usando o método de Newton determina-se o vetor $\mathbf{s}$ desejado.

A matriz $F'(\mathbf{s})$ do sistema $F'(\mathbf{s})\mathbf{r} = -\mathbf{F}(\mathbf{s})$ é dada por

$$F'(\mathbf{s}) = \begin{bmatrix} -Y_1(x_2) & I & 0 & \dots & 0 \\ 0 & -Y_2(x_3) & I & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & & -Y_{N-1}(x_N) & I \\ B_a & \dots & & 0 & B_b Y_N(b) \end{bmatrix}, \tag{3.12d}$$

onde $Y_i(x_{i+1})$ é a solução de

$$\begin{cases} Y_i' = A(x)Y_i, & x > x_i, \quad i = 1, \dots, N, \\ Y_i(x_i) = I, \end{cases} \tag{3.12e}$$

onde $A(x)$ é a jacobiana de $\mathbf{f}$. A matriz $F'(\mathbf{s})$ é semelhante à matriz do método do *shooting* múltiplo padrão para o caso linear.

3.2.4 - Mais sobre o método de Newton

O método de Newton, usado para resolver equações não lineares, possui as seguintes desvantagens:

1. Cada iteração do método é muito dispendiosa, uma vez que requer a avaliação de uma matriz jacobiana e a solução de um sistema de equações lineares;

2. Sob a hipótese de existência de uma solução isolada, o método somente garante convergência local, isto é, necessita de um dado inicial suficientemente próximo da solução.

Uma maneira de reduzir o custo do método é manter a jacobiana fixa. A idéia é a seguinte: encontrar a matriz jacobiana em um ponto, por exemplo $\mathbf{s}^o$, e usar essa jacobiana para gerar mais de uma iteração, ou seja:

$$\mathbf{s}^{m+1} = \mathbf{s}^m - J(\mathbf{s}^o)^{-1}\mathbf{F}(\mathbf{s}^m).$$

Uma outra maneira de contornar as dificuldades apresentadas é controlar o avanço do método de Newton. Calcula-se $\mathbf{s}^{m+1} = \mathbf{s}^m + \lambda\mathbf{r}$, em vez de calcular $\mathbf{s}^{m+1} = \mathbf{s}^m + \mathbf{r}$,

$0 \leq \lambda \leq 1$. O fator λ é denominado fator de amortecimento e o método resultante é chamado *método de Newton amortecido*, o objetivo do método amortecido é não permitir que a próxima aproximação se afaste, de forma indevida, da aproximação corrente. Esse afastamento está ilustrado na figura 3.3, que considera uma função real f.

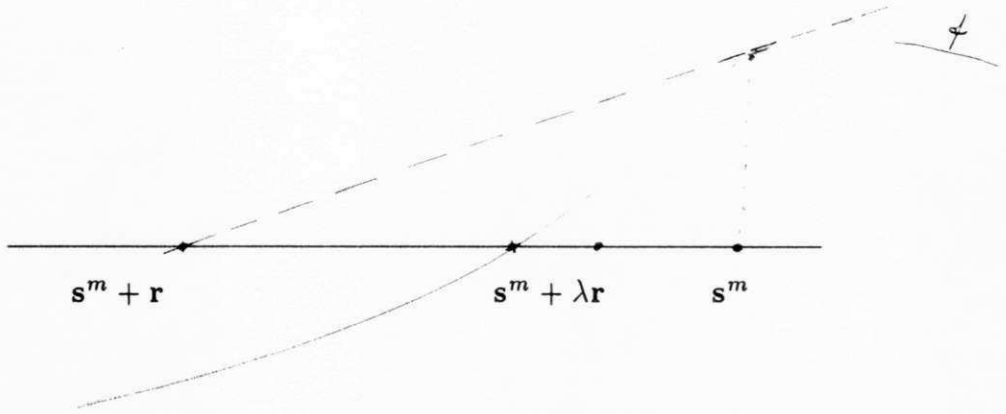


Figura 3.3 - O Método de Newton Amortecido

A principal questão é: como determinar o fator de amortecimento λ, de forma que $\mathbf{s}^{m+1} = \mathbf{s}^m + \lambda\mathbf{r}$ se aproxime mais de $\mathbf{s}^*$, não se conhecendo $\mathbf{s}^*$?

Para essa finalidade define-se uma função g ($g : [0,1] \rightarrow I\!R$), relacionada com alguma norma de $\mathbf{F}(\mathbf{s})$ e que satisfaça às seguintes condições:

i) $g(\mathbf{s}) \geq 0$ e $g(\mathbf{s}^*) = 0$ se e somente se $\mathbf{F}(\mathbf{s}^*) = 0$. $\mathbf{s}^*$ é um mínimo de $g(\mathbf{s})$ se $\mathbf{F}(\mathbf{s}^*) = 0$, e diz-se que um valor λ é melhor que outro se $g(\mathbf{s}^m + \lambda\mathbf{r}) < g(\mathbf{s}^m)$;

ii) A direção de Newton é decrescente com relação a $g(\mathbf{s})$, isto é, $\mathbf{r}^t\nabla g < 0$, onde $\mathbf{r} = -(F'(\mathbf{s}))^{-1}\mathbf{F}(\mathbf{s})$ (∇g é o gradiente de g).

Portanto, o objetivo é determinar um valor de λ que minimize a função g, ou seja, $g(\mathbf{s}^{m+1}) < g(\mathbf{s}^m)$. A seleção de λ aceitável pode ser vista como um processo de *Previsão* e *Correção*.

De acordo com Ascher et al.[1988], um valor de λ é aceitável se satisfizer:

(3.13a) $$g(\mathbf{s}^{m+1}) \leq (1 - 2\lambda\delta)g(\mathbf{s}^m)$$

(3.13b) $$g(\mathbf{s}^{m+1}) \geq (1 - 2\lambda(1 - \delta))g(\mathbf{s}^m)$$

para algum $0 < \delta < 1/2$.

Se $\mathbf{F}(\mathbf{s})$ possuir derivadas segundas limitadas e $\text{cond}(F^{-1}(\mathbf{s}))$ for uniformemente limitada em um domínio $D = \{\mathbf{s} \in I\!R^n; g(\mathbf{s}) < g(\mathbf{s}^o)\}$ que contenha as iterações, $\mathbf{s}^m$, e a raiz $\mathbf{s}^*$, então iniciando o processo por um ponto $\mathbf{s}^o \in D$, o método de Newton amortecido, usando valores aceitáveis para λ em $\mathbf{s}^{m+1} = \mathbf{s}^m + \mathbf{r}$, converge para $\mathbf{s}^*$.

Para determinar valores aceitáveis de λ, pode-se tomar um valor $0 < \rho < 1$ e um novo λ será determinado por $\lambda = \rho\lambda_m$ onde λ_m é um valor previsto para λ. Como prever esses valores será visto no decorrer desta seção. Essa forma de determinar λ pode levar a algoritmos pouco eficientes, no caso de muitos valores de λ serem testados.

Uma forma mais eficiente de determinar valores de λ é aproximar $g(\mathbf{s}^m + \lambda\mathbf{r})$ por uma função quadrática satisfazendo:

$$\Psi(0) = g(\mathbf{s}^m),$$
$$\Psi'(0) = \mathbf{r}^t \nabla g(\mathbf{s}),$$
$$\Psi(\lambda) = g(\mathbf{s}^m + \lambda\mathbf{r}),$$

um mínimo para a função Ψ é dado por

$$\lambda = \frac{-\lambda_m^2 \Psi(0)}{2(\Psi(\lambda_m) - \Psi(0) - \lambda_m \Psi'(0))} \leq \frac{\lambda_m}{2(1-\delta)} \tag{3.13c}$$

Toma-se $\lambda_m = \lambda$, e verifica se a condição (3.13a) é satisfeita; se não for, diminui λ, usando (3.13c) até que as condições de aceitabilidade sejam satisfeitas.

Para cada correção de λ somente uma avaliação de g é necessária e quando a condição (3.13a) for satisfeita, (3.13b) também o é.

A função g de uma forma geral pode ser tomada como $g(\mathbf{s}) = \frac{1}{2}||M\mathbf{F}(\mathbf{s})||_2^2$, M matriz não singular. A função g assim definida satisfaz i) e ii). A matriz M pode ser $F(\mathbf{s}^m)^{-1}$ e neste caso g varia de iteração em iteração e para deixar isso bem claro escreve-se

$$g(\mathbf{s}^m + \lambda\mathbf{r}) = \frac{1}{2}||(F'(\mathbf{s}^m))^{-1}\mathbf{F}(\mathbf{s}^m + \lambda\mathbf{r})||_2^2.$$

O método de Newton com g assim definida pode ser cíclico ($\mathbf{s}^m = \mathbf{s}^o$, para algum m). Para superar essa dificuldade, definem-se alguns parâmetros como o menor valor de λ (λ_{min}), se λ obtido for menor que λ_{min} então nenhuma convergência foi obtida. Um segundo parâmetro que deve ser definido é um limite para a distância entre λ e λ_m em uma troca em (3.13c).

Quanto à previsão, vários métodos podem ser utilizados, por exemplo, tomar o primeiro valor para λ_m, $\lambda_m^{(o)} = 1$; isto é normalmente usado quando $m = 0$, e também pode-se usar informações de λ anteriormente determinados para prever um novo valor, por exemplo,

$$\lambda_m^{(0)} = \begin{cases} \lambda_{m-1}, & \text{se } \lambda_{m-1} < \lambda_{m-2}(1-\delta), \\ \min(1, 2\lambda_{m-1}), & \text{caso contrário.} \end{cases}$$

O método de Newton amortecido tem a vantagem de detectar falhas rapidamente, o que não ocorre no método de *Newton Completo* ($\lambda = 1$), e no caso de mais de uma solução, após encontrar a solução mais próxima do valor inicial $\mathbf{s}^o$, é possível retornar ao problema para procurar outras soluções.

3.3 - Método das Diferenças Finitas de Passo Simples

Neste método uma solução aproximada do PVC/EDO é encontrada pela resolução de um sistema de equações algébricas resultante da substituição das derivadas por quocientes de diferenças na equação diferencial dada, em uma malha Π previamente escolhida, e das condições de contorno que a solução exata deve satisfazer, obtendo uma solução discreta $\mathbf{y}_i \equiv \mathbf{y}_\Pi(x_i)$. A solução aproximada $\mathbf{y}_\Pi(x)$ para todo $x \in [a, b]$ pode ser determinada usando interpolação. O método é denominado de passo simples quando a solução em um ponto x_{i+1} de Π depender apenas de informações sobre a solução no ponto x_i da malha.

A forma geral desse método para PVC de primeira ordem é encontrada resolvendo $\mathbf{y}' = \mathbf{f}(x, \mathbf{y})$ em cada intervalo $[x_i, x_{i+1}]$ da malha Π, substituindo a integral pela regra de integração numérica, obtendo

$$\frac{\mathbf{y}(x_{i+1}) - \mathbf{y}(x_i)}{h_i} = \sum_{j=1}^{k} \alpha_j \mathbf{f}(\xi_i, \mathbf{y}(\xi_i)),$$

onde $\{\xi_j\}_{j=1}^k$ é uma sequência não decrescente de pontos em $[x_i, x_{i+1}]$, e $h_i = x_{i+1} - x_i$ e os α_j são parâmetros dependentes do polinômio interpolante usado para substituir $\mathbf{f}$ na integral, e k é o grau do polinômio interpolante usado na integração numérica. A precisão da integração depende da escolha dos pontos ξ_j no interior do intervalo. Se forem considerados pontos da forma $\xi_i = x_i + \rho_j h_i$, $j = 1, \ldots, k$ onde $h_i = x_{i+1} - x_i$ e $0 < \rho_1 < \cdots < \rho_k < 1$ e ρ_j satisfizerem a condição de ortogonalidade

$$\int_0^1 p(x) \prod_{j=1}^{k} (x - \rho_j) = 0, \qquad p(x) \in P_s \ (s < k),$$

onde P_s é espaço das funções polinomiais de grau $< s$, então a precisão do método será $O(h^{2k})$. Pontos determinados dessa forma são chamados *pontos de Gauss.*

Estabilidade do método: Considere o problema linear (2.3a), o operador diferencial linear

$$L\mathbf{y}(x) = \mathbf{y}'(x) - \mathbf{A}(x)\mathbf{y}(x), \qquad a < x < b,$$

e o correspondente operador diferença

$$L_\Pi \mathbf{y}_i = \frac{\mathbf{y}_{i+1} - \mathbf{y}_i}{h_i} - \Psi(\mathbf{y}_i, \mathbf{y}_{i+1}; x_i, h_i)^*.$$

O método das diferenças finitas de passo simples é *consistente* de ordem p ($p > 0$), se para toda solução de $\mathbf{y}' = A(x)\mathbf{y}$ existem constantes c e $h_o > 0$ tais que para toda malha Π com $h = \max_{1<i<N} h_i \leq h_o$, se tenha

$$\tau[\mathbf{y}] = \max_{1 \leq i \leq N} ||\tau_i[\mathbf{y}]|| \leq ch^p,$$

onde $\tau_i[\mathbf{y}]$ é o erro de truncamento local definido por:

$$\tau_i[\mathbf{y}] = L_\Pi \mathbf{y}(x_i) - L_\Pi \mathbf{y}_i, \qquad 1 \leq i \leq N.$$

O método é *estável* se existe uma constante $K > 0$, tal que as funções $\mathbf{y}_\Pi$ satisfazem

$$||\mathbf{y}_\Pi|| \leq K \max_{1 \leq i \leq N+1} \{||B_a\mathbf{y}_1 + B_b\mathbf{y}_{N+1}||, \max_{1 \leq j \leq N} ||L_\Pi \mathbf{y}_j||\}.$$

K terá valor moderado se a constante de condicionamento do PVC, k, também tiver. A solução aproximada será convergente para a solução exata se

$$\max_{1 \leq i \leq N+1} ||\mathbf{y}_i - \mathbf{y}(x_i)|| \to 0 \quad \text{quando} \quad h \to 0,$$

e a convergência ocorre quando o método de passo simples for consistente e estável conforme Ascher et al.[1988].

Nessa classe serão estudados os seguintes métodos: implícito de Runge-Kutta e o da colocação. Será estabelecida a equivalência do método implícito de Runge-Kutta com o método da colocação para PVC de primeira ordem, como também para o método da colocação para PVC de ordem superior.

---

*A função $\Psi(\mathbf{y}_i, \mathbf{y}_{i+1}; x_i, h_i)$ depende do método usado para determinar a solução do problema discretizado, por exemplo, usando fórmulas implícitas de Runge-Kutta $\Psi(\mathbf{y}_i, \mathbf{y}_{i+1}; x_i, h_i) = \sum_{j=1}^{k} \beta_j \mathbf{k}_j, \quad 1 \leq i \leq N$.

3.3.1 - Método Implícito de Runge-Kutta.

Este método usa avaliações da derivada da equação diferencial

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y})$$

nos pontos interiores do intervalo $[x_i, x_{i+1}]$ da malha Π definida em (3.4) para determinar uma solução aproximada $\mathbf{y}_\Pi$, da seguinte forma: Considera a equação geral do método de passo simples

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h_i \sum_{j=1}^{k} \beta_j \mathbf{k}_j, \qquad 1 \leq j \leq N, \tag{3.14a}$$

$$\text{com} \qquad k_j = \mathbf{f}_{ij} = \mathbf{f}(x_{ij}, \mathbf{y}_i + h_i \sum_{l=1}^{k} \alpha_{jl} \mathbf{f}_{il}), \qquad 1 \leq j \leq k,$$

onde $x_{ij} = x_i + \rho_j h_i$, $1 \leq j \leq k$, $1 \leq i \leq N$ com $0 \leq \rho_1 \leq \rho_2 \leq \cdots \leq \rho_k \leq 1$. Os pontos x_{ij} são denominados pontos de colocação.

Considerando o caso linear, seja o PVC

$$\begin{cases} \mathbf{y}' = A(x)\mathbf{y} + \mathbf{q}(x) & \text{se} \quad a < x < b, \\ B_a \mathbf{y}(a) + B_b \mathbf{y}(b) = \mathbf{d}, \end{cases} \tag{3.14b}$$

e a malha Π, a solução aproximada $\mathbf{y}_\Pi$ deve satisfazer as condições de contorno $B_a \mathbf{y}_1 + B_b \mathbf{y}_{N+1} = \mathbf{d}$ e os pontos de colocação em cada subintervalo da malha Π, ou seja, em cada intervalo $[x_i, x_{i+1}]$, a solução aproximada deve satisfazer à formulação de Runge-Kutta

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h_i \sum_{j=1}^{k} \beta_j (A(x_{ij})\mathbf{y}_{ij} + \mathbf{q}(x_{ij})), \tag{3.14c}$$

onde

$$\mathbf{y}_{ij} = \mathbf{y}_i + h_i \sum_{l=1}^{k} \alpha_{jl} (A(x_{il})\mathbf{y}_{il} + \mathbf{q}(x_{il})).$$

Existem dois tipos de variáveis; os vetores $\mathbf{y}_i$, $i = 1, \ldots, N+1$, determinados nos pontos da malha Π, são denominados variáveis globais e as variáveis, $\mathbf{y}_{ij}$, $i = 1, \ldots, N+1$ e $j = 1, \ldots, k$, determinadas no interior dos subintervalos, são denominadas variáveis locais.

Eliminando as variáveis locais em (3.14c) tem-se:

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h_i \sum_{j=1}^{k} \beta_j (A(x_{ij})(\mathbf{y}_i + h_i \sum_{l=1}^{k} \alpha_{jl}(A(x_{il})\mathbf{y}_{il} + \mathbf{q}(x_{il}))) + \mathbf{q}(x_{ij})) \tag{3.14d}$$

ou, ainda

$$\frac{\mathbf{y}_{i+1} - \mathbf{y}_i}{h_i} = \sum_{j=1}^{k} \beta_j \mathbf{f}_{ij}$$

onde

$$\mathbf{f}_{ij} = A(x_{ij})(\mathbf{y}_i + h_i \sum_{l=1}^{k} \alpha_{jl}(A(x_{il})\mathbf{y}_{il} + \mathbf{q}(x_{il}))) + \mathbf{q}(x_{ij}). \tag{3.14e}$$

Escrevendo (3.14e) na forma matricial, tem-se

$$W_i \mathbf{f}_i = V_i \mathbf{y}_i + \mathbf{q}_i$$

onde

$$W_i = I - h_i \begin{bmatrix} \alpha_{11}A(x_{i1}) & \cdots & \alpha_{1k}A(x_{i1}) \\ \vdots & & \vdots \\ \alpha_{k1}A(x_{ik}) & \cdots & \alpha_{kk}A(x_{ik}) \end{bmatrix},$$

$$V_i = \begin{bmatrix} A(x_{i1}) \\ \vdots \\ A(x_{ik}) \end{bmatrix}, \qquad \mathbf{f}_i = \begin{bmatrix} \mathbf{f}_{i1} \\ \vdots \\ \mathbf{f}_{ik} \end{bmatrix}, \qquad \mathbf{q}_i = \begin{bmatrix} \mathbf{q}(x_{i1}) \\ \vdots \\ \mathbf{q}(x_{ik}) \end{bmatrix},$$

onde $W_i \in \mathbb{R}^{nk \times nk}$, $V_i \in \mathbb{R}^{nk \times n}$ e $\mathbf{f}_i, \mathbf{q}_i \in \mathbb{R}^{nk \times 1}$.

Para h_i suficientemente pequeno, existe $W_i^{-1} = I + O(h_i)$, e a expressão (3.14d) pode ser reescrita na forma

$$\mathbf{y}_{i+1} = \Gamma_i \mathbf{y}_i + \mathbf{r}_i, \qquad 1 \leq i \leq N, \tag{3.14f}$$

onde

$$\Gamma_i = I + h_i B W_i^{-1} V_i \qquad \text{e} \qquad \mathbf{r}_i = h_i B W_i^{-1} \mathbf{q}_i. \quad \text{e} \quad B = [\beta_1 I \quad \cdots \quad \beta_k I].$$

Adicionando as condições de contorno, recai-se em um sistema de equações algébricas como o do método do *shooting* múltiplo. Portanto, este método pode ser considerado como um

método de *shooting* múltiplo onde as resoluções dos PVIs, em cada subintervalo da malha Π, são feitas pelo método de Runge-Kutta, ou que no método de *shooting* múltiplo, a integração dos PVIs, de um ponto de *shooting* para outro, pode ser vista como um processo de eliminação de parâmetros locais.

Para estabelecer a convergência para o método implícito de Runge-Kutta, necessita-se de um limite para o erro global ($e_i = ||\mathbf{y}_i - \mathbf{y}_i(x_i)||$) do tipo $e_i \leq K\, c\, h^p$, $h = \max_i h_i$. Para estabelecer a precisão requerida faz-se a seguinte restrição ao método: os pontos de colocação $\rho_j, j = 1, \ldots, k$, são distintos, ou seja,

$$0 < \rho_1 < \rho_2 < \cdots < \rho_k < 1.$$

Neste caso os valores α_{jl} e β_j são determinados de maneira única da seguinte forma: define-se em cada subintervalo da malha $\Pi, [x_i, x_{i+1}]$, os pontos $x_{ij} = x_i + h_i\rho_j$ e escreve-se $\mathbf{y}'(x)$ como uma soma de interpolantes de Lagrange de ordem k mais o resto Ψ,

(3.15a)
$$\mathbf{y}'(x) = \sum_{l=1}^{k} \mathbf{y}'(x_{il}) L_l\left(\frac{x - x_i}{h_i}\right) + \Psi(x),$$

onde

$$L_l(t) = \frac{\prod_{1 \leq j \neq l < k}(t - \rho_j)}{\prod_{1 \leq j \neq l \leq k}(\rho_l - \rho_j)}, \qquad 1 \leq l \leq k$$

e o resto $\Psi(x) = \mathbf{y}'[x_{i1}, \ldots, x_{ik}, x] \prod_{l=1}^{k}(x - x_{il})$, onde $g[x_{i1}, \ldots, x_{ik}, x]$ é a k-ésima diferença dividida de g conforme Ascher [1986] e Ascher et al.[1988].

Portanto,

$$\begin{aligned}\mathbf{y}(x) - \mathbf{y}(x_i) &= \int_{x_i}^{x} \mathbf{y}'(t)\, dt = \int_{x_i}^{x} \mathbf{f}(t, \mathbf{y}(t))\, dt \\ &= \sum_{l=1}^{k} \mathbf{y}'(x_{il}) \int_{x_i}^{x} \left(L_l\left(\frac{x - x_i}{h_i}\right) + \Psi(x)\right) dx,\end{aligned}$$

e tomando

$$\mathbf{f}_{ij} = \mathbf{y}'(x_{ij})$$

$$\text{e} \qquad \beta_l = \int_{x_i}^{x} \left(L_l\left(\frac{x - x_i}{h_i}\right) + \Psi(x)\right) dx$$

obtemos o método implícito de Runge-Kutta com $\beta_j = \int_0^1 L_j(t)\, dt$ e $\alpha_{jl} = \int_0^{\rho_j} L_l(t)\, dt$, $1 \leq j \neq t \leq k$.

A precisão do método também depende da escolha dos pontos de colocação. Por exemplo, se os pontos escolhidos forem os pontos de Gauss a precisão do método será $O(h^{2k})$.

EXEMPLO 3.5 - Considere o PVC 3.1

Tomando em (3.14a) $k = 2$ e $\rho_1 = \frac{1}{3}$ e $\rho_2 = \frac{2}{3}$, tem-se $x_{i1} = x_i + \frac{1}{3}h_i$, e $x_{i2} = x_i + \frac{2}{3}h_i$, onde x_i é um ponto da malha $\Pi : 0 \leq x_1 < \cdots < x_{N+1} \leq 1$ dada, então (3.15a) tem a forma $L_1(t) = \frac{t-\frac{2}{3}}{-\frac{1}{3}}$ e $L_2(t) = \frac{t-\frac{1}{3}}{\frac{1}{3}}$, e $\beta_1 = \beta_2 = \frac{1}{2}$, $\alpha_{11} = \frac{1}{2}$, $\alpha_{12} = -\frac{1}{6}$, $\alpha_{11} = \frac{2}{3}$, $\alpha_{11} = 0$,

As matrizes de (3.14e) são

$$W = \begin{bmatrix} 1 & -\frac{h_i}{2} & 0 & \frac{h_i}{6} \\ -\frac{h_i\lambda^2}{2} & 1 & \frac{h_i\lambda^2}{6} & 0 \\ 0 & -\frac{2h_i}{3} & 1 & 0 \\ -\frac{2\lambda^2 h_i}{3} & 0 & 0 & 1 \end{bmatrix}, \quad V = \begin{bmatrix} 0 & 1 \\ \lambda^2 & 0 \\ 0 & 1 \\ \lambda^2 & 0 \end{bmatrix}, \quad \mathbf{q}_i = \begin{bmatrix} 0 \\ (1-\lambda^2)e^{x_{i1}} \\ 0 \\ (1-\lambda^2)e^{x_{i2}} \end{bmatrix}. \quad \triangle$$

3.3.2 - Método da Colocação para PVC de primeira ordem.

Uma solução aproximada $\mathbf{y}(x)$ de (2.1a) determinada pelo método da colocação consiste em encontrar um função $\mathbf{y}_\Pi(x)$ que coloca a solução em pontos predeterminados do intervalo, denominados pontos de colocação.

Seja $\mathbf{y}_\Pi(x)$ um polinômio de grau $k+1$, definido em $[x_i, x_{i+1}]$ pelas condições de interpolação

$$\begin{aligned} \mathbf{y}_\Pi(x_i) &= \mathbf{y}_i = \mathbf{y}(x_i) \\ \mathbf{y}'_\Pi(x_{ij}) &= f(x_{ij}, \mathbf{y}_{ij}), \qquad 1 \leq j \leq k \end{aligned}$$

onde $x_{ij} = x_i + \rho_j h_i$, $h_i = x_{i+1} - x_i$, $0 \leq \rho_1 < \rho_2 < \cdots < \rho_k \leq 1$, $\mathbf{y}_{ij} = \mathbf{y}_i + h_i \sum_{l=1}^{k} \alpha_{jl}\mathbf{f}_{il}$, Escrevendo $\mathbf{y}_\Pi(x)$ em termos de sua derivada

$$\mathbf{y}_\Pi(x) - \mathbf{y}_\Pi(x_i) = \int_{x_i}^{x} \mathbf{y}'_\Pi(t)dt$$

e substituindo $\mathbf{y}'_\Pi(x)$ por (3.15a) e usando (3.14a) obtem-se

$$\begin{aligned} \mathbf{y}_\Pi(x_{ij}) &= \mathbf{y}_{ij}, \\ \mathbf{y}_\Pi(x_{i+1}) &= \mathbf{y}_{i+1}. \end{aligned}$$

Se a função polinomial for estendida para o intervalo $[x_{i+1}, x_{i+2}]$ de forma idêntica, obtém-se uma função polinomial por partes continua em x_{i+1}. Estendendo para todo i, $i =$

$1, \ldots, N$, obtém-se uma função polinomial por partes contínua de ordem $k+1$ (grau $< k+1$) em $[a,b]$, que satisfaz a EDO nos pontos de colocação, isto é:

$$\text{(3.16a)} \qquad \begin{cases} \mathbf{y}'_\Pi(x_{ij}) &= \mathbf{f}(x_{ij}, \mathbf{y}_\Pi(x_{ij})), \\ \mathbf{y}_\Pi(x_{ij}) &= \mathbf{y}_{ij}, \end{cases}$$

e também as condições de contorno

$$\text{(3.16b)} \qquad \mathbf{g}(\mathbf{y}_\Pi(a), \mathbf{y}_\Pi(b)) = 0.$$

A função polinomial por partes, contínua , $\mathbf{y}_\Pi(x)$, que satisfaz a EDO nos pontos de colocação (3.16a) e as condições de contorno (3.16b) é denominada solução colocada de (2.1a).

TEOREMA. *O método implícito de Runge-Kutta com as restrições:* $0 \le \rho_1 < \cdots < \rho_k \le 1$ *é equivalente ao método de colocação definido acima. Além disso,*

$$\mathbf{y}_\Pi(x_i) = \mathbf{y}_i, \qquad \mathbf{y}_\Pi(x_{ij}) = \mathbf{y}_{ij}, \qquad 1 \le i \le N, \quad 1 \le j \le k. \qquad \Delta$$

A existência de um solução colocada, para problemas lineares (2.3a), $\mathbf{y}_\Pi(x)$ ou de Runge-Kutta, $\mathbf{y}_\Pi$ é garantida pelos seguintes resultados, demonstrados em Ascher et al.[1988].

TEOREMA. *A solução colocada de k estágios existe para o problema (2.3a) e é obtida de forma estável. Além disso, o erro e suas derivadas satisfazem:*

$$\max_{x_i \le x \le x_{i+1}} ||\mathbf{e}_\Pi(x)|| \le O(h^k) \qquad e$$

$$\max_{x_i \le x \le x_{i+1}} ||\mathbf{e}_\Pi^{(j)}(x)|| = O(h^{k+1-j})(\frac{h}{h_i})^{j-1}, \qquad 1 \le j \le k,$$

onde $\mathbf{e}_\Pi(x) = \mathbf{y}_\Pi(x) - \mathbf{y}(x)$ *e* $h = \max_i h_i$. Δ

TEOREMA. *Se o método de Runge-Kutta satisfaz as restrições dadas acima, então a precisão é de ordem* $O(h^p)$ *para o PVC linear satisfazendo* $A(x), \mathbf{q}(x) \in C^{(p)}[a,b]$, *portanto,*

$$||\mathbf{y}_i - \mathbf{y}(x_i)|| = O(h^p), \qquad 1 \le i \le N$$

e, também, nos pontos de colocação

$$||\mathbf{y}_{ij} - \mathbf{y}(x_{ij})|| = O(h_i^{k+1}) + O(h^p). \qquad \Delta$$

Pode-se observar, que para h pequeno, o erro nos pontos da malha são particularmente pequenos quando comparados com o erro nos pontos de colocação para $p > k+1$ e $h = O(h_i)$.

Considerando, agora, problemas não lineares

$$\text{(3.17a)} \qquad \begin{cases} \mathbf{y}' = \mathbf{f}(x, \mathbf{y}) \\ \mathbf{g}(\mathbf{y}(a), \mathbf{y}(b)) = 0 \end{cases}$$

Uma solução colocada, $\mathbf{y}_\Pi(x)$, pode ser encontrada usando o processo de *quasilinearização*, descrito a seguir.

Pelo processo de linearização em uma vizinhança de uma solução aproximada conhecida $\mathbf{y}_\Pi^m(x)$, recai-se em um problema linear em $\mathbf{z}(x)$,

$$\text{(3.17b)} \qquad \begin{cases} \mathbf{z}' - A(x)\mathbf{z} = -((\mathbf{y}^m)'(x) - \mathbf{f}(x, \mathbf{y}^m(x))) \\ B_a\mathbf{z}(a) + B_b\mathbf{z}(b) = -g(\mathbf{y}^m(a), \mathbf{y}^m(b)) \end{cases}$$

onde

$$A(x) = \frac{\partial \mathbf{f}(x, \mathbf{y}^m(x))}{\partial \mathbf{y}}, \qquad B_a = \frac{\partial \mathbf{g}(\mathbf{y}^m(a), \mathbf{y}^m(b))}{\partial \mathbf{y}(a)}, \qquad B_b = \frac{\partial \mathbf{g}(\mathbf{y}^m(a), \mathbf{y}^m(b))}{\partial \mathbf{y}(b)},$$

e uma solução aproximada da forma

$$\mathbf{y}^{m+1}(x) = \mathbf{y}^m(x) + \mathbf{z}(x)$$

é procurada até que $||\mathbf{z}(x)||$ satisfaça uma tolerância dada, ou algum limite do número de iterações for excedido.

De acordo com Ascher et al.[1988] o método de *quasilinearização* é semelhante ao método de Newton e converge quadraticamente, se a aproximação inicial for suficientemente próxima da solução exata.

TEOREMA. *Seja $\mathbf{y}(x)$ uma solução isolada de (3.17a) com as condições de suavidade necessárias. Então, para cada estágio do método de colocação, existem constantes positivas ρ e h_o tal que para toda malha com $h \leq h_o$, tem-se que:*

i) Existe uma única solução $\mathbf{y}_\Pi(x)$ para as equações de colocação (3.16a) em uma região de raio ρ e centro $\mathbf{y}_\Pi(x)$;

ii) A solução $\mathbf{y}_\Pi(x)$ pode ser obtida pelo método de Newton (ou quasilinearização), que converge quadraticamente, desde que a solução inicial dada $\mathbf{y}_\Pi^o(x)$ esteja suficientemente próxima de $\mathbf{y}(x)$;

iii) O erro estimado será:

$$||\mathbf{y}_i - \mathbf{y}(x_i)|| = O(h^p), \qquad 1 \le i \le N$$
$$||\mathbf{y}_\Pi(x) - \mathbf{y}(x)|| = O(h_i^{k+1}) + O(h^p), \quad x_i \le x \le x_{i+1}, \quad 1 \le i \le N. \qquad \triangle$$

EXEMPLO 3.6 - Considere, novamante, o PVC não linear do exemplo 3.4, e sua conversão em um sistema de primeira ordem.

$$\begin{cases} y_1' = y_2, \\ y_2' = -e^{y_1}, \\ y_1(0) = y_1(1) = 0 \end{cases}$$

onde $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$, e $\mathbf{f}$ e $\mathbf{g}$ de (3.17a) tem a forma $\mathbf{f}(x, \mathbf{y}) = \begin{bmatrix} y_2 \\ -e^{y_1} \end{bmatrix}$ e $\mathbf{g} = \begin{bmatrix} y_1(0) \\ y_1(1) \end{bmatrix}$. A solução aproximada em cada ponto da malha é representada por

$$\mathbf{y}_i^m = \begin{bmatrix} y_{1_i}^m \\ y_{2_i}^m \end{bmatrix}, \qquad 1 \le i \le N+1$$

e (3.17b) tem a forma, linear em $\mathbf{z}$,

$$\frac{\mathbf{z}_{i+1} - \mathbf{z}_i}{h_i} - \frac{1}{2}\begin{bmatrix} 0 & 1 \\ -e^{y_{1_i}^m} & 0 \end{bmatrix}\mathbf{z}_i = -\left(\frac{\mathbf{y}_{i+1}^m - \mathbf{y}_i^m}{h_i} - \mathbf{f}(x, \mathbf{y}^m)\right), \quad 1 \le i \le N,$$
$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\mathbf{z}_1 + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\mathbf{z}_{N+1} = -\begin{bmatrix} y_{1_1}^m \\ y_{1_{N+1}}^m \end{bmatrix}. \qquad \triangle$$

3.3.3 - Método da Colocação para PVC de ordem superior.

Em muitas aplicações, EDOs aparecem na forma de equações de ordem superior, e às vezes a transformação dessas equações em sistemas de primeira ordem não é conveniente.

Considere o PVC/EDO de ordem superior ($m > 1$), linear

(3.18a) $$Lu \equiv u^{(m)} - \sum_{l=1}^{m} c_l(x) u^{(l-1)} = q(x), \quad a < x < b,$$

ou não linear

(3.18b) $$Nu \equiv u^{(m)} - f(x, u, u', \ldots, u^{(m-1)}) = 0, \quad a < x < b,$$

cuja solução $\mathbf{y}(x)$ é denotada por:

$$\mathbf{y}(x) = [u(x)\ u'(x)\ \ldots\ u^{(m-1)}(x)]^t;$$

e as condições de contorno

(3.18c) $$B_a\mathbf{y}(a) + B_b\mathbf{y}(b) = \mathbf{d}.$$

ou

(3.18d) $$\mathbf{g}(\mathbf{y}(a), \mathbf{y}(b)) = 0,$$

Assumindo a existência de uma única solução u, e que os coeficientes $c_i(x)$ do operador linear L sejam suaves, então u geralmente possui m derivadas contínuas a mais que o termo não homogeneo, $q(x)$. Se q for uma função contínua por partes, então $u \in C^{(m-1)}[a,b]$.

Nesta seção será apresentada uma versão do método da colocação para EDO de ordem superior. De uma forma geral, dada uma malha Π em $[a,b]$, o método da colocação procura uma solução colocada da forma:

(3.18e) $$u_\Pi(x) = \sum_{j=1}^{J} \alpha_j \Phi_j(x), \qquad a \le x \le b,$$

onde $\Phi_j(x)$ são funções linearmente independentes conhecidas e definidas em $[a,b]$ e α_j são parâmetros que são determinados pela exigência de $u_\Pi(x)$ satisfazer as m condições de contorno e a EDO nos Nk pontos de colocação em $[a,b]$. Pode-se dizer que $u_\Pi(x) \in P_{k+m,\Pi} \cap C^{(m-1)}[a,b]$, isto é, é um elemento de um espaço de dimensão $J = Nk + m$ que é gerado pelas funções $\Phi_1(x), \Phi_2(x), \ldots, \Phi_J(x)$, onde $P_{k+m,\Pi}$ é o espaço das funções polinomiais por partes, contínuas em $[a,b]$ de ordem $k+m$ para algum $k \ge m$ (grau $< k+m$) e as funções $\Phi_j(x)$ são de classe $C^{(m-1)}[a,b]$ (como a solução exata). Em outras palavras, dada uma malha Π de $[a,b]$ e k pontos de colocação em $[x_i, x_{i+1}]$, onde $x_{ij} = x_i + h_i\rho_j$, $0 \le \rho_1 < \rho_2 < \cdots < \rho_k \le 1$, $1 \le j \le k$, $1 \le i \le N$, a solução colocada $u_\Pi(x)$, é um polinômio de ordem $k+m$ em cada subintervalo de Π e deve satisfazer os Nk pontos de colocação e as m condições de contorno.

A eficiência desse método depende de dois fatores. A escolha das funções bases, $\Phi_j(x)$ de (3.18e) e da escolha da malha Π. As funções bases, como já foi mencionado, são funções contínuas e polinomiais por partes. Nessa classe de funções serão destacadas as bases monomiais e as B-splines.

Bases Monomiais: São funções definidas usando uma representação local de polinômios por partes, isto é, envolvem somente um subintervalo da malha.

A solução $u_\Pi(x)$ da EDO (3.18), para $x_i \leq x \leq x_{i+1}$ pode ser expressa em termos de sua série de Taylor nas vizinhanças de x_i

$$u_\Pi(x) = \sum_{j=1}^{k+m} \frac{(x-x_i)^{j-1}}{(j-1)!} u_\Pi^{(j-1)}(x_i),$$

que pode ser reescrita como segue

$$u_\Pi(x) = \sum_{j=1}^{m} \frac{(x-x_i)^{j-1}}{(j-1)!} y_{ij} + h_i^m \sum_{j=1}^{k} \Phi_j(\frac{x-x_i}{h_i}) z_{ij} \tag{3.19a}$$

onde

$$\begin{aligned}
&y_{ij} = u_\Pi^{(j-1)}(x_i), \quad \text{e} \quad \mathbf{y}_i = (y_{i1} \ \ldots \ y_{im}),\\
&z_{ij} = h_i^{j-1} u_\Pi^{(m+j-1)}(x_i), \quad \text{e} \quad \mathbf{z}_i = (z_{i1} \ \ldots \ z_{ik}),\\
&\mathbf{y}_\Pi(x) = [u_\Pi, u_\Pi' \ \ldots \ u_\Pi^{(m-1)}(x)]^t,\\
&\mathbf{y}_\Pi(x_i) \equiv \mathbf{y}_i
\end{aligned}$$

e seja

$$\Phi_j(x) = \frac{x^{m+j-1}}{(m+j-1)!}, \qquad 1 \leq j \leq k, \quad 0 \leq x \leq 1,$$

polinômios de ordem $k+m$ satisfazendo $\Phi_j^{(l-1)}(0) = 0$, $1 \leq l \leq m$, $1 \leq j \leq k$.

Pode-se observar que tomando $m = 1$ recai-se no método implícito de Runge-Kutta para $u_\Pi(x_{i+1})$.

Reescrevendo o problema linear (3.18a) com a solução u_Π definida em (3.19a), tem-se

$$Lu_\Pi(x) = -\sum_{l=1}^{m} c_l(x) \sum_{j=1}^{m} \frac{(x-x_i)^{j-l}}{(j-l)!} y_{ij} + h_i^m \sum_{j=1}^{k} L\Phi_j(\frac{x-x_i}{h_i}) z_{ij}$$

e, nos pontos de colocação $x_{ij} = x_i + h_i\rho_j$, tem-se a seguinte notação matricial

$$V\mathbf{y}_i + W\mathbf{z}_i = \mathbf{q}_i, \tag{3.19b}$$

onde $\mathbf{q}_i = [q(x_{i1}) \ \ldots \ q(x_{ik})]^t$, e $V_{k\times m}$, $W_{k\times k}$ são matrizes com elementos

$$v_{rj} = -\sum_{l-1}^{j} c_l(x_{ir})\frac{(h_i\rho_r)^{j-l}}{(j-l)!}, \quad 1 \le r \le k, \quad 1 \le j \le m,$$

$$w_{rj} = \Phi_j^{(m)}(\rho_r) - \sum_{l=1}^{m} c_l(x_{ir}) h_i^{m+1-l}\Phi_j^{(l-1)}(\rho_r), \quad 1 \le r, j \le k,$$

respectivamente.

Para estabelecer as condições de continuidade da solução $u_\Pi(x)$, avalia-se $u_\Pi(x)$ e suas $m-1$ derivadas em $x = x_{i+1}$ por (3.19a), e iguala os valores correspondentes a

$$\mathbf{y}_{i+1} = C\mathbf{y}_i + D\mathbf{z}_i, \qquad 1 \le i \le N \tag{3.19c}$$

onde $C_{m\times m}$ é uma matriz triangular superior com elementos

$$c_{rj} = \frac{h_i^{j-r}}{(j-r)!}, \qquad j \ge r \quad 1 \le r \le m,$$

e $D_{m\times k}$ é uma matriz com elementos

$$d_{rj} = h_i^{m+1-r}\Phi_j^{(r-1)}(1), \quad 1 \le r \le m, \quad 1 \le j \le k.$$

As restrições da solução são completadas pelas condições de contorno

$$B_a\mathbf{y}_1 + B_b\mathbf{y}_{N+1} = \mathbf{d}.$$

Conceitualmente, este processo e o método implícito de Runge-Kutta não apresentam diferenças; portanto, o próximo passo é eliminar variáveis locais, como foi feito no método de Runge-Kutta.

Pode-se observar que, quando $h \to 0$,

$$w_{rj} \to \frac{\rho_r^{j-1}}{(j-1)!} = w_{rj}(0)$$

onde a matriz $W(0)$ é uma matriz de Vandermonde. Portanto, para h_i pequeno, W é não singular e

$$W^{-1} = W^{-1}(0) + O(h_i).$$

Usando (3.19b,c) pode-se eliminar $\mathbf{z}_i$, obtendo

$$\mathbf{y}_{i+1} = \Gamma_i\mathbf{y}_i + \mathbf{r}_i \qquad 1 \le i \le N,$$

onde

$$\Gamma_i = C - DW^{-1}V, \qquad \mathbf{r}_i = DW^{-1}\mathbf{q}_i$$
$$C = I + O(h_i) \quad \text{e} \quad D = O(h_i)$$

Novamente, recai-se em um sistema de equações lineares em $u_\Pi(x)$ e suas $m-1$ primeiras derivadas nos pontos da malha. Após obter $\mathbf{y}_i$ pode-se obter $\mathbf{z}_i$ e, então, a solução $u_\Pi(x)$.

EXEMPLO 3.7 - Considerando PVC linear do exemplo 3.1,

$$\begin{cases} u'' = \lambda^2 u + (1-\lambda^2)e^x, & 0 < x < b, \\ u(0) = 1, \quad u(b) = e^b, \end{cases}$$

Tomando uma malha de N subintervalo, com $N = 1$, e $k = 3$ pontos de colocação, tem-se $x_{1j} = \rho_j$, $0 < \rho_1 = \frac{1}{4} < \rho_2 = \frac{1}{2} < \rho_3 = \frac{3}{4} < 1$ e $h_1 = 1$.

A solução

$$u_\Pi(x) = \sum_{j=1}^{2} \frac{(x-x_1)^{j-1}}{(j-1)!} y_{1j} + h_1^m \sum_{j=1}^{3} \Phi_j(\frac{x-x_1}{h_1}) z_{1j}$$

com $\Phi_j(t) = \frac{t^{1+j}}{(1+j)!}$, $1 \le j \le k$, $0 < t < 1$, deve satisfazer a EDO nos pontos de colocação, isto é,

$$u''_\Pi(\rho_i) - \lambda^2 u_\Pi(\rho_i) = (\lambda^2 - 1)\, e^{\rho_i}, \qquad 1 \le i \le 3. \quad \text{ou seja}$$
$$\sum_{j=1}^{3} \Phi''_j(\rho_i) z_{1j} - \lambda^2 (\sum_{j=1}^{2} \frac{{\rho_i}^{j-1}}{(j-1)!} y_{1j} + \sum_{j=1}^{3} \Phi_j(\rho_i) z_{1j}) = (\lambda^2 - 1)\, e^{\rho_i},$$

onde as matrizes $V_{3\times 2}$, $W_{3\times 3}$ dadas em (3.19b) tem a forma

$$W = \begin{bmatrix} 1 - \frac{1}{2!\,4^2}\lambda^2 & \frac{1}{4} - \frac{1}{3!\,4^3}\lambda^2 & \frac{1}{2!\,4^2} - \frac{1}{4!\,4^4}\lambda^2 \\ 1 - \frac{1}{2!\,2^2}\lambda^2 & \frac{1}{2} - \frac{1}{3!\,2^3}\lambda^2 & \frac{1}{2!\,2^2} - \frac{1}{4!\,2^4}\lambda^2 \\ 1 - \frac{9}{2!\,4^2}\lambda^2 & \frac{3}{4} - \frac{3^3}{3!\,4^3}\lambda^2 & \frac{3^2}{2!\,4^2} - \frac{3^4}{4!\,4^4}\lambda^2 \end{bmatrix}, \quad V = \begin{bmatrix} -\lambda^2 & \frac{-\lambda^2}{4} \\ -\lambda^2 & \frac{-\lambda^2}{2} \\ -\lambda^2 & \frac{-3\,\lambda^2}{4} \end{bmatrix},$$

e as condições de continuidade em x_{i+1}

$$u_\Pi(x_2) = \sum_{j=1}^{2} \frac{1}{(j-1)!} y_{1j} + \sum_{j=1}^{3} \Phi_j(1) z_{1j}$$

e

$$u'_{\Pi}(x_2) = y_{12} + \sum_{j=1}^{3} \Phi'_j(1) z_{1j}$$

gerando as matrizes $C_{2\times 2}$ e $D_{2\times 3}$ dadas em (3.19c) tem a forma

$$C = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad D = \begin{bmatrix} \frac{1}{2} & \frac{1}{6} & \frac{1}{24} \\ 1 & \frac{1}{2} & \frac{1}{6} \end{bmatrix}. \qquad \Delta$$

OBSERVAÇÃO. *A representação (3.19a) para $u_{\Pi}(x)$ não é única. Outras escolhas de $\Phi_j(t)$ podem ser consideradas, desde que $\Phi_j^{(l-1)}(0) = 0$, $1 \le l \le m$, $1 \le j \le k$. Por exemplo, Φ pode ser definida de tal forma que $\Phi_j^{(m)}(\rho_r) = \delta_{jr}$, $1 \le j, r \le k$. Neste caso, $z_{ij} = u_{\Pi}(x_{ij})$ e, para $m = 1$, tem-se, $\Phi_j(\rho_r) = \alpha_{rj}$ e $\Phi_j(1) = \beta_j$. Ou ainda, pode-se definir Φ tal que $\Phi_j^{(r-1)}(1) = \delta_{j-k+m,r}$, $1 \le j, r \le k$. O importante nessa escolha é que a matriz D de (3.19c) seja simples conforme Ascher [1986].* Δ

O problema (2.19) pode ser convertido em um sistema equivalente de primeira ordem, se os mesmos k pontos de colocação forem usados. Cada componente de $\mathbf{y}_{\Pi}(x)$ é uma função polinomial por partes contínua de ordem $(k+1)$. O número de parâmetros livres em $\mathbf{y}_{\Pi}(x)$, antes de satisfazer as condições de continuidade e de colocação será $N\,k\,m$, enquanto que de $u_{\Pi}(x)$ é somente $N(k+m)$. Portanto, usar o método para manusear diretamente equações de ordem superior é mais eficiente conforme Ascher et al.[1988].

Mas pode-se usar a teoria já desenvolvida para sistemas de equações de primeira ordem, uma vez que existe a equivalência. Está provado em Bader e Ascher [1987] que assumindo a existência de inteiros $p \ge k \ge m$ tais que se o problema (3.18a,c) seja bem condicionado (isto é, o sistema equivalente de primeira ordem tenha uma constante de condicionamento de valor moderado), tenha coeficientes em $C^{(p)}[a,b]$, tenha uma única solução $u(x)$ em $C^{p+m}[a,b]$ e os k pontos de colocação $\rho_1, \ldots, \rho_k$ sejam os pontos de Gauss. Então para h pequeno o método de colocação descrito acima é estável com constante de estabilidade ckN, onde c é uma constante de valor moderado, e tem uma única solução $u_{\Pi}(x)$. Além disso, as seguintes estimativas de erros valem para os pontos da malha

(3.20a) $$||u^{(j)}(x) - u_{\Pi}^{(j)}(x_i)|| = O(h^p), \quad 1 \le j \le m-1, \quad 1 \le i \le N+1,$$

e para qualquer x em [a,b],

(3.20b)

$$||u^{(j)}(x) - u_{\Pi}^{(j)}(x)|| = h_i^{k+m-j} u^{(k+j)}(x_i) P^{(j)}\left(\frac{x - x_i}{h_i}\right) + O(h_i^{k+m-j+1}) + O(h^p),$$

$$x_i \le x \le x_{i+1}, \quad 1 \le i \le N, \quad 0 \le j \le k+m-1,$$

onde $P(\xi) = \frac{1}{k!(m-1)!} \int_0^{\xi} (t-\xi)^{m-1} \prod_{l=1}^{k} (t-\rho_l)\, dt$.

Considere, agora, o problema não linear (3.18b,d). Dada uma solução inicial $u_\Pi(x)$, e resolvendo, repetidamente, o problema linearizado,

$$\begin{cases} w^{(m)} - \sum_{l=1}^{m} \frac{\partial f(x,\mathbf{y}(u))}{\partial \mathbf{y}_l(u)} w^{(l-1)}(x), = -[u_\Pi^{(m)}(x) - f(x, \mathbf{y}_\Pi(x))], \quad a < x < b, \\ B_a \mathbf{z}(a) + B_b \mathbf{z}(b) = -\mathbf{g}(\mathbf{y}_\Pi(a), \mathbf{y}_\Pi(b)), \end{cases}$$

onde, $\mathbf{z}(x) = [w(x)\ w'(x)\ \ldots\ w^{(m-1)}(x)]^t$,

$$B_a = \frac{\partial g(\mathbf{y}(a), \mathbf{y}(b))}{\partial \mathbf{y}(a)} \qquad \text{e} \qquad B_b = \frac{\partial g(\mathbf{y}(a), \mathbf{y}(b))}{\partial \mathbf{y}(b)},$$

uma solução aproximada da forma

$$\mathbf{y}_\Pi^{s+1}(x) = \mathbf{y}_\Pi^{s}(x) + \mathbf{z}_\Pi(x)$$

é procurada até que $||\mathbf{z}_\Pi(x)||$ satisfaça alguma tolerância dada.

Esse método de *quasilinearização* é assegurado em Ascher et al.[1988] que converge quadraticamente, desde que a aproximação inicial seja suficientemente próxima de $u(x)$, como já foi mencionado no estudo de sistemas de primeira ordem.

EXEMPLO 3.8 - Considere, novamente, o PVC não linear

$$\begin{cases} u'' + e^u = 0, \qquad 0 < x < 1, \\ u(0) = u(1) = 0, \end{cases}$$

usando o método de *quasilinearização* obtém-se

$$\begin{cases} w'' + e^u\, w = -(u'' + e^u), \qquad 0 < x < 1, \\ w(0) = w(1) = 0. \end{cases}$$

que é um problema linear em relação a w.

Tomando uma malha de N subintervalo, com $N = 1$, e $k = 3$ pontos de colocação, tem-se $x_{1j} = \rho_j$, $0 < \rho_1 = \frac{1}{4} < \rho_2 = \frac{1}{2} < \rho_3 = \frac{3}{4} < 1$ e $h_1 = 1$.

As matrizes $V_{3\times 2}$, $W_{3\times 3}$ dadas em (3.19b) e as matrizes $C_{2\times 2}$ e $D_{2\times 3}$ dadas em (3.19c) são identicas às do exemplo 3.7, uma vez que a parte homogênea dos problemas difere apenas por uma constante.

Bases B-Splines: Aproximar a solução u_Π por B-splines de ordem $k+m$, usando colocação, significa determinar uma função no espaço $P_{k+m,\Pi}$ que satisfaça as condições

de continuidade requeridas e as equações de discretização (de colocação) relacionadas com a solução exata da EDO, $Lu_\Pi(x_{ij}) = q_{ij}$ e $B_a y_\Pi(a) + B_b y_\Pi(b) = \mathbf{d}$, ou seja, deve-se encontrar funções Φ_i tais que $u_\Pi(x) = \sum_{i=1}^{Nk+m} \alpha_i \Phi_i(x)$, recaindo em um sistema linear $A\mathbf{a} = \mathbf{b}$, onde A é uma matriz banda.

De acordo com de Boor [1978], para gerar uma base de splines de ordem $k+m$ (B-splines) para o espaço $P_{k+m,\Pi}$ define-se uma sequência $\{t_j\}_{j=1}^{k(N+1)+2m}$ não decrescente a partir de uma sequência estritamente crescente $\Pi = \{x_i\}_{i=1}^{N+1}$ de tal forma que

i) $t_1 \leq t_2 \leq \cdots \leq t_{k+m} \leq x_1$ e $x_{N+1} \leq t_{Nk+m+1} \leq \cdots \leq t_{(N+1)k+2m}$,

ii) para $i = 2, \ldots, N$, x_i ocorrem k vezes em $\{t_j\}_{j=k+m+1}^{Nk+m}$.

Define-se a i-ésima B-spline de ordem $k+m$ para uma sequência não decrescente $\{t_i\}_{i=1}^{(N+1)k+2m}$ por

$$B_i \equiv B_{i,k+m,\Pi}(x) = (t_{i+k+m} - t_i)(\cdot - x)_+^{k+m-1}[t_i, \ldots, t_{i+k+m}]$$

para todo $x \in \mathbb{R}$, onde

$p[t_i, t_{i+1}, \ldots, t_{i+k+m}]$ é a (k+m)-ésima diferença dividida de p,

$(t-x)_+ = \max\{t-x, 0\}$

e $(t)_+^r = (t_+)^r$.

Propriedades:

1. B_i possui um suporte pequeno, isto é, $B_i(x) = 0$ para todo $x \notin [t_i, t_{i+k+m}]$; somente $k+m$ B-splines em qualquer subintervalo $[t_j, t_{j+1}]$ estão em seus suportes, ou seja, $B_i(x) \neq 0$, $x \in [t_j, t_{j+1}]$, $i = 1, \ldots, k+m$.
2. $\sum_i B_i = \sum_{i=r+1-(k+m)}^{s-1} B_i(x) = 1$ para todo $x \in (t_r, t_s)$
3. $B_i(x) > 0$ para todo $x \in (t_i, t_{i+k+m})$ (isto é, B_i é positivo em seu suporte).

Uma função spline de ordem $k+m$ é qualquer combinação linear de B-splines de ordem $k+m$ para a sequência de pontos $\{t_i\}$.

Uma das vantagens de B-splines é o fato das condições de continuidade já estarem embutidas, resultando que somente as equações de discretização devem ser satisfeitas explicitamente. Outra vantagem é o seu pequeno suporte compacto, e também que alguns B-splines e suas derivadas são independentes da malha, facilitando com isso sua avaliação. Infelizmente, as matrizes de discretização para EDO de ordem superior possuem número de condição com um rápido crescimento com o refinamento da malha conforme Ascher et al.[1983].

EXEMPLO 3.9 - Considerando o PVC do exemplo 3.1,

$$\begin{cases} u'' = \lambda^2 u + (1-\lambda^2)e^x, & 0 < x < b, \\ u(0) = 1, \quad u(b) = e^b, \end{cases}$$

uma solução usando B-spline tem a forma

$$u_\Pi(x) = \sum_{j=1}^{Nk+m} \alpha_j \Phi_j(x),$$

tomando $k = 3$ e $N = 2$, $Nk + m = 8$ e $\{t_j\}_{j=1}^{(N+1)k+2m}$, onde $t_1 = \cdots = t_5 \leq 0 < t_6 = t_7 = t_8 = \frac{1}{2} < 1 = t_9 = \cdots = t_{13}$.

A solução usando B-splines de ordem $k + m = 5$, será

$$u_\Pi(x) = \sum_{j=1}^{8} \alpha_j B_{j,5,\Pi}(x),$$

então em cada ponto de colocação

$$\sum_{j=1}^{8} \alpha_j B''_{j,5,\Pi}(x_{ij}) - \lambda^2 \sum_{j=1}^{8} \alpha_j B_{j,5,\Pi}(x_{ij}) = (\lambda^2 - 1)e^{x_{ij}}, \quad 1 \leq i \leq 2,$$

as condições de contorno

$$\sum_{j=1}^{8} \alpha_j B_{j,5,\Pi}(0) = \sum_{j=1}^{8} \alpha_j B_{j,5,\Pi}(1) = 0.$$

gerando $A\alpha = q$, de $Nk + m$ equações com $Nk + m$ variáveis, onde a matriz A é uma matriz de banda. $\triangle$

3.3.4 - Escolha da Malha

Como já foi mencionado, a eficiência do método das diferenças finitas depende da escolha da malha usada na discretização do problema. Essa escolha deve ser tal que o erro de discretização seja controlado preservando a estabilidade do método.

Existem estratégias para essa escolha, por exemplo, pode-se fixar uma malha inicial e refinar essa malha até obter uma solução com a precisão desejada, esse refinamento poderá conter pontos prefixados ou simplesmente ser a duplicação da malha corrente. Estratégias mais complexas poderão ser usadas, como a implementada nos pacotes COLSYS e COLNEW, denominada *método direto*, descrito a seguir. Dado um esquema

de discretização e uma solução inicial, determina-se uma nova malha a partir da solução corrente, e resolve-se o problema na nova malha, repetidamente, até que uma tolerância de erro seja satisfeita.

Por exemplo, considere o problema não linear (3.18b) e assuma a existência de uma solução isolada $\mathbf{y}(x)$. Tomando a solução inicial $\mathbf{y}_\Pi^o$, a malha deve ser escolhida de tal forma que N seja pequeno e o erro $e_i = \max_{x_i \le x \le x_{i+1}} ||\mathbf{y}_\Pi(x) - \mathbf{y}(x)||$ satisfaça a tolerância dada.

Para definir a nova malha Π^*, a partir de uma solução corrente $\mathbf{y}_\Pi(x)$ e uma malha corrente Π, define-se uma função $\Phi(x, \mathbf{y}(x))$, denominada função de controle, com derivadas parciais contínuas em uma vizinhança de raio ρ de $\mathbf{y}(x)$, $a < x < b$, e $\Phi(x, \mathbf{y}(x)) \ge \delta > 0$, para todo $x \in [a, b]$ e para todo δ. E diz-se que a malha Π está equidistribuida com respeito a Φ, se existe alguma constante λ tal que $\int_{x_i}^{x_{i+1}} \Phi(x, \mathbf{y}(x))\, dx = \lambda$, $1 \le i \le N$, $\lambda = \frac{\theta}{N}$ onde $\theta = \int_a^b \Phi(x, \mathbf{y}(x)) dx$.

Para escolher a função de controle, Φ, no método da colocação, usa-se o erro global (3.20b),

$$||\mathbf{y}_\Pi - \mathbf{y}|| = \max_{x_i \le x \le x_{i+1}} ||\mathbf{y}_\Pi(x) - \mathbf{y}(x)|| \le C_i h_i^{k+m} |u^{(k+m)}| + O(h^{(2k)}),$$

onde $C_i = \max_{0<\xi<1} p(\xi)(1 + O(h_i))$ para p definido em (3.20b).

Seja $|u^{(k+m)}|^{\frac{1}{k+m}}$, $x_i \le x \le x_{i+1}$ a função de controle escolhida. Neste caso, não podemos tomar u_Π como uma aproximação de u, pois $|u_\Pi^{(k+m)}|^{\frac{1}{k+m}} = 0$. Mas suponha que $v(x) \in P_{2,\Pi} \cap C[a, b]$ seja uma função linear por partes que interpola $|u_\Pi^{(k+m-1)}(x)|$ nos pontos médios dos subintervalos, $(v(x_{i+\frac{1}{2}}) = u_\Pi^{(k+m-1)}(x_{i+\frac{1}{2}}), 1 \le i \le N)$ e defina a função de controle por

$$\Phi'(x, v(x)) = |v'(x)|^{\frac{1}{k+m}}.$$

Como $p^{(k+m-1)}(\frac{1}{2}) = 0$, tem-se

$$v'_{(x_i+\frac{1}{2})} = u_\Pi^{(k+m)}(x_i + \frac{1}{2}) = u^{(k+m)}(x_i + \frac{1}{2}) + O(h_i^2), \qquad 1 \le i \le N,$$

e então $v'(x) = u^{(k+m)}(x)(1 + O(h))$ é uma aproximação da função de controle $|u^{(k+m)}|^{\frac{1}{k+m}}$ de ordem h, e ainda, $v'(x)$ é uma função constante por partes e

$$\Theta := \int_a^b |v'(x)|^{\frac{1}{k+m}}\, dx$$

é fácil de ser calculada. Considerando a integral

$$\int_{x_i}^{x_{i+1}} |v'(x)|^{\frac{1}{k+m}}\, dx \equiv \frac{\Theta}{N}, \quad 1 \le i \le N, \tag{3.21a}$$

e a função

$$t(x) := \frac{1}{\Theta} \int_a^x |v'(\xi)|^{\frac{1}{k+m}} \, d\xi \tag{3.21b}$$

pode-se determinar x_{i+1} a partir de x_i, fazendo

$$t(x_{i+1}) = \frac{1}{N}. \tag{3.21c}$$

Uma solução colocada $\mathbf{y}_\Pi(x)$, em uma malha Π^* determinada por (3.21) com $N = \Theta(\frac{\widehat{C}}{Tol})^{\frac{1}{k+m}}$, satisfaz

$$||\mathbf{y}^*_\Pi - \mathbf{y}|| \leq \widehat{C}(\frac{\Theta}{N})^{k+m}(1 + O(h)) + O(h^{2k}), \quad \text{onde} \quad \widehat{C} = \max_{0<\xi<1} ||P(\xi)||.$$

3.4 - Método dos Elementos Finitos: Método de Ritz

A idéia básica deste método consiste em encontrar um espaço conveniente de funções que se aproxime do espaço da solução exata do PVC/EDO, e escolher a solução aproximada que minimize um funcional sobre o espaço de soluções aproximadas. A formulação variacional do PVC incorpora as características do problema. As funções de aproximação são, normalmente, polinomiais por partes (Splines) definidas em alguma malha do domínio.

Considere o PVC/EDO de segunda ordem (Problema de Sturm-Louville)

$$Lu(x) = -(p(x)'u)' + q(x)u = f(x)$$

com as condições de contorno

$$u(a) = 0, \qquad \text{e} \qquad u(b) = 0$$

com $p > 0$ e $q \geq 0$, que possui uma única solução u.

O método de Ritz trabalha diretamente com a formulação variacional do problema, ou seja, a equação $Lu = f$ é relacionada com o funcional quadrático

$$I(v) = (Lv, v) - 2(f, v) \tag{3.23a}$$

onde

$$(f, v) = \int_a^b f(x)v(x) \, dx.$$

O funcional $I(v)$ deve ser minimizado, $\frac{\partial I(v)}{\partial v}|_{v=u} = 0$, e $v = u$ se e somente se $Lu = f$. Portanto, o problema de resolver $Lu = f$ é equivalente a minimizar I, ambos produzindo a solução procurada, u.

Construindo $I(v)$ para o problema de Sturm-Louville dado obtém-se:

$$I(v) = \int_a^b (-pv')' + qvdx - 2\int_a^b fvdx$$

integrando por partes e usando as condições de contorno tem-se

$$I(v) = \int_a^b [p(v')^2 + qv^2 - 2fv]\, dx$$

para qualquer $v \in C^{(1)}[a, b]$.

Para determinar a solução, o método de Ritz considera um espaço de dimensão finita que é um subespaço de $C^{(1)}[a, b]$ e procura uma função u_Π que minimize $I(v)$ e que satisfaça as condições de contorno, ou seja, a função u_Π deve ser tal que $I(u) = \min I(v)$ para todo $v \in C_0^{(1)}[a, b] = \{v \in C^{(1)}[a, b], v(a) = v(b) = 0\}$. Seja Π uma malha em [a,b] e $P^0_{k,\Pi,l}$ o espaço das funções polinomiais por partes de ordem k em $C^{(l-1)}[a, b]$ e nulos nos extremos do intervalo ($P^0_{k,\Pi,l} \subset C^{(l-1)}[a, b]$). Escolhe-se uma base $\{\Phi_j(x)\}_{j=1}^J$ para $P^0_{k,\Pi,l}$ e escreve-se $v_\Pi(x) = \sum_{j=1}^J \alpha_j \Phi_j(x)$ para qualquer $v_\Pi(x) \in P^0_{k,\Pi,l}$.

A solução $u_\Pi(x) = \sum_{j=1}^J \widehat{\alpha}_j \Phi_j(x)$ que minimiza I, ou seja, que torna $\frac{\partial I(u)}{\partial \alpha_j} = 0$, $1 \leq j \leq J$ é determinado pela resolução do sistema linear

(3.23b) $$A\widehat{\mathbf{a}} = \widehat{\mathbf{f}},$$

onde $\widehat{\mathbf{a}} = [\widehat{\alpha}_1 \ \ldots \ \widehat{\alpha}_J]^t$ e A e $\widehat{\mathbf{f}}$ são:

$$A = [a_{ij}], \quad a_{ij} = \int_a^b (p(x)\Phi_i'(x)\Phi_j'(x) + q(x)\Phi_i(x)\Phi_j(x))\, dx \qquad 1 \leq i, j \leq J \qquad \text{e}$$

$$\widehat{\mathbf{f}} = [\widehat{f}_1 \ \ldots \ \widehat{f}_J]^t, \quad \widehat{f}_i = \int_a^b f(x)\Phi_i(x)\, dx.$$

Assumindo uma suavidade apropriada nos coeficientes da EDO, Ascher et al.[1988] mostra que *para qualquer inteiro* $n \geq 1$, $l \geq n$, *a solução de Ritz* $u_\Pi(x) \in P^0_{2n,\Pi,l}$ *satisfaz*

$$|u_\Pi - u| = O(h^{2n}).$$

EXEMPLO 3.10 - Seja o PVC linear

$$\begin{cases} -u'' + u = x^2 - \frac{x}{2} - 2, & 0 < x < 0.5 \\ u(0) = u(0.5) = 0 \end{cases}$$

com solução $u(x) = x^2 - \frac{x}{2}$.

Considerando uma malha de $N = 5$ subintervalos com $h_i = 0.1$ para todo $i = 1, \ldots, 5$ e funções $\{\Phi_i\}_{i=1}^{N-1}$ (denominadas elementos lineares) definidas por

$$\Phi_i(x) = \begin{cases} \frac{x - x_{i-1}}{h}, & x_{i-1} \leq x \leq x_i, \\ \frac{x_{i+1} - x}{h}, & x_i < x \leq x_{i+1}, \\ 0 \quad \text{caso contrário}, \end{cases}$$

que formam uma base do espaço das soluções aproximadas, resultando em um sistema linear $A\hat{\mathbf{a}} = \hat{\mathbf{f}}$, onde a matriz A tem elementos

$$a_{ij} = \int_{x_{i-1}}^{x_i} (\Phi_i{}' \, \Phi_j{}' + \Phi_i \Phi_j)\, dx + \int_{x_i}^{x_{i+1}} (\Phi_i{}' \, \Phi_j{}' + \Phi_i \Phi_j)\, dx,$$

$\hat{\mathbf{f}}$ tem elementos

$$f_i = \int_{x_{i-1}}^{x_i} f\Phi_i(x)\, dx + \int_{x_i}^{x_{i+1}} f\Phi_i(x)\, dx,$$

E o sistema (3.23a) tem a forma

$$\begin{bmatrix} 20.067 & -9.983 & 0 & 0 \\ -9.98 & 20.067 & -9.98 & 0 \\ 0 & -9.983 & 20.067 & -9.983 \\ 0 & 0 & -9.983 & 20.067 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} -0.203 \\ -0.205 \\ -0.205 \\ -0.203 \end{bmatrix}. \qquad \triangle$$

4 - IMPLEMENTAÇÕES

Neste capítulo serão descritas e analisadas as implementações disponíveis dos métodos descritos no capítulo anterior. Os métodos cujas implementações não estão disponíveis foram implementados apenas para fins de comparação e estudo do comportamento dos mesmos. Serão analisados os seguintes pacotes: MUS - pacote para resolução de PVC/EDO de primeira ordem usando o método do *shooting* múltiplo com técnica de marcha e reortogonalização; COLSYS e COLNEW - pacotes usando o método da colocação para resolução de PVC/EDO de ordem superior.

4.1 - Implementações Efetuadas

Foram implementados os seguintes métodos para PVC de primeira ordem: superposição e superposição reduzida com *shooting* simples e superposição com *shooting* múltiplo nas versões padrão, padrão com compactação e usando reortogonalização nos pontos de *shooting*, para problemas lineares; *shooting* múltiplo padrão para problemas não lineares e o método dos elementos finitos para PVC de segunda ordem. Para resolver os PVIs nos métodos do valor inicial foi usado o pacote EPISODE - desenvolvido por C. D. Byrne e A. C. Hindamarsh, Argonne National Laboratory, traduzido e adaptado por M. T. Hattori, DSC - CCT - UFPB, e para resolver o sistema linear resultante da discretização do problemas usou-se as rotinas DCOMP e DSOLVE do SEDAN, uma biblioteca de rotinas numéricas para o ensino de cálculo numérico desenvolvida pelo Departamento de Sistemas e Computação.

4.1.1 - Implementação do método do *shooting* simples

A implementação deste método foi feita segundo a descrição dada no capítulo 3, §3.1.

`Algoritmo 4.1` - Superposição com *Shooting* Simples

Entrada: Um PVC/EDO linear de primeira ordem

$$\begin{cases} \mathbf{y}' = A(x)\mathbf{y} + \mathbf{q}(x), & \text{se} \quad a < x < b, \\ B_a\mathbf{y}(a) + B_b\mathbf{y}(b) = \mathbf{d}, \end{cases}$$

especificado de acordo com as exigências do pacote para resolução dos PVIs, (ver apêndice 2) e uma malha de pontos de saída ($a \leq x_1 < x_2 < \cdots < x_J \leq b$).

Saída: a solução do problema nos pontos de saída fornecidos.

1 - Resolver os PVIs.

$$\begin{cases} Y'(x) = A(x)Y(x), \\ Y(a) = I, \end{cases} \quad \text{e} \quad \begin{cases} \mathbf{v}'(x) = A(x)\mathbf{v}(x) + \mathbf{q}(x), \\ \mathbf{v}(a) = 0, \end{cases}$$

para determinar $Y(b)$ e $\mathbf{v}(b)$.

2 - Construir a matriz Q e o vetor $\widehat{\mathbf{d}}$ conforme (3.2a,b)

3 - Determinar $\mathbf{s}$, resolvendo o sistema $Q\mathbf{s} = \widehat{\mathbf{d}}$,

4 - Obter $\mathbf{y}(a) = \mathbf{s}$ por (3.1a)

5 - Integrar o PVI

$$\begin{cases} \mathbf{y}'(x) = A(x)\mathbf{y}(x) + \mathbf{q}(x), \\ \mathbf{y}(a) = \mathbf{s}, \end{cases}$$

obtendo a solução nos pontos de saída predeterminados.

No passo 1 exigiu-se que a matriz solução fundamental satisfizesse a condição $||Y(x)|| < \frac{Tol}{\epsilon_M}$, onde Tol é a tolerância dada, e no passo 3 que a matriz Q do sistema $Q\mathbf{s} = \widehat{\mathbf{d}}$ fosse bem condicionada.

O algoritmo usado na implementação do método da superposição reduzida, usado para PVC com condições de contorno separáveis é identico ao usado em superposição a menos da determinação das condições iniciais $\widehat{Y}(a)$ de (3.3b) e $\mathbf{v}(a)$ de (3.3c). Para determinar esses valores iniciais usou-se a rotina DQRDC - traduzido e adaptado por M.T.Hattori DSC-CCT-UFPB, que utiliza a transformação de Hauseholder para fazer a decomposição QR da matriz B_{a1} obtendo as matrizes H e R dadas em (3.3d,e).

4.1.2 - Implementação do método do *Shooting* Múltiplo

Em todas as versões do método do *shooting* múltiplo a malha é fixa, isto é, os pontos de *shooting* são predeterminados. Se ocorrer erro devido ao crescimento da solução em um intervalo da malha, o usuário deve fornecer uma nova malha e recomeçar a resolução do problema.

O controle de erro na resolução dos PVIs é feita pela própria rotina EPISODE de acordo com a tolerância dada. O controle no crescimento da solução fundamental Y_i é feito da mesma forma que no método do *shooting* simples em cada intervalo da malha dada.

`Algoritmo 4.2` - Método da Superposição com *Shooting* Múltiplo

Entrada: Um PVC/EDO linear de primeira ordem como no algoritmo 4.1 e uma malha Π de $N + 1$ pontos de saída (ou pontos de *shooting*).

Saída: a solução do problema nos pontos de saída fornecidos.

1 - Para $i = 1, \cdots, N$

1.1 - Resolver os PVIs.

$$\begin{cases} Y_i'(x) = A(x)Y_i(x), \\ Y_i(x_i) = F_i, \end{cases} \quad \text{e} \quad \begin{cases} \mathbf{v}_i'(x) = A(x)\mathbf{v}_i(x) + \mathbf{q}(x), \quad x_i < x < x_{i+1} \\ \mathbf{v}_i(x_i) = 0, \end{cases}$$

para determinar $Y_i(x_{i+1})$ e $\mathbf{v}_i(x_{i+1})$.

1.2 - Construir a matriz A e o vetor $\widehat{\mathbf{d}}$ de (3.5e).

2 - Determinar $\mathbf{s}$ resolvendo o sistema $A\mathbf{s} = \widehat{\mathbf{d}}$.

3 - Obter $\mathbf{y}_i(x_i)$ por (3.5h)

A implementação do método do *shooting* múltiplo padrão segue o algoritmo 4.2 com $F_i = I$ para todo $i = 1, \ldots, N$ conforme descrito no capítulo 3, §3.2.

No método de *shooting* múltiplo usando compactação, para determinar o vetor de paramentros $\mathbf{s}$, o passo 1.2 e 2 do algoritmo 4.2 são modificados para determinar as soluções fundamantais Φ_i e $\mathbf{r}_i$ usando (3.7b,c) respectivamente e armazenando essas soluções para todo i. Monta e resolve o sistema (3.7d) para determinar o vetor $\mathbf{s}_1$ e determina $\mathbf{s}_i$ por (3.7a).

O método de *shooting* múltiplo com reortogonalização difere do método padrão apenas na resolução dos PVIs homogêneos do passo 1.1 do algoritmo 4.2, onde os dados iniciais do (i+1)-ésimo intervalo da malha são determinados pela decomposição da solução fundamental do i-ésimo intervalo, isto é, efetua uma decomposição QR de $Y_i(x_{i+1})$ obtendo as matrizes F_{i+1} e Γ_i. Para iniciar o processo considerou-se a matriz $F_1 = I$.

Na implementação de problemas não linerares, as matrizes iniciais $F_i = I$, para todo i, em (3.12e) e a solução aproximada inicial é nula, ou seja, $\mathbf{s}_i = 0$, para todo i, e um valor de $\mathbf{s}$ é considerado aceitável se $||\mathbf{F}(\mathbf{s})|| < Tol$, onde Tol é a tolerância dada.

`Algoritmo 4.3` - método do *shooting* múltiplo para problemas não lineares
Entrada: O PVC de primeira ordem

$$\begin{cases} \mathbf{y}' = \mathbf{f}(x, \mathbf{y}), & \text{se} \quad a < x < b, \\ \mathbf{g}(\mathbf{y}(a), \mathbf{y}(b)) = 0, \end{cases}$$

uma malha Π de $N+1$ pontos de saída (ou pontos de *shooting*) e uma aproximação inicial para a solução.

Saída: A solução do problema nos pontos de malha fornecidos.

1 - Para $i = 1, \cdots, N$

1.1 - Resolver os PVIs

$$\begin{cases} \mathbf{y}_i'(x) = \mathbf{f}(x, \mathbf{y}_i), & x_i < x < x_{i+1} \\ \mathbf{y}_i(x_i) = \mathbf{s}_i, \end{cases}$$

para obter os vetores solução $\mathbf{y}_i(x; \mathbf{s})$.

2 - Verificar se $||\mathbf{F}(s)|| < Tol$, $\mathbf{F}(s)$ dado por (3.12c); se a condição for satisfeita, pare.

3 - Montar a matriz $F'(x) = \frac{\partial \mathbf{F}(x)}{\partial \mathbf{y}}$,

3.1 - montar as matrizes A, B_a e B_b dadas em (3.11c,d),

3.2 - resolver os PVIs (3.12e),

4 - Resolver o sistema $F'(x)\mathbf{r} = -\mathbf{F}(x)$,

5 - Determinar $\mathbf{s}^{m+1} = \mathbf{s}^m + \mathbf{r}$,

6 - Voltar para 1.

4.1.3 - Implementação do Método dos Elementos Finitos

Foi implementado o método de Ritz para PVC/EDO de segunda ordem conforme capítulo 3, §3.4, para determinar a solução, em uma malha Π previamente definida; $u_\Pi(x) = \sum_{j=1}^{J} \widehat{\alpha}_j \Phi_j(x)$ minimiza o funcional I dado em (3.23a) resultando na resolução do sistema linear

$$A\widehat{\mathbf{a}} = \widehat{\mathbf{f}},$$

onde $\widehat{\mathbf{a}} = [\widehat{\alpha}_1, \ldots, \widehat{\alpha}_J)$ e A e $\widehat{\mathbf{f}}$ são:

$$A = (a_{ij}), \quad a_{ij} = \int_a^b (p(x)\Phi_i'(x)\Phi_j'(x) + q(x)\Phi_i(x)\Phi_j(x))\, dx \qquad 1 \le i, j \le J \qquad \text{e}$$

$$\widehat{\mathbf{f}} = (\widehat{f}_1, \ldots, \widehat{f}_J), \quad \widehat{f}_i = \int_a^b f(x)\Phi_i(x)\, dx.$$

Em uma malha Π de N subintervalos definem-se as funções $\{\Phi_i\}_{i=1}^{J}$ usando elementos lineares ou usando splines cúbicos.

Para resolver as integrais foi usado a rotina GQ do SEDAN, que usa quadratura de Gauss-Legendre de ordem n, $2 \le n \le 20$.

`O método de Ritz com elementos lineares`: As funções $\{\Phi_i\}_{i=1}^{N-1}$ são definidas por

$$\Phi_i(x) = \begin{cases} \frac{x - x_{i-1}}{h}, & x_{i-1} \le x \le x_i, \\ \frac{x_{i+1} - x}{h}, & x_i < x \le x_{i+1}, \\ 0 \quad \text{caso contrário,} \end{cases}$$

formam uma base do espaço das soluções aproximadas, resultando em um sistema linear $A\hat{\mathbf{a}} = \hat{\mathbf{f}}$, onde a matriz A tem elementos

$$a_{ij} = \sum_{k=i-1}^{i} \int_{x_k}^{x_{k+1}} (p\,\Phi_j'\Phi_i' + q\,\Phi_j\Phi_i)\,dx,$$

$$\begin{aligned}
&\text{se} \quad i = 1 \quad \Rightarrow \quad i \le j \le i+1,\\
&\text{se} \quad 1 < i < N-1 \quad \Rightarrow \quad i-1 \le j \le i+1,\\
&\text{se} \quad i = N-1 \quad \Rightarrow \quad i-1 \le j \le n-1,
\end{aligned}$$

e $\hat{\mathbf{f}}$ tem elementos

$$f_i = \sum_{k=i-1}^{i} \int_{x_k}^{x_{k+1}} f\Phi_i(x)\,dx.$$

`O método de Ritz com Splines cúbicos:` As funções $\{B_{-1}, B_0, \ldots, B_N, B_{N+1}\}$ formam uma base para os splines cúbicos onde

$$B_j(x) = \begin{cases}
\frac{(x-x_{j-2})^3}{h^3}, & \text{se} \quad x_{j-2} \le x \le x_{j-1},\\
1 + \frac{3}{h}(x-x_{j-1}) + \frac{3}{h^2}(x-x_{j-1})^2 - \frac{3}{h^3}(x-x_{j-1})^3 & \text{se} \quad x_{j-1} \le x \le x_j,\\
1 + \frac{3}{h}(x_{j+1}-x) + \frac{3}{h^2}(x_{j+1}-x)^2 - \frac{3}{h^3}(x_{j+1}-x)^3 & \text{se} \quad x_j \le x \le x_{j+1},\\
\frac{(x_{j+2}-x)^3}{h^3}, & \text{se} \quad x_{j+1} \le x \le x_{j+2},\\
0 & \text{caso contrário.}
\end{cases}$$

As funções v_Π devem satisfazer as condições de contorno

$$v_\Pi(a) = v_\Pi(b) = 0.$$

Uma base $\{\Phi_i\}_{i=1}^N$ definida por

$$\begin{aligned}
\Phi_0(x) &= B_0(x) - 4B_{-1}(x),\\
\Phi_1(x) &= B_0(x) - 4B_1(x),\\
\Phi_j(x) &= B_j(x), \qquad \text{para} \quad 3 \le j \le N-2,\\
\Phi_{N-1}(x) &= B_N(x) - 4B_{N-1}(x),\\
\Phi_N(x) &= B_N(x) - 4B_{N+1}(x),
\end{aligned}$$

gera o espaço das funções v_Π, recaindo em um sistema linear $A\hat{\mathbf{a}} = \mathbf{d}$, onde a matriz A tem os elementos

$$a_{ij} = \sum_{k=a}^{b} \int_{x_k}^{x_{k+1}} (p\Phi_j'\Phi_i' + q\Phi_j\Phi_i)\,dx, \quad \text{com}$$

$a = 0$ e $b = \min\{j+1, i+1\}$ se $1 \le i < 3$ e $1 \le j \le i+3$,

$a = \max\{j-2, i-2\}$ e $b = \min\{j+1, i+1\}$ se $4 \le i \le N-3$ e $i-3 \le j \le i+3$,

$a = \max\{j-2, i-2\}$ e $b = N-1$ se $N-2 \le i < N$ e $i-3 \le j \le N$

e a $\widehat{\mathbf{f}}$ tem elementos

$$\widehat{f}_i = \sum_{k=i-1}^{i+1} \int_{x_k}^{x_{k+1}} f\Phi_i \, dx.$$

4.2 - O Pacote MUS

É uma implementação do método de *shooting* múltiplo para resolver sistemas de PVC/EDOs de primeira ordem, não rígidos, usando reortogonalização nos pontos de *shooting*, usando a técnica de marcha para determinar os pontos de *shooting* e compactação para determinar o vetor $\mathbf{s}_i$ em (3.8a) e $\mathbf{r}$ em §3.2.1. O código foi escrito por R.M.M. Mattheij, G.W.M. Staarink, Economisch Instituut, Katholieke Universiteit, Nijmegen, The Netherlands.

Será descrito como a reortogonalização é implementada, como os PVIs são resolvidos, como os pontos de saída (pontos de *shooting*) são escolhidos e como é feita a escolha da matriz ortogonal F_1 dada no capítulo 3, §3.1.2.

4.2.1 - Reortogonalização

Para formar um algoritmo estável é necessário controlar o crescimento das soluções fundamentais, em cada intervalo da malha Π, e quando alguma tolerância for excedida um novo ponto x_{i+1} da malha será considerado e as colunas de $Y_i(x_{i+1})$ serão reortogonalizadas. Ao fazer a reortogonalização, haverá um desacoplamanto das soluções crescentes e decrescentes, permitindo com isso, construir um algoritmo estável. Dada uma matriz F_1 e o primeiro subintervalo de uma malha de pontos de saída $[x_1, x_2]$, determina-se a solução fundamental $Y_1(x_2)$ satisfazendo $Y_1(x_1) = F_1$ conforme mostrado no capítulo anterior §3.1.2. Decompõe-se $Y_1(x_2) = F_2\widetilde{\Gamma}_1$, onde F_2 é ortogonal e $\widetilde{\Gamma}_1$ é triangular superior, tomando agora F_2 como a matriz inicial no próximo intervalo da malha. De uma forma geral em um subintervalo $[x_i, x_{i+1}]$ determina-se $Y_i(x_{i+1})$ satisfazendo $Y_i(x_i) = F_i$ e decompõe-se a matriz $Y_i(x_{i+1})$ em um produto de matrizes, uma ortogonal e outra triangular superior; a matriz ortogonal F_{i+1} é usada como valor inicial no próximo subintervalo e a matriz triangular superior $\widetilde{\Gamma}_i$ é usada para selecionar o próximo ponto de *shooting*.

4.2.2 - Resolução dos PVIs

A resolução dos PVIs é a parte básica deste método. Em MUS é usado o método de Runge-Kutta-Fehlberg de quarta e quinta ordens (ver apêndice 1). Primeiramante é resolvido o PVI não homogêneo, ou seja, determina-se uma solução particular $\mathbf{v}_i$, em um intervalo $[x_i, x_{i+1}]$ da malha de pontos de saída. Na resolução desse primeiro PVI são determinados os passos e fixados para a resolução dos PVIs homogêneos, usando o método de quarta ordem de Runge-Kutta, no último ponto dessa malha a matriz solução fundamantal é decomposta em uma matriz ortogonal e outra triangular superior.

4.2.3 - Escolha dos pontos de *shooting*

Para determinar os pontos de *shooting* durante o processamento, foi usada a estratégia de controlar o crescimento da solução fundamental. A determinação desses pontos é feita como segue.

Seja x_{i_j} o ponto inicial de um subintervalo da malha de pontos de saída, ou seja, um ponto de *shooting*. Define-se

$$W_1 = \Gamma_{i_j} \quad \text{e} \quad \mathbf{v}_1 = \widehat{\mathbf{d}}_j$$

de (3.8a). Se $||W_1|| \geq M$, onde $M \leq \frac{Tol}{\epsilon_M}$ é o limite permitido, então x_{i_j+1} é o próximo ponto de *shooting*. Se $||W_1|| < M$, então para $s = 1, 2, \ldots$ calcula-se

$$W_{s+1} = \Gamma_{i_j+s} W_s \quad \text{e} \quad \mathbf{v}_{s+1} = \Gamma_{i_j+s}\mathbf{v}_s + \widehat{\mathbf{d}}_{i_j+s},$$

até que $||W_{s+1}|| \geq M$, e o intervalo considerado é $[x_{i_j}, x_{i_j+s+1}]$. Define-se

$$V_j = W_{s+1} \qquad \text{e} \qquad \widetilde{\mathbf{v}}_j = \mathbf{v}_{s+1}$$

e a recorrência para a sequência s_{i_j}, será

$$s_{i_j+1} = V_j s_{i_j} + \widetilde{\mathbf{v}}_j$$

4.2.4 - Determinação da matriz F_1

O principal problema neste método é como encontrar a matriz F_1 adequada. Esta matriz inicial deve ser tal que de alguma forma estabeleça uma ordem nas soluções da matriz fundamental, ou seja, que $\widetilde{\Gamma}_i$ tenha uma ordem nos blocos e que D_i definida em (3.8b) reflita o crescimento das soluções, isto deve ser medido por inspeção dos elementos da diagonal de $\widetilde{\Gamma}_i$. Para se conseguir esse objetivo, assumindo que exista uma dicotomia na solução fundamantal, é suficiente fazer com que a matriz $\widetilde{\Gamma}_1$ tenha diagonal ordenada de forma decrescente, isto é, os elementos da diagonal devem aparecer na ordem decrescente $(a_{ii} \geq a_{jj},\ i < j)$.

No pacote MUS, o procedimento é iniciado em $x = x_1$ com $F_1 = I$ e, em $x = x_2$, verificando se a diagonal de Γ_1 está ordenada, se isso não acontecer, as colunas de $\widetilde{\Gamma}_1$ deverão ser reordenadas de acordo com o valor dos elementos de sua diagonal, usando uma matriz de permutação $P_1^1(2)$ (o dígito 2 no parenteses significa que a verificação está sendo feita em $x = x_2$ e o digito 1 do índice inferior que é a primeira permutação feita). A matriz permutada é novamante decomposta em uma matriz ortogonal e uma triangular superior

$$\widetilde{\Gamma}_1(1)P_1^1(2) = P_2^1(2)\widetilde{\Gamma}_1(2).$$

Se os elementos da diagonal de $\widetilde{\Gamma}_1(2)$ não estiverem ordenados, repete-se o processo. De acordo com Mattheeij e Staarink [1984a], o processo dará resultado após um número finito de passos. Sejam $j = 1, \ldots, p$ os passos dados, onde

$$\widetilde{\Gamma}_1^{j-1}(1)P_1^j(2) = P_2^j(2)\widetilde{\Gamma}_1^j(2).$$

Então

$$\widetilde{\Gamma}_1(1)P_1^1(2)P_1^2(2)\cdots P_1^p(2) = P_2^p(2)\cdots P_2^1(2)\widetilde{\Gamma}_1^p(2).$$

Fazendo

$$S_1(2) = P_1^1(2)P_1^2(2)\cdots P_1^p(2), \quad S_2(2) = P_2^p(2)\cdots P_2^1(2) \quad \text{e} \quad \widetilde{\Gamma}_1(2) = \widetilde{\Gamma}_1^p(2),$$

tem-se

$$\widetilde{\Gamma}_1(1)S_1(2) = S_2(2)\widetilde{\Gamma}_1(2).$$

Se a sequência original F_i e $\widetilde{\Gamma}_i$ for representada por $F_i(1)$ e $\widetilde{\Gamma}_i(1)$, então, usando $F_1(2) = S_1(2)$ como matriz inicial, as sequências induzidas serão $F_i(2)$ e $\widetilde{\Gamma}_i(2)$.

Como se está construindo os intervalos de *shooting* deve-se verificar se a matriz $W_2(2) = \widetilde{\Gamma}_2(2)W_1(2)$ possui diagonal ordenada de forma decrescente. Se isso não ocorrer, permutam-se as colunas de W_2 e faz-se uma decomposição QR como descrito anteriormente, ou seja,

$$W_2(2)S_1(3) = S_3(3)W_2(3)$$

onde $S_3(3)$ é ortogonal e $W_2(3)$ é triangular superior, e considera-se a matriz inicial $F_1^0 = S_1(2)S_1(3)$ para obter a sequência $F_i(3)$ e $\Gamma_i(3)$. Essa adaptação é feita até o ponto final do primeiro subintervalo determinado. Nos pontos seguintes a ordenação é assegurada, se o espaço de solução apresentar uma dicotomia, caso contrário, a ordem encontrada inicialmente não será globalmente aceita. Portanto, as diagonais das matrizes triangualres superior devem ser verificadas, para assegurar a ordenação. Quando não for encontrada a ordenação o processo recomeça. Após ter encontrado uma ordenação satisfatória, o valor de k, dimensão do subspaço de soluções crescentes, é determinado por inspeção.

4.2.5 - Controle do erro

A ordem de crescimento do erro de arredondamanto em (3.8c,d) é determinado por

$$\max_{p,q} ||\Omega_{p,q}||, \quad \max_{j,i} || \prod_{l=j}^{i} E_l ||, \quad \max_{i,j} ||(\prod_{l+j}^{i} D_l)^{-1}||$$

onde $\Omega_{p,q} = \sum_{l=p}^{q-1} \{(\prod_{l+j}^{i} D_l)^{-1} C_l (\prod_{l=j}^{i} E_l)\}$ e para verificar essas normas calcula-se

$$\rho = \max_{i,j} || \prod_{l=j}^{i} \mathrm{diag}(E_l)|| \cdot \max_{i,j} || \prod_{l+j}^{i} \mathrm{diag}(D_l)^{-1})||,$$

e ρ representa o máximo dos elementos da diagonal das matrizes $(D_l)^{-1}$ e E_l e é uma boa estimativa do crescimanto de E e D. Se ρ não for muito grande pode-se esperar que $|| \prod_{l=j}^{i} E_l||, ||(\prod_{l+j}^{i} D_l)^{-1}||$ sejam de ordem 1. A estimativa ρ é usada da seguinte forma: suponha que o usuário queira uma tolerância Tol; se o código detectar $\rho\varepsilon_M > Tol$ indica um acúmulo de erro de arredondamanto. Portanto ρ é um bom estimador para a ampliação global do erro conforme Mattheeij e Staarink [1984a].

4.2.6 - Solução de problemas não lineares

Na solução de problemas não lineares é usado o método de Newton descrito no capítulo 3, §3.2.1 e §3.2.4. Para montar a matriz $F'(\mathbf{s})$ é usado reortogonalização e desacoplamanto como no caso linear e o vetor $\mathbf{r}$ é determinado usando compactação. O processo é iniciado com tolerâncias maiores e executa um refinamento, na tolerância, quando ocorre uma convergência na malha corrente.

4.3 - Os Pacotes COLSYS e COLNEW

Nesta seção serão analisadas as implementações do método de colocação com pontos gaussianos, definidos em §3.3, para problemas de valor de contorno para sistemas de EDO de ordem mista, usando B-splines e bases monomiais, não requerendo a conversão do problema em um sistema de primeira ordem. Os pacotes são denominados COLSYS e COLNEW respectivamente.

O código COLSYS foi escrito por U. Ascher da Universidade de British Columbia, Canada, J. Christiansen e D. Russel da Universidade Simon Fraser, British Columbia, Canadá. O COLNEW é uma modificação de COLSYS feita por U. Ascher e G. Bader da Universidade de Heidelberg, Heidelberg, Alemanha.

4.3.1 - Definição do Problema

Seja o sistema de ordem mista de equações não lineares de ordem

$$1 \leq m_1 \leq m_2 \leq \cdots \leq m_d \leq 4, \qquad 0 \leq d \leq 20, \qquad m^* = \sum_{n=1}^{d} m_n \leq 40,$$

(4.1a)

$$u_n^{(m_n)}(x) = F_n(x, \mathbf{y}(\mathbf{u})), \qquad a < x < b, \qquad n = 1, \ldots, d,$$

onde

$$\mathbf{u}(x) = [u_1(x) \ \ldots \ u_d(x)]^t \quad \text{é uma solução isolada,}$$
$$\mathbf{y}(\mathbf{u}) = [u_1 \ u_1' \ \ldots \ u_1^{(m_1-1)} \ u_2 \ldots \ u_d^{m_d-1)}]^t.$$

O sistema (4.1a) contém m^* condições de contorno separadas, possivelmente não lineares

(4.1b)
$$\mathbf{g}_j(\xi_j; \mathbf{y}(\mathbf{u})) = 0, \qquad j = 1, \ldots, m^*,$$

onde $a \leq \xi_1 < \xi_2 < \cdots < \xi_{m^*} \leq b$.

Se Π for uma malha de $[a, b]$ definida em (3.4), $h_i = x_{i+1} - x_i, \quad i = 1, \ldots, N$ e $h = \max_i h_i$. Uma solução aproximada por colocação é um vetor $\mathbf{v} = [v_1 \ \ldots \ v_d]^t$, tal que $v_n \in P_{k+m_n,\Pi} \cap C^{(m_n-1)}[a, b]$, $n = 1, \ldots, d$ com $k \geq m_d$ o número de pontos de colocação por subintervalos $[x_i, x_{i+1}]$ de Π. Se $\{\rho_j\}_{i=1}^{k}$ forem os pontos de Gauss em $[-1, 1]$, então os pontos de colocação são definidos por

$$x_{ij} = \frac{x_i + x_{i+1}}{2} + \frac{1}{2} h_i \rho_j = x_{i+\frac{1}{2}} + \frac{1}{2} h_i \rho_j, \qquad i = 1, \ldots, N, \quad j = 1, \ldots, k.$$

A solução colocada $\mathbf{v}$ será determinada de tal forma que

$$v_n^{m_n}(x_{ij}) = F_n(x_{ij}, \mathbf{y}(\mathbf{v})), \qquad j = 1, \ldots, k, \quad i = 1, \ldots, N, \quad n = 1, \ldots, d$$

e satisfaça as condições de contorno.

Para a análise, os seguintes aspectos serão considerados:

- Estimativas de erro;
- Seleção da malha;
- Avaliação das funções bases;
- Solução de sistemas lineares;

4.3.1 - Definição do Problema

Seja o sistema de ordem mista de equações não lineares de ordem

$$1 \leq m_1 \leq m_2 \leq \cdots \leq m_d \leq 4, \qquad 0 \leq d \leq 20, \qquad m^* = \sum_{n=1}^{d} m_n \leq 40,$$

(4.1a)

$$u_n^{(m_n)}(x) = F_n(x, \mathbf{y}(\mathbf{u})), \qquad a < x < b, \qquad n = 1, \ldots, d,$$

onde

$$\mathbf{u}(x) = [u_1(x) \ \ldots \ u_d(x)]^t \quad \text{é uma solução isolada,}$$
$$\mathbf{y}(\mathbf{u}) = [u_1 \ u_1' \ \ldots \ u_1^{(m_1-1)} \ u_2 \ldots \ u_d^{m_d-1)}]^t.$$

O sistema (4.1a) contém m^* condições de contorno separadas, possivelmente não lineares

(4.1b)

$$\mathbf{g}_j(\xi_j; \mathbf{y}(\mathbf{u})) = 0, \qquad j = 1, \ldots, m^*,$$

onde $a \leq \xi_1 < \xi_2 < \cdots < \xi_{m^*} \leq b$.

Se Π for uma malha de $[a, b]$ definida em (3.4), $h_i = x_{i+1} - x_i$, $i = 1, \ldots, N$ e $h = \max_i h_i$. Uma solução aproximada por colocação é um vetor $\mathbf{v} = [v_1 \ \ldots \ v_d]^t$, tal que $v_n \in P_{k+m_n,\Pi} \cap C^{(m_n-1)}[a, b]$, $n = 1, \ldots, d$ com $k \geq m_d$ o número de pontos de colocação por subintervalos $[x_i, x_{i+1}]$ de Π. Se $\{\rho_j\}_{i=1}^{k}$ forem os pontos de Gauss em $[-1, 1]$, então os pontos de colocação são definidos por

$$x_{ij} = \frac{x_i + x_{i+1}}{2} + \frac{1}{2} h_i \rho_j = x_{i+\frac{1}{2}} + \frac{1}{2} h_i \rho_j, \qquad i = 1, \ldots, N, \quad j = 1, \ldots, k.$$

A solução colocada $\mathbf{v}$ será determinada de tal forma que

$$v_n^{m_n}(x_{ij}) = F_n(x_{ij}, \mathbf{y}(\mathbf{v})), \qquad j = 1, \ldots, k, \quad i = 1, \ldots, N, \quad n = 1, \ldots, d$$

e satisfaça as condições de contorno.

Para a análise, os seguintes aspectos serão considerados:

- Estimativas de erro;
- Seleção da malha;
- Avaliação das funções bases;
- Solução de sistemas lineares;

- Solução de problemas não lineares.

O pacote COLNEW foi obtido pela substituição das funções B-splines, usadas em COLSYS, pelas bases monomiais de Runge-Kutta. As mudanças foram feitas de modo a alterar o mínimo possível os outros aspectos do pacote. A estimativa de erro e a seleção da malha são iguais em ambos os pacotes, a resolução de equações lineares e não lineares sofreram algumas alterações que serão comentadas no decorrer desta seção.

4.3.2 - Estimativa do Erro

Quando o número de pontos de colocação, k, por subintervalo for muito grande, isto é, $k > m_d$, pelo resultado de convergência (4.2a), o erro local conforme Childs et al.[1979] e Ascher et al. [1979] para $x \in [x_i, x_{i+1}]$ é dado por

(4.3a)
$$e_n^{(l)}(x) = u_n^{(l)}(x) - v_n^{(l)}(x) =$$
$$= \frac{u_n^{(k+m_n)}(x_i)}{2^{k+m_n-l}} P_n^{(l)}\left(\frac{2}{h_i}(x - x_{i+\frac{1}{2}})h_i^{k+m_n-l} + O(h^{k+m_n+1-l}),\right.$$
$$l = 0, \ldots, m_n, \quad n = 1, \ldots, d,$$

onde $P_n(\xi) = \dfrac{d^{k-m_n}}{d\xi^{k-m_n}} p(\xi)$ para $p(\xi) = \dfrac{(\xi^2-1)^k}{2k!}, \qquad \xi \in (-1,1).$

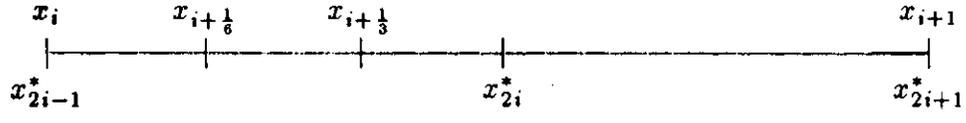
Para obter uma estimativa do erro, e também para a seleção da malha, assume-se que o termo local em (4.3a) é o termo dominante. Isto somente pode ser garantido quando a malha for *quasiuniforme*, isto é, $\frac{h}{\min_{1\le i\le N} h_i}$ for limitado, e h pequeno.

Na prática, não é possível obter a estimativa (4.3a) porque depende explicitamente do raio de convergência superior, $O(h^{k+m_n-l})$, que não é possível ser encontrado. A aproximação de $u_n^{(k+m_n)}(x_i)$ pode ser imprecisa, como também o termo de ordem global negligenciado nem sempre será desprezível.

O cálculo efetivo do erro é feito usando extrapolação, isto é, são calculadas duas soluções aproximadas $v_n(\cdot)$ e $v_n^*(\cdot)$ em malhas diferentes, $\{x_i\}_{i=1}^{N+1}$ e $\{x_i^*\}_{i=1}^{2N+1}$, respectivamente, onde $x_{2i-1}^* = x_i$ e $x_{2i}^* = x_{i+\frac{1}{2}} = (\frac{x_i+x_{i+1}}{2})$ e $e_n^* = u_n(x) - v_n^*(x)$ para $x \in [x_i, x_{i+1}]$ é o erro máximo que deverá ser estimado.

A expressão (4.3a) é usada como segue.

Consideram-se os pontos $x_{2i-\frac{2}{3}}^* = x_{i+\frac{1}{6}}$ e $x_{2i-\frac{1}{3}}^* = x_{i+\frac{1}{3}}$.



Seja

$$\Delta_1 = |v_n(x_{i+\frac{1}{6}}) - v_n^*(x_{i+\frac{1}{6}})| =$$
$$= |e_n(x_{i+\frac{1}{6}}) - e_n^*(x_{i+\frac{1}{6}})| =$$
$$= \frac{|u_n^{(k+m_n)}(x_i)|}{2^{k+m_n}} |P_n(-\frac{2}{3}) - P_n(-\frac{1}{3})| (\frac{h_i}{2})^{k+m_n} + O(h^{k+m_n+1}),$$

e analogamente, tem-se

$$\Delta_2 = |v_n(x_{i+\frac{1}{3}}) - v_n^*(x_{i+\frac{1}{3}})| =$$
$$= \frac{|u_n^{(k+m_n)}(x_i)|}{2^{k+m_n}} |P_n(-\frac{1}{3}) - P_n(\frac{1}{3})| (\frac{h_i}{2})^{k+m_n} + O(h^{k+m_n+1}),$$

para P_n definido em (4.3b). Então

$$\max_{x\in[x_{2i-1}^*, x_{2i}^*]} |e_n^*(x)| = \frac{|P_n|(\Delta_1 + \Delta_2)}{|2^{k+m_n} P_n(-\frac{2}{3}) - P_n(-\frac{1}{3})| + |2^{k+m_n} P_n(-\frac{1}{3}) - P_n(\frac{1}{3})|}$$
$$+ O(h^{k+m_n+1}).$$

Generalizando a expressão acima, para estimar o erro de todos os componentes de $\mathbf{y}(\mathbf{v})$, os pesos que multiplicam $(\Delta_1 + \Delta_2)$ são dados por

$$w_{k,v} = \frac{|P^{(v)}|}{|2^{2k-v} P^{(v)}(-\frac{2}{3}) - P^{(v)}(-\frac{1}{3})| + |2^{2k-v} P^{(v)}(-\frac{1}{3}) - P^{(v)}(\frac{1}{3})|},$$
$$v = 0, \ldots, k-1, \qquad \text{e} \qquad k > m_d.$$

Esses valores de w são previamennte calculados e armazenados no programa como constantes. A estimativa do erro na malha Π^* é dada por

$$\max |e_n^{*(l)}(x)| = w_{k,k-m_n+l}(\Delta_1 + \Delta_2) + O(h^{k+m_n+1}), \qquad l = 0, \ldots, m_n - 1,$$

onde Δ_1 e Δ_2 são tomados para $v_n^{(l)}$, $\quad n = 1, \ldots, d$.

4.3.3 - Seleção da malha

O problema de selecionar uma malha $\{x_i^*\}_{i=1}^{n^*+1}$ que equidistribua o termo local na expressão do erro (4.3a) é resolvido usando uma solução $\mathbf{v}$, previamente calculada na malha corrente $\{x_i\}_{i=1}^{N+1}$.

Dado um conjunto de tolerâncias Tol_j, $\quad j = 1, \ldots, Ntol$, com um conjunto de índices $Ltol_j$, o código tenta satisfazer

(4.4a) $$|y_l(\mathbf{u}) - y_l(\mathbf{v})| \leq Tol_j(1 + |y_l(\mathbf{v})|), \quad l = Ltol_j \quad \text{e} \quad j = 1, \ldots, Ntol,$$

com o menor N possível (N é o número de pontos da malha). Este é o objetivo do algoritmo de seleção da malha, ou seja, deseja-se encontrar uma malha $\{x_i^*\}_{i=1}^{N^*+1}$ de forma que, quando duplicada, produza uma solução que satisfaça (4.4a) com o menor N^*.

Por (4.3a), desprezando o termo global, tem-se

$$\max_{x\in[x_i,x_{i+1}]} |e_n^{(l)}(x)| = C_{k,k-m_n+l}|u_n^{(k+m_n)}(x_i)|h_i^{k+m_n-l}, \quad l = 0,\ldots,m_n-1, \tag{4.4b}$$

onde $C_{k,v} = \frac{|P^{(v)}(k,\cdot)|}{2^{2k-v}}$, são constantes, previamente calculadas e armazenadas.

Para cada j, $1 \leq j \leq Ntol$, sejam $l = Ltol_j, n = Jtol_j$ determina-se $\text{WEIGHT}_j = \frac{C_{k,k-m_n+l}}{Tol_j(1+v^{(l)}(x))}$ e $\text{ROOT}_j = \frac{1}{k+m_n-l}$.

Por (4.4a,b), a meta é encontrar a malha $\{x_i^*\}_{i+1}^{N^*+1}$ para a qual

$$\max_{1\leq j\leq Ntol} \text{WEIGHT}_j|u_n^{(k+m_n)}(x_i^*)|h_i^{\frac{1}{ROOT_j}} \leq 1 \tag{4.4c}$$

para o menor N^* possível.

Como $u_n^{(k+m_n)}(x_i^*)$ é desconhecido, de acordo com a expressão (4.4c), não é possível determinar uma malha. Além disso, em COLSYS a malha final é uma duplicação da malha anterior, logo que uma estimativa do erro é feita.

Sejam

$$S_j(x) = \text{WEIGHT}_j|u_n^{(k+m_n)}(x)| \qquad \text{e}$$
$$S(x) = \max_{1\leq j\leq Ntol} S_j^{ROOT_j}(x)$$

(a função S é usada como a função de controle). Então (4.4c) é equivalente a

$$S(x_i^*)h_i^* \leq 1. \tag{4.4d}$$

A solução colocada que satisfizer (4.4d) deverá satisfazer (4.4a).

Exigindo que a condição

$$\int_{x_i^*}^{x_{i+1}^*} S(x)dx = 1 \tag{4.4e}$$

seja satisfeita, em vez de (4.4c,d), a solução colocada ainda satisfaz (4.4a) conforme Childs et al.[1979]. Mas para verificar (4.4e) ainda se faz necessário conhecer $u_n^{(k+m_n)}(x)$, $n = 1,\ldots,d$.

Uma aproximação para as derivadas de ordem superior, conforme Ascher et al.[1979], pode ser construida da seguinte forma: Dada uma malha $\{x_i\}_{i=1}^{N+1}$ e uma solução colocada v, o polinômio na expressão do erro (4.3a) para a $(k+m_n-1)$-ésima derivada do n-ésimo componente será

$$\frac{1}{2k!}\frac{d^{2k-1}}{d\xi^{2k-1}}(\xi^2-1)^k = \xi$$

e, por (4.3a), tem-se que

$$e^{(k+m_n-1)}(x_{i+\frac{1}{2}}) = O(h^2).$$

Definindo, conforme capítulo 3, §3.3.4

$$\hat{u}_n(x_{i+1}) := \frac{2|v_n^{(k+m_n-1)}(x_{i+1}) - v_n^{(k+m_n-1)}(x_i)|}{x_{i+2}-x_i} =$$

$$= |u^{(k+m_n)}(x_{i+1})| + O(h) = |u^{(k+m_n)}(x)| + O(h),$$

para $x \in [x_i, x_{i+1}]$, $i = 1, \ldots, N-1$, e em todo o intervalo $[a,b]$

$$\hat{u}_n(x) = \begin{cases} \hat{u}_n(x_i) & \text{se} \quad x \in [x_i, x_{i+1}], \quad i = 2, \ldots, N \\ \hat{u}_n(x_2) & \text{se} \quad x \in [x_1, x_2]. \end{cases}$$

Portanto, $|u_n^{(k+m_n)}(x)| = |\hat{u}_n(x)| + O(h)$ e a função de controle será

$$\hat{s}(x) = \max_{1 \le j \le Ntol} (\text{WEIGHT}_j \hat{u}_n(x))^{ROOT_j}, \quad n = Jtol_j,$$

que é uma função constante por partes e possível de ser calculada. $S^*(x_i)h_i^* < 1$ é satisfeito para $\{x_i^*\}_{i=1}^{n^*+1}$ se $\int_{x_i^*}^{x_{i+1}^*} \hat{s}(x)dx = 1, \qquad i = 1, \ldots, N^*$.

Na prática, a ultima expresão conduz a valores de N^* grandes, quando comparados com N, significando que N^* deve ser determinado nos dados iniciais. Também se faz necessária uma estimativa do erro para verificar quando a tolerância deve ser satisfeita. Uma modificação é feita no último critério, e uma nova malha é definida (para algum N^*) de acordo com

$$\int_{x_i^*}^{x_{i+1}^*} \hat{s}(x)dx = \frac{1}{N^*}\sum_{j=1}^{N} \hat{s}(x_j)h_j = \gamma, \qquad i = 1, \ldots, N^*. \tag{4.4f}$$

Duas questões ainda necessitam ser analisadas: quando redistribuir os pontos da malha, em vez de duplicar a malha corrente? E como escolher N^*?

Quando uma solução aproximada na malha corrente $\{x_i\}_{i=1}^{N+1}$ for obtida, os diagnósticos

$$r_1 = \max_i \hat{s}(x_i)h_i, \qquad r_2 = \sum_{i=1}^{N} \hat{s}(x_i)h_i \qquad \text{e} \qquad r_3 = \frac{r_2}{N}$$

são comparados. A razão $\frac{r_1}{r_3}$ dá uma idéia da melhora que poderá ser obtida pela redistribuição. Uma redistribuição somente será feita quando $r_1 \geq 2r_3$. Na redistribuição o valor de N^* é dado por

$$N^* = \min\{\frac{1}{2}\overline{N}, N, \frac{1}{2}\max[N, r_2]\}$$

onde $\overline{N}$ é o número máximo de subintervalos especificado para o armazenamento, permitindo, com isso, uma última duplicação da malha para a obtenção de uma estimativa de erro. A definição de $\overline{N}$ dá uma restrição quanto ao número de vezes que uma malha poderá ser redistribuída antes de ser duplicada.

`Algoritmo` - Seleção da Malha e Estimativa do Erro.

1. Dada uma malha corrente $\{x_i\}_{i=1}^{N+1}$, calcule a solução v.
2. Se uma iteração não linear para v não converge, duplique a malha corrente. Se a nova malha for maior que $\overline{N}$, pare.
   Caso contrário, a malha refinada passa a ser a malha corrente e volta ao passo 1.
3. Se a malha corrente foi obtida pela duplicação de uma malha anterior, e ocorre convergência em ambas, então calcule uma estimativa do erro usando $e = w(\Delta_1 + \Delta_2)$ e verifique se

$$|y_l(\mathbf{u}) - y_l(\mathbf{v})| \leq TOL_j(1 + |y_l(\mathbf{v})|), \quad l = Ltol_j \quad \text{e} \quad j = 1, \ldots, Ntol,$$

   é satisfeito; caso afirmativo, pare.
4. Calcule $r_1, \quad r_2, \quad r_3$.
5. Se $r_1 < 2r_3$ então duplique a malha corrente; a malha refinada passa a ser a malha corrente e volta ao passo 1.
6. Defina N^*, determine $\{x_i^*\}_{i=1}^{N^*+1}$ por (4.4f), onde x_i^* passa a ser a malha corrente e volta ao passo 1.

4.3.4 - Avaliação das Funções bases

O pacote COLSYS foi implementado usando B-splines. As conveniências do uso de B-Splines são: (a) de evitar muitos cálculos repetitivos na resolução de sistemas de equações diferenciais ordinárias; (b) a continuidade dos polinômios por partes nos pontos da malha; (c) existem exatamente k polinômios em cada ponto interior da malha; (d) em muitas

avaliações dos B-Splines os pontos possuem a mesma localização em cada subintervalo da malha.

Considerando, agora, a avaliação dos B-splines e a solução colocada, sabe-se, pelo capítulo 3, §3.3.3, que

$$v_n(x) \in P_{k+m_n,\Pi} \cap C^{(m_n-1)}[a,b], \qquad (1 \leq n \leq d)$$

para uma dada malha $\Pi : a \leq x_1 < x_2 < \cdots < x_{N+1} \leq b$. Se $B_{j,k}$ é o j-ésimo B-spline de ordem k, então

$$v_n(x) = \sum_{j=k-m_n+2}^{N_k} \alpha_{j,n} B_{j,k+m_n}(x).$$

A avaliação dos B-splines e de suas derivadas é feita usando algoritmo de de Boor modificado para usar nesta aplicação particular conforme Ascher et al.[1979]). Os B-splines são usados para, além de avaliar a solução $\mathbf{v}(x)$ e suas derivadas, construir as equações de colocação, resultando em um sistema linear (ou linearizado, no caso de problemas não lineares) onde as variáveis são os coeficientes de B-splines.

A representação da base monomial de Runge-Kutta no pacote COLNEW, é uma adaptação da base monomial, vista no capítulo 3, §3.3.3, para sistemas de equações diferenciais de ordem mista.

Para obter a solução colocada $\mathbf{u}_\Pi(x_i)$ para cada i, $1 \leq i \leq N$ as variáveis $\mathbf{z}_i$ são eliminadas usando (3.19b,c) obtendo $\mathbf{y}_{i+1} = \Gamma_i \mathbf{y}_i + \mathbf{r}i$ dado em (3.19d), onde

$$\mathbf{y}_i = [u_{1\Pi}(x_i)\ u'_{1\Pi}(x_i)\ \ldots\ u_{d\Pi}(x_i)\ \ldots\ u^{m_d-1}_{d\Pi}(x_i)]^t$$
$$\mathbf{z}_i = [u^{(m_n)}_{1\Pi}(x_{i1})\ u^{(m_n)}_{1\Pi}(x_{i2})\ \ldots\ u^{(m_n)}_{d\Pi}(x_{i1})\ \ldots\ u^{(m_n)}_{d\Pi}(x_{ik})]^t.$$

A equação obtida para $\mathbf{y}_i$, $1 \leq i \leq N+1$ é juntada com as obtidas pelas condições de contorno para formar um sistema esparso de ordem $(N+1)m^*$. A matriz resultante é uma matriz bloco diagonal com o i-ésimo bloco tendo $m^* + l_i$ linhas (l_i é o número de condições de contorno envolvidas) e $2m^*$ colunas.

4.3.4 - Solução de Sistemas Lineares

Considera-se, aqui, o método para resolução do conjunto de equações algébricas lineares resultante da aplicação do método de colocação em (4.1a,b) usando bases B-splines,

$$\begin{cases} u_n^{(m_n)}(x) = F_n(x; \mathbf{y}(\mathbf{u})), & a < x < b \\ \mathbf{g}_j(\xi; \mathbf{y}(\mathbf{u})) = 0 \end{cases}$$

rápido nunca será usado e para cada malha o valor inicial de λ será $\lambda_0 = \lambda_{min}$. Assim um eficiente processo é obtido conforme Childs et al.[1979].

A resolução de problemas não lineares em COLNEW também sofreu modificações com a troca das funções bases. Essa troca ocorreu no controle das iterações, que neste caso, usa as variáveis $\{\mathbf{y}_i\}$ e $\{\mathbf{z}_i\}$, definidos em §4.3.4.

5 - EXEMPLOS E COMENTÁRIOS FINAIS

5.1 - Exemplos e Resultados Numéricos

Os exemplos desta seção foram processados em precisão dupla ($\approx$ 16 dígitos decimais) no IBM/4381 da UFPB - CAMPUS II.

Os resultados obtidos são apresentados em tabelas usando as seguintes notações para identificar o método implementado:

SSHTS - método da superposição com *shooting* simples;

SSHTSR - método da superposição reduzida com *shooting* simples;

MSHTP - método da superposição reduzida com *shooting* múltiplo padrão;

MSHTPC - método da superposição com *shooting* múltiplo com compactação;

MSHTP - método da superposição com *shooting* múltiplo padrão com reortogonalização;

MSHTNL - método do *shooting* múltiplo padrão para problemas não linerares;

ELFEL - método dos elementos finitos usando elementos lineares;

ELFSC - método dos elementos finitos usando splines cúbicos;

e as implementações profissionais:

MUS - método da superposição com *shooting* múltiplo com reortogonalização e técnica de marcha,

COLNEW - método da colocação para sistemas de PVC/EDO de ordem mista.

A cópia disponível do pacote COLSYS, que é uma implementação do método da colocação para sistemas de PVC/EDO de ordem mista, usando B-splines, publicada por Ascher et al. [1981], não funcionou para problemas não lineares. Na tentativa de resolver essa dificuldade escreveu-se para o autor, que não esclareceu as dificuldades encontradas. Em vez disso, recomendou a versão que usa bases monomiais, ou seja, o pacote COLNEW. Quanto ao caso linear, não apresentou nenhuma dificuldade, obtendo resultados idênticos aos obtidos por COLNEW.

Nos problemas 1 e 2, apresentados a seguir, foi usado um parâmetro $\lambda > 0$, para analisar o comportamanto dos métodos com o crescimento deste parâmetro.

Em ambos os problemas, quando λ cresce, a norma da matriz solução fundamental, $||Y(x)||$ também cresce, causando instabilidade nos métodos de valor inicial, indicando com isso que a matriz solução fundamantal apresenta um subespaço de solução com crescimento mais rápido que o outro e, neste caso, somente o modo de crescimento dominante é capturado pelo método, tornando o PVI associado mal condicionado.

Problema 1. Considere o Exemplo 3.1 com $b = 1$ e $\lambda = 1, 10, 20, 50$.

$$\begin{cases} u'' - \lambda^2 u = (1 - \lambda^2)e^x, & 0 < x < 1, \\ u(0) = 1, \\ u(1) = e. \end{cases}$$

O problema é bem condicionado, para todos os valores de λ. A sua solução analítica é $u(x) = e^x$.

A solução fundamental $Y(x)$, satisfazendo $Y(0) = I$, tem a forma

$$Y(x) = \begin{bmatrix} \cosh \lambda x & \lambda^{-1} \operatorname{senh} \lambda x \\ \lambda \operatorname{senh} \lambda x & \cosh \lambda x \end{bmatrix}.$$

Para $\lambda = 1$ foi usada uma tolerância de $Tol = 10^{-10}$ nos métodos SSHTS e SSHTSR, e $Tol = 10^{-6}$ para $\lambda = 10$. Para $\lambda > 10$ os métodos falham, devido ao crescimento de λx. Nos métodos do *shooting* múltiplo, MSHTP, MSHTPC e MSHPR, foi usada $Tol = 10^{-10}$ e uma malha fixa de 10 subintervalos. O MSHTPC apresenta comportamanto semelhante ao método do *shooting* simples e, como era esperado, mostrou-se mais instável que as outras versões, falhando para $\lambda = 50$. Nessas implementações, a tolerância dada é a menor permitida, uma vez que valores menores para Tol causam erro na rotina de cálculo dos PVIs.

No pacote MUS, foi usada $Tol = 10^{-6}$ e 10 pontos de saída, com os erros[1] para os

[1]Na determinação do erro foi usada a norma do máxima, ou seja, erro $= \max_{1 \le i \le N} ||e_i||$, onde e_i é a diferença entre a solução exata e a solução aproximada pelo método e N é o número de pontos de saída. Para facilitar a impressão de resultados nas tabelas, $a \times 10^e$ será escrita $a \pm e$.

respectivos valores de λ apresentados como segue

λ	erro exato
1.0	0.31-08
10.0	0.34-10
20.0	0.27-11
50.0	0.90-13

Como pode ser observado, o comportamanto do pacote MUS é completamante diferente das outras implementações do método de valor inicial. Isto se deve ao fato de que, nesta implementação, os modos de crescimento e decrescimento da solução fundamantal estarem desacoplados, e a solução ser determinada na direção onde o crescimento não causa instabilidade, conforme capítulo 4, §4.2.

O pacote COLNEW, com $Tol = 10^{-6}$, resolveu o problema com uma malha de 10 subintervalos, com 4 pontos de colocação por subintervalo da malha, para todos os valores de λ. O erro estimado (conforme capítulo 4 §4.3), o erro exato e o erro nos pontos de saída são dados como segue

λ	erro estimado	erro exato	erro nos pontos de saída
1.0	0.18-08	0.19-08	0.47-12
10.0	0.18-08	0.19-08	0.11-11
20.0	0.16-08	0.18-08	0.45-11
50.0	0.11-08	0.16-08	0.18-10

quando os pontos de saída coincidem com os pontos da malha, observa-se que o erro é muito menor que o erro em um ponto qualquer no intervalo.

A tabela 5.1 contém os erros ocorridos no componente y_1 da solução, onde $\mathbf{y} = [y_1\ y_2]^t = [u(x)\ u'(x)]^t$, nos pontos de saída indicados. Os erros em y_2 se comportam de maneira semelhante aos de y_1.

Pode-se observar que a qualidade da solução obtida por MUS e COLNEW é consideravelmente superior àquela obtida pelas outras implementações.

Tabela 5.1 - Os erros no componente y_1 da solução do Problema 1

$\lambda = 1.0$

x	SSHTS	SSHTSR	MSHTP	MSHTPC	MSHTPR	MUS	COLNEW
0.0	0.22-15	0.00-00	0.44-15	0.22-15	0.66-15	0.22-15	0.00-00
0.1	0.88-07	0.20-11	0.44-15	0.31-02	0.30-02	0.11-08	0.15-12
0.2	0.17-06	0.18-11	0.97-02	0.63-02	0.60-02	0.18-08	0.27-12
0.3	0.26-06	0.15-11	0.14-01	0.96-02	0.91-02	0.25-08	0.37-12
0.4	0.36-06	0.12-11	0.19-01	0.13-01	0.12-01	0.29-08	0.43-12
0.5	0.45-06	0.10-11	0.25-01	0.16-01	0.15-01	0.31-08	0.47-12
0.6	0.56-06	0.74-12	0.30-01	0.20-01	0.19-01	0.31-08	0.47-12
0.7	0.66-06	0.19-12	0.36-01	0.24-01	0.22-01	0.28-08	0.42-12
0.8	0.78-06	0.25-12	0.43-01	0.28-01	0.26-01	0.22-08	0.33-12
0.9	0.96-06	0.11-14	0.49-01	0.32-01	0.30-01	0.13-08	0.19-12
1.0	0.10-05	0.00-00	0.00-00	0.82-07	0.82-07	0.22-15	0.22-15

$\lambda = 10.0$

x	SSHTS	SSHTSR	MSHTP	MSHTPC	MSHTPR	MUS	COLNEW
0.0	0.00-00	0.00-00	0.22-15	0.00-00	0.22-15	0.00-00	0.00-00
0.1	0.33-06	0.10-06	0.69-05	0.11-05	0.45-05	0.24-10	0.44-12
0.2	0.11-05	0.22-06	0.22-04	0.50-05	0.12-04	0.33-10	0.65-12
0.3	0.30-05	0.60-06	0.63-04	0.14-05	0.33-04	0.34-10	0.77-12
0.4	0.84-05	0.16-05	0.17-03	0.41-04	0.90-04	0.22-10	0.88-12
0.5	0.22-04	0.14-05	0.47-03	0.11-03	0.24-03	0.22-10	0.97-12
0.6	0.62-04	0.12-04	0.12-02	0.30-03	0.66-03	0.27-10	0.10-11
0.7	0.16-03	0.32-04	0.35-02	0.83-03	0.18-02	0.31-10	0.11-11
0.8	0.45-03	0.88-04	0.95-02	0.22-02	0.49-02	0.34-10	0.11-11
0.9	0.12-02	0.24-03	0.26-01	0.61-02	0.13-01	0.27-10	0.87-12
1.0	0.30-02	0.10-12	0.51-06	0.82-07	0.82-07	0.22-15	0.22-15

$\lambda = 20.0$

x	SSHTS	SSHTSR	MSHTP	MSHTPC	MSHTPR	MUS	COLNEW
0.0	0.22-15	0.00-00	0.22-15	0.00-00	0.00-00	0.00-00	0.00-00
0.1	0.61-05	0.18-06	0.99-03	0.62-06	0.62-06	0.20-11	0.20-11
0.2	0.13-03	0.12-05	0.20-02	0.47-06	0.47-06	0.27-11	0.25-11
0.3	0.10-02	0.93-05	0.31-02	0.15-06	0.20-06	0.17-11	0.28-11
0.4	0.77-02	0.69-04	0.43-02	0.12-05	0.94-05	0.23-11	0.31-11
0.5	0.56-01	0.51-03	0.57-02	0.11-04	0.92-05	0.27-11	0.34-11
0.6	0.42+00	0.37-02	0.74-02	0.88-04	0.69-04	0.12-11	0.37-11
0.7	0.31+01	0.27-01	0.93-02	0.65-03	0.51-03	0.23-11	0.41-11
0.8	0.22+02	0.26-00	0.11-01	0.48-02	0.51-03	0.22-11	0.45-11
0.9	0.16+03	0.15+01	0.14-01	0.35-01	0.25-01	0.19-11	0.43-11
1.0	0.81+01	0.35-09	0.10-06	0.67-07	0.82-07	0.22-15	0.22-15

$\lambda = 50.0$

x	MSHTP	MSHTPC	MSHTPR	MUS	COLNEW
0.0	0.00-00	0.22-15	0.00-00	0.00-00	0.00-00
0.1	0.31-06	0.31-06	0.33-06	0.51-14	0.83-11
0.2	0.14-06	0.14-06	0.16-06	0.90-13	0.93-11
0.3	0.15-07	0.16-07	0.61-08	0.84-13	0.10-10
0.4	0.10-06	0.16-06	0.81-07	0.79-13	0.11-10
0.5	0.25-06	0.66-05	0.22-06	0.66-13	0.12-10
0.6	0.63-07	0.10-02	0.89-07	0.21-13	0.13-10
0.7	0.39-06	0.20+00	0.40-07	0.39-14	0.15-10
0.8	0.11-03	0.18+02	0.46-04	0.66-13	0.16-10
0.9	0.17-01	0.77+03	0.69-02	0.82-13	0.18-10
1.0	0.24-04	0.13+06	0.82-07	0.22-15	0.68-14

Para o pacote MUS, se for tomado um número de pontos de saída maior que 10, por exemplo 100 pontos, o erro na solução permanece o mesmo, o que era esperado, uma vez que os pontos de *shooting* são determinados com o processamanto, conforme capítulo 4, §4.2. Se a tolerância for menor, digamos $Tol = 10^{-10}$, a precisão do método aumenta consideravelmente, como pode ser observado a seguir

λ	erro exato
1.0	0.31-08
10.0	0.64-14
20.0	0.26-14
50.0	0.97-14

No pacote COLSYS, o uso de uma tolerância, $Tol = 10^{-10}$ produz um erro estimado de 0.20×10^{-10} que é igual ao erro exato. A malha usada foi de 24 subintervalos, com 4 pontos de colocação por subintervalo, para todos os valores de λ.

Problema 2. Seja o problema linear na forma matricial

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}' = \begin{bmatrix} 0 & \lambda \\ \lambda & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \lambda \cos^2 \pi\, x + \frac{2}{\lambda}\pi^2 \cos 2\,\pi\, x \end{bmatrix}, \qquad 0 < x < 1,$$

com condições de contorno

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_1(1) \\ y_2(1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

e solução analítica

$$\begin{bmatrix} y_1(x) \\ y_2(x) \end{bmatrix} = \begin{bmatrix} \frac{e^{\lambda(x-1)}+e^{-\lambda x}}{1+e^{-\lambda}} - \cos^2 \pi\, x \\ \frac{e^{\lambda(x-1)}-e^{-\lambda x}}{1+e^{-\lambda}} + \frac{\pi}{\lambda} \operatorname{sen} 2\pi\, x \end{bmatrix}.$$

A solução fundamanteal $Y(x)$, satisfazendo $Y(0) = I$, tem a forma

$$Y(x) = \begin{bmatrix} \cosh \lambda x & \operatorname{senh} \lambda x \\ \operatorname{senh} \lambda x & \cosh \lambda x \end{bmatrix}.$$

Como no problema 1, quando λ cresce, $||Y(x)||$ também cresce, portanto o comportamento da solução obtida pelos vários métodos de valor inicial é semelhante ao apresentado naquele problema. Neste problema a tolerância dada foi 10^{-6} para todos os métodos. O pacote MUS apresentou os seguintes erros

λ	erro exato
1.0	0.44-06
10.0	0.49-06
20.0	0.48-06
50.0	0.47-06

O pacote COLNEW, com 4 pontos de colocação por subintervalo da malha, apresentou os erros indicados abaixo, onde N é o número de subintervalos da malha.

λ	N	erro estimado	erro exato	erro nos pontos de saída
1.0	20	0.37-07	0.29-07	0.31-12
10.0	40	0.19-07	0.16-07	0.23-12
20.0	36	0.49-07	0.80-07	0.36-07
50.0	80	0.48-07	0.39-07	0.16-07

Os erros em u ou y_1, onde $\mathbf{y} = [y_1\ y_2]^t = [u\ u']^t$ estão listados na tabela 5.2.1. Os erros em y_2 se comportam de maneira semelhante aos de y_1.

Tabela 5.2.1 - Os erros no componente y_1 da solução do problema 2

				$\lambda = 1.0$			
x	SSHTS	SSHTSR	MSHTP	MSHTPC	MSHTPR	MUS	COLNEW
0.0	0.89-17	0.00-00	0.34-17	0.88-17	0.12-15	0.13-16	0.00-00
0.1	0.11-06	0.14-06	0.68-02	0.81-03	0.48-03	0.30-07	0.30-07
0.2	0.62-07	0.12-06	0.13-01	0.16-02	0.98-03	0.10-06	0.33-07
0.3	0.10-07	0.10-06	0.20-01	0.24-02	0.14-02	0.17-06	0.30-07
0.4	0.37-07	0.85-07	0.28-01	0.33-02	0.20-02	0.15-06	0.29-07
0.5	0.23-07	0.13-06	0.35-01	0.42-02	0.25-02	0.23-06	0.21-07
0.6	0.55-07	0.13-06	0.43-01	0.51-02	0.31-02	0.19-06	0.53-07
0.7	0.87-07	0.13-06	0.52-01	0.61-02	0.37-02	0.38-06	0.68-08
0.8	0.12-06	0.14-06	0.61-01	0.72-02	0.43-02	0.44-06	0.91-08
0.9	0.16-06	0.14-06	0.70-01	0.83-02	0.50-02	0.31-06	0.18-08
1.0	0.35-06	0.13-16	0.32-07	0.13-16	0.13-16	0.13-16	0.22-16

$\lambda = 10.0$

x	SSHTS	SSHTSR	MSHTP	MSHTPC	MSHTPR	MUS	COLNEW
0.0	0.82-16	0.00-00	0.98-16	0.82-16	0.12-15	0.00-00	0.00-00
0.1	0.31-07	0.33-07	0.65-05	0.44-05	0.19-05	0.51-07	0.23-12
0.2	0.55-07	0.14-06	0.20-04	0.13-06	0.60-05	0.13-06	0.17-12
0.3	0.13-06	0.41-06	0.56-04	0.38-04	0.16-04	0.19-06	0.81-12
0.4	0.35-06	0.11-05	0.15-03	0.10-03	0.45-04	0.16-06	0.25-13
0.5	0.96-06	0.31-05	0.41-03	0.28-03	0.12-03	0.34-08	0.66-14
0.6	0.26-05	0.84-05	0.11-02	0.76-03	0.33-03	0.23-06	0.25-13
0.7	0.71-05	0.22-04	0.30-02	0.20-02	0.91-03	0.43-06	0.81-13
0.8	0.19-04	0.62-04	0.83-02	0.56-02	0.24-02	0.49-06	0.17-12
0.9	0.52-04	0.16-03	0.22-01	0.15-01	0.67-02	0.34-06	0.23-12
1.0	0.60-03	0.90-12	0.41-06	0.90-12	0.13-16	0.13-16	0.13-16

$\lambda = 20.0$

x	SSHTS	SSHTSR	MSHTP	MSHTPC	MSHTPR	MUS	COLNEW
0.0	0.22-15	0.00-00	0.66-16	0.14-16	0.55-16	0.27-16	0.00-00
0.1	0.38-07	0.10-07	0.24-07	0.24-07	0.27-07	0.48-07	0.77-08
0.2	0.15-06	0.21-06	0.22-07	0.27-07	0.28-07	0.12-06	0.46-08
0.3	0.31-06	0.24-05	0.59-07	0.11-06	0.11-06	0.18-06	0.33-08
0.4	0.20-05	0.18-04	0.58-06	0.72-06	0.75-06	0.15-06	0.80-08
0.5	0.15-04	0.13-03	0.46-05	0.50-05	0.52-05	0.29-08	0.19-07
0.6	0.11-03	0.99-03	0.34-04	0.37-04	0.34-04	0.23-06	0.99-08
0.7	0.83-03	0.73-02	0.25-03	0.27-03	0.28-03	0.42-06	0.30-08
0.8	0.61-02	0.54-01	0.18-02	0.20-02	0.21-02	0.48-06	0.72-08
0.9	0.45-01	0.40-00	0.13-01	0.15-01	0.15-04	0.34-06	0.44-15
1.0	0.25-00	0.37-08	0.46-06	0.37-08	0.44-15	0.00-00	0.13-15

$\lambda = 50.0$

x	MSHTP	MSHTPC	MSHTR	MUS	COLNEW
0.0	0.69-16	0.27-15	0.45-16	0.13-16	0.00-00
0.1	0.24-07	0.20-07	0.20-07	0.52-07	0.22-09
0.2	0.27-07	0.19-07	0.19-07	0.12-06	0.51-10
0.3	0.60-07	0.47-07	0.48-07	0.18-06	0.44-09
0.4	0.75-07	0.00-00	0.57-07	0.15-06	0.50-09
0.5	0.24-08	0.85-05	0.22-10	0.47-07	0.79-09
0.6	0.92-07	0.10-02	0.49-07	0.25-06	0.22-08
0.7	0.27-06	0.95-01	0.41-06	0.42-06	0.30-08
0.8	0.28-04	0.65-00	0.69-04	0.47-06	0.15-09
0.9	0.41-02	0.51+03	0.10-01	0.34-06	0.16-07
1.0	0.14-04	0.65+05	0.71-14	0.13-16	0.13-15

Com o aumento da precisão, $Tol = 10^{-10}$, o erro em MUS permaneceu em 0.47×10^{-06} para todos os valores de λ. No pacote COLNEW, houve um aumento grande no número

de subintervalos da malha, que pode ser visto abaixo

λ	N	erro estimado	erro exato
1.0	160	0.11-11	0.91-12
10.0	192	0.13-11	0.18-11
20.0	36	0.63-13	0.59-11

Para $\lambda = 50$ o tamanho da malha exigida supera o número máximo de subintervalos permitido.

Este problema satisfaz às restrições da implementação do método dos elementos finitos, ou seja, o problema tem a forma $(-p\,u')' + q\,u = f$, com $u(0) = u(1) = 0$. O erro no componente y_1 da solução, usando ELFEL e ELFSC, é mostrado na tabela 5.2.2.

Tabela 5.2.2 - Os erros no componente y_1 da solução do problema 2

	$\lambda = 1.0$		$\lambda = 10.0$		$\lambda = 20.0$		$\lambda = 50.0$	
x	ELFEL	ELFSC	ELFEL	ELFSC	ELFEL	ELFSC	ELFEL	ELFSC
0.0	0.00-00	0.00-00	0.00-00	0.00-00	0.00-00	0.00-00	0.00-00	0.00-00
0.1	0.28-04	0.82-04	0.22-01	0.35-03	0.74-01	0.23-02	0.19-00	0.62-02
0.2	0.19-03	0.44-04	0.14-01	0.11-03	0.17-01	0.59-03	0.25-01	0.11-01
0.3	0.41-03	0.40-04	0.26-02	0.91-04	0.25-02	0.37-03	0.37-03	0.71-02
0.4	0.59-03	0.93-04	0.60-02	0.11-03	0.11-01	0.75-04	0.14-01	0.42-02
0.5	0.66-03	0.11-03	0.91-02	0.13-03	0.15-01	0.25-03	0.16-01	0.36-02
0.6	0.59-03	0.93-04	0.60-02	0.11-03	0.11-01	0.75-04	0.14-01	0.42-02
0.7	0.41-03	0.40-04	0.26-02	0.91-03	0.25-02	0.37-03	0.37-03	0.71-02
0.8	0.19-03	0.44-04	0.14-01	0.11-03	0.17-01	0.59-03	0.25-01	0.11-01
0.9	0.28-04	0.82-04	0.22-01	0.35-03	0.74-01	0.23-02	0.10-00	0.63-02
1.0	0.00-00	0.00-00	0.66-15	0.66-15	0.15-14	0.15-14	0.41-14	0.41-14

Observa-se que a implementação usando splines cúbicos (ELFSC) apresenta uma pequena vantagem sobre a implementação usando elementos lineares (ELFEL), quanto à qualidade da solução, mas ainda inferior àquelas apresentadas pelos pacotes MUS e COLNEW.

Problema 3. Considere problema linear

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}' = \begin{bmatrix} 1 - 19\cos 2x & 0 & 1 + 19\,\mathrm{sen}\,2x \\ 0 & 19 & 0 \\ -1 + 19\,\mathrm{sen}\,2x & 0 & 1 + 19\cos 2x \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} + \begin{bmatrix} (-1 + 19\cos 2x - 19\,\mathrm{sen}\,2x)e^x \\ -18\,e^x \\ (1 - 19\cos 2x - 19\,\mathrm{sen}\,2x)e^x \end{bmatrix}, \qquad 0 < x < b,$$

com condições de contorno não separáveis

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1(0) \\ y_2(0) \\ y_3(0) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1(b) \\ y_2(b) \\ y_3(b) \end{bmatrix} = \begin{bmatrix} 1+e^b \\ 1+e^b \\ 1+e^b \end{bmatrix},$$

e solução analítica

$$\begin{bmatrix} y_1(x) \\ y_2(x) \\ y_3(x) \end{bmatrix} = \begin{bmatrix} e^x \\ e^x \\ e^x \end{bmatrix}.$$

A matriz solução fundamental é da forma

$$Y(x) = \begin{bmatrix} \operatorname{sen} x & 0 & -\cos x \\ 0 & 1 & 0 \\ \cos x & 0 & \operatorname{sen} x \end{bmatrix} \operatorname{diag}(e^{20x}, e^{19x}, e^{-18x}).$$

O método SSHTS falhou para intervalos com $b \geq 1$, o mesmo aconteceu com os métodos MSHTP, MSHTPC e MSHTR, para $b > \pi$.

A tabela 5.3 mostra os erros no componente y_1 da solução nos pontos de saída indicados, ocorridos na resolução do problema usando uma tolerância de 10^{-6} e $b = \pi$. Para os métodos MSHTP, MSHTPC e MSHTR, foi tomada uma malha fixa de 10 subintervalos. No pacote MUS também foram considerado 10 pontos de saída com a mesma tolerância dada.

Em MUS, se a tolerância for menor, $Tol = 10^{-10}$, o resultado se torna mais preciso, com um erro de 0.24×10^{-13}.

Tabela 5.3 - Os erros no componente y_1 da solução do problema 3

x	MSHTP	MSHTR	MUS
0.00	0.30-15	0.11-05	0.61-10
0.31	0.34-06	0.40-06	0.54-10
0.62	0.12-06	0.11-06	0.38-10
0.94	0.18-06	0.31-06	0.81-10
1.25	0.45-06	0.14-06	0.83-10
1.57	0.34-06	0.35-06	0.84-10
1.88	0.33-06	0.60-06	0.23-10
2.19	0.11-05	0.15-06	0.44-10
2.51	0.17-15	0.11-05	0.92-10
2.82	0.10-04	0.10-04	0.87-10
3.14	0.96-05	0.77-05	0.61-10

Problema 4 . Seja o PVC não linear

$$\begin{cases} u'' + e^u = 0, & 0 < x < 1, \\ u(0) = u(1) = 0, \end{cases}$$

com solução analítica

$$u(x) = -2\ln\frac{\cosh((x-\frac{1}{2})\frac{\theta}{2})}{\cosh(\frac{\theta}{4})}, \quad \theta = 1.5171646.$$

Este problema foi resolvido com uma solução inicial, $u(0) = 0.0$ e $u'(0) = 0.54935$. O método do *shooting* múltiplo, MSHTNL, com tolerâncias 10^{-11} e 10^{-6} e com uma malha fixa de 10 subintervalos, convergiu após 12 e 6 iterações,respectivamente. No pacote MUS, com 10 pontos de saída e uma tolerância inicial de 10^{-2} e final de 10^{-6}, convergiu após 2 iterações com um erro de 0.26×10^{-07}. O aumento dos pontos de saída não alterou os resultados. No pacote COLNEW, com $Tol = 10^{-6}$, o problema foi resolvido com uma malha de 10 subintervalos e 3 pontos de colocação por subintervalo da malha, e convergiu após 1 iteração, com um erro estimado igual ao erro exato de 0.14×10^{-08} e nos pontos da malha o erro foi de 0.18×10^{-09}.

A tabela 5.4 apresenta os erros ocorridos em u nos pontos de saída indicados.

Tabela 5.4 - erros na solução do problema 4

x	MSHTNL	MUS	COLNEW
0.0	0.00-00	0.54-19	0.00-00
0.1	0.33-03	0.74-08	0.64-10
0.2	0.65-03	0.14-07	0.11-09
0.3	0.97-03	0.20-07	0.15-09
0.4	0.12-02	0.25-07	0.17-09
0.5	0.15-02	0.26-07	0.18-09
0.6	0.18-02	0.25-07	0.17-09
0.7	0.21-02	0.20-07	0.15-09
0.8	0.23-02	0.14-07	0.11-09
0.9	0.25-02	0.75-08	0.64-09
1.0	0.40-12	0.00-00	0.42-16

Com uma tolerância menor, $Tol = 10^{-10}$, o pacote MUS, com 10 pontos de saída, que convergiu após 2 iterações, apresentou um erro de 0.17×10^{-09}. O pacote COLNEW, usando uma malha com 40 subintervalos, convergiu após 1 iteração. O erro estimado foi de

0.14×10^{-11} e o erro exato de 0.17×10^{-09}, com erro igual nos pontos de saída, os quais coincidem com pontos da malha.

Problema 5. Este problema, apresentado em Ascher et al.[1981], exemplo 2, descreve pequenas deformações finitas de uma tampa esférica delgada, de densidade constante, sujeita a variações de pressão externa, distribuída quadrática e assimetricamente e de uma variação de pressão interna, distribuída uniformemente.

$$\frac{\epsilon^4}{\mu}[\Phi'' + \frac{1}{x}\Phi' - \frac{1}{x^2}\Phi] + \Psi(1 - \frac{1}{x}\Phi) - \Phi = -\gamma x(1 - \frac{1}{2}x^2), \qquad 0 < x < 1,$$
$$\mu[\Psi'' + \frac{1}{x}\Psi' - \frac{1}{x^2}\Psi] - \Phi(1 - \frac{1}{2x}\Phi) = 0,$$

sujeito às condições de contorno

$$\Phi = x\Psi' - 0.3\Psi + 0.7x = 0, \quad \text{em} \quad x = 0 \quad \text{e} \quad 1,$$

onde Φ representa as mudanças do ângulo de deformação da casca esférica e Ψ é a função tensão. Este problema não tem solução analítica conhecida.

Foi resolvido somente pelo pacote COLNEW (este problema apresenta uma singularidade, $\frac{\epsilon^4}{\mu}$ multiplicando o termo de mais alta ordem da equação, e as outras implementações não tratam desses casos), com duas soluções iniciais

a. $\Phi = \Psi = 0,$ e

b. $\Phi = \begin{cases} 2x, & x \le x_t, \\ 0, & x > x_t, \end{cases} \qquad \Psi = \begin{cases} -2x + \gamma x(1 - \frac{1}{2}x^2), & x \le x_t, \\ -\gamma x(1 - \frac{1}{2}x^2), & x > x_t, \end{cases}$

onde $x_t = \sqrt{2(\gamma - 1)\gamma}$ e as constantes $\epsilon = 10^{-2}$, $\mu = \epsilon$. Com uma tolerância de 10^{-6} e 4 pontos de colocação por subintervalos da malha.

Usando a solução inicial **a**, o resultado obtido está listado na tabela 5.5a, a malha usada foi de 52 subintervalos, dos quais 65% estão contidos no intervalo $[0.9, 1.0]$. As estimativas dos erros para Φ, Φ', Ψ, Ψ' foram 0.26×10^{-08}, 0.23×10^{-05}, 0.35×10^{-09} e 0.58×10^{-07}, respectivamente.

Tabela 5.5a - Resultado do problema 5

x	Φ	Φ'	Ψ'	Ψ'
0.0	0.00-00	0.47-01	0.17-25	-0.11+01
0.1	0.47-02	0.47-01	-0.10-00	-0.10+10
0.2	0.94-02	0.47-01	-0.21-00	-0.10+01
0.3	0.41-01	0.47-01	-0.31-00	-0.94+01
0.4	0.18-01	0.47-01	-0.40-00	-0.82-00
0.5	0.23-01	0.46-01	-0.48-00	-0.67-00
0.6	0.28-01	0.40-01	-0.53-00	-0.48-00
0.7	0.30-01	0.55-02	-0.57-00	-0.26-00
0.8	0.25-01	-0.14-00	-0.59-00	-0.60-01
0.9	-0.12-01	-0.75-00	-0.59-00	-0.28-01
1.0	-0.24-13	0.11+03	-0.62-00	-0.88-00

Usando a solução inicial **b**, o resultado foi obtido com uma malha de 112 subintervalos, dos quais 35% estão contidos no intervalo $[0.3, 0.4]$ e 28% estão contidos em $[0.9, 1.0]$. O resultado obtido está na tabela 5.5b. As estimativas de erro para Φ, Φ', Ψ, Ψ' foram 0.24×10^{-07}, 0.86×10^{-05}, 0.73×10^{-09} e 0.94×10^{-07}, respectivamente.

Tabela 5.5b - Resultado do problema 5

x	Φ	Φ'	Ψ'	Ψ'
0.0	0.00+00	0.20+01	0.29-27	-0.90+00
0.1	0.20+00	0.20+01	-0.90-01	-0.91+00
0.2	0.40+00	0.20+01	-0.18+00	-0.96+00
0.3	0.61+00	0.20+01	-0.28+00	-0.10+01
0.4	0.83+00	-0.24+00	-0.39+00	-0.12+01
0.5	0.22-01	0.12+00	-0.48+00	-0.67+00
0.6	0.28-01	0.41-01	-0.53+00	-0.48+00
0.7	0.30-01	0.55-02	-0.57+00	-0.26+00
0.8	0.25-01	-0.14+00	-0.59+00	-0.60-01
0.9	-0.12-01	-0.75+00	-0.59+00	-0.28-01
1.0	-0.24-13	0.11+03	-0.62+00	-0.88+00

5.2 - Comentários Finais

Para um PVC/EDO bem condicionado, o método da superposição, com *shooting* simples ou múltiplo, frequentemente calcula soluções de problemas de valor inicial mal condicionados, o que leva a dificuldades quanto à estabilidade do método, principalmente

se a solução for calculada em intervalos muito longos. Nestes casos, somente informações sobre o modo de crescimento rápido são capturadas, e, eventualmente, o método falha.

Os resultados obtidos nas implementações dos métodos do *shooting* simples e do *shooting* múltiplo, nas versões MSHTP, MSHTPC e MSHTR, confirmam as dificuldades apresentadas teoricamante, ou seja, estes métodos devem ser usados somente quando os problemas não apresentarem um subespaço de solução com crescimento dominante.

As implementações do método de elementos finitos apresentaram resultados bastante pobres. Estes resultados não retratam dificuldades no método em si, mas deficiência nas implementações, que foram feitas, como nos outros métodos, diretamente a partir de sua conceituação.

A implementação profissional do *shooting* múltiplo, o pacote MUS, supera o problema da instabilidade e obtém resultados aceitáveis, tanto no caso linear como no caso não linear, de acordo com uma tolerância dada.

A implementação profissional do método da colocação, o pacote COLNEW, apresentou bons resultados, tanto no caso linear como no caso não linear, confirmando as alegações feitas sobre seu desempenho na literatura especializada.

Comparando o pacote COLNEW com o pacote MUS, o primeiro é mais eficiente, no sentido de que resolve uma classe de problemas maior que MUS e também porque na maioria dos casos testados (veja parágrafo anterior), os resultados apresentados por COLNEW têm uma precisão maior que os apresentados por MUS.

Para realizar os testes, apresentados no parágrafo anterior, não foi usado nenhum programa de teste, por não tê-los disponíveis. Tanto a escolha dos problemas, como a comparação dos pacotes foram feitas sem nenhuma metodologia, apenas resolvendo alguns problemas com solução analítica conhecida e observando a precisão da resposta obtida.

APÊNDICE 1

Fórmulas de Runge-Kutta-Fehlberg

Sejam o PVI $\mathbf{y}' = \mathbf{f}(x, \mathbf{y})$, $a < x < b$ e $\mathbf{y}(a) = \mathbf{y}_1$, a malha $\Pi : a = x_1 < \cdots < x_{N+1} = b$, a solução aproximada $\{\mathbf{y}_i : i = 1, \ldots, N+1\}$ e $h = h_i = x_{i+1} - x_i$.

A forma geral das fórmulas de Runge-Kutta de k-ésima ordem é

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h \sum_{j=1}^{k} \beta_j \mathbf{g}_j \qquad i = 1 = 1, 2, \ldots, N+1$$

onde $\mathbf{g}_j = \mathbf{f}(x_i + h\rho_j, \mathbf{y}_i + h \sum_{l=1}^{k} \alpha_{jl}\mathbf{g}_l)$ e ρ_j, β_j, α_{jl} são constantes conhecidas tal que $0 \leq \rho_j \leq 1$.

As fórmulas são chamadas explícitas se $\alpha_{jl} = 0$ para $l \geq j$ e implícitas caso contrário.

Fórmula explícita de Runge-Kutta-Classica de 4ª ordem

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h\left(\frac{1}{6}\mathbf{g}_1 + \frac{2}{6}\mathbf{g}_2 + \frac{2}{6}\mathbf{g}_3 + \frac{1}{6}\mathbf{g}_4\right),$$

onde

$$\begin{aligned}
\mathbf{g}_1 &= \mathbf{f}(x_i, \mathbf{y}_i), \\
\mathbf{g}_2 &= \mathbf{f}(x_i + \frac{h}{2}, \mathbf{y}_i + \frac{h}{2}\mathbf{g}_1), \\
\mathbf{g}_3 &= \mathbf{f}(x_i + \frac{h}{2}, \mathbf{y}_i + \frac{h}{2}\mathbf{g}_2), \\
\mathbf{g}_4 &= \mathbf{f}(x_i + h, \mathbf{y}_i + h\mathbf{g}_3).
\end{aligned}$$

Fórmulas explícitas de Runge-Kutta-Fehlberg de 4ª e 5ª ordem. Nesta familia de formulas os g_i's da fórmula de 4ª ordem e de 5ª ordem são os mesmos.

Fórmula de 4ª ordem

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h\left(\frac{25}{210}\mathbf{g}_1 + \frac{1408}{2565}\mathbf{g}_3 + \frac{2197}{4104}\mathbf{g}_4 - \frac{1}{5}\mathbf{g}_5\right),$$

fórmula de 5ª ordem

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h\,(\frac{16}{135}\mathbf{g}_1 + \frac{6656}{12825}\mathbf{g}_3 + \frac{28561}{56430}\mathbf{g}_4 - \frac{9}{5065}\mathbf{g}_5 + \frac{2}{55}\mathbf{g}_6),$$

onde

$$\begin{aligned}
\mathbf{g}_1 &= \mathbf{f}(x_i, \mathbf{y}_i),\\
\mathbf{g}_2 &= \mathbf{f}(x_i + \frac{h}{4}, \mathbf{y}_i + \frac{h}{4}\mathbf{g}_1),\\
\mathbf{g}_3 &= \mathbf{f}(x_i + \frac{3h}{8}, \mathbf{y}_i + h\,(\frac{3}{32}\mathbf{g}_1 + \frac{9}{32}\mathbf{g}_2)),\\
\mathbf{g}_4 &= \mathbf{f}(x_i + \frac{12\,h}{13}, \mathbf{y}_i + h\,(\frac{1932}{2197}\mathbf{g}_1 - \frac{7200}{2197}\mathbf{g}_2 + \frac{7296}{2197}\mathbf{g}_3)),\\
\mathbf{g}_5 &= \mathbf{f}(x_i + h, \mathbf{y}_i + h\,(\frac{439}{216}\mathbf{g}_1 - 8\mathbf{g}_2 + \frac{3680}{513}\mathbf{g}_3 - \frac{845}{4104}\mathbf{g}_4)),\\
\mathbf{g}_6 &= \mathbf{f}(x_i + \frac{h}{2}, \mathbf{y}_i + h\,(-\frac{8}{27}\mathbf{g}_1 + 2\mathbf{g}_2 - \frac{3544}{2565}\mathbf{g}_3 + \frac{1859}{4104}\mathbf{g}_4 - \frac{11}{40}\mathbf{g}_5)).
\end{aligned}$$

APÊNDICE 2

Exemplo da Eespecificação dos Problemas e da Estrutura da Matriz Resultante da Discretização de cada Método.

Todos os métodos estudados para resolver numéricamente um PVC/EDO recaem na solução de um sistema linear, $A\,\mathbf{y} = \mathbf{b}$. Serão exemplificadas, aqui, a matriz A resultante em cada método implementado e nos pacotes MUS e COLNEW, assim como a especificação do problema pelo usuário.

Exemplo Linear

Considere o problema 2, capítulo 5, §5.1, com $\lambda = 10$.

$$\begin{cases} u'' = 10^2 u + 10\cos^2 \pi\, x + \frac{2\,\pi^2}{10}\cos 2\,\pi\, x \\ u(0) = u(1) = 0. \end{cases}$$

Nas implementações do método do *shooting* múltiplo (MSHTP, MSHTC, MSHTR), o problema é especificado de acordo com as exigências do pacote EPISODE, usado para resolver os PVIs, da seguinte forma.

```
C
C
      SUBROUTINE DFFUN (N,X,Y,YP)
      INTEGER N
      DOUBLE PRECISION X, Y(1), YP(1), LAMBDA. PI
      LAMBDA = 10.D0
      PI = DARCOS(-1)
      YP(1) = LAMBDA * Y(2)
      YP(2) = LAMBDA * Y(1) + (LAMBDA * (DCOS(PI * X)**2) +
     1            (((2.DO * PI**2) / LAMBDA) * (DCOS(2.D0 * PI * X))).
      RETURN
      END
C
C
      SUBROUTINE PEDERV (N,NMAX,X,Y,DY,N0)
      INTEGER N, NMAX, N0
      DOUBLE PRECISION X, Y(NMAX,13), DY(1), LAMBDA
```

```
      LAMBDA = 10.D0
      DY(1) = 0.D0
      DY(2) = LAMBDA
      DY(3) = LAMBDA
      DY(4) = 0.D0
      RETURN
      END
C
C
      SUBROUTINE FUNC (N,X,Y,YP)
      INTEGER N
      DOUBLE PRECISION X, Y(1), YP(1), LAMBDA
      LAMBDA = 10.D0
      YP(1) = LAMBDA * Y(2)
      YP(2) = LAMBDA * Y(1)
      RETURN
      END
C
C
      SUBROUTINE CCONT (N,BA,BB,BETA)
      INTEGER N
      DOUBLE PRECISION BA(N,N), BB(N,N), BETA(N)
      BA(1,1) = 1.D0
      BA(1,2) = 0.D0
      BA(2,1) = 0.D0
      BA(2,2) = 0.D0
      BB(1,1) = 0.D0
      BB(1,2) = 0.D0
      BB(2,1) = 1.D0
      BB(2,2) = 0.D0
      BETA(1) = 0.D0
      BETA(2) = 0.D0
      RETURN
      END
C
C
```

Em todas as implementações o problema foi resolvido com o número de pontos de *shooting* fixo, N = 10. O sistema de equações resultante, conforme capítulo 3, §3.1.2, onde $Y_i, F_{i+1}, B_a, B_b \in \mathbb{R}^{2\times 2}$ e $\tilde{\mathbf{s}}_i, \mathbf{v}_i \in \mathbb{R}^{2\times 1}$.

Em MSHTP as matrizes $F_i = I$, os vetores $\mathbf{v}_{oi} = 0$ para $i = 1, 2, \ldots, 10$. No sistema linear $A\,\widetilde{\mathbf{s}} = \widehat{\mathbf{d}}$, a matriz $A \in I\!R^{20\times 20}$ e o vetor $\widehat{\mathbf{d}} \in I\!R^{20\times 1}$ e A tem a forma

$$A = \begin{bmatrix}
\times & \times & \times & \times & & & & & & & & \\
\times & \times & \times & \times & & & & & & & & \\
 & & \times & \times & \times & \times & & & & & & \\
 & & \times & \times & \times & \times & & & & & & \\
 & & & & & & \ddots & & & & & \\
 & & & & & & & \times & \times & \times & \times & \\
 & & & & & & & \times & \times & \times & \times & \\
\times & \times & & & & & & & & \times & \times & \\
\times & \times & & & & & & & & \times & \times &
\end{bmatrix},$$

Em MSHTC os vetores $\widetilde{\mathbf{s}}$ são encontrados explorando a expressão recursiva (3.7a), onde as matrizes $\Gamma_i = Y_i(x_{i+1})$, são usadas para determinar as matrizes Φ_i e os vetores $\mathbf{r}_i$ em (3.7b,c) e

$$\widetilde{\mathbf{s}}_i = \Phi_i \widetilde{\mathbf{s}}_1 + \mathbf{r}_i.$$

O vetor $\widetilde{\mathbf{s}}_1$ é determinado resolvendo o sistema

$$(B_a + B_b\,\Phi_{N+1})\widetilde{\mathbf{s}}_1 = \mathbf{d} - B_b\,\mathbf{r}_{N+1}$$

ou seja,

$$\begin{bmatrix} 0.100{+}01 & 0.00{+}00 \\ 0.125{+}05 & 0.125{+}05 \end{bmatrix} \begin{bmatrix} \widetilde{s}_1 \\ \widetilde{s}_2 \end{bmatrix} = \begin{bmatrix} 0.000{+}00 \\ -0.125{+}05 \end{bmatrix}.$$

Em MSHTR a matriz $Y_i(x_{i+1})$ é decomposta no produto $F_{i+1}\widehat{\Gamma}_i$, onde F_{i+1} é ortogonal e $\widehat{\Gamma}_i$ é triangular superior.

A matriz F_1 foi tomada como a identidade e os vetores $\widetilde{\mathbf{s}}$ são determinados pela resoluão

do sistema $A\tilde{\mathbf{s}} = \hat{\mathbf{d}}$, onde a matriz a tem a mesma estrutura da matriz do MSHTP

$$A = \begin{bmatrix} -\Gamma_1 & I & 0 & \dots & 0 \\ 0 & -\Gamma_2 & I & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & & -\Gamma_{N-1} & I \\ B_a & \dots & & 0 & B_b\Gamma_N \end{bmatrix} =$$

$$= \begin{bmatrix} \times & \times & \times & \times & & & & & & \\ \times & \times & \times & \times & & & & & & \\ & & \times & \times & \times & \times & & & & \\ & & \times & \times & \times & \times & & & & \\ & & & & & & \ddots & & & \\ & & & & & & & \times & \times & \times & \times \\ & & & & & & & \times & \times & \times & \times \\ \times & \times & & & & & & & & \times & \times \\ \times & \times & & & & & & & & \times & \times \end{bmatrix},$$

No pacote MUS o problema linear é especificado como segue

```
C
C
      SUBROUTINE FLIN(T,Y,F)
      DOUBLE PRECISION T, Y(2), F(2), LAMBDA
      LAMBDA = 10.D0
      F(1) = LAMBDA * Y(2)
      F(2) = LAMBDA * Y(1)
      RETURN
      END
C
C
      SUBROUTINE FDIF (T,Y,F)
      DOUBLE PRECISION T, Y(2), F(2)
      CALL FLIN (T,Y,F)
      F(1) = F(1)
      F(2) = F(2) + (LAMBDA * (DCOS(PI * X)**2) +
     1          (((2.DO * PI**2) / LAMBDA) * (DCOS(2.D0 * PI * X))).
      RETURN
      END
C
C
```

Este pacote explora a forma recursiva, como em MSHTC para determinar $\mathbf{s}_i$ e a reortogonalização da matriz $Y_i(x_{i+1})$ como em MSHTR conforme capítulo 4, §4.2, obtendo a matriz Φ_i e vetor $\mathbf{r}_i$.

Na implementação do método de elementos finitos descritos no capítulo 3, §3.4, o problema é especificado da seguinte forma

```
C
C
      DOUBLE PRECISION FUNCTION P(X)
      DOUBLE PRECISION X
      P = 0.1D1
      RETURN
      END
C
C
      DOUBLE PRECISION FUNCTION Q(X)
      DOUBLE PRECISION X, LAMBDA
      LAMBDA = 0.10D2
      Q = -LAMBDA**2
      RETURN
      END
C
C
      DOUBLE PRECISION FUNCTION FF(X)
      DOUBLE PRECISION X, LAMBDA
      LAMBDA = 0.10D2
      PI = DARCOS(-1.D0)
      FF = LAMBDA**2 * (DCOS(PI * X)**2 +
     1      (2.DO * PI**2) * (DCOS(2.D0 * PI * X))
      RETURN
      END
C
C
```

A malha considerada na resuloção do problema é fixa com $N = 10$. A matriz A do sistema resultante, $A\mathbf{y} = \mathbf{d}$, é uma matriz de banda e simétrica.

Na versão usando Splines cúbicos (ELFSC), a matriz resultante $A \in I\!R^{(10+1)\times(10+1)}$ na forma mostrada abaixo.

$$A = \begin{bmatrix} \times & \times & \times & \times & & & & & & \\ \times & \times & \times & \times & \times & & & & & \\ \times & \times & \times & \times & \times & \times & & & & \\ \times & \times & \times & \times & \times & \times & \times & & & \\ & \times & \times & \times & \times & \times & \times & \times & & \\ & & \times & \times & \times & \times & \times & \times & \times & \\ & & & \times & \times & \times & \times & \times & \times & \times \\ & & & & \times & \times & \times & \times & \times & \times \\ & & & & & \times & \times & \times & \times & \times \\ & & & & & & \times & \times & \times & \times \end{bmatrix},$$

Na versão usando Elementos Lineares (ELFEL), a matriz resultante $A \in \mathbb{R}^{(10-1)\times(10-1)}$ na forma mostrada a seguir

$$A = \begin{bmatrix} \times & \times & & & & & & & \\ \times & \times & \times & & & & & & \\ & \times & \times & \times & & & & & \\ & & \times & \times & \times & & & & \\ & & & \times & \times & \times & & & \\ & & & & \times & \times & \times & & \\ & & & & & \times & \times & \times & \\ & & & & & & \times & \times & \times \\ & & & & & & & \times & \times \end{bmatrix}$$

No pacote COLNEW, o problema linear é especificado como segue.

```
C
C
      SUBROUTINE FSUB ( X,Z,F )
      DOUBLE PRECISION Z(1), F(1), X, LAMBDA, PI
      LAMBDA = 10.D0
      PI = DARCOS(-1.DO)
      F(1) = LAMBDA * Z(2)
      F(2) = LAMBDA * Z(1) + (LAMBDA * (DCOS(PI * X)**2) +
     1           (((2.DO * PI**2) / LAMBDA) * (DCOS(2.D0 * PI * X))).
      RETURN
      END
C
C
      SUBROUTINE DFSUB ( X,Z,DF )
      DOUBLE PRECISION Z(2), DF(2,2), X, LAMBDA
```

```
      LAMBDA = 10.D0
      DF(1,1) = 0.D0
      DF(1,2) = LAMBDA
      DF(2,1) = LAMBDA
      DF(2,2) = 0.D0
      RETURN
      END
C
C
      SUBROUTINE GSUB ( I,Z,G )
      INTEGER I
      DOUBLE PRECISION Z(1), G
      G = Z(1)
      RETURN
      END
C
C
      SUBROUTINE DGSUB ( I,Z,DG )
      INTEGER I
      DOUBLE PRECISION Z(2), DG(2)
      DG(1) = 1.D0
      DG(2) = 0.D0
      RETURN
      END
C
C
      SUBROUTINE EXACT ( X,U )
      DOUBLE PRECISION U(2), X, PI, LAMBDA
C . . . . . A SOLUCAO EXATA
      PI = DARCOS(-1.D0)
      LAMBDA = 10.D0
      U(1) = (DEXP(LAMBDA * (X - 1)) + DEXP(-LAMBDA * X)) /
             (1.D0 + DEXP(-LAMBDA)) - (DCOS(PI * X))**2
      U(2) = (DEXP(LAMBDA * (X - 1)) - DEXP(-LAMBDA * X)) /
             (1.D0 + DEXP(-LAMBDA)) + (PI/LAMBDA) * (DSIN(2.D0 * PI * X))
      RETURN
      END
C
C
```

O problema foi resolvido usando 4 pontos de colocação por subintervalo da malha com uma $Tol = 10^{-6}$. A determinação da solução depende da resolução de dois sistema

lineares, conforme (3.19), um para determinar as variáveis locais $\mathbf{z}_i$ em

$$W_i\mathbf{z}_i + V_i\mathbf{y}_i = \mathbf{q}_i$$

onde $W_i \in I\!R^{4\times 4}$ e $V_i \in I\!R^{4\times 2}$ triangular superior e o sistema para determinar $\mathbf{y}_i$, onde $\mathbf{y}_{i+1} = \Gamma_i\mathbf{y}_i + \mathbf{r}_i$, $\Gamma_i \in I\!R^{2\times 2}$ a matriz A do sistema resultante, $A\mathbf{y} = \mathbf{b}$, tem estrutura semelhante a do MSHTP

$$A = \begin{bmatrix} \times & \times & \times & \times & & & & & & & & \\ \times & \times & \times & \times & & & & & & & & \\ & & \times & \times & \times & \times & & & & & & \\ & & \times & \times & \times & \times & & & & & & \\ & & & & & & \ddots & & & & & \\ & & & & & & & \times & \times & \times & \times \\ & & & & & & & \times & \times & \times & \times \\ & & & & & & & & & \times & \times \\ & & & & & & & & & \times & \times \end{bmatrix},$$

A matriz A é uma matriz diagonal em blocos, e suas colunas são convenientemente rearranjadas para usar o pacote SOLVEBLOK na resolução do sistema. A malha usada para resolver o problema foi de 20 subintervalos, gerando portanto uma matriz de 20 blocos, cujos 19 primeiros de dimensões 3 e o último com dimensão 4.

Exemplo Não Linear

Seja o PVC

$$\begin{cases} u'' = e^u, & 0 < x < 1 \\ u(0) = u(1) = 0 \end{cases}$$

Na implemantação do método do *shooting* múltiplo padrão, MSHTNL, o problema é especificado da seguinte forma

```
C
C
      SUBROUTINE DFFUNL (N,X,Y,YP)
      INTEGER N
      DOUBLE PRECISION X, Y(1), YP(1)
      YP(1) = Y(2)
      YP(2) = -DEXP(Y(1))
      RETURN
      END
```

```
C
C
      SUBROUTINE DDFFNL (N,X,Y,YP)
      INTEGER N,I
      DOUBLE PRECISION X, Y(1), YP(1), YAUX(20,50)
      COMMON/ JAUX / YAUX, I
      YP(1) = Y(2)
      YP(2) = -DEXP(YAUX(1,I)) * Y(1)
      RETURN
      END
C
C
      SUBROUTINE DPDRNL (N,NMAX,X,Y,DY,N0)
      INTEGER N, NMAX, I
      DOUBLE PRECISION X, Y(NMAX,13), DY(1), YAUX(20,50)
      COMMON /JAUX/ YAUX, I
      DY(1) = 0.D0
      DY(2) = -DEXP(YAUX(1,I))
      DY(3) = 1.D0
      DY(4) = 0.D0
      RETURN
      END
C
C
      SUBROUTINE PDERNL (N,NMAX,X,Y,DY,N0)
      INTEGER N, NMAX, N0
      DOUBLE PRECISION X, Y(NMAX,13), DY(1)
      DY(1) = 0.D0
      DY(2) = -DEXP(Y(1,1))
      DY(3) = 1.D0
      DY(4) = 0.D0
      RETURN
      END
C
C
      SUBROUTINE CCONT (N,NMAX,M1,BA,BB)
      INTEGER N, NMAX
      DOUBLE PRECISION YAUX(20,50), BA(N,N), BB(N,N)
      COMMON /JAUX/ YAUX, I
      BA(1,1) = 1.D0
      BA(1,2) = 0.D0
```

```
      BA(2,1) = 0.D0
      BA(2,2) = 0.D0
      BB(1,1) = 0.D0
      BB(1,2) = 0.D0
      BB(2,1) = 1.D0
      BB(2,2) = 0.D0
      RETURN
      END
C
C
      SUBROUTINE CCONL (N,Y0,Y,G)
      INTEGER N
      DOUBLE PRECISION Y0(N), Y(N), G(N)
      G(1) = Y0(1)
      G(2) = Y(1)
      RETURN
      END
C
C
```

A convergência ocorreu na sétima iteração com o vetor $\mathbf{F}(\mathbf{s})$ e a A matriz $F'(\mathbf{s})$ tendo a mesma estrutura da matriz A do sistema linear do MSHTP

$$F(\mathbf{s}) = \begin{bmatrix} \times & \times & \times & \times & & & & & & & & \\ \times & \times & \times & \times & & & & & & & & \\ & & \times & \times & \times & \times & & & & & & \\ & & \times & \times & \times & \times & & & & & & \\ & & & & & & \ddots & & & & & \\ & & & & & & & \times & \times & \times & \times \\ & & & & & & & \times & \times & \times & \times \\ \times & \times & & & & & & & & \times & \times \\ \times & \times & & & & & & & & \times & \times \end{bmatrix},$$

No pacote MUS, a especificação do problema é da forma abaixo

```
C
C
      SUBROUTINE FDIF (T,Y,F)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION Y(2), F(2)
      F(1) = Y(2)
      F(2) = -DEXP(Y(1))
```

```
      RETURN
      END
C
C
C
      SUBROUTINE X0T (T,X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(2)
      X(1) = 0.D0
      X(2) = 0.54935d0
      RETURN
      END
C
C
      SUBROUTINE G(N,XA,XB,FG,DGA,DGB)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XA(N), XB(N), FG(N), DGA(N,N), DGB(N,N)
      DO 1100 I = 1 , N
            DO 1100 J = 1 , N
                  DGA(I,J) = 0.D0
                  DGB(I,J) = 0.D0
1100        CONTINUE
            DGA(1,1) = 1.D0
            DGB(2,1) = 1.D0
            FG(1) = XA(1)
            FG(2) = XB(1)
      RETURN
      END
C
C
```

A determinação do vetor $\mathbf{r}_i$, $F'(\mathbf{s})\,\mathbf{r}_i = -\mathbf{F}(\mathbf{s})$, explora a forma recursiva semelhante a (3.7a)

$$-Y_i(x_{i+1})\mathbf{r}_i + F_{i+1}\mathbf{r}_{i+1} = \mathbf{s}_{i+1} - \mathbf{y}_i(x_{i+1};\mathbf{s}_i)$$

usando o mesmo procedimento do caso linear e obtendo matrizes com a mesma estrutura.

No pacote COLNEW o problema não linear é especificado como segue.

```
C
C
      SUBROUTINE FSUB ( X,Z,F )
```

```
      DOUBLE PRECISION Z(1), F(1), X
      F(1) = - DEXP(Z(1))
      RETURN
      END
C
C
      SUBROUTINE DFSUB ( X,Z,DF )
      IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION Z(1), DF(1,1), X
      DF(1,1) = -DEXP(Z(1))
      DF(1,2) = 0.D0
      RETURN
      END
C
C
      SUBROUTINE GSUB ( I,Z,G )
      INTEGER I
      DOUBLE PRECISION Z(1), G
      G = Z(1)
      RETURN
      END
C
C
      SUBROUTINE DGSUB ( I,Z,DG )
      INTEGER I
      DOUBLE PRECISION Z(1), DG(1)
      DG(1) = 1.D0
      DG(2) = 0.D0
      RETURN
      END
C
C
      SUBROUTINE EXACT ( X,U )
      DOUBLE PRECISION U(1), X, TETA
C . . . . . A SOLUCAO EXATA
      TETA= 1.517646D0
      U(1) = -2*DLOG(DCOSH((X-0.5D0)*TETA/2.D0)/(DCOSH(TETA/2.D0)))
      U(2) = -4*DSINH((X-0.5)*TETA/2.D0)/(DCOSH((X-0.5)*TETA/2.D0)*TETA)
      RETURN
      END
C
```

```
C
      SUBROUTINE SOLUTN (X,Z,DMVAL)
      DOUBLE PRECISION Z(2), DMVAL(2)
      Z(1) = 0.D0
      Z(2) = 0.54935D0
      DMVAL(1) = 0.D0
      DMVAL(2) = 0.D0
      RETURN
      END
C
C
```

O problema não linear é resolvido usando 3 pontos de colocação por subintervalo da malha, e o processo de *quasilinearização*, conforme §4.3.5. A solução convergiu na primeira iteração, com uma malha de 10 subintervalos. Com as matrizes resultantes $V \in \mathbb{R}^{3\times 2}$, $W \in \mathbb{R}^{3\times 3}$ e $A \in \mathbb{R}^{20\times 20}$. A matriz A, como no caso linear, é convenientemente rearranjada, por colunas, na forma diagonal em blocos, com 9 blocos 3×4 e 1 bloco 4×4 para o uso do pacote SOLVEBLOK na resolução do sistema linear.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Ascher, Uri. 1986. *Collocation for two-point boundary value problems revisited, SIAM J. Numer. Anal.* **23**, 576 - 609.

[2] Ascher, Uri. 1980. *Solving boundary-value problems with a spline-collocation code, J. Comp. Phys.* **34**, 401-413.

[3] Ascher, U., Christiansen, J. and Russel, R.D. 1979. *A collocation solver for mixed order systems of boundary-value problems, Math. Comp.* **33**, 650 - 679.

[4] Ascher, U., Christiansen, J. and Russel, R.D. 1981a. *Collocation software for boundary-value ODEs, ACM Trans. Math. Softw.* **7**, 209-222.

[5] Ascher, U., Christiansen, J. and Russel, R.D. 1981b. *Algorithm 569 - COLSYS: collocation software for boundary-value ODEs [D2], ACM Trans. Math. Softw.* **7**, 223-229.

[6] Ascher, U., Mattheij, R. M. M. and Russel, R. D. 1988. *Numerical Solution of Boundary-value problems for Ordinary Differential Equations.* Prentice-Hall, Englewood Cliffs, New Jersey.

[7] Ascher, U., Pruss, S. and Russel, R. D. 1983. *On spline basis selection for solving differential equations,SIAM J. Numer. Anal.* **20**, 121-142.

[8] Bader, G. and Ascher, U. 1987. *A new basis implementation for a mixed order boundary-value ODE solver,SIAM J. Sci. Stat. Comput.* **8**, 483-500.

[9] Boor, Carl de. 1978. *A practical Guide to Splines.* Springer-Verlag, Berlin.

[10] Boor, Carl de and Swartz, B. 1973. *Collocation at Gaussian points, SIAM J. Numer. Anal.* **10**, 583-606.

[11] Boor, Caarl de. and Weiss, R. 1980a. *SOLVEBLOK: a package for solving almost block diagonal linear systems, ACM Trans. Math. Softw.* **6**, 80-87.

[12] Boor, Caarl de. and Weiss, R. 1980b. *Algorithm 546: SOLVEBLOK [F4], ACM Trans. math. softw.* **6**, 88-91.

[13] Childs B. et al. 1979. *Codes for Boundary-value Problems in Ordinary Differential Equations.* Lecture Notes in Computer Science, Springer-Verlag, Berlin.

[14] Coddington, E. A. and Levinson, N. 1955. *Theory of Ordinary Differential Equations.* McGraw-Hill, New York.

[15] Dongarra, J. J., et al. 1979. *LINPACK. User's Guide.* Society for Industrial and Applied Mathematics Publications, Philadelphia.

[16] Golub, G. H. and Van Loan, C. F. 1983. *Matrix Computations.* Johns Hopkins Univ. Press, Baltimore.

[17] Johnston, R. L. 1982. *Numerical Methods: A Software Approach.* John Wiley & Sons, New York.

[18] Lentini, M., Osborne, M. R. and Russel, R. D. 1985. *The close relationships between methods for solving two-point boundary-value problems*, *SIAM J. Anal. Numer.* **22**, 280-309.

[19] Manteuffel, Thomas A. and White, Andrew D. JR. 1986. *On the efficient numerical solution of systems of second order boundary-value problems*, *SIAM J. Numer. Anal.* **23**, 998-1006.

[20] Mattheij, R. M. M. and Staarink, G. W. M. 1984a. *On optimal shooting intervals*, *Math. Comp.* **42**, 25-40.

[21] Mattheij, R. M. M. and Staarink, G. W. M. 1984b. *An efficient algorithm for solving general linear two point BVP*, *SIAM J. Sci. Stat. Comp.* **5**, 745-763.

[22] Narici, Bachman. 1966. *Functional Analysis.* Academic Press, London.

[23] Oden, J. T and Reddy J. N. 1976. *Introduction to the Matematical Theory of Finite Elements.* John Wiley & Sons, New York.

[24] Prenter, P. M. 1975. *Splines and Variational Methods.* John Wiley & Sons, New York.

[25] Scott, M. R. and Watts, Herman A. 1977. *Computational solution of linear two point boundary-value problems via orthonormalizations*, *SIAM J. Numer. Anal.* **14**, 40-70.

[26] Soutomayor, Jorge. 1979. *Lições de Equações Diferenciais Ordinárias.* Coleção Projeto Euclides, IMPA-CNPq, Rio de Janeiro.

[27] Strang, G. and Fix, G. J. 1973. *An Analysis of the Finite Element Methods.* Prentice Hall, Englewood Cliffs, New Jersey.