

CELSO BERNARDO NÓBREGA FREITAS

Integração numérica de sistemas
não lineares semi-implícitos
via teoria de controle geométrico

SÃO PAULO
2007

CELSO BERNARDO NÓBREGA FREITAS

Integração numérica de sistemas
não lineares semi-implícitos
via teoria de controle geométrico

Monografia apresentada ao Instituto de Matemática e
Estatística para a conclusão do Curso de Bacharelado em
Matemática Aplicada Computacional

Área de Concentração: Engenharia de Sistemas

Orientador: Paulo Sérgio Pereira da Silva

Dedicatória

Dedico este trabalho a meus pais, que estão sempre ao meu lado.

Agradecimentos

Agradeço ao Professor Paulo Sérgio pelo apoio e motivação.

Resumo

Este trabalho estuda duas abordagens para solução de sistemas de controle não lineares semi-implícitos. Ambos os métodos tentam encontrar, via controle geométrico, um sistema explícito cujas soluções converjam para as soluções do sistema original. O primeiro método aplica somente técnicas de desacoplamento e estabilização por realimentação de estado, ao passo que o segundo método, considera um campo vetorial no espaço de estados e de entradas de maneira a obter um novo sistema que siga as soluções do sistema original. Como será visto nos experimentos numéricos, ambos os métodos asseguram convergência, mas o segundo método é numericamente mais estável que o primeiro.

Abstract

This work studies two approaches to solve semi-implicit non-linear control systems. Both methods try to find, via geometric control, a completely explicit in which its solutions converge to the solutions of the original system. The first method uses only the decoupling and feedback state stabilization techniques, while the second method, considers a vectorial field on state space and the input space, in order to obtain a new system that follow the solutions of the original system. As will be shown in the numerical experiments, both methods assure convergence, but the second method is numerically more stable than the first one.

SUMÁRIO

Introdução.....	1
Capítulo 1: Fundamentos Teóricos para o Método I	3
Teoria do Desacoplamento	3
Existência de Coordenadas Complementares \hat{x}	9
Estrutura do Método I	17
Sistemas Semi-Implícitos	13
Capítulo 2: Fundamentos Teóricos para o Método II	15
Estrutura do Método II	16
Teoremas de Convergência para o Método II	18
Obtendo Coordenadas Complementares \hat{x}	19
Capítulo 3: Algoritmos e Construção de Simulação.....	21
Obtendo a Aplicação $Y^{(\rho)}$	21
Obtendo Coordenadas Complementares \hat{x}	22
Construindo Implementação Método I.....	23
Construindo Implementação Método II	25
Capítulo 4: Experimentos Numéricos	28
Exemplo 1: Matrizes Mal Condicionadas	28
Exemplo 2: Pêndulo	39
Conclusão	45
Apêndice.....	46
Referências Bibliográficas	52

Introdução

Neste trabalho lidaremos com Equações Diferenciais Algébricas (DAEs), em particular com sistemas não lineares semi-implícitos invariantes no tempo da forma

$$\begin{cases} \dot{x}(t) = f[x(t), u(t)] \\ y(t) = h[x(t)] = 0 \end{cases}$$

Onde $x \in \mathbb{R}^n$ é o vetor de estados do sistema, $u \in \mathbb{R}^m$ é o vetor de entradas e $y \in \mathbb{R}^l$ é o vetor de saídas. Para um aberto $\Omega \in \mathbb{R}^n$, $f: \Omega \rightarrow \mathbb{R}^n$ exprime a dinâmica do sistema e $h: \Omega \rightarrow \mathbb{R}^l$ representa o conjuntos de restrições para os estados $x \in \mathbb{R}^n$.

Vejam um sistema no qual $\dim u = 2$ e $\dim y = 1$, escrito explicitamente à direita.

$$\begin{cases} \dot{x}_1 = x_2 + u_1 \\ \dot{x}_2 = u_2 \\ y = x_2 = 0 \end{cases} \Leftrightarrow \begin{cases} \dot{x}_1 = u_1 \end{cases}$$

Verificamos que u_2 deve ser identicamente nulo para que a restrição $x_2 = 0$ seja obedecida, por esse motivo u_2 pode ser chamado de pseudo-entrada [3]. Vemos também, a partir do lado direito da expressão acima, que o mesmo não ocorre com u_1 , já que este pode ser escolhido arbitrariamente. Assim, a entrada u_1 recebe a denominação de verdadeira entrada [3].

Analisemos agora outro exemplo simples, onde $\dim u = \dim y = 1$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u \\ y = x_2 = 0 \end{cases} \Leftrightarrow \begin{cases} \dot{x}_1 = 0 \end{cases}$$

Neste caso não existem entradas verdadeiras, por conseguinte o sistema será completamente determinado. A esta classe de sistemas na qual a entrada e a saída possuem a mesma dimensão, dá-se o nome sistemas quadrados, e é nela que nos concentraremos.

Estudaremos dois métodos para a obtenção de soluções desta família de sistemas. Ambos tentam encontrar um sistema explícito cujas soluções converjam para as soluções do sistema original, utilizando para isso a teoria geométrica do desacoplamento [2].

Nos sistemas explícitos assim gerados, poderemos aplicar então as técnicas tradicionais de integração numérica.

O primeiro método, rotulado de Método I [1], considera uma transformação de coordenadas sobre o espaço de estados e uma realimentação de estado estática na forma $u = \alpha(x) + \beta(x)v$, obtendo um sistema modificado.

Este sistema modificado será desacoplado, i.e., cada saída y_i será controlada separadamente por uma das componentes v_i . Mais ainda, mesmo para uma condição inicial que não obedeça exatamente à restrição, os estados do sistema serão levados cada vez mais próximos da restrição. Formalmente, para uma condição inicial x_0 tal que $|h(x_0)| < \varepsilon$, $\varepsilon > 0$, as saídas y_i convergirão assintoticamente para o vetor nulo.

O segundo método, chamado de Método II [1,3], foi elaborado com o intuito de obter melhor convergência do que o Método I em algumas situações encontradas em problemas reais de controle, dentre elas, matrizes mal condicionadas e não linearidades [1].

No Método II é considerada uma transformação de coordenadas sobre o produto cartesiano do espaço de estados e de entradas, obtendo outro sistema modificado e desacoplado. Com base neste novo sistema tenta-se estabelecer um campo vetorial τ , no qual as soluções do sistema $\begin{pmatrix} \dot{x} \\ \dot{u} \end{pmatrix} = \tau(x, u)$ convirjam para as soluções do sistema original de forma mais robusta.

A Teoria do Desacoplamento e os resultados necessários para Método I podem ser encontrados no Capítulo 1. Ainda neste capítulo, discutiremos sobre a existência de coordenadas complementares que formarão a transformação de coordenadas e veremos como um sistema explícito pode ser equivalente a um sistema semi-implícito.

No Capítulo 2 exporemos os fundamentos do Método II, mencionando teoremas de convergência. Apresentaremos também argumentos que nos permitirão compor um difeomorfismo local através de uma escolha numérica de coordenadas complementares.

A combinação dos resultados teóricos, para os dois métodos, organizada em formato de algoritmo está presente no Capítulo 3, juntamente com especificações de construção das simulações. Por fim, no Capítulo 4, realizaremos experimentos numéricos para aferir a precisão dos métodos, comparando-os sob diferentes aspectos.

Capítulo 1: Fundamentos Teóricos para o Método I

Este capítulo abrange aspectos teóricos referentes ao Método I, onde analisamos como e sob quais condições é possível construir uma transformação de coordenadas e uma realimentação de estado para um sistema de controle explícito, tal que, cada uma das saídas y_i seja afetada por uma e apenas uma entrada u_i . Trataremos de sistemas quadrados, i.e sistemas cuja saída e entrada possuam dimensão de mesmo tamanho.

Discutiremos sobre a existência de coordenadas complementares, que comporão o uma transformação de coordenadas diferenciável, e como sistemas explícitos dados numa estrutura especial são equivalentes a semi-implícitos.

Teoria do Desacoplamento

Seja o seguinte sistema não linear afim explícito, dado por sua representação de estado:

$$\begin{cases} \dot{x}(t) = f[x(t)] + \sum_{i=1}^m g_i[x(t)]u_i(t) \\ y(t) = h[x(t)] \end{cases} \quad (1.1a)$$

$$(1.1b)$$

Onde

$t \in I$ é a variável de tempo

$x(t) \in X \subset \mathbb{R}^n$ é o espaço de estado

$u(t) \in U \subset \mathbb{R}^m$ é o espaço de entrada

$y(t) \in Y \subset \mathbb{R}^l$ é o espaço de saída (com $l = m$ para sistemas quadrados)

$f(\cdot) \in C^\infty : X \rightarrow X$ descreve a dinâmica do sistema com entrada nula ($u \equiv 0$)

$g_i(\cdot) \in C^\infty : X \rightarrow X$ descreve como a entrada u_i modifica a dinâmica do sistema

$h(\cdot) \in C^\infty : X \rightarrow Y$ associa um estado do sistema a sua saída

A fim de simplificar a notação, omitiremos a dependência de t ou x quando não isto prejudicar a leitura.

Denotaremos $y_i(t_0, x_0, t, u), i \in \{1, \dots, m\}$ as saídas do sistema (1.1) no instante t , aplicando-se a entrada u , a partir de da condição inicial x_0 no instante t_0 .

Definição 1.1 Para o sistema (1.1), uma saída y_i é dita invariante sob uma entrada u_j em X_0 se e somente se para todo $u = (u_1, \dots, u_j, \dots, u_m)$ e $\tilde{u} = (u_1, \dots, \tilde{u}_j, \dots, u_m)$

$$y_i(t_0, x_0, t, u) = y_i(t_0, x_0, t, \tilde{u}), \quad \forall x_0 \in X_0 \quad (1.2)$$

Ou seja, a entrada u_j não altera a saída y_i , independentemente da escolha de u_j .

Teorema 1.2 Considere o conjunto de funções abaixo

$$L_{g_j} L_{X_1} \dots L_{X_k} h_i(x), \quad \forall k \geq 0, X_1, \dots, X_k \in \{f, g_1, \dots, g_m\}, x \in X_0 \quad (1.3)$$

A saída y_i é invariante sob uma entrada u_j se (1.3) é identicamente nulo para todo $x \in X_0$

Prova: Vide [2]

Definição 1.3 O Sistema (2.1) é chamado de desacoplado em X_0 se e somente se, após reordenação das entradas, valem as seguintes propriedades:

- (i) Para cada $i \in \{1, \dots, m\}$, a saída y_i é invariante sob as entradas u_j , sempre que $j \neq i$
- (ii) A saída y_i é não invariante com respeito à entrada u_i , $i \in \{1, \dots, m\}$.

Utilizando o Teorema 1.2, obtemos uma condição suficiente para (i)

$$\forall j \neq i, L_{g_j} L_{X_1} \dots L_{X_k} h_i(x) = 0, \quad \forall k \geq 0, X_1, \dots, X_k \in \{f, g_1, \dots, g_m\}, x \in X_0 \quad (1.4)$$

De (1.3), extraímos a expressão da influência de u_i sobre y_i

$$L_{g_j} L_{X_1} \dots L_{X_k} h_i(x), \quad \forall k \geq 0, X_1, \dots, X_k \in \{f, g_1, \dots, g_m\}, x \in X_0 \quad (1.5)$$

Para (ii), consideraremos um subconjunto de (1.5)

$$L_{g_i} L_f^k h_i(x), \quad \forall k \geq 0, x \in X_0 \quad (1.6)$$

Proposição 1.4 Caso (1.5) seja identicamente nula

$$L_{g_i} L_f^k h_i(x) = 0, \quad \forall k \geq 0, x \in X_0 \quad (1.7)$$

Então, todas as funções em (1.5) também serão identicamente nulas.

Prova

Para $k = 0$

$$L_{g_i} L_f^0 h_i(x) = 0 \Rightarrow L_{g_i} h_i(x) = 0 \quad (1.8)$$

Para $k > 0$

$$L_{g_i} L_{X_1} \dots L_{X_k} h_i(x) = \begin{cases} L_{g_i} L_{X_1} \dots L_{g_i} h_i(x) = 0, & \text{(de (2.8))} \\ L_{g_i} L_{X_1} \dots L_{g_j} h_i(x) = 0, & \text{(de (2.4))} \\ L_{g_i} L_{X_1} \dots L_{X_{k-1}} L_f h_i(x), & \text{(a partir da validade da expressão (1.7) para } k-1) \end{cases}$$

Assumindo-se que (1.4) seja verdadeira e (1.7) seja falsa, a seguinte construção é admissível para todo $i \in \{1, \dots, m\}$

$$y_i(x) = h_i(x)$$

Derivando-se y_i em relação a t

$$\dot{y}_i = \frac{d}{dt}(y_i(x(t))) = \frac{\partial h_i}{\partial x} \dot{x} = \frac{\partial h_i}{\partial x} \left(f(x) + \sum_{k=1}^m g_k(x) u_k \right) = L_f h_i + L_{g_i} h_i u \doteq h_i^{(1)}(x, u)$$

Se $L_{g_i} h_i \equiv 0$, $h_i^{(1)}(x) = L_f h_i$. Neste caso, derivaremos a expressão mais uma vez.

$$\dot{h}_i^{(1)} = \frac{d}{dt}(h_i^{(1)}(x(t))) = \frac{\partial h_i^{(1)}}{\partial x} \dot{x} = \frac{\partial h_i^{(1)}}{\partial x} \left(f(x) + \sum_{k=1}^m g_k(x) u_k \right) = L_f^2 h_i^{(1)} + L_{g_i} L_f h_i^{(1)} u \doteq h_i^{(2)}(x, u)$$

Novamente, se $L_{g_i} L_f h_i \equiv 0$, $h_i^{(2)}(x) = L_f^2 h_i$.

De modo geral, pode-se gerar $h_i^{(k)}$, enquanto $h_i^{(k-1)}(x, u) = h_i^{(k-1)}(x)$.

$$\dot{h}_i^{(k-1)} = \frac{d}{dt}(h_i^{(k-1)}(x(t))) = \frac{\partial h_i^{(k-1)}}{\partial x} \dot{x} = \frac{\partial h_i^{(k-1)}}{\partial x} \left(f + \sum_{k=1}^m g_k u_k \right) = L_f^k h_i^{(k-1)} + L_{g_i} L_f^{k-1} h_i^{(k-1)} u \doteq h_i^{(k)}(x, u)$$

Definição 1.5 Definiremos números característicos ρ_i , inteiros não negativos, para todo $i \in \{1, \dots, m\}$, como o menor k tal que $h_i^{(k)}$ dependa explicitamente de u , ou seja

$$\left\{ \begin{array}{l}
 y_i \doteq h_i^{(0)} (= y_i^{(0)}(x)) \\
 \dot{y}_i^{(0)} = \frac{\partial h_i^{(0)}}{\partial x} \dot{x} = L_f h_i^{(0)} + L_{g_i} h_i^{(0)} u_i = L_f h_i^{(0)} \doteq h_i^{(1)} (= y_i^{(1)}(x)) \\
 \vdots \\
 \dot{y}_i^{(k-1)} = \frac{\partial h_i^{(k-1)}}{\partial x} \dot{x} = L_f^k h_i^{(k-1)} + L_{g_i} L_f^{k-1} h_i^{(k-1)} u_i = L_f^k h_i^{(k-1)} \doteq h_i^{(k)} (= y_i^{(k)}(x)) \\
 \vdots \\
 \dot{y}_i^{(\rho_i-1)} = \frac{\partial h_i^{(\rho_i-1)}}{\partial x} \dot{x} = L_f^{\rho_i} h_i^{(\rho_i-1)} + L_{g_i} L_f^{\rho_i-1} h_i^{(\rho_i-1)} u_i \doteq h_i^{(\rho_i)} (= y_i^{(\rho_i)}(x, u))
 \end{array} \right. \quad (1.9a)$$

$$\text{Lembrando que } \left\{ \begin{array}{l}
 \forall (x, k) \in (X_0, \{0, \dots, \rho_{i-2}\}), L_{g_i} L_f^{k-1} h_i^{(k-1)} \equiv 0 \\
 \forall x \in X_0, L_{g_i} L_f^{\rho_i-1} h_i^{(\rho_i-1)} \neq 0
 \end{array} \right. \quad (1.9b)$$

Logo, no conjunto $X_0 = \{x \in X \mid L_{g_i} L_f^{\rho_i-1} h_i^{(\rho_i-1)} u_i \neq 0, i \in [1..m]\}$, a entrada u_i influencia a saída y_i instantaneamente.

Definição 1.6 O sistema (1.1) é dito fortemente desacoplado se (1.4) é válida e se existem número característicos ρ_1, \dots, ρ_m como em (1.9). Diremos que esta propriedade é global se X_0 coincidir com X ou local caso contrário [2]

Se o sistema (1.1) é não desacoplado, podemos tentar alterar a dinâmica do sistema adicionando uma realimentação.

Definição 1.7 Dado o sistema (1.1), uma realimentação de estado estática [2] tem o formato

$$u = \alpha(x) + \beta(x)v \quad (1.10)$$

Onde $\alpha: X \rightarrow R^m$, $\beta: X \rightarrow R^{m \times m}$ e $v \in R^m$. Sendo $\det(\beta(x)) \neq 0$ para todo $x \in X_0$. Aplicando-se (1.10) à (1.1) obtém-se o sistema modificado em malha fechada

$$\dot{x}(t) = \tilde{f}(x) + \sum_{i=1}^m \tilde{g}_i(x) v_i \quad (1.11)$$

Onde

$$\tilde{f}(x) = f(x) + \sum_{j=1}^m \tilde{g}_j(x) \alpha_j(x) \quad (1.12a)$$

$$\tilde{g}_i(x) = \sum_{j=1}^m g_j(x) \beta_{ji}(x), i \in [1..m] \quad (1.12b)$$

$$v = -\beta^{-1}(x) \alpha(x) + \beta^{-1} u \quad (1.12c)$$

Neste ponto, enunciaremos formalmente o problema de desacoplamento sugerido na introdução do capítulo.

Problema 1.8 O Problema de desacoplamento para o sistema original (1.1) consiste na obtenção de uma realimentação de estado estática (1.10) tal que o sistema modificado (1.11) torne-se fortemente desacoplado conforme Definição 1.6.

Lema 1.9 Uma realimentação de estado estática para o sistema (1.1) não altera os números característicos ρ_1, \dots, ρ_m .

Definição 1.10 Para qualquer sistema (1.1), é possível tentar o processo (1.9) e, caso existam inteiros finitos não negativos ρ_1, \dots, ρ_m , combinar os resultados como abaixo

$$y^{(\rho)} = a(x) + b(x)u \quad (1.13)$$

onde

$$y^{(\rho)} = \begin{pmatrix} y^{(\rho_1)} \\ \vdots \\ y^{(\rho_m)} \end{pmatrix}, \quad a(x) = \begin{pmatrix} L_f^{\rho_1} h_1 \\ \vdots \\ L_f^{\rho_m} h_m \end{pmatrix}, \quad b(x) = \begin{pmatrix} L_g L_f^{\rho_1-1} h_1 \\ \vdots \\ L_g L_f^{\rho_m-1} h_m \end{pmatrix} \doteq \begin{pmatrix} L_{g_1} L_f^{\rho_1-1} h_1 & \cdots & L_{g_m} L_f^{\rho_1-1} h_1 \\ \vdots & & \vdots \\ L_{g_1} L_f^{\rho_m-1} h_m & \cdots & L_{g_m} L_f^{\rho_m-1} h_m \end{pmatrix}$$

A matriz $m \times m$ $b(x)$ é chamada de matriz de desacoplamento [2].

Teorema 1.11 Seja o sistema (1.1), com números característicos finitos ρ_1, \dots, ρ_m . Então o Problema 1.7 de desacoplamento é solúvel através de uma realimentação de estado estática se e somente se

$$\text{posto } b(x) = m, \quad \forall x \in X \quad (1.14)$$

Prova

Trabalharemos primeiro com o sentido de equivalência mais simples, assumindo que posto $b(x) = m, \forall x \in X$ é válida, logo podemos derivar as saídas do sistema (2.1b), extraíndo a matriz de desacoplamento $b(x)$ de acordo com a definição 2.8

$$y^{(\rho)} = a(x) + b(x)u \quad (1.15)$$

A hipótese garante a existência de $b^{-1}(x)$, por conseguinte, imporemos (α, β) em uma realimentação de estado estática (1.10) $u = \alpha(x) + \beta(x)v$, onde

$$\alpha(x) = -b^{-1}(x)a(x) \quad (1.16a)$$

$$\beta(x) = b^{-1}(x) \quad (1.16b)$$

Combinando (1.15) e (1.16)

$$y^{(\rho)} = a + b(\alpha + \beta v) \quad (1.17a)$$

$$y^{(\rho)} = a + b(-b^{-1}a + b^{-1}v) \quad (1.17b)$$

$$y^{(\rho)} = v \quad (1.17c)$$

A partir de (1.17c) verifica-se que o sistema em malha fechada (1.11) é desacoplado em termos da Definição 1.5, pois existem, por hipótese, os números característicos finitos ρ_1, \dots, ρ_m e, além disso, cada uma das saídas y_i é influenciada por uma e apenas uma entrada v_i .

Agora, assumiremos a existência de uma solução para o problema de desacoplamento. Podemos projetar uma realimentação de estado estática $u = \alpha(x) + \beta(x)v$, tal que o sistema modificado (1.11) torne-se desacoplado.

Cabe ressaltar que a combinação de (1.10) e (1.13) resulta na expressão da matriz de desacoplamento para o sistema modificado (1.11)

$$y^{(\rho)} = a + bu \quad (1.18a)$$

$$y^{(\rho)} = a + b(\alpha + \beta v) \quad (1.18b)$$

$$y^{(\rho)} = (a + b\alpha) + (b\beta)v \quad (1.18c)$$

$$y^{(\rho)} = \tilde{a} + \tilde{b}v \quad (1.18d)$$

Além disso, por hipótese, existem números característicos finitos ρ_1, \dots, ρ_m que, pelo Lema 1.7, são os mesmos para o sistema original (1.1) e o modificado (1.11).

Já que o sistema modificado é desacoplado temos

$$\tilde{b}(x) = \begin{pmatrix} L_{\tilde{g}_1} L_{\tilde{f}}^{\rho_1-1} h_1 & \cdots & L_{\tilde{g}_m} L_{\tilde{f}}^{\rho_1-1} h_1 \\ \vdots & & \vdots \\ L_{\tilde{g}_1} L_{\tilde{f}}^{\rho_m-1} h_m & \cdots & L_{\tilde{g}_m} L_{\tilde{f}}^{\rho_m-1} h_m \end{pmatrix} = \begin{pmatrix} L_{\tilde{g}_1} L_{\tilde{f}}^{\rho_1-1} h_1 & & 0 \\ & \ddots & \\ 0 & & L_{\tilde{g}_m} L_{\tilde{f}}^{\rho_m-1} h_m \end{pmatrix} \quad (1.19)$$

Onde $\det(\tilde{b}(x)) \neq 0$, porque a não invariância de y_i sob u_i implica em $L_{\tilde{g}_1} L_{\tilde{f}}^{\rho_1-1} h_1 \neq 0$.

Temos também que $\det(\beta(x)) \neq 0$ em uma região X (vide Definição 1.7).

Como $\tilde{b}(x) = b(x)\beta(x)$, chegamos a tese (1.14) de que posto $b(x) = m$, $\forall x \in X_0$

Teorema 1.12 Sejam $\rho \doteq \sum_{i=1}^m \rho_i$, denominado grau relativo do sistema, e a aplicação $S : X \rightarrow R^\rho$ definida por

$$\left(h_1(x), L_f h_1(x), \dots, L_f^{\rho_1-1} h_1(x), h_2(x), \dots, h_m(x), \dots, L_f^{\rho_m-1} h_m(x) \right) \quad (1.20)$$

Quando a propriedade (1.14) é válida para o sistema (1.1), então

$$\text{posto } S(x) = \rho \quad (1.21)$$

Prova: Consultar apêndice.

A título de melhor visualização, introduziremos uma notação alternativa para a aplicação $S(x)$ em (1.20).

$$Y^{(\rho-1)}(x) = \left(y_1^{(0)}(x), \dots, y_1^{(\rho_1-1)}(x), \dots, y_m^{(0)}(x), \dots, y_m^{(\rho_m-1)}(x) \right) \quad (1.22)$$

Existência de Coordenadas Complementares \hat{x}

Proposição 1.13 Seja um conjunto de funções independentes $\varphi \doteq \{\varphi_i : U \rightarrow \mathbb{R}^n, i \in \{1, \dots, \rho\}\}$ sendo x_0 pertence ao aberto $U \subset \mathbb{R}^n$. Então, existe outro conjunto de funções $\hat{x} \doteq \{\hat{x}_i : V \rightarrow \mathbb{R}^n, i \in \{1, \dots, n - \rho\}, V \subset U\}$ tal que a aplicação $\psi : V \rightarrow \mathbb{R}^n$ [3]

$$\psi(x) = \begin{pmatrix} \varphi(x) \\ \hat{x}(x) \end{pmatrix} \quad (1.23)$$

é um difeomorfismo local em V .

Prova

Tomando-se a matriz jacobiana

$$J\varphi(x) = \begin{pmatrix} \frac{\partial \varphi_1}{\partial x_1} & \dots & \frac{\partial \varphi_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial \varphi_\rho}{\partial x_1} & \dots & \frac{\partial \varphi_\rho}{\partial x_n} \end{pmatrix} \quad (1.24)$$

A independência das funções em φ permite nos reordenar as colunas de (1.24) de tal maneira que as ρ primeiras colunas de $J\varphi$ formem uma submatriz não singular

$$J\varphi(x) = \begin{pmatrix} \frac{\partial \varphi_1}{\partial x_1} & \dots & \frac{\partial \varphi_1}{\partial x_\rho} & \frac{\partial \varphi_1}{\partial x_{\rho+1}} & \dots & \frac{\partial \varphi_1}{\partial x_n} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial \varphi_\rho}{\partial x_1} & \dots & \frac{\partial \varphi_\rho}{\partial x_\rho} & \frac{\partial \varphi_\rho}{\partial x_{\rho+1}} & \dots & \frac{\partial \varphi_\rho}{\partial x_n} \end{pmatrix} \quad (1.25)$$

Note que, por abuso de notação, a seqüência dos índices x_i foi mantida apesar da possível permutação das colunas de (1.24)

Indicaremos por $x_{\{k, \dots, k+l\}}$ as coordenadas de x restritas aos índices $\{k, \dots, k+l\}$.

$$W(x) \doteq \frac{\partial \varphi}{\partial x_{\{\rho+1, \dots, n\}}}. \quad (1.26)$$

Observe que

$$\frac{\partial \psi}{\partial x} \Big|_{x_0} = \begin{pmatrix} \frac{\partial \varphi}{\partial x} \\ \frac{\partial \hat{x}}{\partial x} \end{pmatrix} \Big|_{x_0} = \begin{pmatrix} \frac{\partial \varphi}{\partial x_{\{1, \dots, \rho\}}} & W(x) \\ 0 & I_{n-\rho} \end{pmatrix} \Big|_{x_0} \quad (1.27)$$

é não singular pois os elementos em sua diagonal são diferentes de zeros em X_0 . Portanto, $\psi(x)$ é um difeomorfismo numa vizinhança V de x_0 , de acordo com o Teorema da Aplicação Inversa [4].

Estrutura do Método I

Reunindo os resultados dos Teoremas 1.12 e 1.13, concluímos que posto $b(x) = m$, $\forall x \in X_0$ (2.14) é uma condição necessária e suficiente para que

$$z \doteq \psi(x) = \left(Y^{(\rho-1)}(x), \hat{x}(x) \right) \quad (1.28)$$

seja um difeomorfismo numa vizinhança V de x_0 .

Considerando a derivada em t de (1.28)

$$\dot{z} = \frac{\partial \psi(x)}{\partial x} \dot{x} \quad (1.29)$$

Iremos substituir (1.1a) em (1.29)

$$\dot{z} = \frac{\partial \psi(x)}{\partial x} f(x) + \sum_{i=1}^m \frac{\partial \psi(x)}{\partial x} g_i(x) u_i \quad (1.30a)$$

$$\dot{z} = \frac{\partial \psi(\psi^{-1}(z))}{\partial x} f(\psi^{-1}(z)) + \sum_{i=1}^m \frac{\partial \psi(\psi^{-1}(z))}{\partial x} g_i(\psi^{-1}(z)) u_i \quad (1.30b)$$

Utilizaremos a notação $(\psi_* f)(z) \doteq \frac{\partial \psi(\psi^{-1}(z))}{\partial x} f(\psi^{-1}(z))$, sugerindo a representação de f no espaço de coordenadas $z = \psi(x)$. Logo (1.30) equivale a

$$\dot{z} = (\psi_* f)(z) + \sum_{i=1}^m (\psi_* g_i)(z) u_i \quad (1.30c)$$

Ademais, utilizando a notação (1.22) para (1.9), vamos reescrever (1.29) na forma canônica

$$\left\{ \begin{array}{l} \dot{y}_1^{(0)} = y_1^{(1)} \\ \vdots \\ \dot{y}_1^{(\rho_1-2)} = y_1^{(\rho_1-1)} \\ \dot{y}_1^{(\rho_1-1)} = a_1 + b_1 u \\ \vdots \\ \dot{y}_m^{(0)} = y_m^{(1)} \\ \vdots \\ \dot{y}_m^{(\rho_m-2)} = y_m^{(\rho_m-1)} \\ \dot{y}_m^{(\rho_m-1)} = a_m + b_m u \\ \dot{\hat{x}} = \eta(\hat{x}, Y^{(\rho-1)}, u) \end{array} \right. \quad (1.31)$$

Onde a_i e b_i correspondem a i -ésima linha da matriz $a(x)$ e a matriz de desacoplamento $b(x)$ conforme Definição 1.10 e $\eta(\hat{x}, Y^{(\rho-1)}, u) \doteq (\hat{x}_* f)(z) + \sum_{i=1}^m (\hat{x}_* g_i)(z) u_i$

Por causa da relação entre η e sistemas semi-implícitos, como será visto na próxima seção, η recebe o nome de dinâmica zero.

Se impusermos a realimentação $u = \alpha(x) + \beta(x)v$ segundo (1.16) e acrescentarmos ao sistema as saídas (1.1b), com a mesma notação (1.22), obteremos

$$\left\{ \begin{array}{l} \dot{y}_1^{(0)} = y_1^{(1)} \\ \vdots \\ \dot{y}_1^{(\rho_1-2)} = y_1^{(\rho_1-1)} \\ \dot{y}_1^{(\rho_1-1)} = v_1 \\ \vdots \\ \dot{y}_m^{(0)} = y_m^{(1)} \\ \vdots \\ \dot{y}_m^{(\rho_m-2)} = y_m^{(\rho_m-1)} \\ \dot{y}_m^{(\rho_m-1)} = v_m \end{array} \right. \quad (1.32a)$$

$$\left\{ \begin{array}{l} y_1 = y_1^{(0)} \\ \vdots \\ y_m = y_m^{(0)} \end{array} \right. \quad (1.32b)$$

$$\left\{ \dot{\hat{x}} = \eta(\hat{x}, Y^{(\rho-1)}, v) \right. \quad (1.32c)$$

Temos então que em uma região X_0 tal que posto $b(x) = m, \forall x \in X_0$ (1.14) a transformação de coordenadas $z \doteq \psi(x) = (Y^{(\rho-1)}(x), \hat{x}(x))$ (1.28) aplicada ao sistema original (1.1) gera (1.31).

Adicionando-se a realimentação de estado estática (1.16) chegamos a um sistema em malha fechada (1.11) no formato (1.32), fortemente desacoplado (Teorema 1.6).

Observe também que, ao tratar de sistemas quadrados, não restam entradas para que possamos controlar diretamente as coordenadas \hat{x} .

A estrutura (1.32) será a base para a construção do Método I. Para controlar a velocidade de convergência, escolheremos cada uma das entradas v_i segundo um parâmetro $\gamma > 0$, onde

$$v_i = - \sum_{j=0}^{\rho_i-1} \alpha_{ij} y_i^{(j)}, \quad \forall i = \{1, \dots, m\} \quad (1.33a)$$

$$\pi_i(s) = (s + \gamma)^{\rho_i} = s^{\rho_i} + \sum_{j=0}^{\rho_i-1} \alpha_{ij} s^j \quad (1.33b)$$

Observe que o sistema (1.32a,b) com (1.33) é linear e possui todos autovalores iguais à $-\gamma$, o que implica em $\lim_{t \rightarrow \infty} y_i = 0 \forall i \in \{1, \dots, m\}$. Veja também que (1.32c) evolui somente de acordo com a dinâmica zero η .

Uma abordagem algorítmica deste método será feita no Capítulo 3.

Sistemas Semi-Implícitos

Consideremos agora sistemas semi-implícitos

$$\begin{cases} \dot{x}(t) = f[x(t)] + \sum_{i=1}^m g_i[x(t)]u_i(t) & (1.34a) \\ y(t) = h[x(t)] \equiv 0 & (1.34b) \end{cases}$$

Ao contrário do sistema explícito (1.1) não mais podemos escolher arbitrariamente as entradas u para o sistema semi-implícito (1.34) devido à restrição (1.34b). De um ponto de vista geométrico, podemos dizer que, no espaço $M = X_0 \times U \subset \mathbb{R}^{n \times m}$, estamos presos à superfície de nível $h(x) = 0$, que depende implicitamente de u devido às relações diferenciais expressas por (1.34a).

Estabeleceremos o conjunto

$$\Gamma = \{(x, u) \in M \mid Y^{(\rho)}(x, u) = 0\} \quad (1.35)$$

Vejam agora que, para uma condição inicial $(x_0, u_0) \in \Gamma$ o sistema (1.32) torna-se

$$\begin{cases} \dot{Y}^{(\rho-1)} = 0 & (1.36a) \\ \dot{\hat{x}} = \eta(\hat{x}, Y^{(\rho-1)}, 0) & (1.36b) \end{cases}$$

que coincide com a solução do sistema implícito (1.34).

O adjetivo semi-implícito provém do fato de que $\forall (x, u) \in \Gamma$, $\psi(x, u) = (\hat{x}, 0)$, uma escolha de \hat{x}_0 , e, por conseguinte de $\psi^{-1}(\hat{x}_0) \in \mathbb{R}^{n-\rho}$, determina as demais ρ coordenadas de x , por isso, podemos entender x como um pseudo-estado e \hat{x} como o verdadeiro estado do sistema [3], ou ainda, como uma subvariedade de dimensão $n - \rho$ dentro da variedade M .

Ao lidarmos com sistemas quadrados, se escolhermos u de modo que a dinâmica do sistema (1.1) evolua somente sobre Γ , abdicamos ao controle das coordenadas $\hat{x}(x)$, que serão regidas exclusivamente pela dinâmica zero $\eta(\hat{x}, Y^{(\rho)})$. Em outras palavras temos que os sistemas quadrados (1.34) são sempre completamente determinados, ou seja, u não é uma entrada livre, mas sim uma pseudo-entrada.

Exemplos ilustrativos e algumas discussões sobre sistemas completamente determinados podem ser encontrados na Introdução deste trabalho.

Capítulo 2: Fundamentos Teóricos para o Método II

Iremos estudar na primeira seção deste capítulo a teoria do Método II, que é uma alternativa para resolução de sistemas semi-implícitos da forma¹

$$\begin{cases} \dot{x}(t) = f[x(t)] + \sum_{i=1}^m g_i[x(t)]u_i(t) & (2.1a) \\ y(t) = h[x(t)] \equiv 0 & (2.1b) \end{cases}$$

Assim como em (1.35), indicaremos por Γ o conjunto de todos $(x, u) \in M = X \times U \subset \mathbb{R}^{n \times m}$ satisfazendo (2.1).

A arte na implementação de algoritmos para esta classe de problemas reside em lidar com erros numéricos. Na prática, podemos não dispor de uma condição inicial $\zeta_0 \doteq (x_0, u_0) \in \Gamma$, ao invés disso, teremos

$$\tilde{\zeta}_0 \in (M \setminus \Gamma) \text{ tal que } \|\tilde{\zeta}_0 - \zeta_0\| < \varepsilon, \varepsilon > 0 \quad (2.2)$$

Ou ainda, mesmo quando conseguirmos uma condição inicial exatamente em Γ , erros decorrentes de precisão finita podem gerar situações como em (2.2). Com o objetivo de alcançar maior robustez, estudaremos o Método II.

Para este método, tentaremos construir um campo vetorial τ sobre a variedade M

$$\frac{d}{dt} \begin{pmatrix} x \\ u \end{pmatrix} = \tau(x(t), u(t)) \quad (2.3)$$

onde as soluções $\zeta(t) = (\tilde{x}(t), \tilde{u}(t))$ do sistema explícito (2.3) converjam para as soluções do sistema semi-implícito (3.1).

Na segunda seção exporemos a teoria de um processo para a obtenção de coordenadas complementares \hat{x} que também seja robusto do ponto de vista numérico.

¹ Veja a descrição das aplicações envolvidas neste sistema em (1.1)

Estrutura do Método II

Corolário 2.1 (do Capítulo 1) Considere os conjuntos $M \doteq (X_0 \times U) \in \mathbb{R}^{n \times m}$ e $N \in \mathbb{R}^{n \times m}$ e aplicação $\Psi : M \rightarrow N$

$$\Psi(x, u) = (\hat{x}, Y^{(\rho)}) \doteq z \quad (2.4)$$

então M e N são localmente difeomórficas por Ψ , com $Y^{(\rho)} \doteq (Y^{(\rho-1)}, y^{(\rho)})$ e $Y^{(\rho-1)}, y^{(\rho)}$ e \hat{x} definidas respectivamente por (2.22), (1.13) e Proposição 1.13.

Prova

Denotaremos $\frac{\partial \Psi(x, u)}{\partial(x, u)}$ por $T(x, u)$, então

$$T(x, u) = \left[\begin{array}{cc} \left. \begin{array}{cc} \frac{\partial \hat{x}}{\partial x} & \frac{\partial \hat{x}}{\partial u} \end{array} \right\}^{n-\rho} \\ \left. \begin{array}{cc} \frac{\partial Y^{(\rho-1)}}{\partial x} & \frac{\partial Y^{(\rho-1)}}{\partial u} \end{array} \right\}^{\rho} \\ \left. \begin{array}{cc} \frac{\partial y^{(\rho)}}{\partial x} & \frac{\partial y^{(\rho)}}{\partial u} \end{array} \right\}^m \end{array} \right] \quad (2.5)$$

Lembrando que $Y^{(\rho-1)}$ são funções independentes quando posto $b(x) = m$, $\forall x \in X_0$ (1.14), então a Proposição 1.13 garante a existência de uma aplicação \hat{x} tal que o posto da submatriz

$\left(\begin{array}{c} \frac{\partial \hat{x}}{\partial x} \\ \frac{\partial Y^{(\rho-1)}}{\partial x} \end{array} \right)$ seja máximo. Além disso, por construção, temos que $\frac{\partial Y^{(\rho-1)}}{\partial u} \equiv 0$ e $\frac{\partial y^{(\rho)}}{\partial u} \neq 0$ de acordo

com (1.9) e, mesmo com a imposição de $\frac{\partial \hat{x}}{\partial u} \equiv 0$, temos

$$T(x, u) = \left(\begin{array}{cc} \left(\begin{array}{c} \frac{\partial \hat{x}}{\partial x} \\ \frac{\partial Y^{(\rho-1)}}{\partial x} \end{array} \right) & 0 \\ \frac{\partial y^{(\rho)}}{\partial x} & \left(\frac{\partial y^{(\rho)}}{\partial u} \right) \end{array} \right) \quad (2.6)$$

que é não singular, porque os elementos em sua diagonal são não singulares em X_0

Utilizando o Teorema da Aplicação Inversa [4], concluímos que M e N são localmente difeomórficas por Ψ .

Da mesma forma que no Método I, escolheremos um parâmetro de convergência $\gamma > 0$ e montaremos um sistema auxiliar, chamado de injeção de saída generalizada, nas coordenadas $(\hat{x}, Y^{(\rho)})$ como segue

$$\begin{cases} \dot{\hat{x}} = \frac{\partial \hat{x}}{\partial x} (f(x) + g(x)u) \\ \dot{y}_1^{(0)} = -\gamma y_1^{(0)} \\ \vdots \\ \dot{y}_1^{(\rho_1)} = -\gamma y_1^{(\rho_1)} \\ \vdots \\ \dot{y}_m^{(0)} = -\gamma y_m^{(0)} \\ \vdots \\ \dot{y}_m^{(\rho_m)} = -\gamma y_m^{(\rho_m)} \end{cases} \quad (2.7')$$

Que será reescrito na forma vetorial como

$$\begin{cases} \dot{\hat{x}} = \frac{\partial \hat{x}}{\partial x} (f(x) + g(x)u) & (2.7a) \\ \dot{Y}^{(\rho)} = -\gamma Y^{(\rho)} & (2.7b) \end{cases}$$

Note que não usufruímos do fato que $\dot{y}_i^{(k)} = y_i^{(k+1)}$, $\forall (i, k) \in (\{1, \dots, m\}, \{0, \dots, \rho_i - 2\})$, decorrente da própria construção de $Y^{(\rho)}$ em (1.9), como feito em (1.32).

Ao invés disso, optamos por um sistema no qual $Y^{(\rho)}(t) = Y^{(\rho)}(t_0)e^{-\gamma(t-t_0)}$ devido a (2.7b), logo $\lim_{t \rightarrow \infty} Y^{(\rho)} = 0$. Assim (2.7) converge assintoticamente para uma solução em Γ .

Em particular, o sistema sempre evoluirá em Γ para qualquer condição inicial $(x_0, u_0) \in \Gamma$.

Voltaremos a discutir este tema na próxima seção.

A idéia agora é recuperar as coordenadas originais (x, u) , preservando as boas características do sistema (2.7) nas coordenadas $(\hat{x}, Y^{(\rho)})$. Seja o sistema

$$\frac{d}{dt} \begin{pmatrix} x \\ u \end{pmatrix} = \tau(x(t), u(t)) \quad (2.8)$$

Onde $\tau(x, u)$ obedece à

$$T(x, u)\tau(x, u) = \begin{pmatrix} \frac{\partial \hat{x}}{\partial x}(f(x) + g(x)u) \\ -\gamma Y^{(\rho)} \end{pmatrix} \quad (2.9)$$

Com esta formulação obteremos um campo vetorial $\tau(x, u)$ cuja representação no espaço $(\hat{x}, Y^{(\rho)})$, dada por $(\Psi_*\tau)(z)$, coincide com o sistema (2.8), i.e.

$$\langle d\hat{x}, \tau \rangle = \frac{\partial \hat{x}}{\partial x}(f(x) + g(x)u) \quad (2.10a)$$

$$\langle dY^{(\rho)}, \tau \rangle = -\gamma Y^{(\rho)} \quad (2.10b)$$

Chama-se a atenção do leitor que a expressão de $T(x, u)$ pode ser obtida algebricamente e que o sistema (2.7), no qual (2.7b) é linear, será resolvido por métodos numéricos. Um passo-a-passo para este método é assunto do Capítulo 3.

Teoremas de Convergência para o Método II

Apresentam-se três teoremas a fim de formalizar os conceitos abordados até aqui.

Teorema 2.2 Seja $\zeta(t) = (\bar{x}(t), \bar{u}(t))$ uma solução do sistema (3.6) com $\zeta(t_0) \in \Gamma$. Então $\bar{x}(t) = \pi_x(\zeta(t))$ é uma solução de (3.1) com entrada $u(t)$. Reciprocamente, se $\bar{x}(t) \in C^\infty$ é uma solução de (3.1), então existe uma $\bar{u}(t)$ tal que $\zeta(t) = (\bar{x}(t), \bar{u}(t))$ é uma solução de (3.6) para uma condição inicial $(\pi_x(\zeta(t_0)), \bar{u}(t_0)) \in \Gamma$.

Teorema 2.3 Seja $\zeta(t) = (\bar{x}(t), \bar{u}(t))$ uma solução de (3.6) com condição inicial ζ_0 . Se $\zeta(t)$ é bem definida para todo $t \in [t_0, t_1]$, então $\|Y^{(\rho)}(t)\| \leq e^{-\gamma t} \|Y^{(\rho)}(t_0)\|$ para todo $t \in [t_0, t_1]$.

Teorema 2.4 Seja $L \subset M = \mathbb{R}^n \times \mathbb{R}^m$ um conjunto compacto. Sejam $\varepsilon > 0$ e $L_1 = \{\mu \in M \mid \text{dist}(\mu, L) < \varepsilon\}$. Assumiremos que toda solução $\zeta(t)$ de (3.6) com condição inicial $\zeta(t_0) \in L_1$ é bem definida e está dentro de um conjunto compacto $V \in M$ para todo $t \in [t_0, t_1]$. Então existe $\delta > 0$ tal que, se $\zeta(t)$ é uma solução de (18) com condição inicial em L e $\|Y^{(\rho)}(t_0)\| < \delta$, então existem $\kappa_1, \kappa_2 > 0$ e uma solução $x(t)$ de (3.1) onde, para todo $t \in [t_0, t_1]$, teremos que $\|\pi_x(\zeta(t)) - x(t)\| \leq \kappa_1 \|Y^{(\rho)}(t_0)\| e^{\kappa_2(t-t_0)}$

Prova: Vide [5]

Obtendo Coordenadas Complementares \hat{x}

Tomando por base a Proposição 1.13, sabemos que para todo conjunto φ de $\rho+m$ funções independentes, existe um conjunto \hat{x} de $n-\rho$ funções que completam a transformação de coordenadas $\Psi(x,u) = (\varphi(x), \hat{x}(x))$. Todavia, conseguir a expressão analítica de \hat{x} pode ser uma tarefa árdua.

Como artifício, apresentaremos abaixo uma asserção que nos permitirá operar com uma escolha local de uma matriz constante que representará \hat{x} , mesmo sem obtê-la diretamente.

Proposição 3.5 O campo τ definido por (3.9) tem as seguintes propriedades que não dependem da escolha de $\hat{x}(x)$

1. $\langle dY^{(\rho)}, \tau \rangle = -\gamma Y^{(\rho)}$
2. Para um ponto (\bar{x}, \bar{u}) de Γ , $\tau(\bar{x}, \bar{u})$ não depende da escolha de $\hat{x}(x)$
3. Para um ponto (\bar{x}, \bar{u}) de Γ , $\dot{x} = \langle dx, \tau(\bar{x}, \bar{u}) \rangle = f(\bar{x}) + g(\bar{x})\bar{u}$

Prova Consultar [3]

Agora, nosso problema converteu-se em descobrir uma matriz R tal que o posto de $\begin{pmatrix} R \\ \frac{\partial Y^{(\rho-1)}}{\partial x} \end{pmatrix}$ em (\bar{x}, \bar{u}) seja máximo, de modo a conseguirmos uma $T(\bar{x}, \bar{u})$ invertível, i.e., um difeomorfismo local Ψ ao redor de (\bar{x}, \bar{u}) .

Qualquer matriz $R_{(n-\rho) \times n}$ de linhas linearmente independentes de $H_{\rho \times n} \doteq \frac{\partial Y^{(\rho-1)}}{\partial x}$ seria uma escolha plausível, entretanto, gostaríamos de confeccionar R menos suscetível a erros numéricos.

Um recurso de álgebra computacional que garante maior robustez são as matrizes ortogonais. Lançaremos mão então de um resultado simples da álgebra linear sobre complementos ortogonais.

Lema 3.6 Seja $V = \{v_1, \dots, v_p\}$ um conjunto de vetores linearmente independentes no \mathbb{R}^n e o seu complemento ortogonal $V^\perp \doteq \{x \in \mathbb{R}^n / \langle x, v \rangle = 0, \forall v \in V\}$. Então $\mathbb{R}^n = V \oplus V^\perp$ [6]

Um processo fácil para extração de um complemento ortogonal V^\perp é a decomposição QR [6]. Dada uma matriz de posto máximo $S_{n \times s}$, obtêm-se uma matriz ortogonal $q_{n \times n}$ e uma matriz triangular superior de diagonal positiva $r_{n \times s}$ tais que $qr = S$. Subdividiremos a matriz q como segue

$$n \left\{ \begin{array}{c} \left[\begin{array}{cc} q_1 & q_2 \end{array} \right] \\ \underbrace{\hspace{1.5cm}}_s \quad \underbrace{\hspace{1.5cm}}_{n-s} \end{array} \right\} = \left[\begin{array}{c} q \end{array} \right] \quad (2.11)$$

Similarmente ao processo de Gramm-Schmidt, a decomposição QR gera em q_1 a ortogonalização dos vetores coluna de S e em q_2 seu complemento ortogonal.

Fazendo $S = H^T$ e computando as matrizes q_1 , q_2 e r pelo processo acima, $R = q_2^T$ implicará em uma matriz

$$T(\bar{x}, \bar{u}) = \left(\begin{array}{cc} \left(\begin{array}{c} q_2^T \\ \frac{\partial Y^{(\rho-1)}}{\partial x} \end{array} \right) & 0 \\ \frac{\partial y^\rho}{\partial x} & \left(\frac{\partial y^\rho}{\partial u} \right) \end{array} \right) \quad (2.12)$$

localmente invertível porque os elementos em sua diagonal possuem determinante não nulo, pelo Lema 3.6.

Capítulo 3: Algoritmos e Construção de Simulação

Apresentam-se aqui os algoritmos referentes aos Métodos I e II para a solução de problemas semi-implícitos não lineares. Como introdução a cada um dos algoritmos, foi inserida uma brevíssima discussão sobre as ferramentas envolvidas e menções aos tópicos relacionados da teoria presentes nos Capítulos 1 e 2. O código fonte destes algoritmos está disponível no Apêndice. Manipularemos DAEs da forma

$$\left\{ \begin{array}{l} \dot{x}(t) = f[x(t), u(t)] \end{array} \right. \quad (3.1a)$$

$$\left\{ \begin{array}{l} y(t) = h[x(t)] = 0 \end{array} \right. \quad (3.1b)$$

de acordo com (1.1), substituindo $f[x(t)] + \sum_{i=1}^m g_i[x(t)]u_i(t)$ por $f[x(t), u(t)]$, ou seja, lidaremos com sistema não necessariamente afins, discutindo as implicações desta generalização na Seção Construção da Implementação do Método I.

Ressalta-se novamente que nossos sistemas serão quadrados logo, completamente determinados.

Obtendo a Aplicação $Y^{(\rho)}$

Um trabalho a ser feito para ambos os métodos é a obtenção de $Y^{(\rho)} \doteq (Y^{(p-1)}, y^{(\rho)})$, (1.22) e (1.13), utilizando cálculo simbólico.

Esta aplicação, cujo posto é pleno, será construída de acordo com (1.9), para que consigamos posteriormente um sistema fortemente desacoplado pelo Teorema 1.3. Visto que $\dim u = \dim y$, derivaremos sucessivamente as $l = m$ saídas do sistema enquanto elas não dependam explicitamente da entrada u .

Observe que denotamos $\tilde{\rho} \doteq \{\rho_1, \dots, \rho_m\}$

Algoritmo 3.1

Função TransformaCoordenadas(f, h, x, u), Devolve $(Y^{(\rho-1)}, y^{(\rho)}, \tilde{\rho})$

$Y^{(\rho-1)} \leftarrow \{ \}, y^{(\rho)} \leftarrow \{ \}, \tilde{\rho} \leftarrow \{ \}$

Para cada $i \in \{1, \dots, m\}$, Faça:

$k \leftarrow 0$

$\bar{h} \leftarrow h_i$

Enquanto $\frac{\partial \bar{h}}{\partial u} \equiv 0$

$Y^{(\rho-1)} \leftarrow Y^{(\rho-1)} \cup \{\bar{h}\}$

$\bar{h} \leftarrow \left\langle \frac{\partial \bar{h}}{\partial x}, f \right\rangle$

$k \leftarrow k + 1$

Fim do Enquanto

$\tilde{\rho} \leftarrow \tilde{\rho} \cup \{k\}$,

$y^{(\rho)} \leftarrow y^{(\rho)} \cup \{\bar{h}\}$

Fim do Para cada

Lembrando que se $k > 1$, na i -ésima entrada e k -ésima iteração, $\bar{h} = h_i^{(k)} = L_f^k h_i^{(k-1)}$ pois

$$\dot{y}_i^{(k-2)} = \frac{d}{dt} h_i^{(k-2)} = \frac{\partial h_i^{(k-2)}}{\partial(x, u)} \begin{pmatrix} \dot{x} \\ \dot{u} \end{pmatrix} = \frac{\partial h_i^{(k-2)}}{\partial x} f = h_i^{(k-1)} \quad (3.2)$$

De (3.1b) temos que as restrições $h(x)$ não dependem explicitamente de u , então $\bar{h} = h_i^{(k)} = L_f^k h_i^{(k-1)}$ é válida para $k = 1$.

Definição 3.2 É denominado índice de uma DAE o inteiro $k^* = \max\{\tilde{\rho} + 1\}$. Esta é uma maneira de se medir a dificuldade de integração de um sistema (3.1), visto que k^* expressa o número mínimo de derivações necessárias para chegarmos a $Y^{(\rho)}(x)$.

Obtendo Coordenadas Complementares \hat{x}

Outra tarefa comum a ambos os métodos é a obtenção de \hat{x} para compor um difeomorfismo $\psi(x) = (Y^{(\rho-1)}, \hat{x})$ (1.28). Sempre que $Y^{(\rho-1)}$ formar um conjunto linearmente, o Teorema 1.13 garante a existência de \hat{x} . Utilizaremos aqui a técnica do complemento ortogonal, inspirada no Lema 3.6 e na decomposição QR, para uma escolha local de \hat{x} (Lema 3.5).

Algoritmo 3.3

Função CoordenadasComplementares(H), Devolve (R)

$$(\rho, n) \leftarrow \dim(R)$$

$$(q, r) \leftarrow \text{DecomposiçãoQR}(H)$$

$$R \leftarrow \begin{pmatrix} q_{1,\rho+1} & \cdots & q_{1,n} \\ \vdots & & \vdots \\ q_{n,\rho+1} & \cdots & q_{n,n} \end{pmatrix}^T$$

Construindo Implementação Método I

Descreveremos a rotina que fornecesse os elementos para o diagrama de simulação do Método I. Aproveitaremos as relações $\dot{y}_i^{(k)} = y_i^{(k+1)} \quad \forall (i, k) \in (\{1, \dots, m\}, \{0, \dots, \rho_i - 2\})$ de $Y^{(\rho)}$, inerentes a própria construção (1.9), chegando a um sistema (1.31) nas coordenadas $\psi(x) \doteq (Y^{(\rho-1)}, \hat{x}) = z$, fortemente desacoplado pelo Teorema 1.6.

$$\left\{ \begin{array}{l} \dot{y}_i^{(k)} = y_i^{(k+1)} \\ \dot{y}_i^{(\rho_i-1)} = a_i + b_i u \end{array} \right\} \forall (i, k) \in (\{1, \dots, m\}, \{0, \dots, \rho_i - 2\}) \quad (3.3)$$

$$\dot{\hat{x}} = \eta(\hat{x}, Y^{(\rho-1)}, u)$$

Neste método, dependemos da separação de $y^{(\rho)}$ em $a(x) + b(x)u$, conforme (1.13), para que possamos definir uma realimentação de estado estática via (1.16)

$$u = \alpha(x) + \beta(x)v, \text{ onde } \alpha(x) = -b^{-1}(x)a(x), \beta(x) = b^{-1}(x)$$

Verifica-se facilmente que esta separação é possível somente para sistemas afins, portanto, ao contrário do Método II, este método é aplicável somente para esta classe de problemas.

A nova entrada v será parametrizada por γ , como em (1.33).

$$v_i = -\sum_{j=0}^{\rho_i-1} \alpha_{ij} y_i^{(j)}, \quad \forall i = \{1, \dots, m\}, \text{ onde } \pi_i(s) = (s + \gamma)^{\rho_i} = s^{\rho_i} + \sum_{j=0}^{\rho_i-1} \alpha_{ij} s^j \quad (3.4)$$

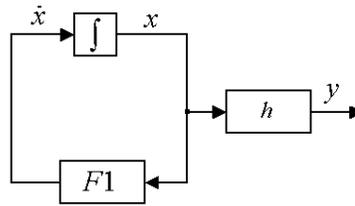
No algoritmo a seguir será utilizada uma função $\text{ConstroiFunção}(\text{Exp}_{\text{Simb}}, \text{Var}_{\text{Sim}})$ Retorna(Função_{Numérica}). Esta Função_{Numérica} será definida pela substituição das variáveis simbólicas Var_{Sim} por valores numéricos em Exp_{Simb} .

Algoritmo 3.4

Função ConstróiImplementaçãoI(γ, f, h, x, u), Devolve (Implementação)

ConstroiFunção($\gamma, \{ \}$)
 ConstroiFunção($f, \{x, u\}$)
 ConstroiFunção($h, \{x\}$)
 $(Y^{(\rho-1)}, y^{(\rho)}, \tilde{\rho}) \leftarrow \text{TransformaCoordenadas}(f, h, x, u)$
 ConstroiFunção($\frac{\partial Y^{(\rho-1)}}{\partial x}, \{x\}$)
 $a \leftarrow y^{(\rho)}|_{u=0}, \quad b \leftarrow (y^{(\rho)} - a)|_{u=1}$
 $\alpha \leftarrow -b^{-1}a, \quad \beta \leftarrow -b^{-1}$
 $v_i \leftarrow -\sum_{k=0}^{\rho_i} \text{CoeficientesPolinômio}(j, \rho_i, \gamma)y_i^{(k)} \quad \forall i \in \rho$
 $u_{\text{realimentação}} \leftarrow \alpha + \beta v$
 ConstroiFunção($u_{\text{realimentação}}, x$)
 Para cada $i \in \{1, \dots, m\}$
 $\dot{y}_i^{(k)} \leftarrow y_i^{(k+1)} \quad \forall k \in \{0, \dots, \rho_i - 2\}$
 $\dot{y}_i^{(\rho_i-1)} \leftarrow a_i + b_i u_i$
 Fim do Para cada
 ConstroiFunção($\dot{Y}^{(\rho-1)}, \{x\}$)

Diagrama 3.5



Retornaremos ao espaço tangente das coordenadas originais do sistema calculando

$$\dot{x} = T^{-1}\dot{z}, \text{ onde } T(x) \doteq \frac{\partial \psi(x)}{\partial x}.$$

O leitor deve notar que as variáveis numéricas no próximo algoritmo estão assinaladas com uma barra e que a função $F1$ do diagrama não precisa ser reconstruída, já ela é parametrizada pelas saídas da função ConstróiImplementaçãoI.

Algoritmo 3.6

Função $F1(\bar{x})$, Devolve $(\dot{\bar{x}})$

$$\bar{u} = u_{\text{realimentação}} \Big|_{\bar{x}}$$

$$R \leftarrow \text{CoordenadasComplementares} \left(\left(\frac{\partial Y^{(\rho-1)}}{\partial x} \Big|_{\bar{x}} \right)^T \right)$$

$$\bar{T} \leftarrow \left(\begin{array}{c} R \\ \frac{\partial Y^{(\rho-1)}}{\partial x} \end{array} \right) \Big|_{\bar{x}}$$

$$\bar{\tau} \leftarrow \left(\begin{array}{c} Rf \\ \dot{Y}^{(\rho-1)} \end{array} \right) \Big|_{\left(\begin{array}{c} \bar{x} \\ \bar{u} \end{array} \right)}$$

$$\dot{\bar{x}} \leftarrow \bar{T}^{-1} \bar{\tau}$$

Construindo Implementação Método II

Exibimos a seguir a rotina que criará as entradas do esquema de simulação para o Método II.

Já que não podemos escolher livremente as entradas u para sistemas semi-implícitos (Capítulo 1, Seção Sistemas Semi-Implícitos) optaremos por um difeomorfismo sobre o espaço dos estados e das entradas $\Psi(x, u) \doteq (\hat{x}, Y^{(\rho)}) = z$ conforme Corolário 2.1.

Outra diferença em relação ao primeiro método é que não incluiremos agora as relações derivativas entre as funções em $Y^{(\rho)}$ no sistema nas coordenadas $(\hat{x}, Y^{(\rho)})$.

Nosso sistema nas novas coordenadas será uma injeção de saída generalizada com parâmetro de velocidade de convergência γ , como em (2.7)

$$\begin{cases} \dot{\hat{x}} = \frac{\partial \hat{x}}{\partial x} (f(x) + g(x)u) \\ \dot{Y}^{(\rho)} = -\gamma Y^{(\rho)} \end{cases} \quad (3.5)$$

Algoritmo 3.7

Função ConstróiImplementaçãoII(γ, f, h, x, u), Devolve (Implementação)

ConstoiFunção($\gamma, \{ \}$)

ConstoiFunção($f, \{x, u\}$)

ConstoiFunção($h, \{x\}$)

$(Y^{(\rho-1)}, y^{(\rho)}, \tilde{\rho}) \leftarrow \text{TransformaCoordenadas}(f, h, x, u)$

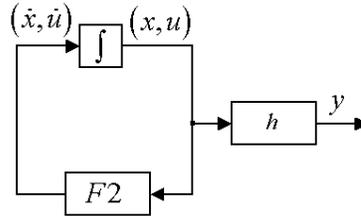
ConstoiFunção($Y^{(\rho-1)}, \{x\}$)

ConstoiFunção($\frac{\partial Y^{(\rho-1)}}{\partial x}, \{x\}$)

ConstoiFunção($y^{(\rho)}, \{x, u\}$)

ConstoiFunção($\frac{\partial y^{(\rho)}}{\partial(x, u)}, \{x, u\}$)

Diagrama 3.8



Podemos interpretar $F2$ como um campo vetorial τ no espaço (x, u) (2.6-7), tal que $Y^{(\rho)}(t) = Y^{(\rho)}(t_0)e^{-\gamma(t-t_0)}$, ou seja, que as soluções $\zeta(t) = (\bar{x}(t), \bar{u}(t))$ do sistema simulado converjam assintoticamente para uma solução na variedade Γ . Veja os Teoremas 2.2-4 para uma formalização destes conceitos.

Novamente retornaremos ao espaço tangente original calculando $\dot{x} = T^{-1}\dot{z}$

Por construção, temos que $T(x) \doteq \frac{\partial \Psi(x, u)}{\partial(x, u)}$ é da forma (Corolário 2.1)

$$T(x, u) = \begin{pmatrix} \begin{pmatrix} \frac{\partial \bar{x}}{\partial x} \\ \frac{\partial Y^{(\rho-1)}}{\partial x} \end{pmatrix} & 0 \\ \frac{\partial y^{(\rho)}}{\partial x} & \begin{pmatrix} \frac{\partial y^{(\rho)}}{\partial u} \end{pmatrix} \end{pmatrix} \quad (3.6)$$

Como antes, as variáveis numéricas foram assinaladas com uma barra e $F2$ é recebe parâmetros de ConstróiImplementaçãoII.

Algoritmo 3.9

$$\begin{aligned} & \text{Função } F2\left(\begin{matrix} \bar{x} \\ \bar{u} \end{matrix}\right), \text{ Devolve } \begin{pmatrix} \dot{\bar{x}} \\ \dot{\bar{u}} \end{pmatrix} \\ & R \leftarrow \text{CoordenadasComplementares} \left(\left(\left. \frac{\partial Y^{(\rho-1)}}{\partial x} \right|_{\bar{x}} \right)^T \right) \\ & \bar{T} \leftarrow \left(\begin{array}{cc} R & 0 \\ \frac{\partial Y^{(\rho-1)}}{\partial x} & 0 \\ \frac{\partial y^{(\rho)}}{\partial x} & \frac{\partial y^{(\rho)}}{\partial x} \end{array} \right)_{\left(\begin{matrix} \bar{x} \\ \bar{u} \end{matrix} \right)} \\ & \bar{\tau} \leftarrow \left(\begin{array}{c} Rf \\ -\gamma Y^{(\rho)} \end{array} \right)_{\left(\begin{matrix} \bar{x} \\ \bar{u} \end{matrix} \right)} \\ & \begin{pmatrix} \dot{\bar{x}} \\ \dot{\bar{u}} \end{pmatrix} \leftarrow \bar{T}^{-1} \bar{\tau} \end{aligned}$$

Note que a última passagem dos Algoritmos 4.5-6 será calculada através de Eliminação de Gauss, sem necessidade de uma inversão numérica de T_1 .

Capítulo 4: Experimentos Numéricos

Realizaremos aqui um estudo empírico dos métodos através de simulações. As técnicas de aferição utilizadas serão: (i) comparação das soluções obtidas através dos métodos com a solução analítica, (ii) análise do erro numérico de y em relação à restrição $y = h(x) = 0$ e (iii) comparação entre as soluções obtidas pelo Método I e II.

Trataremos de algumas variantes de dois exemplos, a saber, matrizes mal condicionadas [1] e o pêndulo simples [7].

Exemplo 1: Matrizes Mal Condicionadas

Seja o sistema linear

$$\dot{w}(t) = Aw(t) + Bu(t) \quad (4.7a)$$

$$y(t) = Cw(t) = 0 \quad (4.7b)$$

Onde

$$A = T_1^{-1} \tilde{A} T_1, \quad B = T_1^{-1} \tilde{B}, \quad C = \tilde{C} T_1^2 \quad (4.8c)$$

Com $(\tilde{A}, \tilde{B}, \tilde{C})$ oriundos do sistema linear original

$$\dot{x}(t) = \tilde{A}x(t) + \tilde{B}u(t) \quad (4.9a)$$

$$y(t) = \tilde{C}x(t) = 0 \quad (4.9b)$$

onde

$$\tilde{A} = \begin{pmatrix} 0 & -100 & -100 & -100 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad \tilde{C} = (0 \quad 1 \quad 0 \quad 0) \quad (4.9c)$$

Observe que (4.6) é igual a

$$\begin{cases} \dot{x}_1 = -100x_1 - 100x_2 - 100x_3 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = u \\ y = x_2 = 0 \end{cases} \quad (4.7')$$

² T_1 foi invertida simbolicamente nestas simulações

de onde se extrai a expressão analítica da solução de (4.6)

$$\begin{aligned} w(t) &\equiv w_0, \quad \forall (w_0, 0) \in \Gamma \\ \Gamma &= \left\{ (w, u) \in \mathbb{R}^4 \times \mathbb{R}^1 \text{ tal que } w = T_1^{-1}(c, 0, 0, 0)^T, c \in \mathbb{R} \text{ e } u = 0 \right\} \end{aligned} \quad (4.10)$$

Começaremos pelo caso mais simples, tomando $T_1 = I$

$$\text{Veja que } \begin{cases} y = w_2 \doteq y^{(0)} \\ \dot{y}^{(0)} = w_3 \doteq y^{(1)} \\ \dot{y}^{(1)} = w_4 \doteq y^{(2)} \\ \dot{y}^{(2)} = u \doteq y^{(3)} \end{cases}, \text{ então } \rho = 3, Y^{(\rho-1)} = (w_2, w_3, w_4) \text{ e } y^{(\rho)} = u \text{ (Algoritmo 3.1).}$$

A escolha $\hat{x} = w_1$ completaria $Y^{(\rho-1)}$ para uma base de \mathbb{R}^4 .

O sistema para o Método I (vide 4.3 e 4.4) é

$$\begin{cases} \dot{y}^{(0)} = y^{(1)} \\ \dot{y}^{(1)} = y^{(2)} \\ \dot{y}^{(2)} = u \\ \dot{\hat{x}} = \frac{\partial \hat{x}}{\partial x} (Aw(t) + Bu(t)) \end{cases} = \begin{cases} \dot{y}^{(0)} = y^{(1)} \\ \dot{y}^{(1)} = y^{(2)} \\ \dot{y}^{(2)} = -\gamma^3 y^{(0)} - 3\gamma^2 y^{(1)} - 3\gamma y^{(2)} \\ \dot{\hat{x}} = -100y^{(0)} - 100y^{(1)} - 100y^{(2)} \end{cases}$$

O sistema para o Método II (vide 4.5) é

$$\begin{cases} \dot{y}^{(0)} = -\gamma y^{(0)} \\ \dot{y}^{(1)} = -\gamma y^{(1)} \\ \dot{y}^{(2)} = -\gamma y^{(2)} \\ \dot{\hat{x}} = \frac{\partial \hat{x}}{\partial x} (Aw(t) + Bu(t)) \end{cases} = \begin{cases} \dot{y}^{(0)} = -\gamma y^{(0)} \\ \dot{y}^{(1)} = -\gamma y^{(1)} \\ \dot{y}^{(2)} = -\gamma y^{(2)} \\ \dot{\hat{x}} = -100y^{(0)} - 100y^{(1)} - 100y^{(2)} \end{cases}$$

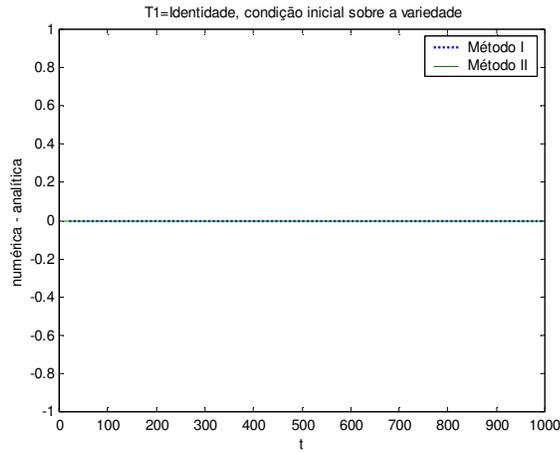
Salvo a utilização de uma escolha local de $\frac{\partial \hat{x}}{\partial x}$ como descrito no Algoritmo 4.3, estas serão nossas simulações.

Seja uma condição inicial $(w_0, u_0) \in \Gamma$ com $w_0 = T_1^{-1}x_0$, $x_0 = (2 \ 0 \ 0 \ 0)^T$ e $u_0 = 0$.

Consideraremos $\gamma = 1$ como padrão. Utilizaremos o método de integração numérica Dormand-Prince (ode45 MatLab), com tamanho de passo variável e tolerância relativa RTOL=1e-4.

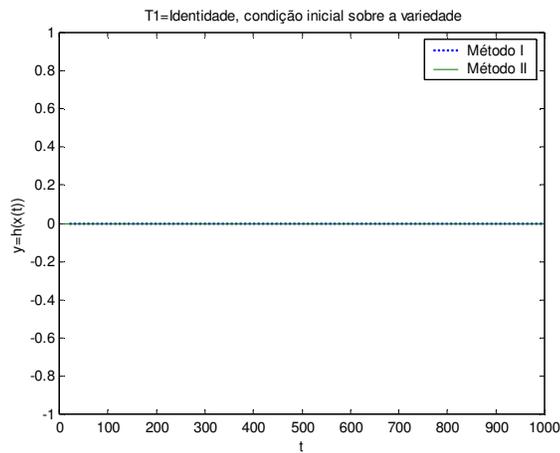
Observe que o eixo vertical na próxima figura identificado por “numérica - analítica” significa $\max_{k \in \{1, \dots, 4\}} (w_k(t) - w_{0k})$ para cada t . Veja expressão analítica da solução (4.10).

Figura 4.7



Para este experimento trivial os estados w gerados numericamente por ambos os métodos acompanharam exatamente a solução analítica.

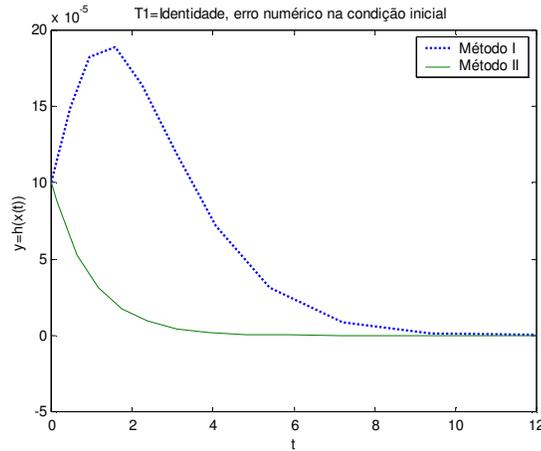
Figura 4.8



Veja que o desvio em relação à variedade foi nulo para ambos os métodos, o que reforça a simulação anterior.

Utilizaremos agora uma condição inicial com um erro numérico em relação à variedade, $(\tilde{w}_0, \tilde{u}_0) \notin \Gamma$ com $\tilde{w}_0 = T_1^{-1} \tilde{x}_0$, $\tilde{x}_0 = \begin{pmatrix} 2 & 10^{-4} & 10^{-4} & 10^{-4} \end{pmatrix}$ e $\tilde{u}_0 = 10^{-4}$.

Figura 4.9



Observamos que ambos os métodos foram capazes de acentuar o erro da condição inicial nos primeiros 10s de simulação. O Método I apresentou um pico de quase de duas vezes o erro original. Lembrando que no neste método $\dot{y}_i^{(k)} = y_i^{(k+1)}$ $\forall (i,k) \in (\{1, \dots, m\}, \{0, \dots, \rho_i - 2\})$ e que a condição inicial está fora da variedade, i.e $Y^{(\rho)} \neq 0$, $y_i^{(k+1)}$ se afastará ainda mais de zero enquanto $\dot{y}_i^{(k)} > 0$. Contrariamente, o Método II iniciou instantaneamente a se aproximar da variedade, sem aumento de erro inicial, e ainda mais rápido do que o Método I para $\gamma = 1$.

Note também que a tolerância relativa utilizada poderia ser refinada para suavizar o bico observado no Método I por volta de 2s na Figura 4.9.

Vamos analisar o resultado da mesma simulação anterior para $t > 10$.

Figura 4.10a

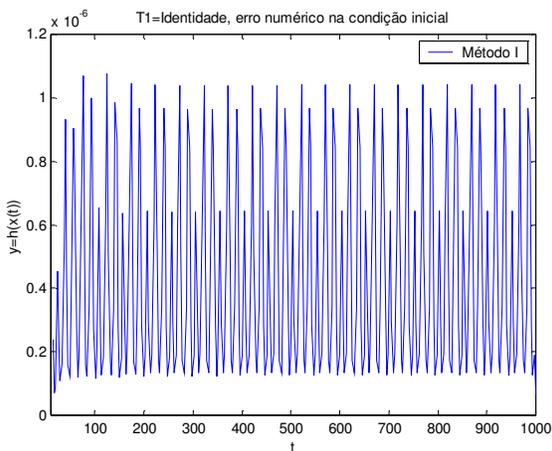
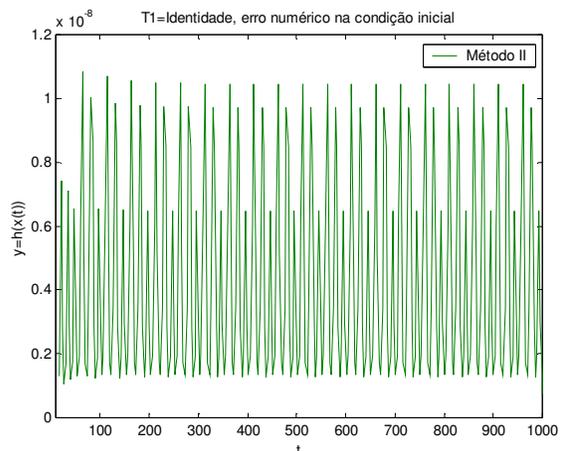


Figura 4.10b



Após o transitório, o Método II apresentou ruído da ordem de 10^{-8} e Método I de 10^{-6} , 100 vezes maior. Notamos também um erro médio diferente de zero para ambos os métodos.

Analogamente ao experimento da Figura 4.7, veremos aqui o comportamento das soluções numéricas em relação à solução analítica que, apesar do erro numérico na condição inicial, consideramos igual a $w(t) \equiv w_0$.

Figura 4.11a

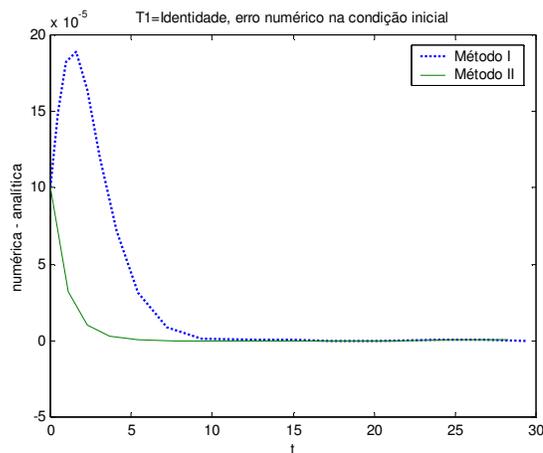
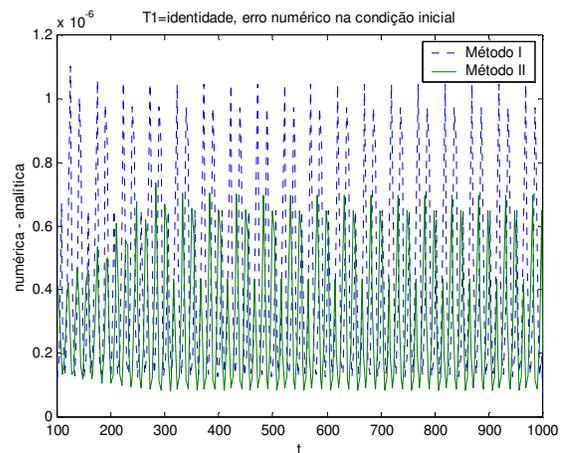


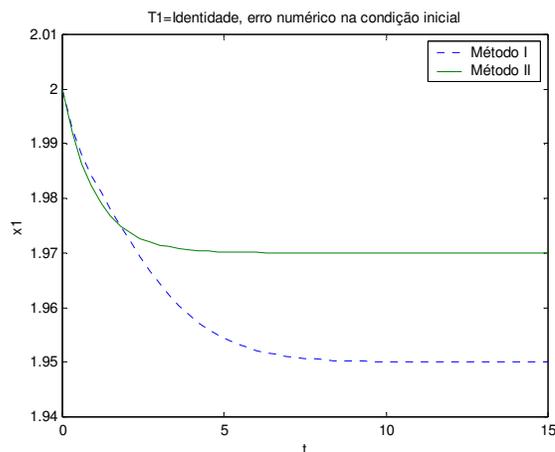
Figura 4.11b



A Figura 4.11a confirma os resultados exibidos na Figura 4.9, mostrando que, além do desvio $h(x)$ relativo à variedade, o estado do sistema também foi corrigido assintoticamente.

Ao contrário da Figura 4.7, temos que os dois métodos na Figura 4.11b exibiram erro médio diferente de zero em relação à solução analítica. A próxima figura auxiliará a entender este fenômeno.

Figura 4.12



De fato, ambos os métodos levaram o sistema a convergir para uma solução na variedade, ainda que nesta solução $w(t_0) \neq w_0$, devido a presença do erro numérico na condição inicial. O estado w_1 convergiu para aproximadamente 1.95 no Método I e no Método II para 1.97.

Deseja-se estudar o comportamento destes métodos para matrizes mal condicionadas e para isso escolheremos

$$T_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 5 & 6 & 7 \\ 7 & 8 & 9.00000001 & 10 \\ 11 & 12 & 13 & 14.00000001 \end{pmatrix}$$

Considere $(w_0, u_0) \in \Gamma$ definida anteriormente.

Figura 4.13a

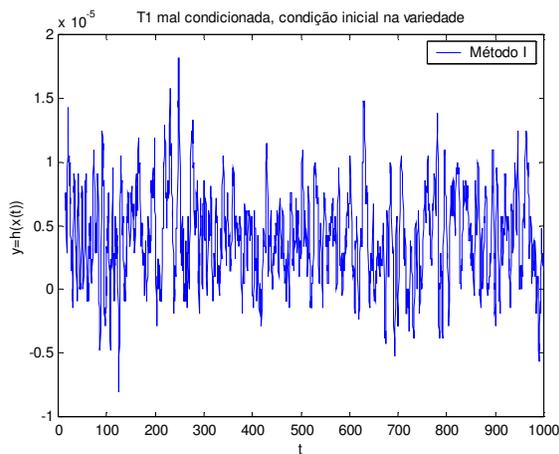
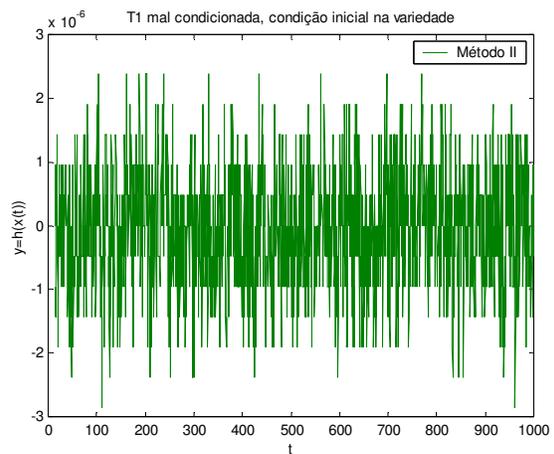


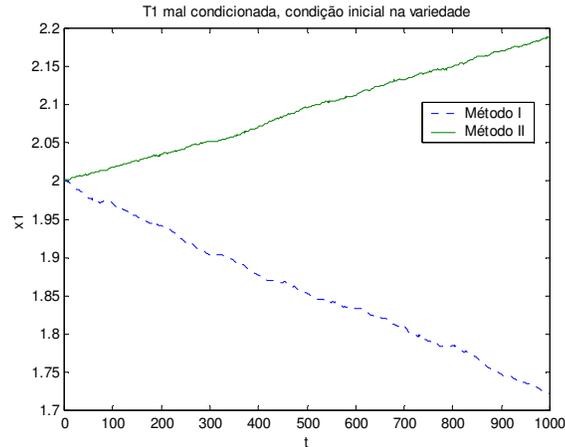
Figura 4.13b



O experimento numérico ilustrado pelas figuras 4.13a-b mostra que mesmo para uma T_1 mal condicionada, os métodos acompanham aproximadamente a restrição. Desta vez temos que o Método I apresenta ruído em y da ordem de 10^{-5} e Método II de 10^{-6} .

Comparando com as Figuras 4.10a-b, constatamos que os dois métodos sofreram perda de acuidade com a utilização de uma T_1 mal condicionada, entretanto, o Método II continua obtendo melhor resultado (aproximadamente 10 vezes mais preciso que o Método I).

Figura 4.14



Apesar de a saída y na simulação da Figura 4.13 seguir aproximadamente a restrição, o estado x_1 , correspondente à dinâmica zero, distanciou-se da solução analítica atingindo uma diferença da ordem de 10^{-1} em $t = 1000$ nos dois métodos. Este experimento foi refeito com refinamento da tolerância relativa do passo de integração até 10^{-7} , sem modificação qualitativa no comportamento de x_1 . Concluímos então que a análise da saída y em relação à restrição não é suficiente para atestar a qualidade dos métodos.

Tentaremos então outro ensaio, diminuindo gradativamente a estabilidade numérica das matrizes. Seja a família de matrizes (veja que o exemplo anterior equivale à T_{18})

$$T_{1k} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 5 & 6 & 7 \\ 7 & 8 & 9+10^{-k} & 10 \\ 11 & 12 & 13 & 14+10^{-k} \end{pmatrix}$$

Estudaremos o comportamento de x_1 para diferentes valores de k .

Tabela 4.15

k	$\text{rcond}(T_{1k})$	$\max_{t \in [0,1000]} x_1 - 2 , (w_0, u_0) \in \Gamma$	
		Método I	Método II
0	1,44E-03	2,16E-04	2,22E-04
1	1,39E-04	2,22E-04	2,22E-04
2	1,38E-05	2,39E-04	2,23E-04
3	1,38E-06	2,20E-04	2,18E-04
4	1,38E-07	2,63E-04	2,86E-04
5	1,38E-08	2,72E-04	2,42E-04
6	1,38E-09	5,46E-03	3,18E-03
7	1,38E-10	3,62E-02	2,86E-02
8	1,38E-11	2,89E-01	1,75E-01

Utilizamos o número de condicionamento rcond como medida de estabilidade de matrizes. Se uma matriz T é bem condicionada $\text{rcond}(T)$ será próximo de 1, caso contrário, $\text{rcond}(T)$ se aproximará de 0.

Notamos que até $k = 5$ o erro de x_1 foi da ordem de 10^{-4} . A partir daí, o erro começa a crescer acompanhando o crescimento de rcond .

Ademais, para um k fixo e uma condição inicial na variedade, este experimento não detectou diferença de acuidade maior do que $3,00\text{E-}05$ entre os métodos, ao analisar o estado x_1 numérico versus analítico.

Exibem-se abaixo os resultados de testes comparando a saída y com a restrição. Observe que optamos por $k \in \{1, \dots, 5\}$ para que o erros decorrentes de x_1 não prejudicassem o experimento.

Tabela 4.16

k	Método I		
	$(\tilde{w}_0, \tilde{u}_0) \notin \Gamma$		$(w_0, u_0) \in \Gamma$
	$\text{ET1} \doteq \max_{t \in [0,15]} y $	$\text{EE1} \doteq \max_{t \in [15,1000]} y $	$\text{E1} \doteq \max_{t \in [0,1000]} y $
0	1,88E-04	2,27E-06	2,16E-06
1	1,88E-04	2,21E-06	2,22E-06
2	1,88E-04	2,62E-06	2,39E-06
3	1,88E-04	2,26E-06	2,19E-06
4	1,88E-04	2,37E-06	2,49E-06
5	1,88E-04	2,54E-06	2,22E-06

Tabela 4.17

k	Método II		
	$(\tilde{w}_0, \tilde{u}_0) \notin \Gamma$		$(w_0, u_0) \in \Gamma$
	$\text{ET2} \doteq \max_{t \in [0,15]} y $	$\text{EE2} \doteq \max_{t \in [15,1000]} y $	$\text{E2} \doteq \max_{t \in [0,1000]} y $
0	1,00E-04	7,38E-07	1,21E-07
1	1,00E-04	7,39E-07	1,74E-06
2	1,00E-04	7,37E-07	9,04E-07
3	1,00E-04	7,36E-07	1,92E-06
4	1,00E-04	7,04E-07	9,14E-06
5	1,00E-04	1,25E-06	1,27E-05

Os erros de transitório para condição inicial com erro numérico, ET1 e ET2, mantiveram-se constantes, independente dos valores de k escolhidos. Estes erros foram equivalentes aos ilustrados na Figuras 4.9 e Figura 4.11a.

Os erros em regime estacionário para $(\tilde{w}_0, \tilde{u}_0) \notin \Gamma$, EE1 e EE2, diferiram no máximo em $1,14E-05$ dos erros de saída para condição inicial na variedade. Isto significa que ambos os métodos conseguiram corrigir o erro inerente a $(\tilde{w}_0, \tilde{u}_0) \notin \Gamma$, levando o sistema aproximadamente para a solução gerada por $(w_0, u_0) \in \Gamma$.

O próximo quadro, proveniente das Tabelas 4.16-17, nos permite comparar de maneira mais direta a qualidade dos métodos.

Tabela 4.18

k	EE1/EE2	E1/E2
0	3,07	17,88
1	2,99	1,28
2	3,56	2,64
3	3,07	1,14
4	3,36	0,27
5	2,03	0,18

Em média, EE1 é 3 vezes maior que EE2. Outro fato interessante é que E1/E2 apresentou maior variância que E1/E2 e o segundo método foi melhor que o primeiro somente para $k \in \{0, \dots, 3\}$.

Queremos estudar agora o comportamento dos métodos para diferentes valores de γ , escolhemos para as próximas simulações T_{15} . Lembrando que continuamos a utilizar o método de integração numérica Dormand-Prince com tolerância relativa do passo de integração $1e-4$.

Figura 4.19a

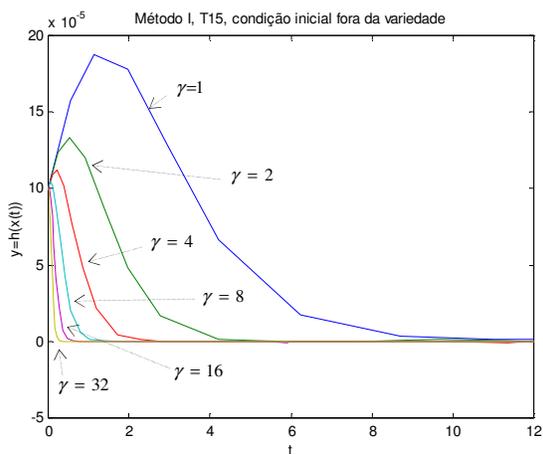
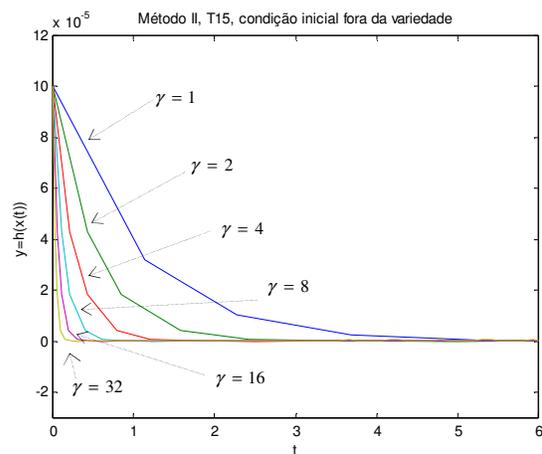


Figura 4.19b



Como esperado para ambos os métodos (vide comentários de (3.7) e (2.39)), a velocidade de convergência aumenta com γ . Entretanto, uma rápida convergência para a variedade também requerer uma rápida dinâmica de $Y^{(\rho)}$, o que pode acarretar problemas numéricos.

Analisaremos melhor este experimento, cujo intervalo de simulação foi $t \in [0,1000]$, através das tabelas abaixo (novamente toma-se $x_1 \equiv 2$ como solução analítica). A influência de γ sobre a restrição y consta na Tabela 4.20 e sobre x_1 na Tabela 4.21.

Definimos t^* como o menor \bar{t} tal que $|y(t)| < e_p \forall t > \bar{t}$, de maneira análoga, definimos \hat{t}^* como o menor \bar{t} tal que $|x_1(t)| < \hat{e}_p \forall t > \bar{t}$.

Tabela 4.20

γ	TP \doteq Tempo de Processamento		$e_p \doteq \max_{t \in [15,1000]} y $		t^*		$e_T \doteq \max_{t \in [0, t^*]} y $	
	Método I	Método II	Método I	Método II	Método I	Método II	Método I	Método II
1	1,33	1,24	2,54E-06	1,25E-06	10,994	5,418	1,88E-04	1,00E-04
2	2,50	2,36	3,13E-06	1,41E-06	4,203	2,419	1,34E-04	1,00E-04
4	4,98	4,80	6,24E-07	1,37E-06	2,835	1,209	1,12E-04	1,00E-04
8	10,25	9,52	4,82E-07	2,28E-06	1,398	0,605	1,04E-04	1,00E-04
16	21,66	18,99	3,65E-07	1,21E-06	0,771	0,302	1,01E-04	1,00E-04
32	45,00	37,86	1,10E-07	2,42E-06	0,349	0,151	1,00E-04	1,00E-04
64	97,62	71,59	1,10E-07	4,53E-06	0,215	0,049	1,00E-04	1,00E-04
128	223,29	151,25	3,45E-08	4,32E-06	0,115	0,025	1,00E-04	1,00E-04
256	521,60	279,29	1,77E-08	1,53E-05	0,056	0,012	1,00E-04	1,00E-04

Tabela 4.21

γ	$\hat{e}_m \doteq \text{média}_{t \in [15,100]} x_1 - 2$		$\hat{e}_p \doteq \max_{t \in [15,100]} x_1 - \hat{e}_m $		\hat{t}^*		$\hat{e}_T \doteq \max_{t \in [0, \hat{t}^*]} x_1 - \hat{e}_m $	
	Método I	Método II	Método I	Método II	Método I	Método II	Método I	Método II
1	5,00E-02	3,01E-02	2,71E-04	1,89E-04	10,994	5,418	5,00E-02	3,01E-02
2	3,70E-03	1,50E-02	2,92E-04	8,61E-05	4,203	2,419	6,34E-03	1,50E-02
4	1,05E-02	7,50E-03	2,67E-04	1,23E-04	2,278	1,209	1,05E-02	7,51E-03
8	1,58E-02	3,80E-03	2,99E-04	1,21E-04	1,398	0,396	1,78E-02	3,77E-03
16	1,81E-02	1,90E-03	3,10E-04	8,11E-05	0,771	0,198	3,85E-02	1,91E-03
32	1,91E-02	1,00E-03	2,83E-04	5,69E-05	0,349	0,099	8,05E-02	9,81E-04
64	1,96E-02	5,13E-04	3,00E-04	6,21E-05	0,215	0,027	1,68E-01	5,13E-04
128	1,98E-02	2,76E-04	2,68E-04	5,69E-05	0,115	0,013	3,37E-01	2,76E-04
256	1,99E-02	1,52E-04	2,20E-04	8,17E-05	0,056	0,003	6,87E-01	1,52E-04

Observamos que o tempo de processamento (TP) nos dois métodos cresce com γ , além disso, o TP para o Método I cresce aproximadamente duas vezes mais rápido do que para o Método II.

O erro de y em regime permanente (e_p) para o Método I decresce com o aumento de γ e para o Método II é aproximadamente constante com média $3,78E-06$. E o erro de x_1 em regime permanente (\hat{e}_p) oscila ao redor de $2,79E-04$ e $9,53E-05$ para os métodos I e II, respectivamente.

Temos que t^* e \hat{t}^* decrescem exponencialmente com o aumento de γ nos dois métodos e que o Método II atinge t^* e \hat{t}^* mais rápido que o Método I.

O erro cometido durante o transitório (e_T) para o segundo método decai com γ , atingindo o erro da condição inicial $y(h(x_0))=1,00E-04$ para $\gamma \geq 32$, ao passo que no segundo método e_T é constante igual $1,00E-04$.

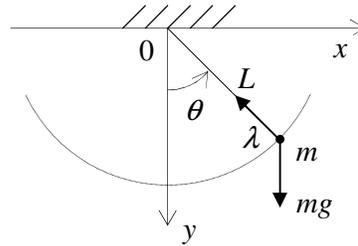
Os resultados mais importantes deste experimento referem-se aos erros de transitório (\hat{e}_T) e médio (\hat{e}_m) relativos a x_1 . Para o Método II, \hat{e}_T e \hat{e}_m tendem a diminuir com o aumento γ , em contrapartida, estas medidas aumentam com γ no Método I.

Portanto, se desejamos melhorar características de convergência para a restrição e que a solução a ser seguida na variedade tenha condição inicial real mais próxima da condição inicial dada, devemos optar pelo Método II para esta classe de problemas.

Exemplo 2: Pêndulo

Consideremos o problema do pêndulo simples sem amortecimento, ilustrado na figura abaixo, onde m é a massa fixada na extremidade inferior do fio de comprimento L , g é a aceleração da gravidade, λ é a tensão no fio e θ é a coordenada angular contada a partir da vertical no sentido anti-horário.

Figura 4.22



Escolhemos $m = 1$ e chegamos a seguinte formulação do problema

$$\begin{cases} \dot{x} = -\lambda x \\ \dot{y} = -\lambda y - g \\ x^2 + y^2 - L^2 = 0 \end{cases} \quad (4.11)$$

Reescrevermos (4.11) como

$$\begin{cases} \dot{x} = f(x, u) \\ y = h(x) = 0 \end{cases} = \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -u x_1 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = -u x_3 - g \\ y = x_1^2 + x_2^2 - L^2 \end{cases} \quad (4.12)$$

$$\text{Veja que } \begin{cases} y = x_1^2 + x_2^2 - L^2 \doteq y^{(0)} \\ \dot{y}^{(0)} = 2(x_1 x_2 + x_3 x_4) \doteq y^{(1)} \\ \dot{y}^{(1)} = 2[(x_2^2 + x_4^2) - (x_1^2 + x_3^2 + x_3 g)u] \doteq y^{(2)} \end{cases}$$

Então $\rho = 2$, $Y^{(\rho)} = (y^{(0)}, y^{(1)}, y^{(2)})$ conforme Algoritmo 3.1.

Note que para este exemplo simples a escolha de \hat{x} para completar $Y^{(\rho-1)}$ até uma base de \mathbb{R}^4 não é tão trivial como no exemplo anterior.

O sistema para o Método I (vide 4.3 e 4.4) é

$$\begin{cases} \dot{y}^{(0)} = y^{(1)} \\ \dot{y}^{(1)} = y^{(2)} \\ \dot{\hat{x}} = \frac{\partial \hat{x}}{\partial x}(f(x)) \end{cases}$$

O sistema para o Método II (vide 4.5) é

$$\begin{cases} \dot{y}^{(0)} = -\lambda y^{(0)} \\ \dot{y}^{(1)} = -\lambda y^{(1)} \\ \dot{y}^{(2)} = -\lambda y^{(2)} \\ \dot{\hat{x}} = \frac{\partial \hat{x}}{\partial x}(f(x)) \end{cases}$$

Consideraremos $\gamma=1$ como padrão. Utilizaremos o método de integração numérica Dormand-Prince (ode45 MatLab), com tamanho de passo variável e tolerância relativa RTOL=1e-4. Assumiremos também para as simulações $L=1$, $g=9.8$.

Seja uma condição inicial $(x_0, u_0) \in \Gamma$ com $x_0 = (1 \ 0 \ 0 \ 0)^T$ e $u_0 = 0$.

Avaliaremos o resultado da simulação após o transitório.

Figura 4.23a

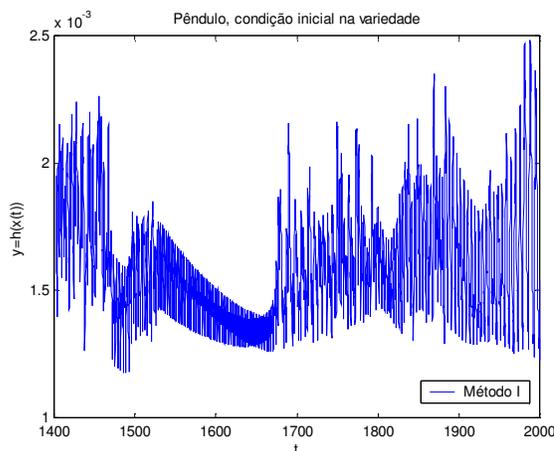
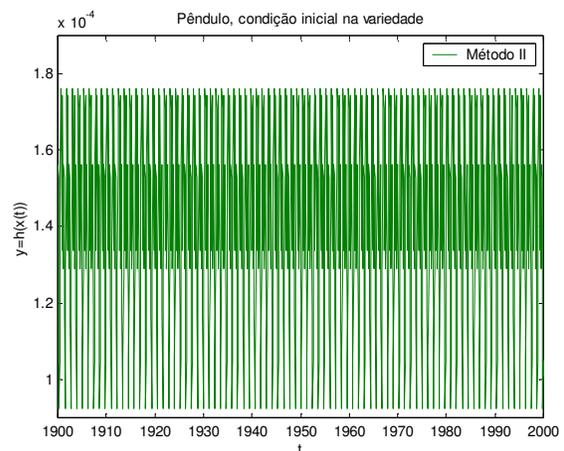


Figura 4.23b



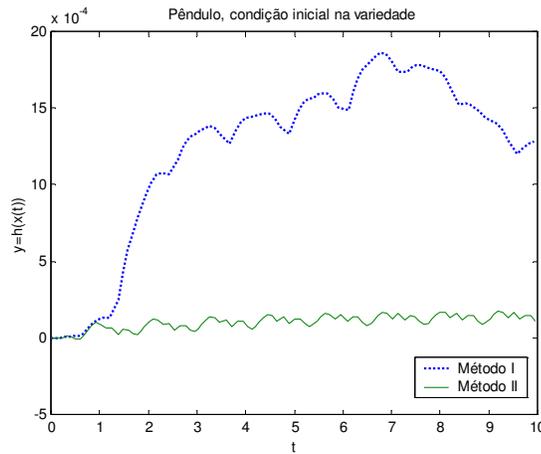
Assim como para o Exemplo 1, verificamos que tanto o Método I quanto o Método II mantiveram o erro relativo à restrição próximo de zero.

Os erros apresentados foram respectivamente da ordem de 10^{-3} e 10^{-4} , entretanto, a distribuição do segundo método foi homogênea e isto não aconteceu com o primeiro método.

Outro fato importante é que a distribuição dos erros não se concentrou ao redor de zero, mas sim em torno de $1.7e-3$ para o primeiro método e $1.4e-4$ para o segundo. Este fenômeno também ocorreu no exemplo linear (vide Figura 4.10a-b)

Segue abaixo a mesma simulação para t perto da origem.

Figura 4.24



Sabendo que o período do pêndulo é dado pela expressão [7]

$$T = 4 \sqrt{\frac{l}{g}} \int_0^{\pi/2} \frac{d\phi}{\sqrt{1 - \left| \sin \frac{\theta_0}{2} \right|^2 \sin^2 \phi}} \quad (4.13)$$

Averiguaremos se os estados gerados pela simulação acompanham a solução analítica medindo a diferença absoluta entre os ângulos θ e θ_0 , após o tempo de simulação igual $t = T$. O cálculo da integral elíptica acima será feito numericamente, até uma precisão da ordem de 10^{-14} .

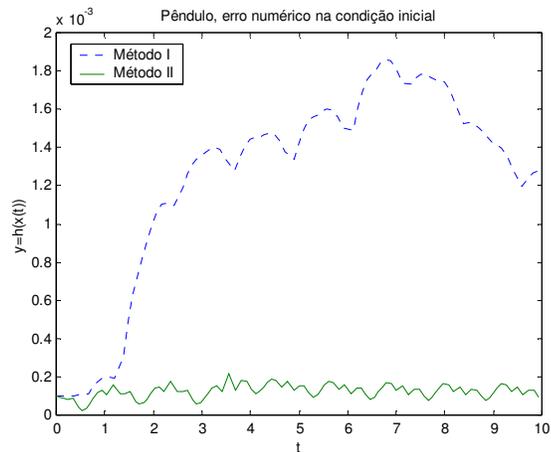
Para esta mesma condição inicial, tanto para o Método I quanto para o Método II obtivemos $|\theta_T - \theta_0| \approx 2,40E-03$.

Seja agora uma condição inicial $(\tilde{x}_0, \tilde{u}_0) \notin \Gamma$,

$$\tilde{x}_0 = (-1.00005 \quad 0 \quad 0 \quad 0)^T \text{ e } \tilde{u}_0 = 0.1.$$

Veremos a simulação para t perto da origem.

Figura 4.25



Ambos os métodos acentuaram o erro da condição inicial e apresentaram ruídos semelhantes ao da Figura 4.24 aproximadamente após $t = 3s$

Seguem os resultados desta simulação após o transitório

Figura 4.26a

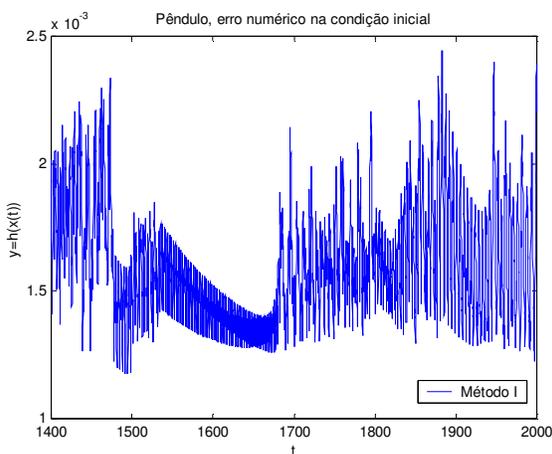
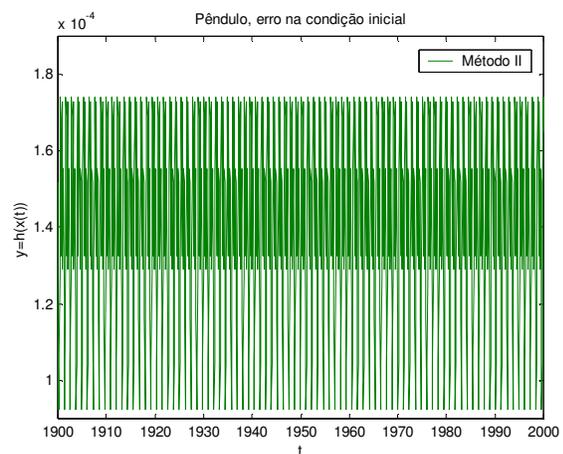


Figura 4.26b



Podemos notar que as Figuras 4.26a-b são muito semelhantes às Figuras 4.23a-b, concluímos daí que, para este exemplo, o erro em relação à variedade em regime estacionário não foi afetado pelo erro numérico na condição inicial.

A medida de proximidade da solução analítica $|\theta_T - \theta_0|$ foi coletada, obtendo 8,30E-03 para o primeiro método e 7,40E-03 para o segundo. Notamos que um erro na condição inicial de cerca de 1,00E-04 originou $|\theta_T - \theta_0|$ aproximadamente 8 vezes maior do que para uma condição inicial justa na restrição.

Ainda com $(\tilde{x}_0, \tilde{u}_0) \notin \Gamma$, estudaremos o efeito de γ neste problema não linear, de maneira semelhante ao que fizemos no Exemplo 1.

Figura 4.27a

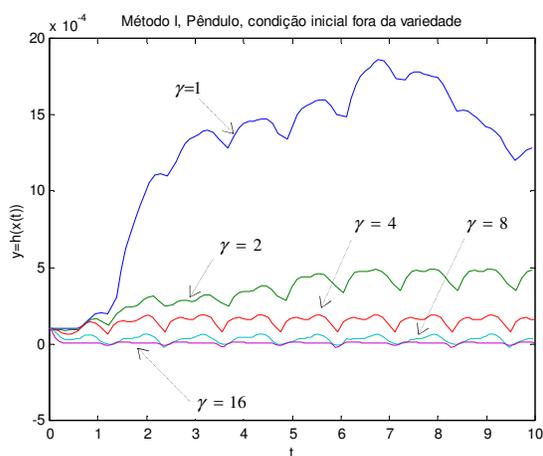
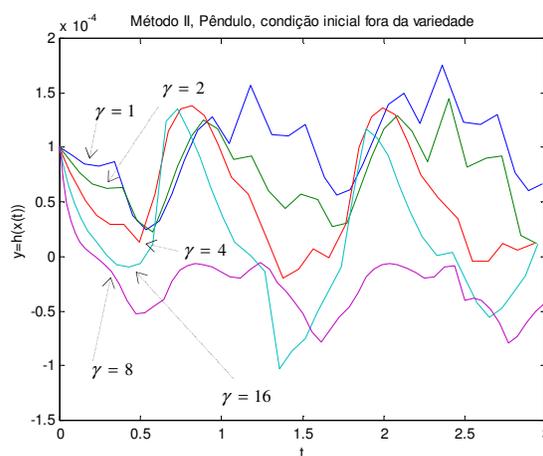


Figura 4.27b



Para o Método I, podemos ver que o aumento de γ leva a soluções mais próximas da restrição. Ao analisar o Método II perto da origem, percebemos a princípio que o crescimento de γ impõe uma velocidade maior de convergência, por outro lado, erros numéricos podem prejudicar a precisão deste método.

Vejamos a tabela abaixo para mais informações sobre este ensaio.

Tabela 4.28

γ	TP \doteq Tempo de Processamento		$ \theta_T - \theta_0 $		$e_T \doteq \max_{t \in [0,3]} y $		$e_p \doteq \max_{t \in [3,100]} y $	
	Método I	Método II	Método I	Método II	Método I	Método II	Método I	Método II
1	2,03	2,69	8,30E-03	7,40E-03	1,35E-03	1,75E-04	2,00E-03	2,14E-04
2	2,08	2,77	3,90E-03	8,50E-03	3,13E-04	1,45E-04	4,90E-04	1,81E-04
4	2,30	2,78	2,30E-03	6,80E-03	1,86E-04	1,38E-04	1,92E-04	1,69E-04
8	3,17	3,34	1,60E-03	4,10E-03	1,00E-04	1,36E-04	6,29E-05	1,42E-04
16	5,38	4,53	1,80E-03	2,60E-03	1,00E-04	1,00E-04	2,74E-05	8,02E-05
32	9,05	7,17	1,20E-03	2,90E-03	1,00E-04	1,00E-04	1,71E-05	5,81E-05
64	20,45	10,78	1,00E-03	1,50E-03	1,00E-04	1,00E-04	1,13E-05	5,19E-05
128	27,61	18,31	5,00E-04	9,00E-04	1,00E-04	1,00E-04	5,23E-06	5,45E-05
256	48,36	26,72	4,00E-04	5,00E-04	1,00E-04	1,00E-04	3,50E-06	8,25E-05

O tempo de processamento (TP) continuou crescendo com γ . E o TP do Método I cresceu aproximadamente duas vezes mais rápido que para no Método II.

A distância entre a solução numérica e a analítica, medida por $|\theta_T - \theta_0|$, diminuiu exponencialmente com γ para ambos os métodos. Temos que primeiro método foi um pouco mais preciso que o segundo, excetuando-se $\gamma=1$.

O erro de transitório (e_T) diminuiu exponencialmente com γ , atingindo o erro numérico da condição inicial para $\gamma=8$ no primeiro método e para $\gamma=16$ no segundo.

O erro em regime permanente (e_r) também diminuiu exponencialmente com γ nos dois casos. Observamos que para $\gamma=\{1,2,4\}$ o e_r foi menor para o Método II e que para valores mais altos de γ o Método I foi mais preciso.

Conclusão

Com base na teoria, foram elaborados algoritmos que geraram um sistema explícito, tanto pelo Método I quanto pelo Método II, a partir de um sistema semi-implícito dado num formato especificado no trabalho.

Utilizando estas ferramentas, foram feitos experimentos numéricos que concordaram com os resultados teóricos de convergência estudados.

Verificou-se que ambos os métodos foram capazes de corrigir assintoticamente os erros numéricos em relação à restrição, mesmo para sistemas possuindo matrizes mal condicionadas ou não linearidades.

Com o exemplo de matrizes mal condicionadas, pode-se concluir que a análise somente da saída em relação à restrição não é suficiente para aferir a qualidade dos métodos, pois os estados afetados pela dinâmica zero podem diferir da solução analítica, mesmo com erro de saída próximo de zero.

Vimos que, abaixo de determinado fator de estabilidade, os dois métodos passaram a divergir igualmente da solução analítica, acompanhando o decréscimo do número de condicionamento (r_{cond}). E que o fator de velocidade de convergência γ também modifica a estabilidade do sistema. Constatamos neste exemplo que o Método II é de fato mais preciso que o Método I, principalmente porque ele reduz o erro em relação à solução analítica.

No exemplo do pêndulo, o aumento de γ ocasionou a melhora de características de convergência para a variedade e para a solução analítica, de maneira bastante semelhante em ambos os métodos. Contudo, observamos que o segundo método foi um pouco melhor que o primeiro para $\gamma > 1$.

Estudos subseqüentes podem ser feitos para analisar diversos outros aspectos destes problemas. A aplicação de teoria de métodos numéricos com o objetivo de prever a acuidade dos métodos é um deles. Outro estudo possível é sobre a complexidade computacional para a geração das soluções.

Generalizações destes algoritmos também poderiam ser desenvolvidas de forma a lidar com sistemas variantes no tempo, restrições de saída dependentes explicitamente da entrada ou sistemas não completamente determinados.

Apêndice

Encontram-se neste apêndice a prova do Teorema 2.12 e os códigos fonte desenvolvidos em Matlab utilizados para as simulações.

Teorema 1.12 Sejam $\rho \doteq \sum_{i=1}^m \rho_i$, denominado grau relativo do sistema, e a aplicação $S : X \rightarrow R^\rho$ definida por

$$(h_1(x), L_f h_1(x), \dots, L_f^{\rho_1-1} h_1(x), h_2(x), \dots, h_m(x), \dots, L_f^{\rho_m-1} h_m(x)) \quad (1.20)$$

Quando a propriedade (1.14) é válida para o sistema (1.1), então

$$\text{posto } S(x) = \rho \quad (1.21)$$

Prova

Suporemos por absurdo que, para algum $x_0 \in X$, o $\text{rank } S(x) \neq \rho$. Para isso os vetores

$$dh_1(x_0), dL_f h_1(x_0), \dots, dL_f^{\rho_1-1} h_1(x_0), dh_2(x_0), \dots, dh_m(x_0), \dots, dL_f^{\rho_m-1} h_m(x_0) \quad (A.1)$$

devem ser linearmente dependentes. Isto equivale a existência de números reais não todos nulos c_{ik} , $i \in \{1, \dots, m\}$, $k = \{0, \dots, \rho_i - 1\}$, tais que

$$\sum_{i=1}^m \sum_{k=0}^{\rho_i-1} c_{ik} dL_f^k h_i(x_0) = 0 \quad (A.2)$$

Aplicando o campo vetorial $g_j(\cdot)$ à função $\sum_{i=1}^m \sum_{k=0}^{\rho_i-1} c_{ik} L_f^k h_i(x)$

$$L_{g_j} \left(\sum_{i=1}^m \sum_{k=0}^{\rho_i-1} c_{ik} L_f^k h_i(x) \right) = \left\langle \sum_{i=1}^m \sum_{k=0}^{\rho_i-1} c_{ik} L_f^k h_i(x), g_j \right\rangle = \sum_{i=1}^m c_{i\rho_i-1} b_{ij}(x) \quad (A.3)$$

Onde $b_{ij}(x)$ é a matriz de desacoplamento (1.13).

Salienta-se que a última passagem de (A.3) é válida pela definição de número característico (1.9), já que, para $L_f^k h_i(x) \equiv 0$, $k \in \{1, \dots, \rho_i - 2\}$.

A expressão (A.3) anula-se quando nela substituirmos (A.2), entretanto a hipótese (1.14) diz que $\text{posto } b(x) = m$, $\forall x \in X$. Para que isto aconteça $c_{i\rho_i-1} = 0$, $i \in \{1, \dots, m\}$.

Deste modo (A.2) reduz-se a

$$\sum_{i=1}^m \sum_{k=0}^{\rho_i-2} c_{ik} dL_f^k h_i(x_0) = 0 \quad (\text{A.4})$$

Aplicando o campo vetorial $[g_j, f](\cdot)$ à função $\sum_{i=1}^m \sum_{k=0}^{\rho_i-2} c_{ik} L_f^k h_i(x)$

$$L_{[g_j, f]} \left(\sum_{i=1}^m \sum_{k=0}^{\rho_i-2} c_{ik} L_f^k h_i(x) \right) = \quad (\text{A.5a})$$

$$L_f L_{g_j} \left(\sum_{i=1}^m \sum_{k=0}^{\rho_i-2} c_{ik} L_f^k h_i(x) \right) - L_{g_j} L_f \left(\sum_{i=1}^m \sum_{k=0}^{\rho_i-2} c_{ik} L_f^k h_i(x) \right) \quad (\text{A.5b})$$

O primeiro termo de (A.5b) é igual a zero, pois $L_f^k h_i(x) \equiv 0$, $k \in \{1, \dots, \rho_i - 2\}$. Então, utilizando o mesmo argumento que em (A.3), (A.5a) equivale a

$$L_{g_j} \left(\sum_{i=1}^m \sum_{k=0}^{\rho_i-2} c_{ik} L_f^{k+1} h_i(x) \right) = \sum_{i=1}^m c_{i\rho_i-1} b_{ij}(x) \quad (\text{A.6})$$

Novamente, a expressão (A.6) anula-se quando nela substituimos (A.2), entretanto (1.14) diz que posto $b(x) = m$, $\forall x \in X$. Para que isto aconteça $c_{i\rho_i-2} = 0$, $i \in \{1, \dots, m\}$.

A repetição deste raciocínio implica em $c_{ik} = 0$, $i \in \{1, \dots, m\}$, $k = \{0, \dots, \rho_i - 1\}$. Concluimos então a tese (1.21) de que $\text{rank } S(x) = \rho$.

Código fonte do Algoritmo 3.1

```

function[Y, YP, JY, JYP, p] = TransformaCoordenadas(f, h, x, u)

%Inicializando variáveis auxiliares
Y = []; JY = [];
YP = []; JYP = [];
p = [];

[n, aux] = size(f);
[l, aux] = size(h);

%Construindo JY, JYP, Y, YP, p
for i=1:l

    k = 0;
    h_ = h(i);
    jhu = jacobian(h_, u);

    while (all(jhu == 0))

        Y = [Y; h_];

        jhx = jacobian(h_, x);

        JY = [JY; jhx];

        h_ = jhx*f;
        jhu = jacobian(h_, u);

        k = k + 1;
    end

    p = [p; k];

    YP = [YP;
         h_];

    jhx = jacobian(h_, x);

    JYP = [JYP;
           jhx jhu];

end

```

Código fonte do Algoritmo 3.3

```

function[R] = CoordenadasComplementares(JYb)

[rho, n] = size(JYb);
[q, r] = qr(JYb');
R = q(:, (rho+1):(n))';

```

Código fonte do Algoritmo 3.4

```

function[] = ConstroiImplementacaoI(gama, f, h, x, u, cte, vcte)

%Inicializando variáveis
v = []; Yaux = []; xu = [x; u]; l = length(u);

ConstroiFuncao('gama', gama, [], [], [] );
ConstroiFuncao('f' , f , xu, cte, vcte);
ConstroiFuncao('h' , h , x , cte, vcte);

[Y, YP, JY, JYP, p] = TransformaCoordenadas(f, h, x, u);

ConstroiFuncao('JY' , JY , xu, cte, vcte);

%Separando YP = a(x) + b(x)u
a = valor(YP, u, zeros(1,l));
b = valor(YP - a, u, ones(1,l));

%Construção da Realimentação de estado
invb = inv(b); alfa = -invb*a; beta = invb;

for i=1:l;
    v=[v; -CoefPoli(p(i),gama)* Y(sum(p(1:length(p)<i)) + 1:p(i))];
end

feedback = alfa + beta * v;

ConstroiFuncao('feedback', feedback, x, cte, vcte);

for i=1:l;
    for k=1:p(i)-1
        Yaux = [Yaux; Y(sum(p(1:length(p)<i)) + k + 1)];
    end
    Yaux = [Yaux; a(i) + b(i,:)*u(i)];
end

ConstroiFuncao('Yaux', Yaux, xu, cte, vcte);

```

Código fonte de funções auxiliares do Algoritmo 3.4

```

function[fx] = Valor(f, x, x_)
%Calcula o valor de f(x) no ponto x_

[n, aux] = size(x);
string = '[';

for i=1:n
    if (i==n)
        string = strcat(string, char(x(i)), '=', num2str(x_(i)));
    else
        string = strcat(string, char(x(i)), '=', num2str(x_(i)), ',');
    end
end

string = strcat(string, ']');
fx = maple('eval', f, string);

```

```

function [fatn] = Fatorial(n)

if (n==0)
    fatn = 1;
else
    fatn = prod(1:n);
end

function [alfa] = CoefPoli(n, gama)

alfa = zeros(1,n);

for r=1:n
    alfa(n-r+1) = Fatorial(n)/( Fatorial(r) * Fatorial(n-r)) * gama^r;
end

function [] = ConstroiFuncao(out, f, x, cte, vcte)

fid = fopen(strcat(out, '.m'), 'w');

fprintf(fid, strcat('function [out] = ', out, '(x)'));

[n, m ]=size(f);
[k, aux]=size(x);

%Estruturando as variáveis simbólicas como um vetor numérico
if (k>0)
    fprintf(fid, strcat('\n\n', char(37), 'Recuperando variáveis
simbólicas\n'));

    for i=1:k
        fprintf(fid, strcat(char(x(i)), '=', 'x(', num2str(i), '), '); \n'));
    end
end

%Atribuindo valor às ctes
[k, aux ]=size(cte);
if (k~=0)
    fprintf(fid, strcat('\n\n', char(37), 'Atribuindo valores às constantes\n'));

    for i=1:k
        fprintf(fid, strcat(char(cte(i)), '=', num2str(vcte(i)), '); \n'));
    end
end

%Escrevendo função
fprintf(fid, strcat('\n\n', char(37), 'Escrevendo a função\n'));

fprintf(fid, strcat('out', '=zeros(', num2str(n), ', ', num2str(m), '); \n\n'));

for i=1:n
    for j=1:m
        if isnumeric(f(i,j))
            fprintf(fid, strcat('out', '(', num2str(i), ', ', num2str(j), ')=', num2str(f(i,j)), '); \n'));
        else
            fprintf(fid, strcat('out', '(', num2str(i), ', ', num2str(j), ')=', char(f(i,j)), '); \n'));
        end
    end
end

status = fclose(fid);

```

Código fonte do Algoritmo 3.6

```
function[fxb] = F1(xb)
%Método I

ub = feedback(xb); xub = [xb; ub];

JYb = JY(xub);
R = CoordenadasComplementares(JYb);

Tb = [R;
      JYb];

taub = [R*f(xub);
        Yaux(xub)];

fxb = Tb\taub;
```

Código fonte do Algoritmo 3.7

```
function[] = ConstroiImplementacaoII(gama, f, h, x, u, cte, vcte)

%Inicializando variáveis
xu = [x; u];

ConstroiFuncao('gama', gama, [], [], []);
ConstroiFuncao('f', f, xu, cte, vcte);
ConstroiFuncao('h', h, x, cte, vcte);

[Y, YP, JY, JYP, p] = TransformaCoordenadas(f, h, x, u);

ConstroiFuncao('Y', Y, xu, cte, vcte);
ConstroiFuncao('JY', JY, xu, cte, vcte);
ConstroiFuncao('YP', YP, xu, cte, vcte);
ConstroiFuncao('JYP', JYP, xu, cte, vcte);
```

Código fonte do Algoritmo 3.9

```
function[fxub] = F2(xub)
%Método II

JYb = JY(xub);
R = CoordenadasComplementares(JYb);
JYPb = JYP(xub);

[m, aux] = size(JYPb); [rho, n] = size(JYb);

Tb = [ R, zeros(n-rho, m);
      JYb, zeros(rho, m);
      JYPb];

taub = [R*f(xub);
        -gama*Y(xub);
        -gama*YP(xub)];

fxub = Tb\taub;
```

Referências Bibliográficas

[1] SILVA, P. S. P. DELALEU, E; SOUZA, I. S. *On Geometric control and numeric integration of DAES's*. In: IFAC World Congress'05, 2005, Praga, 2005, v.1, p-16

[2] NIJMEIJER, H.; SHAFT, A. J. VAN DER. *Nonlinear dynamical control systems*. New York: Springer-Verlag, 1990. xiii, 467 p.

[3] SOUZA, I. S., *Geometria do desacoplamento e integração numérica de equações diferenciais não lineares implícitas*, Dissertação de Mestrado, Escola Politécnica da USP, 2006

[4] LIMA, E. L, *Curso de Análise*, vol 2, IMPA, 1981, 557p.

[5] PEREIRA DA SILVA, P. S; WATANABE, C. J. *Some geometric properties of differential-algebraic equations*. Article submitted, available in 2002.

[6] STRANG, G. *Linear algebra and its applications*, 2nd ed. New York: Academic Press, 1980.

[7] FIGUEIREDO, D.; NEVES, A. F.; *Equações Diferenciais Aplicadas*, 2ª ed. IMPA, 2002, 301p.