

# 9

## CAPÍTULO

# DETERMINAÇÃO DE UMA SOLUÇÃO APROXIMADA DA EQUAÇÃO DE DUFFING POR MEIO DE ALGORITMO GENÉTICO

*Eliabe Jacinto Silva<sup>1</sup>*  
*Donald Mark Santee<sup>1</sup>*

**Agradecimentos:** Ao professor Dr. Donald Mark Santee, à Universidade Federal de Goiás e a todos que contribuíram para este trabalho.

**Resumo:** Este trabalho visou testar uma abordagem do algoritmo genético (GA, do inglês, *genetic algorithm*) para a determinação de uma solução apro-

---

<sup>1</sup> Unidade Acadêmica Especial de Matemática e Tecnologia, Universidade Federal de Goiás – Regional Catalão, Catalão, Brasil.  
E-mail de contato: eliabyjsilva@gmail.com

ximada da equação de Duffing, na tentativa de obter resultados significantes e abrangentes no espaço dos parâmetros, maiores do que os resultados obtidos por métodos de perturbação, principalmente sobre as hipóteses de pequena não linearidade e/ou pequenas amplitudes de vibração. Por meio do auxílio de um método numérico (Runge-Kutta de 4ª ordem), comparamos os resultados obtidos com os resultados alcançados pelo algoritmo genético, possibilitando calcular a aptidão dos dados. Essa solução poderá servir de referência para o desenvolvimento de novos métodos de perturbação.

**Palavras-chave:** Algoritmo genético. Equação de Duffing. Estruturas.

**Abstract:** This paper tests the genetic algorithm (GA) approach to determine an approximate solution for the Duffing equation, and aims to achieve significant results and that has a range in the parameter space, higher than the results obtained by perturbation methods, especially for the cases of small non-linearity and/or small vibration amplitudes. Through the aid of a numerical method (4<sup>th</sup> order Runge-Kutta) it will be compared with the results obtained by genetic algorithm (GA), allowing to calculate the suitability of the results. This solution can serve as a reference for developing new disruption methods.

**Keywords:** Genetic algorithm. Duffing equation. Structures.

## 1 INTRODUÇÃO

O uso de colunas é muito comum em construções civis, por isso seu estudo merece minuciosa atenção e os critérios de dimensionamento devem ser bem estabelecidos para garantir a segurança da estrutura, evitando o comprometimento dela. Em particular, colunas semienterradas são usadas em chaminés, torres, estruturas *off-shore*, etc.

Essas colunas podem aparecer totalmente enterradas, desenterradas ou semienterradas nos mais diferenciados tipos de solos. Existem modelos e soluções propostas para o caso totalmente enterrado e totalmente desenterrado, como o que se encontra em Queiroz e Santee (2007). Já para o caso semienterrado, poucas são as referências na literatura, existindo uma carência de estudos nesse tópico.

Alguns modelos de colunas semienterradas são mais conhecidos, como os propostos por Sampaio (2004). Estudos anteriores, que fazem parte do presente projeto de pesquisa, desenvolveram um modelo matemático que descreve o comportamento de uma coluna semienterrada em um solo quadraticamente não linear do tipo “com enfraquecimento”.

## 2 EQUAÇÃO DE DUFFING

O modelo matemático obtido a partir de funções de forma quadráticas e com um carregamento transversal periódico recai na equação diferencial ordinária conhecida como equação de Duffing e possui a seguinte forma geral:

$$\ddot{x} + c\dot{x} + w_0^2x + \alpha x^3 = F\cos(\Omega t) \quad (1)$$

na qual as constantes  $c$ ,  $w_0$  e  $\alpha$  dependem das características geométricas da coluna e das propriedades do solo.

$w_0$  = frequência natural de oscilação

$c$  = amortecimento

$\alpha$  = termo de não linearidade

$F$  = forçamento

A solução da Equação diferencial (1) descreve como a coluna se deforma com o tempo e não possui solução exata. As soluções para essa equação são obtidas por integração numérica ou, no caso da necessidade de uma fórmula algébrica para a solução, por métodos de perturbação. Essas soluções possuem uma abrangência limitada às hipóteses que foram efetuadas na obtenção da respectiva solução aproximada.

Uma estratégia ainda não utilizada para chegar a uma solução aproximada para a Equação (1) é o algoritmo genético, que tem mostrado resultados bem interessantes, chegando a reproduzir e criar patentes novas (LINDEN, 2012).

Utilizou-se o Octave para implementar o algoritmo de Runge-Kutta de 4ª ordem, além de um *script* organizador e da utilização do algoritmo genético do programa.

## 3 RUNGE-KUTTA DE 4º ORDEM

Em pesquisas, determinou-se que seria ideal obter uma solução da equação de Duffing por meio de um método numérico, de modo que o método “Runge-Kutta de 4ª ordem” foi escolhido para obtermos resultados que serão comparados com os resultados de um algoritmo genético (GA) nas mesmas condições.

Esse método consiste em se fazer mudanças no método de Euler para se conseguir um método baseado na série de Taylor de 2ª ordem, de tal forma que elimine o cálculo de derivadas de 2ª ordem.

## 4 MÉTODO DA MUDANÇA DE VARIÁVEL

Foi feita uma mudança de variáveis no sistema original (1) para que possa ser aplicado o método de Runge-Kutta de 4ª ordem.

A mudança transforma a equação Diferencial Ordinária de 2ª ordem em um sistema de 1ª ordem (FRIEDLAND, 2005), utilizado pelo método de Runge-Kutta.

Equação de Duffing:

$$\ddot{x} + c\dot{x} + w_0^2 x + \alpha x^3 = F \cos(\Omega t) \quad (1)$$

## 4.1 Mudança de variável da equação de Duffing

Isolando  $\ddot{x}$ , obtemos:

$$\ddot{x} = -c\dot{x} - w_0^2 x - \alpha x^3 + F \cos(\Omega t) \quad (2)$$

Fazendo [ $x_1 = x$  e  $x_2 = \dot{x}$ ]

Se derivamos  $x_1 = x$  dos dois lados, temos:

$$x_1 = x \quad \dot{x}_1 = \dot{x}$$

Sabemos que  $x_1 = x$   $\dot{x}_1 = \dot{x}$  logo  $\dot{x}_1 = x_2$

Se derivamos  $x_2 = \dot{x}$  dos dois lados, temos:

$$x_2 = \dot{x} \quad \dot{x}_2 = \ddot{x}$$

Juntando os dados, temos:

$$x_1 = x$$

$$x_2 = \dot{x}$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \ddot{x}$$

Substituindo na Equação (2), temos:

$$\ddot{x} = c\dot{x} - w_0^2 x - \alpha x^3 + F \cos(\Omega t) \quad (2)$$

$$\dot{x}_1 = x_2 \quad (3)$$

$$\dot{x}_2 = -cx_2 - w_0^2 x_1 + \alpha x_1^3 + F \cos(\Omega t) \quad (4)$$

## 4.2 Octave

O Octave é um *software* de alto desempenho destinado a fazer cálculos com matrizes, podendo funcionar como uma calculadora ou como uma linguagem de programação científica (FORTRAN, Pascal, C etc.). Os comandos do Octave são mais próximos da forma como escrevemos expressões algébricas, tornando mais simples o seu uso. Atualmente, o Octave é definido como um sistema interativo e uma linguagem de programação para computação técnica e científica em geral, integrando a capacidade de fazer cálculos, visualização gráfica e programação (TONINI; SCHETTINO, 2002).

Utilizaremos o Octave para desenvolver os algoritmos, Runge-Kutta de 4ª ordem, algoritmo organizador, e utilizaremos o algoritmo genético do programa.

### 4.3 Aplicando o Método de Runge-Kutta de 4ª ordem

Consideraremos as seguintes condições iniciais mostradas na Tabela 1:

$$function[f1,f2] = tt(t,x1,x2)$$

$$f1=x2;$$

$$f2=F*cos(Frequência da Força *t)-w0^2*x1+alpha*x1^3-c*x2;$$

Tabela 1. Condições iniciais

Condições iniciais Runge-Kutta de 4ª ordem		
Numero de pontos	N	40
Pi	$\pi$	3,14
Tempo inicial	t(1)	0
x1 inicial	x1(1)	1
x2 inicial	x2(1)	0
Passo	H	$(2*\pi)/N$
Amortecimento	C	0,05
Frequência natural de oscilação	W0	1
Alpha	A	-1
Frequência da força	$\Omega$	1
Força	F	0.021

```

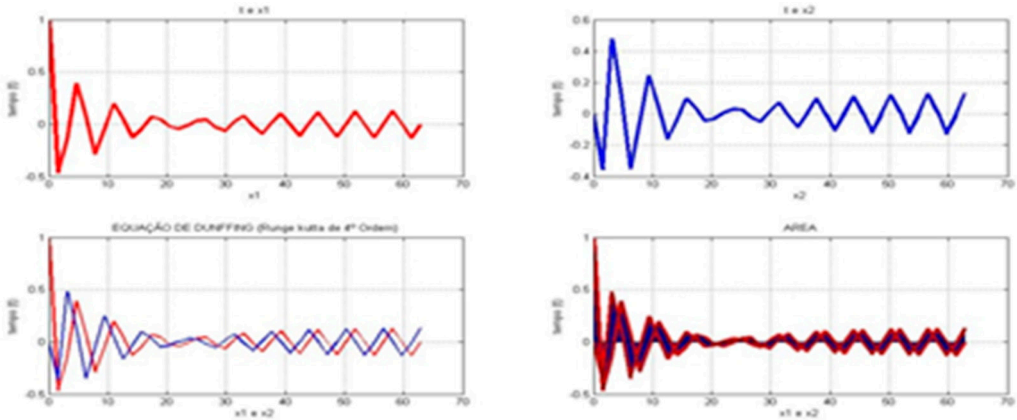
1
2
3
4 clear
5 clear all
6
7 %CONDIÇÕES INICIAIS
8 -----
9 N=4; PI=3.14; t(1)=0; x1(1)=1; x2(1)=0; h = (2*PI)/N;
10 -----
11
12 for n=1 :10*N
13
14     [k1(1),k1(2)] = tt(t(n),x1(n),x2(n));
15     [k2(1),k2(2)]=tt(t(n)+(h/2),x1(n)+(h/2)*k1(1),x2(n)+(h/2)*k1(2));
16     [k3(1),k3(2)]=tt(t(n)+(h/2),x1(n)+(h/2)*k2(1),x2(n)+(h/2)*k2(2));
17     [k4(1),k4(2)] =tt(t(n)+h,x1(n)+h*k3(1),x2(n)+h*k3(2));
18
19     x1(n+1)=x1(n)+((h./6)*(k1(1)+2*k2(1)+2*k3(1)+k4(1)));
20     x2(n+1)=x2(n)+((h./6)*(k1(2)+2*k2(2)+2*k3(2)+k4(2)));
21
22     t(n+1)=t(n)+h;
23
24 end
25

```

Figura 1. Parte Principal do método Runge-Kutta de 4ª ordem.

## 4.4 Resultados gráficos do Runge-Kutta de 4ª ordem

Esses são os resultados encontrados com o método de Runge-Kutta de 4ª ordem com a função utilizando as mesmas condições no GA, e número de ponto igual a 40. Os resultados estão dispostos em  $t$ ,  $x1$  e  $x2$ .



**Figura 2.** Gráficos do Runge-Kutta de 4ª ordem.

Os resultados numéricos são inicialmente armazenados em um arquivo de texto para podermos analisar logo em sequência.

## 5 APLICANDO O ALGORITMO GENÉTICO

### 5.1 Algoritmos evolucionários

Algoritmos evolucionários usam modelos computacionais dos processos naturais de evolução como ferramenta para resolver problemas. Os algoritmos evolucionários funcionam mantendo uma população de estruturas, denominadas indivíduos ou cromossomos, que se comportam de forma semelhante à evolução das espécies. Nessas estruturas, são aplicados os chamados operadores genéticos, como recombinação e mutação, entre outros. Cada indivíduo recebe uma avaliação que é uma quantificação da sua qualidade como solução do problema em questão. Com base nesta avaliação, serão aplicados os operadores genéticos de forma a simular a sobrevivência do mais apto (LINDEN, 2012).

Segundo Linden (2012), os algoritmos evolucionários são extremamente dependentes de fatores probabilísticos, tanto na fase de iniciação da população quanto na fase de evolução (durante a seleção dos pais, principalmente); isso faz com que os seus resultados raramente sejam perfeitamente reprodutíveis. Os al-

goritmos evolucionários são heurísticos que não asseguram a obtenção do melhor resultado possível em todas as suas execuções.

## 5.2 O que são algoritmos genéticos?

Algoritmos genéticos (GA) são um ramo dos algoritmos evolucionários e podem ser definidos como uma técnica de busca baseada no processo biológico de evolução natural. São técnicas heurísticas de otimização global; os GAs diferem de métodos como o de Hill Climbing, que seguem a derivada da função e cujo objetivo é encontrar o seu máximo, ficando retidos em máximos locais. Nos GAs, é criada uma população de indivíduos, em sequência são aplicados os operadores genéticos nessa população: seleção, recombinação (*crossover*) e mutação. Os operadores irão submeter cada indivíduo a uma qualidade como solução do problema e, eventualmente, deverão gerar um indivíduo que caracterizará uma boa solução (LINDEN, 2012).

Segundo Linden (2012), GAs são técnicas probabilísticas e não técnicas determinísticas. Assim, um GA com a mesma população inicial e o mesmo conjunto de parâmetros pode encontrar soluções diferentes a cada vez que é executado. Trabalham com uma grande população de pontos, sendo uma heurística de busca no espaço de soluções, diferencia-se dos esquemas enumerativos pelo fato de não procurar em todos os pontos possíveis, mas sim em um subconjunto desses pontos.

## 5.3 Esquema de um GA

Os algoritmos genéticos são um ramo da computação evolucionária, e, por consequente, seu funcionamento é extremamente similar, pode ser resumido algorítmicamente através dos seguintes passos:

- a) Inicie a população de cromossomos;
- b) Avalie cada cromossomo na população;
- c) Selecione os pais para gerar novos cromossomos;
- d) Aplique os operadores de recombinação e mutação a estes pais de forma a gerar os indivíduos da nova geração;
- e) Apague os velhos membros da população;
- f) Avalie todos os novos cromossomos e insira-os na população;
- g) Se o tempo acabou ou o melhor cromossomo satisfaz os requerimentos e o desempenho, retorne-o. Caso contrário, volte ao passo c.

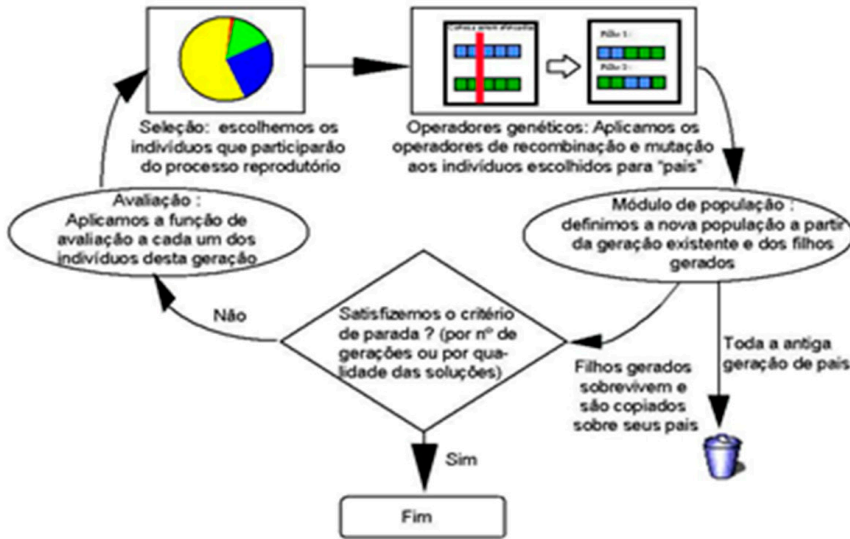


Figura 3. Esquema de um GA. Fonte: Linden (2012).

## 6 APLICAÇÃO DO ALGORITMO GENÉTICO

Função com as mesmas condições iniciais aplicadas ao método de Runge-Kutta:

$$\begin{cases} f(1) = x(2); \\ f(2) = F * \cos(\text{Frequência da Força} * t) - w0^2 * x(1) + \text{alpha} * x(1)^3 - c * x(2); \end{cases}$$

Preenchimento do algoritmo:

Tabela 2. Condições iniciais

Condições iniciais algoritmo genético		
Função de <i>fitness</i>	Fitness function	PIVIC
Número variável	Numbervariable	2
Limite superior	Lower	[0,0]
Limite inferior	Upper	[100,100]
Tamanho da população	Populationsize	64
Fração de mutação da população	Paralelo front populationfraction	0,7



Condições iniciais algoritmo genético		
Gráfico	Grafic	Paralelo front e distance
Amortecimento	C	0,05
Frequência natural de oscilação	W0	1
Alpha	A	-1
Frequência da força	$\Omega$	1
Força	F	0.021

## 7 RESULTADOS

### 7.1 Algoritmo organizador

Após vários testes utilizando os métodos Runge-Kutta de 4ª ordem e algoritmo genético, foi feito um algoritmo para manipular os dados e organizar de forma coesa os resultados, possibilitando assim enxergarmos os gráficos de ambos os resultados e, principalmente, para podermos calcular a aptidão do GA com relação ao Runge-Kutta.

- a) O algoritmo criado, primeiro, lê os resultados das fontes dos dois métodos, que estão salvos em um arquivo de texto, e os insere de forma organizada em uma planilha no Excel.

$$\text{aptidão} = \exp \left\{ - \sum_{i=0}^n \left[ x_{GA}(t_i) - x_{\text{runge-kutta}}(t_i) \right]^2 \right\}$$

Figura 4. Aptidão final.

Preenchimento dos vetores e parte do cálculo da aptidão:

```

for i=1:m
    %valores de x1 Ag e x1 rk4
    x(i,1)=R1(i);
    x(i,2)=R3(i);
    %valores de x2 Ag e x2 rk4
    y(i,1)=R2(i);
    y(i,2)=R4(i);

    resultados(i,1)=(x(i,1)-x(i,2))^2;
    resultados(i,2)=(y(i,1)-y(i,2))^2;
end

```

Figura 5. Trecho do código, primeira parte do cálculo da aptidão.

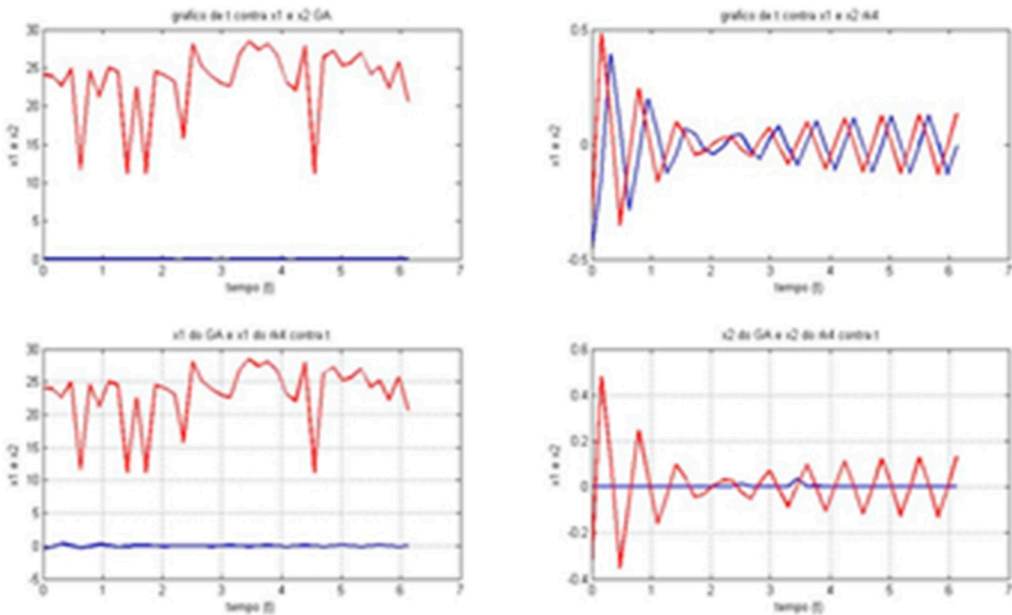
- b) Em seguida inserimos os resultados do cálculo em uma tabela no Excel, no qual é feito o restante dos cálculos.

```
excel3= xlswrite('t.xlsx',resultados,2,'B6');
```

**Tabela 3.** Resultado final da aptidão.

Aptidão x1	Aptidão x2
1,4398E-221	0,99980745

- c) E, por último, o algoritmo plota os gráficos de R1, R2, R3 e R4, correspondentes aos valores de x1 e x2 do Ga e x1 e x2 do Runge-Kutta.



**Figura 6.** Gráficos de x1 e x2 de ambos os métodos.

## 8 CONCLUSÃO

Neste capítulo, foi apresentada a equação de Duffing, que é um modelo matemático para uma coluna semienterrada e um sistema formado por um conjunto de equações diferenciais ordinárias não lineares que descreve o seu comportamento. A principal contribuição deste trabalho foi a verificação da eficiência do algoritmo genético na aproximação da solução da equação diferencial (equação de Duffing). Pôde-se perceber que a função de aptidão desempenha um papel fundamental na eficiência do método: a função de aptidão utilizada.

$$\begin{cases} f(1) = x(2); \\ f(2) = F \cdot \cos(\text{Frequência da Força} \cdot t) - w_0^2 \cdot x(1) + \alpha \cdot x(1)^3 - c \cdot x(2); \end{cases}$$

Não foi uma boa função, uma vez que permitiu grandes diferenças entre a solução do GA e a solução numérica da equação diferencial, fazendo com que a solução do GA diferísse substancialmente da solução do Runge-Kutta, conforme pode ser observado nos gráficos da Figura 6. Como perspectivas para trabalhos futuros, serão desenvolvidas alternativas mais eficientes para a função de aptidão.

## REFERÊNCIAS

- FRIEDLAND, B. **Control system design an introduction to state-space methods**. Mineola: Dover Publications, 2005.
- LINDEN, R. **Algoritmos genéticos**. 2. ed. Brasport, 2012. 428 p.
- QUEIROZ, T. A.; SANTEE, D. M. **Aperfeiçoamento de métodos numéricos com aplicação comparativa e estudo de caso para o problema da coluna parcialmente enterrada sujeita ao caos**. Porto: CMNE, 2007.
- SAMPAIO, A. E. G. **Análise do comportamento dinâmico de colunas semi-enterradas**. 2004. Dissertação (Mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2004.
- TONINI, A. M.; SCHETTINO, D. N. **Matlab para Engenharia**. 1. ed. Belo Horizonte, 2002. Disponível em: <[http://www.netsoft.inf.br/aulas/7\\_EAC\\_Sistemas\\_Realimentados/6\\_MatLab\\_para\\_Engenharia.pdf](http://www.netsoft.inf.br/aulas/7_EAC_Sistemas_Realimentados/6_MatLab_para_Engenharia.pdf)>. Acesso em: 20 set. 2014.

