Métodos Computacionais para Análise Estrutural: Aplicações em Python e Octave

Paulo de Souza Silva

2022-02-15

Sumário

Motivação

Este livro tem como objetivo ser um material auxiliar para os alunos da disciplina de **Métodos Computacionais para Análise Estrutural** do curso de **Engenharia Aeroespacial** da **Universidade Federal do ABC**. No entanto, o mesmo, tenta abordar de forma clara e concisa os temas referentes a *Álgebra Linear, Cálculo Numério* e *Modelagens Estruturais* para todos aqueles que são intusiastas nesses tópicos.

Por que ler este livro?

Sobre os Autores

Paulo de Souza é graduando do curso de Engenharia Aeroespacial. Seus interesses são nas áreas de simulação estrutural de aeronaves e aprendizado de máquina com aplicação em frentes da engenharia mecânica.

Parte I

Conhecendo o Octave e o Python

Capítulo 1

Introdução ao Octave

O GNU Octave, ou simplesmente Octave, é um *software* livre para execução e desenvolvimento de operações matemáticas.

1.1 Comandos Básicos em Octave

1.1.1 Operações Matemáticas e Lógicas

As operações matemáticas simples, adição, subtração, multiplicação e divisão podem ser executadas de maneira direta, como é apresentado na sequência

```
2 + 5
2 - 5
2 * 5
2 / 5
```

```
## ans = 7
## ans = -3
## ans = 10
## ans = 0.4000
```

1.1.2 Vetores e Matrizes

Esta é uma forma de montar uma matriz 3×3 em **Octave**

```
A = [1 2 3;4 5 6;7 8 9]
```

```
## A =
##
##    1   2   3
##    4   5   6
##    7   8   9
```

Existem alguns comandos que automatizam a criação de algumas matrizes; em algumas ocasiões serão necessárias o uso de **matrizes de zeros** ou da **matriz identidade**, os comandos para elaborar ambas são respectivamente:

```
A = zeros(3,3) %matriz de zeros de dimensao 3 por 3
```

```
## A =
##
##    0   0   0
##    0   0   0
##    0   0   0
```

```
A = eye(3,3) %matriz identidade de dimensao 3 por 3
```

```
## A =
##
## Diagonal Matrix
##
##    1   0   0
##    0   1   0
##    0   0   1
```

1.1.3 Gráficos

Para gerar o gráficos bidimensionais, define-se os valores do eixo X e depois aplica-se a equação desejada; por exemplo, para $y = x^2 | \ x \in [-5, 5]$ pode ser feito

```
x = -5:0.1:5;
y = x.^2;

plot(x,y,'r-')
grid on
print -djpg parabola.jpg
```

OBS: O comando `print -djpg parabola.jpg` não é necessário no Octave, os autores o deixaram no corpo do código apenas para salvar a imagem e assim aplica-lá ao livro.

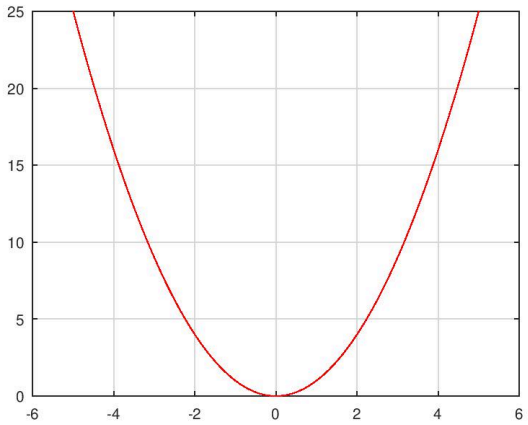

Figura 1.1: Parábola definida de -5 à 5

1.2 Programação em Octave

1.2.1 Estruturas de Repetição

1.2.2 Funções *function*

```
function I = InerciaRet(h,b)
% Momento de Inercia de uma barra de seção retangular
% h - altura da seção
% b - base da seção

I = h^3*b/12
```

É possível então chamar a função pela *Janela de Comandos* ou de outro *script*

```
altura = 2;
base = 3;
I = InericaRet(altura,base)

## I =
##
##   2
```

1.2.3 Leitura de documentos externos

1.2.4 Bibliotecas

Existem diversas bibliotecas extras no Octave, aqui serão apresentadas apenas duas que por ventura são utilizadas ao decorrer do livro, para mais detalhes consulte...

Capítulo 2

Introdução ao Python

2.1 Comandos Básicos em Python

Esta é uma forma de montar uma matriz 3×3 em **Python**

```
#Matriz 3x3
import numpy as np

M = np.array([[1,2,3],
              [4,5,6],
              [7,8,9]])
print(M)
```

```
## [[1 2 3]
##  [4 5 6]
##  [7 8 9]]
```

Parte II

Resolução de Sistemas e Aproximação de Funções

Capítulo 3

Sistemas de Equações

3.1 Equações Não-Lineares

3.1.1 Método de Newton

3.1.2 Método das Secantes

3.1.3 Método de Regula Falsi

3.1.4 Sistemas de Equações não lineares

3.1.5 Equações Polinomiais

3.2 Solução de Equações Lineares: Métodos Exatos

3.2.1 Decomposição LU

3.2.2 Eliminação de Gauss

3.2.3 Decomposição de Cholesky

Se uma matriz **A** for **simétrica** e **positiva definida**, então A pode ser decomposta da forma:

$$A = GG^T$$

- **Matriz Simétrica**

  Uma matriz simétrica é aquela no qual suas componentes $a_{i,j}$ tem valores iguais as componentes $a_{j,i}$, isto é:

  $$a_{i,j} = a_{j,i}$$

- **Matriz Positiva Definida**

Uma matriz é dita positiva definida se ocorre um dos seguintes fatos:
1. Os autovalores de A são todos positivos
2. Os menores principais são positivos
3. $v^T Av > 0, \forall v \neq 0$

Abaixo um o teorema 1 define o método para execução de Cholesky

Teorema 1 *Se uma matriz* ***A*** *for* ***simétrica*** *e* ***positiva definida****, então* existe uma única *matriz triangular G, com elementos diagonais positivos tal que* $A = GG^T$.

Isto é, para um caso 3×3 é possível representar

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} g_{11} & 0 & 0 \\ g_{21} & g_{22} & 0 \\ g_{31} & g_{32} & g_{33} \end{pmatrix} \begin{pmatrix} g_{11} & g_{21} & g_{31} \\ 0 & g_{22} & g_{32} \\ 0 & 0 & g_{33} \end{pmatrix}$$

Para encontrar os coeficientes da matriz G, é então feito os seguintes processos

$$g_{11} = \sqrt{a_{11}} \tag{i}$$

$$g_{i1} = \frac{a_{i1}}{g_{11}} \tag{ii}$$

$$g_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} g_{ik}^2 \right)^{1/2} \tag{iii}$$

$$g_{ij} = \frac{\left(a_{ij} - \sum_{k=1}^{j-1} g_{ik} g_{jk} \right)}{g_{jj}} \tag{iv}$$

Os passos (iii) e (iv) devem ser intercalados, vista a depedência dos fatores de uma equação na outra, desta forma é indicado o cálculo do valor da diagonal principal e os valores abaixo deste.

Por exemplo, após calcular a componente g_{22} calcular as demais componentes da coluna 2 (pelo passo iv), e somente ao término deste, ir para a componente g_{33}.

3.2.3.1 Função de Cholesky (Python)

```
import numpy as np

#Funcao que verifica a simetria
def simetrica(U):
  for i in range(len(U)):
    for j  in range(len(U[0])):
      if U[i][j] != U[j][i]:
        return 0
  return 1

#Funcao que verifica se a matriz é positiva definida
def pos_def(U):
  auval,auvec = np.linalg.eig(U)
  cont = 0;
  for i in range(len(auval)):
    if auval[i]>0:
```

```
      cont = cont+1
  if cont == len(U):
    return 1
  return 0

#Funcao que faz a soma para os termos G[i][i]
def SomaCho1(U,i):
  soma1 = 0
  for k in range(0,i):
    soma1 = soma1 + np.power(U[i][k],2)
  return soma1

#Funcao que faz a soma para os termos G[i][j]
def SomaCho2(U,i,j):
  soma2 = 0
  for k in range(0,j):
    soma2 = soma2 + U[i][k]*U[j][k]
  return soma2

#Funcao para o Calculo da Decomposicao de Cholesky
def MeuCholesly(A):
  G = np.zeros((len(A),len(A)))
  if simetrica(A) == pos_def(A):
    G[0][0] = np.sqrt(A[0][0])
    for i in range(1,len(A)):
      G[i][0] = A[i][0]/G[0][0]

    for j in range(1,len(A)):
      for i in range(j,len(A)):
        if i==j:
          G[i][j] = np.sqrt(A[i][i] - SomaCho1(G,i))
          #print(SomaCho1(G,i))
        else:
          G[i][j] = (A[i][j] - SomaCho2(G,i,j))/G[j][j]
          #print(SomaCho2(G,i,j))
    G = np.round(G,4)
    return G
  else:
    return print('Não é possível usar Decomposição de Cholesky')
```

Exemplo matriz 3×3

```
A = np.array([[6,15,55],
              [15,55,225],
              [55,225,979]])

MeuCholesly(A)
```

```
## array([[ 2.4495,  0.    ,  0.    ],
##        [ 6.1237,  4.1833,  0.    ],
##        [22.4537, 20.9165,  6.1101]])
```

Exemplo matriz 4×4

```
M = np.array([[9,0,-27,18],
              [0,9,-9,-27],
              [-27,-9,99,-27],
              [18,-27,-27,121]])

MeuCholesly(M)
```

```
## array([[ 3.,  0.,  0.,  0.],
##        [ 0.,  3.,  0.,  0.],
##        [-9., -3.,  3.,  0.],
##        [ 6., -9.,  0.,  2.]])
```

3.2.3.2 Comando linalg.Cholesky

3.3 Solução de Equações Lineares: Métodos Iterativos

3.3.1 Método de Gauss Seidel

Capítulo 4

Footnotes and citations

4.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one [1].

4.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [Xie, 2021] (check out the last code chunk in index.Rmd to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [Xie, 2015] (this citation was added manually in an external file book.bib). Note that the `.bib` files need to be listed in the index.Rmd with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: https://rstudio.github.io/visual-markdown-editing/#/citations

[1] This is a footnote.

Capítulo 5

Blocks

5.1 Equações

Aqui está uma equação.

$$f\left(k\right) = \binom{n}{k} p^k \left(1-p\right)^{n-k} \tag{5.1}$$

You may refer to using `\@ref(eq:binom)`, like see Equation (5.1).

5.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref(thm:tri)`, for example, check out this smart theorem 2.

Teorema 2 *For a right triangle, if c denotes the* length *of the hypotenuse and a and b denote the lengths of the **other** two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html.

5.3 Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html

Capítulo 6

Sharing your book

6.1 Publishing

HTML books can be published online, see: https://bookdown.org/yihui/bookdown/publishing.html

6.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

6.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book- all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

https://pkgs.rstudio.com/bookdown/reference/gitbook.html

Or use:

```
?bookdown::gitbook
```

Referências Bibliográficas

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL http://yihui.org/knitr/. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2021. URL https://CRAN.R-project.org/package=bookdown. R package version 0.24.