



UNIVERSIDADE FEDERAL DE UBERLÂNDIA



## FICHA DE COMPONENTE CURRICULAR

## COURSE OUTLINE

<b>CÓDIGO / COURSE CODE :</b>		<b>COMPONENTE CURRICULAR / COURSE TITLE :</b> Métodos Computacionais / Computational Methods		
<b>UNIDADE ACADÊMICA OFERTANTE / ORGANIZATION :</b> Faculdade de Engenharia Elétrica - Programa de Pós-Graduação em Engenharia Elétrica Faculty of Electrical Engineering - Postgraduate Program in Electrical Engineering				<b>SIGLA / ACRONYM :</b> FEELT - PPGEELT
<b>CH TOTAL TEÓRICA / LECTURE HOURS :</b> 45 horas / hours	<b>CH TOTAL PRÁTICA / LABORATORY HOURS :</b> 0 horas / hours	<b>CH TOTAL / TOTAL HOURS :</b> 45 horas / hours	<b>CRÉDITOS / CREDITS :</b> 3	<b>TIPO / TYPE:</b> Optativa / Elective
<b>Curso / Degree :</b> Mestrado e Doutorado / Master and PhD		<b>Requisito / Requirement :</b> Sem requisitos / No Requirements		

## 1. OBJETIVOS / STUDY GOALS

O ensino de métodos computacionais para estudantes de Engenharia é fundamental, pois fornece uma base sólida para a resolução de problemas complexos e a análise de sistemas do mundo real. Os objetivos de ensinar métodos numéricos incluem: compreensão dos fundamentos matemáticos, resolução de problemas complexos, modelagem e simulação, aplicação em engenharia, uso eficiente de software de engenharia, avaliação de erros e incertezas, desenvolvimento de algoritmos, preparação para pesquisa e desenvolvimento, integração com a teoria, e o desenvolvimento de habilidades computacionais. Em resumo, o ensino de métodos computacionais na Engenharia visa equipar os estudantes com as habilidades necessárias para abordar problemas práticos, desenvolver soluções inovadoras e aplicar ferramentas computacionais avançadas em suas carreiras profissionais.

The teaching of computational methods to engineering students is crucial, as it provides a solid foundation for solving complex problems and analyzing real-world systems. The objectives of teaching numerical methods include understanding mathematical fundamentals, solving complex problems, modeling and simulation, application in engineering, efficient use of engineering software, evaluation of errors and uncertainties, algorithm development, preparation for research and development, integration with theory, and the development of computational skills. In summary, the teaching of computational methods in engineering aims to equip students with the necessary skills to address practical problems, develop innovative solutions, and apply advanced computational tools in their professional careers.

## 2. EMENTA / COURSE CONTENTS

Introdução à Computação. Equações Algébricas Lineares. Equações Não-lineares. Equações com Autovalores. Interpolação e Ajuste de Curvas. Integração Numérica. Solução Numérica de Equações Diferenciais Simples. Introdução à Equações Diferenciais Parciais.

Introduction to Programming. Linear Algebraic Equations. Nonlinear Equations. Equations with Eigenvalues. Interpolation and Curve Fitting. Numerical Integration. Numerical Solution of Simple Differential Equations. Introduction to Partial Differential Equations.

## 3. PROGRAMA / PROGRAM

**1. Introdução à Programação.**

- 1.1. Introdução.
- 1.2. Software e hardware.
- 1.3. Sub-rotinas matemáticas e funções.
- 1.4. Erros.
- 1.5. Conclusões.

**2. Equações Algébricas Lineares.**

- 2.1. Introdução.
- 2.2. Eliminação Gaussiana.
- 2.3. Solução de equações utilizando fatoração LU.

- 2.4. Equações com matriz de coeficientes simétricos.
  - 2.4.1. Formas quadráticas e definidas positivas.
  - 2.4.2. Método de *Cholesky*.
- 2.5. Equações em banda.
- 2.6. Armazenamento compacto para largura de banda variáveis.
- 2.7. Pivoteamento.
  - 2.7.1. Mau condicionamento.
- 2.8. Equações com variáveis prescritas.
- 2.9. Métodos iterativos.
  - 2.9.1. O processo iterativo.
  - 2.9.2. Sistemas muito esparsos.
  - 2.9.3. O método de *Gauss-Seidel*.
  - 2.9.4. Sobre relaxação sucessiva.
- 2.10. Métodos do Gradiente.
  - 2.10.1. O método da "descida mais rápida".
  - 2.10.2. O método do "gradiente conjugado".
  - 2.10.3. Convergência de métodos iterativos.
  - 2.10.4. Pré-condicionamento.
- 2.11. Comparação entre métodos diretos e iterativos.

### **3. Equações Não-lineares.**

- 3.1. Introdução.
- 3.2. Substituição iterativa.
- 3.3. Raízes múltiplas e outras dificuldades.
- 3.4. Métodos de Interpolação.
  - 3.4.1. Método da bissecção.
  - 3.4.2. Método da posição falsa.
- 3.5. Métodos de extrapolação.
  - 3.5.1. Método de *Newton-Raphson*.
  - 3.5.2. O método de *Newton-Raphson* modificado.
- 3.6. Aceleração de convergência.
- 3.7. Sistemas de equações não-lineares.
  - 3.7.1. Substituição iterativa para sistemas.
  - 3.7.2. *Newton-Raphson* para sistemas.
  - 3.7.3. Método de *Newton-Raphson* modificado para sistemas.

### **4. Equações com Autovalores.**

- 4.1. Introdução.
- 4.2. Ortogonalidade de Normalização de Autovetores.
- 4.3. Métodos de solução para equações com autovalores.
- 4.4. Iteração vetorial.
  - 4.4.1. Iteração deslocada.
  - 4.4.2. Iteração deslocada inversa.
- 4.5. Cálculo de autovalores intermediários – deflação.
- 4.6. O problema genérico de autovalores  $Ax = \lambda Bx$ .
  - 4.6.1. Conversão do problema genérico para a forma padrão.
- 4.7. Métodos de Transformação.
  - 4.7.1. Diagonalização de Jacobi.
  - 4.7.2. Transformação de *Householder* para a forma tridiagonal.
  - 4.7.3. Transformação LR para autovalores de matrizes tridiagonais.

4.7.4. Redução de Lanczos para a forma tridiagonal.

4.8. Método do polinômio característico.

4.8.1. Calculando determinantes de matrizes tridiagonais.

4.8.2. A propriedade da sequência de *Sturm*.

## **5. Interpolação e Ajuste de Curvas.**

5.1. Introdução.

5.2. Polinômios interpoladores.

5.2.1. Polinômios de Lagrange.

5.2.2. Métodos de Diferenças.

5.2.3. Funções *Spline*.

5.3. Diferenciação numérica.

5.3.1. Erros nas fórmulas de diferenças.

5.4. Ajuste de Curvas.

5.4.1. Mínimos quadrados.

## **6. Integração Numérica.**

6.1. Introdução.

6.2. Regras de *Newton-Cotes*.

6.2.1. Introdução.

6.2.2. Regra do ponto único (regra do retângulo,  $n=1$ ).

6.2.3. Regra dos dois pontos (regra do trapézio,  $n=2$ ).

6.2.4. Regra dos três pontos (regra de Simpson,  $n=3$ ).

6.2.5. Regra de *Newton-Cotes* de Ordens Mais elevadas ( $n>3$ ).

6.2.6. Precisão das regras de *Newton-Cotes*.

6.2.6.1. Erro na regra do retângulo.

6.2.6.2. Erro na regra do trapézio.

6.2.6.3 Erro na regra de *Simpson*.

6.2.7. Resumo das Regras de *Newton-Cotes*.

6.2.8. Regras com Repetição.

6.2.8.1 Regra do trapézio repetida.

6.2.8.2 Regra de Simpson repetida.

6.2.9. Observações sobre as regras de *Newton-Cotes*.

6.3. Regras de Gauss-Legendre.

6.3.1. Introdução.

6.3.2. Regra do ponto único (regra do ponto médio,  $n=1$ ).

6.3.3. Regra de dois pontos.

6.3.4. Regra de três pontos ( $n=3$ ).

6.3.5. Mudando os limites de integração.

6.3.6. Precisão das regras de *Gauss-Legendre*.

6.3.7. Resumo das regras de *Gauss-Legendre*.

6.4. Regras de integração especiais.

6.4.1. Introdução.

6.4.2. Regras de *Gauss-Laguerre*.

6.4.3. Regras de *Gauss-Chebyshev*.

6.4.4. Coeficientes de ponderação fixos.

6.4.5. Métodos híbridos.

6.4.6. Pontos amostrados fora do intervalo de integração.

6.5. Integrais múltiplas.

6.5.1. Introdução.

6.5.2. Integração sobre uma área retangular.

6.5.3. Integração sobre a área de um quadrilátero genérico.

## **7. Solução Numérica de Equações Diferenciais Simples.**

7.1. Introdução.

7.2. Definições e tipos de equações diferenciais.

7.3. Problemas de valor inicial.

7.3.1. Métodos de passo único.

7.3.1.1. Método de *Euler*.

7.3.1.2. Método de *Euler* modificado.

7.3.1.3. Método do ponto médio.

7.3.1.4. Método de *Runge-Kutta*.

7.3.1.5. Precisão dos métodos de passo único.

7.3.1.6. Redução de equações de ordem elevada.

7.3.1.7. Solução de equações de primeira ordem simultâneas.

7.3.1.8. Métodos *theta* para equações lineares.

7.3.2. Métodos preditores-corretores.

7.3.2.1. Método *Euler-Trapezoidal*.

7.3.2.2. Método de *Milne*.

7.3.2.3. Método *Adams-Moulton-Bashforth*.

7.3.2.4. Precisão dos métodos preditores-corretores.

7.3.3. Equações rígidas.

7.3.4. Propagação do erro – estabilidade numérica.

7.3.5. Observações conclusivas sobre problemas de valor inicial.

7.4. Problemas de valor de contorno.

7.4.1. Métodos de diferenças finitas.

7.4.2. Métodos do chute.

7.4.3. Métodos dos resíduos ponderados.

7.4.3.1. Colocação.

7.4.3.2. Subdomínio.

7.4.3.3. Mínimos quadrados.

7.4.3.4. Método de *Galerkin*.

7.4.3.5. Observações conclusivas sobre procedimentos dos resíduos ponderados.

## **8. Introdução à Equações Diferenciais Parciais.**

8.1. Introdução.

8.2. Algumas definições.

8.3. Equações de primeira ordem.

8.4. Equações de segunda ordem.

8.5. Diferenças finitas em duas dimensões.

8.6. Sistemas elípticos.

8.7. Sistemas parabólicos.

8.7.1. Estabilidade das diferenças finitas.

8.8. Sistemas hiperbólicos.

## **1. Introduction to Programming.**

1.1. Introduction.

1.2. Software and hardware.

1.3. Mathematical subroutines and functions.

1.4. Errors.

1.5. Conclusions.

## **2. Linear Algebraic Equations.**

2.1. Introduction.

2.2. Gaussian Elimination.

2.3. Solution of equations using LU factorization.

2.4. Equations with symmetric coefficient matrices.

2.4.1. Quadratic forms and positive definite matrices.

2.4.2. Cholesky's method.

2.5. Banded equations.

2.6. Compact storage for variable bandwidth.

2.7. Pivoting.

2.7.1. Ill-conditioning.

2.8. Equations with prescribed variables.

2.9. Iterative methods.

2.9.1. The iterative process.

2.9.2. Very sparse systems.

2.9.3. Gauss-Seidel method.

2.9.4. Successive over-relaxation.

2.10. Gradient methods.

2.10.1. The "steepest descent" method.

2.10.2. The "conjugate gradient" method.

2.10.3. Convergence of iterative methods.

2.10.4. Preconditioning.

2.11. Comparison between direct and iterative methods.

## **3. Nonlinear Equations.**

3.1. Introduction.

3.2. Iterative substitution.

3.3. Multiple roots and other difficulties.

3.4. Interpolation methods.

3.4.1. Bisection method.

3.4.2. False position method.

3.5. Extrapolation methods.

3.5.1. Newton-Raphson method.

3.5.2. Modified Newton-Raphson method.

3.6. Acceleration of convergence.

3.7. Systems of nonlinear equations.

3.7.1. Iterative substitution for systems.

3.7.2. Newton-Raphson for systems.

3.7.3. Modified Newton-Raphson for systems.

## **4. Equations with Eigenvalues.**

4.1. Introduction.

4.2. Orthogonality and Normalization of Eigenvectors.

4.3. Solution methods for equations with eigenvalues.

4.4. Vector iteration.

4.4.1. Shifted iteration.

4.4.2. Inverse shifted iteration.

4.5. Calculation of intermediate eigenvalues – deflation.

4.6. The generic eigenvalue problem  $Ax = \lambda Bx$ .

4.6.1. Conversion of the generic problem to standard form.

4.7. Transformation methods.

4.7.1. Jacobi diagonalization.

4.7.2. Householder transformation to tridiagonal form.

4.7.3. LR transformation for eigenvalues of tridiagonal matrices.

4.7.4. Lanczos reduction to tridiagonal form.

4.8. Characteristic polynomial method.

4.8.1. Calculating determinants of tridiagonal matrices.

4.8.2. The Sturm sequence property.

## **5. Interpolation and Curve Fitting.**

5.1. Introduction.

5.2. Interpolating polynomials.

5.2.1. Lagrange polynomials.

5.2.2. Difference methods.

5.2.3. Spline functions.

5.3. Numerical differentiation.

5.3.1. Errors in difference formulas.

5.4. Curve fitting.

5.4.1. Least squares.

## **6. Numerical Integration.**

6.1. Introduction.

6.2. Newton-Cotes rules.

6.2.1. Introduction.

6.2.2. Single point rule (rectangle rule,  $n=1$ ).

6.2.3. Two-point rule (trapezoidal rule,  $n=2$ ).

6.2.4. Three-point rule (Simpson's rule,  $n=3$ ).

6.2.5. Higher Order Newton-Cotes rules ( $n>3$ ).

6.2.6. Accuracy of Newton-Cotes rules.

6.2.6.1. Error in rectangle rule.

6.2.6.2. Error in trapezoidal rule.

6.2.6.3. Error in Simpson's rule.

6.2.7. Summary of Newton-Cotes rules.

6.2.8. Repeated rules.

6.2.8.1 Repeated trapezoidal rule.

6.2.8.2 Repeated Simpson's rule.

6.2.9. Observations on Newton-Cotes rules.

6.3. Gauss-Legendre rules.

6.3.1. Introduction.

6.3.2. Single point rule (midpoint rule,  $n=1$ ).

6.3.3. Two-point rule.

6.3.4. Three-point rule ( $n=3$ ).

6.3.5. Changing integration limits.

6.3.6. Accuracy of Gauss-Legendre rules.

6.3.7. Summary of Gauss-Legendre rules.

6.4. Special integration rules.

6.4.1. Introduction.

6.4.2. Gauss-Laguerre rules.

6.4.3. Gauss-Chebyshev rules.

- 6.4.4. Fixed weighting coefficients.
- 6.4.5. Hybrid methods.
- 6.4.6. Sampling points outside the integration interval

## 6.5. Multiple integrals.

- 6.5.1. Introduction.
- 6.5.2. Integration over a rectangular area.
- 6.5.3. Integration over the area of a generic quadrilateral.

## **7. Numerical Solution of Simple Differential Equations.**

### 7.1. Introduction.

### 7.2. Definitions and types of differential equations.

### 7.3. Initial value problems.

#### 7.3.1. Single-step methods.

- 7.3.1.1. Euler's method.
- 7.3.1.2. Modified Euler's method.
- 7.3.1.3. Midpoint method.
- 7.3.1.4. Runge-Kutta methods.
- 7.3.1.5. Accuracy of single-step methods.
- 7.3.1.6. Reduction of higher-order equations.
- 7.3.1.7. Solution of simultaneous first-order equations.
- 7.3.1.8. Theta methods for linear equations.

#### 7.3.2. Predictor-corrector methods.

- 7.3.2.1. Euler-Trapezoidal method.
- 7.3.2.2. Milne's method.
- 7.3.2.3. Adams-Moulton-Bashforth method.
- 7.3.2.4. Accuracy of predictor-corrector methods.

#### 7.3.3. Stiff equations.

#### 7.3.4. Error propagation – numerical stability.

#### 7.3.5. Concluding remarks on initial value problems.

### 7.4. Boundary value problems.

#### 7.4.1. Finite difference methods.

#### 7.4.2. Shoot methods.

#### 7.4.3. Weighted residual methods.

- 7.4.3.1. Collocation.
- 7.4.3.2. Subdomain.
- 7.4.3.3. Least squares.
- 7.4.3.4. Galerkin's method
- 7.4.3.5. Concluding remarks on weighted residual procedures.

## **8. Introduction to Partial Differential Equations.**

### 8.1. Introduction.

### 8.2. Some definitions.

### 8.3. First-order equations.

### 8.4. Second-order equations.

### 8.5. Two-dimensional finite differences.

### 8.6. Elliptic systems.

### 8.7. Parabolic systems.

#### 8.7.1. Stability of finite differences.

### 8.8. Hyperbolic systems.

**4. BIBLIOGRAFIA BÁSICA / TEXT BOOK**

1. GRIFFITHS, D. V.; SMITH, I. M. **Numerical Methods for Engineers**. 2ª edição [S.l.]: Chapman and Hall/CRC, 2006. 492 p. ISBN 978-1584884019.
2. PRESS, W. H. *et al.* **Numerical Recipes: The Art of Scientific Computing**. 3ª edição. Reino Unido: Cambridge University Press, 2007. 1235 p. ISBN 978-0521880688.
3. CHAPRA, S. C.; CANALE, R. P. **Numerical Methods for Engineers**. 8ª edição [S.l.]: McGraw Hill, 2021.

1. GRIFFITHS, D. V.; SMITH, I. M. **Numerical Methods for Engineers**. 2ª edição [S.l.]: Chapman and Hall/CRC, 2006. 492 p. ISBN 978-1584884019.
2. PRESS, W. H. *et al.* **Numerical Recipes: The Art of Scientific Computing**. 3ª edição. Reino Unido: Cambridge University Press, 2007. 1235 p. ISBN 978-0521880688.
3. CHAPRA, S. C.; CANALE, R. P. **Numerical Methods for Engineers**. 8ª edição [S.l.]: McGraw Hill, 2021.

**5. BIBLIOGRAFIA COMPLEMENTAR / ADDITIONAL READING**

1. HSU, T. **Applied Engineering Analysis**. 1ª edição [S.l.]: Wiley, 2018. 528 p. ISBN 978-1119071204.
2. CONTE, S. D.; DE BOOR, C. **Elementary Numerical Analysis: an Algorithmic Approach**. 3ª edição [S.l.]: McGraw-Hill Book Company, 1980. 408 p. ISBN 978-0070662285.
3. ISSACSON, E.; KELLER, H. B. **Analysis of Numerical Methods**. Edição revisada [S.l.]: Dover Publications Inc., 1994. 576 p. ISBN 978-048668093.
4. HAMMING, R. W. **Numerical Methods for Scientists and Engineers**. 2ª Edição Revisada [S.l.]: Dover Publications Inc., 1987. 752 p. ISBN 978-0486652412.
5. HOFFMAN, J. D.; FRANKEL, S. **Numerical Methods for Engineers and Scientists**. 2ª Edição Revisada e Expandida [S.l.]: CRC Press, 2001. 838 p. ISBN 978-0824704438.
6. DECHAUMPHAI, P.; WANSOPHARK, N., **Numerical Methods in Science and Engineering: Theories with Matlab, Mathematica, Fortran, C and Python Programs**. 1ª Edição [S.l.]: Alpha Science International Ltd., 2022.

1. HSU, T. **Applied Engineering Analysis**. 1ª edição [S.l.]: Wiley, 2018. 528 p. ISBN 978-1119071204.
2. CONTE, S. D.; DE BOOR, C. **Elementary Numerical Analysis: an Algorithmic Approach**. 3ª edição [S.l.]: McGraw-Hill Book Company, 1980. 408 p. ISBN 978-0070662285.
3. ISSACSON, E.; KELLER, H. B. **Analysis of Numerical Methods**. Edição revisada [S.l.]: Dover Publications Inc., 1994. 576 p. ISBN 978-048668093.
4. HAMMING, R. W. **Numerical Methods for Scientists and Engineers**. 2ª Edição Revisada [S.l.]: Dover Publications Inc., 1987. 752 p. ISBN 978-0486652412.
5. HOFFMAN, J. D.; FRANKEL, S. **Numerical Methods for Engineers and Scientists**. 2ª Edição Revisada e Expandida [S.l.]: CRC Press, 2001. 838 p. ISBN 978-0824704438.
6. DECHAUMPHAI, P.; WANSOPHARK, N., **Numerical Methods in Science and Engineering: Theories with Matlab, Mathematica, Fortran, C and Python Programs**. 1ª Edição [S.l.]: Alpha Science International Ltd., 2022.

**6. APROVAÇÃO / APPROVAL**

Ficha de Disciplina homologada na 366ª Reunião Ordinária do Colegiado do Programa de Pós-Graduação em Engenharia Elétrica.  
Curricular Component approved at 366th Regular Board Meeting of the Postgraduate Program in Electrical Engineering.

PROF. DR. LUIZ CARLOS GOMES DE FREITAS  
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica  
Head of the Postgraduate Program in Electrical Engineering  
Portaria de Pessoal UFU Nº 3675, de 30 de Junho de 2023

PROF. DR. SÉRGIO FERREIRA DE PAULA SILVA  
Diretor da Faculdade de Engenharia Elétrica  
Director of the Faculty of Electrical Engineering  
Portaria de Pessoal UFU Nº 1225, de 31 de Março de 2021



Documento assinado eletronicamente por **Luiz Carlos Gomes de Freitas, Coordenador(a)**, em 02/02/2024, às 13:43, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [https://www.sei.ufu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **5153671** e o código CRC **DB1E6E7E**.



