

1. Problema computacional

- a. Entradas
- b. Saídas
- c. Especificação (como as entradas estão relacionadas com a saída)
 - i. “Promessa” do desenvolvedor
 - ii. O que o usuário quer

2. Problema: Achar um valor aproximado de uma função com informações de derivadas

- a. Entradas: um valor x , um valor a , “informações das derivadas em a ”, M (maior valor da n -derivada no intervalo)
- b. Saída: um valor y e um valor E
- c. Especificação: $|y - f(x)| \leq E$
- d. Algoritmos:
 - i. Taylor de ordem 0
 - ii. Taylor de ordem 1
 - iii. Taylor de ordem 2
 - iv. Taylor de ordem n

3. Problema: Encontrar raiz aproximadamente (Resolver equações não-lineares) (uma variável e uma equação)

- a. Entrada: uma função f , erro, um intervalo
- b. Saída: um número real x ou aviso que não teve troca de sinal no intervalo
- c. Especificação: $f(r)=0$ e $|x-r| \leq \text{erro}$
- d. Algoritmos
 - i. Bisseção
 - ii. Newton (não fez análise de erro) (não precisa se preocupar com o erro)

4. Problema: Interpolação Polinomial

- a. Entrada: n pontos: $(x_0, y_0) \dots (x_n, y_n)$
- b. Saída: um polinômio $F(x)$ com grau no máximo n
- c. Especificação: Para todo $1 \leq i \leq n$, $F(x_i) = y_i$
- d. Algoritmos
 - i. Vandermonde (retorna os coeficientes ou *retorna a função $F(x)$*)
 - ii. Polinômio de Lagrange (retorna a função $F(x)$)

5. Problema: Regressão Polinomial (“Interpolação aproximada”)

- a. Entrada: n pontos: $(x_0, y_0) \dots (x_n, y_n)$ e um natural k
- b. Saída: um polinômio $F(x)$ com grau no máximo k
- c. Especificação: Para todo $1 \leq i \leq n$, $F(x_i)$ **aproximadamente** y_i
- d. Algoritmo
 - i. Vandermonde + Mínimos quadrados

6. Problema: Regressão com coeficiente lineares (generalização do problema anterior)

- a. Entrada: n pontos: $(x_0, y_0) \dots (x_n, y_n)$, e k funções $f_1(x), f_2(x), \dots, f_k(x)$ (um vetor de funções)
- b. Saída: uma função $F(x) = c_1 f_1(x) + \dots + c_k f_k(x)$ ou coeficientes
- c. Especificação: Para todo $1 \leq i \leq n$, $F(x_i)$ **aproximadamente** y_i
- d. Algoritmo
 - i. Vandermonde (Matriz $n+1$ por k) + Mínimos quadrados

7. Problema: Regressão com coeficientes não lineares. Regressão com o modelo exponencial, potência e geométrica (mencionado na aula 24) (uma função para cada modelo)

- a. Entrada: n pontos: $(x_0, y_0) \dots (x_n, y_n)$
- b. Saída: modelo escolhido com o coeficientes ou coeficientes
- c. Especificação: Para todo $1 \leq i \leq n$, $F(x_i)$ **aproximadamente** y_i
- d. Algoritmo
 - i. Troca de variável (dicionário ou linearização) + Vandermonde + Mínimos quadrados + troca de variável

8. Problema: Interpolação 2D

- a. Entrada: $x_0, x_1, y_0, y_1, z_0, z_1, z_2, z_3$
- b. Saída: um polinômio de grau 2 de duas variáveis
- c. Especificação: Para todo $1 \leq i \leq n$, $F(x_i, y_i) = z_i$
- d. Algoritmo
 - i. Lagrange 2D

9. Calcular a norma de um vetor v

- a. Entrada: um vetor v e um vetor
- b. Saída: um número z
- c. Especificação $z = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2}$

10. Problema: Resolver um sistema linear denso aproximadamente (Mínimos Quadrados) (resolver n equações lineares com n variáveis)

- a. Entrada: Uma matriz densa $n \times n$ A e um vetor b
- b. Saída: Um vetor $x^* n \times 1$
- c. Especificação: Ax^* aproximadamente b ($x^* = \operatorname{argmin} \|Ax - b\|$)
- d. Algoritmo: Resolve o sistema $A^t Ax = A^t b$

11. Problema: Resolver exatamente uma sistema linear denso

- a. Entrada: Uma matriz densa $n \times n$ A e um vetor b
- b. Saída: Um vetor $x n \times 1$
- c. Especificação: $Ax = b$
- d. Algoritmo: Resolver com LU $O(n^3)$

12. Problema: Resolver um sistema triangular superior

- a. Entrada: Uma matriz triangular superior $n \times n$ U e um vetor b
- b. Saída: Um vetor $x n \times 1$
- c. Especificação: $Ux = b$

- d. Algoritmo: Substituição Reversa $O(n^2)$

13. Problema: Resolver um sistema triangular inferior

- a. Entrada: Uma matriz triangular inferior $n \times n$ L e um vetor b
- b. Saída: Um vetor x $n \times 1$
- c. Especificação: $Lx=b$
- d. Algoritmo: Substituição Direta $O(n^2)$

14. Problema: Resolver um sistema diagonal

- a. Entrada: Uma matriz triangular diagonal $n \times n$ D e um vetor b
- b. Saída: Um vetor x $n \times 1$
- c. Especificação: $Dx=b$
- d. Algoritmo: Só fazer n divisões $O(n)$

15. Problema: achar a inversa de uma matriz

- a. Entrada: uma matrix inversível A $n \times n$ (não-singular)
- b. Saída: uma matrix B $n \times n$
- c. Especificação: $A*B=B*A=I$
- d. Algoritmos
 - i. Resolver com LU

16. Problema: Decomposição LU

- a. Entrada: uma matrix A $n \times n$
- b. Saída: uma matrix triangular inferior L e um matrix triangular superior U
- c. Especificação: $A=LU$

17. Problema: PVC (Problema de Valores no Contorno) (só vamos resolver

- a. **Problema específico:** $y''(x)=c_1+c_2y(x)+c_3y'(x)$, $y(a)=y_a$ e $y(b)=y_b$
- b. Entrada: a, b, y_a, y_b , n (natural), c_1, c_2, c_3
- c. Saída: vetor de $y_2, y_3, \dots, y_{(n+1)}$ (temos escolhas aqui)
- d. Especificação: " $y(x_i)$ aproximadamente y_i "
- e. Algoritmo: Criar uma matriz com diferenças finitas e resolver um sistema linear

18. Problema: Integração Numérica (com o número de intervalos)

- a. Entrada: uma função f (de uma variável), um natural n (número de intervalos), float a , float b
- b. Saída:
- c. Especificação:
- d. Algoritmo: Método do Trapézio

19. Problema: Integração Numérica (com o erro)

- a. Entrada: uma função f (de uma variável), erro máximo, float a , float b , $M=$ maior valor de $f''(x)$ no intervalo
- b. Saída: um número S
- c. Especificação: $|(integral \text{ de } f \text{ no intervalo } a,b) - S| \leq \text{erro_máximo}$
- d. Algoritmo:
 - i. Soma de Riemann (Método do retângulo - interpolação constante) (não precisa fazer na biblioteca)
 - ii. Método do Trapézio (interpolação linear)
 - iii. Método de Simpson (interpolação quadrática) (não fizemos e não precisa fazer na biblioteca)

20. Bônus: Problema: Integração Dupla

21. Derivada Numérica com uma função contínua

- a. Entrada: uma função f (de uma variável), x , h
- b. Saída: uma função $z(x)$, (erro máximo como bonus)
- c. Especificação: $z(x)$ aproximadamente $y'(x)$
- d. Algoritmo:
 - i. Diferenças “para frente”
 - ii. Diferenças “para trás”
 - iii. Diferenças “centradas”

22. Derivada Numérica com uma função discreta

- a. Entrada: um vetor f com valores da função e um vetor x com os valores discretos de x
- b. Saída: um vetor y_linha (primeira ou segunda derivada)
- c. Especificação: y_linha aproximadamente y'
- d. Algoritmo:
 - i. Diferenças “para frente”
 - ii. Diferenças “para trás”
 - iii. Diferenças “centradas”

3 exemplos por função!!!!

Plots são bons exemplos

23. Problema: Resolver um sistema linear (métodos iterativos) (Não precisa implementar)

- a. Algoritmos
 - i. Gauss-Seidel
 - ii. Gauss-Jacobi