INTRODUÇÃO AOS MÉTODOS ESPECTRAIS
PTC 5725 (18/09/2025)

Prof. Dr. Osvaldo Guimarães

osvaldo.guimaraes@alumni.usp.br

Monitoria: Pedro Lee

Bibliografia

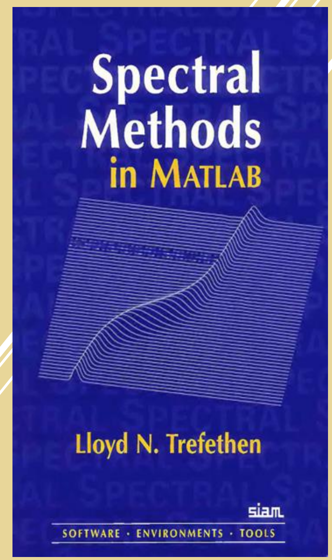
Trefethen, L. N. (2000). *Spectral Methods in MATLAB*. SIAM. (Excelente, prático e o melhor ponto de partida).

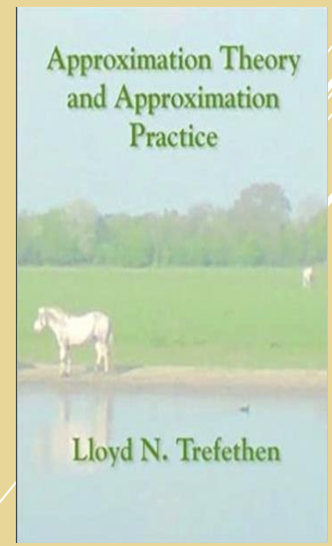
Boyd, J. P. (2001). *Chebyshev and Fourier Spectral Methods*. Dover Publications.

Canuto, C., Hussaini, M. Y., Quarteroni, A., & Zang, T. A. (2006). *Spectral Methods: Fundamentals in Single Domains*. Springer.

Shen, J., Tang, T., & Wang, L. L. (2011). *Spectral Methods: Algorithms, Analysis and Applications*. Springer.

Approximation theory and approximation practice
http://www.chebfun.org/ATAP/





Capítulos 5, 6 e 7
Chebyshev differentiation matrices

... por melhor que seja o instrumento de medição de que disponhamos, dada uma função contínua arbitrária no intervalo [a, b], existe um polinômio que consegue confundir nosso instrumento, o qual não vai poder distinguir entre esse polinômio e a função inicialmente introduzida.

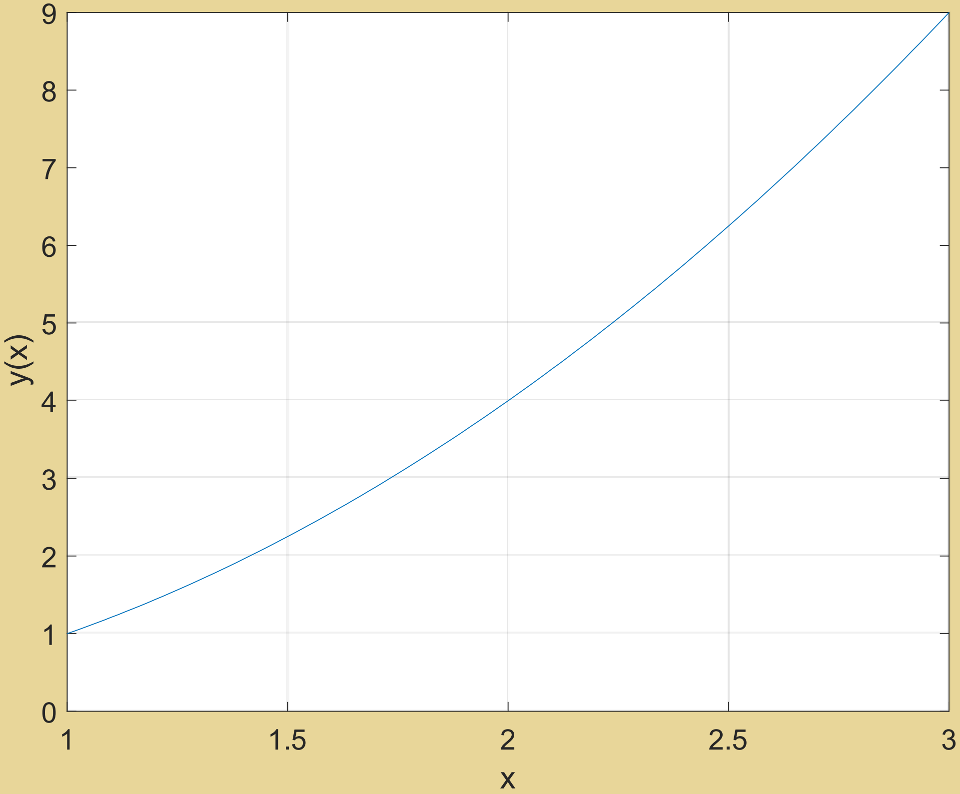
Como a avaliação numérica de um polinômio exige apenas operações algébricas (soma, multiplicação), facilmente processadas em um computador, este resultado parece indicar um caminho insuperável para a construção de algoritmos de aproximação.

MOURA, C. A. D. (2002). Análise Funcional para aplicações - Posologia. Rio de Janeiro: Ciência Moderna, 217 p.

Se $f(x)$ é uma função contínua no intervalo fechado $[a,b]$, nesse intervalo existe uma seqüência de polinômios $P_n(x)$ tal que: $\lim_{n \to \infty} P_n(x) = f(x)$, uniformemente no intervalo $[a,b]$.

O critério de convergência uniforme garante que: qualquer que seja $\varepsilon > 0$, existe um número N, independente de x no intervalo $[a,b]$, tal que $|f(x) - P_n(x)| < \varepsilon$ para todo $n \geq N$.

Interpolação de Lagrange



$$y_k = [1,4,9] \qquad x_k = [1,2,3]$$

$$\ell_1(x) = C_1(x-2)(x-3) \quad \text{e} \quad \ell_1(1) = 1 \Rightarrow C_1 = \frac{1}{(1-2)(1-3)} = \frac{1}{2}$$

$$\ell_2(x) = C_2(x-1)(x-3) \quad \text{e} \quad \ell_2(2) = 4 \Rightarrow C_2 = \frac{4}{(2-1)(2-3)} = -4$$

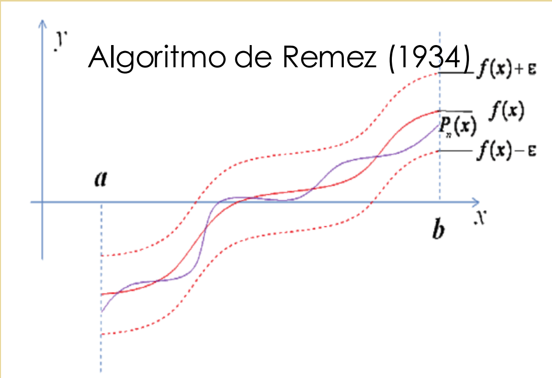
$$\ell_3(x) = C_3(x-1)(x-2) \quad \text{e} \quad \ell_3(3) = 9 \Rightarrow C_3 = \frac{9}{(3-1)(3-2)} = \frac{9}{2}$$

$$\ell_j(x) = \frac{\prod_{k=0,k\neq j}^{n}(x-x_k)}{\prod_{k=0,k\neq j}^{n}(x_j-x_k)}$$

$$\ell_1(x) = \frac{(x-2)(x-3)}{(1-2)(1-3)}$$

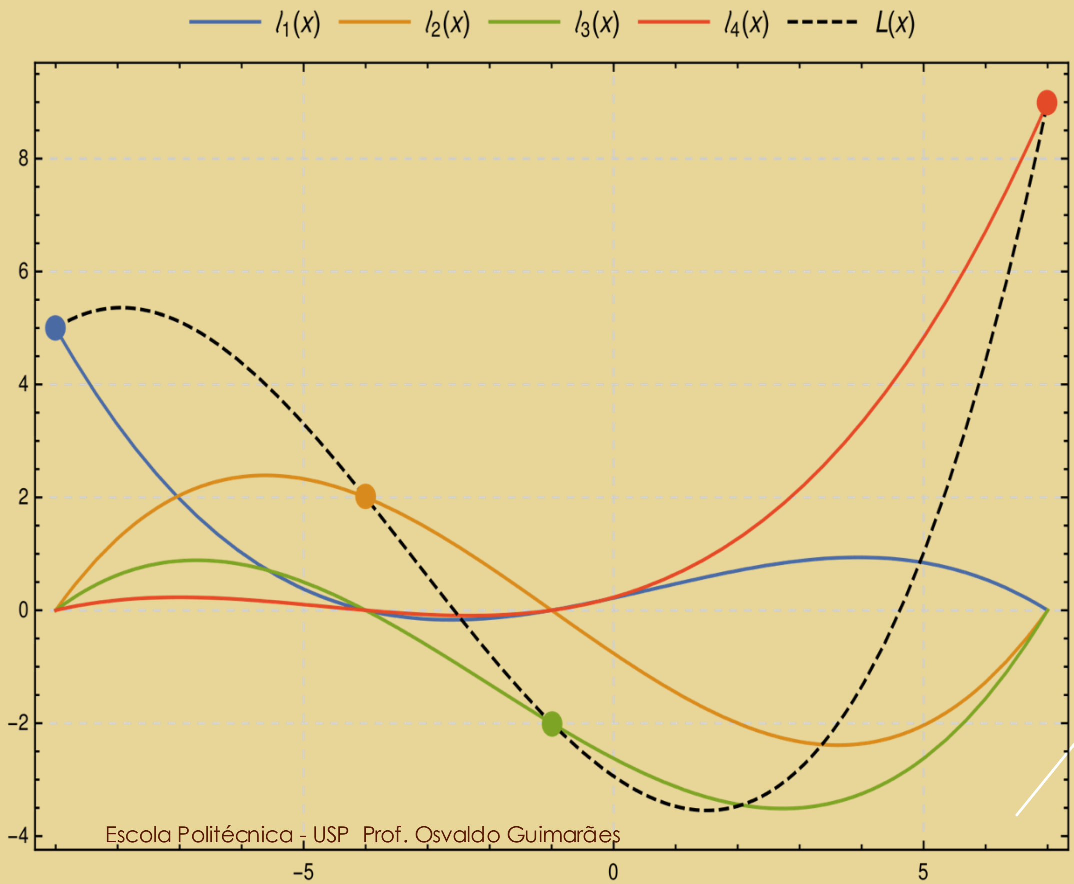
Interpolação de Lagrange

Função polyfit no Matlab



$$\ell_j(x) = \frac{\prod_{k\neq j}(x - x_k)}{\prod_{k\neq j}(x_j - x_k)}$$

$$L(x) = \sum_{j=0}^{n} \ell_j(x)\cdot f_j$$



Alternativa

$$\begin{bmatrix} x^0 & x^1 & x^2 \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = y$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 9 \end{bmatrix}$$

$$\ell_j(x) = \frac{\prod_{k \neq j} (x - x_k)}{\prod_{k \neq j} (x_j - x_k)} \qquad L(x) = \sum_{j=0}^{n} \ell_j(x) \cdot f_j$$

Considere a matriz G e um dado valor de x: $G = \begin{bmatrix} x - x_0 & x_0 - x_1 & x_0 - x_2 & \cdots & x_0 - x_n \\ x_1 - x_0 & x - x_1 & x_1 - x_2 & \cdots & x_1 - x_n \\ x_2 - x_0 & x_2 - x_1 & x - x_2 & \cdots & x_2 - x_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n - x_0 & x_n - x_1 & x_n - x_2 & \cdots & x - x_n \end{bmatrix}$

$$\ell_j(x) = \frac{\left[\prod_{k \neq j} (x - x_k)\right](x - x_j)}{\left[\prod_{k \neq j} (x_j - x_k)\right](x - x_j)} = \frac{G_d}{G_j}$$

G_d é o mesmo p/ todo ℓ_j.

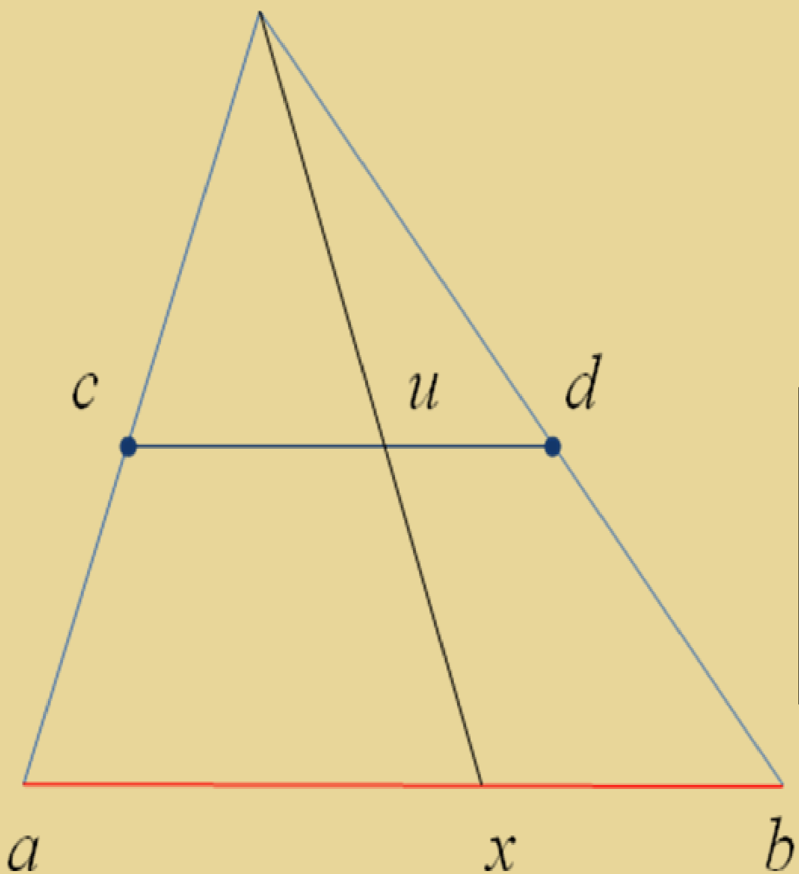
G_d é o produto da diagonal de G e G_j é produto dos elementos da linha $j+1$.

Assim, $L(x) = \sum_j \frac{G_d}{G_j} \cdot y_j = G_d \left[\frac{1}{G_j}\right] \cdot y_j$

```matlab
function yv = Lagrange_Interp(xk,yk,xv)
%%
xk = xk(:); yk = yk(:); xv = xv(:); %Make sure they are columns vector
N = length(xk);  M = length(xv); yv = zeros(M,1);
X = repmat(xk,1,N);
dX = X' - X;
%% Loop for xv
for j = 1:M
    dv = xv(j) - xk;
    dXv = dX + diag(dv);
    prod_diag = prod(dv);
        if prod_diag == 0
            aa = find(xk == xv(j));
            yv(j) = yk(aa);
        else
        iGj = (1./prod(dXv))*yk; %summation
        yv(j) = prod_diag*iGj;
        end
end
%%
```

Condição necessária para convergência uniforme polinomial

$$n \to \infty \Rightarrow x^n \to 0, \text{ logo } |x| < 1 \text{ e } [c,d] \equiv [-1,1]$$



$$\frac{u-c}{d-c} = \frac{x-a}{b-a} \text{ e } [c,d] = [-1,1] \Rightarrow \boxed{x = \frac{b-a}{2}u + \frac{b+a}{2}}$$

$$\boxed{\begin{array}{l} \dfrac{dF}{dx} = \dfrac{dG}{du}\dfrac{du}{dx} = \dfrac{dG}{du} \cdot \dfrac{2}{(b-a)}, \quad k = \dfrac{2}{(b-a)} \\ \dfrac{dF}{dx} = \dfrac{dG}{du}k, \quad \dfrac{d^nF}{dx^n} = \dfrac{d^nG}{du^n} \cdot k^n \end{array}}$$

Inversa

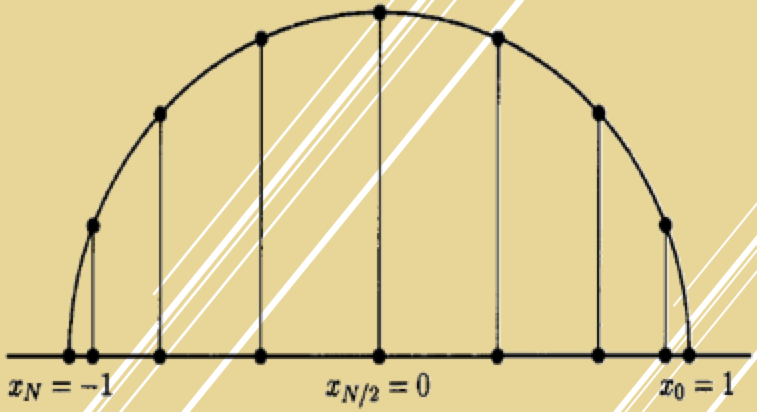
$$u = \frac{2x-b-a}{b-a}$$

J.-P. Berrut and L. N. Trefethen, "Barycentric Lagrange Interpolation," *SIAM Rev.*, vol. 46, no. 3, pp. 501–517, 2004.

```matlab
% p9.m - polynomial interpolation in equispaced and Chebyshev pts

N = 16;
xx = -1.01:.005:1.01; clf
for i = 1:2
  if i==1, s = 'equispaced points'; x = -1 + 2*(0:N)/N; end
  if i==2, s = 'Chebyshev points';  x = cos(pi*(0:N)/N); end
  subplot(2,2,i)
  u = 1./(1+16*x.^2);
  uu = 1./(1+16*xx.^2);
  p = polyfit(x,u,N);                 % interpolation
  pp = polyval(p,xx);                 % evaluation of interpolant
  plot(x,u,'.','markersize',13)
  line(xx,pp)
  axis([-1.1 1.1 -1 1.5]), title(s)
  error = norm(uu-pp,inf);
  text(-.5,-.5,['max error = ' num2str(error)])
end
```

Chebyshev-Lobatto points:

$$x = -\cos\left(\frac{k\pi}{n}\right), \text{ com } k = 0:n$$



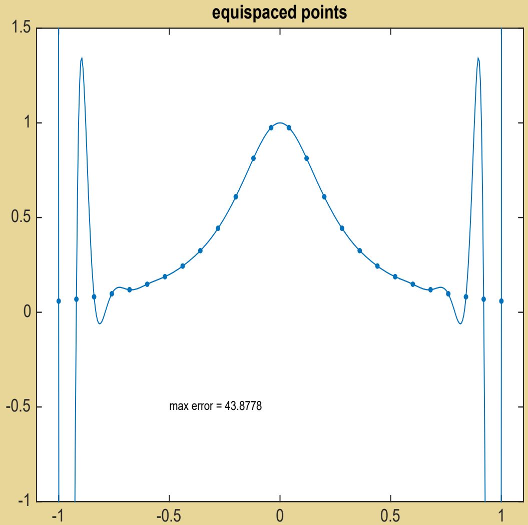
Interpolação de:

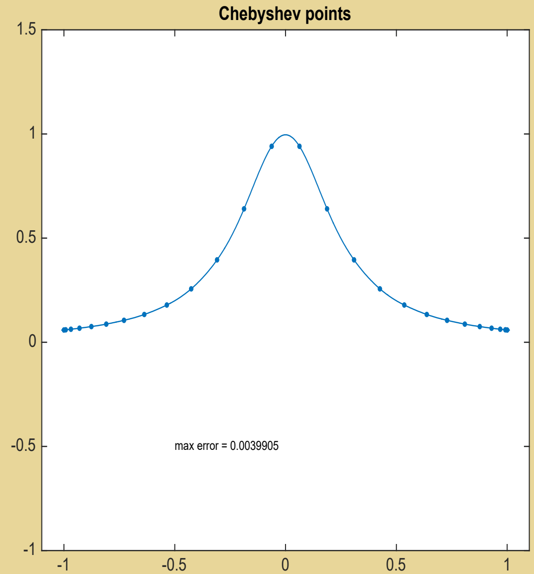
$$f(x) = \frac{1}{1+16x^2}$$

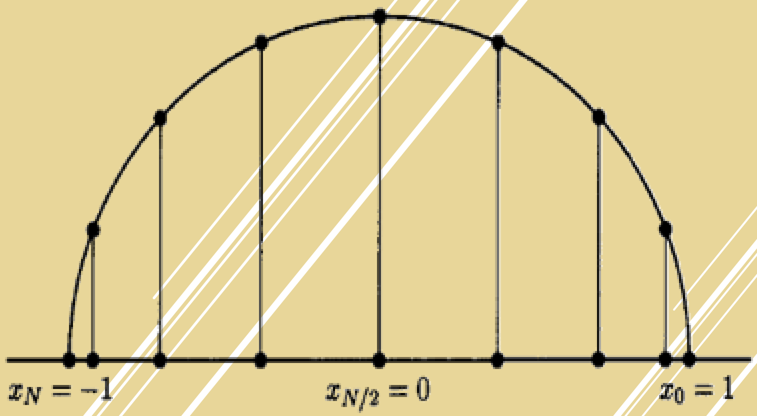
Chebyshev-Lobatto points:

$$x = -\cos\left(\frac{k\pi}{n}\right),\ \text{com}\ k = 0:n$$

N = 25







Interpolação de:

$$f(x) = \frac{1}{1+16x^2}$$

Matriz de diferenciação

$$D_{ij} = \frac{a_i}{a_j\left(x_i - x_j\right)}, \quad i \neq j \quad \text{e} \quad D_{jj} = \prod_{k \neq j}\left(x_j - x_k\right)^{-1}$$

Queremos: $D_{n+1,n+1} \cdot f_{n+1} = f'_{n+1}$

Para qualquer *grid*

$$\ell_j(x) = \frac{\prod_{k \neq j}\left(x - x_k\right)}{\prod_{k \neq j}\left(x_j - x_k\right)} \square 1 \qquad L(x) = \sum_{j=0}^{n} \ell_j(x) \cdot f_j$$

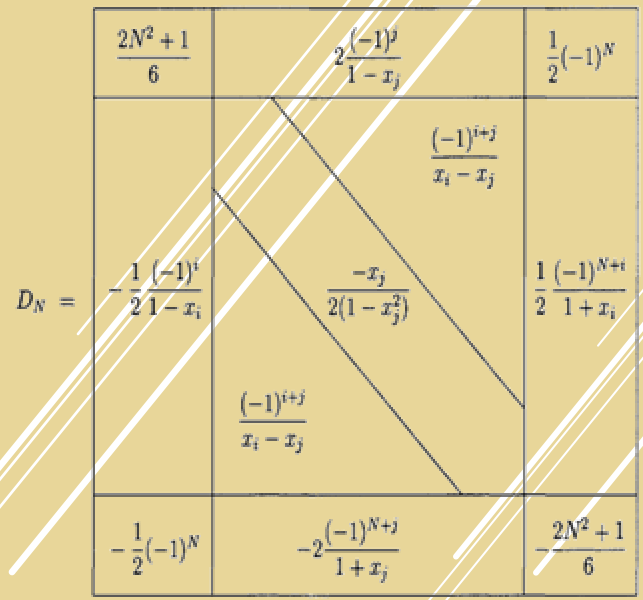
Pondo-se $a_j = \prod_{k \neq j}\left(x_j - x_k\right)$

$\ell_j(x) = \frac{1}{a_j} \prod_{k \neq j}\left(x - x_k\right)$, assim

$$\ln\left[a_j \cdot \ell_j(x)\right] = \ln\left[\prod_{k \neq j}\left(x - x_k\right)\right]$$

$\ln\left(a_j\right) + \ln\left(\ell_j(x)\right) = \ln\left[\prod_{k \neq j}\left(x - x_k\right)\right]$, derivando

$$\frac{\ell'_j(x)}{\ell_j(x)} = \sum_{k \neq j}\left(x - x_k\right)^{-1} \Rightarrow \ell'_j(x) = \ell_j(x) \cdot \sum_{k \neq j}\left(x - x_k\right)^{-1}$$

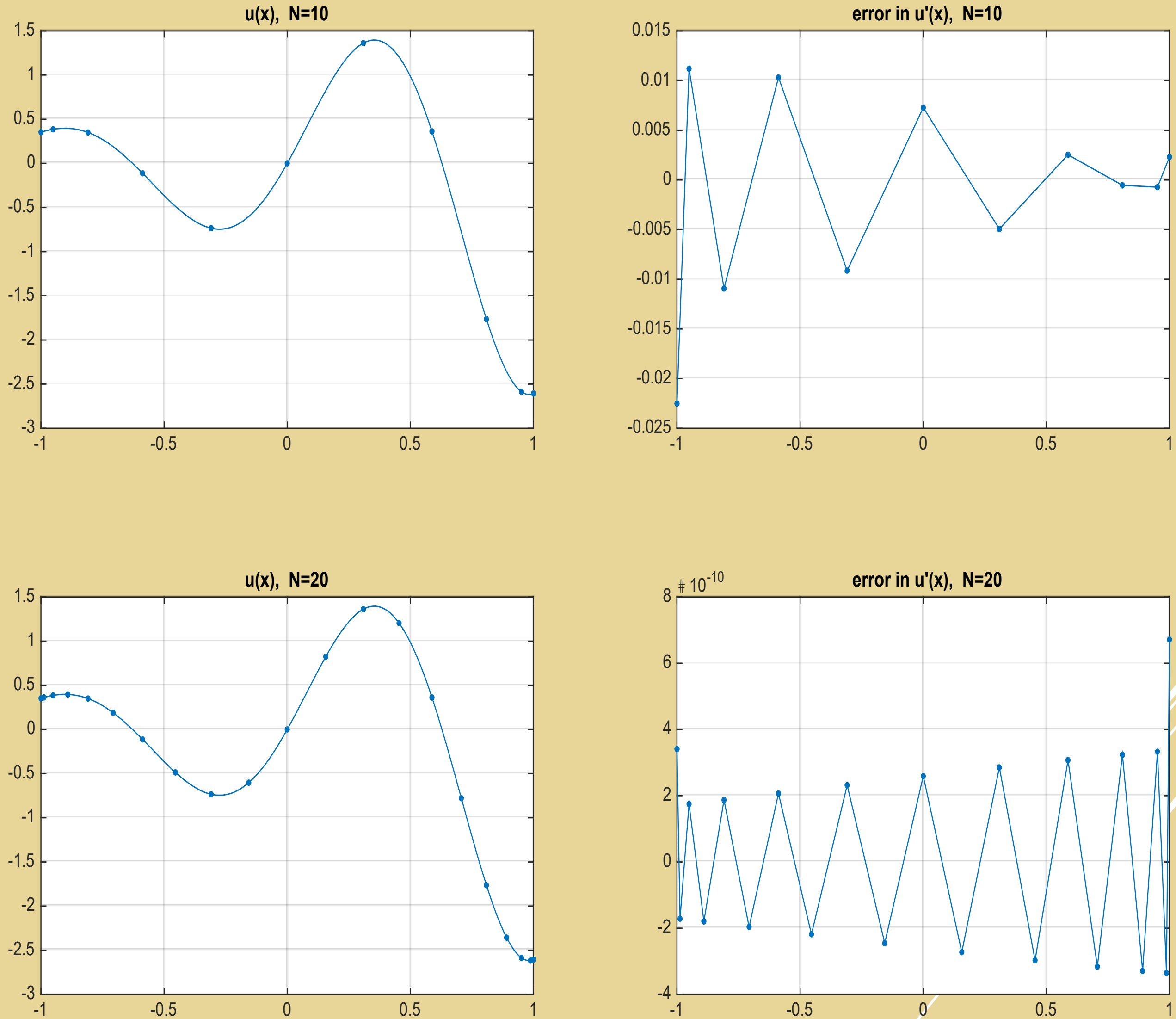


```
function D = Generalized_Diff_Mat(xs)

%%
n = length(xs) - 1;
xs = xs(:); %ensure that xs is a column vector
X = repmat(xs,1,n+1);
dX = X-X' + eye(n+1);
aj = prod(dX).';
D = (aj*(1./aj).')./dX;
D = D - diag(sum(D.'));
```

```
Command Window
>> 1 + 2^-52

ans =

    1.0000

>> format long
>> 1 + 2^-52

ans =

   1.000000000000000

>> 2^-52

ans =

     2.220446049250313e-16

>>
```

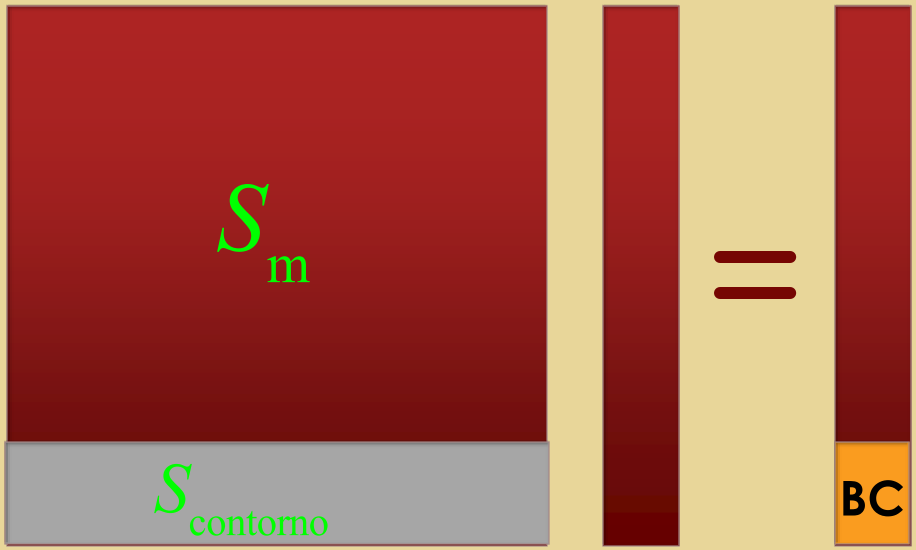
```
% p11.m - Chebyshev differentation of a smooth function

  xx = -1:.01:1; uu = exp(xx).*sin(5*xx); clf
  for N = [10 20]
    [D,x] = cheb(N); u = exp(x).*sin(5*x);
      subplot('position',[.15 .66-.4*(N==20) .31 .28])
      plot(x,u,'.','markersize',14), grid on
      line(xx,uu)
      title(['u(x),  N=' int2str(N)])
    error = D*u - exp(x).*(sin(5*x)+5*cos(5*x));
      subplot('position',[.55 .66-.4*(N==20) .31 .28])
      plot(x,error,'.','markersize',14), grid on
      line(x,error)
      title(['  error in u''(x),  N=' int2str(N)])
  end
```

Derivada de: $f\left(x\right)=e^{x}\cdot\sin\left(5x\right)$



Programa 11

Collocation



$$S \cdot \vec{y} = \vec{r}$$

Boundary Value Problems (BVP)

Programa 13

$$y_{xx} = e^{4x}, \quad y(\pm 1) = 0$$

$$\text{Analiticamente: } y = \left[e^{4x} - x \cdot \sinh(4) - \cosh(4) \right] / 16.$$

```
 %% BVP p.13 - Alternative code
N = 25;
x = -cos( (0:N)*pi/N ).'; D = Generalized_Diff_Mat(x); D2 = D^2;
D2([1,N+1],:) = 0; D2(1,1) = 1; D2(N+1,N+1) = 1;
f = exp(4*x(2:N)); f = [0;f;0];
u = D2\f;
```

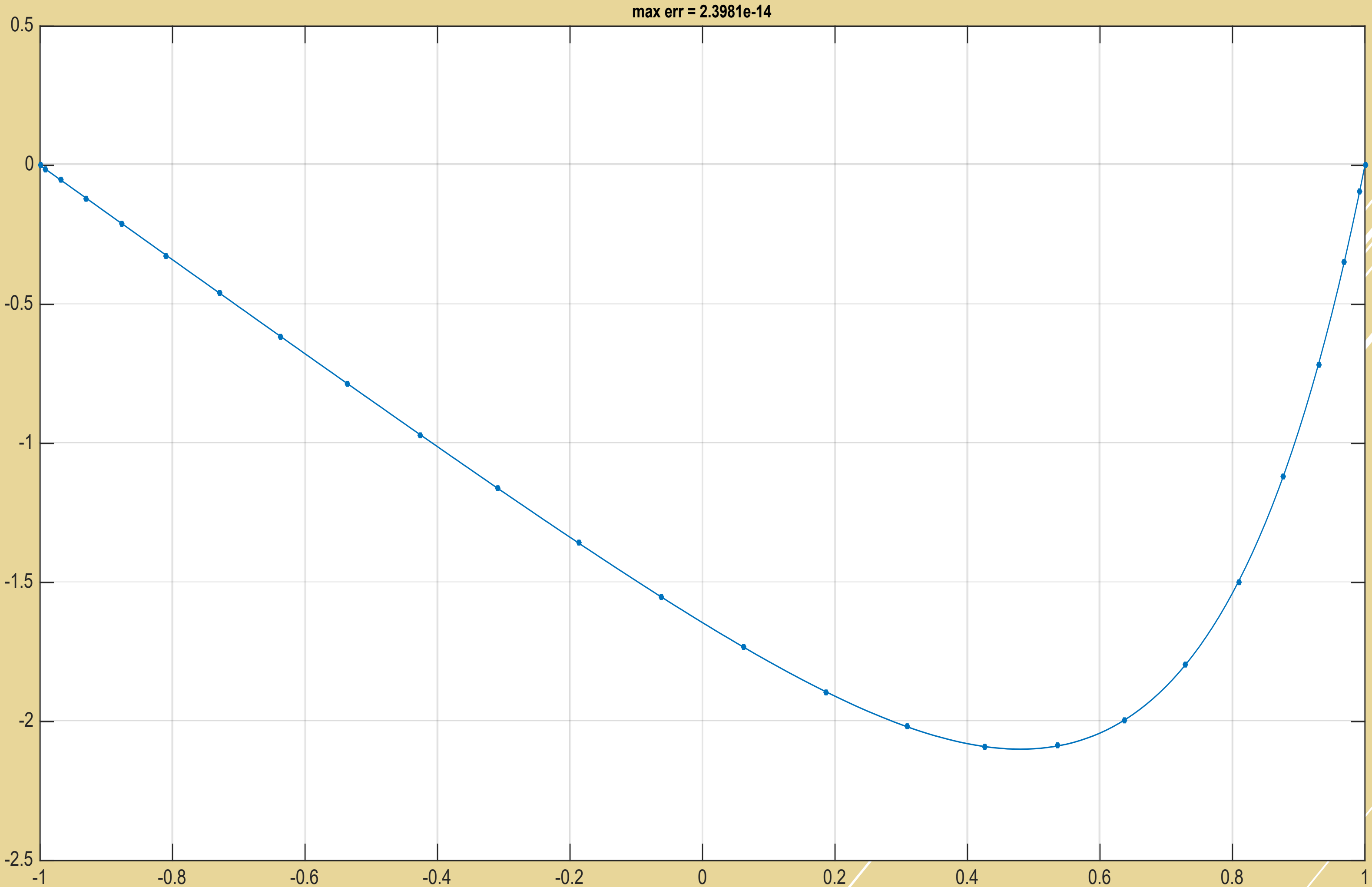
Equação matricial

$$D \cdot (D \cdot y) = f(x),$$

$$D^2 \cdot y = f,$$

□Linha 1 de D^2 : $[1\ 0\ ...0]$ e $f_1 = y(-1) = 0$

□Linha $(n+1)$ de D^2 : $[0......1]$ e $f_{n+1} = y(+1) = 0$


max err = 2.3981e-14
0.5
0
-0.5
-1
-1.5
-2
-2.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

Tarefa

1) Escreva um código para obter os valores de $f\left(x_k\right)$ com a interpolação de Lagrange feita para os $\left(n+1\right)$ pontos $f\left(x_j\right)$.

2) Mostre que o polinômio interpolante de Lagrange para n+1 pontos é o único polinômio de grau $\leq n$ que passa por todos esses pontos.

3) Demonstre a expressão da matriz de diferenciação generalizada e escreva um código para obtê-la.

4) Analise o artigo sobre interpolação baricêntrica e escreva um código para ela ser efetuada para x_j, dados x_k e y_k.

Funções para testar no intervalo $x \in \left[-1;1\right] \subset$ » :

e^{-x}, $\sin\left(\pi x\right)$, $\sin\left(\pi x\right)$, $10^4 \cos\left(\pi x\right)$ e $32x^6 - 48x^4 + 18x^2 - 1$.

$$x_k = -1 + k\frac{2}{11} \quad k = 0,1,\ldots,11 \quad \text{e} \qquad x_i = -1 + i\frac{2}{120} \quad i = 1,2,\ldots,120$$

Linguagem: Julia ou Pyton
Enviar junto com o program.jl um pdf pelo email:
osvaldo.guimaraes@alumni.usp.br c/ comentários e documentação. Prazo: até 4ª feira 01/10/25