Interpolação usando Polinômio de Lagrange

Danilo Souza
Iago Medeiros

Universidade Federal do Pará
Instituto de Tecnologia
Faculdade de Engenharia da Computação e Telecomunicações

15 de Dezembro de 2014

Agenda

Introdução

Interpolação: introduzir partes em algo para completar o todo

- Encontrar funções que representam um fenômeno

Introdução

Interpolação: introduzir partes em algo para completar o todo

- Encontrar funções que representam um fenômeno
- A partir de alguns pontos de medições do mundo real

Introdução

Interpolação: introduzir partes em algo para completar o todo

- Encontrar funções que representam um fenômeno
- A partir de alguns pontos de medições do mundo real
- Porque polinômios?

Introdução

Interpolação: introduzir partes em algo para completar o todo

- Encontrar funções que representam um fenômeno
- A partir de alguns pontos de medições do mundo real
- Porque polinômios?
  - Única função que pode ser encontrada usando as 4 operações elementares

Introdução

Interpolação: introduzir partes em algo para completar o todo

- Encontrar funções que representam um fenômeno
- A partir de alguns pontos de medições do mundo real
- Porque polinômios?
  - Única função que pode ser encontrada usando as 4 operações elementares
  - Polinômio de Lagrange

Agenda

Definindo o Polinômio

O polinômio de Lagrange é definido conforme:

$$
\begin{aligned}
P_0 &= (x - x_1)(x - x_2)\cdots(x - x_n) \\
P_1 &= (x - x_0)(x - x_2)\cdots(x - x_n) \\
P_2 &= (x - x_0)(x - x_1)\cdots(x - x_n) \\
&\vdots \\
P_n &= (x - x_0)(x - x_1)\cdots(x - x_{n-1})
\end{aligned}
\tag{1}
$$

Propriedades do Polinômio

Propriedades do método:

$$P_i(x_j) = 0; i \neq j$$
$$P_i(x_j) \neq 0; i = j \qquad (2)$$

A equação do Polinômio

$$P_n(x) = b_0 p_0(x) + b_1 p1(x) + \cdots + b_k p_k(x) + \cdots + b_n p_n(x) \quad (3)$$

Onde $b_0, b_1, b_2, \cdots, b_n$ *são constantes a serem determinadas*

Encontrando os coeficientes

Generalizando a solução de b, temos:

$$b_i = \frac{y_i}{P_i(x_i)} \tag{4}$$

Juntando as equações 3 e 4 temos:

$$P_n(x) = \sum_{i=0}^{n} \frac{y_i P_i(x)}{P_i(x_i)}$$
$$P_n(x) = \sum_{i=0}^{n} y_i \prod_{j=0}^{n}{}_{i \neq j} \frac{(x - x_j)}{(x_i - x_j)} \tag{5}$$

Generalizando

A equação implementada no código foi:

$$P_n(x) = Prod_x \sum_{i=0}^{n} \frac{y_i}{Dif_i Prod_i} \tag{6}$$

Agenda

Cabeçalho

```
% Declaracao de variaveis
n = length(x_inicial);
dif(1,n) = 0;
produto = ones(1,n);
mat_dif(n,n) = 1;
x = valor;
prod_x = 1;
```

Principal

```
for i=1:n
    % Calcula a diferenca x - x(i); i=1,2,...,n
    dif(1,i) = x - x_inicial(i);
    % Calcula o valor de Prod_x
    prod_x = prod_x*dif(i);
    for j=1:n
        if i==j
            mat_dif(i,j) = 1;
        else
            % Constroi a matriz de diferencas
            mat_dif(i,j) = x_inicial(i)-x_inicial(j);
        end
        % Usa o resultado da matriz de diferencas
        produto(1,i) = produto(1,i)*mat_dif(i,j);
    end
end
```

Resultado

```
%mat_dif(find(eye(3))) = [1 1 1];
resultado = zeros(1,n);

% Calcula cada termo da solucao
for i=1:n
    resultado(1,i) = y_inicial(1,i)/(dif(1,i)*produto(1,i));
end

% Formula final
p = prod_x*(sum(resultado));
```

Agenda

Conclusão

- Mesmo não sendo o melhor método, ele tem uma boa velocidade e desempenho.

Conclusão

- Mesmo não sendo o melhor método, ele tem uma boa velocidade e desempenho.
- Bom para aplicações em tempo real

Conclusão

- Mesmo não sendo o melhor método, ele tem uma boa velocidade e desempenho.
- Bom para aplicações em tempo real
- Simples de ser implementado