

# Implementação do método interpolação de Lagrange

A implementação do método de interpolação de Lagrange, para a lista de exercícios 3 da disciplina de matemática computacional (CAP-239-4) do Instituto Nacional de Pesquisas Espaciais, foi feito utilizando a linguagem de programação C++, os códigos implementados e conceitos são apresentados nas seções abaixo.

## Ideia base e implementação

A ideia base utilizada para a implementação do método pode ser apresentada em duas etapas.

**Preparação dos dados:** Como já apresentado durante as aulas da disciplina, a interpolação pelo método de lagrange é feita basicamente através do ajuste dos chamados polinômios de Lagrange, sendo estes apresentados da seguinte forma

$$l_k(x) = \frac{(x - x_0) \dots (x - x_{k-1}) (x - x_{k+1}) \dots (x - x_n)}{(x_k - x_0) \dots (x_k - x_{k-1}) (x_k - x_{k+1}) \dots (x_k - x_n)}$$

Ao visualizar a forma dos polinômios de lagrange é possível perceber que, o numerador é composto por operações que envolvem o valor termo independente, que será interpolado e que, no denominador está presente apenas os elementos já conhecidos.

Seguindo esta ideia, esta parte do código, nomeada de **Preparação dos dados**, separa todos os dados que serão utilizados no numerador, na geração de cada polinômio de lagrange, ao mesmo tempo que já calcula os elementos do denominador, também referente a cada polinômio gerado.

**Interpolação:** Através dos dados gerados na etapa anterior, a interpolação pode ser aplicada, esta que funciona da seguinte forma.

Para cada polinômio de Lagrange que precisa ser gerado, é inserido o termo independente  $X$  e então, todos os dados que foram separados para os numeradores na etapa anterior são subtraídos por  $X$  e tem seu produtório calculado, com esta etapa feita, os elementos que já estão calculados do denominador, referentes ao polinômio que está sendo gerado são utilizados, para dividir o resultado do produtório.

Desta forma, os valores podem ser interpolados.

## Exemplo de utilização

Para deixar claro cada etapa dentro do código, abaixo um exemplo é listado

```

#include <iostream>

#include "xtensor/xarray.hpp"
#include "xtensor/xview.hpp"

#include "LagrangeInterpolator.hpp"

int main()
{
    xt::xarray<double> x({1, 3, 5, 7});
    xt::xarray<double> y({2, 4, 3, 4});

    // Etapa de preparação dos dados
    interp::LagrangeInterpolator l(x, y);
    l.fit();

    // Etapa de interpolação dos dados
    xt::xarray<double> result = l.interp(xt::xarray<double>({7}));
    std::cout << result.data()[0] << std::endl;

    return 0;
}

```

O nome `fit` foi utilizado na etapa de preparação dos dados simplesmente por seguir um padrão de nomenclatura inspirado na biblioteca [scikit-learn](#)

## Considerações sobre o código

Com o código implementado, é necessário que algumas considerações sobre o mesmo sejam feitas.

**Forma da implementação:** A maneira de implementação é simples e foi criada considerando somente as características fundamentais das formas da equação de Lagrange.

**Otimização:** O código ganha em simplicidade, porém, com o aumento da quantidade de pontos pode ser que o mesmo passe a enfrentar problemas de otimização, isso ocorre já que, a maneira como o mesmo foi implementado realiza múltiplos cálculos e consome recursos auxiliares para tais operações.

Um contorno a esta questão foi feito utilizando bibliotecas que já possuem otimizações de implementação, porém, é interessante deixar claro que para grandes quantidades de pontos, as operações de combinações na geração dos métodos de Lagrange devem ser reconsiderados.