

---

# MATLAB - Comando polyfit, polyval, lsqcurvefit

Ana Maria A. C. Rocha

Universidade do Minho, Escola de Engenharia, Departamento de Produção e Sistemas

---

## Comando polyfit

O comando `polyfit` calcula os coeficientes de um polinómio de grau `n` que melhor se ajusta aos dados `Y`, no sentido de mínimos quadrados. Calcula também o resíduo.

$$[p,S] = \text{polyfit}(X,Y,n)$$

### Argumentos de entrada

- `X` - é o vetor com os pontos.
- `Y` - é o vetor com os valores da função nos pontos.
- `n` - é o grau do polinómio que se quer construir.

### Argumentos de saída:

- `p` - é o vetor com os coeficientes do polinómio, em potências decrescentes

$$p_n(x) = c_1x^n + c_2x^{n-1} + \dots + c_nx + c_{n+1}$$

- `S` - é uma estrutura para obter uma estimativa do erro. A estrutura contém os seguintes campos:

- `R` - matriz triangular superior da decomposição QR da matriz de Vandermonde de `X`
- `df` - graus de liberdade
- `normr` - é a norma dos resíduos, i.e., é a raiz quadrada da soma do quadrado dos resíduos (erro).

## Comando polyval

O comando `polyval` calcula o valor do polinómio `p` num ponto ou conjunto de pontos `x`. `p` é um vetor de comprimento `n + 1` cujos elementos são os coeficientes do polinómio, em potências decrescentes.

$$y = \text{polyval}(p,x)$$

### Argumentos de entrada

- `p` - é o vetor com os coeficientes do polinómio, em potências decrescentes.
- `x` - é o ponto ou vetor de pontos para os quais se pretende calcular o valor do polinómio.

### Argumentos de saída:

- `y` - é o valor ou vetor de valores do polinómio.

## Comando `lsqcurvefit`

O comando `lsqcurvefit` calcula os coeficientes de um modelo  $M(x)$  que melhor se ajusta aos dados  $y$ , no sentido de mínimos quadrados. Calcula também o resíduo.

$$[c, \text{resnorm}] = \text{lsqcurvefit}(\text{fun}, c0, x, y)$$

### Argumentos de entrada

- `fun` - é a função com o modelo.

A função tem dois argumentos de entrada: `c` (vetor dos coeficientes do modelo) e `x` (variável do modelo). Usar os operadores `./` e `.*` e `.^` na definição da função.

- Especificar `fun` como uma função

```
[c,resnorm] = lsqcurvefit(@fun,c0,x,y)
function M = fun(c,x)
M = ... ;
end
```

- Especificar `fun` como um identificador de função anónima:

```
[c,resnorm] = lsqcurvefit(@(c,x)...,c0,x,y)
```

- Especificar a função `fun`:

```
fun = @(c,x)...;
[c,resnorm] = lsqcurvefit(fun,c0,x,y)
```

- `c0` - é o vetor com a aproximação inicial aos coeficientes.
- `x` - é o vetor com os pontos.
- `y` - é o vetor com os valores da função nos pontos.

### Argumentos de saída:

- `c` - é o vetor com os coeficientes do modelo.
- `resnorm` - é a soma do quadrado dos resíduos (erro).

## Técnica dos Mínimos Quadrados

- A técnica dos mínimos quadrados faz o ajuste de curvas, para um conjunto de pontos, por exemplo, obtidos numa experiência.
- Na aplicação da técnica dos mínimos quadrados, o polinómio/modelo que se ajusta ao conjunto de pontos `x` não tem de passar por nenhum dos pontos (se acontecer, é mera casualidade).
- A técnica dos mínimos quadrados é distinta da interpolação, dado que na interpolação a curva ajustada passa por todos os pontos.