

Quick Reference

Polinomio <T>

Polynomial declaration, p.e. `Polinomio<double> pol;`

make(int d,int m)

Build internally the polynomial in d -dimensions with $m-1$ order. This is the first function that must be call before to employ any of the following functions.

deriva(string ip, int orden)

Derivate the polynomial, e.g, `deriva("x",1)` is the first derivative with respect of x . You need to use the make function previously.

eval(const T *x, int d, T *px, int dim_px)

Evaluate the polynomial in the point x in d -dimensions and the result is stored in the vector px with length dim_px . Observe that dim_px must be equal than the value returned by `get_M()`. You need to use the make function previously.

Matrix<T> build_tnt(const T *x, int n, int d)

Given a vector x with size n and each element in d -dimensions, this function evaluate the polynomial at each element of x an the values are stored in the matrix P of size $n \times m$, where m are the elements of the base in the polynomial. You need to use the make function previously.

int get_M(void)

Return the number of the elements in the base of the polynomial. Recall that this value depends of d and m used firstly in make function. You need to use the make function previously.

int get_d(void)

Get the d -dimension of the data, e.g. return 1 if the data is 1- d . You need to use the make function previously.

int get_m(void)

Get the degree of the polynomial. In particular, each element in pol has at most $m-1$ in the degree. You need to use the make function previously.

Author.

José Antonio Muñoz-Gómez
Universidad de Guadalajara
CUCSUR
June-2009