MOP Reference Manual

0.1.0

Michael Carley
m.j.carley@bath.ac.uk

Contents

Chapter 1

MOP File Index

1.1 MOP File List

Here is a list of all documented files with brief descriptions:

Chapter 2

MOP File Documentation

2.1 mop.c File Reference

Functions

- gint **mop_number_of_terms** (gint dim, gint order)
- mop_polynomial * **mop_polynomial_alloc** (gint np, gint dim, gint order)
- gint **mop_polynomial_free** (mop_polynomial *p)
- mop_polynomial_workspace * **mop_polynomial_workspace_alloc** (gint np, gint dim, gint order)
- gint **mop_polynomial_workspace_free** (mop_polynomial_workspace *w)
- gint **mop_polynomial_set_points** (mop_polynomial *p, gdouble *x, gdouble *w, gint n)
- gint **mop_polynomial_write** (mop_polynomial *p, FILE *f)
- gint **mop_polynomial_basis_power** (mop_polynomial *p, gint order, gdouble tol, mop_polynomial_workspace *w)
- gint **mop_polynomial_basis_points** (mop_polynomial *p, gint order, gdouble tol, mop_polynomial_workspace *w)
- gint **mop_polynomial_make** (mop_polynomial *p, mop_polynomial_workspace *w)
- gint **mop_polynomial_write_latex** (mop_polynomial *p, FILE *f)
- gint **mop_polynomial_normalize** (mop_polynomial *p, mop_polynomial_workspace *w)
- gint **mop_polynomial_eval** (mop_polynomial *p, gdouble *x, gdouble *P)
- gint **mop_polynomial_eval_base** (mop_polynomial *p, gdouble *P)
- gint **mop_polynomial_transform** (mop_polynomial *p, gdouble *f, gint n, gdouble *c, mop_polynomial_workspace *w)
- gint **mop_interpolate** (mop_polynomial *p, gdouble *c, gint n, gdouble *x, gdouble *f)
- gint **mop_interpolation_weights** (mop_polynomial *p, gdouble *x, gdouble *v, mop_polynomial_workspace *w)
- gint **mop_polynomial_differentiate** (mop_polynomial *p, gdouble *x, gint *d, gdouble *P)
- gint **mop_differentiation_weights** (mop_polynomial *p, gdouble *x, gint *d, gdouble *v, mop_polynomial_workspace *w)

2.1.1 Detailed Description

Author:

Michael Carley

Date:

Wed Nov 14 17:53:17 2007

2.1.2 Function Documentation

gint mop_differentiation_weights (mop_polynomial * *p*, gdouble * *x*, gint * *d*, gdouble * *v*, mop_polynomial_workspace * *w*)

Compute the weights for direct differentiation at a point, based on values at the base points of a system of orthogonal polynomials, so that $\partial^{d_1+d_2+\cdots} f(\mathbf{x})/\partial x_1^{d_1} x_2^{d_2} \ldots \approx \sum v_i f_i$. Remember to use mop_polynomial_index to map base points to real points.

Parameters:

- ***p*** a mop_polynomial;
- ***x*** interpolation point;
- ***d*** array of derivative orders;
- ***v*** differentiation weights;
- ***w*** a suitably sized mop_polynomial_workspace.

Returns:

0 on success.

gint mop_interpolate (mop_polynomial * *p*, gdouble * *c*, gint *n*, gdouble * *x*, gdouble * *f*)

Evaluate a function expanded in orthogonal polynomials, $f \approx \sum_i c_i P_i(\mathbf{x})$.

Parameters:

- ***p*** mop_polynomial for the expansion;
- ***c*** coefficients of the expansion, from **mop_polynomial_transform** (p. 8);
- ***n*** number of function values, as in **mop_polynomial_transform** (p. 8);
- ***x*** coordinates of evaluation point;
- ***f*** value(s) of function(s).

Returns:

0 on success.

gint mop_interpolation_weights (mop_polynomial * *p*, gdouble * *x*, gdouble * *v*, mop_polynomial_workspace * *w*)

Compute the weights for direct interpolation at a point, based on values at the base points of a system of orthogonal polynomials, so that $f(\mathbf{x}) \approx \sum v_i f_i$. Remember to use mop_polynomial_index to map base points to real points.

Parameters:

- ***p*** a mop_polynomial;
- ***x*** interpolation point;
- ***v*** interpolation weights;
- ***w*** a suitably sized mop_polynomial_workspace.

Returns:

0 on success.

gint mop_number_of_terms (gint *dim*, gint *order*)

The number of monomials required for a multi-variable polynomial of a given order.

Parameters:

dim dimension of system;

order order of polynomial.

Returns:

number of terms in general polynomial of order *order* in *dim* dimensions.

mop_polynomial* mop_polynomial_alloc (gint *np*, gint *dim*, gint *order*)

Allocate a mop_polynomial.

Parameters:

np (maximum) number of points in basis;

dim dimension of system;

order maximum order of polynomial to be generated.

Returns:

pointer to new mop_polynomial.

gint mop_polynomial_basis_points (mop_polynomial * *p*, gint *order*, gdouble *tol*, mop_polynomial_workspace * *w*)

Generate the basis for a mop_polynomial by selecting sufficient points to match the number of monomial powers supplied.

Parameters:

p a mop_polynomial, which should have been initialized by a call to **mop_polynomial_set_points** (p. 8);

order maximum order of monomial to employed;

tol tolerance to be used in determining rank of basis matrix;

w suitably sized mop_polynomial_workspace.

Returns:

0 on success

gint mop_polynomial_basis_power (mop_polynomial * *p*, gint *order*, gdouble *tol*, mop_polynomial_workspace * *w*)

Generate the basis for a mop_polynomial by selecting sufficient monomial powers to match the number of points in the mop_polynomial, using the method of Xu, Yuan, 2004, 'On discrete orthogonal polynomials of several variables', Advances in Applied Mathematics, 33:615–632, doi:10.1016/j.aam.2004.03.002.

Parameters:

p a mop_polynomial, which should have been initialized by a call to **mop_polynomial_set_points** (p. 8);

order maximum order of monomial to employed;

tol tolerance to be used in determining rank of basis matrix;

w suitably sized mop_polynomial_workspace.

Returns:

0 on success

gint mop_polynomial_differentiate (mop_polynomial * *p*, gdouble * *x*, gint * *d*, gdouble * *P*)

Evaluate derivatives of a mop_polynomial at a point *x*.

Parameters:

p mop_polynomial to evaluate;

x coordinates of evaluation point;

d orders of differentiation for each dimension;

P array containing values of *p* at *x*.

Returns:

0 on success.

gint mop_polynomial_eval (mop_polynomial * *p*, gdouble * *x*, gdouble * *P*)

Evaluate a mop_polynomial at a point *x*.

Parameters:

p mop_polynomial to evaluate;

x coordinates of evaluation point;

P array containing values of *p* at *x*.

Returns:

0 on success.

gint mop_polynomial_eval_base (mop_polynomial * *p*, gdouble * *P*)

Evaluate a mop_polynomial at its base points. Note that the base points used are those in the index list of *p* and they are used in the order in that list. To connect a given value of orthogonal polynomial to a particular base point, use mop_polynomial_index.

Parameters:

p mop_polynomial to evaluate;

P array of *p* evaluated at its base points so that $P_i(x_j)$ is P[j∗mop_polynomial_nterms(p)+i].

Returns:

0 on success.

gint mop_polynomial_free (mop_polynomial ∗ *p*)

Free a mop_polynomial and associated memory

Parameters:

p mop_polynomial to free.

Returns:

0 on success.

gint mop_polynomial_make (mop_polynomial ∗ *p*, mop_polynomial_workspace ∗ *w*)

Generate the discrete orthogonal polynomials associated with *p*, using the method of Xu, Yuan, 2004, 'On discrete orthogonal polynomials of several variables', Advances in Applied Mathematics, 33:615–632, doi:10.1016/j.aam.2004.03.002.

Parameters:

p a mop_polynomial, which should have been initialized by a call to **mop_polynomial_basis_power** (p. 5) or **mop_polynomial_basis_points** (p. 5);

w a suitably sized mop_polynomial_workspace, generated by **mop_polynomial_workspace_alloc** (p. 8).

Returns:

0 on success.

gint mop_polynomial_normalize (mop_polynomial ∗ *p*, mop_polynomial_workspace ∗ *w*)

Scale the coefficients of a mop_polynomial to give unit inner product, $\sum_i P_j^2(x_i) w_i \equiv 1$.

Parameters:

p mop_polynomial to normalize;

w a mop_polynomial_workspace of appropriate size.

Returns:

0 on success.

gint mop_polynomial_set_points (mop_polynomial * *p*, gdouble * *x*, gdouble * *w*, gint *n*)

Set points and weights to be used in generating sets of orthogonal polynomials. The data are copied into *p* so the arrays can be reused.

Parameters:

p a mop_polynomial of appropriate size;

x array of points of the same dimension as *p*;

w array of weights, one for each *x*;

n number of points.

Returns:

0 on success.

gint mop_polynomial_transform (mop_polynomial * *p*, gdouble * *f*, gint *n*, gdouble * *c*, mop_polynomial_workspace * *w*)

Calculate the coefficients of an expansion of a function f in the polynomial p.

Parameters:

p mop_polynomial for the expansion;

f value(s) of function(s) at base points of *p*;

n number of values at each base point;

c coefficients of expansion, $f \approx \sum_i c_i P_i(\mathbf{x})$;

w a mop_polynomial_workspace of appropriate size.

Returns:

0 on success.

mop_polynomial_workspace* mop_polynomial_workspace_alloc (gint *np*, gint *dim*, gint *order*)

Allocate a mop_polynomial_workspace for use in generating orthogonal polynomials.

Parameters:

np (maximum) number of points in basis;

dim dimension of system;

order maximum order of polynomial to be generated.

Returns:

pointer to new mop_polynomial_workspace.

gint mop_polynomial_workspace_free (mop_polynomial_workspace * *w*)

Free a mop_polynomial_workspace and associated memory

Parameters:

w mop_polynomial_workspace to free.

Returns:

0 on success.

gint mop_polynomial_write (mop_polynomial * *p*, FILE * *f*)

Write a mop_polynomial to a file.

Parameters:

p mop_polynomial to write;

f file pointer.

Returns:

0 on success.

gint mop_polynomial_write_latex (mop_polynomial * *p*, FILE * *f*)

Write a mop_polynomial as a fragment of LaTeX code.

Parameters:

p mop_polynomial to write;

f file pointer to write to.

Returns:

0 on success.