
xtensor-interpolate

Apr 10, 2018

1	Installation	3
1.1	CMake	3
2	API Reference	5

Spline interpolation in `xtensor` via internal bindings to Paul Dierckx's FITPACK.

`xtensor-interpolate` is licensed under the **BSD-2-Clause** license. See the LICENSE file for details.

1.1 CMake

```
mkdir build
cd build
cmake ..
make install
```

This will install the `xtensor-interpolate` headers, compile and install the FITPACK interface libraries (`.so` on Linux, `.dll` on Windows), and place the CMake project configuration file in the appropriate location so that other projects can use CMake's `find_package(xtensor-interpolate)`.

The `xtensor-interpolate` API is closely modelled after Python's `scipy.interpolate`.

Defined in `xtensor-interpolate/xinterpolate.hpp`.

template <class E1, class E2, class E3>

auto xt::interpolate::splrep(const xexpression<E1> &x, const xexpression<E2> &y, const xexpression<E3> &w, double xb, double xe, int k, double s)

Find the B-spline representation of a 1-D curve.

Given the set of data points $(x[i], y[i])$ determine a smooth spline approximation of degree k on the interval $xb \leq x \leq xe$.

Return A tuple containing the vector of knots, τ , the B-spline coefficients, c , and the degree of the spline, k . It is not recommended to manually adjust any of these values.

Parameters

- xy : The data points defining a curve $y = f(x)$.
- k : The order of the spline fit. It is recommended to use cubic splines.

Exceptions

- `std::runtime_error`: Thrown if input data is not 1-D or not the same length, or if k does not satisfy $0 \leq k \leq 5$.
- `std::runtime_error`: Thrown if FITPACK encounters any errors.

template <class E1, class E2>

auto xt::interpolate::splrep(const xexpression<E1> &x, const xexpression<E2> &y, int k = 3, double s = 0.0)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

template <class E, class... Args>

auto xt::interpolate::splev(const xexpression<E> &x, const std::tuple<Args...> &tck, int der = 0, int ext = 0)

Evaluate a B-Spline or its derivatives.

Return An array of interpolated values corresponding to the input parameter x .

Parameters

- `x`: An array of points at which to return the value of the smoothed spline or its derivatives.
- `tck`: The tuple returned by `splrep` containing the knots, coefficients, and degree of the spline.
- `der`: The order of derivative of the spline to compute (must be less than or equal to `k`).
- `ext`: Controls the value returned for elements of `x` not in the interval defined by the knot sequence.
 - if 0, return the extrapolated value.
 - if 1, return 0
 - if 2, throw exception.
 - if 3, return the boundary value.

Exceptions

- `std::runtime_error`: Thrown if FITPACK encounters any errors.

```
template <class... Args>
```

```
auto xt::interpolate::splint(double a, double b, const std::tuple<Args...> &tck)
```

Evaluate the definite integral of a B-spline between two points.

Return The resultant integral.

Note This routine assumes the spline is 0 outside of its data points.

Parameters

- `ab`: The endpoints defining the bounds of integration.
- `tck`: A tuple containing the knots, B-spline coefficients, and degree of the spline.

```
template <class... Args>
```

```
auto xt::interpolate::sproot(const std::tuple<Args...> &tck, int mest = 10)
```

Find the roots of a cubic B-spline.

Return An array giving the roots of the spline.

Warning The degree of the spline must be 3.

Warning The number of knots must be greater than or equal to 8.

Parameters

- `tck`: A tuple containing the knots, B-spline coefficients, and degree of the spline.
- `mest`: An estimate of the number of roots.

Exceptions

- `std::runtime_error`: Thrown if the input data is invalid, or if FITPACK encounters any errors.

```
template <class E, class... Args>
```

```
auto xt::interpolate::spalde(const xexpression<E> &x, const std::tuple<Args...> &tck)
```

Evaluate all derivatives of a B-spline.

Return An `xarray` of all derivatives of the spline up to order `k` for each point in `x`.

Parameters

- `x`: A set of points at which to evaluate the derivatives.
- `tck`: A tuple containing the knots, B-spline coefficients, and degree of the spline.

Exceptions

- `std::runtime_error`: Thrown if FITPACK encounters any errors.

template <class E, class... Args>

`auto xt::interpolate::splder(const xexpression<E> &x, const std::tuple<Args...> &tck, int nu = 1, int
ext = 0)`

Compute the spline representation of the derivative of a given spline.

Return A tuple containing the vector of knots, \mathbf{t} , the B-spline coefficients, \mathbf{c} , and the degree of the spline, k

- nu for the evaluated derivative. It is not recommended to manually adjust any of these values.

Parameters

- \mathbf{x} : An array of points at which to evaluate the derivative.
- \mathbf{tck} : A tuple containing the knots, B-spline coefficients, and degree of the spline.
- nu : Order of derivative to evaluate.
- ext : Controls low-level behaviour during derivation.
 - If 0, the spline is extrapolated from the end spans for points not in the support.
 - If 1, the spline evaluates to 0 at those points.
 - If 2, the function will throw.

Exceptions

- `std::runtime_error`: Thrown if $0 \leq n \leq k$ does not hold, where k is the order of the spline.
- `std::runtime_error`: Thrown if FITPACK encounters any errors.

X

xt::interpolate::spalde (C++ function), 6
xt::interpolate::splder (C++ function), 7
xt::interpolate::splev (C++ function), 5
xt::interpolate::splint (C++ function), 6
xt::interpolate::splrep (C++ function), 5
xt::interpolate::sproot (C++ function), 6