
xtensor

Johan Mabilie, Sylvain Corlay and Wolf Vollprecht

Apr 26, 2019

INSTALLATION

1	Installation	3
2	Changelog	5
3	Containers and views	25
4	Readers and writers	27

Multi-dimensional arrays with broadcasting and lazy computing.

xtensor is a C++ library meant for numerical analysis with multi-dimensional array expressions.

xtensor provides

- an extensible expression system enabling **lazy broadcasting**.
- an API following the idioms of the **C++ standard library**.
- tools to manipulate array expressions and build upon *xtensor*.

INSTALLATION

Although `xtensor` is a header-only library, we provide standardized means to install it, with package managers or with `cmake`.

Besides the `xtensor` headers, all these methods place the `cmake` project configuration file in the right location so that third-party projects can use `cmake`'s `find_package` to locate `xtensor` headers.

1.1 Using the conda package

A package for `xtensor` is available on the conda package manager.

```
conda install -c conda-forge xtensor
```

1.2 Using the Debian package

A package for `xtensor` is available on Debian.

```
sudo apt-get install xtensor-dev
```

1.3 Using the Spack package

A package for `xtensor` is available on the Spack package manager.

```
spack install xtensor
spack load --dependencies xtensor
```

1.4 From source with `cmake`

You can also install `xtensor` from source with `cmake`. This requires that you have the `xtl` library installed on your system. On Unix platforms, from the source directory:

```
mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX=path_to_prefix ..
make install
```

On Windows platforms, from the source directory:

```
mkdir build
cd build
cmake -G "NMake Makefiles" -DCMAKE_INSTALL_PREFIX=path_to_prefix ..
nmake
nmake install
```

`path_to_prefix` is the absolute path to the folder where `cmake` searches for dependencies and installs libraries. `xtensor` installation from `cmake` assumes this folder contains `include` and `lib` subfolders.

See the build-options section for more details about `cmake` options.

1.5 Including `xtensor` in your project

The different packages of `xtensor` are built with `cmake`, so whatever the installation mode you choose, you can add `xtensor` to your project using `cmake`:

```
find_package(xtensor REQUIRED)
target_include_directories(your_target PUBLIC ${xtensor_INCLUDE_DIRS})
target_link_libraries(your_target PUBLIC xtensor)
```


CHANGELOG

2.1 0.20.4

- Buffer adaptor default constructor #1524

2.2 0.20.3

- Fix xbuffer adaptor #1523

2.3 0.20.2

- Fixed broadcast linear assign #1493
- Fixed do_stirdes_match #1497
- Removed unused capture #1499
- Upgraded to xtl 0.6.2 #1502
- Added missing methods in xshared_expression #1503
- Fixed iterator types of xcontainer #1504
- Typo correction in external-structure.rst #1505
- Added extension base to adaptors #1507
- Fixed shared expression iterator methods #1509
- Strided view fixes #1512
- Improved range documentation #1515
- Fixed ravel and flatten implementation #1511
- Fixed xfixed_adaptor temporary assign #1516
- Changed struct -> class in xiterator_adaptor #1513
- Fixed argmax for expressions with strides 0 #1519
- Add has_linear_assign to sdynamic_view #1520

2.4 0.20.1

- Add a test for mimetype rendering and fix forward declaration #1490
- Fix special case of view iteration #1491

2.5 0.20.0

2.5.1 Breaking changes

- Removed `xmasked_value` and `promote_type_t` #1389
- Removed deprecated type `slice_vector` #1459
- Upgraded to `xtl` 0.6.1 #1468
- Added `keep_dims` option to reducers #1474
- `do_strides_match` now accept an addition base stride value #1479

2.5.2 Other changes

- Add `partition`, `argpartition` and `median` #991
- Fix tets on `avx512` #1410
- Implemented `xcommon_tensor_t` with tests #1412
- Code reorganization #1416
- `reshape` now accepts `initializer_list` parameter #1417
- Improved documentation #1419
- Fixed `noexcept` specifier #1418
- `view` now accepts `lvalue` slices #1420
- Removed warnings #1422
- Added `reshape` member to `xgenerator` to make `arange` more flexible #1421
- Add `std::decay_t` to `shape_type` in `strided view` #1425
- Generic `reshape` for `xgenerator` #1426
- Fix out of bounds accessing in `xview::compute_strides` #1437
- Added quick reference section to documentation #1438
- Improved getting started `CMakeLists.txt` #1440
- Added periodic indices #1430
- Added build section to narrative documentation #1442
- Fixed `linspace` corner case #1443
- Fixed `type-o` in documentation #1446
- Added `xt::xpad` #1441
- Added warning in `resize` documentation #1447

- Added `in_bounds` method #1444
- `xstrided_view_base` is now a CRTP base class #1453
- Turned `xfunctor_applier_base` into a CRTP base class #1455
- Removed out of bound access in `data_offset` #1456
- Added `xaccessible` base class #1451
- Refactored `operator[]` #1460
- Splitted `xaccessible` #1461
- Refactored `size` #1462
- Implemented `nanvar` and `nanstd` with tests #1424
- Removed warnings #1463
- Added `periodic` and `in_bounds` method to `xoptional_assembly_base` #1464
- Updated documentation according to last changes #1465
- Fixed `flatten_sort_result_type` #1470
- Fixed `unique` with expressions not defining `temporary_type` #1472
- Fixed `xstrided_view_base` constructor #1473
- Avoid signed integer overflow in integer printer #1475
- Fixed `xview::inner_backstrides_type` #1480
- Fixed compiler warnings #1481
- `slice_implementation_getter` now forwards its `lice` argument #1486
- `linspace` can now be reshaped #1488

2.6 0.19.4

- Add missing include #1391
- Fixes in `xfunctor_view` #1393
- Add tests for `xfunctor_view` #1395
- Add `empty` method to `fixed_shape` #1396
- Add accessors to slice members #1401
- Allow adaptors on shared pointers #1218
- Fix `eye` with negative index #1406
- Add documentation for shared pointer adaptor #1407
- Add `nanmean` function #1408

2.7 0.19.3

- Fix `arange` #1361.
- Adaptors for C stack-allocated arrays #1363.

- Add support for optionals in `conditional_ternary` #1365.
- Add tests for ternary operator on `xoptionals` #1368.
- Enable ternary operation for a mix of `xoptional<value>` and `value` #1370.
- `reduce` now accepts a single reduction function #1371.
- Implemented `share` method #1372.
- Documentation of `shared` improved #1373.
- `make_lambda_xfunction` more generic #1374.
- `minimum/maximum` for `xoptional` #1378.
- Added missing methods in `uvector` and `svector` #1379.
- Clip `xoptional_assembly` #1380.
- Improve `gtest` `cmake` #1382.
- Implement ternary operator for scalars #1385.
- Added missing `at` method in `uvector` and `svector` #1386.
- Fixup binder environment #1387.
- Fixed `resize` and `swap` of `svector` #1388.

2.8 0.19.2

- Enable CI for C++17 #1324.
- Fix assignment of masked views #1328.
- Set `CMAKE_CXX_STANDARD` instead of `CMAKE_CXX_FLAGS` #1330.
- Allow specifying traversal order to `argmin` and `argmax` #1331.
- Update section on differences with NumPy #1336.
- Fix accumulators for shapes containing 1 #1337.
- Decouple `XTENSOR_DEFAULT_LAYOUT` and `XTENSOR_DEFAULT_TRAVERSAL` #1339.
- Prevent ambiguity with `xsimd::reduce` #1343.
- Require `xtl` 0.5.3 #1346.
- Use concepts instead of `SFINAE` #1347.
- Document good practice for `xtensor`-based API design #1348.
- Fix rich display of tensor expressions #1353.
- Fix `xview` on fixed tensor #1354.
- Fix issue with `keep_slice` in case of `dynamic_view` on `view` #1355.
- Prevent installation of `gtest` artifacts #1357.

2.9 0.19.1

- Add string specialization to `lexical_cast` #1281.
- Added HDF5 reference for `xtensor-io` #1284.
- Fixed view index remap issue #1288.
- Fixed gcc 8.2 deleted functions #1289.
- Fixed reducer for 0d input #1292.
- Fixed `check_element_index` #1295.
- Added comparison functions #1297.
- Add some tests to ensure chrono works with xexpressions #1272.
- Refactor `functor_view` #1276.
- Documentation improved #1302.
- Implementation of shift operators #1304.
- Make functor adaptor stepper work for proxy specializations #1305.
- Replaced `auto&` with `auto&&` in `assign_to` #1306.
- Fix namespace in `xview_utils.hpp` #1308.
- Introducing `flatten_indices` and `unravel_indices` #1300.
- Default layout parameter for `ravel` #1311.
- Fixed `xvie_stepper` #1317.
- Fixed assignment of view on view #1314.
- Documented indices #1318.
- Fixed shift operators return type #1319.

2.10 0.19.0

2.10.1 Breaking changes

- Upgraded to `xtl 0.5` #1275.

2.10.2 Other changes

- Removed type-o in docs, minor code style consistency update #1255.
- Removed most of the warnings #1261.
- Optional bitwise fixed #1263.
- Prevent macro expansion in `std::max` #1265.
- Update `numpy.rst` #1267.
- Update `getting_started.rst` #1268.
- keep and drop `step_size` fixed #1270.

- Fixed typo in `xadapt` #1277.
- Fixed typo #1278.

2.11 0.18.3

- Exporting optional dependencies #1253.
- 0-D HTML rendering #1252.
- Include `nlohmann_json` in `xio` for mime bundle repr #1251.
- Fixup `xview` scalar assignment #1250.
- Implemented *from_indices* #1240.
- `xtensor_forward.hpp` cleanup #1243.
- default layout-type for *unravel_from_strides* and *unravel_index* #1239.
- `xfunction` iterator fix #1241.
- `xstepper` fixes #1237.
- `print_options` io manipulators #1231.
- Add syntactic sugar for reducer on single axis #1228.
- Added view vs. adapt benchmark #1229.
- added precisions to the installation instructions #1226.
- removed data interface from dynamic view #1225.
- add `xio` docs #1223.
- Fixup `xview` assignment #1216.
- documentation updated to be consistent with last changes #1214.
- prevents macro expansion of `std::max` #1213.
- Fix minor typos #1212.
- Added missing assign operator in `xstrided_view` #1210.
- `argmax` on axis with single element fixed #1209.

2.12 0.18.2

- expression tag system fixed #1207.
- optional extension for generator #1206.
- optional extension for `xview` #1205.
- optional extension for `xstrided_view` #1204.
- optional extension for reducer #1203.
- optional extension for `xindex_view` #1202.
- optional extension for `xfunctor_view` #1201.
- optional extension for broadcast #1198.

- extension API and code cleanup #1197.
- `xscalar` optional refactoring #1196.
- Extension mechanism #1192.
- Many small fixes #1191.
- Slight refactoring in `step_size` logic #1188.
- Fixup call of `const` overload in assembly storage #1187.

2.13 0.18.1

- Fixup `xio` forward declaration #1185.

2.14 0.18.0

2.14.1 Breaking changes

- Assign and `trivial_broadcast` refactoring #1150.
- Moved array manipulation functions (`transpose`, `ravel`, `flatten`, `trim_zeros`, `squeeze`, `expand_dims`, `split`, `atleast_Nd`, `atleast_1d`, `atleast_2d`, `atleast_3d`, `flip`) from `xstrided_view.hpp` to `xmanipulation.hpp` #1153.
- iterator API improved #1155.
- Fixed `where` and `nonzero` function behavior to mimic the behavior from NumPy #1157.
- `xsimd` and functor refactoring #1173.

2.14.2 New features

- Implement `rot90` #1153.
- Implement `argwhere` and `flatnonzero` #1157.
- Implemented `xexpression_holder` #1164.

2.14.3 Other changes

- Warnings removed #1159.
- Added missing include #1162.
- Removed unused type alias in `xmath/average` #1163.
- Slices improved #1168.
- Fixed `xdrop_slice` #1181.

2.15 0.17.4

- perfect forwarding in `xoptional_function` constructor #1101.
- fix issue with `base_simd` #1103.
- `XTENSOR_ASSERT` fixed on Windows #1104.
- Implement `xmasked_value` #1032.
- Added `setdiff1d` using `stl` interface #1109.
- Added test case for `setdiff1d` #1110.
- Added missing reference to `diff` in `From numpy to xtensor` section #1116.
- Add `amax` and `amin` to the documentation #1121.
- `histogram` and `histogram_bin_edges` implementation #1108.
- Added `numpy` comparison for `interp` #1111.
- Allow multiple return type reducer functions #1113.
- Fixes average bug + adds `Numpy` based tests #1118.
- Static `xfunction` cache for fixed sizes #1105.
- Add negative reshaping axis #1120.
- Updated `xmasked_view` using `xmasked_value` #1074.
- Clean documentation for views #1131.
- Build with `xsimd` on Windows fixed #1127.
- Implement `mime_bundle_repr` for `xmasked_view` #1132.
- Modify `shuffle` to use identical algorithms for any number of dimensions #1135.
- Warnings removal on windows #1139.
- Add permutation function to `random` #1141.
- `xfunction_iterator` permutation #933.
- Add `bincount` to `xhistogram` #1140.
- Add contiguous iterable base class and remove layout param from storage iterator #1057.
- Add `storage_iterator` to view and strided view #1045.
- Removes `data_element` from `xoptional` #1137.
- `xtensor` default constructor and scalar assign fixed #1148.
- Add `resize` / `reshape` to `xfixed_container` #1147.
- Iterable refactoring #1149.
- `inner_strides_type` imported in `xstrided_view` #1151.

2.16 0.17.3

- `xslice` fix #1099.
- added missing `static_layout` in `xmasked_view` #1100.

2.17 0.17.2

- Add experimental TBB support for parallelized multicore assign #948.
- Add inline statement to all functions in xnpv #1097.
- Fix strided assign for certain assignments #1095.
- CMake, remove gtest warnings #1085.
- Add conversion operators to slices #1093.
- Add optimization to unchecked accessors when contiguous layout is known #1060.
- Speedup assign by computing any layout on vectors #1063.
- Skip resizing for fixed shapes #1072.
- Add xsimd apply to xcomplex functors (conj, norm, arg) #1086.
- Propagate contiguous layout through views #1039.
- Fix C++17 ambiguity for GCC 7 #1081.
- Correct shape type in argmin, fix svector growth #1079.
- Add interp function to xmath #1071.
- Fix valgrind warnings + memory leak in xadapt #1078.
- Remove more clang warnings & errors on OS X #1077.
- Add move constructor from xtensor <-> xarray #1051.
- Add global support for negative axes in reducers/accumulators allow multiple axes in average #1010.
- Fix reference usage in xio #1076.
- Remove occurrences of std::size_t and double #1073.
- Add missing parantheses around min/max for MSVC #1061.

2.18 0.17.1

- Add std namespace to size_t everywhere, remove std::copysign for MSVC #1053.
- Fix (wrong) bracket warnings for older clang versions (e.g. clang 5 on OS X) #1050.
- Fix strided view on view by using std::addressof #1049.
- Add more adapt functions and shorthands #1043.
- Improve CRTP base class detection #1041.
- Fix rebind container ambiguous template for C++17 / GCC 8 regression #1038.
- Fix functor return value #1035.

2.19 0.17.0

2.19.1 Breaking changes

- Changed strides to `std::ptrdiff_t` #925.
- Renamed `count_nonzeros` in `count_nonzero` #974.
- homogenize `xfixed` constructors #970.
- Improve `random::choice` #1011.

2.19.2 New features

- add `signed char` to `npv` deserialization format #1017.
- `simd` assignment now requires convertible types instead of same type #1000.
- shared expression and automatic `xclosure` detection #992.
- average function #987.
- added `simd` support for complex #985.
- `argsort` function #977.
- propagate fixed shape #922.
- added `xdrop_slice` #972.
- added doc for `xmasked_view` #971.
- added `xmasked_view` #969.
- added `dynamic_view` #966.
- added ability to use negative indices in `keep slice` #964.
- added an easy way to create lambda expressions, square and cube #961.
- `noalias` on `rvalue` #965.

2.19.3 Other changes

- `xshared_expression` fixed #1025.
- fix `make_xshared` #1024.
- add tests to evaluate shared expressions #1019.
- fix `where` on `xview` #1012.
- basic usage replaced with getting started #1004.
- avoided installation failure in absence of `nlohmann_json` #1001.
- code and documentation clean up #998.
- removed `g++` “pedantic” compiler warnings #997.
- added missing header in `basic_usage.rst` #996.
- warning pass #990.

- added missing include in `xview` #989.
- added missing `<map>` include #983.
- `xislice` refactoring #962.
- added missing operators to `noalias` #932.
- cmake fix for Intel compiler on Windows #951.
- fixed `xsimd` abs deduction #946.
- added `islice` example to view doc #940.

2.20 0.16.4

- removed usage of `std::transform` in `assign` #868.
- add strided assignment #901.
- `simd` activated for conditional ternary functor #903.
- `xstrided_view` split #905.
- assigning an expression to a view throws if it has more dimensions #910.
- faster random #913.
- `xoptional_assembly_base` storage type #915.
- new tests and warning pass #916.
- norm immediate reducer #924.
- add `reshape_view` #927.
- fix immediate reducers with 0 strides #935.

2.21 0.16.3

- `simd` on mathematical functions fixed #886.
- `fill` method added to containers #887.
- access with more arguments than dimensions #889.
- unchecked method implemented #890.
- `fill` method implemented in view #893.
- documentation fixed and warnings removed #894.
- negative slices and new range syntax #895.
- `xview_stepper` with implicit `xt::all` bug fix #899.

2.22 0.16.2

- Add include of `xview.hpp` in example #884.
- Remove FS identifier #885.

2.23 0.16.1

- Workaround for Visual Studio Bug #858.
- Fixup example notebook #861.
- Prevent expansion of min and max macros on Windows #863.
- Renamed `m_data` to `m_storage` #864.
- Fix regression with respect to random access stepping with views #865.
- Remove use of CS, DS and ES qualifiers for Solaris builds #866.
- Removal of precision type #870.
- Make json tests optional, bump xtl/xsimd versions #871.
- Add more benchmarks #876.
- Forbid simd fixed #877.
- Add more asserts #879.
- Add missing `batch_bool` typedef #881.
- `simd_return_type` hack removed #882.
- Removed test guard and fixed dimension check in `xscalar` #883.

2.24 0.16.0

2.24.1 Breaking changes

- `data` renamed in `storage`, `raw_data` renamed in `data` #792.
- Added layout template parameter to `xstrided_view` #796.
- Remove equality operator from `stepper` #824.
- `dynamic_view` renamed in `strided_view` #832.
- `xtensorf` renamed in `xtensor_fixed` #846.

2.24.2 New features

- Added strided view selector #765.
- Added `count_nonzeros` #781.
- Added implicit conversion to scalar in `xview` #788.
- Added tracking allocators to `xutils.hpp` #789.
- `xindexslice` and `shuffle` function #804.
- Allow `xadapt` with dynamic layout #816.
- Added `xtensorf` initialization from C array #819.
- Added policy to allocation tracking for throw option #820.
- Free function `empty` for construction from shape #827.

- Support for JSON serialization and deserialization of xtensor expressions #830.
- Add `trapz` function #837.
- Add `diff` and `trapz(y, x)` functions #841.

2.24.3 Other changes

- Added fast path for specific assigns #767.
- Renamed internal macros to prevent collisions #772.
- `dynamic_view` unwrapping #775.
- `xreducer_stepper` copy semantic fixed #785.
- `xfunction` copy constructor fixed #787.
- warnings removed #791.
- `xscalar_stepper` fixed #802.
- Fixup `xadapt` on const pointers #809.
- Fix in owning buffer adaptors #810.
- Macros fixup #812.
- More fixes in `xadapt` #813.
- Mute unused variable warning #815.
- Remove comparison of steppers in assign loop #823.
- Fix reverse iterators #825.
- gcc-8 fix for template method calls #833.
- refactor benchmarks for upcoming release #842.
- `flip` now returns a view #843.
- initial warning pass #850.
- Fix warning on `diff` function #851.
- `xsimd` assignment fixed #852.

2.25 0.15.9

- missing layout method in `xfixed` #777.
- fixed uninitialized backstrides #774.
- update xtensor-blas in binder #773.

2.26 0.15.8

- comparison operators for slices #770.
- use default-assignable layout for strided views. #769.

2.27 0.15.7

- nan related functions #718.
- return types fixed in dynamic view helper #722.
- xview on constant expressions #723.
- added decays to make const value_type compile #727.
- iterator for constant strided_view fixed #729.
- strided_view on xfunction fixed #732.
- Fixes in xstrided_view #736.
- View semantic (broadcast on assign) fixed #742.
- Compilation prevented when using ellipsis with xview #743.
- Index of xiterator set to shape when reaching the end #744.
- xscalar fixed #748.
- Updated README and related projects #749.
- Perfect forwarding in xfunction and views #750.
- Missing include in xassign.hpp #752.
- More related projects in the README #754.
- Fixed stride computation for xtensorf #755.
- Added tests for backstrides #758.
- Clean up has_raw_data ins strided view #759.
- Switch to ptrdiff_t for slices #760.
- Fixed xview strides computation #762.
- Additional methods in slices, required for xframe #764.

2.28 0.15.6

- zeros, ones, full and empty_like functions #686.
- squeeze view #687.
- bitwise shift left and shift right #688.
- ellipsis, unique and trim functions #689.
- xview iterator benchmark #696.
- optimize stepper increment #697.
- minmax reducers #698.
- where fix with SIMD #704.
- additional doc for scalars and views #705.
- mixed arithmetic with SIMD #713.
- broadcast fixed #717.

2.29 0.15.5

- assign functions optimized #650.
- transposed view fixed #652.
- exceptions refactoring #654.
- performances improved #655.
- view data accessor fixed #660.
- new dynamic view using variant #656.
- alignment added to fixed xtensor #659.
- code cleanup #664.
- xtensorf and new dynamic view documentation #667.
- qualify namespace for compute_size #665.
- make xio use dynamic_view instead of view #662.
- transposed view on any expression #671.
- docs typos and grammar plus formatting #676.
- index view test assertion fixed #680.
- flatten view #678.
- handle the case of pointers to const element in xadapt #679.
- use quotes in #include statements for xtl #681.
- additional constructors for svector #682.
- removed test_xsemantics.hpp from test CMakeLists #684.

2.30 0.15.4

- fix gcc-7 error w.r.t. the use of assert #648.

2.31 0.15.3

- add missing headers to cmake installation and tests #647.

2.32 0.15.2

- xshape implementation #572.
- xfixed container #586.
- protected xcontainer::derived_cast #627.
- const reference fix #632.
- xgenerator access operators fixed #643.

- contiguous layout optimization #645.

2.33 0.15.1

- `xarray_adaptor` fixed #618.
- `xtensor_adaptor` fixed #620.
- fix in `xreducer` steppers #622.
- documentation improved #621. #623. #625.
- warnings removed #624.

2.34 0.15.0

2.34.1 Breaking changes

- change `reshape` to `resize`, and add throwing `reshape` #598.
- moved to modern `cmake` #611.

2.34.2 New features

- `unravel` function #589.
- random access iterators #596.

2.34.3 Other changes

- upgraded to google/benchmark version 1.3.0 #583.
- `XTENSOR_ASSERT` renamed into `XTENSOR_TRY`, new `XTENSOR_ASSERT` #603.
- `adapt` fixed #604.
- VC14 warnings removed #608.
- `xfunctor_iterator` is now a random access iterator #609.
- removed `old-style-cast` warnings #610.

2.35 0.14.1

2.35.1 New features

- `sort`, `argmin` and `argmax` #549.
- `xscalar_expression_tag` #582.

2.35.2 Other changes

- accumulator improvements #570.
- benchmark cmake fixed #571.
- allocator_type added to container interface #573.
- allow conda-forge as fallback channel #575.
- arithmetic mixing optional assemblies and scalars fixed #578.
- arithmetic mixing optional assemblies and optionals fixed #579.
- operator== restricted to xtensor and xoptional expressions #580.

2.36 0.14.0

2.36.1 Breaking changes

- xadapt renamed into adapt #563.
- Naming consistency #565.

2.36.2 New features

- add random::choice #547.
- evaluation strategy and accumulators. #550.
- modulus operator #556.
- adapt: default overload for 1D arrays #560.
- Move semantic on adapt #564.

2.36.3 Other changes

- optional fixes to avoid ambiguous calls #541.
- narrative documentation about xt::adapt #544.
- xfunction refactoring #545.
- SIMD acceleration for AVX fixed #557.
- allocator fixes #558. #559.
- return type of view::strides() fixed #568.

2.37 0.13.2

- Support for complex version of isclose #512.
- Fixup static layout in xstrided_view #536.
- xexpression::operator[] now take support any type of sequence #537.

- Fixing `xinfo` issues for Visual Studio. #529.
- Fix const-correctness in `xstrided_view`. #526.

2.38 0.13.1

- More general floating point type #518.
- Do not require functor to be passed via rvalue reference #519.
- Documentation improved #520.
- Fix in `xreducer` #521.

2.39 0.13.0

2.39.1 Breaking changes

- The API for `xbuffer_adaptor` has changed. The template parameter is the type of the buffer, not just the value type #482.
- Change `edge_items` print option to `edgeitems` for better numpy consistency #489.
- `xtensor` now depends on `xtl` version `~0.3.3` #508.

2.39.2 New features

- Support for parsing the `npz` file format #465.
- Creation of optional expressions from value and boolean expressions (optional assembly) #496.
- Support for the explicit cast of expressions with different value types #491.

2.39.3 Other changes

- Addition of broadcasting bitwise operators #459.
- More efficient optional expression system #467.
- Migration of benchmarks to the Google benchmark framework #473.
- Container semantic and adaptor semantic merged #475.
- Various fixes and improvements of the strided views #480. #481.
- Assignment now performs basic type conversion #486.
- Workaround for a compiler bug in Visual Studio 2017 #490.
- MSVC 2017 workaround #492.
- The `size()` method for containers now returns the total number of elements instead of the buffer size, which may differ when the smallest stride is greater than 1 #502.
- The behavior of `linspace` with integral types has been made consistent with numpy #510.

2.40 0.12.1

- Fix issue with slicing when using heterogeneous integral types #451.

2.41 0.12.0

2.41.1 Breaking changes

- `xtensor` now depends on `xtl` version *0.2.x* #421.

2.41.2 New features

- `xtensor` has an optional dependency on `xsimd` for enabling simd acceleration #426.
- All expressions have an additional safe access function (`at`) #420.
- `norm` functions #440.
- `closure_pointer` used in iterators returning temporaries so their `operator->` can be correctly defined #446.
- `expressions` tags added so `xtensor` expression system can be extended #447.

2.41.3 Other changes

- Preconditions and exceptions #409.
- `isclose` is now symmetric #411.
- concepts added #414.
- narrowing cast for mixed arithmetic #432.
- `is_xexpression` concept fixed #439.
- `void_t` implementation fixed for compilers affected by C++14 defect CWG 1558 #448.

2.42 0.11.3

- Fixed bug in length-1 statically dimensioned tensor construction #431.

2.43 0.11.2

- Fixup compilation issue with latest clang compiler. (missing `constexpr` keyword) #407.

2.44 0.11.1

- Fixes some warnings in julia and python bindings

2.45 0.11.0

2.45.1 Breaking changes

- `xbegin / xend`, `xcbegin / xcend`, `xrbegin / xrend` and `xcrbegin / xcrend` methods replaced with classical `begin / end`, `cbegin / cend`, `rbegin / rend` and `crbegin / crend` methods. Old `begin / end` methods and their variants have been removed. [#370](#).
- `xview` now uses a `const` stepper when its underlying expression is `const`. [#385](#).

2.45.2 Other changes

- `xview` copy semantic and move semantic fixed. [#377](#).
- `xoptional` can be implicitly constructed from a scalar. [#382](#).
- build with Emscripten fixed. [#388](#).
- STL version detection improved. [#396](#).
- Implicit conversion between signed and unsigned integers fixed. [#397](#).

CONTAINERS AND VIEWS

Containers are in-memory expressions that share a common implementation of most of the methods of the xexpression API. The final container classes (`xarray`, `xtensor`) mainly implement constructors and value semantic, most of the xexpression API is actually implemented in `xstrided_container` and `xcontainer`.

3.1 layout

Defined in `xtensor/xlayout.hpp`

Warning: doxygenenum: Cannot find enum “`xt::layout_type`” in doxygen xml output for project “xtensor” from directory: `../xml`

Warning: doxygenfunction: Cannot find function “`xt::compute_layout`” in doxygen xml output for project “xtensor” from directory: `../xml`

3.2 xcontainer

Defined in `xtensor/xcontainer.hpp`

Warning: doxygenclass: Cannot find class “`xt::xcontainer`” in doxygen xml output for project “xtensor” from directory: `../xml`

3.3 xstrided_container

Defined in `xtensor/xcontainer.hpp`

Warning: doxygenclass: Cannot find class “`xt::xstrided_container`” in doxygen xml output for project “xtensor” from directory: `../xml`

READERS AND WRITERS

Read && writers are in-memory expressions that share a common implementation of most of the methods of the xexpression API. The final container classes (`xarray`, `xtensor`) mainly implement constructors and value semantic, most of the xexpression API is actually implemented in `xstrided_container` and `xcontainer`.

4.1 layout

Defined in `xtensor/xcontainer.hpp`

Warning: doxygenenum: Cannot find enum “`xt::layout_type`” in doxygen xml output for project “xtensor” from directory: `../xml`

Warning: doxygenfunction: Cannot find function “`xt::compute_layout`” in doxygen xml output for project “xtensor” from directory: `../xml`

4.2 node1

Defined in `cyber/node/node.h`

Warning: doxygenclass: Cannot find class “`apollo::cyber::Node`” in doxygen xml output for project “xtensor” from directory: `../xml`

4.3 xstrided_container

Defined in `xtensor/xcontainer.hpp`

Warning: doxygenclass: Cannot find class “`xt::xstrided_container`” in doxygen xml output for project “xtensor” from directory: `../xml`