NemoMath - Bug #1510
trunk doesn`t build on MacOS Lion 64bit

05/27/2013 01:57 PM - Anonymous

Status:	Resolved	**Start date:**	05/27/2013
Priority:	Normal	**Due date:**	
Assignee:	M. Rolf	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	NemoMath 0.4		

Description

See: https://ci.cor-lab.org/view/macos/job/nemomath-trunk-macos/label=MAC_OS_lion_64bit/12/console

```
In file included from
/Users/jenkins/workspace/nemomath-trunk-macos/label/MAC_OS_lion_64bit/nemomath/examples/ExampleMappingArithmetics.cpp:2: In
s.cpp:2:
In file included from
/Users/jenkins/workspace/nemomath-trunk-macos/label/MAC_OS_lion_64bit/nemomath/examples/../src/nemo/MatrixMath.h:4:
In file included from
/Users/jenkins/workspace/nemomath-trunk-macos/label/MAC_OS_lion_64bit/nemomath/src/nemo/Matrix.h:12:
In file included from /usr/local/include/eigen3/Eigen/Core:248:
/usr/local/include/eigen3/Eigen/src/Core/util/Memory.h:718:23: error: no member named 'forward' in namespace 'std'
    ::new(p) T(std::forward<Args>(args)...);
             ~~~~~^
/usr/local/include/eigen3/Eigen/src/Core/util/Memory.h:718:31: error: 'Args' does not refer to a value
    ::new(p) T(std::forward<Args>(args)...);
                          ^
...
```

Maybe just a missing / wrong header include?

Associated revisions

Revision 294 - 08/24/2013 03:31 PM - anordman

Adds += and -= operator to Vector and Matrix

- libc++ uses += and -= operator for sort
- thanks to f. sowade for digging into this

refs #1510

History

#1 - 05/27/2013 02:26 PM - M. Rolf

We had a similar problem before. Likely solution: add " -stdlib=libc++" to the CXX flags.

See
http://marshall.calepin.co/llvmclang-and-standard-libraries-on-mac-os-x.html

Actually, this flag should have been there already... maybe it has been dropped in some CI refactoring?

#2 - 05/27/2013 02:46 PM - Anonymous

Adding -stdlib=libc++ to the compiler flags now produces different, much earlier errors:
https://ci.cor-lab.org/view/GradSchool/job/nemomath-trunk-macos/label=MAC_OS_lion_64bit/13/console

```
In file included from
/Users/jenkins/workspace/nemomath-trunk-macos/label/MAC_OS_lion_64bit/nemomath/examples/ExampleCollections.cpp:2:
/Users/jenkins/workspace/nemomath-trunk-macos/label/MAC_OS_lion_64bit/nemomath/examples/../src/nemo/Vector.h:406:15: error: no
member named 'max' in namespace 'std'
                using std::max;
                    ~~~~~^
/Users/jenkins/workspace/nemomath-trunk-macos/label/MAC_OS_lion_64bit/nemomath/examples/../src/nemo/Vector.h:410:15: error: no
member named 'max' in namespace 'std'
                using std::max;
                    ~~~~~^
/Users/jenkins/workspace/nemomath-trunk-macos/label/MAC_OS_lion_64bit/nemomath/examples/../src/nemo/Vector.h:414:15: error: no
member named 'max' in namespace 'std'
                using std::max;
                    ~~~~~^
/Users/jenkins/workspace/nemomath-trunk-macos/label/MAC_OS_lion_64bit/nemomath/examples/../src/nemo/Vector.h:418:15: error: no
member named 'min' in namespace 'std'
                using std::min;
                    ~~~~~^
/Users/jenkins/workspace/nemomath-trunk-macos/label/MAC_OS_lion_64bit/nemomath/examples/../src/nemo/Vector.h:422:15: error: no
member named 'min' in namespace 'std'
                using std::min;
                    ~~~~~^
/Users/jenkins/workspace/nemomath-trunk-macos/label/MAC_OS_lion_64bit/nemomath/examples/../src/nemo/Vector.h:426:15: error: no
member named 'min' in namespace 'std'
                using std::min;
                    ~~~~~^
/Users/jenkins/workspace/nemomath-trunk-macos/label/MAC_OS_lion_64bit/nemomath/examples/../src/nemo/Vector.h:1117:9: error: implicit
instantiation of undefined template 'std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> >'
        out << std::string("[");
...
```

Seems to be a similar issue though.

#3 - 05/27/2013 03:56 PM - Anonymous

Now also setting linker flags accordingly. Next error:

```
In file included from /Users/jenkins/workspace/nemomath-trunk-macos/label/MAC_OS_lion_64bit/nemomath/examples/ExampleVector.cpp:2:
In file included from /usr/bin/../lib/c++/v1/iostream:38:
In file included from /usr/bin/../lib/c++/v1/ios:216:
In file included from /usr/bin/../lib/c++/v1/__locale:15:
In file included from /usr/bin/../lib/c++/v1/string:434:
/usr/bin/../lib/c++/v1/algorithm:3585:17: error: no viable overloaded '+='
            __m += __delta;
            ~~~ ^  ~~~~~~~
```

#4 - 05/27/2013 04:13 PM - M. Rolf

- Status changed from New to In Progress

Probably, it fits this description:
http://lists.cs.uiuc.edu/pipermail/llvmbugs/2010-July/013919.html

"Clang is correct to reject this code; GCC shouldn't allow it to compile." ....

http://clang.llvm.org/compatibility.html#dep_lookup

#5 - 08/12/2013 03:50 PM - Anonymous

- Status changed from In Progress to Feedback
- Assignee set to M. Rolf

Any ideas how to proceed?

#6 - 08/24/2013 10:30 AM - Anonymous

Doesn't seem to be a problem with clang, since it builds without problems using clang++ on precise64.
Occurs when also using libc++ instead of libstdc++.

#7 - 08/24/2013 10:42 AM - Anonymous

- Status changed from Feedback to In Progress

MathVector<T>::Iterator is missing the += operator which is used by libc++ sort.

#8 - 08/24/2013 10:53 AM - Anonymous

- Status changed from In Progress to New

#9 - 08/24/2013 04:04 PM - Anonymous

- Status changed from New to Resolved
- % Done changed from 0 to 100

Resolved. Tests not compiling on MacOS now, see #1600