Subject: Eigen updated
Posted by koldo on Sat, 09 May 2020 09:03:20 GMT
View Forum Message <> Reply to Message

From 2018 Eigen algebra library had no new version, and since interesting improvements have been accumulated, we have updated to the last master commit.
This commit has not given any compilation or functioning problems. However, please check your applications just in case.

Subject: Re: Eigen updated
Posted by Novo on Sat, 09 May 2020 13:17:44 GMT
View Forum Message <> Reply to Message

koldo wrote on Sat, 09 May 2020 05:03From 2018 Eigen algebra library had no new version, and since interesting improvements have been accumulated, we have updated to the last master commit.
This commit has not given any compilation or functioning problems. However, please check your applications just in case.
And reference/Eigen_demo is broken with CLANG on Linux.

Subject: Re: Eigen updated
Posted by koldo on Sat, 09 May 2020 17:26:18 GMT
View Forum Message <> Reply to Message

Quote:And reference/Eigen_demo is broken with CLANG on Linux.Bad luck. I do not have CLANG in Linux. I will try to install it. With GCC in Linux and with CLANG in Windows there is no problem. Thank you for reporting.

Subject: Re: Eigen updated
Posted by koldo on Sat, 09 May 2020 17:35:08 GMT
View Forum Message <> Reply to Message

I have just compiled and run Eigen_demo in Ubuntu 18.04 with CLANG and GCC and works perfectly.

Subject: Re: Eigen updated
Posted by Novo on Sat, 09 May 2020 18:08:30 GMT
View Forum Message <> Reply to Message

koldo wrote on Sat, 09 May 2020 13:35I have just compiled and run Eigen_demo in Ubuntu 18.04 with CLANG and GCC and works perfectly.
Ubuntu 20.04

./umk reference Eigen_demo CLANG -bus
Running full rebuild now ...
I was getting this:
BLITZ: eigen_demo.cpp non-linear.cpp fft.cpp
In file included from
/home/ssg/.local/soft/bb-worker/worker/upp/build/.cache/upp.out/Eigen_demo/CLANG.Debug.Debug_Full.Main.Shared/Eigen_demo$blitz.cpp:7:
/home/ssg/.local/soft/bb-worker/worker/upp/build/reference/Eigen_demo/non-linear.cpp:26:9:
error: reference to 'Vector' is ambiguous
static Vector<double> _x, _y;
^
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/Core/Vcont.h:104:7: note: candidate found by name lookup is 'Upp::Vector'
class Vector : public MoveableAndDeepCopyOption< Vector<T> > {
^
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen/src/Core/Matrix.h:549:1: note: candidate found by name lookup is 'Eigen::Vector'
using Vector = Matrix<Type, Size, 1>;
^
In file included from
/home/ssg/.local/soft/bb-worker/worker/upp/build/.cache/upp.out/Eigen_demo/CLANG.Debug.Debug_Full.Main.Shared/Eigen_demo$blitz.cpp:7:
/home/ssg/.local/soft/bb-worker/worker/upp/build/reference/Eigen_demo/non-linear.cpp:46:1:
error: reference to 'Vector' is ambiguous
Vector<double> Thurber_functor::_x, Thurber_functor::_y;
^
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/Core/Vcont.h:104:7: note: candidate found by name lookup is 'Upp::Vector'
class Vector : public MoveableAndDeepCopyOption< Vector<T> > {
^
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen/src/Core/Matrix.h:549:1: note: candidate found by name lookup is 'Eigen::Vector'
using Vector = Matrix<Type, Size, 1>;
^
2 errors generated.
Eigen_demo: 3 file(s) built in (0:12.87), 4291 msecs / file
There were errors. (0:13.04)

Subject: Re: Eigen updated
Posted by Novo on Sat, 09 May 2020 18:15:11 GMT
View Forum Message <> Reply to Message

After full rebuild:
reference/ScatterDraw_Demo
./umk reference ScatterDraw_Demo CLANG -bus
ScatterDraw_Demo.cpp

```
/home/ssg/.local/soft/bb-worker/worker/upp/build/reference/ScatterDraw_Demo/ScatterDraw_Dem
o.cpp:16:2: error: reference to 'Vector' is ambiguous
        Vector<Pointf> s1;
        ^
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/Core/Vcont.h:104:7: note: candidate
found by name lookup is 'Upp::Vector'
class Vector : public MoveableAndDeepCopyOption< Vector<T> > {
      ^
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen/src/Core/Matrix.h:549
:1: note: candidate found by name lookup is 'Eigen::Vector'
using Vector = Matrix<Type, Size, 1>;
^
1 error generated.
ScatterDraw_Demo: 1 file(s) built in (0:01.08), 1088 msecs / file
ScatterDraw: 8 file(s) built in (0:08.75), 1094 msecs / file
There were errors. (0:09.92)
program finished with exit code 1
```

It is still broken ...

Subject: Re: Eigen updated
Posted by Novo on Sat, 09 May 2020 20:59:19 GMT
View Forum Message <> Reply to Message

I double-checked that with TheIDE against GIT rev. 1189b35
The problem is reproducible.

Subject: Re: Eigen updated
Posted by koldo on Sat, 09 May 2020 21:56:48 GMT
View Forum Message <> Reply to Message

Dear Novo

Yes, there was other problem. I forgot that I have also added VectorXd data access to ScatterDraw, so it should have to be also fully checked. Now it is done and, I think it is finally solved. Thank you!

Subject: Re: Eigen updated
Posted by Novo on Sat, 09 May 2020 23:37:14 GMT
View Forum Message <> Reply to Message

No problem.
Do not thank me, thank a build automation system :roll:

Subject: Re: Eigen updated
Posted by koldo on Sun, 10 May 2020 06:56:09 GMT
View Forum Message <> Reply to Message

:lol: From now on I will use it not partially, but massively.

Subject: Re: Eigen updated
Posted by coolman on Fri, 15 May 2020 11:44:26 GMT
View Forum Message <> Reply to Message

Novo wrote on Sat, 09 May 2020 20:15After full rebuild:
...
/home/ssg/.local/soft/bb-worker/worker/upp/build/reference/ScatterDraw_Demo/ScatterDraw_Demo.cpp:16:2: error: reference to 'Vector' is ambiguous
Vector<Pointf> s1;
^
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/Core/Vcont.h:104:7: note: candidate found by name lookup is 'Upp::Vector'
class Vector : public MoveableAndDeepCopyOption< Vector<T> > {
^
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen/src/Core/Matrix.h:549:1: note: candidate found by name lookup is 'Eigen::Vector'
using Vector = Matrix<Type, Size, 1>;
^
1 error generated.
...

It is still broken ...

Hi,

There is similar problem with Array in my personal application. And when you are using SQL the Select is ambiguous too. Build on Ubuntu 18.04 CLANG.

BR, Radek

Subject: Re: Eigen updated
Posted by Didier on Fri, 15 May 2020 16:25:55 GMT
View Forum Message <> Reply to Message

Hello Radek,

I had the same problem last week : it was my application that had problems.
The problem was the use of 'using namespace Upp' in some header files ... which leads to these ambiguity errors

Maybe it's the same for you

Subject: Re: Eigen updated
Posted by koldo on Fri, 15 May 2020 17:39:39 GMT
View Forum Message <> Reply to Message

Didier wrote on Fri, 15 May 2020 18:25Hello Radek,

I had the same problem last week : it was my application that had problems.
The problem was the use of 'using namespace Upp' in some header files ... which leads to these ambiguity errors

Maybe it's the same for youI agree with Didier. Please try to put #include <ScatterDraw/ScatterDraw.h> before using namespace Upp;, and tell us if this solves the problem.

Subject: Re: Eigen updated
Posted by coolman on Fri, 15 May 2020 18:04:07 GMT
View Forum Message <> Reply to Message

koldo wrote on Fri, 15 May 2020 19:39Didier wrote on Fri, 15 May 2020 18:25Hello Radek,

I had the same problem last week : it was my application that had problems.
The problem was the use of 'using namespace Upp' in some header files ... which leads to these ambiguity errors

Maybe it's the same for youI agree with Didier. Please try to put #include <ScatterDraw/ScatterDraw.h> before using namespace Upp;, and tell us if this solves the problem.

Hi,

I don't use 'using namespace Upp' in header files. Only in code files after all includes.

BR, Radek

Subject: Re: Eigen updated
Posted by Didier on Fri, 15 May 2020 20:28:47 GMT
View Forum Message <> Reply to Message

I think it still is the source of you're problem.
In my case I removed ALL the 'using namespace Upp' (in header and in .cpp files) and replaced

them with 'namespace Upp { }'

Subject: Re: Eigen updated
Posted by coolman on Sat, 16 May 2020 06:20:39 GMT
View Forum Message <> Reply to Message

Hi,

Problem was with the include of <ScatterCtrl/ScatterCtrl.h> and <ScatterCtrl/PieCtrl.h>. They were in a wrong place. The 'using namespace Upp;' was correct.

Thanks, Radek

Subject: Re: Eigen updated
Posted by Klugier on Tue, 19 May 2020 22:19:06 GMT
View Forum Message <> Reply to Message

Hello Koldo,

If I may suggest something, I would never use unstable code in the production environment. By introducing nightly build you risk all the problems resulting from this fact. One of that problem is that library can let you down at the most inopportune moment. I think Eigen is one of that libraries with the well establish release cycle and we should respect it. The same thing is true for Upp. You can hit the moment when some important thing doesn't work like we want to. The best example for that is initial GTK 3 port.

Currently, we have 3.3.7 and they are working hard to bring 3.4 or 3.3.8. In my opinion you should be patient and wait for that release. It will happen sooner or later. I highly recommend to revert current master version and back to stable 3.3.7.

Sincerely,
Klugier

Subject: Re: Eigen updated
Posted by koldo on Wed, 20 May 2020 10:56:08 GMT
View Forum Message <> Reply to Message

Dear Klugier

I think you're totally disoriented as to the cause of this problem.

On the one hand, although the problem appeared when updating Eigen, it was actually because of adding a new option, dependent on Eigen, to ScatterDraw.
In other words, this problem would have appeared even if Eigen had not been updated.

Moreover, the update was made immediately after Mirek announced the official release of version 2020.1.

Finally, the problem was minimal and quickly resolved.

Subject: Re: Eigen updated
Posted by Novo on Wed, 20 May 2020 16:42:45 GMT
View Forum Message <> Reply to Message

I'm sorry to say this, but Eigen_demo is broken again. This time after Mirek's commit.

template<> EIGEN_STRONG_INLINE Packet16b pset1<Packet16b>(const bool& from) { return _mm_set1_epi8(static_cast<char>(from)); }

^~~~~~~
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen/src/Core/arch/SSE/PacketMath.h:227:93: error: use of undeclared identifier '_mm_castsi128_ps'; did you mean 'Upp::_mm_castsi128_ps'?
template<> EIGEN_STRONG_INLINE Packet4f pset1frombits<Packet4f>(unsigned int from) { return _mm_castsi128_ps(pset1<Packet4i>(from)); }
                                                                                ^~~~~~~~~~~~~~~~
                                                                                Upp::_mm_castsi128_ps
/usr/lib/llvm-10/lib/clang/10.0.0/include/emmintrin.h:4930:1: note: 'Upp::_mm_castsi128_ps' declared here
_mm_castsi128_ps(__m128i __a)
^
In file included from
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen.cpp:2:
In file included from
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen.h:19:
In file included from
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen/Eigen:1:
In file included from
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen/Dense:1:
In file included from
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen/Core:193:
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen/src/Core/arch/SSE/PacketMath.h:227:110: error: cannot initialize a parameter of type 'Upp::__m128i' (vector of 2 'long long' values) with an rvalue of type 'Eigen::internal::Packet4i' (aka 'int')
template<> EIGEN_STRONG_INLINE Packet4f pset1frombits<Packet4f>(unsigned int from) { return _mm_castsi128_ps(pset1<Packet4i>(from)); }
                                                                                      ^~~~~~~~~~~~~~~~~~~~
/usr/lib/llvm-10/lib/clang/10.0.0/include/emmintrin.h:4930:26: note: passing argument to parameter

'__a' here
_mm_castsi128_ps(__m128i __a)
^
In file included from
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen.cpp:2:
In file included from
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen.h:19:
In file included from
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen/Eigen:1:
In file included from
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen/Dense:1:
In file included from
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen/Core:193:
/home/ssg/.local/soft/bb-worker/worker/upp/build/uppsrc/plugin/Eigen/Eigen/src/Core/arch/SSE/Pa
cketMath.h:229:79: error: use of undeclared identifier '_mm_setzero_ps'; did you mean
'Upp::_mm_setzero_ps'?
template<> EIGEN_STRONG_INLINE Packet4f pzero(const Packet4f& /*a*/) { return
_mm_setzero_ps(); }
^~~~~~~~~~~~~~
Upp::_mm_setzero_ps
/usr/lib/llvm-10/lib/clang/10.0.0/include/xmmintrin.h:1903:1: note: 'Upp::_mm_setzero_ps' declared
here
_mm_setzero_ps(void)
^
fatal error: too many errors emitted, stopping now [-ferror-limit=]
20 errors generated.
plugin/Eigen: 1 file(s) built in (0:00.88), 886 msecs / file
Eigen_demo: 3 file(s) built in (0:00.91), 305 msecs / file

Subject: Re: Eigen updated
Posted by koldo on Thu, 21 May 2020 13:12:59 GMT
View Forum Message <> Reply to Message

Thank you Novo

Yes, the new version of Core/Ops.h adds an #include <smmintrin.h> that declares __m128i, but in some way smashes the rest of references to __m128i in other include files. And Eigen uses vectorisation.

I am working for hours on it... I will try to solve the problem.
(And this is not because of Eigen's new version :))

Subject: Re: Eigen updated
Posted by Sender Ghost on Thu, 21 May 2020 14:50:15 GMT

View Forum Message <> Reply to Message

Hello.

Please check second variant of the patch in Redmine #2035, which may fix reference/Eigen_demo build.

Subject: Re: Eigen updated
Posted by koldo on Thu, 21 May 2020 15:48:07 GMT
View Forum Message <> Reply to Message

Great Sender

With your first patch, now MSVC works in W10 (MinGW and CLANG worked before).
However the problem remains in Ubuntu.
I have added information in your Redmine bug as in Core/Oops.h, in line 395, RGBA should have to be dword.

Subject: Re: Eigen updated
Posted by koldo on Fri, 22 May 2020 07:25:34 GMT
View Forum Message <> Reply to Message

Dear Sender

Some changes have been included in last version, but not all.
I am overwhelmed, vectorization and intrinsics are out of my knowledge.

It is like if including #include <smmintrin.h> in an include (Core/Blit.h), hides __m128* unions.
However I do not know why. Maybe caused by name mangling?
Replacing #ifdef CPU_X86 with #if defined(CPU_X86) && defined(__SSE2__) just avoids the problem by avoiding #include <smmintrin.h>.

Maybe these new options in Core/Blit.h require adding new compiler flags?

Subject: Re: Eigen updated
Posted by Sender Ghost on Fri, 22 May 2020 08:08:06 GMT
View Forum Message <> Reply to Message

koldo wrote on Fri, 22 May 2020 07:25
Replacing #ifdef CPU_X86 with #if defined(CPU_X86) && defined(__SSE2__) just avoids the problem by avoiding #include <smmintrin.h>

The __SSE2__ define was used for <emmintrin.h> include, because of _mm_set1_epi32 usage.
There is a possibility to use <immintrin.h> include for available intrinsics:

#ifdef __SSE2__
#include <emmintrin.h>
#endif

or <x86intrin.h>:

#include <immintrin.h>

But may need to check for related define where related intrinsic(s) used, I guess.

Some examples of how this was done for Eigen:
 uppsrc/plugin/Eigen/Eigen/src/Core/util/ConfigureVectorizati on.h:236-238:

#ifdef __SSE4_1__
 #define EIGEN_VECTORIZE_SSE4_1
#endif

 uppsrc/plugin/Eigen/Eigen/src/Core/util/ConfigureVectorizati on.h:350-352:

#ifdef EIGEN_VECTORIZE_SSE4_1
 #include <smmintrin.h>
#endif

for OpenGL Mathematics (GLM):
uppsrc/plugin/glm/simd/platform.h:242:

#define GLM_ARCH_SSE2 (GLM_ARCH_SSE2_BIT | GLM_ARCH_SSE)

uppsrc/plugin/glm/simd/platform.h:296-297:

elif defined(__SSE2__) || defined(__x86_64__) || defined(_M_X64) || defined(_M_IX86_FP)
define GLM_ARCH (GLM_ARCH_SSE2)

uppsrc/plugin/glm/simd/platform.h:340-342:

#elif GLM_ARCH & GLM_ARCH_SSE2_BIT
include <emmintrin.h>
#endif//GLM_ARCH

Possible to fix Eigen build issue, if place usage of intrinsic(s) (e.g. in uppsrc/Core) inside of C/C++ (*.c/*.cpp) instead of header (*.h) files. Maybe there are other methods.

Subject: Re: Eigen updated
Posted by koldo on Fri, 22 May 2020 08:23:22 GMT
View Forum Message <> Reply to Message

Quote:Possible to fix Eigen build issue, if place usage of intrinsic(s) (e.g. in uppsrc/Core) inside of C++ (*.cpp) instead of header (*.h) files. Maybe there are other methods.Yes, that would solve it all.
The problem of this is that inline functions have to be in .h, and Core/Blit.h includes inline void memsetd().

Subject: Re: Eigen updated
Posted by Sender Ghost on Fri, 22 May 2020 09:42:22 GMT
View Forum Message <> Reply to Message

koldo wrote on Fri, 22 May 2020 08:23Quote:Possible to fix Eigen build issue, if place usage of intrinsic(s) (e.g. in uppsrc/Core) inside of C++ (*.cpp) instead of header (*.h) files. Maybe there are other methods.Yes, that would solve it all.
The problem of this is that inline functions have to be in .h, and Core/Blit.h includes inline void memsetd().
Looks like, the Eigen build issue with <smmintrin.h> include was because of its include inside of (Upp) namespace.
I attached newer patch in Redmine #2035. But if there is a need to support CPUs without SSE2 instructions, there is a need to provide other implementations, based on __SSE2__ (or other) check, I guess.

Subject: Re: Eigen updated
Posted by koldo on Fri, 22 May 2020 10:49:41 GMT
View Forum Message <> Reply to Message

Awesome Sender, you got it!
I have asked Mirek and Tom here to consider your patch.
I owe you one.

Subject: Re: Eigen updated
Posted by Novo on Fri, 29 May 2020 22:10:52 GMT
View Forum Message <> Reply to Message

This is me again with bad news.
I tried to compile Eigen_demo for Windows using CLANG (actually, I compiled on Linux using wine), and CLANG never fished compilation. I guess this is a bug with CLANG shipped with Upp.
Or it is a bug with Eigen_demo itself ...
Command used for compilation:
wine umk reference Eigen_demo CLANG -bus

Subject: Re: Eigen updated
Posted by koldo on Sat, 30 May 2020 09:32:37 GMT

View Forum Message <> Reply to Message

Dear Novo

I cannot reproduce your problem in my W10, using CLANG from U++ and MSVC, both 32 and 64 bits.
Anyway, now it is just updated the very last version, with tiny changes. Please check it just in case, and send the error messages.
Thank you for your support.

Subject: Re: Eigen updated
Posted by Novo on Sun, 31 May 2020 05:05:15 GMT
View Forum Message <> Reply to Message

I was able to compile it in Release configuration.
$ wine umk reference Eigen_demo CLANGx64 -brus
----- Core (CLANG SHARED BLITZ WIN32) (1 / 4)
----- plugin/Eigen (CLANG SHARED BLITZ WIN32) (2 / 4)
----- plugin/z (CLANG SHARED BLITZ WIN32) (3 / 4)
----- Eigen_demo (MAIN CLANG SHARED BLITZ WIN32) (4 / 4)
BLITZ: eigen_demo.cpp non-linear.cpp fft.cpp
Eigen_demo: 3 file(s) built in (1:37.71), 32570 msecs / file
Linking...
Z:\home\ssg\.local\soft\bb-worker\worker\wine-upp\build\.cache\upp.out\CLANGx64.Blitz.Shared\Eigen_demo.exe (2488320 B) linked in (0:00.69)

It takes a minute and a half.
I never saw it finishing compilation in Debug.
TheIde and Clang are from March 3-rd 2020. I guess this is a release.
Tested on Linux with wine.

Subject: Re: Eigen updated
Posted by koldo on Sun, 31 May 2020 18:44:02 GMT
View Forum Message <> Reply to Message

Dear Novo

I have never used wine with U++. However, I will try to run it all to try to reproduce your problem.
On the other side, I can promise that it works everyday in tenths of computers with W10.