C++ and Linear Algebra

Guy Davidson
#include `<C++>`

includecpp.org
What to expect

What is linear algebra?
What to expect

What is linear algebra?
What is a linear algebra library?
What to expect

What is linear algebra?
What is a linear algebra library?
Customising the library
What to expect

What is linear algebra?
What is a linear algebra library?
Customising the library
Applications in colour
What to expect

What is linear algebra?
What is a linear algebra library?
Customising the library
Applications in colour
Applications in geometry
What is Linear Algebra?
What is linear algebra?

“The branch of mathematics concerning linear equations and linear functions, and their representation through matrices and vector spaces”
What is linear algebra?

◊ “The branch of mathematics concerning linear equations and linear functions, and their representation through matrices and vector spaces”

◊ \[ a_1x_1 + a_2x_2 + \ldots + a_nx_n = b \]
What is linear algebra?

◊ “The branch of mathematics concerning linear equations and linear functions, and their representation through matrices and vector spaces”

◊ $a_1x_1 + a_2x_2 + ... + a_nx_n = b$

◊ Geometry
What is linear algebra?

- “The branch of mathematics concerning linear equations and linear functions, and their representation through matrices and vector spaces”
- \(a_1x_1 + a_2x_2 + ... + a_nx_n = b\)
- Geometry
- Colour
What is linear algebra?

◊ “The branch of mathematics concerning linear equations and linear functions, and their representation through matrices and vector spaces”

◊ $a_1x_1 + a_2x_2 + ... + a_nx_n = b$

◊ Geometry

◊ Colour

◊ Solving simultaneous equations
What is linear algebra?

- Matrix
What is linear algebra?

- **Matrix**

- $\begin{bmatrix}
  a_{11} & a_{12} & \ldots & a_{1n} \\
  a_{21} & a_{22} & \ldots & a_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{m1} & a_{m2} & \ldots & a_{mn}
\end{bmatrix}$
What is linear algebra?

- Matrix-scalar multiplication
What is linear algebra?

◊ Matrix-scalar multiplication

◊ \( b \times \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} = \begin{bmatrix} b\times a_{11} & b\times a_{12} & \ldots & b\times a_{1n} \\ b\times a_{21} & b\times a_{22} & \ldots & b\times a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b\times a_{m1} & b\times a_{m2} & \ldots & b\times a_{mn} \end{bmatrix} \)
What is linear algebra?

- Matrix addition
What is linear algebra?

- **Matrix addition**

\[
\begin{bmatrix}
a_{11} & a_{12} & \ldots & a_{1n} \\
a_{21} & a_{22} & \ldots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \ldots & a_{mn}
\end{bmatrix}
+ \begin{bmatrix}
b_{11} & b_{12} & \ldots & b_{1n} \\
b_{21} & b_{22} & \ldots & b_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
b_{m1} & b_{m2} & \ldots & b_{mn}
\end{bmatrix}
= \begin{bmatrix}
a_{11}+b_{11} & a_{12}+b_{12} & \ldots & a_{1n}+b_{1n} \\
a_{21}+b_{21} & a_{22}+b_{22} & \ldots & a_{2n}+b_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1}+b_{m1} & a_{m2}+b_{m2} & \ldots & a_{mn}+b_{mn}
\end{bmatrix}
\]
What is linear algebra?

- Matrix-matrix multiplication
What is linear algebra?

Matrix-matrix multiplication

\[
\begin{bmatrix}
a_{11} & a_{12} & \ldots & a_{1n} \\
a_{21} & a_{22} & \ldots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \ldots & a_{mn}
\end{bmatrix}
\begin{bmatrix}
b_{11} & b_{12} & \ldots & b_{1m} \\
b_{21} & b_{22} & \ldots & b_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
b_{n1} & b_{n2} & \ldots & b_{nm}
\end{bmatrix}
= 
\begin{bmatrix}
a_{11} \cdot b_{11} & a_{12} \cdot b_{12} & \ldots & a_{1n} \cdot b_{1n} \\
a_{21} \cdot b_{21} & a_{22} \cdot b_{22} & \ldots & a_{2n} \cdot b_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} \cdot b_{n1} & a_{m2} \cdot b_{n2} & \ldots & a_{mn} \cdot b_{nm}
\end{bmatrix}
\]
What is linear algebra?

◊ Matrix-matrix multiplication

◊ \[
\begin{bmatrix}
a_{11} & a_{12} & \ldots & a_{1n} \\
a_{21} & a_{22} & \ldots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \ldots & a_{mn}
\end{bmatrix} \times
\begin{bmatrix}
b_{11} & b_{12} & \ldots & b_{1m} \\
b_{21} & b_{22} & \ldots & b_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
b_{n1} & b_{n2} & \ldots & b_{nm}
\end{bmatrix} =
\begin{bmatrix}
a_{1}b_{1} & a_{1}b_{2} & \ldots & a_{1}b_{n} \\
a_{2}b_{1} & a_{2}b_{2} & \ldots & a_{2}b_{n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m}b_{1} & a_{m}b_{2} & \ldots & a_{m}b_{n}
\end{bmatrix}
\]

◊ \( A \times B \neq B \times A \)
What is linear algebra?

- Square matrix
What is linear algebra?

- Square matrix

\[
\begin{bmatrix}
a_{11} & a_{12} & \ldots & a_{1n} \\
a_{21} & a_{22} & \ldots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \ldots & a_{nn}
\end{bmatrix}
\]
What is linear algebra?

- Identity matrix
What is linear algebra?

diamond Identity matrix

diamond $I = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix}$
What is linear algebra?

◊ Identity matrix

◊ \( I = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix} \)

◊ \( A \cdot I = I \cdot A = A \)
What is linear algebra?

- Determinant of $A = |A|$
What is linear algebra?

- Determinant of $A = |A|
- Inverse of $A = A^{-1}$
- $A * A^{-1} = A^{-1} * A = I$
What is linear algebra?

- Determinant of $A = |A|
- Inverse of $A = A^{-1}$
- $A \cdot A^{-1} = A^{-1} \cdot A = I$
What is linear algebra?

- operator+()
- operator-()
- operator*()
- operator/()
- operator++(), operator--()
- operator>>()
What is linear algebra?

- Vector
What is linear algebra?

- Vector
- Single row or single column
What is linear algebra?

- Vector
- Single row or single column
- Inner product
What is linear algebra?

◊ Vector
◊ Single row or single column
◊ Inner product

◊ \[ \begin{bmatrix} a & b \end{bmatrix} \times \begin{bmatrix} x \end{bmatrix} = (a \times x) + (b \times y) \]
   \[ \begin{bmatrix} y \end{bmatrix} \]
What is linear algebra?

- Vector
- Single row or single column
- Outer product
What is linear algebra?

- Vector
- Single row or single column
- Outer product

\[
[a] \times [x \ y] = [a \times x \ a \times y] \\
[b] \quad [b \times x \ b \times y]
\]
What is linear algebra?

- Vector
- Single row or single column
- Abstraction problem
What is linear algebra?

- Vector
- Single row or single column
- Abstraction problem
- Naming problem
What is linear algebra?

- $ax + by = e$
- $cx + dy = f$
What is linear algebra?

- $ax + by = e$
- $cx + dy = f$
- $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} e \\ f \end{bmatrix}$
What is linear algebra?

º \( ax + by = e \)
º \( cx + dy = f \)
º \([a \ b] \ * \ [x] = [e]\)
º \([c \ d] \ [y] \ [f]\)
º \( M \ * \ [x] = [e]\)
º \([y] \ [f]\)
What is linear algebra?

- \( ax + by = e \)
- \( cx + dy = f \)

- \( \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} e \\ f \end{bmatrix} \)

- \( M \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} e \\ f \end{bmatrix} \)

- \( \begin{bmatrix} x \\ y \end{bmatrix} = M^{-1} \begin{bmatrix} e \\ f \end{bmatrix} \)
What is linear algebra?

- $2x + 3y = 8$
- $x - 2y = -3$
What is linear algebra?

- $2x + 3y = 8$
- $x - 2y = -3$

$M = \begin{bmatrix} 2 & 3 \\ 1 & -2 \end{bmatrix}$
What is linear algebra?

- $2x + 3y = 8$
- $x - 2y = -3$
- $M = \begin{bmatrix} 2 & 3 \\ 1 & -2 \end{bmatrix}$
- $M^{-1} = |M|^{-1} \times \text{adjugate}(M)$
What is linear algebra?

- $2x + 3y = 8$
- $x - 2y = -3$

- $M = \begin{bmatrix} 2 & 3 \\ 1 & -2 \end{bmatrix}$

- $M^{-1} = |M|^{-1} \times \text{adjugate}(M)$

- $|M| = (2 \times -2) - (1 \times 3)$
  - $= -7$
What is linear algebra?

- $2x + 3y = 8$
- $x - 2y = -3$

- $M = \begin{bmatrix} 2 & 3 \\ 1 & -2 \end{bmatrix}$

- $M^{-1} = |M|^{-1} \times \text{adjugate}(M)$

- $|M| = (2 \times -2) - (1 \times 3) = -7$

- $\text{adjugate}(M) = \begin{bmatrix} -2 & -3 \\ -1 & 2 \end{bmatrix}$
What is linear algebra?

- $2x + 3y = 8$
- $x - 2y = -3$
- $M = \begin{bmatrix} 2 & 3 \\ 1 & -2 \end{bmatrix}$
- $M^{-1} = -7^{-1} \times \begin{bmatrix} -2 & -3 \\ -1 & 2 \end{bmatrix}$
What is linear algebra?

- $2x + 3y = 8$
- $x - 2y = -3$

- $M = \begin{bmatrix} 2 & 3 \\ 1 & -2 \end{bmatrix}$

- $M^{-1} = -7^{-1} \times \begin{bmatrix} -2 & -3 \\ -1 & 2 \end{bmatrix}$

- $[x] = -7^{-1} \times \begin{bmatrix} -2 & -3 \end{bmatrix} \times \begin{bmatrix} 8 \\ -3 \end{bmatrix}$
What is linear algebra?

- $2x + 3y = 8$
- $x - 2y = -3$

- $M = \begin{bmatrix} 2 & 3 \\ 1 & -2 \end{bmatrix}$
- $M^{-1} = -7^{-1} \times \begin{bmatrix} -2 & -3 \\ -1 & 2 \end{bmatrix}$
- $\begin{bmatrix} x \\ y \end{bmatrix} = -7^{-1} \times \begin{bmatrix} -2 & -3 \\ -1 & 2 \end{bmatrix} \times \begin{bmatrix} 8 \\ -3 \end{bmatrix}$
- $\begin{bmatrix} x \\ y \end{bmatrix} = \frac{\left( ( -2 \times 8 ) + ( -3 \times -3 ) \right) }{ -7 } = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
What is a Linear Algebra Library?
What is a Linear Algebra Library?

- 68000, fixed point
What is a Linear Algebra Library?

- 68000, fixed point
- 80286, fixed point, C/C++
What is a Linear Algebra Library?

- 68000, fixed point
- 80286, fixed point, C/C++
- 80486, floating point, C++
What is a Linear Algebra Library?

- 68000, fixed point
- 80286, fixed point, C/C++
- 80486, floating point, C++
- SSE2, Pentium IV
What is a Linear Algebra Library?

- 68000, fixed point
- 80286, fixed point, C/C++
- 80486, floating point, C++
- SSE2, Pentium IV
- AVX, (Sandy bridge)
What is a Linear Algebra Library?

- 68000, fixed point
- 80286, fixed point, C/C++
- 80486, floating point, C++
- SSE2, Pentium IV
- AVX, (Sandy bridge)
- N4860, P1385
What is a Linear Algebra Library?

- Optimisations available through specialisation
What is a Linear Algebra Library?

- Optimisations available through specialisation
- Matrix size
What is a Linear Algebra Library?

- Optimisations available through specialisation
- Matrix size
- Float
What is a Linear Algebra Library?

- Optimisations available through specialisation
- Matrix size
- Float
- SIMD instruction set
What is a Linear Algebra Library?

- Optimisations available through specialisation
- Matrix size
- Float
- SIMD instruction set
- Cache line size
What is a Linear Algebra Library?

- Optimisations available through specialisation
- Matrix size
- Float
- SIMD instruction set
- Cache line size
- Dense
What is a Linear Algebra Library?

- Matrix
What is a Linear Algebra Library?

- Matrix
- Vector
What is a Linear Algebra Library?

- Matrix
- Vector
- Infix operator overloads
What is a Linear Algebra Library?

- Matrix
- Vector
- Infix operator overloads
- \( M + M, V - V, a \cdot M, V / a \ldots \)
What is a Linear Algebra Library?

- Matrix
- Vector
- Infix operator overloads
- M+M, V·V, a*M, V/a...
- V*V, V*M, M*V, M*M
What is a Linear Algebra Library?

◊ operator >>
What is a Linear Algebra Library?

- `operator >>`
- `operator []`
What is a Linear Algebra Library?

- operator `>>`
- operator `[]`
- m(i,j)
What is a Linear Algebra Library?

- operator `>>`
- operator `[]`
- `m(i,j)`
- `m[i,j]`
What is a Linear Algebra Library?

- operator `>>`
- operator `[]`
- `m(i,j)`
- `m[i,j]`
- `m[i][j]`
What is a Linear Algebra Library?

◊ operator *
What is a Linear Algebra Library?

- operator *
- $6 \times 9$
What is a Linear Algebra Library?

- operator *
- 6 x 9
- operator x
What is a Linear Algebra Library?

- operator *
- $6 \times 9$
- operator x
- operator *
What is a Linear Algebra Library?

- operator *
- 6 x 9
- operator x
- operator *
- 6 x 9
What is a Linear Algebra Library?

$v^w$
What is a Linear Algebra Library?

- Hadamard product

- \((3, 2) \times (4, 2) = (12, 4)\)

- \[
\begin{bmatrix}
3 & 2 \\
4 & 2 \\
\end{bmatrix}
\times
\begin{bmatrix}
4 & 1 \\
2 & 2 \\
\end{bmatrix}
= 
\begin{bmatrix}
12 & 2 \\
8 & 4 \\
\end{bmatrix}
\]
What is a Linear Algebra Library?

- BLAS (Basic Linear Algebra Subprograms)
What is a Linear Algebra Library?

- BLAS (Basic Linear Algebra Subprograms)
- BLAS++
What is a Linear Algebra Library?

- BLAS (Basic Linear Algebra Subprograms)
- BLAS++

```cpp
void blas::axpy(int64_t n, float alpha, float const* x, int64_t incx,
                 float* y, int64_t incy);
```
What is a Linear Algebra Library?

◊ BLAS (Basic Linear Algebra Subprograms)
◊ BLAS++

◊ void blas::axpy(int64_t n, float alpha,
                   float const* x, int64_t incx,
                   float* y, int64_t incy);

◊ Boost.uBLAS
What is a Linear Algebra Library?

- asum: vector 1 norm (sum)
- axpy: add vectors
- copy: copy vector
- dot: dot product
- dotu: dot product, unconjugated
- iamax: max element
- nrm2: vector 2 norm
- rot: apply Givens plane rotation
- rotg: generate Givens plane rotation
- rotm: apply modified Givens plane rotation
- rotmg: generate modified Givens plane rotation
- scal: scale vector
- swap: swap vectors
What is a Linear Algebra Library?

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asum</td>
<td>general matrix-vector multiply</td>
</tr>
<tr>
<td>axpy</td>
<td>general matrix rank 1 update</td>
</tr>
<tr>
<td>copy</td>
<td>hermitian matrix-vector multiply</td>
</tr>
<tr>
<td>dot</td>
<td>hermitian rank 1 update</td>
</tr>
<tr>
<td>dotu</td>
<td>hermitian rank 2 update</td>
</tr>
<tr>
<td>iamax</td>
<td>symmetric matrix-vector multiply</td>
</tr>
<tr>
<td>nrm2</td>
<td>symmetric rank 1 update</td>
</tr>
<tr>
<td>rot</td>
<td>symmetric rank 2 update</td>
</tr>
<tr>
<td>rotg</td>
<td>triangular matrix-vector multiply</td>
</tr>
<tr>
<td>rotm</td>
<td>triangular matrix-vector solve</td>
</tr>
<tr>
<td>scal</td>
<td></td>
</tr>
<tr>
<td>swap</td>
<td></td>
</tr>
</tbody>
</table>
What is a Linear Algebra Library?

- **asum**
- **gemv**
- **gemm**
- General matrix multiply: \( C = AB + C \)
- **axpy**
- **ger**
- **hemm**
- Hermitian matrix multiply
- **copy**
- **hemv**
- **herk**
- Hermitian rank \( k \) update
- **dot**
- **her**
- **her2k**
- Hermitian rank 2\( k \) update
- **dotu**
- **her2**
- **symm**
- Symmetric matrix multiply
- **iamax**
- **symv**
- **syrk**
- Symmetric rank \( k \) update
- **nrm2**
- **syr**
- **syr2k**
- Symmetric rank 2\( k \) update
- **rot**
- **syr2**
- **trmm**
- Triangular matrix multiply
- **rotg**
- **trmv**
- **trsm**
- Triangular solve matrix
- **rotm**
- **trsv**
- **rotmg**
- **scal**
- **swap**

@hatcat01
What is a Linear Algebra Library?

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asum</td>
<td>general matrix multiply: ( C = AB + C )</td>
</tr>
<tr>
<td>axpy</td>
<td>hermitian matrix multiply</td>
</tr>
<tr>
<td>copy</td>
<td>hermitian rank ( k ) update</td>
</tr>
<tr>
<td>dot</td>
<td>hermitian rank 2( k ) update</td>
</tr>
<tr>
<td>dotu</td>
<td>symmetric matrix multiply</td>
</tr>
<tr>
<td>iamax</td>
<td>symmetric rank ( k ) update</td>
</tr>
<tr>
<td>nrm2</td>
<td>symmetric rank 2( k ) update</td>
</tr>
<tr>
<td>rot</td>
<td>triangular matrix multiply</td>
</tr>
<tr>
<td>rotg</td>
<td>triangular solve matrix</td>
</tr>
<tr>
<td>rotm</td>
<td></td>
</tr>
<tr>
<td>rotmg</td>
<td></td>
</tr>
<tr>
<td>scal</td>
<td></td>
</tr>
<tr>
<td>swap</td>
<td></td>
</tr>
</tbody>
</table>

P1673R2: A free function linear algebra interface based on the BLAS
What is a Linear Algebra Library?

- Eigen
What is a Linear Algebra Library?

- Eigen
- Matrix and vector templates
What is a Linear Algebra Library?

- Eigen
- Matrix and vector templates
- Dynamic or static dimensions
What is a Linear Algebra Library?

- Eigen
- Matrix and vector templates
- Dynamic or static dimensions
- Span
What is a Linear Algebra Library?

- Eigen
- Matrix and vector templates
- Dynamic or static dimensions
- Span
- Member function API
What is a Linear Algebra Library?

```cpp
#include <iostream>
#include <Eigen/Dense>
using namespace Eigen;
using namespace std;

int main() {
    MatrixXd m = MatrixXd::Random(3, 3);
    m = (m + MatrixXd::Constant(3, 3, 1.2)) * 50;
    cout << "m =" << endl << m << endl;
    VectorXd v(3);
    v << 1, 2, 3;
    cout << "m * v =" << endl << m * v << endl;
}
```
What is a Linear Algebra Library?

- Dlib
What is a Linear Algebra Library?

- Dlib
- Expression templates
What is a Linear Algebra Library?

◊ Blaze
# What is a Linear Algebra Library?

```cpp
#include <iostream>
#include <blaze/Math.h>
using blaze::StaticVector;
using blaze::DynamicVector

int main() {
    StaticVector<int,3UL> a{ 4, -2, 5 }
    DynamicVector<int> b( 3UL );
    b[0] = 2;
    b[1] = 5;
    b[2] = -3;
    DynamicVector<int> c = a + b;
    std::cout << "c =\n" << c << "\n";
}
```
What is a Linear Algebra Library?

- https://wg21.link/P1385
- Syntax proposal
- Reserve some identifiers
- Boost.QVM
Customising the library
Customising the library

- Element type
Customising the library

- Element type
- Element arrangement
Customising the library

◊ Element type
◊ Element arrangement
◊ std::math::matrix<float, 3, 3> ml;
Customising the library

- Element type
- Element arrangement
  - `std::math::matrix<float, 3, 3> m1;`
  - `std::math::matrix<float> m2;`
Customising the library

◊ Designing storage engines
Customising the library

- Designing storage engines
- `automatic_storage<T, R, C>`
Customising the library

◊ Designing storage engines
◊ automatic_storage<T, R, C>
◊ dynamic_storage<T, A>
Customising the library

- Designing storage engines
- `automatic_storage<T, R, C>`
- `dynamic_storage<T, A>`
- `std::math::matrix<automatic_storage<float, 3, 3>> ml;`
Customising the library

♀ Designing storage engines
♀ `automatic_storage<T, R, C>`
♀ `dynamic_storage<T, A>`
♀ `std::math::matrix<automatic_storage<float, 3, 3>> m1;`
♀ `std::math::matrix<dynamic_storage<float, std::allocator>> m2;`
Customising the library

✧ Designing storage engines

✧ automatic_storage<T, R, C>
✧ dynamic_storage<T, A>
✧ std::math::matrix<automatic_storage<float, 3, 3>> m1;
✧ std::math::matrix<dynamic_storage<float, std::allocator>> m2;
✧ using geometry = automatic_storage<float, 3, 3>;
    std::math::matrix<geometry> m1;
Customising the library

diamond mdspan : P0009
Customising the library

diamond mdspan : P0009

diamond Multidimensional arrays are a foundational data structure for science and engineering codes, as demonstrated by their extensive use in Fortran for five decades. A multidimensional array is a view to a memory extent through a layout mapping from a multi-index space (domain) to that extent (range).
Customising the library

- mdspan : P0009

- Traditional layout mappings have been specified as part of the language. For example, Fortran specifies column major layout and C specifies row major layout. Such a language-imposed specification requires significant code refactoring to change an array's layout and requires significant code complexity to implement non-traditional layouts such as tiling in modern linear algebra or structured grid application domains.
Customising the library

- `mdspan : P0009`

- A multidimensional array view abstraction with polymorphic layout is required to enable changing array layout without extensive code refactoring and maintenance of functionally redundant code. Layout polymorphism is a critical capability; however, it is not the only beneficial form of polymorphism.
Customising the library

- `mdspan : P0009`
- `template <ptrdiff_t... Extents> class extents;`
Customising the library

- `mdspan : P0009`
- `template <ptrdiff_t... Extents> class extents;`
- `dynamic_extent`
Customising the library

matrix_storage_engine<T, extents<R, C>, A>;
Customising the library

- `matrix_storage_engine<T, extents<R, C>, A>;
- `matrix<matrix_storage_engine<float, extents<3, 3>, void>>;`
Customising the library

- `matrix_storage_engine<T, extents<R, C>, A>;
- `matrix<matrix_storage_engine<float, extents<3, 3>, void>>;
- `matrix<matrix_storage_engine<float, dynamic_extents, std::allocator<T>>>;`
Customising the library

- `matrix_storage_engine<T, extents<R, C>, A>;
- `matrix<matrix_storage_engine<float, extents<3, 3>, void>>;
- `matrix<matrix_storage_engine<float, dynamic_extents, std::allocator<T>>>;
- `matrix_storage_engine<T, extents<R, C>, A, L>;
Customising the library

- `matrix_storage_engine<T, extents<R, C>, A>;
- `matrix<matrix_storage_engine<float, extents<3, 3>, void>>;
- `matrix<matrix_storage_engine<float, dynamic_extents, std::allocator<T>>>;
- `matrix_storage_engine<T, extents<R, C>, A, L>;
- `matrix<matrix_storage_engine<float, extents<3, 3>, void, matrix_layout::row_major>>;`
Customising the library

- `matrix_storage_engine<T, extents<R, C>, A>;
- `matrix<matrix_storage_engine<float, extents<3, 3>, void>>;
- `matrix<matrix_storage_engine<float, dynamic_extents, std::allocator<T>>>;
- `matrix_storage_engine<T, extents<R, C>, A, L>;
- `matrix<matrix_storage_engine<float, extents<3, 3>, void, matrix_layout::row_major>>;
- `matrix<matrix_storage_engine<float, dynamic_extents, std::allocator<T>>, matrix_layout::column_major>;
Customising the library

```cpp
#include <iostream>

int main()
{
    std::cout << 1 + 2.5;
}
```
Customising the library

```cpp
#include <iostream>

int main()
{
    std::cout << 1 + 2.5;
}

3.5
```
Customising the library

- `#include <iostream>

int main()
{
    std::cout << 1 + 2.5;
}

3.5

double operator+(int, double)?
Customising the library

```
#include <iostream>

int main()
{
    std::cout << 1 + 2.5;
}

3.5

double operator+(int, double)?
double operator+(double, double)
```
Customising the library

```cpp
#include <iostream>
#include <complex>

int main()
{
    std::cout << std::complex<double>(3., 3.);
}
```
Customising the library

```cpp
#include <iostream>
#include <complex>

int main()
{
    std::cout << std::complex<double>(3., 3.);
}

(3,3)
```
Customising the library

```cpp
#include <iostream>
#include <complex>

int main()
{
    std::cout << std::complex<int>(3., 3.);
}
```
Customising the library

```cpp
#include <iostream>
#include <complex>

int main()
{
    std::cout << std::complex<int>(3., 3.);
}
(3,3)
```
Customising the library

```cpp
#include <iostream>
#include <complex>

int main()
{
    std::cout << std::complex<int>(3.7, 3.2);
}
```
Customising the library

```cpp
#include <iostream>
#include <complex>

int main()
{
    std::cout << std::complex<int>(3.7, 3.2);
}

(3,3)
```
Customising the library

```cpp
#include <iostream>
#include <complex>

int main()
{
    std::cout << std::complex<int>(3.7, 3.2) + std::complex<int>(4, 4);
}
```
Customising the library

```cpp
#include <iostream>
#include <complex>

int main()
{
    std::cout << std::complex<int>(3.7, 3.2) + std::complex<int>(4, 4);
}

(7,7)
```
Customising the library

```cpp
#include <iostream>
#include <complex>

int main()
{
    std::cout << std::complex<float>(3.7, 3.2) + std::complex<float>(4, 4);
}
```
Customising the library

```cpp
#include <iostream>
#include <complex>

int main()
{
    std::cout << std::complex<float>(3.7, 3.2) + std::complex<float>(4, 4);
}

(7.7, 7.2)
```
Customising the library

#include <iostream>
#include <complex>

int main()
{
    std::cout << std::complex<float>(3.7, 4)
              + std::complex<float>(4, 3.2);
}

@hatcat01
Customising the library

```cpp
#include <iostream>
#include <complex>

int main()
{
    std::cout << std::complex<float>(3.7, 4) + std::complex<float>(4, 3.2);
}
```

(7.7, 7.2)
# Customising the library

```cpp
#include <iostream>
#include <complex>

int main()
{
    std::cout << std::complex<float>(3.7, 4) + std::complex<double>(4, 3.2);
}
```
Customising the library

- `binary '+' : 'std::complex<float>' does not define this operator or a conversion to a type acceptable to the predefined operator`
Customising the library

using double_33_a = matrix_storage_engine<double, extents<3, 3>,
void, matrix_layout::row_major>;
Customising the library

```cpp
// using double_33_a = matrix_storage_engine<double, extents<3, 3>,
// void, matrix_layout::row_major>;

using float_33_d = matrix_storage_engine<float, extents<3, 3>,
std::allocator<T>, matrix_layout::column_major>;
```
Customising the library

```cpp
// using double_33_a = matrix_storage_engine<double, extents<3, 3>,
// void, matrix_layout::row_major>;

using float_33_d = matrix_storage_engine<float, extents<3, 3>,
std::allocator<T>, matrix_layout::column_major>;

matrix<double_33_a> m1 = get_auto_mat();
```
Customising the library

```cpp
using double_33_a = matrix_storage_engine<double, extents<3, 3>,
    void, matrix_layout::row_major>;
using float_33_d = matrix_storage_engine<float, extents<3, 3>,
    std::allocator<T>, matrix_layout::column_major>;

matrix<double_33_a> m1 = get_auto_mat();
matrix<float_33_d> m2 = get_dyna_mat();
```
Customising the library

```cpp
using double_33_a = matrix_storage_engine<double, extents<3, 3>,
    void, matrix_layout::row_major>;

using float_33_d = matrix_storage_engine<float, extents<3, 3>,
    std::allocator<T>, matrix_layout::column_major>;

matrix<double_33_a> m1 = get_auto_mat();

matrix<float_33_d> m2 = get_dyna_mat();

auto m3 = m1 + m2;
```
Customising the library

- matrix_storage_engine<double, extents<3, 3>, void, row_major>;
- matrix_storage_engine<float, extents<3, 3>, void, row_major>;
- => matrix_storage_engine<double, extents<3, 3>, void, row_major>;
Customising the library

◊ matrix_storage_engine<double, extents<3, 3>, void, row_major>;
◊ matrix_storage_engine<float, extents<3, 3>, void, row_major>;
◊ => matrix_storage_engine<double, extents<3, 3>, void, row_major>;
Customising the library

- `matrix_storage_engine<double, extents<3, 3>, void, row_major>;
- `matrix_storage_engine<float, extents<3, 3>, void, row_major>;
- `=> matrix_storage_engine<double, extents<3, 3>, void, row_major>;;
Customising the library

```cpp
struct matrix_operation_traits {
    template <typename OTR, class T1, class T2> addition_element_traits;
    template <typename OTR, class T1, class T2> addition_engine_traits;
    template <typename OTR, class T1, class T2> addition_arithmetic_traits;
    template <typename OTR, class T1, class T2> subtraction_element_traits;
    ...
    template <typename OTR, class T1, class T2> multiplication_element_traits;
    ...
    template <typename OTR, class T1, class T2> addition_element_traits;
};
```
Customising the library

```cpp
template<class T, ptrdiff_t R, ptrdiff_t C, class COT = void>
using fixed_size_matrix =
    basic_matrix<matrix_storage_engine<T, extents<R, C>>,
                 void, matrix_layout::row_major>, COT>;

template<class COT = void>
using matrix_33f =
    basic_matrix<matrix_storage_engine<float, extents<3, 3>>,
                 void, matrix_layout::row_major>, COT>;
```
Customising the library

Multiplication
Customising the library

- Multiplication
- $O(n^3)$
Customising the library

- Multiplication
- $O(n^3)$
- Strassen’s algorithm – $O(n^{2.807})$
Customising the library

- Multiplication
- $O(n^3)$
- Strassen’s algorithm – $O(n^{2.807})$
- Best result – $O(n^{2.3728639})$
Customising the library

```cpp
struct custom_operation_traits {
    template <typename OTR, class T1, class T2>
    using addition_element_traits =
        std::matrix_operation_traits::addition_element_traits<OTR, T1, T2>;
    template <typename OTR, class T1, class T2>
    using addition_engine_traits =
        std::matrix_operation_traits::addition_engine_traits<OTR, T1, T2>;
    template <typename OTR, class T1, class T2>
    using addition_arithmetic_traits =
        custom_addition_arithmetic_traits<OTR, T1, T2>;
    ...
};
```
Customising the library

basic_matrix<
Customising the library

- `basic_matrix`
- `matrix_storage_engine`
Customising the library

◊ `basic_matrix<`
◊
  `matrix_storage_engine<`
◊
  `element_type,`
Customising the library

- basic_matrix<
  - matrix_storage_engine<
    - element_type,
    - extents<R, C>,
Customising the library

- basic_matrix<
  - matrix_storage_engine<
    - element_type,
    - extents<R, C>,
    - allocator,
Customising the library

basic_matrix<

  matrix_storage_engine<

    element_type,
    extents<R, C>,
    allocator,
    layout>,

Customising the library

- `basic_matrix<`
  - `matrix_storage_engine<`
    - `element_type,`
    - `extents<R, C>,`
    - `allocator,`
    - `layout>,`
  - `matrix_operation_traits>;`
Customising the library

```cpp
template <typename T>
using complex_scalar_storage =
    matrix_storage_engine<std::complex<T>>, extents<1, 1>, void>;

template <typename T>
using complex_scalar =
    basic_matrix<complex_scalar_storage<T>>;

custom_scalar<float> c1{2.2f, 3.3f};
custom_scalar<double> c2{4.4, 5.5};
auto c3 = c1 + c2;
```
Applications in colour
Applications in colour
Applications in colour
Applications in colour
Applications in colour
Applications in colour
Applications in colour
Applications in colour
Applications in colour
Applications in colour
Applications in colour

$2.0\sqrt{x}$

$2.0\sqrt{x}$

$2.0\sqrt{x}$

$2.0\sqrt{x}$

$2.0\sqrt{x}$

$2.0\sqrt{x}$
<table>
<thead>
<tr>
<th>2.2√x</th>
<th>2.2√x</th>
<th>2.2√x</th>
<th>2.2√x</th>
<th>2.2√x</th>
<th>2.2√x</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.8√x</td>
<td>1.8√x</td>
<td>1.8√x</td>
<td>1.8√x</td>
<td>1.8√x</td>
<td>1.8√x</td>
</tr>
</tbody>
</table>

Applications in colour
Applications in colour
Applications in colour

❖ Take a standard human
Applications in colour

- Take a standard human
- Put them in a standard environment
Applications in colour

- Take a standard human
- Put them in a standard environment
- Measure how they perceive electromagnetic waves, via matching the colours of lights
Applications in colour

✧ Take a standard human
✧ Put them in a standard environment
✧ Measure how they perceive electromagnetic waves, via matching the colours of lights
✧ Build a function that maps wavelengths to perception, giving 3 values (X, Y, Z)
Applications in colour

- Take a standard human
- Put them in a standard environment
- Measure how they perceive electromagnetic waves, via matching the colours of lights
- Build a function that maps wavelengths to perception, giving 3 values (X, Y, Z)
- Add some mathematical constraints (values > 0, Y = relative luminance [0, 100])
Applications in colour

 normalized cone response (linear energy)

<table>
<thead>
<tr>
<th>Wavelength (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
</tr>
</tbody>
</table>

S  M  L
Applications in colour

◊ Humans separate colour from brightness
Applications in colour

- Humans separate colour from brightness

- Normalise:
  \[ x = \frac{X}{X + Y + Z} \]
  \[ y = \frac{Y}{X + Y + Z} \]
  \[ z = \frac{Z}{X + Y + Z} = (1 - x - y) \]
Applications in colour

◊ Humans separate colour from brightness

◊ Normalise:
  \[ x = \frac{X}{X + Y + Z} \]
  \[ y = \frac{Y}{X + Y + Z} \]
  \[ z = \frac{Z}{X + Y + Z} = (1 - x - y) \]

◊ xyY colour space
  x and y are colour
  Y is relative luminance
Applications in colour
Applications in colour

- Small change in a value has the same effect in perceived colour
Applications in colour

- Small change in a value has the same effect in perceived colour
- XYZ values are not perceptually uniform
Applications in colour

- Small change in a value has the same effect in perceived colour
- XYZ values are not perceptually uniform
- Inefficient, like storing sound volume in raw values rather than in dB. $100\text{dB}=1^{100}$
Applications in colour

diamond 1996: Microsoft + HP
Applications in colour

- 1996: Microsoft + HP
- IEC 61966-2-1:1999
Applications in colour

🔹 1996: Microsoft + HP
🔹 IEC 61966-2-1:1999
🔹 Default colour space where NO COLOUR SPACE INFORMATION is provided
Applications in colour

<table>
<thead>
<tr>
<th>Chromaticity</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
<th>White point</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0.6400</td>
<td>0.3000</td>
<td>0.1500</td>
<td>0.3127</td>
</tr>
<tr>
<td>y</td>
<td>0.3300</td>
<td>0.6000</td>
<td>0.0600</td>
<td>0.3290</td>
</tr>
<tr>
<td>Y</td>
<td>0.2126</td>
<td>0.7152</td>
<td>0.0722</td>
<td>1.0000</td>
</tr>
</tbody>
</table>
Applications in colour
Applications in colour
Applications in colour

\[
\begin{bmatrix}
R_{\text{linear}} \\
G_{\text{linear}} \\
B_{\text{linear}}
\end{bmatrix} =
\begin{bmatrix}
+3.24096994 & -1.53738318 & -0.49861076 \\
-0.96924364 & +1.8759675 & +0.04155506 \\
+0.05563008 & -0.20397696 & +1.05697151
\end{bmatrix}
\begin{bmatrix}
X_{D65} \\
Y_{D65} \\
Z_{D65}
\end{bmatrix}
\]

\[
\gamma(u) = \begin{cases} 
12.92u & u \leq 0.0031308 \\
1.055u^{1/4} - 0.055 & \text{otherwise}
\end{cases}
\]

\[
\gamma^{-1}(u) = \begin{cases} 
\frac{u}{12.92} & u \leq 0.04045 \\
\left(\frac{u + 0.055}{1.055}\right)^{2.4} & \left(\frac{200u + 11}{211}\right)^{12/5} & \text{otherwise}
\end{cases}
\]

\[
\begin{bmatrix}
X_{D65} \\
Y_{D65} \\
Z_{D65}
\end{bmatrix} =
\begin{bmatrix}
0.41239080 & 0.35758434 & 0.18048079 \\
0.21263901 & 0.71816868 & 0.07219323 \\
0.01933082 & 0.11919478 & 0.95053215
\end{bmatrix}
\begin{bmatrix}
R_{\text{linear}} \\
G_{\text{linear}} \\
B_{\text{linear}}
\end{bmatrix}
\]
Applications in colour

- Brightness perception is logarithmic
Applications in colour

- Brightness perception is logarithmic
- XYZ defines absolute perceptual colours
Applications in colour

- Brightness perception is logarithmic
- XYZ defines absolute perceptual colours
- The xyY colour space is linear
Applications in colour

- Brightness perception is logarithmic
- XYZ defines absolute perceptual colours
- The xyY colourspace is linear
- Linear interpolation is valid on linear colourspaces
Applications in colour

- Brightness perception is logarithmic
- XYZ defines absolute perceptual colours
- The xyY colourspace is linear
- Linear interpolation is valid on linear colourspaces
- sRGB is defined relative to xyY
Applications in colour

- Brightness perception is logarithmic
- XYZ defines absolute perceptual colours
- The xyY colourspace is linear
- Linear interpolation is valid on linear colourspaces
- sRGB is defined relative to xyY
- The transfer function is non-linear and expensive
Applications in colour

- Brightness perception is logarithmic
- XYZ defines absolute perceptual colours
- The xyY colourspace is linear
- Linear interpolation is valid on linear colourspaces
- sRGB is defined relative to xyY
- The transfer function is non-linear and expensive
- sRGB is non-linear
Applications in colour

- Brightness perception is logarithmic
- XYZ defines absolute perceptual colours
- The xyY colourspace is linear
- Linear interpolation is valid on linear colourspaces
- sRGB is defined relative to xyY
- The transfer function is non-linear and expensive
- sRGB is non-linear
- Linear interpolation is invalid on sRGB
Applications in colour

\( (x + y) / 2 \)
Applications in colour

- $(x + y) / 2$
- $(\sqrt{x} + \sqrt{y}) / 2 < \sqrt{(x + y) / 2}$
Applications in colour

- \((x + y) / 2\)
- \((\sqrt{x} + \sqrt{y}) / 2 < \sqrt{(x + y) / 2}\)
- \(x = 9, \ y = 16\)
Applications in colour

- $(x + y) / 2$
- $(\sqrt{x} + \sqrt{y}) / 2 < \sqrt{(x + y) / 2}$
- $x = 9, \ y = 16$
- $(\sqrt{9} + \sqrt{16}) / 2 = 3.5$
Applications in colour

- $\frac{x + y}{2}$
- $\frac{\sqrt{x} + \sqrt{y}}{2} < \sqrt{\frac{x + y}{2}}$
- $x = 9, \quad y = 16$
- $\frac{\sqrt{9} + \sqrt{16}}{2} = 3.5$
- $\sqrt{\frac{9 + 16}{2}} = 3.535$
Applications in colour

- $(x + y) / 2$
- $(\sqrt{x} + \sqrt{y}) / 2 < \sqrt{((x + y) / 2)}$
- $x = 9, y = 16$
- $(\sqrt{9} + \sqrt{16}) / 2 = 3.5$
- $\sqrt{(9 + 16) / 2} = 3.535$

- `template <class T>
  constexpr std::midpoint(T a, T b) noexcept;`
Applications in colour

\[ \frac{x + y}{2} \]
\[ \frac{\sqrt{x} + \sqrt{y}}{2} \leq \sqrt{\left( \frac{x + y}{2} \right)} \]
\[ x = 9, \ y = 16 \]
\[ \frac{\sqrt{9} + \sqrt{16}}{2} = 3.5 \]
\[ \sqrt{\left( \frac{9 + 16}{2} \right)} = 3.535 \]

//template <class T>
 constexpr std::midpoint(T a, T b) noexcept;

//constexpr float std::lerp(float a, float b, float t) noexcept
Applications in colour
Applications in colour

| 0 | 0 | 0 | 0 | 0 | 0 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 28 | 28 | 28 | 28 | 28 | 28 | 28 | 34 | 34 | 34 | 34 | 34 | 34 | 34 | 34 | 38 | 38 | 38 | 38 | 38 | 38 | 42 | 42 |
|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 2 | 42 | 42 | 46 | 46 | 46 | 50 | 50 | 50 | 50 | 53 | 53 | 53 | 56 | 56 | 59 | 59 | 61 | 61 | 61 | 64 | 64 | 64 | 66 | 66 | 66 | 69 | 69 | 71 | 71 | 71 | 73 | 73 | 73 | 73 | 75 | 75 | 77 | 77 | 79 | 79 | 81 | 81 |
| 3 | 83 | 85 | 85 | 86 | 86 | 88 | 88 | 90 | 92 | 92 | 93 | 95 | 95 | 96 | 96 | 98 | 99 | 99 | 101 | 102 | 104 | 104 | 105 | 106 | 106 | 108 | 108 | 109 | 110 | 112 | 112 | 113 | 114 | 115 | 117 | 117 | 117 | 117 | 117 | 117 |
| 117 | 118 | 119 | 120 | 121 | 122 | 124 | 124 | 125 | 126 | 127 | 128 | 128 | 130 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 | 145 | 146 |

Max error 6
Total error 127
Applications in colour

- libSDL
Applications in colour

- libSDL
- SFML
Applications in colour

- libSDL
- SFML
- DearImGui
Applications in colour

- libSDL
- SFML
- Dear ImGui
- Flash
Applications in colour

- libSDL
- SFML
- Dear ImGui
- Flash
- Unity
Applications in colour

- libSDL
- SFML
- Dear ImGui
- Flash
- Unity
- Godot
Applications in colour

- libSDL
- SFML
- Dear ImGui
- Flash
- Unity
- Godot
- OGRE
Applications in colour

◊ CRYENGINE
Applications in colour

- CRYENGINE
- MatLab
Applications in colour

- CRYENGINE
- MatLab
- OpenCV
Applications in colour

- CRYENGINE
- MatLab
- OpenCV
- SVG and CSS
Applications in colour

- CRYENGINE
- MatLab
- OpenCV
- SVG and CSS
- Qt
Applications in colour

- CRYENGINE
- MatLab
- OpenCV
- SVG and CSS
- Qt
- Unreal Engine
Applications in colour
Applications in colour

Wrong

Wrong

Wrong

Wrong

Wrong
Applications in geometry
Applications in geometry

◊ "The branch of mathematics concerned with questions of shape, size, relative position of figures and the properties of space."
Applications in geometry

"The branch of mathematics concerned with questions of shape, size, relative position of figures and the properties of space."
Applications in geometry
Applications in geometry

Proposition 1

To construct an equilateral triangle on a given finite straight-line.

Let \( AB \) be the given finite straight-line.

So it is required to construct an equilateral triangle on the straight-line \( AB \).

Let the circle \( BCD \) with center \( A \) and radius \( AB \) have been drawn [Post. 3], and again let the circle \( ACE \) with center \( B \) and radius \( BA \) have been drawn [Post. 3]. And let the straight-lines \( CA \) and \( CE \) have been joined from the point \( C \), where the circles cut one another, to the points \( A \) and \( B \) (respectively) [Post. 1].

And since the point \( A \) is the center of the circle \( CDB \), \( AC \) is equal to \( AB \) [Def. 1.15]. Again, since the point \( B \) is the center of the circle \( CAE \), \( BC \) is equal to \( BA \) [Def. 1.15]. But \( CA \) was also shown (to be) equal to \( AB \). Thus, \( CA \) and \( CB \) are each equal to \( AB \). But things equal to the same thing are also equal to one another [CN. 1]. Thus, \( CA \) is also equal to \( CB \). Thus, the three (straight-lines) \( CA \), \( AB \), and \( BC \) are equal to one another.

Thus, the triangle \( ABC \) is equilateral, and has been constructed on the given finite straight-line \( AB \). (Which is) the very thing it was required to do.
Applications in geometry

- René Descartes
- b. 31st March 1596
- d. 11th February 1650
Applications in geometry
Applications in geometry
Applications in geometry

\[ x, y \in \mathbb{R} \{ y - 2x - 3 = 0 \} \]
Applications in geometry

$y = 2x + 3$
Applications in geometry

\[ a_1x_1 + a_2x_2 + \ldots + a_nx_n = b \]
Applications in geometry

\[ a_1x_1 + a_2x_2 + \ldots + a_nx_n = b \]

\[ a_1x_1 + a_2x_2 = b \]
Applications in geometry

\[
\diamond a_1 x_1 + a_2 x_2 + \ldots + a_n x_n = b
\]

\[
\diamond a_1 x_1 + a_2 x_2 = b
\]

\[
\diamond a x + b y = c
\]
Applications in geometry

◊ $a_1x_1 + a_2x_2 + \ldots + a_nx_n = b$

◊ $a_1x_1 + a_2x_2 = b$

◊ $ax + by = c$

◊ $by = -ax + c$
Applications in geometry

- $a_1 x_1 + a_2 x_2 + ... + a_n x_n = b$
- $a_1 x_1 + a_2 x_2 = b$
- $ax + by = c$
- $by = -ax + c$
- $y = mx + c$
Applications in geometry

- (x, y)
- Translate
- (x, y) + (a, b) = (x + a, y + b)
Applications in geometry

- $(x, y)$
- Translate
- $(x, y) + (a, b) = (x + a, y + b)$
Applications in geometry

- $(x, y)$
- Scale
- $(x, y) \times 2 = (2x, 2y)$
- $(x, y) \times (2, 0) = (2x, 2y)
- (0, 2)$
Applications in geometry

- $(x, y)$
- Scale
- $(x, y) \times 2 = (2x, 2y)$
- $(x, y) \times (2\ 0) = (2x, 2y)$
- $(0\ 2)$
Applications in geometry

- $(x, y)$
- Shear
- $(x, y) \times (1 \ 2) = (x, 2x + y)$
  
  $(0 \ 1)$
Applications in geometry

- $(x, y)$
- Shear
- $(x, y) \cdot (1 \ 2) = (x, 2x + y)$
  
  $(0 \ 1)$
Applications in geometry

- $(x, y)$
- Reflect
- $(x, y) \times (1 \ 0) = (x, -y)$
  $(0 \ -1)$
Applications in geometry

- $(x, y)$
- Reflect
- $(x, y) \times (1 \ 0) = (x, -y)$
  $(0 \ -1)$
Applications in geometry

- $(x, y)$
- Rotate
  
  - $(x, y) \cdot (\cos a \ - \sin a)$
  
  
  \[
  \begin{pmatrix}
  \sin a & \cos a \\
  -\cos a & \sin a
  \end{pmatrix}
  \]
  
  
  = $(x \cdot \cos a + y \cdot \sin a,$
  
  
  $-x \cdot \sin a + y \cdot \cos a$)
Applications in geometry

- $(x, y)$
- Rotate
- $(x, y) \cdot (\cos a \; - \sin a)$
  
  $(\sin a \; \cos a)$
  
  $= (x \; \cos a \; + \; y \; \sin a, \; -x \; \sin a \; + \; y \; \cos a)$

@hatcat01
Applications in geometry

- Boost.Geometry
Applications in geometry

- Boost.Geometry
- Barend Gehrels
Applications in geometry

- Boost.Geometry
- Barend Gehrels
- Geometry classes
Applications in geometry

- Boost.Geometry
- Barend Gehrels
- Geometry classes
- Dimension agnostic
Applications in geometry

- Boost.Geometry
- Barend Gehrels
- Geometry classes
- Dimension agnostic
- Distance
Applications in geometry

- Boost.Geometry
- Barend Gehrels
- Geometry classes
- Dimension agnostic
- Distance
- Coordinate-system agnostic
Applications in geometry

$y = 2x + 3$
Applications in geometry

```c
struct line
{
  float gradient;
  float y_intercept;
};
```
Applications in geometry

```c
struct line
{
    float gradient;
    float y_intercept;
};

struct line_segment
{
    point p1;
    point p2;
};
```
Applications in geometry

◊ Q
Applications in geometry

- $\mathbb{Q}$
- 3244.7482
Applications in geometry

```cpp
struct line
{
    std::vector<point> points;
};
```
Applications in geometry

```c
# struct line
{
    float gradient;
    float y_intercept;
};
```
Applications in geometry

```c
typedef struct line
{
    float gradient;
    float y_intercept;
    point pl;
    point p2;
} point p2;
```
Applications in geometry

```c
struct line
{
    float gradient;
    float y_intercept;
    point p_begin;
    point p_end;
};
```
Applications in geometry

```c
struct bezier
{
    point p_begin;
    point p_control;
    point p_end;
};
```
Applications in geometry

\[ y = x - 1 \]
\[ y = 2x - 4 \]
Applications in geometry

- $y = x - 1$
- $y = 2x - 4$
- $0 = x - 3$
- $x = 3$
Applications in geometry

\[ y = x^2 \]
\[ y = x + 3.9 \]
Applications in geometry

- $y = x^2$
  - $y = x + 3.9$

- $0 = x^2 - x - 3.9$
  - $x = 0.5 \pm \sqrt{16.6}/2$
Applications in geometry

\[ y = x - 2.3 \]
\[ y = x/3 \]
Applications in geometry

- $y = x - 2.3$
- $y = x/3$
- $0 = 2x/3 - 2.3$
- $x = 3.45$
Applications in geometry

bool intersects(line a, line b);
Applications in geometry

- bool intersects(line a, line b);
- FLT_MIN vs FLT_EPSILON
Applications in geometry

bool intersects(line a, line b);
FLT_MIN vs FLT_EPSILON
bool intersects(line a, line b, float epsilon);
What to expect

What is linear algebra?
What is a linear algebra library?
Customising the library
Applications in colour
Applications in geometry
https://wg21.link/p1385
C++ and Linear Algebra

Guy Davidson