Math.NET for Unity

version 1.0.1

NOVEMBER 26, 2016

BIMUR STUDIOS LTD

Introduction

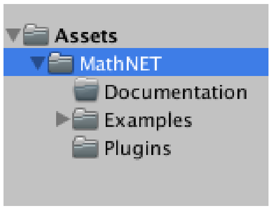
Math.NET for Unity is a port of the official Math.NET Numerics library (http://numerics.mathdotnet.com/). Math.NET Numerics is an open source mathematics library written in C#. It contains classes for Linear Algebra, Probability and Statistics, Special Functions, Regression, Interpolation, Numerical Integration and Differentiation, Function (or signal) Sampling and Optimization. The original Math.NET Numerics library is MIT licensed and is available at http://numerics.mathdotnet.com/ .

Limitations

Math.NET supports most major linear algebra providers (i.e. BLAS and LAPACK), however, due to licensing limitations these are not included within Math.NET for Unity. Instead, a managed provider is used that requires no additional setup. The down side of this is that linear algebra operations, that depend on these providers, are not platform optimised. That said, it is highly advisable that expensive numerical operations that depend on linear algebra are performed off-line, e.g. A.I. neural network training, machine learning algorithms etc.

Installation

Importing Math.NET for Unity from the Asset Store should add the following directory tree in your Assets folder. Note that it is advisable that you remove completely any previous versions of Math.NET for Unity before re-importing any updates.



```
Assets
  MathNET
    Documentation
    Examples
    Plugins
```

The only folder necessary for using the library is MathNET/Plugins that contains the dll along with XML documentation. To keep your source tree "clean" its best if you remove the Documentation and Examples folders once you're familiar enough with the library.

Within the Documentation folder, appart from documentation you will also find a zip file of the source code along with a large number of unit tests (~19000), that are from the original Math.NET library. Some unit tests are known to have convergence issues with mono (5 out of 19000) and have been removed.

Unsupported Platforms

Math.NET for Unity does not support legacy Unity platforms, e.g. Blackberry or Flash.

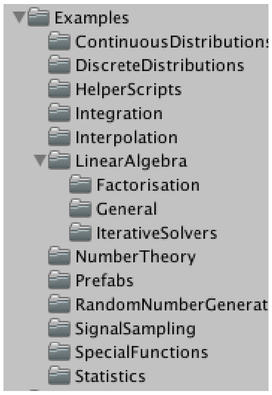
Linear Algebra Providers

Math.NET supports a number of native linear algebra providers, e.g. ATLAS, OpenBLAS, MKL, Apple's Accelerate framework, cuBLAS and others. In extension Math.NET for Unity also supports these third party native providers, however, as mentioned already these are not included. Nevertheless, it's a good point to consider for games that have to solve several (even small) linear systems periodically as it can be optimize performance, see for example Christoph Ruegg's blog that compares a native provider (MKL) to the managed provider used in Math.NET.

Quick Start

If you're already familiar with Math.NET Numerics just import Math.NET for Unity and start using it as you would the original. All namespaces and functions have been preserved.

For everyone else that is not familiar with the Math.NET library an excellent place to start is the Examples folder (see figure to the left).



There are hunders of usage examples with ample documentation within this folder. Usually the code contains links to relevant external sources, for example Wolfram Alpha and Wikipedia. A word of caution, for all its virtues Wikipedia is sometimes not a very accurate source of information on mathematics (and other things), so it is best to cross reference this information just to be on the safe side. Wolfram Alpha is much more reliable, but it is often very brief in its explanations. An example factorisation is given below:

```
MathDisplay.WriteLine("<b>Cholesky factorisation</b>");

// Format matrix output to console
var formatProvider = (CultureInfo)CultureInfo.InvariantCulture.Clone();
formatProvider.TextInfo.ListSeparator = " ";

// Create square, symmetric, positive definite matrix
var matrix = DenseMatrix.OfArray(new[,] { { 2.0, 1.0 }, { 1.0, 2.0 } });
MathDisplay.WriteLine(@"Initial square, symmetric, positive definite matrix");
MathDisplay.WriteLine(matrix.ToString("#0.00\t", formatProvider));
MathDisplay.WriteLine();

// Perform Cholesky decomposition
```

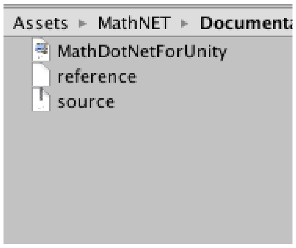
```
var cholesky = matrix.Cholesky();
MathDisplay.WriteLine(@"Perform Cholesky decomposition");
```

Documentation

This file along with the reference.zip archive comprise the documentation for Math.NET for Unity. The zip archive contains the approximate equivalent of 1300 A4 pages of documentation compiled from the original Math.NET Numerics library. Another source of information is the http://numerics.mathdotnet.com/ page. Lastly, the examples discussed above and the unit tests (see next section) are a good place to see how the library can be used. Note that as Math.NET for Unity evolves there will be increasingly more features not found in the original Math.NET Numerics library, therefore the documentation included in the reference.zip archive is the most authoritative source of information for Math.NET for Unity. That said, Math.NET for Unity will maintain API compatiblity with Math.NET Numerics, namely, functionality that is already available will continue to perform as expected.

Note: A nice chm reader for OSX is iCHM and for linux kCHMViewer is reliable.

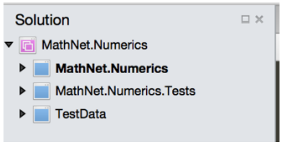
Source Code and Unit Tests (*Pro version ONLY*)



The source code along with the unit tests project are contained in the source.zip file. Make sure that you extract this zip archive outside of the Assets folder, otherwise this will result in name collisions with the Math.NET for Unity dll in the Plugins folder. Also, although you could in principle use the Math.NET for Unity source code within your project, its best if you use the dll as this may reduce compilation times for your project.

Running the Unit Tests

Once you extract the source.zip archive in some folder (other than the Assets folder) open the MathNet.Numerics.sln solution. This solution (see left) has three projects:

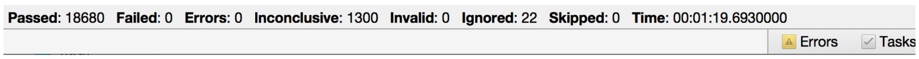


* MathNet.Numerics – This compiles the Math.NET for Unity library.

* MathNet.Numerics.Tests – This project contains all the unit tests for the Math.NET for Unity.

* TestData – This contains some supplementary data and is used only within the MathNet.Numerics.Tests.

To run the tests simply click on the unit tests tab to the right of monodevelop and press run all. Note that 1300 tests will be inconclusive, this is to be expected as certain operations in linear algebra are valid only for matrices that adhere to certain restrictions. When these restrictions are not met, the test is not performed and is registered as *inconclusive*. Once the test are complete you should see the following at the bottom bar of monodevelop:



Passed: 18680 **Failed:** 0 **Errors:** 0 **Inconclusive:** 1300 **Invalid:** 0 **Ignored:** 22 **Skipped:** 0 **Time:** 00:01:19.6930000

Support

Math.NET Numerics is a fairly complete library for mathematics, however, its possible that the good folks at http://numerics.mathdotnet.com/ have missed a function that could be useful. So, if you have suggestions or questions on how to use the library be sure to check out the forums (www.bismur.co.uk/forums) or contact us via email:

Bismur Studios Asset Support email: **asset.support@bismur.co.uk**

And don’t forget to check out www.bismur.co.uk for any updates and news.

Thank you for purchasing Math.NET for Unity, we hope you enjoy it and don’t forget to review this asset!