Vorcyc Mathematics （以下简称 VMath） 是一套基于.NET 8的算法库，其中包含了基础数学、位运算、线性代数、统计学、机器学习、数字信号处理等领域的相关算法，其目的旨在为开发人员提供一套统一的、标准的、高性能的并且能运行在异构设备上的API集。

我司已有一套数学算法库 Vorcyc MathHub ，它在未来将不再继续维护，而 VMath 将会作为一个全新的产品完全替换 MathHub。两者最大的区别是 MathHub 使用C/C++ 编写且面向本机编译，而VMath 则采用C#编写并且面向托管代码和利用AoT技术面向本机代码编译。

VMath 只支持.NET 8或更高版本，目的是为了充分利用.NET的最新特性以优化性能：包括.NET 5引入的 SIMD 指令编译和 .NET 7引入的泛型数学功能。

**Vorcyc.Mathematics.Numerics 命名空间**

1. BigFloat

一个表示高精度浮点数的结构体。

功能：

• 表示高精度浮点数

• 支持加、减、乘、除等运算

• 支持与其他数值类型的转换

• 支持平方根、幂运算等高级数学运算

使用示例：

BigFloat a = new BigFloat(123.456);

BigFloat b = new BigFloat("789.012");

BigFloat resultAdd = a + b;

BigFloat resultSub = a - b;

BigFloat resultMul = a \* b;

BigFloat resultDiv = a / b;

BigFloat resultSqrt = BigFloat.Sqrt(a);

BigFloat resultPow = BigFloat.Pow(a, 2);

Console.WriteLine($"Add: {resultAdd}");

Console.WriteLine($"Subtract: {resultSub}");

Console.WriteLine($"Multiply: {resultMul}");

Console.WriteLine($"Divide: {resultDiv}");

Console.WriteLine($"Sqrt: {resultSqrt}");

Console.WriteLine($"Pow: {resultPow}");

2. Complex<T>

一个表示复数的结构体，具有泛型数值类型。

功能：

• 表示复数

• 支持加、减、乘、除等运算

• 支持极坐标转换

• 支持求模、求共轭等操作

使用示例：

Complex<double> c1 = new Complex<double>(1.0, 2.0);

Complex<double> c2 = new Complex<double>(3.0, 4.0);

Complex<double> resultAdd = c1 + c2;

Complex<double> resultSub = c1 - c2;

Complex<double> resultMul = c1 \* c2;

Complex<double> resultDiv = c1 / c2;

double magnitude = Complex<double>.Abs(c1);

Complex<double> conjugate = Complex<double>.Conjugate(c1);

Console.WriteLine($"Add: {resultAdd}");

Console.WriteLine($"Subtract: {resultSub}");

Console.WriteLine($"Multiply: {resultMul}");

Console.WriteLine($"Divide: {resultDiv}");

Console.WriteLine($"Magnitude: {magnitude}");

Console.WriteLine($"Conjugate: {conjugate}");

3. ComplexFp32

一个表示复数的结构体，具有单精度浮点数分量。

功能：

• 表示复数

• 支持加、减、乘、除等运算

• 支持极坐标转换

• 支持求模、求共轭等操作

使用示例：

ComplexFp32 c1 = new ComplexFp32(1.0f, 2.0f);

ComplexFp32 c2 = new ComplexFp32(3.0f, 4.0f);

ComplexFp32 resultAdd = c1 + c2;

ComplexFp32 resultSub = c1 - c2;

ComplexFp32 resultMul = c1 \* c2;

ComplexFp32 resultDiv = c1 / c2;

float magnitude = ComplexFp32.Abs(c1);

ComplexFp32 conjugate = ComplexFp32.Conjugate(c1);

Console.WriteLine($"Add: {resultAdd}");

Console.WriteLine($"Subtract: {resultSub}");

Console.WriteLine($"Multiply: {resultMul}");

Console.WriteLine($"Divide: {resultDiv}");

Console.WriteLine($"Magnitude: {magnitude}");

Console.WriteLine($"Conjugate: {conjugate}");

4. Int24

一个表示24位有符号整数的结构体。

功能：

• 表示24位有符号整数

• 支持加、减、乘、除等运算

• 支持与其他数值类型的转换

使用示例：

Int24 a = new Int24(123456);

Int24 b = new Int24(654321);

Int24 resultAdd = a + b;

Int24 resultSub = a - b;

Int24 resultMul = a \* b;

Int24 resultDiv = a / b;

Console.WriteLine($"Add: {resultAdd}");

Console.WriteLine($"Subtract: {resultSub}");

Console.WriteLine($"Multiply: {resultMul}");

Console.WriteLine($"Divide: {resultDiv}");

5. IntRange

一个表示整数范围的结构体。

功能：

• 表示整数范围

• 检查值是否在范围内

• 计算范围的交集

使用示例：

IntRange range = new IntRange(1, 10);

bool isInside = range.IsInside(5);

IntRange intersection = range.Intersection(new IntRange(5, 15));

Console.WriteLine($"IsInside: {isInside}");

Console.WriteLine($"Intersection: [{intersection.Min}, {intersection.Max}]");

6. Number

一个表示数字的类，作为数字列表，具有任意精度。

功能：

• 表示任意精度的数字

• 支持加、减、乘、除等运算

• 支持与其他数值类型的转换

使用示例：

Number n1 = new Number("12345678901234567890");

Number n2 = new Number("98765432109876543210");

Number resultAdd = n1 + n2;

Number resultSub = n1 - n2;

Number resultMul = n1 \* n2;

Number resultDiv = n1 / n2;

Console.WriteLine($"Add: {resultAdd}");

Console.WriteLine($"Subtract: {resultSub}");

Console.WriteLine($"Multiply: {resultMul}");

Console.WriteLine($"Divide: {resultDiv}");

7. Point<T>

一个表示二维平面上点的结构体，具有泛型数值类型。

功能：

• 表示二维平面上的点

• 支持加、减运算

• 支持距离计算

使用示例：

Point<int> p1 = new Point<int>(1, 2);

Point<int> p2 = new Point<int>(3, 4);

Point<int> resultAdd = p1 + p2;

Point<int> resultSub = p1 - p2;

double distance = Point<int>.Distance(p1, p2);

Console.WriteLine($"Add: {resultAdd}");

Console.WriteLine($"Subtract: {resultSub}");

Console.WriteLine($"Distance: {distance}");

8. PointFp32

一个表示二维平面上点的结构体，具有单精度浮点数坐标。

功能：

• 表示二维平面上的点

• 支持加、减运算

• 支持距离计算

使用示例：

PointFp32 p1 = new PointFp32(1.0f, 2.0f);

PointFp32 p2 = new PointFp32(3.0f, 4.0f);

PointFp32 resultAdd = p1 + p2;

PointFp32 resultSub = p1 - p2;

float distance = PointFp32.Distance(p1, p2);

Console.WriteLine($"Add: {resultAdd}");

Console.WriteLine($"Subtract: {resultSub}");

Console.WriteLine($"Distance: {distance}");

9. Range<T>

一个表示值范围的类，具有泛型数值类型。

功能：

• 表示值范围

• 检查值是否在范围内

• 计算范围的交集和并集

使用示例：

Range<int> range = new Range<int>(1, 10);

bool isInside = range.IsInside(5);

Range<int> intersection = range.Intersection(new Range<int>(5, 15));

Range<int> union = range.Union(new Range<int>(5, 15));

Console.WriteLine($"IsInside: {isInside}");

Console.WriteLine($"Intersection: [{intersection.Minimum}, {intersection.Maximum}]");

Console.WriteLine($"Union: [{union.Minimum}, {union.Maximum}]");

10. Rational<T>

一个表示有理数的结构体，具有泛型整数类型。

功能：

• 表示有理数

• 支持加、减、乘、除等运算

• 支持与其他数值类型的转换

使用示例：

Rational<int> r1 = new Rational<int>(1, 2);

Rational<int> r2 = new Rational<int>(3, 4);

Rational<int> resultAdd = r1 + r2;

Rational<int> resultSub = r1 - r2;

Rational<int> resultMul = r1 \* r2;

Rational<int> resultDiv = r1 / r2;

Console.WriteLine($"Add: {resultAdd}");

Console.WriteLine($"Subtract: {resultSub}");

Console.WriteLine($"Multiply: {resultMul}");

Console.WriteLine($"Divide: {resultDiv}");

11. RectangleFP32

一个表示具有浮点数坐标和尺寸的矩形结构体。

功能：

• 表示矩形

• 检查点是否在矩形内

• 计算矩形的交集和并集

使用示例：

RectangleFP32 rect = new RectangleFP32(0.0f, 0.0f, 10.0f, 20.0f);

bool contains = rect.Contains(5.0f, 5.0f);

RectangleFP32 intersection = RectangleFP32.Intersect(rect, new RectangleFP32(5.0f, 5.0f, 10.0f, 20.0f));

RectangleFP32 union = RectangleFP32.Union(rect, new RectangleFP32(5.0f, 5.0f, 10.0f, 20.0f));

Console.WriteLine($"Contains: {contains}");

Console.WriteLine($"Intersection: {intersection}");

Console.WriteLine($"Union: {union}");

12. Size<T>

一个表示二维尺寸的结构体，具有泛型数值类型。

功能：

• 表示二维尺寸

• 支持加、减、乘、除等运算

使用示例：

Size<int> size1 = new Size<int>(10, 20);

Size<int> size2 = new Size<int>(5, 5);

Size<int> resultAdd = size1 + size2;

Size<int> resultSub = size1 - size2;

Size<int> resultMul = size1 \* 2;

Size<int> resultDiv = size1 / 2;

Console.WriteLine($"Add: {resultAdd}");

Console.WriteLine($"Subtract: {resultSub}");

Console.WriteLine($"Multiply: {resultMul}");

Console.WriteLine($"Divide: {resultDiv}");

13. SizeFp32

一个表示二维尺寸的结构体，具有单精度浮点数尺寸。

功能：

• 表示二维尺寸

• 支持加、减、乘、除等运算

使用示例：

SizeFp32 size1 = new SizeFp32(10.0f, 20.0f);

SizeFp32 size2 = new SizeFp32(5.0f, 5.0f);

SizeFp32 resultAdd = size1 + size2;

SizeFp32 resultSub = size1 - size2;

SizeFp32 resultMul = size1 \* 2.0f;

SizeFp32 resultDiv = size1 / 2.0f;

Console.WriteLine($"Add: {resultAdd}");

Console.WriteLine($"Subtract: {resultSub}");

Console.WriteLine($"Multiply: {resultMul}");

Console.WriteLine($"Divide: {resultDiv}");

14. UInt24

一个表示24位无符号整数的结构体。

功能：

• 表示24位无符号整数

• 支持加、减、乘、除等运算

• 支持与其他数值类型的转换

使用示例：

UInt24 a = new UInt24(123456);

UInt24 b = new UInt24(654321);

UInt24 resultAdd = a + b;

UInt24 resultSub = a - b;

UInt24 resultMul = a \* b;

UInt24 resultDiv = a / b;

Console.WriteLine($"Add: {resultAdd}");

Console.WriteLine($"Subtract: {resultSub}");

Console.WriteLine($"Multiply: {resultMul}");

Console.WriteLine($"Divide: {resultDiv}");