

# LIBRAPID HIGH PERFORMANCE COMPUTING

## **Current Features** **LibRapid (C++/Python)**

Toby “Pencilcaseman” Davis

January 2022



# Contents

<b>1</b>	<b>Array Storage, Operations and Calculations</b>	<b>3</b>
1.1	Extent . . . . .	4
1.1.1	Constructors . . . . .	4
1.1.2	Properties . . . . .	4
1.2	Stride . . . . .	4
1.2.1	Constructors . . . . .	4
1.2.2	Properties . . . . .	4
1.3	Array . . . . .	4
1.3.1	Constructors . . . . .	4
1.3.2	Properties . . . . .	4
1.3.3	Fill functions . . . . .	4
1.3.4	Clone Functions . . . . .	4
1.3.5	Mapping Functions . . . . .	4
1.3.6	Arithmetic Operations . . . . .	4
1.3.7	Reshaping . . . . .	5
1.3.8	Matrix Operations . . . . .	5
1.3.9	Casting . . . . .	5
1.3.10	String Conversions . . . . .	5
1.3.11	BLAS, Warm-up and Multi-threaded Optimization . . . . .	5
1.3.12	Filled Array Construction . . . . .	5
1.3.13	More Array Operations . . . . .	5
1.3.14	Implicit Casting . . . . .	5
1.4	Colors . . . . .	6
1.4.1	RGB and HSL Classes . . . . .	6
1.4.2	Conversions . . . . .	6
1.4.3	Convert between RGB and HSL values . . . . .	6
1.4.4	Merging . . . . .	6
1.4.5	Simple merging functionality between two RGB/HSL values . . . . .	6
1.4.6	Predefined Colors . . . . .	6
1.4.7	Foreground and Background Formatting . . . . .	7
1.5	Timing and General Utilities . . . . .	8
1.5.1	Get the current number of seconds since the Unix Epoch as a 64-bit float . . . . .	8
1.5.2	Halt program execution for a given number of seconds . . . . .	8
1.5.3	Get the rows and columns of the current terminal window . . . . .	8
1.6	Mathematics . . . . .	9
1.6.1	Constants . . . . .	9
1.6.2	List Calculations . . . . .	10
1.6.3	Mapping a Number to a Range . . . . .	10
1.6.4	Generate a random integer in a given range . . . . .	10
1.6.5	Generate a random floating point value in a given range . . . . .	10
1.6.6	An optimized function to calculate powers of 10 . . . . .	10
1.6.7	Round a number to a given number of decimal places . . . . .	10
1.6.8	Round a number to a given number of significant figures . . . . .	10
1.6.9	Vectors and Vector Arithmetic . . . . .	10

## Chapter 1

# Array Storage, Operations and Calculations

A multi-dimensional array library supporting runtime-defined datatypes, custom mapping functions and highly optimized routines.

## 1.1 Extent

### 1.1.1 Constructors

Default Constructor

Construct from various STL types

Construct another Extent

### 1.1.2 Properties

Access dimensions, find automatic values, etc.

## 1.2 Stride

### 1.2.1 Constructors

Default Constructor

Construct from various STL types

Construct another Stride

Generate new stride from an Extent object

### 1.2.2 Properties

Access dimensions, triviality, contiguity, etc.

## 1.3 Array

### 1.3.1 Constructors

Construct from Extent object, with datatype and CPU/GPU parameters

Create from arbitrary input data

Create from an existing Array

### 1.3.2 Properties

Access dimensions, stride, location, datatype, etc.

Get and set items within the Array (array or scalar values)

### 1.3.3 Fill functions

Fill an array with a particular value, or with randomly generated values

### 1.3.4 Clone Functions

Create a raw clone of an array

Create a raw clone of an array and cast to a different datatype

Create a raw clone of an array on a different device (CPU/GPU)

### 1.3.5 Mapping Functions

Apply a kernel to any number of input arrays, optimizing the loops and resulting in a dramatic speed improvement

### 1.3.6 Arithmetic Operations

Binary operations where both the LHS and RHS are Arrays

Binary operations where only the LHS is an Array

Binary operations where only the RHS is an Array

In-place binary operations where both the LHS and RHS are Arrays

In-place binary operations where only the LHS is an Array

### 1.3.7 Reshaping

Reshape an array in-place based on a list-like object (list, Extent, etc.)

Reshape an array based on a list-like object (list, Extent, etc.), returning the result

### 1.3.8 Matrix Operations

Transpose an array in-place, or return the transposed Array

Calculate vector-vector products, matrix-vector products, matrix-matrix products, and higher-dimensional products as well

### 1.3.9 Casting

Convert a value from an Array into a Boolean, integer, floating point or complex datatype

### 1.3.10 String Conversions

Return a string representation of an Array object, formatted so that decimal points align and large arrays have the middle values stripped out

Output an Array with Python's `print()` function, C++'s `std::cout` function or `fmt::print()`

### 1.3.11 BLAS, Warm-up and Multi-threaded Optimization

Check if LibRapid was compiled with BLAS

Set the number of threads LibRapid uses

Get the number of threads LibRapid uses

Run some benchmarks and optimize the number of threads LibRapid uses

### 1.3.12 Filled Array Construction

Create an exact replica of an Array, except filled with zeros, ones or random numbers

### 1.3.13 More Array Operations

Binary arithmetic on two Array-like objects, returning the result

Binary arithmetic on two Array-like objects, storing the result in a third Array

Compute the dot product on two Array-like objects, either returning the result or storing the result in a third Array

Stack and concatenate multiple Array into a single, larger Array

### 1.3.14 Implicit Casting

Implicit cast between scalar types and list-like types in C++ and Python

## 1.4 Colors

### 1.4.1 RGB and HSL Classes

### 1.4.2 Conversions

### 1.4.3 Convert between RGB and HSL values

### 1.4.4 Merging

### 1.4.5 Simple merging functionality between two RGB/HSL values

### 1.4.6 Predefined Colors

clear

bold

blink

black

red

green

yellow

blue

magenta

cyan

white

brightBlack

brightRed

brightGreen

brightYellow

brightBlue

brightMagenta

brightCyan

brightWhite

blackBackground

redBackground

greenBackground

yellowBackground

blueBackground

magentaBackground

cyanBackground

whiteBackground

brightBlackBackground

brightRedBackground

brightGreenBackground

brightYellowBackground

`brightBlueBackground`

`brightMagentaBackground`

`brightCyanBackground`

`brightWhiteBackground`

### 1.4.7 Foreground and Background Formatting

Automatically print a foreground or background color, allowing for colored printing



## 1.5 Timing and General Utilities

1.5.1 Get the current number of seconds since the Unix Epoch as a 64-bit float

1.5.2 Halt program execution for a given number of seconds

1.5.3 Get the rows and columns of the current terminal window

## 1.6 Mathematics

### 1.6.1 Constants

Convert between RGB and HSL values

Backhouse's Constant

Bloch-Landau Constant

Cahen's Constant

$$\sqrt[3]{2}$$

$$\sqrt[3]{3}$$

$$\pi/180$$

$$180/\pi$$

Dottie Number

Euler's Number  $e$

Gravity of Earth

$e^\pi$ , Gelfond's Constant

Euler-Mascheroni Constant  $\gamma$

Feller-Tornier Constant

Gauss's Constant

Gieseking Constant

Golden Angle

Golden Ratio  $\phi$

Golomb-Dickman Constant

$$2\pi$$

$$\pi/2$$

Khinchin's Constant

Laplace limit  $\lambda$

Speed of Light

$$\ln 2$$

$$\ln 3$$

$$\ln 5$$

Loch's Constant

Area of the Mandelbrot Fractal

Meissel-Mertens Constant

Mill's Constant

MRB Constant

Niven's Constant

Parabolic Constant  $P_2$

$$\pi$$

$$\pi/\ln 2$$

$$\pi^2/6$$

Plastic Number

Somos' Quadratic Recurrence Constant

Reciprocal Fibonacci Constant

$\sqrt{2}$

$\sqrt{3}$

$\sqrt{5}$

$\sqrt{e}$

$\sqrt{\pi}$

$\tau$

Twin Primes Constant

$2^{\sqrt{2}}$

Wallis Constant

$\zeta(3)$

Conversions

### 1.6.2 List Calculations

Calculate the product of a list-like object

Find the minimum or maximum value of a list-like object

### 1.6.3 Mapping a Number to a Range

$$\frac{(x - \min_i) \cdot (\max_o - \min_o)}{\max_i - \min_i} + \min_o$$

### 1.6.4 Generate a random integer in a given range

### 1.6.5 Generate a random floating point value in a given range

### 1.6.6 An optimized function to calculate powers of 10

$$10^n$$

### 1.6.7 Round a number to a given number of decimal places

### 1.6.8 Round a number to a given number of significant figures

### 1.6.9 Vectors and Vector Arithmetic

Support for n-dimensional vectors with any datatype

Construct vectors from various objects and types

Binary Vector arithmetic with Vectors or scalar values as the RHS

In-place binary arithmetic

Vector magnitude

Vector magnitude squared

Inverse vector magnitude

Vector dot product

Vector cross product

Access to any component of vectors, including variables for x, y, z, and w

Explicit typedefs for Vec2i, Vec2f, Vec2d, Vec3i, Vec3f, Vec3d, Vec4i, Vec4f, and Vec4d

String formatting for Vector objects