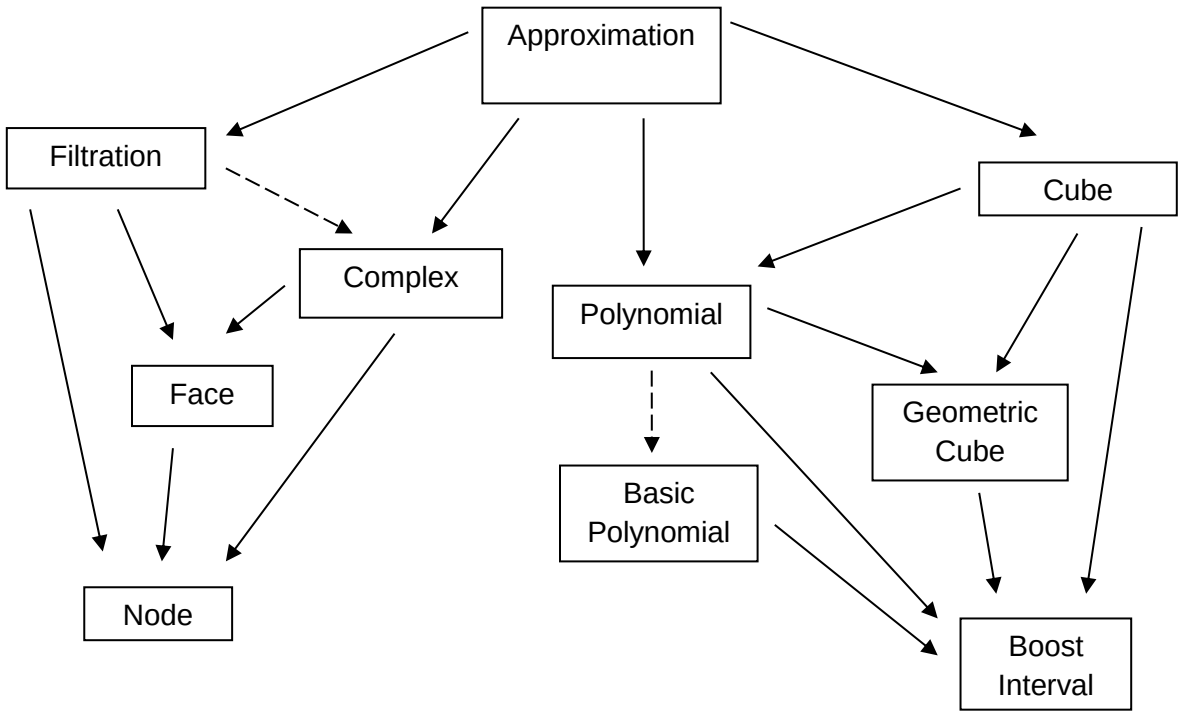
Reference Guide

This document summarizes each of the Classes and Structures defined in this software package. Refer to http://www.boost.org/ for documentation on the Boost Interval library.

In order to compute persistent homology there are two main tasks: (1) testing the local properties of the primary function and (2) manipulating the CW complexes we use to approximate the super-level sets. The classes and structures used in this program are largely divided between accomplishing one of these two tasks. After a brief summary of each class/structure we present a more detailed description.



The figure above describes the dependency relationships between the various classes and structures. Dashed arrows represent class inheritance. For the most part in the code, class names contain the prefix "C" and the structures have the prefix "S" (i.e. the Approximation class is defined as "CApproximation".)

Approximation
This class is used to create discrete approximations of a function's super-level sets and the corresponding filtration.

Basic Polynomial
This class stores the user's input function and calculates the value and derivative of that function on Geometric Cubes. Despite the name, the input function does not have to be a polynomial.

Complex
This class is a quad-tree and is used to store the discrete approximation to a super-level set of the input function.

Cube
This class represents a 2 dimensional dyadic cube by a vector of integers. This class is used to check whether the input function is well behaved locally, on that dyadic cube.

Face
This structure represents a single cell from a dyadic cube.

Filtration
This class is derived from the base class Complex. It represents a CW complex on the entire unit square and is used to export a filtered algebraic chain complex.

Geometric Cube
This structure represents a 0-, 1-, or 2-dimensional dyadic cube by two Intervals. Primarily serves as the input to the Basic Polynomial class.

Node
This class represents a 2 dimensional dyadic cube and is the building block of the class Complex. This class stores which cells from the dyadic cube are included in the Complex and in Filtration, they store when each cell is born.

Polynomial
This class is derived from the base class Basic Polynomial. The Polynomial class and which improves the precision of the interval arithmetic and test whether the function's value/derivatives on a GeometricCube are above/below a certain value.

APPROXIMATION

STORED DATA

Maximum Depth

- Description: An unsigned integer which define the maximum number times the Verified Approximation function is allowed to subdivide the unit cube before terminating.

Polynomial

- Description: This is the function whose persistent homology the user wishes to study.

PUBLIC FUNCTIONS

Define Interval Subdivisions

- Inputs: An unsigned integer
- Description: Defines the amount of preprocessing used in Polynomial's interval arithmetic.

Define Maximum Depth

- Inputs: An unsigned integer
- Description: Changes the member Maximum Depth to equal the user's input.

Intersect Complexes

- Inputs: Two Complexes
- Outputs: None; this function modifies the second of the input complexes
- Description: Removes cells from the second complex so that it is then equal to the intersection of the two given.

Modify Output

- Inputs: A bottom threshold (type double), a step size (type double), a maximum number of steps (type unsigned integer) and a sub/super-level set directive (type Boole)
- Description: This function uses the output from Perseus to create analogous files with the real-valued birth/death times corresponding to the thresholds used.

Verify Approximation

- Inputs: A Complex, a threshold (type double), and a sub/super-level set directive (Boole)
- Description: Modifies the input Complex to construct a verified approximation to the sub/super-level set (corresponding to the input being true/false) of the Polynomial at the given threshold.

Verify Filtration

- Inputs: A bottom threshold (type double), a step size (type double), a maximum number of steps (type unsigned integer), a sub/super-level set directive (Boole) and a Filtration
- Outputs: Modifies the Filtration object and externally produces a text file.
- Description: Constructs a filtration corresponding to the sub/super-level sets of the primary function. The thresholds used for the filtration start at the "bottom threshold" and increase by the amount "step size" to include a total number of thresholds equal to the "maximum number of steps". For each threshold, this function runs Verify Approximation, generating a verified Complex. After taking the appropriate intersections, it adds these complexes to the input Filtration. Lastly, the Filtration exports a filtered chain complex.
- Note: *Including from the Filtration object given to this function, Verify Filtration stores at most three Complexes in memory at any one time.*

BASIC POLYNOMIAL

PUBLIC FUNCTIONS

Value

- Inputs: Geometric Cube.
- Outputs: Interval
- Description: The function calculates the image of the polynomial on the geometric cube.

Derivative

- Inputs: Geometric Cube.
- Outputs: 2x1 vector of Intervals
- Description: The image of the polynomial's Jacobian on the geometric cube.

COMPLEX

STORED DATA

Root

- Description: A Node representing the entire unit square and the root of the quad-tree.

PRIVATE FUNCTIONS

Infer Faces

- Inputs: A Boolean vector of length four, representing which of cube's vertices are in the super-level set.
- Outputs: A Boolean vector of length nine, representing the entire cube's cells which are included in the approximation, exactly analogous to the Faces member in the Node class.
- Description: A cell is added to the approximation if and only if all of its vertices are in the approximation.

PROTECTED FUNCTIONS

Redundant Cells

- Inputs: A Face
- Outputs: A vector of Faces
- Description: This function returns all of the Faces which are in the Complex and intersect the input Face.

PUBLIC FUNCTIONS

Add Cube

- Inputs: The location of a cube (vector of unsigned integers) and which vertices of the cube are in the approximation (length four vector of Boolean values).
- Description: This function will add a node to the complex corresponding to the input location, and use Infer Faces to store the necessary cells.

Boundary

- Inputs: A Face and Boolean value
- Outputs: A vector of Faces
- Description: If the Boolean input is true/false, this function return all of the faces in the boundary with a coefficient of +1 / -1.

Create CW Complex

- Description: When an approximation is initially constructed, a single point in the unit cube may be contained in multiple cells. For each node in the complex (and each cell on that node), this function will find the corresponding redundant cells, and will remove which ever cells are necessary to make the resultant structure an actual CW-complex.

Obtain Carrier

- Inputs: A Face (with correct Path data; the pointer to Node doesn't matter)
- Outputs: A vector of Faces
- Description: This function returns all of the faces in the complex which intersect a given Face. This can be used to find the carriers of a cell from a different complex.

CUBE

STORED DATA

Location

- Description: A vector of unsigned integers which provide the location of the cube in the class Complex.

PUBLIC FUNCTIONS

Get Geometric Representation

- Outputs: Geometric Cube
- Description: Returns the Geometric Cube corresponding to the Cube object.

Get Location

- Outputs: An vector of unsigned integers
- Description: Returns the object member “Location”.

Local Approximation

- Inputs: A polynomial and a threshold (of type double).
- Outputs: A Boolean vector (of length 4)
- Description: Calculates which of the vertices of the cube have an image above the threshold, and therefore in the approximation.

Print

- Description: Uses `std::cout` to print the object member Location.

Subdivide Cube

- Outputs: A vector of 4 cubes whose geometric union equals the initial cube.

Verify

- Inputs: A Polynomial, and a threshold (of type double).
- Outputs: One of three integers with the following interpretation:
    - **-1)** One of the cube’s vertices had an image which was neither above nor below the threshold.
    - **0)** The cube failed to be verified.
    - **1)** The cube succeeded in being verified.
- Description: Checks whether the cube is analytically verified by seeing if
    - *(Verification by Sign)* The image of the function on the cube is bounded away from the threshold
    - *(Strong Verification by Derivative)* The image of both partial derivatives of the function on the cube are bounded away from zero
    - *(Weak Verification by Derivative)* The image of one of the partial derivatives of the function on the cube is bounded away from zero and two specific edges are verified by sign.

FACE

STORED DATA

Path

- Description: A vector of unsigned integers which provide the location of the dyadic cube containing the Face; analogous to the location member of the Cube class.

Pointer to Node

- Description: A pointer to the Node representing the dyadic cube which contains the face.
- Note: *Both the Path member and the Pointer to Node member uniquely describe the location of a dyadic cube in a Complex. When a Face structure is used, one of these members will be defined, but not necessarily the other.*

Face Number

- Description: An integer corresponding to one of the 9 cells in a dyadic cube. Index the faces is given thusly:
    - The number **0** corresponds to the single 2-cell
    - The numbers **1-4** correspond to the 1-cells, ordered counter clockwise, starting with the bottom edge
    - The numbers **5-8** correspond to the vertices, ordered as follows::
      (0,0) ; (0,1) ; (1,0) ; (1,1)

PUBLIC FUNCTIONS

Birth Time

- Description: Returns when the cell was born in the Filtration.

Check Cell

- Description: Determines whether the cell is in the Complex.

Define Export Order

- Input: An Integer
- Description: Defines the order in which the cell was written to the export file to be the user's input.

Define Birth Time

- Input: An integer
- Description: Defines the birth time of the cell in the Filtration to be the input of the user.

Delete

- Description: Removes the given cell from the Complex.

Export Order

- Description: Returns the reference number of the cell in relation to when that cell was written to the export file.

FILTRATION

ADDITIONAL DATA

Zero, One and Two – Cell Lists

- Description: Three vectors of Faces, which linearly store the cells of dimensions zero, one and two in the Filtration.

PUBLIC FUNCTIONS

Add Complex

- Input: Complex, threshold (type integer)
- Description: Adds all of the cells from the complex into the filtration, at the given threshold value, refining any cells as need be.
- Note: *It is assumed that complexes will be added to the filtration in increasing order of real thresholds. Since the super-level sets shrink as the real threshold increase, it is assumed that the integer thresholds passed to Add Complex will be decreasing.*

Export

- Output: Externally produces a text file.
- Description: Generates a file which lists all of the 0, 1 and 2-cells in the complex, along with their birth times, and boundary information.
    - The output file lists each cell of the CW complex on its own line of the file. First zero-dimensional cells are listed, then one-dimensional cells, and then finally two-dimensional cells. Before listing cells of a new dimension, the number "-1" is inserted on a new line. For each cell, we include integers which encode both where the cell appears in the filtration and its boundary information.
    - The first number on each line corresponds to the birth time of the cell. This is a positive integer valued number. Cells with a higher birth time are born later. The condition that all cells are born by `max_steps+1` is imposed.
        - For sub-level set filtration a cell with birth time “1” corresponds to a cell born at the bottom threshold.
        - For a super-level set filtration a cell with birth time “1” corresponds to a cell born at the top threshold.
    - The second number on each line corresponds to the total number of elements in the boundary of the cell. After that, the elements in the boundary of the cell are listed by their reference number. Each element in the boundary is preceded by a coefficient (being either 1 or -1).
    - A cell's reference number is given by the order in which appeared after an instance of "-1". For example, the reference number corresponding to the first 1-cell appearing in the file is “0”.

Initialize

- Input: An integer
- Description: All cells in the Filtration must be born by a certain point. This function specifies that point to be the input integer.
- Note: *A Filtration has to perform the Initialize function before the Add Complex function can be called.*

GEOMETRIC CUBE

STORED DATA

Dimension

- Description: This integer defines the ambient dimension of the cube.
- Note: *While in the current software release this variable is always set to 2, the hope is that future versions will allow the user to study functions defined on cubes of higher dimensions.*

Coordinates

- Description: This is a vector of intervals corresponding to the coordinates of the cube.

PUBLIC FUNCTIONS

Subdivide

- Output: A vector of Geometric Cubes
- Description: Divides the original geometric cube in half in every dimension and returns all of the constituent pieces. If the original geometric cube is a vertex then this function returns a vector containing the original geometric cube.

Print

- Description: Uses `std::cout` to print the coordinates of the geometric cube.

NODE

STORED DATA

Parent

- Description: A pointer to the parent node.

Children

- Description: A vector of pointers to nodes (the children of the given node).

Faces

- Description: A vector of type Boolean, describing which of the faces in the corresponding cube are included in the approximation. Indexing the faces is given thusly:
    - The first element is the single 2-cell
    - The next elements are the 1-cells, ordered counter clockwise, starting with the bottom edge
    - The vertices are then listed and ordered as follows:: (0,0) ; (0,1) ; (1,0) ; (1,1)

FILTRATION DATA

Birth Time

- Description: A vector of integers corresponding to the faces, describing the cell's birth time.

Export Order

- Description: A vector of integers corresponding to the faces, describing the cell's order in which it is written to the output file.

PUBLIC FUNCTIONS

Filtered Subdivide

- Description: When a Complex is being added to a Filtration, certain cells need to be refined into smaller cells. This function subdivides a Node (from a Filtration object) into four new nodes and defines the birth time of each of the new cells (in the new nodes) to be the birth time of the cell (from the original node) in which they are contained.

Leaf Check

- Output: Boolean
- Description: This function returns a Boolean value describing whether or not the given node is a leaf, i.e. if the vector Children is empty.
- Note: *For any node, either the Children or Faces vector should be empty.*

Split

- Description: This function creates four new nodes and stores pointers to them.

Store Faces

- Inputs: A Boolean vector describing which faces the user wishes to store.
- Description: This function copies the input to the object's face vector

POLYNOMIAL

STORED DATA

Interval Subdivisions

- Description: An unsigned integer which dictates how many times a Geometric Cube is subdivided when performing interval arithmetic.

PRIVATE FUNCTIONS

Multiple Subdivisions

- Inputs: Geometric Cube
- Outputs: Vector of Geometric Cubes
- Description: Subdivides the input geometric cube “Interval Subdivisions” many times and returns the constituent pieces.

Value Efficient

- Inputs: Geometric Cube.
- Outputs: Interval
- Description: Calculates the image of the polynomial on the geometric cube. Subdivides the cube several times to obtain tighter Interval bounds.

Derivative Efficient

- Inputs: Geometric Cube.
- Outputs: Length-2 vector of Intervals
- Description: Calculates the image of the polynomial’s Jacobian on the geometric cube. Subdivides the cube several times to obtain tighter Interval bounds.

PUBLIC FUNCTIONS

Define Interval Subdivision

- Inputs: Unsigned Integer
- Description: Redefines the object member “Interval Subdivisions” according to the user’s input.

Value Test

- Inputs: Geometric Cube, Threshold (type double)
- Outputs: Boolean;
- Description: This function determines whether the image of the geometric cube has a minimum which is greater than the threshold or has a maximum which is lower than the threshold.

Derivative Test

- Inputs: Geometric Cube
- Outputs: Length-2 vector of Booleans;
- Description: The i^{th} Boolean value refers to whether or the i^{th} partial derivative of the function is bounded away from zero on the geometric cube.