```
Page 1/2
Name: Jude Onvia
Student ID: V00947095
Email: judeonyia10@gmail.com
Course: ECE596C
Section: T01
Assignment ID: cpp_arithmetic
Assignment Title: Interval Arithmetic and Applications
Submission Source: https://github.com/uvic-seng475-2020-05/cpp_arithmetic-JudeOn
via.qit
Commit ID: 301c29dfd89ae92f8524e4fb633cefe4c7df11b1
Submitted Files
______
drwxrwxr-x 135 2020-06-15 00:50 ./app

-rw-rw-r- 4120 2020-06-15 00:50 ./app/delaunay_triangulation.cpp

-rw-rw-r- 2427 2020-06-15 00:50 ./app/test_interval.cpp

-rw-rw-r- 3586 2020-06-15 00:50 ./app/test_kernel.cpp
-rw-rw-r--
              22880 2020-06-15 00:50 ./app/triangulation_2.hpp
             740 2020-06-15 00:50 ./CMakeLists.txt
-rw-rw-r--
               144 2020-06-15 00:50 ./IDENTIFICATION.txt
-rw-rw-r--
                 24 2020-06-15 00:50 ./include
drwxrwxr-x
                 56 2020-06-15 00:50 ./include/ra
drwxrwxr-x
              6290 2020-06-15 00:50 ./include/ra/interval.hpp
-rw-rw-r--
-rw-rw-r-- 11224 2020-06-15 00:50 ./include/ra/kernel.hpp
-rw-rw-r-- 113754 2020-06-15 00:50 ./README.pdf
Results
_____
                  Operation Target
Package
                                                 Status
nonprog
                  generate ---
                                                OK (0.0s)
interval_oriq
                 generate ---
                                                OK (0.3s)
                  configure ---
interval_oriq
                                                OK (2.3s)
interval_orig
                 build test_interval
                                                OK (0.8s)
interval_orig
                 build
                            test_kernel
                                                OK (5.2s)
                  generate ---
                                                OK (0.4s)
interval_sane
                  configure ---
interval_sane
                                                OK (2.1s)
```

Normally, an operation is indicated as having a status of either "OK" or "FAIL". A status of "?" indicates that the operation could not be performed for some reason (e.g., due to an earlier error or being a manual step). The time (in seconds) required for an operation is denoted by an expression consisting of a number followed by the letter "s" (e.g., "5.0s"). In the case of a test that consists of multiple test cases, the number of failed test cases and total number of test cases is expressed as a fraction (e.g., "10/50" means 10 test cases failed out of 50 test cases in total). The length (in lines) of the log file generated by an operation is denoted by an expression consisting of a number followed by the letter "L" (e.g., "10L"). To ascertain the reason for the failure of an operation, check the contents of the log file provided.

delaunay_triangu OK (8.2s)

FAIL (2 0.9s 95L)

OK (0.4s)

OK (2.1s)

FAIL (2 4.5s 183L)

build test_interval

test_kernel

build

build

generate ---

configure ---

interval_sane

interval_sane

deltri_oriq

deltri_orig deltri_oriq Legend

Package: nonprog

Nonprogramming exercises

Package: interval_orig

The code as originally submitted by the student.

Build target: test_interval

Build the test_interval program.

Build target: test_kernel

Build the test_kernel program.

Package: interval_sane

Code with modifications to perform API sanity checking.

Build target: test_interval

Build the (dummy) test_interval program.

Build target: test_kernel

Build the (dummy) test_kernel program.

Package: deltri_orig

The code as originally submitted by the student.

Build target: delaunay_triangulation

Build the delaunay_triangulation program.

CMakeFiles/test_interval.dir/app/test_interval.cpp.o -c $/home/jude on yia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jude on yia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jude on yia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jude on yia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jude on yia/Documents/ECE596C_Assgn_3/cpp_arithmetic-Jude on yia/Documents/ECE596C_Assgn_3/cpp_arithmetic-Documents/ECE596C_Assgn_3/cpp_arithmetic-Documents$ 61 eOnyia/cktmp/package-interval_sane/source/app/test_interval.cpp

udeOnyia/cktmp/package-interval_sane/source/include -pedantic-errors

[50%] Building CXX object CMakeFiles/test_interval.dir/app/test_interval.cpp.o

-I/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J

/home/frodo/public/ugls_lab-4.0.70/bin/c++

-frounding-math -std=gnu++17 -o

59

60

```
Jun 15, 20 0:50
```

Log: interval sane build test interval

Page 2/2

```
In file included from
   /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
   eOnyia/cktmp/package-interval_sane/source/app/test_interval.cpp:4:
   /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
   eOnyia/cktmp/package-interval_sane/source/include/ra/interval.hpp:116:73:
  error: C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@
   M-^Y or
   âM-^@M-^X-std=gnu++2aâM-^@M-^Y [-Wpedantic]
     116 | typename interval<real_type>::statistics interval<real_type>::stat_ =
70
   {.indeterminate_result_count = 0, .arithmetic_op_count = 0};
71
72
73
74
   /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
   eOnyia/cktmp/package-interval_sane/source/include/ra/interval.hpp:116:106:
75
   error: C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@
   M-^Y or
   âM-^@M-^X-std=gnu++2aâM-^@M-^Y [-Wpedantic]
77
           typename interval<real_type>::statistics interval<real_type>::stat_ =
   {.indeterminate_result_count = 0, .arithmetic_op_count = 0};
80
81
   qmake[3]: *** [CMakeFiles/test_interval.dir/app/test_interval.cpp.o] Error 1
82
   gmake[3]: Leaving directory
   '/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
   deOnyia/cktmp/package-interval_sane/derived'
   gmake[2]: *** [CMakeFiles/test_interval.dir/all] Error 2
86
   gmake[2]: Leaving directory
87
   '/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
88
   deOnyia/cktmp/package-interval_sane/derived'
89
   qmake[1]: *** [CMakeFiles/test_interval.dir/rule] Error 2
90
   gmake[1]: Leaving directory
91
   '/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
   deOnyia/cktmp/package-interval_sane/derived'
93
   gmake: *** [test_interval] Error 2
  ERROR: build failed to generate executable test_interval
```

Log: interval sane build test kernel Jun 15, 20 0:50 Page 1/4 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake -S/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J udeOnyia/cktmp/package-interval_sane/source -B/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J udeOnyia/cktmp/package-interval_sane/derived --check-build-system CMakeFiles/Makefile.cmake 0 /usr/bin/gmake -f CMakeFiles/Makefile2 test_kernel gmake[1]: Entering directory '/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju deOnyia/cktmp/package-interval_sane/derived' 10 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake 11 -S/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J 12 udeOnyia/cktmp/package-interval_sane/source 13 -B/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J 14 udeOnyia/cktmp/package-interval_sane/derived --check-build-system 15 CMakeFiles/Makefile.cmake 0 17 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake -E 18 cmake_progress_start /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud 19 eOnyia/cktmp/package-interval_sane/derived/CMakeFiles 2 20 /usr/bin/gmake -f CMakeFiles/Makefile2 CMakeFiles/test_kernel.dir/all 21 gmake[2]: Entering directory 22 '/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju deOnyia/cktmp/package-interval_sane/derived' 25 /usr/bin/gmake -f CMakeFiles/test_kernel.dir/build.make CMakeFiles/test_kernel.dir/depend 26 gmake[3]: Entering directory 27 \home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju 28 29 deOnyia/cktmp/package-interval_sane/derived' 30 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud 31 32 eOnyia/cktmp/package-interval_sane/derived && /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake -E cmake_depends "Unix Makefiles" /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud 35 eOnyia/cktmp/package-interval_sane/source 36 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud 37 eOnyia/cktmp/package-interval_sane/source 38 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud eOnyia/cktmp/package-interval_sane/derived /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud 41 eOnyia/cktmp/package-interval_sane/derived 42 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud 43 eOnyia/cktmp/package-interval_sane/derived/CMakeFiles/test_kernel.dir/DependInfo 44 .cmake --color= 45 Scanning dependencies of target test_kernel 46 gmake[3]: Leaving directory '/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju 48 deOnyia/cktmp/package-interval_sane/derived' 49 /usr/bin/gmake -f CMakeFiles/test_kernel.dir/build.make 50 CMakeFiles/test_kernel.dir/build 51 gmake[3]: Entering directory 52 '/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju 53 54 deOnyia/cktmp/package-interval_sane/derived' [50%] Building CXX object CMakeFiles/test_kernel.dir/app/test_kernel.cpp.o /home/frodo/public/ugls_lab-4.0.70/bin/c++ -I/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J 57 udeOnyia/cktmp/package-interval_sane/source/include -pedantic-errors

/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud

-frounding-math -std=gnu++17 -o

CMakeFiles/test_kernel.dir/app/test_kernel.cpp.o -c

eOnyia/cktmp/package-interval_sane/source/app/test_kernel.cpp

59

60

61

```
In file included from
   /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
   eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:6,
   /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
67
   eOnyia/cktmp/package-interval_sane/source/app/test_kernel.cpp:3:
   /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
   eOnyia/cktmp/package-interval_sane/source/include/ra/interval.hpp:116:73:
   error: C++ designated initializers only available with aM-^0M-^X-std=c++2aaM-^0
   M-^Y or
   âM-^@M-^X-std=gnu++2aâM-^@M-^Y [-Wpedantic]
72
     116 | typename interval<real_type>::statistics interval<real_type>::stat_ =
73
   {.indeterminate_result_count = 0, .arithmetic_op_count = 0};
74
75
76
   /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
77
   eOnyia/cktmp/package-interval_sane/source/include/ra/interval.hpp:116:106:
78
   error: C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@
79
   M-^Y or
   âM-^@M-^X-std=gnu++2aâM-^@M-^Y [-Wpedantic]
80
     116 | typename interval<real_type>::statistics interval<real_type>::stat_ =
81
   {.indeterminate_result_count = 0, .arithmetic_op_count = 0};
82
85
   In file included from
   /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
86
   eOnyia/cktmp/package-interval_sane/source/app/test_kernel.cpp:3:
87
   /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
88
   eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:25:35: warning:
89
   declaration âM-^@M-^Xstruct ra::math::indeterminate_resultâM-^@M-^Y does not dec
   lare anything
91
               using idr = typename ra::math::indeterminate_result;
                                               ^~~~~~~~~~~~~~~~~~
   /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
93
   eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:27:32: warning:
   declaration âM-^@M-^Xclass CGAL::MP_FloatâM-^@M-^Y does not declare anything
95
      27
               using exct = typename CGAL::MP_Float;
96
97
98
   In file included from
   /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
   eOnyia/cktmp/package-interval_sane/source/app/test_kernel.cpp:3:
   /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
   eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:253:59: error:
102
   C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@M-^Y or
103
   aM^-\ensuremath{^{\circ}}M^-\xspace X-std=gnu++2aaM^-\ensuremath{^{\circ}}M^-\xspace Y
   [-Wpedantic]
104
     253 | typename Kernel<Real>::Statistics Kernel<Real>::stat_ =
   {.orientation_total_count = 0, .orientation_exact_count = 0,
106
   .preferred_direction_total_count = 0, .preferred_direction_exact_count = 0,
   .side_of_oriented_circle_total_count = 0, .side_of_oriented_circle_exact_count
108
   = 0 };
109
110
   /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
111
   eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:253:89: error:
   C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@M-^Y or
   aM-^0M-^X-std=gnu++2aaM-^0M-^Y
   [-Wpedantic]
114
     253 |
115
            typename Kernel<Real>::Statistics Kernel<Real>::stat_ =
   {.orientation_total_count = 0, .orientation_exact_count = 0,
116
   .preferred_direction_total_count = 0, .preferred_direction_exact_count = 0,
117
   .side_of_oriented_circle_total_count = 0, .side_of_oriented_circle_exact_count
118
119
```

```
Log: interval sane build test kernel
Jun 15, 20 0:50
                                                                                 Page 3/4
120
121
    /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
122
    eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:253:119: error:
123
    C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@M-^Y or
    aM^-\M-x-std=gnu++2aaM^-\M-xY
    [-Wpedantic]
             typename Kernel<Real>::Statistics Kernel<Real>::stat_ =
      253
    {.orientation_total_count = 0, .orientation_exact_count = 0,
127
    .preferred_direction_total_count = 0, .preferred_direction_exact_count = 0,
128
    .side_of_oriented_circle_total_count = 0, .side_of_oriented_circle_exact_count
129
    = 0;
130
131
132
    /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
133
    eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:253:157: error:
    C++ designated initializers only available with aM-^0M-^X-std=c++2aaM-^0M-^Y or
    aM^-\ensuremath{^{\circ}}M^-\xspace X-std=gnu++2aaM^-\ensuremath{^{\circ}}M^-\xspace Y
    [-Wpedantic]
136
             typename Kernel<Real>::Statistics Kernel<Real>::stat_ =
137
    {.orientation_total_count = 0, .orientation_exact_count = 0,
138
    .preferred_direction_total_count = 0, .preferred_direction_exact_count = 0,
139
   .side_of_oriented_circle_total_count = 0, .side_of_oriented_circle_exact_count
   = 0  ;
142
143
144
    /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
145
    eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:253:195: error:
146
    C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@M-^Y or
    aM^-\ensuremath{^{\circ}}M^-\xspace X-std=gnu++2aaM^-\ensuremath{^{\circ}}M^-\xspace Y
148
    [-Wpedantic]
      253 l
             typename Kernel<Real>::Statistics Kernel<Real>::stat_ =
149
    {.orientation_total_count = 0, .orientation_exact_count = 0,
150
    .preferred_direction_total_count = 0, .preferred_direction_exact_count = 0,
    .side_of_oriented_circle_total_count = 0, .side_of_oriented_circle_exact_count
152
    = 0;
153
154
155
    /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
157
   eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:253:237: error:
    C++ designated initializers only available with aM-^0M-^X-std=c++2aaM-^0M-^Y or
159
    aM-^0M-^X-std=gnu++2aaM-^0M-^Y
    [-Wpedantic]
160
      253 | typename Kernel<Real>::Statistics Kernel<Real>::stat_ =
161
162
    {.orientation_total_count = 0, .orientation_exact_count = 0,
    .preferred_direction_total_count = 0, .preferred_direction_exact_count = 0,
163
    .side_of_oriented_circle_total_count = 0, .side_of_oriented_circle_exact_count
164
    = 0 };
165
166
167
168
169
    gmake[3]: *** [CMakeFiles/test_kernel.dir/app/test_kernel.cpp.o] Error 1
170
    gmake[3]: Leaving directory
172
    '/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
173
    deOnyia/cktmp/package-interval_sane/derived'
    gmake[2]: *** [CMakeFiles/test_kernel.dir/all] Error 2
174
    gmake[2]: Leaving directory
175
    `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
176
    deOnyia/cktmp/package-interval_sane/derived'
```

```
Log: interval_sane build test_kernel
Jun 15, 20 0:50
                                                                                         Page 4/4
178 gmake[1]: *** [CMakeFiles/test_kernel.dir/rule] Error 2
gmake[1]: Leaving directory '/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
deOnyia/cktmp/package-interval_sane/derived'
gmake: *** [test_kernel] Error 2
183 ERROR: build failed to generate executable test_kernel
```

```
../commit history
Jun 15, 20 0:50
                                                                              Page 1/2
    commit 2e5f4130237556ce16fe8499c8eb2cd218206a6c
   Author: JudeOnyia <60678029+JudeOnyia@users.noreply.github.com>
            Thu Jun 11 01:23:01 2020 -0700
        First Commit
   commit 0fd81b4d2225985328adcc597aaf191bb0cb2421
   Author: Jude Onyia <judeonyia10@gmail.com>
           Thu Jun 11 18:12:27 2020 -0700
10
        1) wrote indeterminate_result class
11
        2) wrote rounding_mode_saver class
12
        3) wrote statistics class
13
        4) wrote default constructor and compound operator overloads (+,-,*)
14
        5) wrote two argument constructor
15
        6) wrote upper, lower, is_singleton and sign member functions
16
17
   commit 4d3078e339c39e99c68a742c23c559fa9c688e26
18
   Author: Jude Onyia <judeonyia10@gmail.com>
19
           Fri Jun 12 02:43:44 2020 -0700
21
        1) Wrote clear_statistics member function
22
        2) Wrote get_statistics member function
        3) Wrote binary add, sub, mult non-member functions
25
        4) Wrote less than operator overload
        5) Wrote stream inserter
26
27
   commit e75ac306de5ff98af6a01dcae95bf65a26f5400a
28
   Author: JudeOnyia <60678029+JudeOnyia@users.noreply.github.com>
29
   Date:
           Fri Jun 12 15:34:29 2020 -0700
30
32
        Starting Kernel class
33
   commit 8d2796f420bc33bacd8221ec3cf952fec7cad7d0
34
   Author: Jude Onyia <judeonyia10@gmail.com>
           Fri Jun 12 20:58:07 2020 -0700
38
        1) Wrote the types needed for the Kernel class
39
        2) Wrote the Orientation and Oriented_side enum classes
40
        3) Wrote Statistics struct
        4) Wrote a template function to compute the determinant of a
41
           3 by 3 matrix
42
        5) Wrote the orientation member function that uses interval
43
           arithmetic or exact arithmetic if interval yields indetermine
44
           result
45
46
   commit 68ffe60f61a40f2c717223f08d1813940de3ed7c
    Author: Jude Onyia < judeonyia10@gmail.com>
48
            Sat Jun 13 01:09:36 2020 -0700
   Date:
49
50
        1) Wrote static function for solving determinant of 4 by 4 matrix
51
        2) Wrote side_of_oriented_circle member function
52
53
   commit 4b77c1fe22e0c876a07c3a18482872e696002f97
   Author: Jude Onyia <judeonyia10@gmail.com>
   Date:
           Sat Jun 13 18:55:44 2020 -0700
57
58
        1) Wrote preferred_direction member function
        2) Wrote is_strictly_convex_quad member function
59
60
```

commit 26168d5d9d4049b1fb342bc49a4ece6f58d15306

Author: Jude Onyia <judeonyia10@gmail.com>

61

```
../commit history
Jun 15, 20 0:50
                                                                              Page 2/2
            Sat Jun 13 23:11:03 2020 -0700
   Date:
        1) Wrote is_locally_delaunay_edge memeber function
65
        2) Wrote is_locally_pd_delaunay_edge member function
66
        3) Wrote clear_statistics member function
67
        4) Wrote get_statistics member function
   commit 5db9ce8ba4110ace1ff6126d5ceda2164c813dee
   Author: Jude Onyia <judeonyia10@gmail.com>
71
            Sun Jun 14 00:52:55 2020 -0700
72
   Date:
73
        Added some static specifiers and did some const correctness
74
75
   commit 87a35d587bab40497e037dcb34c70dbc2a67bb24
76
    Author: JudeOnyia <60678029+JudeOnyia@users.noreply.github.com>
            Sun Jun 14 14:57:03 2020 -0700
   Date:
79
        Starting delaunay triangulation
80
81
   commit f479bf137e763c2e18e916d5c3f47ece25eecdb5
   Author: Jude Onyia <judeonyia10@gmail.com>
           Sun Jun 14 22:20:15 2020 -0700
        1) Wrote function to check if half edge is flippable
        2) Modified is_strictly_convex function in kernel.hpp
87
88
   commit 3813050045c41cffeade0ff47c11d88fc6816a55
89
   Author: Jude Onyia <judeonyia10@gmail.com>
90
   Date:
            Mon Jun 15 00:31:57 2020 -0700
91
        1) applied the Lawson local optimization procedure
93
    commit 301c29dfd89ae92f8524e4fb633cefe4c7df11b1
   Author: JudeOnyia <60678029+JudeOnyia@users.noreply.github.com>
96
            Mon Jun 15 00:45:17 2020 -0700
   Date:
98
        1) Removed traingulation_2_demo
99
100
        2) Commented out cmakelists line of demo
```

Name: Jude Onyia

Student ID: V00947095

Course: ECE 596C

Due Date: June 19, 2020

Assignment 3: Non – Programming Exercise

6.1)

Due to exceptions that could be thrown for several reasons (i.e. division by zero, lack of memory, etc.), performing any necessary clean-up of an object must be put in a finalizer of that object. When exception is thrown, the program is violently removed from that code block, it does not execute any code further down the line from that exception. The only code guaranteed to execute is the code within the finalizer of that object. Therefore, any clean-up necessary must be done in the finalizer.

6.2 a)

During the stack unwinding process, objects are destroyed in the following sequence: die3, die, countdown, hello, i, bjarne, herb, dv, u, z.

6.2 b)

During the stack unwinding process, the only object destroyed is s.

6.3 a)

If there is insufficient memory to allocate for second buffer, an exception will be thrown, and the program will be violently ripped from the function without freeing the first buffer. To guarantee that both buffers are freed if an exception occurs, one can use a class for the buffers, where a char pointer is a data member. Overloads on operator= and operator[] can be used to assign the char pointer to a space in memory and access it, respectively. The finalizer of the class can be used to safely deallocate the memory if an exception is thrown. Both buf1 and buf2 will then be objects of this class. In the incident where buf1 is created and memory runs out while attempting to create buf2, as the exception is thrown, it is guaranteed that the finalizer of buf1 will be called to free up that space before leaving the function.

6.3 b)

If the formatting flags are changed and outputting the integer to the ostream causes an exception to be thrown, the program will be violently ripped from the function without running the code that restores the formatting flag. To prevent this, the line that changes the formatting flag and outputs the integer can be surrounded by a try clause. The body of the associated catch clause can restore the old formatting flags if an exception occurs.

6.3 c)

If the queue has a number of elements in it, and there is insufficient amount of memory left to push another element, q.push_back(value) might throws an exception. If it does and the user of the class does not catch it, the program could terminate without freeing the memory allocated for the elements already in the queue. To prevent this, a finalizer for the class can be explicitly defined to free

the queue when called. This will ensure that when an exception is thrown, the memory allocated for elements in the queue are freed before the object is destroyed and the program terminated.

6.5)

The function 'analyze' cannot throw an exception because the function has the noexcept specifier. This specifier indicates to the compiler that code for exceptions is not needed for this function, therefore, the compile will not generate code for exceptions for this function. This means the function is incapable of throwing an exception. If an attempt is made to throw an exception in this noexcept function, this will result in a fatal error.

The function 'doWork' does not have a noexcept specifier, therefore, it may or may not throw an exception. The compiler will generate code for exceptions for this function. Therefore, this function is capable of throwing an exception.

```
Jun 15, 20 0:50 CMakeLists.txt Page 1/1
```

```
# Specify Minimum Required Version
   cmake_minimum_required(VERSION 3.1 FATAL_ERROR)
   # Specify Project and Language
  project(cpp_arithmetic LANGUAGES CXX)
   # Set CXX Flags with the -frounding-math option
   set (CMAKE_CXX_FLAGS "-frounding-math")
10 # Find CGAL Library
find_package(CGAL REQUIRED)
12
# Set Include directories and libraries
  include_directories(include ${CGAL_INCLUDE_DIRS})
14
  link_libraries(${CGAL_LIBRARY} ${GMP_LIBRARIES})
15
17 # Add Executable Program
add_executable(test_interval app/test_interval.cpp)
  add_executable(test_kernel app/test_kernel.cpp)
20 #add_executable(triangulation_2_demo app/triangulation_2_demo.cpp app/triangulat
   ion_2.hpp)
21 add_executable(delaunay_triangulation app/delaunay_triangulation.cpp app/triangu
   lation_2.hpp)
```

```
Jun 15, 20 0:50
```

include/ra/interval.hpp

Page 1/4

```
#ifndef INTERVAL_HPP
   #define INTERVAL HPP
   #include < cfenv >
   #include<stdexcept>
   #include<algorithm>
   #include<iostream>
   namespace ra::math {
       struct indeterminate_result : public std::runtime_error{
9
10
            using std::runtime_error::runtime_error;
11
       };
12
       class rounding_mode_saver {
13
           public:
14
                rounding_mode_saver() : current_round_mode(std::fegetround()) {}
                ~rounding_mode_saver() {
16
17
                    std::fesetround(current_round_mode);
18
                rounding_mode_saver(rounding_mode_saver&&) = delete;
19
                rounding_mode_saver(const rounding_mode_saver&) = delete;
20
21
                rounding_mode_saver& operator=(rounding_mode_saver&&) = delete;
                rounding_mode_saver& operator=(const rounding_mode_saver&) = delete;
22
           private:
                int current_round_mode;
24
25
       };
26
       template<class T>
27
       class interval{
28
           public:
29
                using real_type = T;
30
31
32
                struct statistics{
                    // The total number of indeterminate results encountered.
33
                    unsigned long indeterminate_result_count;
34
                    // The total number of interval arithmetic operations.
35
                    unsigned long arithmetic_op_count;
36
37
                };
38
39
                interval(real_type set_value = real_type(0)) : lower_(set_value), up
   per_(set_value) {}
                interval(real_type lower, real_type upper) : lower_((lower<upper)? 1</pre>
40
   ower:upper), upper_((upper>lower)? upper:lower) {}
                real_type lower() const { return lower_; }
41
                real_type upper() const { return upper_; }
42
43
                static void incr_indeterminate_result_count() { ++(stat_.indetermina
44
   te_result_count); }
                static void incr_arithmetic_op_count() { ++(stat_.arithmetic_op_coun
45
   t); }
46
                interval& operator+=(const interval& obj) {
47
                    real_type upper_Temp(0);
48
49
                    real_type lower_Temp(0);
                    rounding_mode_saver rms; // Save the rounding mode that should b
50
   e restored
                    std::fesetround(FE_DOWNWARD);
51
52
                    lower_Temp = lower_ + obj.lower();
53
                    std::fesetround(FE_UPWARD);
                    upper_Temp = upper_ + obj.upper();
54
                    lower_ = lower_Temp;
55
                    upper_ = upper_Temp;
56
57
                    incr_arithmetic_op_count();
```

```
include/ra/interval.hpp
Jun 15, 20 0:50
                                                                                  Page 2/4
                     return *this;
59
                 interval& operator = (const interval& obj) {
60
                     real_type upper_Temp(0);
61
                     real_type lower_Temp(0);
62
                     rounding_mode_saver rms; // Save the rounding mode that should b
    e restored
                     std::fesetround(FE_DOWNWARD);
64
                     lower_Temp = lower_ - obj.upper();
65
                     std::fesetround(FE_UPWARD);
66
                     upper_Temp = upper_ - obj.lower();
67
                     lower_ = lower_Temp;
68
                     upper_ = upper_Temp;
69
                     incr_arithmetic_op_count();
70
                     return *this;
71
72
73
                 interval& operator*=(const interval& obj) {
                     real_type upper_Temp(0);
74
                     real_type lower_Temp(0);
75
                     rounding_mode_saver rms; // Save the rounding mode that should b
76
    e restored
                     std::fesetround(FE_DOWNWARD);
77
                     lower_Temp = std::min(std::min((lower_*obj.lower()),(lower_*obj.
    upper())), std::min((upper_*obj.lower()), (upper_*obj.upper())));
79
                     std::fesetround(FE_UPWARD);
                     upper_Temp = std::max(std::max((lower_*obj.lower()), (lower_*obj.
80
    upper())), std::max((upper_*obj.lower()), (upper_*obj.upper())));
                     lower_ = lower_Temp;
81
                     upper_ = upper_Temp;
82
                     incr_arithmetic_op_count();
83
                     return *this;
84
85
                 bool is_singleton() const { return (lower_==upper_)? true : false; }
87
88
                 int sign() const {
89
                     if((lower_ < real_type(0)) && (upper_ < real_type(0))) { return -</pre>
90
    1; }
                     else if((lower_ > real_type(0)) && (upper_ > real_type(0))){ ret
    urn 1; }
                     else if((lower_ == real_type(0)) && (upper_ == real_type(0))){ \mathbf{r}
92
    eturn 0; }
                     else{
93
                          incr_indeterminate_result_count();
94
                          throw indeterminate_result("Indeterminate sign of interval");
95
96
                     }
                 static void clear_statistics() {
                     stat_.indeterminate_result_count = 0;
100
                     stat_.arithmetic_op_count = 0;
101
102
103
                 static void get_statistics(statistics& stat) {
104
                     stat = stat_;
105
106
107
108
            private:
109
                 real_type upper_;
                 real_type lower_;
110
                 static statistics stat ;
111
112
```

```
include/ra/interval.hpp
Jun 15, 20 0:50
                                                                                 Page 3/4
        };
113
114
        template < class real_type >
115
        typename interval<real_type>::statistics interval<real_type>::stat_ = { .inde
116
    terminate_result_count = 0, .arithmetic_op_count = 0);
117
        template < class real_type>
118
        interval<real_type> operator+(const interval<real_type>& obj_A, const interv
    al<real_type>& obj_B) {
120
            real_type upper_Temp(0);
            real_type lower_Temp(0);
121
            rounding_mode_saver rms; // Save the rounding mode that should be restor
122
    ed
            std::fesetround(FE_DOWNWARD);
123
            lower_Temp = obj_A.lower() + obj_B.lower();
124
            std::fesetround(FE_UPWARD);
125
            upper_Temp = obj_A.upper() + obj_B.upper();
126
            interval<real_type> result(lower_Temp,upper_Temp);
127
            result.incr_arithmetic_op_count();
128
            return result;
129
        }
130
131
        template < class real_type>
132
        interval<real_type> operator-(const interval<real_type>& obj_A, const interv
    al<real_type>& obj_B){
            real_type upper_Temp(0);
134
            real_type lower_Temp(0);
135
            rounding_mode_saver rms; // Save the rounding mode that should be restor
136
    ed
            std::fesetround(FE_DOWNWARD);
137
            lower_Temp = obj_A.lower() - obj_B.upper();
138
139
            std::fesetround(FE_UPWARD);
            upper_Temp = obj_A.upper() - obj_B.lower();
140
            interval<real_type> result(lower_Temp,upper_Temp);
141
            result.incr_arithmetic_op_count();
142
            return result;
143
        }
144
145
146
        template < class real_type >
        interval<real_type> operator*(const interval<real_type>& obj_A, const interv
147
    al<real_type>& obj_B){
            real_type upper_Temp(0);
148
            real_type lower_Temp(0);
149
            rounding_mode_saver rms; // Save the rounding mode that should be restor
150
    ed
            std::fesetround(FE_DOWNWARD);
151
152
            lower_Temp = std::min(std::min((obj_A.lower()*obj_B.lower()), (obj_A.lowe
    r()*obj_B.upper())), std::min((obj_A.upper()*obj_B.lower()), (obj_A.upper()*obj_
    B.upper()));
            std::fesetround(FE_UPWARD);
153
            upper_Temp = std::max(std::max((obj_A.lower()*obj_B.lower()),(obj_A.lowe
154
    r()*obj_B.upper())), std::max((obj_A.upper()*obj_B.lower()), (obj_A.upper()*obj_
    B.upper()));
155
            interval<real_type> result(lower_Temp,upper_Temp);
            result.incr_arithmetic_op_count();
156
            return result;
157
        }
158
159
        template<class real_type>
160
        bool operator<(const interval<real_type>& obj_A, const interval<real_type>&
161
    obj_B) {
            if( (obj_A.upper()) < (obj_B.lower()) ) { return true; }</pre>
162
```

```
include/ra/interval.hpp
Jun 15, 20 0:50
                                                                                   Page 4/4
             else if( (obj_A.lower()) >= (obj_B.upper()) ) { return false; }
163
164
             else{
                 interval<real_type>::incr_indeterminate_result_count();
165
                 throw indeterminate_result("Indeterminate less than operator");
166
             }
167
        }
168
169
        template<class real_type>
170
        std::ostream& operator<<(std::ostream& outStream, const interval<real_type>&
171
     obj){
             outStream<<"["<<(obj.lower())<<","<<(obj.upper())<<"]";
172
             return outStream;
173
174
175
176
177
178 #endif
```

```
#include"ra/interval.hpp"
   #include<iostream>
   int main(){
        using itvf = typename ra::math::interval<float>;
        using stat = typename ra::math::interval<float>::statistics;
        using std::cout;
        using std::endl;
9
10
        // Test default constructor
        itvf obj_A;
11
        cout<<"obj_A: "<<(obj_A.lower())<<" "<<(obj_A.upper())<<endl;
12
        // Test 2 value constructor
13
        itvf obj_B(2.89, 2.91);
14
        itvf obj_C(3.91, 3.89);
15
        cout << "obj_B: " << (obj_B.lower()) << " " << (obj_B.upper()) << endl;
16
        cout << "obj_C: "<< (obj_C.lower()) << " "<< (obj_C.upper()) << endl;
17
18
        // Test Compound add, sub, mult
        cout << "obj_C+=obj_B: "<< ((obj_C+=obj_B).lower()) << " "<< (obj_C.upper()) << endl;
19
        cout < "obj_C-=obj_B: "<< ((obj_C-=obj_B).lower()) << " "<< (obj_C.upper()) << endl;
20
        cout << "obj_C*=obj_B: "<< ((obj_C*=obj_B).lower()) << " "<< (obj_C.upper()) << endl;
21
        // Test Aliasing of Compound add, sub, mult
22
        cout << "obj_B+=obj_B: "<< ((obj_B+=obj_B).lower()) << " "<< (obj_B.upper()) << endl;
23
        cout < "obj_B-=obj_B: " << ((obj_B-=obj_B).lower()) << " " << (obj_B.upper()) << endl;
24
        cout < "obj_B*=obj_B: "<< ((obj_B*=obj_B).lower()) << " "<< (obj_B.upper()) << endl;
25
        // Test is_singleton
26
        itvf obj_D(1.7);
27
        cout<<"obj_D.is_singleton() must be true: "<<(obj_D.is_singleton())<<endl;</pre>
28
        cout << "obj_C.is_singleton() must be false: " << (obj_C.is_singleton()) << endl;</pre>
29
30
        // Test sign()
        itvf obj_E(-7.335, -7.339);
31
        itvf obj_F(3.442,3.448);
32
        itvf obj_G(-0.04, 0.12);
33
        cout<<"Neg sign: "<< (obj_E.sign()) <<endl;</pre>
34
        cout<<"Pos sign: "<<(obj_F.sign())<<endl;</pre>
35
        cout << "Zero no sign: " << (obj_A.sign()) << endl;</pre>
36
        //cout<<"Indeteminate for sign: "<<(obj_G.sign())<<endl;</pre>
37
        // Test get_statistics and clear_statistics
38
39
        stat st;
40
        obj_C.get_statistics(st);
        cout << "get_statistics of obj_C: indet="<< (st.indeterminate_result_count) << " arith=" << (st.</pre>
41
   arithmetic_op_count) <<endl;
        obj_C.clear_statistics();
42
        obj_C.get_statistics(st);
43
        cout << "clear_statistics of obj_C: indet="<< (st.indeterminate_result_count) << " arith="<< (st</pre>
44
    .arithmetic_op_count) <<endl;</pre>
45
        // Test Binary add, sub, and mult
        itvf obj_H = obj_E + obj_F;
46
        itvf obj_I = obj_E - obj_F;
47
        itvf obj_J = obj_E * obj_F;
48
        cout << "[-7.339, -7.335] + [3.442, 3.448]: "< obj_H << endl;
49
        cout << "[-7.339,-7.335] - [3.442,3.448]: "<<obj_I<<endl;
50
        cout << "[-7.339,-7.335] * [3.442,3.448]: "<<obj_J<<endl;
51
52
        // Test less than
        itvf obj_L(-7.342, -7.337);
53
        cout << "Must be true: " << (obj_E < obj_F) << endl;</pre>
        cout<<"Must be false: "<<(obj_A < obj_E) <<endl;</pre>
55
        //cout<<"Indeterminate: "<<(obj_L < obj_E)<<endl;</pre>
56
57
58
59
60
```

Jun 15, 20 0:50	app/test_interval.cpp	Page 2/2
61 return 0; 62 }		-

```
#ifndef KERNEL_HPP
   #define KERNEL_HPP
   #include <CGAL/Cartesian.h>
   #include <CGAL/MP Float.h>
   #include <cstddef>
   #include"ra/interval.hpp"
   namespace ra::geometry {
       // A geometry kernel with robust predicates.
9
       template<class R>
10
       class Kernel {
11
           public:
12
                // The type used to represent real numbers.
13
                using Real = R;
14
15
                // The type used to represent points in two dimensions.
16
                using Point =typename CGAL::Cartesian<R>::Point_2;
17
18
                // The type used to represent vectors in two dimensions.
19
                using Vector = typename CGAL::Cartesian<R>::Vector_2;
20
21
                // Type used for interval class
22
                using itv = typename ra::math::interval<R>;
                // Type used for indeterminate_result class
25
                using idr = typename ra::math::indeterminate_result;
                // Type used for exact arithmetic
26
                using exct = typename CGAL::MP_Float;
27
28
                // The possible outcomes of an orientation test.
29
                enum class Orientation : int {
30
                    right\_turn = -1,
31
32
                    collinear = 0,
                    left_turn = 1,
33
34
35
                // The possible outcomes of an oriented-side-of test.
36
                enum class Oriented_side : int {
37
                    on_negative_side = -1,
38
39
                    on_boundary = 0,
40
                    on_positive_side = 1,
                };
41
42
                // The set of statistics maintained by the kernel.
43
                struct Statistics {
44
                    // The total number of orientation tests.
45
                    std::size_t orientation_total_count;
46
                    // The number of orientation tests requiring exact arithmetic
                    std::size_t orientation_exact_count;
48
                    // The total number of preferred-direction tests.
49
                    std::size_t preferred_direction_total_count;
50
                    // The number of preferred-direction tests requiring exact arith
51
   metic
                    std::size_t preferred_direction_exact_count;
52
53
                    // The total number of side-of-oriented-circle tests.
                    std::size_t side_of_oriented_circle_total_count;
54
                    // The number of side-of-oriented-circle tests requiring exact a
   rithmetic
56
                    std::size_t side_of_oriented_circle_exact_count;
                };
57
58
                // Since a kernel object is stateless, construction and destruction
59
   are trivial
```

```
include/ra/kernel.hpp
Jun 15, 20 0:50
                                                                              Page 2/5
                Kernel() = default;
                ~Kernel() = default;
61
62
                // The kernel type is both movable and copyable.
63
                // Since a kernel object is stateless, a copy/move operation is triv
64
    ial
                Kernel(const Kernel&) = default;
65
                Kernel& operator=(const Kernel&) = default;
66
                Kernel (Kernel&&) = default;
67
                Kernel& operator=(Kernel&&) = default;
68
69
                // Member function to compute 3 by 3 determinant
70
                template<class DR>
71
                static DR determinant_3_by_3(const DR (&m)[3][3]) {
72
                    73
    m[0][2]*m[1][0]*m[2][1])) - ( (m[2][0]*m[1][1]*m[0][2])+(m[2][1]*m[1][2]*m[0][0]
    ])+(m[2][2]*m[1][0]*m[0][1]));
                    return det;
74
75
76
                // Member function to compute 4 by 4 determinant
77
                template<class DDR>
78
                static DDR determinant_4_by_4(const DDR (&m)[4][4]) {
79
                    DDR sub0[3][3] = {m[1][1], m[1][2], m[1][3], m[2][1], m[2][2], m[2][3]
    ],m[3][1],m[3][2],m[3][3];
                    DDR sub1[3][3] = {m[0][1], m[0][2], m[0][3], m[2][1], m[2][2], m[2][3]
81
    ],m[3][1],m[3][2],m[3][3];
                    DDR sub2[3][3] = {m[0][1],m[0][2],m[0][3],m[1][1],m[1][2],m[1][3}
82
    ],m[3][1],m[3][2],m[3][3]};
                    DDR sub3[3][3] = {m[0][1], m[0][2], m[0][3], m[1][1], m[1][2], m[1][3]
    ],m[2][1],m[2][2],m[2][3]};
                    DDR det = (m[0][0]*determinant_3_by_3(sub0)) - (m[1][0]*determin
    ant_3_by_3(sub1)) + (m[2][0]*determinant_3_by_3(sub2)) - (m[3][0]*determinant_3_
   by_3(sub3));
                    return det;
85
86
87
                // Determines how the point c is positioned relative to the
88
89
                // directed line through the points a and b (in that order).
90
                // Precondition: The points a and b have distinct values.
                Orientation orientation (const Point & a, const Point & b, const Point &
91
     c) const {
                    try{
92
                        ++(stat_.orientation_total_count);
93
                        itv ax(a.x()); itv ay(a.y()); itv bx(b.x()); itv by(b.y());
94
    itv cx(c.x()); itv cy(c.y());
95
                        itv matrix[3][3] = \{ax, bx, cx, ay, by, cy, itv(1.0), itv(1.0), itv(
    1.0)};
                        itv det = determinant_3_by_3 (matrix);
96
                        if( (det.sign()) == -1 ) { return (Orientation::right_turn);
97
     }
                        else if( (det.sign()) == 1 ) { return (Orientation::left_tur
98
    n); }
99
                        else{ return (Orientation::collinear); }
100
                    catch(const idr& e) {
101
                        ++(stat_.orientation_exact_count);
102
103
                        exct matrix[3][3] = \{(a.x()), (b.x()), (c.x()), (a.y()), (b.y())\}
    , (c.y()), exct(1.0), exct(1.0), exct(1.0) \};
                        exct det = determinant_3_by_3(matrix);
104
                        if(det < exct(0)) { return (Orientation::right_turn); }</pre>
105
                        else if(det > exct(0)) { return (Orientation::left_turn); }
106
```

```
include/ra/kernel.hpp
Jun 15, 20 0:50
                                                                                 Page 3/5
                         else{ return (Orientation::collinear); }
107
                     }
108
                 }
109
110
                 // Determines how the point d is positioned relative to the
111
112
                 // oriented circle passing through the points a, b, and c
                 // (in that order).
113
                 // Precondition: The points a, b, and c are not collinear.
114
                 Oriented_side side_of_oriented_circle(const Point& a, const Point& b
115
      const Point& c, const Point& d) const{
116
                     try{
                         ++(stat_.side_of_oriented_circle_total_count);
117
                         itv ax(a.x()); itv ay(a.y()); itv bx(b.x()); itv by(b.y());
118
    itv cx(c.x()); itv cy(c.y());
                         itv dx(d.x()); itv dy(d.y());
119
                         itv ar((ax*ax)+(ay*ay)); itv br((bx*bx)+(by*by)); itv cr((cx)
120
    *cx) + (cy*cy));
121
                         itv dr((dx*dx)+(dy*dy));
                         itv matrix[4][4] = {ax,bx,cx,dx,ay,by,cy,dy,ar,br,cr,dr,itv(
122
    1.0), itv(1.0), itv(1.0), itv(1.0)};
123
                         itv det = determinant_4_by_4 (matrix);
                         if( (det.sign()) == -1 ) { return (Oriented_side::on_negativ
124
    e_side); }
                         else if( (det.sign()) == 1 ) { return (Oriented_side::on_pos
125
    itive_side); }
                         else{ return (Oriented_side::on_boundary); }
126
127
                     catch(const idr& e) {
128
                         ++(stat_.side_of_oriented_circle_exact_count);
129
                         exct ax(a.x()); exct ay(a.y()); exct bx(b.x()); exct by(b.y(
130
    ));
131
                         exct cx(c.x()); exct cy(c.y());
                         exct dx(d.x()); exct dy(d.y());
132
                         exct ar((ax*ax)+(ay*ay)); exct br((bx*bx)+(by*by)); exct cr(
133
    (cx*cx)+(cy*cy));
                         exct dr((dx*dx)+(dy*dy));
134
                         exct matrix[4][4] = \{ax, bx, cx, dx, ay, by, cy, dy, ar, br, cr, dr, exc
135
    t(1.0), exct(1.0), exct(1.0), exct(1.0)};
136
                         exct det = determinant_4_by_4 (matrix);
                         if(det < exct(0)) { return (Oriented_side::on_negative_side);</pre>
137
     }
                         else if(det > exct(0)){ return (Oriented_side::on_positive_s
138
    ide); }
                         else{ return (Oriented_side::on_boundary); }
139
                     }
140
141
142
                 // Determines if, compared to the orientation of line
143
                 // segment cd, the orientation of the line segment ab is
144
                 // more close, equally close, or less close to the
145
                 // orientation of the vector v.
146
                 // The value returned is 1, 0, or -1 if, compared to the
147
                 // orientation of cd, the orientation of ab is more close,
148
                 // equally close, or less close to the orientation of v,
149
                 // respectively.
150
                 // Precondition: The points a and b have distinct values; the
151
                 // points c and d have distinct values; the vector v is not
152
153
                 // the zero vector.
                 int preferred_direction(const Point& a,const Point& b,const Point& c
154
    , const Point& d, const Vector& v) const{
                     try{
155
                         ++(stat_.preferred_direction_total_count);
156
```

```
include/ra/kernel.hpp
Jun 15, 20 0:50
                                                                               Page 4/5
                         itv ax(a.x()); itv ay(a.y()); itv bx(b.x()); itv by(b.y());
157
                         itv cx(c.x()); itv cy(c.y()); itv dx(d.x()); itv dy(d.y());
158
                         itv vx(v.x()); itv vy(v.y());
159
                         itv first = ((dx-cx)*(dx-cx)) + ((dy-cy)*(dy-cy));
160
                         itv second_inc = ((bx-ax)*(vx)) + ((by-ay)*(vy));
161
162
                         itv second = second_inc * second_inc;
                         itv third = ((bx-ax)*(bx-ax)) + ((by-ay)*(by-ay));
163
                         itv fourth_inc = ((dx-cx)*(vx)) + ((dy-cy)*(vy));
164
                         itv fourth = fourth_inc * fourth_inc;
165
                         itv result = (first * second) - (third * fourth);
166
                         return (result.sign());
167
168
                     catch(const idr& e) {
169
                         ++(stat_.preferred_direction_exact_count);
170
                         exct ax(a.x()); exct ay(a.y()); exct bx(b.x()); exct by(b.y(
171
    ));
                         exct cx(c.x()); exct cy(c.y()); exct dx(d.x()); exct dy(d.y(
172
    ));
                         exct vx(v.x()); exct vy(v.y());
173
                         exct first = ((dx-cx)*(dx-cx)) + ((dy-cy)*(dy-cy));
174
                         exct second_inc = ((bx-ax)*(vx)) + ((by-ay)*(vy));
175
                         exct second = second_inc * second_inc;
176
                         exct third = ((bx-ax)*(bx-ax)) + ((by-ay)*(by-ay));
177
                         exct fourth_inc = ((dx-cx)*(vx)) + ((dy-cy)*(vy));
178
179
                         exct fourth = fourth_inc * fourth_inc;
                         exct result = (first * second) - (third * fourth);
180
                         if( result < exct(0) ) { return -1; }</pre>
181
                         else if( result > exct(0) ) { return 1; }
182
                         else { return 0; }
183
                     }
184
                }
185
186
                // Tests if the quadrilateral with vertices a, b, c, and d
187
                // specified in CCW order is strictly convex.
188
                // Precondition: The vertices a, b, c, and d have distinct
189
                // values and are specified in CCW order.
190
                bool is_strictly_convex_quad(const Point& a, const Point& b, const Poi
191
    nt& c, const Point& d) const {
192
                     Orientation t_1 = orientation(a,b,c);
193
                     Orientation t_2 = orientation(b, c, d);
                     Orientation t_3 = orientation(c,d,a);
194
                     Orientation t_4 = orientation(d,a,b);
195
                     //Orientation left = Orientation::left_turn;
196
                     //if((t_1==left) \&\& (t_2==left) \&\& (t_3==left) \&\& (t_4==left))
197
    { return true; }
                     if( (t_2==t_1) && (t_3==t_1) && (t_4==t_1) ){ return true; }
198
199
                     else{ return false; }
200
201
                // Tests if the flippable edge, with endpoints a and c and
202
                // two incident faces abc and acd, is locally Delaunay.
203
                // Precondition: The points a, b, c, and d have distinct
204
                // values; the quadrilateral abcd must be strictly convex.
205
                bool is_locally_delaunay_edge(const Point& a,const Point& b,const Po
206
    int& c,const Point& d)const{
                     Oriented_side tst = side_of_oriented_circle(a,b,c,d);
207
                     if(tst == Oriented_side::on_positive_side) { return false; }
208
209
                     else{ return true; }
                }
210
211
                // Tests if the flippable edge, with endpoints a and c and
212
                // two incident faces abc and acd, has the preferred-directions
213
```

```
include/ra/kernel.hpp
Jun 15, 20 0:50
                                                                                 Page 5/5
                 // locally-Delaunay property with respect to the first and
214
                 // second directions u and v.
215
                 // Precondition: The points a, b, c, and d have distinct values;
216
                 // the vectors u and v are not zero vectors; the vectors u and
217
                 // v are neither parallel nor orthogonal.
218
219
                 bool is_locally_pd_delaunay_edge(const Point& a,const Point& b,const
     Point& c, const Point& d, const Vector& u, const Vector& v) const {
                     Oriented_side tst = side_of_oriented_circle(a,b,c,d);
220
                     if(tst == Oriented_side::on_positive_side) { return false; }
221
                     else if(tst == Oriented_side::on_negative_side) { return true; }
222
                     else{
223
                         if((preferred_direction(a,c,b,d,u)) > 0) { return true; }
224
225
                         else if( ((preferred_direction(a,c,b,d,u))==0) && ((preferre
    d_{direction(a,c,b,d,v))>0) ){
                              return true;
226
227
228
                         else{ return false;}
                     }
229
                 }
230
231
                 // Clear (i.e., set to zero) all kernel statistics.
232
                 static void clear_statistics() {
233
                     stat_.orientation_total_count=0;
234
                     stat_.orientation_exact_count=0;
235
236
                     stat_.preferred_direction_total_count=0;
                     stat_.preferred_direction_exact_count=0;
237
                     stat_.side_of_oriented_circle_total_count=0;
238
                     stat_.side_of_oriented_circle_exact_count=0;
239
                 }
240
241
                 // Get the current values of the kernel statistics.
242
                 static void get_statistics(Statistics& statistics) {
243
                     statistics = stat_;
244
245
246
            private:
247
                 static Statistics stat_;
248
249
250
        };
251
        template<class Real>
252
        typename Kernel<Real>::Statistics Kernel<Real>::stat_ = {.orientation_total_
    count = 0, .orientation_exact_count = 0, .preferred_direction_total_count = 0, .
    preferred_direction_exact_count = 0, .side_of_oriented_circle_total_count = 0, .
    side_of_oriented_circle_exact_count = 0};
254
255
256
257
   #endif
258
```

```
#include <CGAL/Cartesian.h>
   #include"ra/kernel.hpp"
   #include<iostream>
3
   int main(){
        using std::cout;
        using std::endl;
        using knlD = typename ra::geometry::Kernel<double>;
        using ort = typename ra::geometry::Kernel<double>::Orientation;
9
10
        using ort_side = typename ra::geometry::Kernel<double>::Oriented_side;
        using stat = typename ra::geometry::Kernel<double>::Statistics;
11
        using point = typename CGAL::Cartesian<double>::Point_2;
12
        using vector = typename CGAL::Cartesian<double>::Vector_2;
13
14
        knlD obj_A;
15
        point a(0,0); point b(2,2); point c(2,0); point d(1,1); point e(0,2);
16
        // Test orientation
17
        cout << "Test for right(true): "<< ((obj_A.orientation(a,b,c)) == (ort::right_turn)) <<e</pre>
18
   ndl;
        cout << "Test for collinear(true): "<< ((obj_A.orientation(a,b,d)) == (ort::collinear)) <</pre>
19
   <endl;
        cout<<"Test for left(true): "<<((obj_A.orientation(a,b,e)) == (ort::left_turn))<<end</pre>
20
   1;
        // Test side_of_oriented_circle
21
22
        point ac(0,0); point bc(2,0); point cc(0,2); point dc(1,1); point ec(2,2); p
   oint fc(3,3);
        cout << "Test for on_negative_side(true): "<< ((obj_A.side_of_oriented_circle(ac,bc,cc,fc))</pre>
23
   ) == (ort_side::on_negative_side)) <<endl;</pre>
        cout<<"Test for on_boundary(true): "<<((obj_A.side_of_oriented_circle(ac,bc,cc,ec))</pre>
24
       (ort_side::on_boundary)) <<endl;</pre>
        cout << "Test for on_positive_side(true): "<< ((obj_A.side_of_oriented_circle(ac,bc,cc,dc))</pre>
25
   ) == (ort_side::on_positive_side)) << endl;</pre>
        // Test preferred_direction
26
        point ap(0,0); point bp(2,2); point cp(2,0); point dp(0,2);
27
        vector up(2,1); vector vp(1,0); vector wp(-1,2);
28
        cout<<"Test for more close(true): "<<((obj_A.preferred_direction(ap,bp,cp,dp,up)) ==</pre>
   1) <<endl;
        cout << "Test for equally close(true): "<< ((obj_A.preferred_direction(ap,bp,cp,dp,vp)) ==</pre>
30
        cout<<"Test for less close(true): "<<((obj_A.preferred_direction(ap,bp,cp,dp,wp)) == -</pre>
   1) <<endl;
        // Test is_strictly_convex_quad
32
        point asc(0,0); point bsc(1,0); point csc(2,2); point dsc(1,2);
33
        point anc(0,0); point bnc(1,0); point cnc(2,0); point dnc(2,1);
34
        cout<<"Test for convexity(true): "<<(obj_A.is_strictly_convex_quad(asc,bsc,csc,dsc))</pre>
35
   <endl;
36
        cout << "Test for convexity(false): "<< (obj_A.is_strictly_convex_quad (anc, bnc, cnc, dnc))</pre>
   <<endl;
        //Test is_locally_delaunay_edge
37
        point ad(0,0); point bd(1,0); point cd(1,1); point dd(0,1);
38
        point adn(0,0); point bdn(5,0); point cdn(5,5); point ddn(2,3);
39
        cout << "Test edge for locally delaunay(true): "<< (obj_A.is_locally_delaunay_edge (ad,bd,cd,</pre>
40
   dd)) < < endl;
        cout < "Test edge for locally delaunay(false): "<< (obj_A.is_locally_delaunay_edge (adn, bdn,
41
   cdn,ddn))<<endl;
        // Test is_locally_pd_delaunay_edge
42
        cout << "Test edge for pref dir loc delaunay(true): "<< (obj_A.is_locally_pd_delaunay_edge (ad,
43
   bd, cd, dd, up, vp)) <<endl;
        cout<<"Test edge for pref dir loc delaunay(false): "<<(obj_A.is_locally_pd_delaunay_edge(ad</pre>
44
   ,bd,cd,dd,wp,vp))<<endl;
        // Test get_statistics and clear_statistics
45
46
        stat st;
```

```
Jun 15, 20 0:50
```

app/test_kernel.cpp

Page 2/2

```
obj_A.get_statistics(st);
             cout << "Current statistics: ort tot="<< (st.orientation_total_count) << " ort ex="<< (st.orien</pre>
48
      tation_exact_count) << " pd tot=" << (st.preferred_direction_total_count) << " pd ex=" << (s
      t.preferred_direction_exact_count) << "crc tot=" << (st.side_of_oriented_circle_total_
      count) << " crc ex="<<(st.side_of_oriented_circle_exact_count) <<endl;</pre>
              obj_A.clear_statistics();
              obj_A.get_statistics(st);
50
              cout<<"Cleared statistics: ort tot="<< (st.orientation_total_count) <<" ort ex="<< (st.orien</pre>
      tation_exact_count) << " pd tot=" << (st.preferred_direction_total_count) << " pd ex=" << (st.preferred_direction_total_count) </pre>
      t.preferred_direction_exact_count) << " crc tot=" << (st.side_of_oriented_circle_total_
      count) << " crc ex="<< (st.side_of_oriented_circle_exact_count) << endl;</pre>
52
53
54
55
      }
56
```

app/delaunay_triangulation.cpp

```
#include<iostream>
   #include"ra/kernel.hpp"
   #include <CGAL/Cartesian.h>
   #include "triangulation_2.hpp"
   #include<vector>
   using Kernel = CGAL::Cartesian<double>;
   using kernel = ra::geometry::Kernel<double>;
   using Triangulation = trilib::Triangulation_2<Kernel>;
10
   using hEit = Triangulation::Halfedge_iterator;
   using hE_hand = Triangulation::Halfedge_handle;
11
   using point = CGAL::Cartesian<double>::Point_2;
12
   using vector = CGAL::Cartesian<double>::Vector_2;
13
14
   using half_edge = Triangulation::Halfedge_handle;
17
   using vertex = Triangulation::Vertex_handle;
18
19
   bool is_flippable(hE_hand edg_iter, kernel& obj){
20
21
       //if((*edg_iter).is_border_edge()) { return false; }
       //else{
22
            point a = edg_iter->vertex()->point();
23
            point b = edg_iter->next()->vertex()->point();
24
25
            point c = edg_iter->opposite()->vertex()->point();
            point d = edg_iter->opposite()->next()->vertex()->point();
26
            return (obj.is_strictly_convex_quad(a,b,c,d));
27
       //}
28
29
   }
30
   int main(int argc, char** argv) {
31
32
       using std::cout;
       using std::endl;
33
       Triangulation tri(std::cin);
34
       kernel obj;
35
       hE_hand tmp_hand;
36
37
38
39
       std::vector<hE_hand> suspect_list;
       for (auto halfedgeIter = tri.halfedges_begin(); halfedgeIter !=tri.halfedges
40
   _end(); ++++halfedgeIter) {
           tmp_hand = &*halfedgeIter;
41
            if (is_flippable(tmp_hand,obj)) {
42
                suspect_list.push_back(tmp_hand);
43
            }
44
45
       }
46
       while(!(suspect_list.empty())){
            hE_hand it = suspect_list.back();
48
            if (is_flippable(it,obj)) {
49
                point c = it->vertex()->point();
50
                point d = it->next()->vertex()->point();
51
                point a = it->opposite()->vertex()->point();
52
53
                point b = it->opposite()->next()->vertex()->point();
                if(!(obj.is_locally_pd_delaunay_edge(a,b,c,d,vector(1,0),vector(1,1)
54
   ))){
                    it = tri.flip_edge(it);
55
56
                    suspect_list.push_back(it->next());
                    suspect_list.push_back(it->next()->next());
57
                    suspect_list.push_back(it->opposite()->next());
58
                    suspect_list.push_back(it->opposite()->next()->next());
59
60
                }
```

```
app/delaunay triangulation.cpp
Jun 15, 20 0:50
                                                                                    Page 2/3
61
             suspect_list.pop_back();
62
63
        }
64
        cout << "Resulting Triangulation" << endl;
65
66
        tri.output_off(cout);
67
         /*
68
        int count = 0;
69
        for (auto halfedgeIter = tri.halfedges_begin(); halfedgeIter !=tri.halfedges
70
    _end(); ++++halfedgeIter) {
             if(is_flippable(halfedgeIter,obj)) { ++count; }
71
             cout<<halfedgeIter->vertex()->point()<<endl;</pre>
72
73
        cout<< "Count was: "<<count<<endl;</pre>
74
75
76
77
78
79
        vertex other_quad_pt_a; vertex other_quad_pt_b;
80
        int num_of_edge_faces = 0;
81
         // Iterate over every edge
        for (auto halfedgeIter = tri.halfedges_begin(); halfedgeIter !=tri.halfedges
    _end(); ++++halfedgeIter) {
             vertex he_a = halfedgeIter->vertex();
84
             vertex he_b = halfedgeIter->opposite()->vertex();
85
86
             // Check if that edge is flippable
87
             // iterate over faces
88
             for (auto faceIter = tri.faces_begin(); faceIter != tri.faces_end(); ++f
89
    aceIter) {
                 half_edge h = faceIter->halfedge();
90
                 vertex face_a = h->vertex();
91
                 vertex face_b = h->opposite()->vertex();
92
                 vertex face_c = h->next()->vertex();
93
                 if(((face_a==he_a)\&\&(face_b==he_b))) | ((face_a==he_b)\&\&(face_b==he_b))
94
    _a)) ){
95
                      ++num_of_edge_faces;
96
                      if(num_of_edge_faces == 1) { other_quad_pt_a = face_c; }
                      if(num_of_edge_faces == 2) { other_quad_pt_b = face_c; }
97
98
99
                 else if ((face_a==he_a)\&\&(face_c==he_b)) \mid ((face_a==he_b)\&\&(face_a==he_b))
100
    c==he_a)) ) {
                      ++num_of_edge_faces;
101
102
                      if(num_of_edge_faces == 1) { other_quad_pt_a = face_b; }
                      if(num_of_edge_faces == 2) { other_quad_pt_b = face_b; }
103
104
                 else if ((face_b=-he_a)\&\&(face_c=-he_b)) \mid ((face_b=-he_b)\&\&(face_b=-he_b)\&\&(face_b=-he_b))
105
    c==he_a)) ) {
                      ++num_of_edge_faces;
106
                      if(num_of_edge_faces == 1) { other_quad_pt_a = face_a; }
107
                      if(num_of_edge_faces == 2) { other_quad_pt_b = face_a; }
108
109
                 else{}
110
111
             // If edge is attached to 2 faces, check if quad is convex (if edge is f
112
    lippable)
             if(num_of_edge_faces == 2) {
113
                 if(obj.is_strictly_convex_quad((he_a.point()), (other_quad_pt_a.point
114
    ()), (he_b.point()), (other_quad_pt_b.point()))) {
```

```
app/delaunay_triangulation.cpp
                                                                                     Page 3/3
Jun 15, 20 0:50
                      // It is flippable
115
                  }
116
117
             num_of_edge_faces = 0;
118
119
         }
*/
120
121
122
123
124
125
126
127
    OFF
128
    -1 -1 0
130
131 1 -1 0
132 1 1 0
133 -1 1 0
134 0 0 1
   3 0 1 4
135
   3 1 2 4
136
   3 2 3 4
   3 0 4 3
138
    */
139
140
141
142
    OFF
143
    980
144
147 1 -1 0
148 1 0 0
   1 1 0
149
   0 1 0
150
   -1 1 0
151
152 -1 0 0
153 0 0 0
154 3 0 1 8
155 3 0 8 7
156 3 1 2 3
   3 1 3 8
157
158 3 3 4 8
    3 8 4 5
3 8 5 7
159
160
   3 7 5 6
```