Readme.

For my creating Benchmarks I'm using 3 files:

1. benchmarks_arrangement.C

2. benchmark.h

3. Input_data.h

In the following I'll explain what I'm using for what.

benchmarks_arrangement.C : It's the main file .C . Here I'm creating main typedefs and reading options from command line.

benchmark.h : Here It's header file where I defined class Bench. I'll introduce all members and aim of this class later.

Input_data.h : Here I defined functions for creating data and reading it form files.

Class Bench.

This class has 3 functions:

first – create data set and put it in Arc container ;

second – put the data set to arrangement and make benchmarks;

third – create html and latex tables and put there results of benchmarks.

Creation:

Bench benchmark(char* HtmlFileName,char*LatexFileName,char* DxfFileName,const bool ONLY_DXF).

HtmlFileName -name of output html file where html table will be put.

LatexFileName -name of output Latex document where Latex table will be put.

DxfFileName – name of Dxf File.

ONLY_DXF – flag to create benchmarks only with dxf files.

Bench class can work in 2 regime. First it can create all data sets, insert them in arrangement, make benchmarks and put all results in tables. Or it can make benchmarks only with dxf files.

How to use:

All you need is:

1) Create an object of class Bench;

Bench bench(Htmlfilename,Texfilename,Dxffilename[i]);

put necessary information about what is benchmarked by using function: kernel(std::string);
It is used for descriptions in Latex and html tables.

```
      bench.kernel("Circular kernel  Circular arc traits");
```

2) make typedefs of Kernel , Traits , and ArcContainer and put them like parameters of template in one functions:

   Compute<Kernel,Traits,ArcContainer>(Dxffilename);

   or

   Compute_no_dxf<Kernel,Traits,ArcContainer>(); (For benchmarks without dxf files)

   After that Bench class will do benchmarks with data types which you have put in template parameters. And it will create two tables with results (latex and html tables) and put it in files with file names which you have set in constructor.

   If you want to use another typedef and put results in the same table, you can just repeat all steps described before just with another information and typedef by using the same Bench object.

Descriptions of folders:

arrangement_traits – benchmarks with arrangement traits

bff_reader – testing program for extended version of parser. It could be useful for creating of .bff reader. This program not finished and it creates only poligons and circles.

DxfArrayBenchmarks – benchmarks only for .dxf files

dxf_to_bff – converter from dxf to bff.

incremental insertion – in this benchmarks is using incremental insertion function.

parser – source of extended version of parser.