

```
.....
GPData.cpp
.....
# include <iostream>
# include <iomanip>
# include <cstdlib>
# include <cmath>
# include <fstream>
# include <vector>
# include <algorithm>
# include <cstring>
# include <string>

# include "Angles.hpp"
# include "GPData.hpp"

using namespace std;

//-----
// Purpose:
//
//   GnuPlot Data Class Implementation
//
// Discussion:
//
//
// Licensing:
//
//   This code is distributed under the GNU LGPL license.
//
// Modified:
//
//   2013.06.21
//
// Author:
//
//   Young Won Lim
//
// Parameters:
//
//-----
//
// void GPData::set_prefix(Angles *Ang);
// void GPData::set_suffix(Angles *Ang);
// void GPData::set_fname(Angles *Ang, const char *preStr, const char *inStr);
// void GPData::set_title(Angles *Ang, const char *inStr);
//-----
```

```
//-----  
// Class Figures' Member Functions  
//-----  
GPData::GPData()  
{  
  
    strcpy(GnuTerm, "");  
  
    strcpy(fname, "");  
    strcpy(fext, "");  
  
    strcpy(title, "");  
    strcpy(xlabel, "");  
    strcpy(ylabel, "");  
  
    strcpy(LorAstr, "");  
    strcpy(lStr, "");  
    strcpy(nStr, "");  
    strcpy(xStr, "");  
    strcpy(thStr, "");  
  
    strcpy(udata, "");  
  
    strcpy(prefix, "");  
    strcpy(suffix, "");  
  
    nPoints = 0;  
  
    useSubRange = 0;  
    valSubRange = 0;  
  
}  
  
GPData::GPData(string GTerm, int nPt)  
{  
  
    GPData();  
  
    strcpy(GnuTerm, GTerm.c_str());  
    strcpy(fext, GTerm.c_str());  
    nPoints = nPt;  
  
}  
  
GPData::GPData(string GTerm)
```

```

{
  GPData();

  strcpy(GnuTerm, GTerm.c_str());
  strcpy(fext, GTerm.c_str());
}

//-----
// prefix : Leaf_10 / All_10
//-----
void GPData::set_prefix(Angles *Ang)
{
  (Ang->getLeaf()) ? strcpy(LorAStr, "Leaf_") : strcpy(LorAStr, "All_");
  sprintf(lStr, "%d", Ang->getnIters());

  strcpy(prefix, LorAStr);
  strcat(prefix, lStr);
}

//-----
// suffix : n4096x1th0.55
//-----
void GPData::set_suffix(Angles *Ang)
{
  sprintf(nStr, "n%d", nPoints);
  // sprintf(xStr, " ");
  // sprintf(thStr, " ");
  *xStr = 0;
  *thStr = 0;
  if (useSubRange) sprintf(xStr, "x%d", valSubRange) ;
  if (Ang->getUseTh()) sprintf(thStr, "th%g", Ang->getThreshold() );

  sprintf(suffix, nStr, xStr, thStr);
}

//-----
// fname : egbX.____.circle_ang.____.eps
//-----
void GPData::set_fname(Angles *Ang, const char *preStr, const char *inStr)
{
  if (strcmp(GnuTerm, "wxt") == 0) {
    strcpy(fname, "");
    strcpy(fext, "");
    return;
  }
}

```

```
}

strcpy(fext, GnuTerm);
sprintf(fname, "%s.%s.%s.%s.%s", preStr, prefix, inStr, suffix, fext);
}

//-----
// title : Angle Tree (prefix suffix)
//-----
void GPData::set_title(Angles *Ang, const char *inStr)
{
    sprintf(title, "%s (%s %s)", inStr, prefix, suffix);
}

//-----
// xlabel :
//-----
void GPData::set_xlabel(const char *inStr)
{
    sprintf(xlabel, inStr);
}

//-----
// ylabel :
//-----
void GPData::set_ylabel(const char *inStr)
{
    sprintf(ylabel, inStr);
}

::::::::::::
GPData.hpp
::::::::::::
using namespace std;

#define useXSampling    10;
#define useXPartition  20;
#define useXSubtree    30;

//-----
// Purpose:
//
// Class GPData Interface Files
```

```
//
// Discussion:
//
// Licensing:
//
//   This code is distributed under the GNU LGPL license.
//
// Modified:
//
//   2013.06.21
//
// Author:
//
//   Young Won Lim
//
// Parameters:
//
//-----
#define GPSLEN 255

//-----
// Data structure for gnuplot call
//-----

class GPData {
public:
  GPData();
  GPData(string GTerm);
  GPData(string GTerm, int nPt);

  void set_prefix(Angles *Ang);
  void set_suffix(Angles *Ang);
  void set_fname(Angles *Ang, const char *preStr, const char *inStr);
  void set_title(Angles *Ang, const char *inStr);
  void set_xlabel(const char *inStr);
  void set_ylabel(const char *inStr);

//-----
  char GnuTerm  [GPSLEN]; // wxt, eps, emf ...

  char prefix   [GPSLEN]; // eg. Leaf_10, All_10
  char suffix   [GPSLEN]; // eg. n4192x1th1.5 ...

  char fname    [GPSLEN]; // output file name
  char fext     [GPSLEN]; // output file extension ie eps, emf, ...

  char title    [GPSLEN];
```

```
char xlabel [GPSLEN];
char ylabel [GPSLEN];

char udata [GPSLEN]; // used columns

char LorAStr [GPSLEN]; // Leaf or All
char lStr [GPSLEN]; // level
char nStr [GPSLEN]; // number of nodes
char xStr [GPSLEN]; // sampling factor
char thStr [GPSLEN]; // threshold
//-----

int nPoints;
int nSamples;
int nParts;

//-----
// useSubRnage : useXSampling 10;
// useSubRnage : useXPartition 20;
// useSubRnage : useXSubtree 30;
//-----
int useSubRange;
int valSubRange;

};
```