

```

// PointF.h
#pragma once
#define GSQN 30          // number of Gaussian quadrature points
class PointF
{
public:
    PointF(void);
    ~PointF(void);

    // Gaussian-Lagrrrier quadrature routine
    double infintg(double (PointF::*grand)(double));
    // integrand for Gaussian-Lagrrrier quadrature
    double integrandS(double x);
    // Set values of gage position (so) and distance (yo) for the stress value
    void SetGPos(double x, double y) {gl=x; dy = y;};
    void SetOrdern(int i) {nOrder = i;};
    void SetNVal(int i);
    // Compute stress due to F
    double FunS(double y);
    ////////////////////////////////////////
    // A modified Gaussian quadrature
    double intek(double a, double b, double (PointF::*grand)(double));
    double Fun1(double v);
    double Fun2(double u);
    double kFun1(double v);
    double kFun2(double u);
    double FunF(double a);
    double kFunF(double a);
private:
    int N;          // number of Gaussian-Lagrrrier quadrature points
    int nOrder;
    static double ab[GSQN];
    static double wb[GSQN];
    static double add[GSQN];
    static double wkb[GSQN];
    double (PointF::*fpt)(double x);
    double (PointF::*gpt)(double x);
    double (PointF::*hpt)(double x);
    double ao;      // current normalized depth of cut
    double Ga;      // current value of G(a)
    double yo;      // normalized distance on the plane of cut for stress due to F
    double so;      // distance from the position of F to the plane of cut
    double gl;      // Gage length
    double dy;      // Offset of gage center
public:
    double Legend(double x);
    double CFun(double aVal);
    double AveStn(double aDepth);
};

```