Experimenting with the Java Computer Language in Engineering Calculations: Application to Statically Indeterminate, Rigid, Multi-Bearing Shaft Analysis

Tonye K. Jack, Gbanaibolou Jombo

***Abstract*— Not many engineering programs written in the Java Computer Language have been applied successfully to complex engineering projects. In this article, the Java computer language is applied to the analysis of statically indeterminate beam models using the Domain and Patch exact Beam analysis method. Java source codes and subroutines for the numerical three-support beam model example presented are listed, to act as reusable productivity tool kit to aid developers to minimize development time and effort.**

***Index Terms*— Beam Deflection Analysis, Engineering Programs, Java Programming, Shaft Analysis, Shaft Design, Rigid Bearing Analysis, Statically Indeterminate Beam Analysis, Structural Analysis**

I. INTRODUCTION

Shafts on bearing supports are often modeled as beams to allow for analysis and determination of supporting reactions, and output response to external loadings in terms of deflected shape. Such bearing analyses often follow a two step process – (a.) rigid bearing analysis, and (b.) flexibility of supporting bearing structure [1]. Mischke, [2], has provided methods of shaft analysis. A supporting program in the BASIC computer language has been written by Zarrugh based on the method of [2], and reported in [3]. For statically determinate two-bearing support models, rigid-body mechanics by way of, the equations of force and moment equilibrium are satisfactory to analyze such models. Tables of beam formulae in Engineering Handbooks may be satisfactory in meeting the Engineer's need, once acceptable models have been formulated. However, for more complex shafts, often long and having higher degrees of redundancy, with interconnected coupled and coupling units, and supported on several supports usually three or more, the conditions of static equilibrium together with additional information on the beam shape and behaviour in response to loadings is required to evaluate the shaft model. Whilst Tables of Beam Formulae together with the Method of Superposition can be applied to such Complex Shafts models when broken down to simpler beam models, for which appropriate formulae is obtainable from beam tables, errors can occur in reducing such complex models into simpler cases.

Manuscript Received on June 25, 2013

Tonye K. Jack, University Teacher, Department of Mechanical Engineering Port Harcourt, Rivers State, Nigeria.

Gbanaibolou Jombo, Studying for the Masters Degree in Design of Rotating Machines at the Cranfield University in England

Other beam analysis methods such as the Energy, Singularity Function, amongst others can also be applied [4]-[6]. The Double Integration exact solution method of *Domain and Patch* [4] is often the best approach. Though lengthy in analysis method, with several parameters to consider, the step-by-step approach allows for calculation review to trace for errors. This makes it appropriate for the method to be computerized.

In the sections that follow, the theoretical basis for the domain and patch method is explained, through a step-by-step algorithm. A hand-calculation approach for determining all unknown parameters using a model example application of a three-bearing support shaft system is shown. The method of java program application to the model example calculation is then given with all class source codes, code development and structure flow-diagrams, the interface development technique, with the identifiable input requirements.

II. THE METHOD OF DOMAIN AND PATCH

A. The General Beam Flexure Formula

The general *beam flexure double integral formula* defined by (1), applied to beam models allows for the slope and, hence, deflection to be obtained through a two-stage integration process.

$$M = EI\frac{d^2y}{dx^2} \tag{1}$$

For typical horizontal beam models, the coordinate interpretation with reference to (1) implies that, the horizontal neutral axis of the beam is along the *x*-axis, and the vertical, *y*-axis which gives response deflection due to external loading effect after a second integration of (1). The first integration gives the slope. Equation (1) is thus, the rate of change of the deflection, *y*, along the horizontal beam, *x*-axis, when subjected to vertical force or curvature moment loadings. The *EI*, often called the "Stiffness Constant" defines the material and geometric property of the beam. A model often assumed for strength of materials is a homogenous, elastic material with unidirectional material and geometric linearity - an isotropic material [7]. A convenient solution approach to (1) allows for the equation to be written as (1a):

$$\frac{d^2y}{dx^2} = \frac{M}{EI} \tag{1a}$$

This allows for, and clarifies the two-step integration to obtain the slope in the first instance, and then the deflection when applied to each of the identified and defined *domains*.

B. *Proposed Solution Algorithm*

The step-by-step application of (1) to beam models by the method of *domain* and *patch*, which allows for backtracking for error checks is as follows, based on the model of fig. (1):
Step 1: Define the coordinate and apply force and moment equilibrium conditions to write-out the reaction equations
Step 2: Define the Domains
Step 3: For each of the domains, apply the flexure formula to obtain the slope and deflection equations.
Step 4: Apply Boundary conditions
Step 5: Apply the Patch conditions of: Equality of Slopes and Deflections at the beginning and end of any two follow-up domains
Step 6: Solve derived equations simultaneously or by any convenient method to obtain required reactions, slopes or deflections.

C. *Shaft Model*

In the fig.(1) is shown a two-span beam model of total length, *L*, on three-simple supports with two concentrated forces, F_1 and F_2 on either span. The span lengths are respectively, *a*, and *b*, on the left-, and right-sides. The corresponding free-body diagram is shown in fig. (2).The coordinate system adopted is indicated.

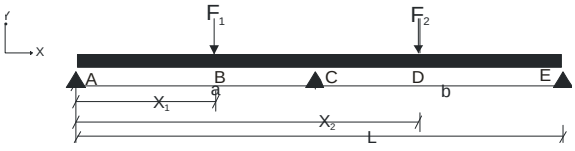


Fig. (1) : Shaft Model as a Beam on Three Rigid Bearing Supports

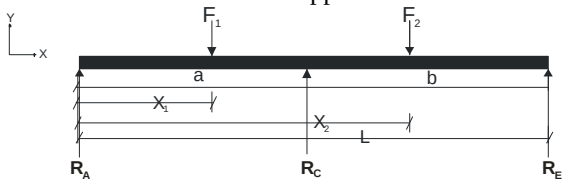


Fig. (2) Free-Body-Diagram of Shaft Beam Model with Support Reactions

III. MATHEMATICAL SOLUTION FORMULATION AND ANALYSIS

Mathematical Method of solution follows the approach of [8], [9], and [4].

A. *Reaction Equations*

Force and moment equilibrium are applied to obtain the reaction equations.
Summation of the forces on the y-axis,

$$+\uparrow \sum F_y = 0; \quad R_A + R_C + R_E - F_1 - F_2 = 0 \qquad (2)$$

Simplifying, $R_A + R_C + R_E = F_1 + F_2$ (2a)

$$R_E = F_1 + F_2 - R_A - R_C \qquad (2b)$$

Summation of the moments about point *E*,

$$\sum M_E = 0;$$

$$-R_A L - R_c b + F_1(L - x_1) + F_2(L - x_2) = 0$$

$$R_A L + R_c b = F_1(L - x_1) + F_2(L - x_2) \qquad (3)$$

$$R_C = \frac{1}{b}\left[F_1(L - x_1) + F_2(L - x_2) - R_A L\right] \qquad (3a)$$

B. *Domains: Definitions and Analysis Formulation*

Domain definition allows for a load-to-load determination of the moment equation applicable between those particular set of loads. The sections to consider on a load-to-load basis are: R_A-to-F_1, F_1-to-R_C, R_C-to-F_2, and F_2-to-R_E. Thus, for the model example, *four domains* are identified. The domains are usually expressed as inequalities and written as:

Domains:

$$(0 \le x \le x_1), (x_1 \le x \le a), (a \le x \le x_2), (x_2 \le x \le L)$$

Analysis for first domain: $(0 \le x \le x_1)$;

The moment equation for this domain is given by (4), applicable at any point, *x*, from A:

$$M_x = R_A x \qquad (4)$$

Substituting in (1a), gives:

$$\frac{d^2 y}{dx^2} = \frac{1}{EI}(R_A x) \qquad (5)$$

A first integration gives the slope, $(dy/dx)_{AB}$,

$$\left(\frac{dy}{dx}\right)_{AB} = \frac{1}{EI}\left(R_A \frac{x^2}{2} + C_1\right) \qquad (6)$$

A second integration gives the deflection, y_{AB},

$$y_{AB} = \frac{1}{EI}\left(R_A \frac{x^3}{6} + C_1 x + C_2\right) \qquad (7)$$

Analysis for second domain: $(x_1 \le x \le a)$;
The moment equation for this domain is given by (8),

$$M_x = R_A x - F_1(x - x_1) \qquad (8)$$

Thus, by (1a):

$$\frac{d^2 y}{dx^2} = \frac{1}{EI}\left[R_A x - F_1(x - x_1)\right] \qquad (9)$$

The Slope, $(dy/dx)_{BC}$, is given by (10):

$$\left(\frac{dy}{dx}\right)_{BC} = \frac{1}{EI}\left[R_A \frac{x^2}{2} - F_1 \frac{(x - x_1)^2}{2} + C_3\right] \qquad (10)$$

(3)

The Deflection, y_{BC}, is given by (11):

(3a)

$$y_{BC} = \frac{1}{EI}\left[R_A \frac{x^3}{6} - F_1 \frac{(x - x_1)^3}{6} + C_3 x + C_4\right] \qquad (11)$$

Analysis for third domain: $(a \le x \le x_2)$;

The moment equation for this domain is given by (12),

$$M_x = R_A x - F_1(x - x_1) + R_C(x - a)) \quad (12)$$

Equation (1a) now applied, gives,

$$\frac{d^2y}{dx^2} = \frac{1}{EI}[R_A x - F_1(x - x_1) + R_C(x - a)] \quad (13)$$

The slope, $(dy/dx)_{CD}$ is,

$$\left(\frac{dy}{dx}\right)_{CD} = \frac{1}{EI}\left[R_A\frac{x^2}{2} - F_1\frac{(x - x_1)}{2} + R_C\frac{(x - a)^2}{2} + C_5\right] \quad (14)$$

The deflection, y_{CD} is,

$$y_{CD} = \frac{1}{EI}\left[R_A\frac{x^3}{6} - F_1\frac{(x - x_1)^3}{6} + R_C\frac{(x - a)^3}{6} + C_5x + C_6\right] \quad (15)$$

Analysis for fourth domain: $(x_2 \le x \le L)$;

The Moment equation is given by (16),

$$M_x = R_A x - F_1(x - x_1) + R_C(x - a) - F_2(x - x_2) \quad (16)$$

And from (1a),

$$\frac{d^2y}{dx^2} = \frac{1}{EI}[R_A x - F_1(x - x_1) + R_C(x - a) - F_2(x - x_2)] \quad (17)$$

The slope, $(dy/dx)_{DE}$ is,

$$\left(\frac{dy}{dx}\right)_{DE} = \frac{1}{EI}\left[R_A\frac{x^2}{2} - F_1\frac{(x - x_1)^2}{2} + R_C\frac{(x - a)^2}{2} - F_2\frac{(x - x_2)^2}{2} + C_7\right] \quad (18)$$

The deflection, y_{DE}, is,

$$y_{DE} = \frac{1}{EI}\left[R_A\frac{x^3}{6} - F_1\frac{(x - x_1)^3}{6} + R_C\frac{(x - a)^3}{6} - F_2\frac{(x - x_2)^3}{6} + C_7x + C_8\right] \quad (19)$$

C. Boundary Conditions

The boundary conditions, or BC, allow for determination of some of the arbitrary constants of integration, and the unknowns. These boundary conditions are usually obtained from conditions at the type of supports. Shames, [4] further notes that, for statically indeterminate beams, the degree of redundancies provides additional boundary conditions equal to the degree of redundancies. For the three-support shaft beam model, the degree of redundancy is one i.e., 1. Thus, an additional BC of one is obtainable from the model under consideration.

At the simple support A, i.e., x=0, deflection, y=0; substitution in (7) gives,

$$C_2 = 0$$

At support C: $x = a$, deflection, y=0, substituting in (11) and (15), we obtain:

For $(x_1 \le x \le a)$:

$$0 = R_A\frac{a^3}{6} - F_1\frac{(a - x_1)^3}{6} + C_3a + C_4 \quad (20)$$

For $(a \le x \le x_2)$:

$$0 = R_A\frac{a^3}{6} - F_1\frac{(a - x_1)^3}{6} + C_5a + C_6 \quad (21)$$

Equating (20) and (21)

$C_3a+C_4=C_5a+C_6$ (22)

At simple support E: x=L, deflection, y=0

$$0 = R_A\frac{L^3}{6} - F_1\frac{(L - x_1)^3}{6} + R_C\frac{b^3}{6} - F_2\frac{(L - x_2)^3}{6} + C_7L + C_8 \quad (23)$$

D. Patch Conditions

The defined domains fall within bounds each having a beginning and an end as stipulated by the inequality. Close observation of these domain inequalities shows that conditions at the end of one domain and the beginning of the follow-up domain are same. Thus, the slope and deflection at the end of a domain must equal the slope and deflection of the next follow-up domain. This is then the requirement for the *Patch conditions*. The "Patching" simplifies the analysis process, thus allowing for the slope equations in the two domains to be equated. Same applies for the deflection equations [4].

At $x = x_1$, the domains, just to the left (i.e., $x = x_1^-$) and just to the right (i.e., $x = x_1^+$), which represents the end of a domain -and-beginning of the next domain, equating slopes:

$$\frac{dy}{dx}_{AB} = \frac{dy}{dx}_{BC}$$

$$R_A\frac{x^2}{2} + C_1 = R_A\frac{x^2}{2} + C_3$$

$C_1=C_3$

Again, at $(x = x_1^-)$ and $(x = x_1^+)$, equating deflections:

$y_{AB}=y_{BC}$

$$R_A\frac{x^2}{2} + C_1x + C_2 = R_A\frac{x^2}{2} + C_3x + C_4$$

$C_2=C_4$

At $x = a$, i.e. for the domains $(x = a^-)$ and $(x = a^+)$, equating slopes:

$$\frac{dy}{dx}_{BC} = \frac{dy}{dx}_{CD}$$

$$R_A\frac{x^2}{2} - F_1\frac{(a - x_1)^2}{2} + C_3 = R_A\frac{x^2}{2} - F_1\frac{(a - x_1)^2}{2} + C_5$$

$C_3=C_5$

From (22)

$C_4=C_6$

Again, at $x = x_2$, i.e. for the domains $(x = x_2^-)$ and $(x = x_2^+)$, equating slopes:

$$\frac{dy}{dx}_{CD} = \frac{dy}{dx}_{DE}$$

$$R_A\frac{x_2^2}{2} - F_1\frac{(x_2 - x_1)^2}{2} + R_C\frac{(x_2 - a)^2}{2} + C_5 = R_A\frac{x_2^2}{2} - F_1\frac{(x_2 - x_1)^2}{2} + R_C\frac{(x_2 - a)^2}{2} + C_7$$

$C_5=C_7$

At $x = x_2$, i.e. for the domains $(x = x_2^-)$ and $(x = x_2^+)$, equating deflections:

$y_{CD} = y_{DE}$

$$R_A\frac{x_2^3}{6} - F_1\frac{(x_2 - x_1)^3}{6} + R_C\frac{(x_2 - a)^3}{6} + C_5x_2 + C_6 =$$

$$R_A\frac{x_2^3}{6} - F_1\frac{(x_2 - x_1)^3}{6} + R_C\frac{(x_2 - a)^3}{6} + C_7x_2 + C_8$$

$C_6=C_8$

$$\therefore \quad C_2 = C_4 = C_6 = C_8 = 0$$

$$C_1 = C_3 = C_5 = C_7$$

From (20)

$$R_A\frac{a^3}{6}-F_1\frac{(a-x_1)^3}{6}+C_3a=0$$

$$C_3=F_1\frac{(a-x_1)^3}{6a}-R_A\frac{a^2}{6}$$

Since $C_1=C_3$

$$C_1=F_1\frac{(a-x)^3}{6a}-R_A\frac{a^2}{6}$$

From (23):

$$R_A\frac{L^3}{6}-F_1\frac{(L-x_1)^3}{6}+R_C\frac{b^3}{6}-F_2\frac{(L-x_2)^3}{6}+C_7L=0$$

$$R_A\frac{L^3}{6}-F_1\frac{(L-x_1)^3}{6}+\frac{b^2}{6}\left[F_1(L-x_1)+F_2(L-x_2)-R_AL\right]-F_2\frac{(L-x_2)^3}{6}$$
$$+\left(F_1\frac{(a-x_1)^3}{6a}-R_A\frac{a^2}{6}\right)L=0$$

$$R_A\frac{L^3}{6}-F_1\frac{(L-x_1)^3}{6}+F_1\frac{b^2}{6}(L-x_1)+F_2\frac{b^2}{6}(L-x_2)-R_A\frac{b^2}{6}L-F_2\frac{(L-x_2)^3}{6}$$
$$+F_1\frac{(a-x_1)^3L}{6a}-R_A\frac{a^2}{6}L=0$$

$$R_A\frac{L^3}{6}-R_A\frac{b^2}{6}L-R_A\frac{a^2}{6}L-F_1\frac{(L-x_1)^3}{6}+F_1\frac{b^2}{6}(L-x_1)+F_2\frac{b^2}{6}(L-x_2)-F_2\frac{(L-x_2)^3}{6}$$
$$+F_1\frac{(a-x_1)^3L}{6a}=0$$

$$R_A\frac{\left(L^2-b^2-a^2\right)L}{6}+\frac{F_1}{6}\left[-(L-x_1)^3+b^2(L-x_1)+\frac{(a-x_1)^3L}{a}\right]$$
$$+\frac{F_2}{6}\left[b^2(L-x_2)-(L-x_2)^3\right]=0$$

$$R_A\frac{abL}{3}+\frac{F_1}{6}\left[b^2(L-x_1)-(L-x_1)^3+\frac{(a-x_1)^3L}{a}\right]$$
$$+\frac{F_2}{6}\left[b^2(L-x_2)-(L-x_2)^3\right]=0$$

$$R_A\frac{abL}{3}=-\frac{F_1}{6}\left[b^2(L-x_1)-(L-x_1)^3+\frac{(a-x_1)^3L}{a}\right]$$
$$-\frac{F_2}{6}\left[b^2(L-x_2)-(L-x_2)^3\right]=0$$

$$R_A=\frac{1}{abL}\left(\frac{F_1}{2}\left[(L-x_1)^3-b^2(L-x_1)-\frac{(a-x_1)^3L}{a}\right]+\frac{F_2}{2}\left[(L-x_2)^3-b^2(L-x_2)\right]\right)$$

IV. HANDLING CHANGING SHAFT CROSS-SECTIONS

Use of the uniform diameter cross-sectioned shaft beam model in figs. (1) and (2), represents the ideal. Practical industrial shafts are often of varying cross-sections, and referred to as stepped-shafts. Solution methods for stepped-shafts available in the program allow for abutments for bearings, impellers, couplings and other such equipment parts features to be handled through the application of the concept of effective diameter.

A. The Concept of Effective Diameter

To install various mountings (gears, pulleys, impellers, etc.) on a stepped-shaft, the diameter of each section needs to be determined. And, to determine the diameter of the shaft sections will require that the shaft designer has knowledge of the bending moment and torque distribution throughout the length of the shaft [7]. For a constant torque applied at one end of a full shaft length, with one end fixed, the Moment Equilibrium equation is applied, and the following torsional twist, θ, relation:

$$\theta=\frac{TL}{GJ} \tag{24}$$

With, $\frac{T}{\theta}=\frac{GJ}{L}$, being defined as the torsional rigidity of the shaft, - the torque required to produce a displacement of θ, radians. Where, for a solid shaft, polar moment of inertia relation:

$$J=\frac{\pi D^4}{32} \tag{24a}$$

For non-uniform diameter shafts connected in series or with its flexible elements in series it is common practice to replace each section of the actual shaft model with an equivalent length of shaft of some reference diameter [10]. It can be shown that:

$$\theta_e=\theta_1+\theta_2+\theta_3+\theta_4+\theta_5+\theta_6 \tag{25}$$

$$\frac{32TL_e}{\pi GD_e^{\,4}}=\frac{32TL_1}{\pi GD_1^{\,4}}+\frac{32TL_2}{\pi GD_2^{\,4}}+\frac{32TL_3}{\pi GD_3^{\,4}}+\frac{32TL_4}{\pi GD_4^{\,4}}+\frac{32TL_5}{\pi GD_5^{\,4}}+\frac{32TL_6}{\pi GD_6^{\,4}}$$

$$\frac{32TL_e}{\pi GD_e^{\,4}}=\frac{32T}{\pi G}\left(\frac{L_1}{D_1^{\,4}}+\frac{L_2}{D_2^{\,4}}+\frac{L_3}{D_3^{\,4}}+\frac{L_4}{D_4^{\,4}}+\frac{L_5}{D_5^{\,4}}+\frac{L_6}{D_6^{\,4}}\right) \tag{26}$$

$$\frac{L_e}{D_e^{\,4}}=\left(\frac{L_1}{D_1^{\,4}}+\frac{L_2}{D_2^{\,4}}+\frac{L_3}{D_3^{\,4}}+\frac{L_4}{D_4^{\,4}}+\frac{L_5}{D_5^{\,4}}+\frac{L_6}{D_6^{\,4}}\right) \tag{27}$$

The equivalent length, L_e is obtained from,

$$\Sigma L_e=L_1+L_2+L_3+L_4+L_5+L_6 \tag{28}$$

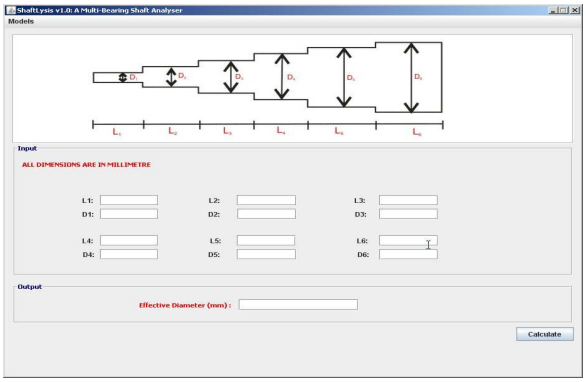


Fig. (3): Interface for Equivalent Diameter calculation

Where, θ_e, L_e, D_e, are the equivalent or effective, twist, diameter, and lengths respectively. The stepped-shaft being viewed as one which has been reduced to its effective diameter, having same torsional stiffness as the sum of each of the sections twists' combined. We can solve for the equivalent diameter as:

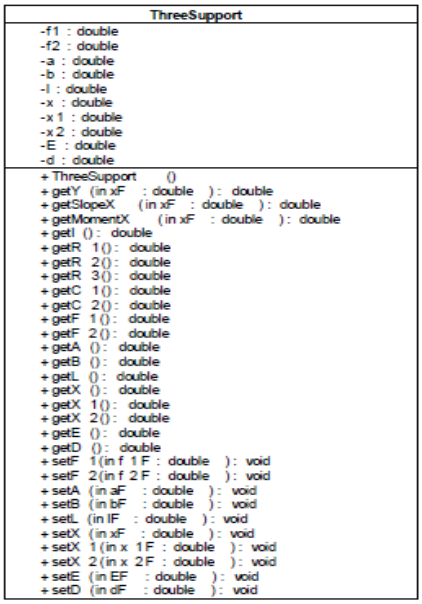
$$D_e^{\,4}=\frac{L_e}{\left(\frac{L_1}{D_1^{\,4}}+\frac{L_2}{D_2^{\,4}}+\frac{L_3}{D_3^{\,4}}+\frac{L_4}{D_4^{\,4}}+\frac{L_5}{D_5^{\,4}}+\frac{L_6}{D_6^{\,4}}\right)}$$

$$D_e=\sqrt[4]{\frac{L_e}{\left(\frac{L_1}{D_1^{\,4}}+\frac{L_2}{D_2^{\,4}}+\frac{L_3}{D_3^{\,4}}+\frac{L_4}{D_4^{\,4}}+\frac{L_5}{D_5^{\,4}}+\frac{L_6}{D_6^{\,4}}\right)}} \tag{29}$$

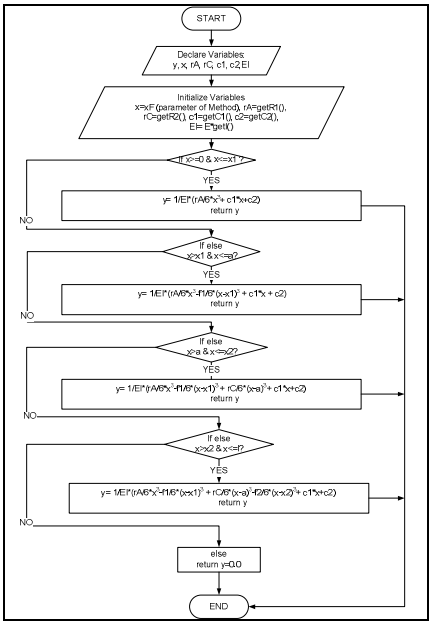
V. THE JAVA PROGRAM –CLASSES, METHODS, ATTRIBUTES, CLASS DIAGRAMS AND FLOWCHARTS FOR CLASSES

The building blocks in object-oriented programs such as Java are the classes. The classes are a collection of methods which work together for the effective execution of the program [11]. The ISO standardised graphical language adopted for incremental object-oriented software development process is the Unified Modelling Language (UML). Developed by Rational Software, now an IBM subsidiary, it is more like a framework of methods, activities and sequence graphic diagrams of visual models of association of data structures for best practice efficient program development [12, 13]. In the sections that follow, a breakdown of classes for the three-support beam model are defined and grouped together in a class diagram for the full program architecture. The Structure flow diagrams by way of case diagrams of association and relationships of the class object attributes, methods and sequence of operation then follows.

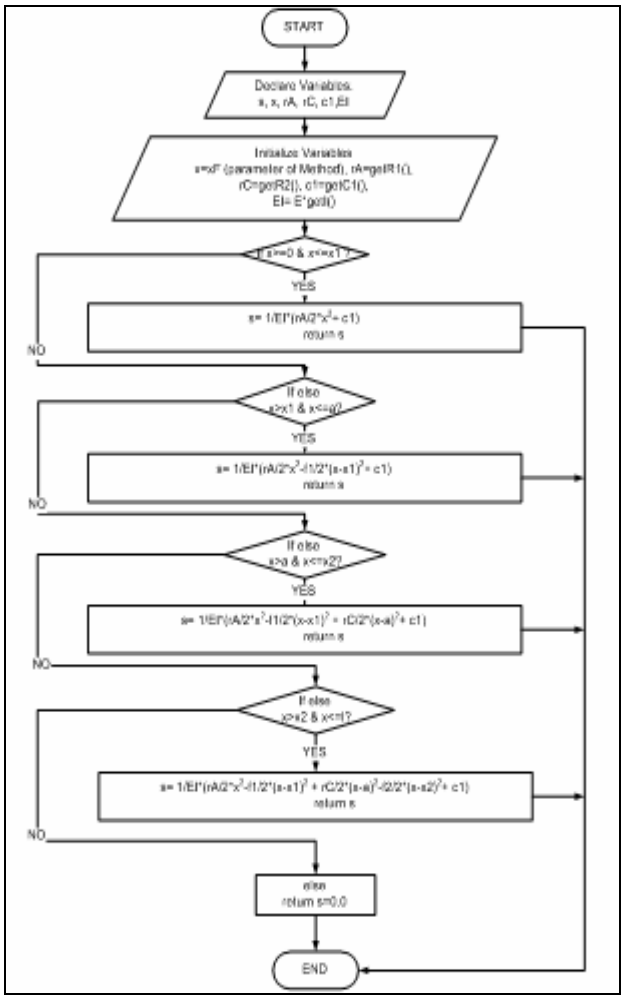
A. Three-support Beam Model UML Class Diagram



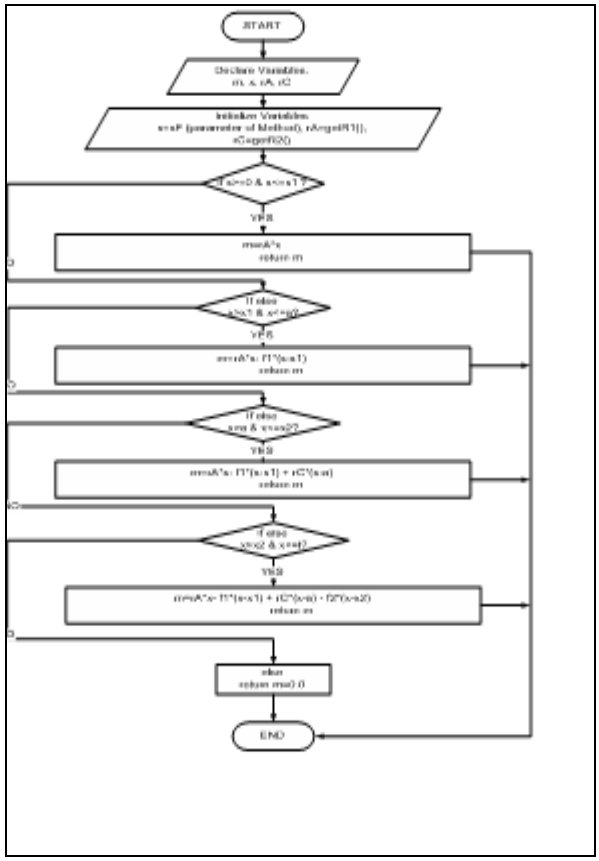
B. Flow diagram for getY(double):double



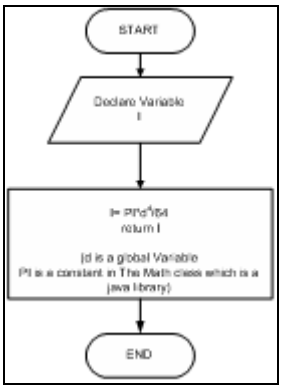
C. Flow diagram for getSlopeX(double):double



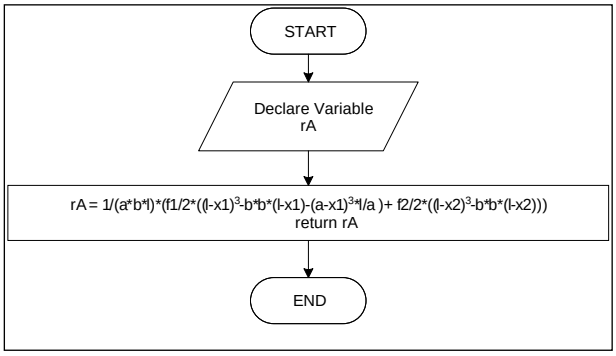
D. Flow diagram for getMomentX(double):double



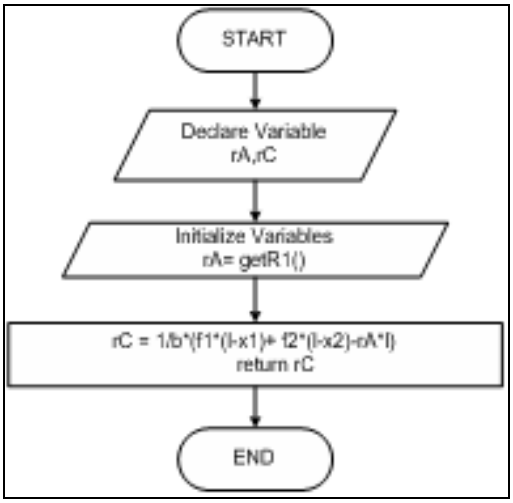
E. Flow diagram for getI():double



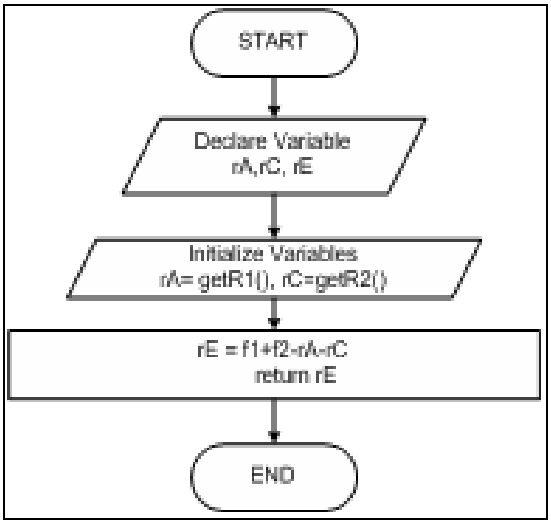
F. Flow diagram for getR1():double



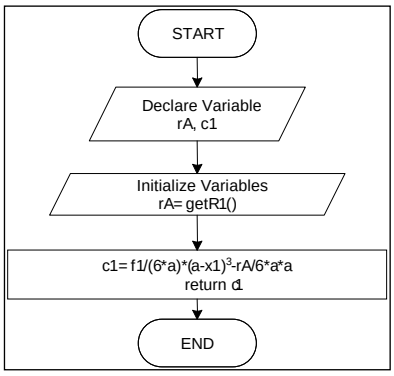
G. Flow diagram for getR2():double



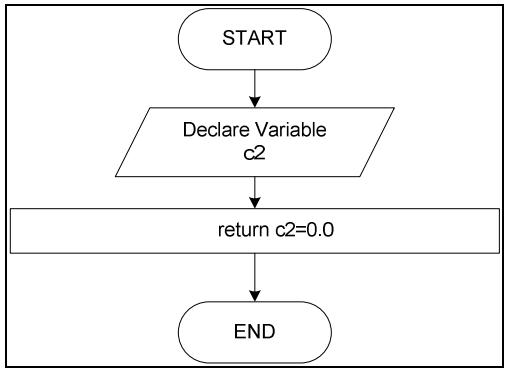
H. Flow diagram for getR3():double



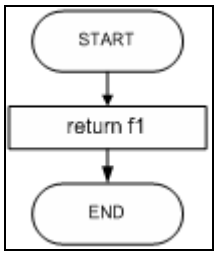
I. Flow diagram for getC1():double



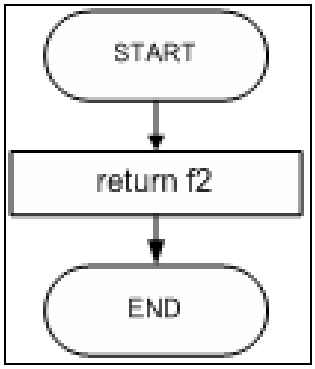
J. Flow diagram for getC2():double



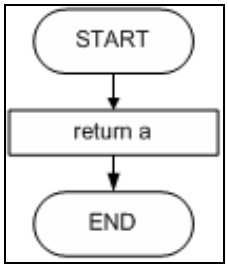
K. Flow diagram for ***getF1():double***



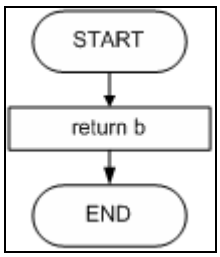
L. Flow diagram for getF2():double



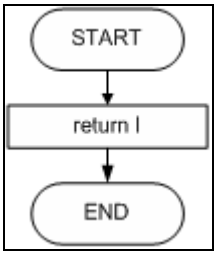
M. Flow diagram for getA():double



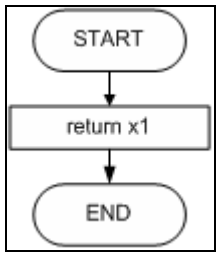
N. Flow diagram for getB():double



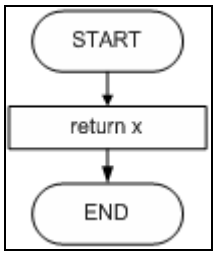
O. Flow diagram for getL():double



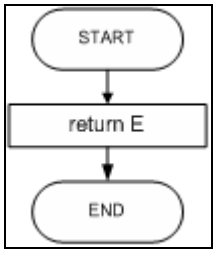
P. Flow diagram for getX1():double



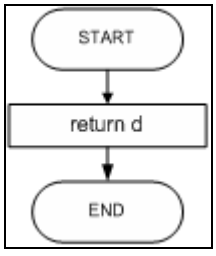
Q. Flow diagram for getX():double



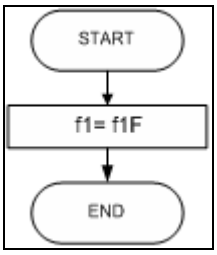
R. Flow diagram for getE():double



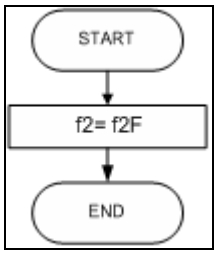
S. Flow diagram for getD():double



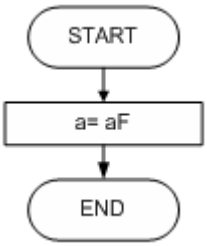
T. Flow diagram for setF1(double):void



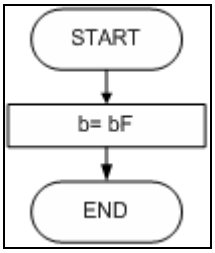
U. Flow diagram for setF2(double):void



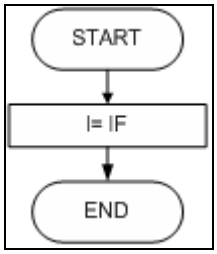
V. Flow diagram for setA(double):void



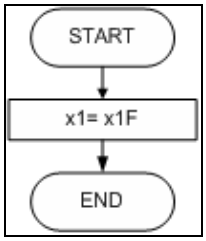
W. Flow diagram for setB(double):void



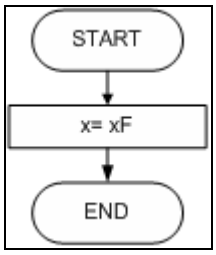
X. Flow diagram for setL(double):void



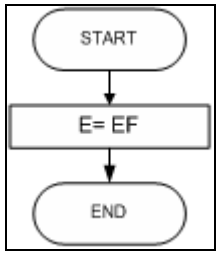
Y. Flow diagram for setX1(double):void



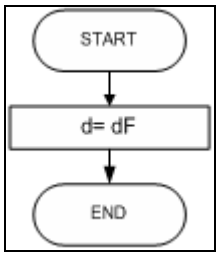
Z. Flow diagram for setX(double):void



AA. Flow diagram for setE(double):void



BB. Flow diagram for setD(double):void



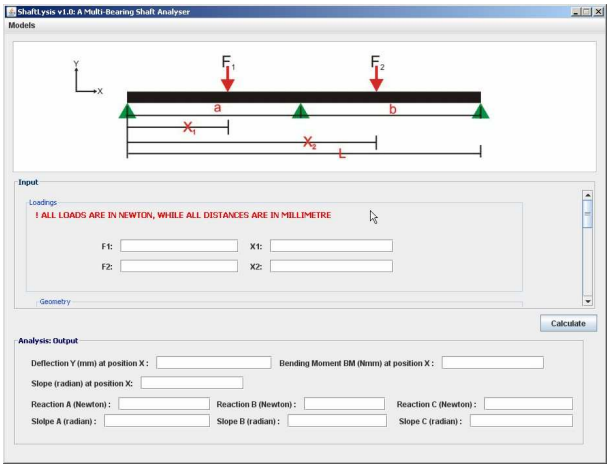


Fig. (4): Interface for Beam Analysis Calculations

VI. CONCLUSION

The method illustrated in this paper was extensively applied successfully by Jombo, [9] to other Beam Type Models of: Shaft beam on two-simple supports, four-support shaft beam model, beams of varying loads with single overhang, and beams with varying load of double overhang. Attempts have been made by Ezeh, [14] in applications of the Java language to Gear Design Calculations with some success. The source codes provided is to act as adaptable blueprint to aid program developers to apply the Java language to other engineering applications, since in the real world there are many objects which are of same type and share certain characteristics and common elements for which they can be grouped into classes and inherit features from other formed objects [15]-[16]. Java's platform independence has great appeal.

APPENDIX 1: THE THREE SUPPORT BEAM CLASS SOURCE CODE FOR DEVELOPED PROGRAM

```
package beamModels;

import static java.lang.Math.*;

/**
 * A class that models a beam supported on three supports separated
by
 * a and b respectively and supporting each of the load f1 and f2 on
 * each of the spans respectively.
 *
 * @author Jombo Gbanaibolou
 */
public class ThreeSupport {

    // input variable declarations
    // the force inputs in newton.
    private double f1;
    private double f2;

    // distance between the supports in millimetre.
    private double a;
    private double b;

    private double l;       // span of beam in mm

    private double x;       // any distance along the beam

    // positions of the respective forces in millimetre.
    private double x1;
    private double x2;

    private double E;       // young's modulus of elasticity in N/mm^2.
    private double d;       // effective diameter of the shaft in mm.

    public ThreeSupport(){}

    // getter methods for theanalysis properties

    public double getY(double xF){

        double y;           // deflection of the beam along any position x
        double x= xF;           // any position along the beam in mm
        double rA= getR1();
        double rC= getR2();
        double c1= getC1();
        double c2= getC2();
        double EI= E*getI();          // flexural rigidity of the shaft in
Nmm^2

        if(x>=0 & x<=x1){
            y= 1/EI*(rA/6*pow(x,3)+ c1*x+c2);
            return y;
        }
        else if (x>x1 & x<=a){
            y= 1/EI*(rA/6*pow(x,3)-f1/6*pow((x-x1),3)+ c1*x+c2);
            return y;
        }
        else if (x>a & x<=x2){
            y= 1/EI*(rA/6*pow(x,3)-f1/6*pow((x-x1),3) +
rC/6*pow((x-a),3)+ c1*x+c2);
            return y;
        }
        else if (x>x2 & x<=l){
            y= 1/EI*(rA/6*pow(x,3)-f1/6*pow((x-x1),3) +
rC/6*pow((x-a),3)-f2/6*pow((x-x2),3)+ c1*x+c2);
            return y;
        }
```

```
        else
            return y=0;


    }

    public double getSlopeX(double xF){

        double s;               //slope along any position x on the
        double x= xF;           // any position along the beam
        double rA= getR1();
        double rC= getR2();
        double c1= getC1();
        double EI= E*getI();          // flexural rigidity of the shaft in
Nmm^2

        if(x>=0 & x<=x1){
            s= 1/EI*(rA/2*pow(x,2)+ c1);
            return s;
        }
        else if (x>x1 & x<=a){
            s= 1/EI*(rA/2*pow(x,2)-f1/2*pow((x-x1),2)+ c1);
            return s;
        }
        else if (x>a & x<=x2){
            s= 1/EI*(rA/2*pow(x,2)-f1/2*pow((x-x1),2) +
rC/2*pow((x-a),2)+ c1);
            return s;
        }
        else if (x>x2 & x<=l){
            s= 1/EI*(rA/2*pow(x,2)-f1/2*pow((x-x1),2) +
rC/2*pow((x-a),2)-f2/2*pow((x-x2),2)+ c1);
            return s;
        }
        else
            return s=0;
    }

    public double getMomentX(double xF){

        double m;           // bending moment along any positon on the
beam in Nmm
        double x= xF;       // any position along the beam in mm.
        double rA= getR1();
        double rC= getR2();

        if(x>=0 & x<=x1){
            m=rA*x;
            return m;
        }
        else if (x>x1 & x<=a){
            m=rA*x- f1*(x-x1);
            return m;
        }
        else if (x>a & x<=x2){
            m=rA*x- f1*(x-x1) + rC*(x-a);
            return m;
        }
        else if (x>x2 & x<=l){
             m=rA*x- f1*(x-x1) + rC*(x-a) - f2*(x-x2);
            return m;
        }
        else
            return m=0;

    }


    // getter methods for properties needed in other calculations

    public double getI(){

        double I;       // Area moment of inertia of the beam
        I= PI*pow(d,4)/64;
        return I;
    }

    public double getR1(){

        double rA;          // reaction at the first support in Newtons

        rA =
1/(a*b*l)*(f1/2*(pow((l-x1),3)-b*b*(l-x1)-pow((a-x1),3)*l/a )+
f2/2*(pow((l-x2),3)-b*b*(l-x2)));
        return rA;
    }

    public double getR2(){

        double rC;      // reaction at the second support
        double rA= getR1();

        rC = 1/b*(f1*(l-x1)+ f2*(l-x2)-rA*l);
        return rC;
    }

    public double getR3(){

        double rE;      // reaction at the third support
        double rA= getR1();
        double rC= getR2();

        rE = f1+f2-rA-rC;
        return rE;
    }

    public double getC1(){

        double c1;          //odd constant of integration
        double rA= getR1();
        c1= f1/(6*a)*pow((a-x1),3)-rA/6*a*a;
        return c1;
    }

    public double getC2(){

        double c2=0;      // even constant of integration
        return c2;
    }

    // getter method for the input variables

    public double getF1(){
        return f1;
    }

    public double getF2(){
        return f2;
    }

    public double getA(){
        return a;
    }

    public double getB(){
        return b;
    }

    public double getL(){
        return l;
```

```
    }

    public double getX(){
        return x;
    }

    public double getX1(){
        return x1;
    }

    public double getX2(){
        return x2;
    }

    public double getE(){
        return E;
    }

    public double getD(){
        return d;
    }

    // setter methods for all the variables

    public void setF1(double f1F){
        f1= f1F;
    }

    public void setF2(double f2F){
        f2= f2F;
    }

    public void setA(double aF){
        a= aF;
    }

    public void setB(double bF){
        b= bF;
    }

    public void setL(double lF){
        l= lF;
    }

    public void setX(double xF){
        x= xF;
    }

    public void setX1(double x1F){
        x1= x1F;
    }

    public void setX2(double x2F){
        x2= x2F;
    }

    public void setE(double EF){
        E= EF;
    }

    public void setD(double dF){
        d= dF;
    }

}
```

APPENDIX 2: THE THREE SUPPORT BEAM PANEL SOURCE CODE

```
package MyPanels;

import beamModels.*;

/**
 *
 * @author Jombo Gbanaibolou
 */
public class ThreeSupportPanel extends javax.swing.JPanel {

    private ThreeSupport three;

    /** Creates new form SimplySupported */
    public ThreeSupportPanel() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this
method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN:initComponents
    private void initComponents() {

        banner = new javax.swing.JPanel();
        imageL = new javax.swing.JLabel();
        inputScrollPane = new javax.swing.JScrollPane();
        inputPanel = new javax.swing.JPanel();
        loadingsPanel = new javax.swing.JPanel();
        f2L = new javax.swing.JLabel();
        f2TF = new javax.swing.JTextField();
        f1L = new javax.swing.JLabel();
        f1TF = new javax.swing.JTextField();
        x1L = new javax.swing.JLabel();
        x2L = new javax.swing.JLabel();
        x1TF = new javax.swing.JTextField();
        x2TF = new javax.swing.JTextField();
        loadingsInstructionL = new javax.swing.JLabel();
        geometryPanel = new javax.swing.JPanel();
        aL = new javax.swing.JLabel();
        bL = new javax.swing.JLabel();
        lenL = new javax.swing.JLabel();
        aTF = new javax.swing.JTextField();
        bTF = new javax.swing.JTextField();
        lenTF = new javax.swing.JTextField();
        diaL = new javax.swing.JLabel();
        positionXL = new javax.swing.JLabel();
        diaTF = new javax.swing.JTextField();
        positionXTF = new javax.swing.JTextField();
        geometryInstructionL = new javax.swing.JLabel();
        MatPropPanel = new javax.swing.JPanel();
        youngL = new javax.swing.JLabel();
        inertiaL = new javax.swing.JLabel();
        inertiaTF = new javax.swing.JTextField();
        youngTF = new javax.swing.JTextField();
        outputPanel = new javax.swing.JPanel();
        deflXL = new javax.swing.JLabel();
        deflXTF = new javax.swing.JTextField();
        momXL = new javax.swing.JLabel();
        momXTF = new javax.swing.JTextField();
        slpXL = new javax.swing.JLabel();
        slpXTF = new javax.swing.JTextField();
        reacAL = new javax.swing.JLabel();
        reacATF = new javax.swing.JTextField();
        reacBL = new javax.swing.JLabel();
        reacBTF = new javax.swing.JTextField();
        slpAL = new javax.swing.JLabel();
```

```
        slpATF = new javax.swing.JTextField();
        slpBL = new javax.swing.JLabel();
        slpBTF = new javax.swing.JTextField();
        reacCL = new javax.swing.JLabel();
        reacCTF = new javax.swing.JTextField();
        slpCL = new javax.swing.JLabel();
        slpCTF = new javax.swing.JTextField();
        calculate = new javax.swing.JButton();

        banner.setBackground(new java.awt.Color(255, 255, 255));

banner.setBorder(javax.swing.BorderFactory.createEtchedBorder()
);

        imageL.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/images/threeSupp
ort.gif"))); // NOI18N

        javax.swing.GroupLayout bannerLayout = new
javax.swing.GroupLayout(banner);
        banner.setLayout(bannerLayout);
        bannerLayout.setHorizontalGroup(

bannerLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
            .addGroup(bannerLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(imageL,
javax.swing.GroupLayout.PREFERRED_SIZE, 723,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(63, Short.MAX_VALUE))
        );
        bannerLayout.setVerticalGroup(

bannerLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
            .addGroup(bannerLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(imageL,
javax.swing.GroupLayout.PREFERRED_SIZE, 174,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );


inputScrollPane.setBorder(javax.swing.BorderFactory.createTitled
Border(javax.swing.BorderFactory.createEtchedBorder(), "Input",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Tahoma", 1, 11), new java.awt.Color(0, 51, 102)));
// NOI18N


loadingsPanel.setBorder(javax.swing.BorderFactory.createTitledB
order(null, "Loadings",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Tahoma", 0, 11), new java.awt.Color(0, 51, 255)));
// NOI18N

        f2L.setText("F2:");

        f1L.setText("F1:");

        x1L.setText("X1:");

        x2L.setText("X2:");

        x1TF.addActionListener(new java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                x1TFActionPerformed(evt);
            }
        });

        x2TF.addActionListener(new java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                x2TFActionPerformed(evt);
            }
        });

        loadingsInstructionL.setFont(new java.awt.Font("Tahoma", 1,
12));
        loadingsInstructionL.setForeground(java.awt.Color.red);
        loadingsInstructionL.setText("! ALL LOADS ARE IN
NEWTON, WHILE ALL DISTANCES ARE IN MILLIMETRE");

        javax.swing.GroupLayout loadingsPanelLayout = new
javax.swing.GroupLayout(loadingsPanel);
        loadingsPanel.setLayout(loadingsPanelLayout);
        loadingsPanelLayout.setHorizontalGroup(

loadingsPanelLayout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)
            .addGroup(loadingsPanelLayout.createSequentialGroup()

.addGroup(loadingsPanelLayout.createParallelGroup(javax.swing.
GroupLayout.Alignment.LEADING)

.addGroup(loadingsPanelLayout.createSequentialGroup()
                    .addGap(114, 114, 114)

.addGroup(loadingsPanelLayout.createParallelGroup(javax.swing.
GroupLayout.Alignment.TRAILING)
                        .addComponent(f1L)
                        .addComponent(f2L))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)

.addGroup(loadingsPanelLayout.createParallelGroup(javax.swing.
GroupLayout.Alignment.LEADING)
                        .addComponent(f2TF,
javax.swing.GroupLayout.DEFAULT_SIZE, 181,
Short.MAX_VALUE)
                        .addComponent(f1TF,
javax.swing.GroupLayout.DEFAULT_SIZE, 181,
Short.MAX_VALUE))
                    .addGap(18, 18, 18)

.addGroup(loadingsPanelLayout.createParallelGroup(javax.swing.
GroupLayout.Alignment.LEADING)
                        .addComponent(x1L,
javax.swing.GroupLayout.Alignment.TRAILING)
                        .addComponent(x2L,
javax.swing.GroupLayout.Alignment.TRAILING))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)

.addGroup(loadingsPanelLayout.createParallelGroup(javax.swing.
GroupLayout.Alignment.LEADING, false)
                        .addComponent(x1TF)
```

```
                    .addComponent(x2TF,
javax.swing.GroupLayout.DEFAULT_SIZE, 189,
Short.MAX_VALUE))
                .addGap(37, 37, 37))

.addGroup(loadingsPanelLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(loadingsInstructionL)))
            .addContainerGap(158,
javax.swing.GroupLayout.PREFERRED_SIZE))
        );
        loadingsPanelLayout.setVerticalGroup(

loadingsPanelLayout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)
            .addGroup(loadingsPanelLayout.createSequentialGroup()
                .addComponent(loadingsInstructionL)
                .addGap(30, 30, 30)

.addGroup(loadingsPanelLayout.createParallelGroup(javax.swing.
GroupLayout.Alignment.BASELINE)
                .addComponent(x1L)
                .addComponent(f1L)
                .addComponent(x1TF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(f1TF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(11, 11, 11)

.addGroup(loadingsPanelLayout.createParallelGroup(javax.swing.
GroupLayout.Alignment.LEADING)

.addGroup(loadingsPanelLayout.createParallelGroup(javax.swing.
GroupLayout.Alignment.BASELINE)
                .addComponent(f2L)
                .addComponent(x2L)
                .addComponent(f2TF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addComponent(x2TF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(23, 23, 23))
        );

geometryPanel.setBorder(javax.swing.BorderFactory.createTitledB
order(null, "Geometry",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Tahoma", 0, 11), new java.awt.Color(0, 51, 255)));
// NOI18N

        aL.setText("a:");

        bL.setText("b:");

        lenL.setText("L (Span):");

        aTF.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                aTFActionPerformed(evt);
            }
        });

        lenTF.addActionListener(new java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                lenTFActionPerformed(evt);
            }
        });

        diaL.setText("Effective Diameter:");

        positionXL.setFont(new java.awt.Font("Tahoma", 1, 11));
        positionXL.setForeground(java.awt.Color.red);
        positionXL.setText("Position X:");

        diaTF.addActionListener(new java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                diaTFActionPerformed(evt);
            }
        });

        geometryInstructionL.setFont(new java.awt.Font("Tahoma",
1, 12));
        geometryInstructionL.setForeground(java.awt.Color.red);
        geometryInstructionL.setText("! ALL DISTANCES ARE IN
MILLIMETRE");

        javax.swing.GroupLayout geometryPanelLayout = new
javax.swing.GroupLayout(geometryPanel);
        geometryPanel.setLayout(geometryPanelLayout);
        geometryPanelLayout.setHorizontalGroup(

geometryPanelLayout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)
            .addGroup(geometryPanelLayout.createSequentialGroup()
                .addGap(56, 56, 56)

.addGroup(geometryPanelLayout.createParallelGroup(javax.swing.
GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
geometryPanelLayout.createSequentialGroup()
                .addComponent(diaL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                .addComponent(diaTF,
javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(40, 40, 40)
                .addComponent(positionXL)
                .addGap(18, 18, 18)
                .addComponent(positionXTF,
javax.swing.GroupLayout.PREFERRED_SIZE, 181,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(103, 103, 103))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
geometryPanelLayout.createSequentialGroup()
                .addComponent(aL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                .addComponent(aTF,
javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(39, 39, 39)
                .addComponent(bL)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                .addComponent(bTF,
javax.swing.GroupLayout.PREFERRED_SIZE, 164,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(54, 54, 54)
                .addComponent(lenL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                .addComponent(lenTF,
javax.swing.GroupLayout.PREFERRED_SIZE, 158,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(68, 68, 68))))
        .addGroup(geometryPanelLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(geometryInstructionL))
    );
    geometryPanelLayout.setVerticalGroup(

geometryPanelLayout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)
        .addGroup(geometryPanelLayout.createSequentialGroup()
            .addComponent(geometryInstructionL)
            .addGap(18, 18, 18)

.addGroup(geometryPanelLayout.createParallelGroup(javax.swing.
GroupLayout.Alignment.BASELINE)
                .addComponent(aL)
                .addComponent(aTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(lenTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(lenL)
                .addComponent(bTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(bL))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED, 19, Short.MAX_VALUE)

.addGroup(geometryPanelLayout.createParallelGroup(javax.swing.
GroupLayout.Alignment.BASELINE)
                .addComponent(positionXL)
                .addComponent(diaTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(diaL)
                .addComponent(positionXTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap())
    );

MatPropPanel.setBorder(javax.swing.BorderFactory.createTitledB
order(null, "Material Property",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Tahoma", 0, 11), new java.awt.Color(0, 51, 255)));
// NOI18N
        youngL.setText("Modulus of Rigidity E (N/mm^2 or MPa) :");

        inertiaL.setText("Area Moment of Inertia I (mm^4):");

        inertiaTF.setEditable(false);

        javax.swing.GroupLayout MatPropPanelLayout = new
javax.swing.GroupLayout(MatPropPanel);
        MatPropPanel.setLayout(MatPropPanelLayout);
        MatPropPanelLayout.setHorizontalGroup(

MatPropPanelLayout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)
            .addGroup(MatPropPanelLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(youngL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                .addComponent(youngTF,
javax.swing.GroupLayout.PREFERRED_SIZE, 179,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                .addComponent(inertiaL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
UNRELATED)
                .addComponent(inertiaTF,
javax.swing.GroupLayout.PREFERRED_SIZE, 167,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        MatPropPanelLayout.setVerticalGroup(

MatPropPanelLayout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
MatPropPanelLayout.createSequentialGroup()
                .addContainerGap(12, Short.MAX_VALUE)

.addGroup(MatPropPanelLayout.createParallelGroup(javax.swing.
GroupLayout.Alignment.BASELINE)
                    .addComponent(youngL)
                    .addComponent(youngTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(inertiaL)
                    .addComponent(inertiaTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap())
        );

        javax.swing.GroupLayout inputPanelLayout = new
javax.swing.GroupLayout(inputPanel);
        inputPanel.setLayout(inputPanelLayout);
        inputPanelLayout.setHorizontalGroup(

inputPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
            .addGroup(inputPanelLayout.createSequentialGroup()
                .addContainerGap()
```

```
.addGroup(inputPanelLayout.createParallelGroup(javax.swing.Gro
upLayout.Alignment.LEADING)
                .addComponent(MatPropPanel,
javax.swing.GroupLayout.PREFERRED_SIZE, 749,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(inputPanelLayout.createParallelGroup(javax.swing.Gro
upLayout.Alignment.TRAILING, false)
                .addComponent(loadingsPanel,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(geometryPanel,
javax.swing.GroupLayout.PREFERRED_SIZE, 749,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap(13, Short.MAX_VALUE))
        );
        inputPanelLayout.setVerticalGroup(

inputPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
inputPanelLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(loadingsPanel,
javax.swing.GroupLayout.PREFERRED_SIZE, 146,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
            .addComponent(geometryPanel,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
            .addComponent(MatPropPanel,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(242, 242, 242))
        );

        inputScrollPane.setViewportView(inputPanel);


outputPanel.setBorder(javax.swing.BorderFactory.createTitledBor
der(javax.swing.BorderFactory.createEtchedBorder(), "Analysis:
Output",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Tahoma", 1, 11), new java.awt.Color(0, 0, 102))); //
NOI18N

        deflXL.setText("Deflection Y (mm) at position X :");

        momXL.setText("Bending Moment BM (Nmm) at position X
:");

        slpXL.setText("Slope (radian) at position X:");

        reacAL.setText("Reaction A (Newton) :");

        reacBL.setText("Reaction B (Newton) :");

        reacBTF.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                reacBTFActionPerformed(evt);
            }
        });

        slpAL.setText("Slolpe A (radian) :");

        slpBL.setText("Slope B (radian) :");

        reacCL.setText("Reaction C (Newton) :");

        reacCTF.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                reacCTFActionPerformed(evt);
            }
        });

        slpCL.setText("Slope C (radian) :");

        javax.swing.GroupLayout outputPanelLayout = new
javax.swing.GroupLayout(outputPanel);
        outputPanel.setLayout(outputPanelLayout);
        outputPanelLayout.setHorizontalGroup(

outputPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
            .addGroup(outputPanelLayout.createSequentialGroup()
                .addGap(23, 23, 23)

.addGroup(outputPanelLayout.createParallelGroup(javax.swing.Gr
oupLayout.Alignment.LEADING, false)

.addGroup(outputPanelLayout.createSequentialGroup()
                    .addComponent(slpXL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                    .addComponent(slpXTF,
javax.swing.GroupLayout.PREFERRED_SIZE, 158,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(outputPanelLayout.createSequentialGroup()

.addGroup(outputPanelLayout.createParallelGroup(javax.swing.Gr
oupLayout.Alignment.LEADING, false)

.addGroup(outputPanelLayout.createSequentialGroup()
                        .addComponent(reacAL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                        .addComponent(reacATF,
javax.swing.GroupLayout.PREFERRED_SIZE, 141,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(outputPanelLayout.createSequentialGroup()
                        .addComponent(slpAL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                        .addComponent(slpATF)))
                    .addGap(10, 10, 10)

.addGroup(outputPanelLayout.createParallelGroup(javax.swing.Gr
oupLayout.Alignment.LEADING)
```

```
.addGroup(outputPanelLayout.createSequentialGroup()
                        .addComponent(reacBL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                        .addComponent(reacBTF,
javax.swing.GroupLayout.PREFERRED_SIZE, 125,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
UNRELATED)
                        .addComponent(reacCL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                        .addComponent(reacCTF,
javax.swing.GroupLayout.DEFAULT_SIZE, 137,
Short.MAX_VALUE))

.addGroup(outputPanelLayout.createSequentialGroup()
                        .addComponent(slpBL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                        .addComponent(slpBTF,
javax.swing.GroupLayout.PREFERRED_SIZE, 153,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(slpCL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                        .addComponent(slpCTF))))

.addGroup(outputPanelLayout.createSequentialGroup()
                    .addComponent(deflXL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                    .addComponent(deflXTF,
javax.swing.GroupLayout.PREFERRED_SIZE, 177,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                    .addComponent(momXL)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                    .addComponent(momXTF,
javax.swing.GroupLayout.PREFERRED_SIZE, 157,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap())
        );
        outputPanelLayout.setVerticalGroup(

outputPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
            .addGroup(outputPanelLayout.createSequentialGroup()
                .addContainerGap()

.addGroup(outputPanelLayout.createParallelGroup(javax.swing.Gr
oupLayout.Alignment.BASELINE, false)
                    .addComponent(deflXL)
                    .addComponent(deflXTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(momXL)
                    .addComponent(momXTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(11, 11, 11)

.addGroup(outputPanelLayout.createParallelGroup(javax.swing.Gr
oupLayout.Alignment.BASELINE)
                    .addComponent(slpXL)
                    .addComponent(slpXTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
UNRELATED)

.addGroup(outputPanelLayout.createParallelGroup(javax.swing.Gr
oupLayout.Alignment.BASELINE, false)
                    .addComponent(reacAL)
                    .addComponent(reacATF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(reacBL)
                    .addComponent(reacBTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(reacCL)
                    .addComponent(reacCTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)

.addGroup(outputPanelLayout.createParallelGroup(javax.swing.Gr
oupLayout.Alignment.BASELINE)
                    .addComponent(slpAL)
                    .addComponent(slpBL)
                    .addComponent(slpBTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(slpCL)
                    .addComponent(slpCTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(slpATF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(25, 25, 25))
        );

        calculate.setText("Calculate");
        calculate.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                calculateActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(this);
        this.setLayout(layout);
```

```
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.A
lignment.LEADING)
                    .addComponent(banner,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(inputScrollPane,
javax.swing.GroupLayout.DEFAULT_SIZE, 800,
Short.MAX_VALUE)
                    .addComponent(calculate,
javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(outputPanel,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap())
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(banner,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                .addComponent(inputScrollPane,
javax.swing.GroupLayout.PREFERRED_SIZE, 205,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                .addComponent(calculate)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED)
                .addComponent(outputPanel,
javax.swing.GroupLayout.PREFERRED_SIZE, 170,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
    }// </editor-fold>//GEN-END:initComponents

    private void
calculateActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_calculateActionPerformed

        //declaration of variables

        three = new ThreeSupport();     // instance of the class
ThreeSupport.
        double f1, f2;      // forces acting on the beam in N
        double x1,x2;       // position of the respective forces in mm
        double a,b;         // distance between supports in mm
        double l;       // span of the beam in mm
        double d;       // diameter of shaft in mm
        double x; // position x along the beam in mm
        double E;   // young's modulus in N/mm^2

        // getting the inputs from the respective JTextField and
initializing the
        // beam object variables

        f1 = Double.parseDouble(f1TF.getText());
        three.setF1(f1);
        f2 = Double.parseDouble(f2TF.getText());
        three.setF2(f2);
        x1 = Double.parseDouble(x1TF.getText());
        three.setX1(x1);
        x2 = Double.parseDouble(x2TF.getText());
        three.setX2(x2);
        a = Double.parseDouble(aTF.getText());
        three.setA(a);
        b = Double.parseDouble(bTF.getText());
        three.setB(b);
        l = Double.parseDouble(lenTF.getText());
        three.setL(l);
        d = Double.parseDouble(diaTF.getText());
        three.setD(d);
        x = Double.parseDouble(positionXTF.getText());
        three.setX(x);
        E = Double.parseDouble(youngTF.getText());
        three.setE(E);

        // calculations
        inertiaTF.setText(String.valueOf(three.getI()));    //area
moment of inertia

        reacATF.setText(String.valueOf(three.getR1()));

        reacBTF.setText(String.valueOf(three.getR2()));

        reacCTF.setText(String.valueOf(three.getR3()));

        //analysis
        deflXTF.setText(String.valueOf(three.getY(x))); // deflection
along any pont x

        momXTF.setText(String.valueOf(three.getMomentX(x)));
//bending moment along any point x

        slpXTF.setText(String.valueOf(three.getSlopeX(x)));     //
slope along any point x

        slpATF.setText(String.valueOf(three.getSlopeX(0)));
//slope at the left support

        slpBTF.setText(String.valueOf(three.getSlopeX(a)));     //
slope at middle support

        slpCTF.setText(String.valueOf(three.getSlopeX(l)));    // slope
at the right support

    }//GEN-LAST:event_calculateActionPerformed

    private void
reacBTFActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_reacBTFActionPerformed
        // TODO add your handling code here:
}//GEN-LAST:event_reacBTFActionPerformed

    private void x1TFActionPerformed(java.awt.event.ActionEvent
evt) {//GEN-FIRST:event_x1TFActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_x1TFActionPerformed
```

```
    private void x2TFActionPerformed(java.awt.event.ActionEvent
evt) {//GEN-FIRST:event_x2TFActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_x2TFActionPerformed

    private void aTFActionPerformed(java.awt.event.ActionEvent
evt) {//GEN-FIRST:event_aTFActionPerformed
        // TODO add your handling code here:
}//GEN-LAST:event_aTFActionPerformed

    private void diaTFActionPerformed(java.awt.event.ActionEvent
evt) {//GEN-FIRST:event_diaTFActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_diaTFActionPerformed

    private void lenTFActionPerformed(java.awt.event.ActionEvent
evt) {//GEN-FIRST:event_lenTFActionPerformed
        // TODO add your handling code here:
}//GEN-LAST:event_lenTFActionPerformed

    private void
reacCTFActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_reacCTFActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_reacCTFActionPerformed

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JPanel MatPropPanel;
    private javax.swing.JLabel aL;
    private javax.swing.JTextField aTF;
    private javax.swing.JLabel bL;
    private javax.swing.JTextField bTF;
    private javax.swing.JPanel banner;
    private javax.swing.JButton calculate;
    private javax.swing.JLabel deflXL;
    private javax.swing.JTextField deflXTF;
    private javax.swing.JLabel diaL;
    private javax.swing.JTextField diaTF;
    private javax.swing.JLabel f1L;
    private javax.swing.JTextField f1TF;
    private javax.swing.JLabel f2L;
    private javax.swing.JTextField f2TF;
    private javax.swing.JLabel geometryInstructionL;
    private javax.swing.JPanel geometryPanel;
    private javax.swing.JLabel imageL;
    private javax.swing.JLabel inertiaL;
    private javax.swing.JTextField inertiaTF;
    private javax.swing.JPanel inputPanel;
    private javax.swing.JScrollPane inputScrollPane;
    private javax.swing.JLabel lenL;
    private javax.swing.JTextField lenTF;
    private javax.swing.JLabel loadingsInstructionL;
    private javax.swing.JPanel loadingsPanel;
    private javax.swing.JLabel momXL;
    private javax.swing.JTextField momXTF;
    private javax.swing.JPanel outputPanel;
    private javax.swing.JLabel positionXL;
    private javax.swing.JTextField positionXTF;
    private javax.swing.JLabel reacAL;
    private javax.swing.JTextField reacATF;
    private javax.swing.JLabel reacBL;
    private javax.swing.JTextField reacBTF;
    private javax.swing.JLabel reacCL;
    private javax.swing.JTextField reacCTF;
    private javax.swing.JLabel slpAL;
    private javax.swing.JTextField slpATF;
    private javax.swing.JLabel slpBL;
    private javax.swing.JTextField slpBTF;
    private javax.swing.JLabel slpCL;
    private javax.swing.JTextField slpCTF;
    private javax.swing.JLabel slpXL;
    private javax.swing.JTextField slpXTF;
    private javax.swing.JLabel x1L;
    private javax.swing.JTextField x1TF;
    private javax.swing.JLabel x2L;
    private javax.swing.JTextField x2TF;
    private javax.swing.JLabel youngL;
    private javax.swing.JTextField youngTF;
    // End of variables declaration//GEN-END:variables

}
```

REFERENCES

[1] R. H. Bannister, "Lecture Notes on Machine Dynamics", Parts I & II, Department of Turbo-machinery ands Engineering Mechanics, Cranfield University, England, 1992.

[2] C. K. Mischke, (2004). *Shafts*, ch. 17, pp. 17.1-17.21, Available: www.digitalengineeringlibrary.com

[3] R. C. Juvinal, K. M. Marshek, *Fundamentals of Machine Component Design*, 2nd ed., John Wiley, 1991, pp. 790-792

[4] I. H. Shames, *Introduction to Solid Mechanics*, 2nd ed., Prentice Hall, New Delhi, 1990, ch. 12, pp. 369-395

[5] J. Den Hartog,, Strength of Materials, Dover, 1961

[6] E. P. Popov, S. Nagarajan, Z. A. Lu, *Mechanics of Materials*, 2nd ed., Prentice Hall, New Jersey, 1976.

[7] A. D. Deutschman, W. J. Michels, C. E. Wilson, Machine Design, Macmillan, New York, 1975, pp. 236-238

[8] T. K. Jack, "Lecture Notes on Strength of Materials II," Department of Mechanical Engineering, Rivers State University of Science and Technology, Port Harcourt, 2006.

[9] G. Jombo, "Computer Program for Multi-bearing Shaft Deflection Calculation and Analysis by Patch and Domain Double Integration Method", B.Tech Degree, Final Year Project Report, Department of Mechanical Engineering, Rivers State University of Science and Technology, Port Harcourt, 2008

[10] W. Ker Wilson, *Practical Solutions of Torsional Vibration Problems*, 3rd ed., John Wiley, New York, 1956, ch. XI, pp. 562-576

[11] E. Currie, *Fundamentals of Programming using Java*, Thomson, 2006

[12] http://en.wikipedia.org.wiki/Unified_Modelling_Lamguage

[13] http://en.wikipedia.org.wiki/Class_Diagrams

[14] V. M. Ezeh, "Computer Assisted Helical Gear Design and Analysis", B.Tech Degree, Final Year Project Report, Department of Mechanical Engineering, Rivers State University of Science and Technology, Port Harcourt, 2008

[15] J. Cowell, *Essential Java Fast*, Springer, 1997

[16] P. Mcbride, *Java Made Simple*, Butterworth-Heinemann, 1997

[17] D. Flanagan, *JavaScript Pocket Reference*, O'reilly, 1998

[18] S. D. Gathman, S. D., "A Text UI for Java AWT, Designing User Interfaces", *Dr. Dobb's Journal*, Sept. 1997.

Tonye K. Jack is a Registered Engineer, and ASME member. He worked on plant maintenance and rotating equipment in the Chemical Fertilizer industry, and on gas turbines in the oil and gas industry. He has Bachelors degree in Mechanical Engineering from the University of Nigeria, and Masters Degrees in Engineering Management from the University of Port Harcourt, and in Rotating Machines Design from the Cranfield University in England. He is a University Teacher in Port Harcourt, Rivers State, Nigeria, teaching undergraduate classes in mechanical engineering. His research interests are on rotating equipment engineering, maintenance, engineering management, engineering computer programs, and applied mechanics.

Gbanaibolou Jombo – Graduated with a Bachelors Degree in Mechanical Engineering from the River State University of Science and Technology in 2008. He is currently studying for the Masters Degree in Design of Rotating Machines at the Cranfield University in England.