Error Propagation in a Simple Function Evaluation

Sean Rodby, Luke Snyder, Autar Kaw

University of South Florida

United States of America

kaw@eng.usf.edu

Website: http://numericalmethods.eng.usf.edu

Version: Maple 12; Last Revised: August 28, 2008

Introduction

The following worksheet illustrates how error propagates in a simple function evaluation. This is due largely to the fact that relative error calculations are based on round-off errors and/or measurement errors that exist inherently in simple devices used to take measurements. The user will enter force, length or width of the cross section, and Young's Modulus data in the *Input* section of the program, as well as the corresponding relative measurement error for each input. From here, Maple will be used to demonstrate how this error propagates. The formula used for illustration is Strain = $\frac{F}{h^2 E}$.

Initialization

```
> restart;
```

Section 1: Input

This is the only section where the user interacts with the program.

Enter inputs to calculate strain based on the formula Strain = $\frac{F}{h^2 E}$.

Enter Force (in N)

```
> Force := 72;
```

$$Force := 72 \tag{3.1}$$

Enter relative measurement error in percentage for force:

```
> RME_Force := 2.5;
```

$$RME\_Force := 2.5 \tag{3.2}$$

Enter length or width of cross section (in m):

```
> h := 4e-3;
```

$$h := 0.004 \tag{3.3}$$

Enter relative measurement error in percentage for h:

```
> RME_h := 2.5;
```

$$RME\_h := 2.5 \tag{3.4}$$

Enter Young's Modulus (in Pa):

> $E := 70e9;$

$$E := 7.0\ 10^{10} \quad (3.5)$$

Enter relative measurement error in percentage for Modulus:

> $RME\_Young := 2.5;$

$$RME\_Young := 2.5 \quad (3.6)$$

This is the end of the user section. All information must be entered before proceeding to the next section. Re-execute the program.

Section 2: Procedure

Calculating the measurement error for each input.

```
> Meas_Error_Force := Force * (RME_Force/100);
  Meas_Error_h := h * (RME_h/100);
  Meas_Error_Young := E * (RME_Young/100);
```

$$Meas\_Error\_Force := 1.800000000$$

$$Meas\_Error\_h := 0.0001000000000$$

$$Meas\_Error\_Young := 1.750000000\ 10^{9} \quad (4.1)$$

First we must calculate the strain given by the equation $\varepsilon = \frac{F}{h^2 E}$.

> $strain := \frac{Force}{h^2\ E}$

$$strain := 0.00006428571429 \quad (4.2)$$

Using the formula for maximum possible error to derive the maximum possible error in the measured strain. Note that different variable names are used because the previous variable names represented actual numbers. Meaning that if the partial differential is taken, in all cases it will be zero. The subs command is then used to input the values given by the user into the partial differential equation.

```
> strain1 := F/((h1^2)*E1);
  partial_wrt_Force := diff(strain1, F);
  a := subs(E1 = E, partial_wrt_Force) :
  partial_wrt_Force := abs(subs(h1 = h, a));
  partial_wrt_h := diff(strain1, h1);
  a := subs(h1 = h, partial_wrt_h) :
  b := subs(E1 = E, a) :
  partial_wrt_h := abs(subs(F = Force, b));
  partial_wrt_Young := diff(strain1, E1);
  a := subs(E1 = E, partial_wrt_Young) :
  b := subs(h1 = h, a) :
  partial_wrt_Young := abs(subs(F = Force, b));
```

$$strain1 := \frac{F}{h1^2\,E1}$$

$$partial\_wrt\_Force := \frac{1}{h1^2\,E1}$$

$$partial\_wrt\_Force := 8.928571431\;10^{-7}$$

$$partial\_wrt\_h := -\frac{2\,F}{h1^3\,E1}$$

$$partial\_wrt\_h := 0.03214285716$$

$$partial\_wrt\_Young := -\frac{F}{h1^2\,E1^2}$$

$$partial\_wrt\_Young := 9.183673469\;10^{-16} \tag{4.3}$$

Calculating the range of value for each measured quantity based in the RME percentages given.

```
> Range_Force := (RME_Force/100) * Force;
  Range_h := (RME_h/100) * h;
  Range_Young := (RME_Young/100) * E;
```

$$Range\_Force := 1.800000000$$

$$Range\_h := 0.0001000000000$$

$$Range\_Young := 1.750000000\;10^{9} \tag{4.4}$$

Calculating the maximum possible error by adding the absolute value of the partial differential equations multiplied by their respective range of variation in the measured value. Then calculating the individual error contributions due to Force, Area, and the length or width dimension respectively.

```
> # With respect to force:
  Error_Force := partial_wrt_Force·Range_Force;

  # With respect to h :
  Error_h := partial_wrt_h * Range_h;

  # With respect to Young's Modulus :
  Error_Young := partial_wrt_Young * Range_Young;

  # Calculating total error by adding all the individual contributions given
      by the user defined inputs.
  Tot_Error := (Error_Force + Error_h + Error_Young);

  # Calculating the percent contribution of force error to the total error.
  Rel_Force_Error := Error_Force·100 / Tot_Error;

  # Calculating the percent contribution of h error to the total error.
  Rel_h_Error := (Error_h/Tot_Error) * 100;

   # Calculating the percent contribution of Young's Modulus error to total error
```

$$Rel\_Young\_Error := \frac{Error\_Young \cdot 100}{Tot\_Error};$$

Calculating the effective range that the axial strain could be within using the total error calculation.

$Low\_strain := (strain - Tot\_Error):$

$High\_strain := (strain + Tot\_Error):$

Calculating the maximum percent relative measurement error for strain.

$$RME\_strain := \frac{100\ Tot\_Error}{strain}:$$

```
printf("\nThe nominal strain calculated at the nominal value for all inputs is, strain = %g.", strain);

printf("\nMinimum value of strain = %g.", Low_strain);
printf("\nMaximum value of strain = %g.", High_strain);
printf("\nThe maximum measurement error for strain is RME_strain = %g %%.", RME_strain);
```

$$Error\_Force := 0.000001607142858$$

$$Error\_h := 0.000003214285716$$

$$Error\_Young := 0.000001607142857$$

$$Tot\_Error := 0.000006428571431$$

$$Rel\_Force\_Error := 25.00000000$$

$$Rel\_h\_Error := 50.00000001$$

$$Rel\_Young\_Error := 24.99999999$$

```
The nominal strain calculated at the nominal value for all inputs is,
strain = 6.42857e-05.
Minimum value of strain = 5.78571e-05.
Maximum value of strain = 7.07143e-05.
The maximum measurement error for strain is RME_strain = 10 %.
```

Conclusion

If a calculation is made with numbers that are not exact, then the calculation itself will have an error. Since the final results of an experiment are not usually directly measured but are some function of one or more of the measured quantities, it is important to understand and utilize propagation of error concepts to better interrpret and represent experimental results.

References

Propagation of Errors.

See: http://numericalmethods.eng.usf.edu/mws/gen/01aae/mws_gen_aae_txt_propagationoferrors.pdf