

# Second Order Derivatives Approximation

Ana Catalina Torres, Autar Kaw

University of South Florida

United States of America

kaw@eng.usf.edu

## Introduction

This worksheet demonstrates the use of Maple to illustrate the approximation of the second order derivative of continuous functions.

A second order derivative approximation uses a point  $h$  ahead and a point  $h$  behind of the given value of  $x$  at which the second derivative of  $f(x)$  is to be found.

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

## Initialization

```
> restart;  
with (plots) :
```

## Section 1: Input

The following simulation approximates the second derivative of a function using Second Order Derivatives Approximation. The user inputs are

- function,  $f(x)$
- point at which the derivative is to be found,  $xv$
- starting step size,  $h$
- number of times user wants to halve the step size,  $n$

The outputs include

- approximate value of the second derivative at the point and initial step size given
- exact value
- true error, absolute relative true error, approximate error and absolute relative approximate error, number of at least correct significant digits in the solution as a function of step size.

[Function  $f(x)$ .

```
> f := x -> exp(2*x);
```

$$f := x \rightarrow e^{2x} \quad (3.1)$$

[Value of  $x$  at which  $f''(x)$  is desired,  $xv$

```
> xv := 4.0;
```

$$xv := 4.0 \quad (3.2)$$

[Starting step size,  $h$

```
> h := 0.2;
                                     h := 0.2                                (3.3)
```

Number of times step size is halved

```
> n := 6;
                                     n := 6                                (3.4)
```

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

## Section 2: Procedure

The following procedure estimates the solution of second derivate of an equation at a point  $xv$ .

$f(x)$  = function

$xv$  = value at which the solution is desired

$h$  = step size value

$n$  = number of times step size is halved

```
> SOD := proc(f, xv, h)
  local deriv :
  deriv :=  $\frac{f(xv+h) - 2 \cdot f(xv) + f(xv-h))}{h^2}$  :
  return (deriv) :
end proc:
```

## Section 3: Calculation

The exact value  $Ev$  of the first derivative of the equation:

First, using the *diff* command the solution is found. In a second step, the exact value of the derivative is shown.

```
> y(x) = f(x);
                                     y(x) = e2x                                (5.1)
```

```
> Soln := diff(diff(f(x), x), x);
                                     Soln := 4 e2x                                (5.2)
```

```
> Ev := evalf(subs(x = xv, Soln));
                                     Ev := 11923.83195                                (5.3)
```

The next loop calculates the following:

Av: Approximate value of the second derivative using Second Order Derivatives Approximation by calling the procedure "SOD"

Ev: Exact value of the second derivative

Et: True Error

et: Absolute relative true percentage error

Ea: Approximate Error

ea: Absolute relative approximate percentage error

Sig: Least number of correct significant digits in an approximation

```
> for i from 0 by 1 to n-1 do
  N[i] := 2i :
```

```

H[i] :=  $\frac{h}{N[i]}$  :
Av[i] := SOD(f, xv, H[i]) :
Et[i] := Ev - Av[i] :
et[i] := abs( $\frac{Et[i]}{Ev}$ ) * 100 :
if (i > 0) then
  Ea[i] := Av[i] - Av[i - 1] :
  ea[i] := abs( $\frac{Ea[i]}{Av[i]}$ ) * 100 :
  Sig[i] := floor( $2 - \log_{10}\left(\frac{ea[i]}{0.5}\right)$ ) :
  if Sig[i] < 0 then
    Sig[i] := 0 :
  end if:
end if:
end do:

```

The loop halves the value of the step size  $n$  times. Each time, the approximate value of the second derivative is calculated and saved in a vector. The approximate error is calculated after at least two approximate values of the second derivative have been saved. The number of significant digits is calculated and written as the lowest real number. If the number of significant digits calculated is less than zero, then is shown as zero.

## Section 4: Spreadsheet

The next table shows the step size value, approximate value, true error, the absolute relative true percentage error, the approximate error, the absolute relative approximate percentage error and the least number of correct significant digits in an approximation as a function of the step size value.

```

> with(Spread) :
  tableoutput := CreateSpreadsheet("Second Order Derivative") :
  SetCellFormula(tableoutput, 1, 2, "Step Size") :
  SetCellFormula(tableoutput, 1, 3, "Approx Value") :
  SetCellFormula(tableoutput, 1, 4, "True Error") :
  SetCellFormula(tableoutput, 1, 5, "Abs Rel True Error") :
  SetCellFormula(tableoutput, 1, 6, "Approx error") :
  SetCellFormula(tableoutput, 1, 7, "Abs Rel Approx Error") :
  SetCellFormula(tableoutput, 1, 8, "Sig Digits");
for i from 0 by 1 to n-1 do
  SetCellFormula(tableoutput, i + 2, 1, i + 1) :
  SetCellFormula(tableoutput, i + 2, 2, evalf(H[i])) :
  SetCellFormula(tableoutput, i + 2, 3, evalf(Av[i])) :
  SetCellFormula(tableoutput, i + 2, 4, evalf(Et[i])) :
  SetCellFormula(tableoutput, i + 2, 5, evalf(et[i])) :
  SetCellFormula(tableoutput, i + 2, 6, evalf(Ea[i])) :

```

```

SetCellFormula(tableoutput, i+2, 7, evalf(ea[i])) :
SetCellFormula(tableoutput, i+2, 8, evalf(Sig[i]));
end do:
EvaluateSpreadsheet(tableoutput) :

```

| Second Order Derivative |     |                |                |              |                      |                |                        |              |   |
|-------------------------|-----|----------------|----------------|--------------|----------------------|----------------|------------------------|--------------|---|
|                         | ... | B              | C              | D            | E                    | F              | G                      | H            | I |
| 1                       |     | "Step Size"    | "Approx Value" | "True Error" | "Abs Rel True Error" | "Approx error" | "Abs Rel Approx Error" | "Sig Digits" |   |
| 2                       | 1   | 0.2            | 12083.66672    | -159.83477   | 1.340464799          | $Ea_0$         | $ea_0$                 | $Sig_0$      |   |
| 3                       | 2   | 0.1000000000   | 11963.63110    | -39.79915    | 0.3337781861         | -120.03562     | 1.003337691            | 1.           |   |
| 4                       | 3   | 0.05000000000  | 11933.77160    | -9.93965     | 0.08335952772        | -29.85950      | 0.2502100845           | 2.           |   |
| 5                       | 4   | 0.02500000000  | 11926.31520    | -2.48325     | 0.02082593927        | -7.45640       | 0.06252056796          | 2.           |   |
| 6                       | 5   | 0.01250000000  | 11924.45440    | -.62245      | 0.005220217818       | -1.86080       | 0.01560490684          | 3.           |   |
| 7                       | 6   | 0.006250000000 | 11923.99360    | -.16165      | 0.001355688345       | -.46080        | 0.003864477083         | 4.           |   |
| 8                       |     |                |                |              |                      |                |                        |              |   |

(6.1)

## Section 5: Graphs

The following graphs show the approximate solution, absolute relative true error, absolute relative approximate error and least number of significant digits as a function of step size.

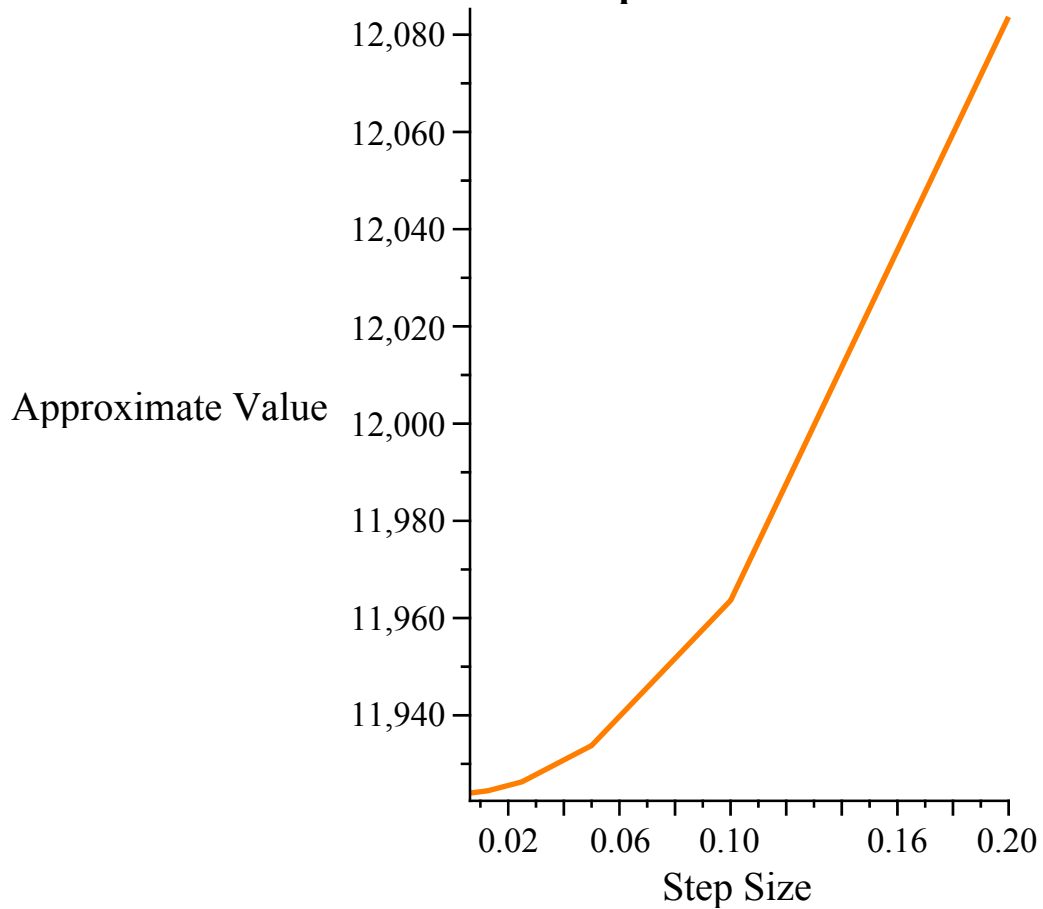
```

> data := [seq([H[i], Av[i]], i = 0 .. n - 1)]:
plot(data, x = H[0] .. H[n - 1], color = coral, thickness = 2, title
= "Approximate Solution of the Second Derivative of a
Function using Second Order Derivatives Approximation as a
Function of Step Size", labels = ["Step Size",
"Approximate Value"], titlefont = [TIMES, BOLD, 12], labelfont
= [TIMES, ROMAN, 12]);
data := [seq([H[i], et[i]], i = 0 .. n - 1)]:
plot(data, x = H[0] .. H[n - 1], color = magenta, thickness = 2, title
= "Absolute Relative True Percentage Error as a Function of
Step Size", labels = ["Step Size", "Absolute Relative
True Error"], titlefont = [TIMES, BOLD, 12], labelfont = [TIMES,
ROMAN, 12]);
data := [seq([H[i], ea[i]], i = 0 .. n - 1)]:
plot(data, x = H[0] .. H[n - 1], color = green, thickness = 2, title
= "Absolute Relative Approximate Percentage Error as a
Function of Step Size", labels = ["Step Size", "Absolute
Relative
Approximate Error "], titlefont = [TIMES, BOLD, 12], labelfont
= [TIMES, ROMAN, 12]);
data := [seq([H[i], Sig[i]], i = 0 .. n - 1)]:
plot(data, x = H[0] .. H[n - 1], color = navy, thickness = 2, title
= "Least Significant Digits Correct as a Function of Step

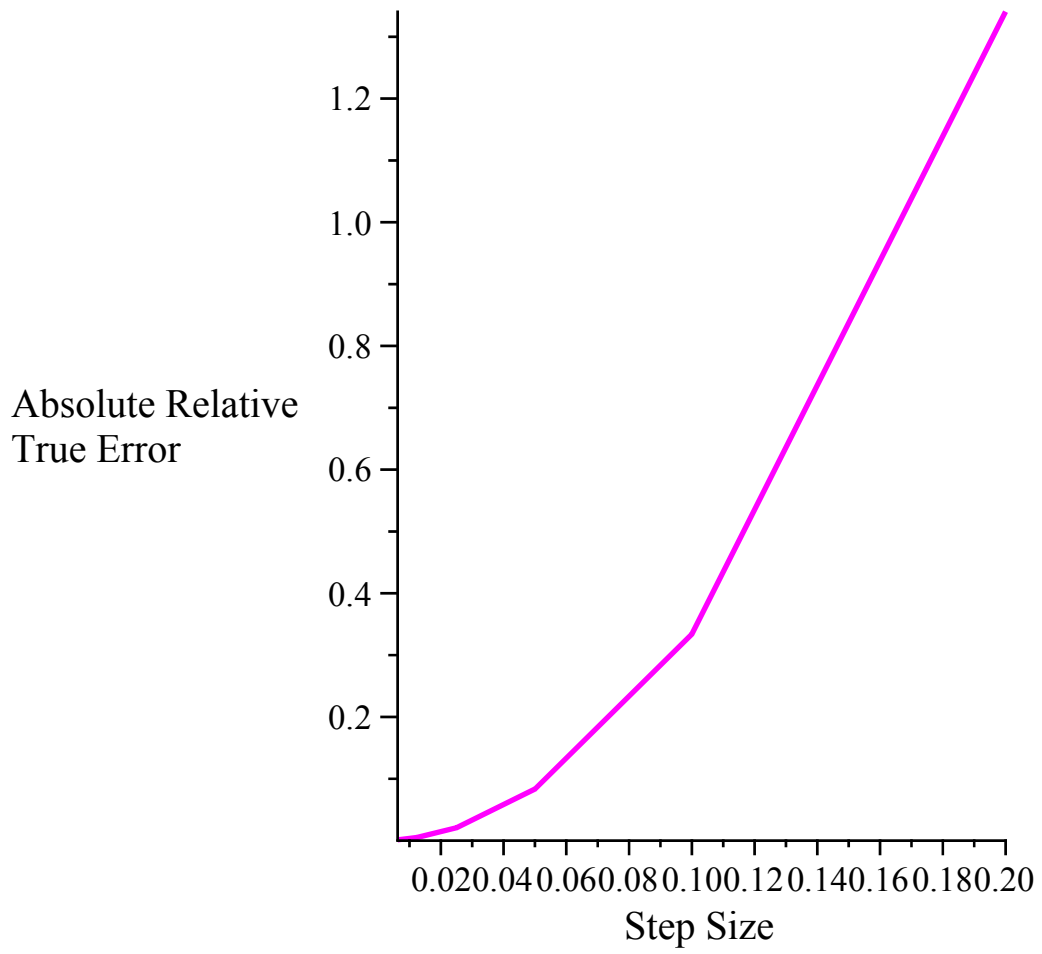
```

```
Size", labels=["Step size", "Least number of  
significant digits"], titlefont=[TIMES, BOLD, 12], labelfont  
=[TIMES, ROMAN, 12]);
```

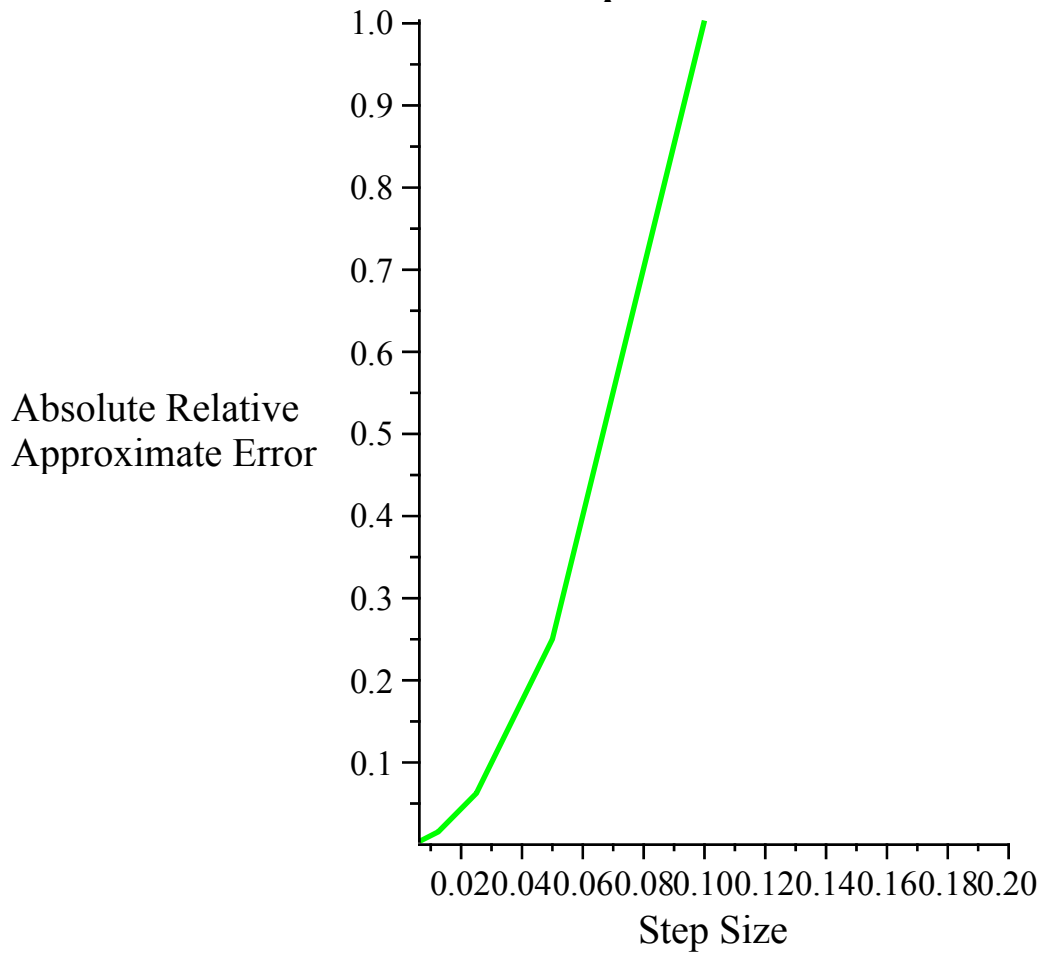
**Approximate Solution of the Second Derivative of a  
Function using Second Order Derivatives Approximation as  
a Function of Step Size**



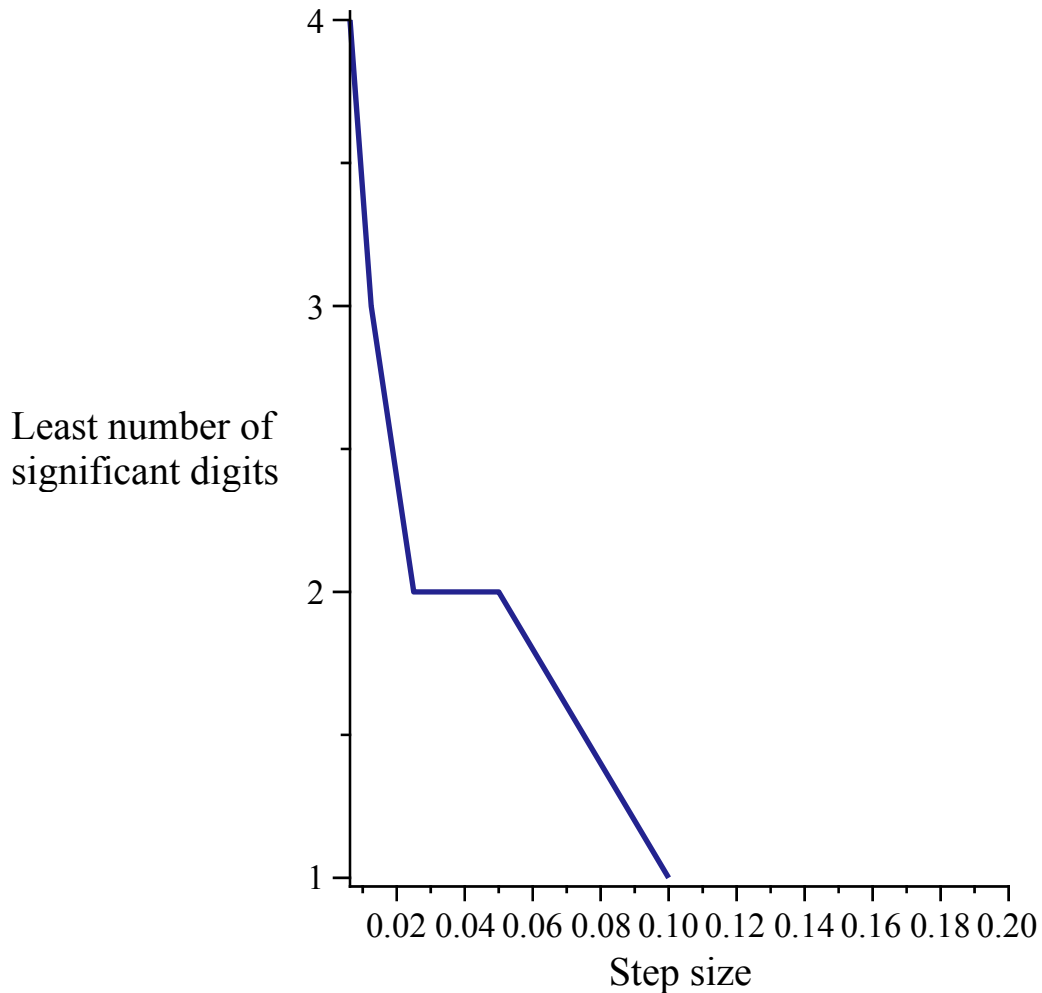
## Absolute Relative True Percentage Error as a Function of Step Size



### Absolute Relative Approximate Percentage Error as a Function of Step Size



## Least Significant Digits Correct as a Function of Step Size



>

## References

Numerical Differentiation of Continuous Functions.  
See <http://numericalmethods.eng.usf.edu/mws/gen/0.2def/>

## Questions

1. The velocity of a rocket is given by

$$v(t) = 2000 \cdot \ln \frac{140000}{140000 - 2100 t} - 9.8 t$$

Use second order derivative approximation method with a step size of 0.25 to find the jerk at  $t=5s$ . Compare with the exact answer and study the effect of the step size.

2. Look at the true error vs. step size data for problem 1. Do you see a relationship between the value of the true error and step size? Is this coincidental?

## Conclusions

To obtain more accurate values of the second derivative using Second Order Derivative Approximation, the step size needs to be small. As the spreadsheet shows, the smaller the step size



value is, the approximation is closest to the exact value. By decreasing the step size, the least number of significant digits that can be trusted increases.

*Legal Notice: The copyright for this application is owned by the author(s). Neither Maplesoft nor the author are responsible for any errors contained within and are not liable for any damages resulting from the use of this material. This application is intended for non-commercial, non-profit use only. Contact the author for permission if you wish to use this application for-profit activities.*