Differentiation of Continuous Functions

Backward Difference Approximation of the First Derivative

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 Backward Difference Approximation of the first derivative of continuous functions.

Backward Difference Approximation of the first derivative uses a point *h* behind of the given value of *x* at which the derivative of *f(x)* is to be found.

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

Initialization

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

Section 1: Input

The following simulation approximates the first derivative of a function using Backward Difference Approximation. The user inputs are

a) function, *f(x)*
b) point at which the derivative is to be found, *xv*
c) starting step size, *h*
d) number of times user wants to halve the step size, *n*

The outputs include

a) approximate value of the derivative at the point and given initial step size
b) exact value
c) 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→exp(2·x);
```

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

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

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

$$xv := 4.0 \quad \textbf{(3.2)}$$

Starting step size, *h*

```
> h := 0.2;
```

$$h := 0.2 \qquad (3.3)$$

Number of times step size is halved

```
> n := 6;
```

$$n := 6 \qquad (3.4)$$

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 first 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

```
> BDD :=proc(f, xv, h)
  local deriv :
  deriv := (f(xv) - f(xv - h)) / h ;
  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);
```

$$y(x) = e^{2x} \qquad (5.1)$$

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

$$Soln := 2\,e^{2x} \qquad (5.2)$$

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

$$Ev := 5961.915974 \qquad (5.3)$$

The next loop calculates the following:

Av: Approximate value of the first derivative using Backward Difference Approximation by calling the procedure "BDD"

Ev: Exact value of the first 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] := 2^i :
```

```
H[i] := h/N[i] :
Av[i] := BDD(f, xv, H[i]) :
Et[i] := Ev - Av[i] :
et[i] := abs(Et[i]/Ev)·100 :
 if (i > 0) then
  Ea[i] := Av[i] - Av[i - 1] :
  ea[i] := abs(Ea[i]/Av[i])·100 :
  Sig[i] := floor(2 - log10(ea[i]/0.5)) :
   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 derivative is calculated and saved in a vector. The approximate error is calculated after at least two approximate values of the 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("Backward Divided Difference") :
  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) :
```

Backward Divided Difference

	...	B	C	D	E	F	G	H
1		"Step Size"	"Approx Value"	"True Error"	"Abs Rel True Error"	"Approx error"	"Abs Rel Approx Error"	"Sig Digits"
2	1	0.2	4913.810460	1048.105514	17.58001150	Ea_0	ea_0	Sig_0
3	2	0.1000000000	5403.560090	558.355884	9.365376608	489.749630	9.063462270	0.
4	3	0.05000000000	5673.513180	288.402794	4.837417959	269.953090	4.758129248	1.
5	4	0.02500000000	5815.321480	146.594494	2.458848710	141.808300	2.438528987	1.
6	5	0.01250000000	5888.009200	73.906774	1.239648031	72.687720	1.234504185	1.
7	6	0.006250000000	5924.808800	37.107174	0.6224035052	36.799600	0.6211103386	1.
8								

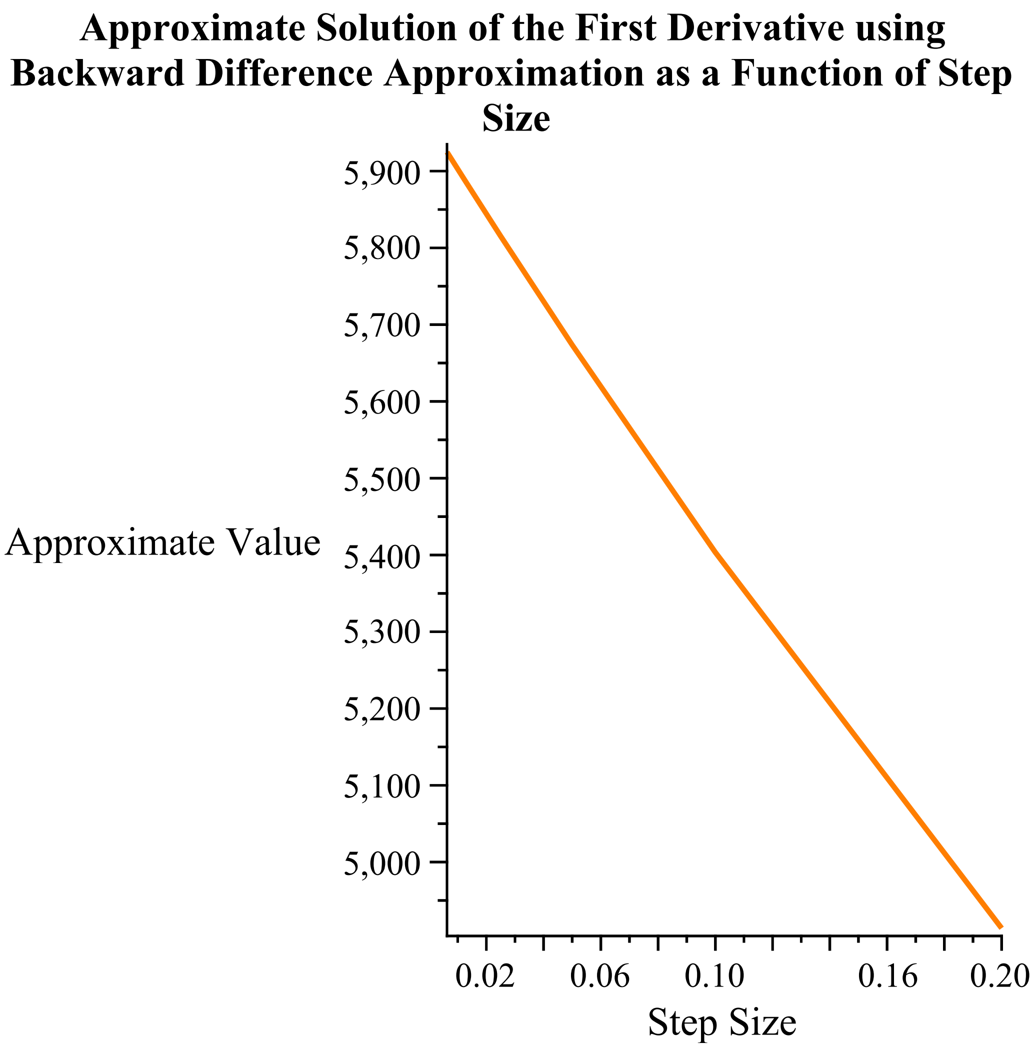
(6.1)

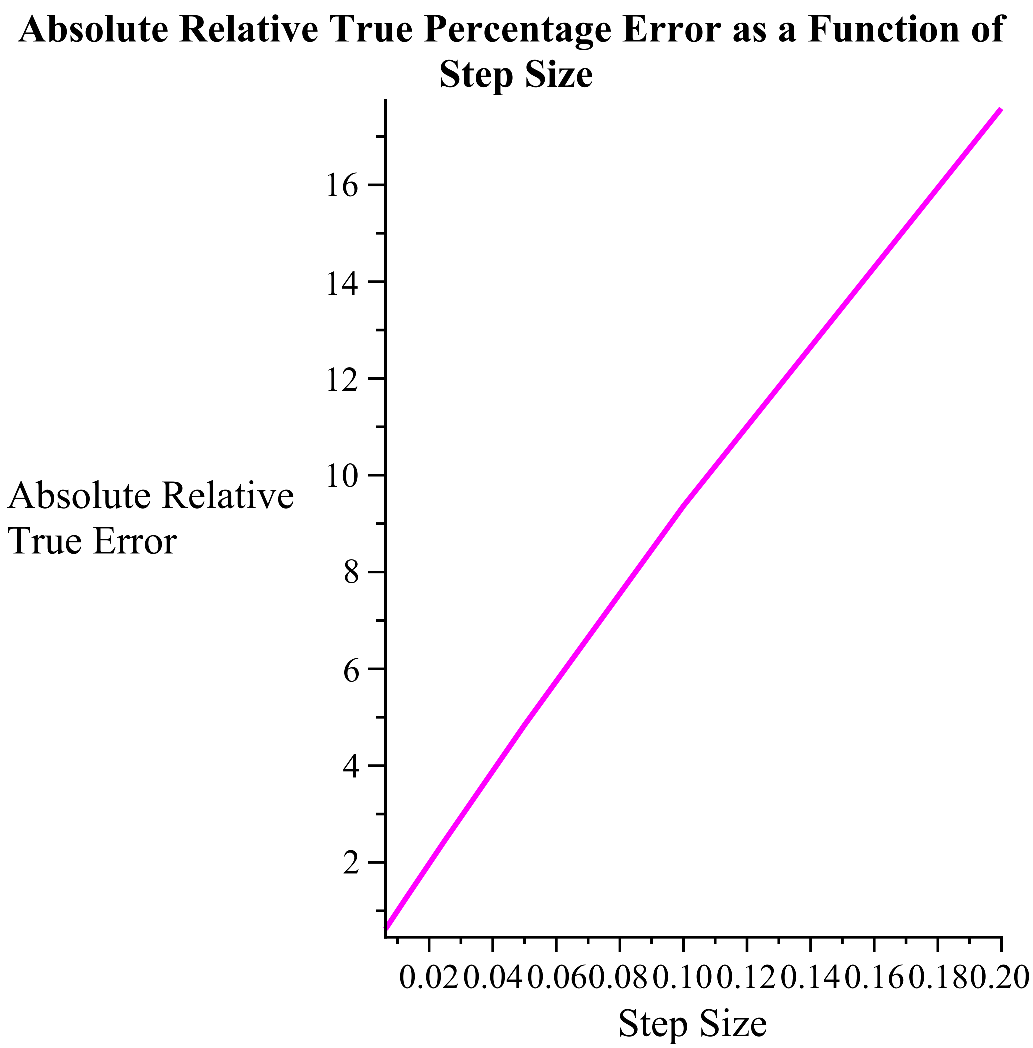
Section 5: Graphs

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.

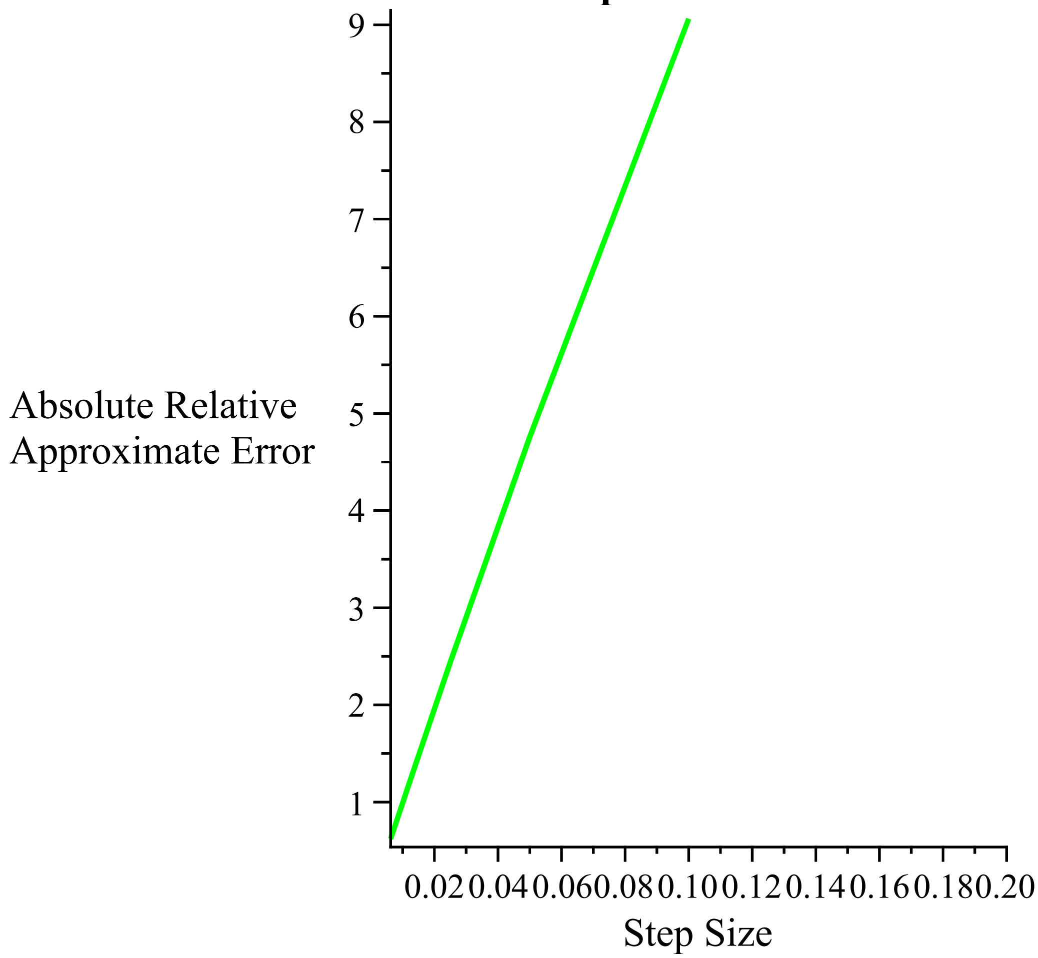
```
> data := [seq([H[i], Av[i]], i=0..n-1)] :
  plot(data, x=H[0]..H[n-1], color=coral, thickness=2, title
    ="Approximate Solution of the First Derivative using
    Backward Difference Approximation as a Function of Step
    Size", labels=["Step Size", "Approximate Value"], titlefont
    =[TIMES, 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
```

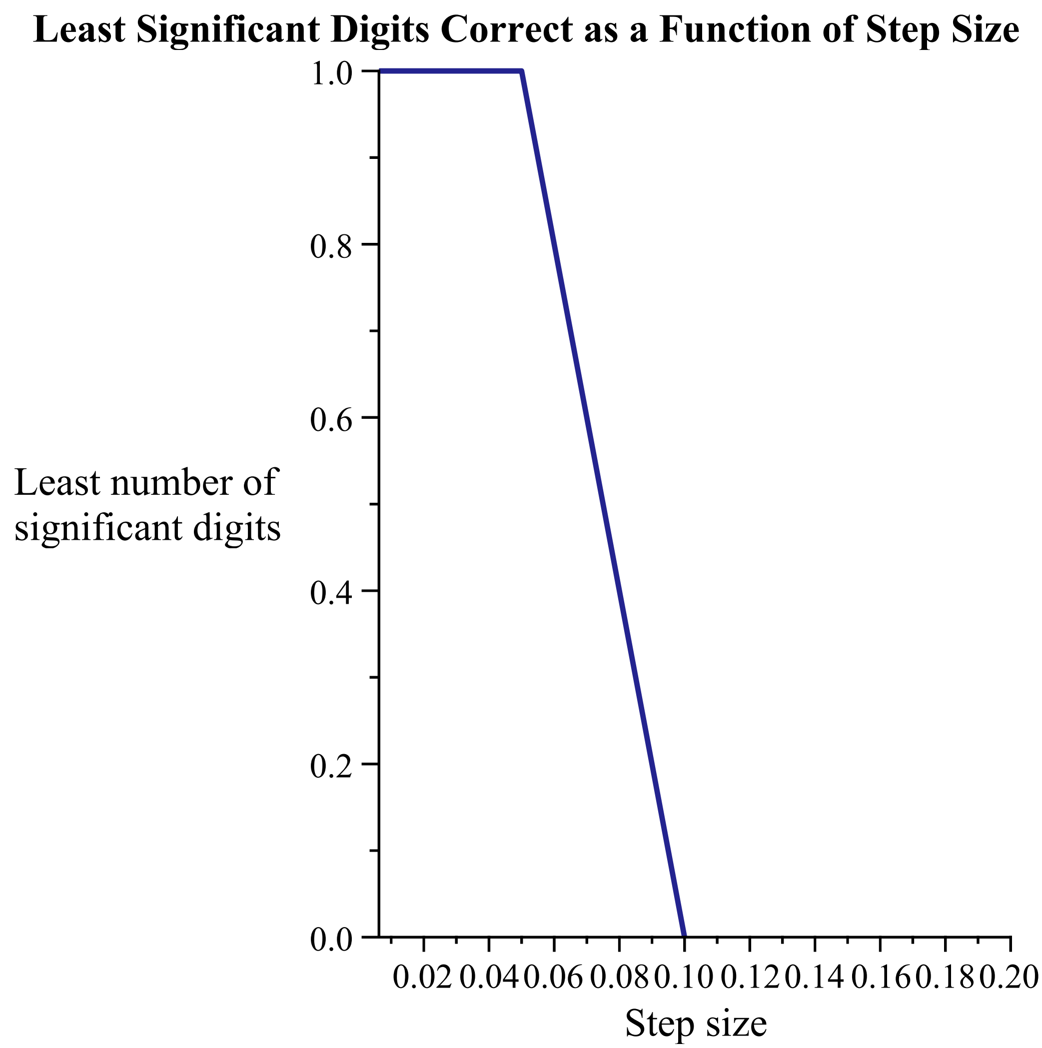
```
significant digits"], titlefont = [TIMES, BOLD, 12], labelfont
    = [TIMES, ROMAN, 12]);
```




Absolute Relative True Percentage Error as a Function of Step Size
16
14
12
10
8
6
4
2
Absolute Relative True Error
0.02 0.04 0.06 0.08 0.10 0.12 0.14 0.16 0.18 0.20
Step Size

Absolute Relative Approximate Percentage Error as a Function of Step Size





>

References

Numerical Differentiation of Continuous Functions.
See http://numericalmethods.eng.usf.edu/mws/gen/02def/mws_gen_dif_txt_continuous.pdf

Questions

1. The velocity of a rocket is given by

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

Use Backward Divided Difference method with a step size of 0.25 to find the acceleration 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 concidential?

Conclusions

To obtain more accurate values of the first derivative using Bacward Difference 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.