

# Re: Testable Predictions by HdB

---

*Source:* <http://sci.tech--archive.net/Archive/sci.math/2005-10/msg01077.html>

---

- *From:* [Han.deBruijn@xxxxxxxxxxxxxxxx](mailto:Han.deBruijn@xxxxxxxxxxxxxxxx)
  - *Date:* 10 Oct 2005 12:31:33 -0700
- 

Han de Bruijn wrote:

> I currently do not have access to a C compiler. But that's  
> just a minor hurdle. I can easily translate a page or two  
> in my favorite speech.

As simple as that: I can not reproduce your results.

Irrespective of the outcome, I want to make sure if we are doing the same thing. Here comes my program and my output and I would like to invite you to post your results too. (It's only twenty lines or so) Otherwise we can't compare.

It is remarked that my own method is implemented somewhat different when compared with your code, namely as I find it should be. May I ask you to conform with this in your C code?

Han de Bruijn

```
----- 8< ----- 8< -----  
program deriv_p;  
{  
#ifndef D  
#define FLOAT float  
#else  
#define FLOAT double  
#endif  
  
FLOAT function(FLOAT x) {  
#ifndef NEG  
return exp(x);  
#else  
return exp(-x);  
#endif  
}  
type  
functie = function(x : double) : double;  
  
function Euler(x : double) : double;
```

## Re: Testable Predictions by HdB

```
begin
Euler := exp(x);
end;

function deriv1(f : funktie; x,h : double) : double;
{
3 points Lagrange
}
var
x1,xr : double;
f1,fr : double;
begin
x1 := x - h;
xr := x + h;
f1 := f(x1);
fr := f(xr);
deriv1 := (fr - f1)/(2 * h);
end;

function deriv2(f : funktie; x,h : double) : double;
{
5 points Lagrange
}
var
x0,x1,x3,x4 : double;
f0,f1,f3,f4 : double;
begin
x0 := x - h * 2;
x1 := x - h;
x3 := x + h;
x4 := x + h * 2;
f0 := f(x0);
f1 := f(x1);
f3 := f(x3);
f4 := f(x4);
deriv2 := (f0 - f1*8 + f3*8 - f4)/(12 * h);
end;

function deriv3(f : funktie; x,h : double) : double;
{
9 points Lagrange
}
var
x0,x1,x2,x3,x5,x6,x7,x8 : double;
f0,f1,f2,f3,f5,f6,f7,f8 : double;
begin
x0 := x - h * 4;
x1 := x - h * 3;
x2 := x - h * 2;
x3 := x - h;
x5 := x + h;
```

```

x6 := x + h * 2;
x7 := x + h * 3;
x8 := x + h * 4;
f0 := f(x0);
f1 := f(x1);
f2 := f(x2);
f3 := f(x3);
f5 := f(x5);
f6 := f(x6);
f7 := f(x7);
f8 := f(x8);

```

```

deriv3 := (f0 * 3 - f1 * 32 + f2 * 168 - f3 * 672 +
f5 * 672 - f6 * 168 + f7 * 32 - f8 * 3)/(840 * h);
end;

```

```

function derivn(f : funktie; x,h,eps : double) : double;
{
HdB Warning: a bit different from _your_ implementation !!
}
var
i, n, hn : integer;
fk, xk : array of double;
sample,sigma,norm : double;
M,d, R : double;
begin
sample := h;
sigma := sample * 2;
norm := sample/(sqrt(2*pi)*sigma);
M := sqrt(2*ln(1/eps));
n := 2*Round(M)+1;
hn := (n-1) div 2;

SetLength(xk,n);
SetLength(fk,n);
for i := -hn to +hn do
begin
xk[i+hn] := x + i * sample;
fk[i+hn] := f(xk[i+hn]);
end;
R := 0;
for i := 0 to n-1 do
begin
d := (x - xk[i])/sigma;
R := R - fk[i] * d * exp(- d*d/2)/sigma;
end;
SetLength(xk,0);
SetLength(fk,0);
derivn := norm * R;
end;

```

```

var
i : integer;
e,h, eps : double;
R1, R2, R3, R4 : double;
begin
{
#ifdef NEG
e = exp(1.0);
#else
e = - exp(- 1.0);
#endif
}
h := 1;
e := exp(1);
eps := 1.E-10;
for i := 0 to 23-1 do
begin
R1 := deriv1(Euler, 1.0, h);
R2 := deriv2(Euler, 1.0, h);
R3 := deriv3(Euler, 1.0, h);
R4 := derivn(Euler, 1.0, h, eps);
Writeln(
',',abs((R1 - e) / e):12
',',abs((R2 - e) / e):12
',',abs((R3 - e) / e):12
',',abs((R4 - e) / e):12
', 2^(-',i,')');
h := h / 2;
end;
end.
----- 8< ----- 8< -----
1.752E-0001 3.754E-0002 2.084E-0003 5.793E+0000 2^(-0)
4.219E-0002 2.146E-0003 6.638E-0006 6.302E-0001 2^(-1)
1.044E-0002 1.312E-0004 2.464E-0008 1.285E-0001 2^(-2)
2.606E-0003 8.153E-0006 9.502E-0011 2.878E-0002 2^(-3)
6.512E-0004 5.089E-0007 3.682E-0013 5.227E-0003 2^(-4)
1.628E-0004 3.179E-0008 3.104E-0015 5.779E-0004 2^(-5)
4.069E-0005 1.987E-0009 5.718E-0015 2.024E-0003 2^(-6)
1.017E-0005 1.242E-0010 8.985E-0015 2.385E-0003 2^(-7)
2.543E-0006 7.761E-0012 1.634E-0015 2.476E-0003 2^(-8)
6.358E-0007 4.911E-0013 8.659E-0015 2.498E-0003 2^(-9)
1.589E-0007 8.691E-0014 6.420E-0014 2.504E-0003 2^(-10)
3.974E-0008 1.983E-0013 2.143E-0013 2.505E-0003 2^(-11)
9.934E-0009 1.363E-0013 2.349E-0013 2.505E-0003 2^(-12)
2.484E-0009 1.363E-0013 1.363E-0013 2.506E-0003 2^(-13)
6.213E-0010 5.823E-0013 6.460E-0013 2.506E-0003 2^(-14)
1.543E-0010 1.425E-0012 1.865E-0012 2.506E-0003 2^(-15)
3.917E-0011 8.054E-0013 1.430E-0012 2.506E-0003 2^(-16)
7.051E-0012 3.656E-0012 4.089E-0012 2.506E-0003 2^(-17)
3.656E-0012 7.225E-0012 8.550E-0012 2.506E-0003 2^(-18)
3.656E-0012 3.656E-0012 6.987E-0013 2.506E-0003 2^(-19)

```

Re: Testable Predictions by HdB

4.648E-0011 6.076E-0011 6.892E-0011 2.506E-0003 2<sup>(-20)</sup>  
4.648E-0011 4.648E-0011 3.465E-0011 2.506E-0003 2<sup>(-21)</sup>  
1.248E-0010 1.819E-0010 2.407E-0010 2.506E-0003 2<sup>(-22)</sup>

---

• **References:**

- ◆ **Re: Testable Predictions by HdB**  
◇ From: Han . deBruijn
- ◆ **Re: Testable Predictions by HdB**  
◇ From: Dik T. Winter
- ◆ **Re: Testable Predictions by HdB**  
◇ From: Han de Bruijn
- ◆ **Re: Testable Predictions by HdB**  
◇ From: Dik T. Winter
- ◆ **Re: Testable Predictions by HdB**  
◇ From: Han de Bruijn

- Prev by Date: **Re: infinity**
- Next by Date: **Re: Please review and comment**
- Previous by thread: **Re: Testable Predictions by HdB**
- Next by thread: **Re: Testable Predictions by HdB**
- Index(es):
  - ◆ **Date**
  - ◆ **Thread**