DBLINT – Double Integrals

Abstract

DBLINT is a program written in 1979 for the HP-34C programmable calculator to compute the numeric value of a definite double integral of a user-specified f(x,y) between given limits. Four worked examples are included.

Keywords: *double integral, definite integration, Gauss-Legendre quadrature, programmable calculator, RPN, HP-34C*

1. Introduction

DBLINT is a short *(67 steps) RPN* program that I wrote in 1979 for the *HP-34C* calculator (will also run with minor modifications in some *RPN* models, such as the *HP-15C*) which can compute the numeric value of a definite double integral of a user-defined function ***f(x, y)*** between specified limits of integration.

The built-in integration functionality already allows for easy computation of arbitrary (definite) integrals like:

$$I = \int_a^b f(x)\,dx$$

But this functionality *can't be nested*, so it can't be used to compute the integral of a function whose definition includes the computation of another integral. Computing double integrals like this is not possible out-of-the-box:

$$I = \int_{x_0}^{x_m} \int_{y_0}^{y_n} f(x,y)\ dy\,dx = \int_{x_0}^{x_m} \left(\int_{y_0}^{y_n} f(x,y)\,dy \right) dx$$

To overcome this limitation, this program uses the built-in integration functionality to compute the ***inner*** integral together with a Gaussian quadrature method to compute the ***outer*** one, namely the fast *3-point Gauss-Legendre* quadrature formula applied over a given number of subintervals. The method is as follows: we want to compute:

$$I = \int_a^b g(x)\,.dx \quad , \quad \textit{where } g(x) = \int_{y_0}^{y_n} f(x,y)\,dy, \quad \textit{which itself is computed via } \int_y^x$$

but first of all the change of variable $x = (b+a)/2 + (b-a)t/2$, $dx = (b-a)/2.dt$ transforms the interval *(a, b)* into the interval *(-1, 1)*. The *3-point Gauss-Legendre* quadrature formula then gives:

$$\int_{-1}^{1} g(x)\,dx = \frac{8}{9} g(0) + \frac{5}{9} \left(g\left(\sqrt{3/5}\right) + g\left(-\sqrt{3/5}\right) \right)$$

which is *exact* for polynomial *g(x)* up to the 5^{th} degree and a 5^{th}-order approximation otherwise, using just 3 evaluations per subinterval. This is far better than *Simpson's Rule*, which only gives 3^{rd}-order accuracy.

2. Program Listing

```
01 ♦LBL A
02 STO 3
03 R↓
04 STO 4
05 R↓
06 X↔Y
07 STO 6
08 -
09 RCL 1
10 ÷
11 STO 7
12 RCL 8
13 STO- 8
14 ♦LBL 2
15 RCL 6
16 RCL 6
17 RCL 7
18 STO+ 6
19 +
20 STO 0
21 X↔Y
22 STO- 0
23 +
24 2
25 STO÷ 0
26 ÷
27 STO 1
28 GSB 0 ►
29 +
30 GSB 1 ►
31 STO 2
32 RCL 1
33 GSB 0 ►
34 -
35 GSB 1 ►
36 STO+ 2
37 5
38 STOx 2
39 RCL 1
40 GSB 1 ►
41 8
42 x
43 STO+ 2
44 RCL 0
45 STOx 2
46 9
47 STO÷ 2
48 RCL 2
49 STO+ 8
50 DSE
51 GTO 2 ►
52 RCL 8
53 RTN
54 ♦LBL 0
55 RCL 0
56 .
57 6
58 √x
59 x
60 RTN
61 ♦LBL 1
62 STO 5
63 RCL 4
64 RCL 3
65 ∫ B
66 RTN
67 ♦LBL B
```

- *67 steps*
- *uses registers R_0 - R_8 , R_I*
- *uses labels A,B,0,1,2*

- *define the function to integrate under* ***♦LBL B***

- *FIX 4 or SCI 4 recommended*
- *RAD mode recommended*

- *the symbols ♦ and ► are purely cosmetic, to indicate branching*

3. Usage Instructions

Step 1: In **PRGM** Mode, key in under *67* **♦LBL B** the sequence of steps which defines the function to integrate ***f(x, y)***, where ***x*** is in register R_5 and ***y*** is in stack register *X*, and end it with **RTN**.

Also, before keying in the function's definition do not forget to *delete* any previous definition from program memory, if there's one, except for *67* **♦LBL B** itself.

Step 2: In **RUN** Mode, store the number of subintervals ***m***: *m* **STO I**

Step 3: Compute the integral: x_0 **ENTER↑** x_m **ENTER↑** y_0 **ENTER↑** y_n **A** *value of integral*

- To try different limits with the same ***f(x, y)***, repeat *Step 2* above.
- To integrate another ***f(x, y)***, go to *Step 1* above but don't forget to *delete* the previous *f(x,y)* first.

Note: The accuracy of the result depends on both the display mode **FIX d** or **SCI d** selected and the number ***m*** of subintervals chosen. It is strongly recommended to use ***d = 4*** or less and ***m = 1*** or ***2***. These choices will usually give about 4 correct places in moderate run times. If more accuracy is needed, first increase ***m*** and as a last resort set **FIX 6** or **SCI 6**. Keep in mind that going from ***m = 1*** to ***2*** more than *duplicates* the running time, while going from **FIX 4** to **FIX 6** increases the running time by a factor of *2-3*. See the ***Examples***.

4. Examples

The following examples can be useful to check that the program is correctly entered and to understand its usage.

4.1 Example 1

Evaluate $I = \int_0^1 \int_1^2 (x^2 + y^2)\, dy\, dx$

First of all, we define the function to integrate, ***f(x,y)***:

In **PRGM** Mode, enter under *67* **♦LBL B** this *6-step* program to define the function *f(x,y)* to be integrated:

67 **♦LBL B**	*68* x^2	*69* RCL 5	*70* x^2	*71* +	*72* RTN

In **RUN** Mode, we'll specify just one subinterval and **FIX 4**: **FIX 4** 1 **STO I**

Finally, enter the limits of integration and compute the integral:

0 **ENTER↑** 1 **ENTER↑** 1 **ENTER↑** 2 **A** <u>*2.6667*</u> **FIX 7** <u>*2.6666666*</u> *(exact is **8/3** so we got 8 correct places)*

4.2 Example 2

Evaluate $I = \int_3^4 \int_1^2 \frac{dy\, dx}{(x+y)^2}$

First, we define the function to integrate, ***f(x,y)***:

In **PRGM** Mode, enter under *67* **♦LBL B** this *6-step* program to define the function *f(x,y)* to be integrated:

67 **♦LBL B**	*68* RCL 5	*69* +	*70* x^2	*71* 1/x	*72* RTN

In **RUN** Mode, we'll specify just one subinterval and **FIX 4**: **FIX 4** 1 **STO I**

Next, enter the limits of integration and compute the integral:

3 **ENTER↑** 4 **ENTER↑** 1 **ENTER↑** 2 **A** <u>0.0408</u> **FIX 6** <u>0.040821</u> *(exact is **Ln(25/24)** so we got 6 places)*

4.3 Example 3

Evaluate $\boldsymbol{I} = \int_{-2.3}^{1.6} \int_{3.9}^{6.1} \left(e^{-x^2} + x^3 - y^3x^2 + 7\right) tan^{-1}(x-2)\, sin(y+3)\, dy\, dx$

First, we define the function to integrate, ***f(x,y)***:

In **PRGM** Mode, enter under *67* **♦LBL B** this *27-step* program to define the function *f(x,y)* to be integrated:

67 **♦LBL B**	*72* -	*77* CHS	*82* +	*87* x	*92* x
68 STO 9	*73* RCL 5	*78* e^x	*83* RCL 5	*88* RCL 9	*93* RTN
69 3	*74* x^2	*79* X↔Y	*84* 2	*89* 3	
70 y^x	*75* x	*80* -	*85* -	*90* +	
71 RCL 5	*76* LST*x*	*81* 7	*86* TAN^{-1}	*91* SIN	

In **RUN** Mode, we'll specify **RAD** mode, 2 subintervals and **SCI 4**: **RAD** **SCI 4** 2 **STO I**

Now, enter the limits of integration and compute the integral:

-2.3 **ENTER↑** 1.6 **ENTER↑** 3.9 **ENTER↑** 6.1 **A** <u>1.3213e03</u> **FIX 2** <u>1321.27</u> *(all 6 places are correct)*

Note: This is a particularly difficult example. First, *f(x,y)* takes 6 sec. to evaluate, which greatly increases run time. Second, the interval of integration is quite wide, which affects accuracy. Still, we *got 6 correct places* in reasonable time.

4.4 Example 4

Evaluate $\boldsymbol{I} = \int_{0}^{\infty} \int_{1}^{\infty} e^{-x^2-y^2}\, dy\, dx$

First of all, we define the function to integrate, ***f(x,y)***:

In **PRGM** Mode, enter under *67* **♦LBL B** this *8-step* program to define the function *f(x,y)* to be integrated:

67 **♦LBL B**	*68* x^2	*69* RCL 5	*70* x^2	*71* +	*72* CHS	*73* e^x	*74* RTN

In **RUN** Mode, we'll specify 3 subintervals and **FIX 4**: **FIX 4** 3 **STO I**

Last, enter the limits of integration (replacing ∞ by 4, as *f(4,4) < 1.27e-14*), and compute the integral:

0 **ENTER↑** 4 **ENTER↑** 0 **ENTER↑** 4 **A** <u>0.7853</u> *(exact is **π/4** so we got 4 correct places despite the finite interval)*

Notes

1. This program is included in *Hewlett-Packard's Solution Book "HP-34C Matemática Avanzada" (Spanish)*
2. This program is featured in my article *"HP Article VA023 - Long Live the HP-34C"*

References

Francis Scheid (1988). *Schaum's Outline of Theory and Problems of Numerical Analysis, 2nd Edition.*

Copyrights