
Subject: int_2D and int_3D

Posted by [Geoff.Cureton](#) on Mon, 26 Nov 2007 09:41:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi all,

I am trying to use int_2D and int_3D, and understand their behaviour so I can properly write the functions that I am trying to integrate.

From the IDL help for int_2D, it appears that I am required to return a vector of function values using a single x value and a vector of y values. Printing out the x and y values passed to the function seems to bear this out.

Printing out the values passed to my 3D function (when using int_3D) indicates a vector of z values for single x and y values, which I concede has a certain symmetry with the int_2D routine. However, the IDL help for int_3D gives the impression that the 3D function being integrated only need return a scalar result for a single (x,y,z) triple.

A simple test case I did for int_3D, where I wrote the function under the assumption that I only needed to return a scalar, worked anyway, perhaps by accident, i.e. if my function is of the form...

$$f(x,y,z) = x + y + z,$$

and I pass it scalar x and y, and a vector z, IDL seems to be able to return a vector result `f(x,y,z[])` without any special coding on my part.

Am I missing something? I would grateful if someone could share their insights regarding the similarities/differences of int_2D and int_3D, and the correct usage of each.

Cheers,
Geoff
