Subject: Re: run-time function creation
Posted by PMan on Wed, 10 Apr 2013 19:57:17 GMT
View Forum Message <> Reply to Message

On Wednesday, April 10, 2013 1:04:49 PM UTC-4, Craig Markwardt wrote:
> On Wednesday, April 10, 2013 10:33:31 AM UTC-4, Paul Mallas wrote:
>
>> I know this is a bit off the wall, but I was wondering if there is a technique or method for doing run-time function creation.
>
>>
>
>>
>
>>
>
>> The reason I ask is I was looking at the IDL integration routines (e.g., qsimp) and this requires a singular argument function name as input. My problem is I have a function with several parameters I need to integrate, but don't have these parameters until I calculate them at run time. If I could somehow create the function dynamically, I could create a function that would satisfy the the qsimp requirement for a single argument function, but prior to run-time I can't.
>
>
>
> Well, I have two answers for you.
>
>
>
> The first answer is that you can use my QPINT1D which is a better integrator than the IDL-standard quadrature functions. Plus, it allows you to enter in a private variable (a structure) for other parameters.
>
>
>
> If you really want run-time functions, then FILE_COMPILE will do that for you. You need to write out a scratch file with the function you want, and then FILE_COMPILE will do the tricky part of compiling it. (not so tricky, but getting the paths right is more work than you might think)
>
>
>
> Craig
>
>
>
> QPINT1D
>
> http://cow.physics.wisc.edu/~craigm/idl/math.html#QPINT1D
>

>
>
> FILE_COMPILE
>
> http://cow.physics.wisc.edu/~craigm/idl/introspect.html#FILE _COMPILE

Craig - I used your qpint1. Excellent! It was just what I needed and it worked great. I even have the imsl lib and qpint1 worked much better than the imsl_intfcn. imsl_intfcn is fast and gives the right answer, but gives tons of floating point warnings and - for some reason - it kills the workbench when I run it. qpint1 appears slightly faster and does not deleterious side effects. Thank you!
