Originally posted by: none

It is actually based on LSODAR, also from ODEPACK. I think this fact is in
IDL's documentation.

The R in LSODAR stands for root, and you can read ODEPACK's documentation to
figure out what the rootfinding capability means. The point may be moot,
however, because I don't know that RSI ever exposed the rootfinding
capability of LSODAR to the user. Even so, since LSODAR can return
STATUS=3, it is conceivable you can get that return status.

The ability to use callbacks to get to IDL functions from ODEPACK is
possible, and may even be documented. If it isn't documented, it may be
proprietary information. If so, I'm not allowed to divulge it. :(

<ushomirs@my-deja.com> wrote in message news:7r90g4rqb1@nnrp1.deja.com...
> Hi,
>
> It looks like IDL 5.2 includes one of the ODEPACK routines, LSODE. But
> that's just the basic routine, FORTRAN ODEPACK on Netlib (which is where
> i think they got the routine) has several more sophisticated routines.
>
> In addition, it seems that RSI has done a sloppy job integrating LSODE
> into IDL. they don't support all the options the original LSODE has,
> including the useful ones, like single-stepping through the integration.
> the documentation seems haphazardly put together. for example, their
> description of output value STATUS=3 is
>
> "The integration was performed successfully, and no roots were found"
>
> what roots?? anyway, this makes me doubt the correctness of the
> implementation.
>
> Anyway, I have 2 questions.
> (1) does anyone know if RSI has plans to integrate the rest of ODEPACK
> routines into IDL??
>
> (2) Since the answer to (1) is probably "No", how do I call an IDL
> routine from a DLM (dynamically loadable module)?? What i'm thinking is
> just compiling all of the odepack routines into a dlm, and writing and
> IDL interface. however, i would need to pass the function that computes
> the right-hand side of the differential equation to the FORTRAN routine

> in a dlm. IDL external development guide describes how to call external
> routines from IDL, but not the other way around.
>
> Please note that "Callable IDL" is probably **not** what's needed here.
> It seems to me that "callable idl" inits a whole separate copy of idl,
> which is clearly not what i need here. (unless i'm wrong on this
> point).
>
>
> greg
>
>
> Sent via Deja.com http://www.deja.com/
> Share what you know. Learn what you don't.
>