Subject: Re: IDL, PV-Wave, Matlab, Khoros
Posted by kevin on Tue, 19 Jan 1993 17:58:15 GMT
View Forum Message <> Reply to Message

alan@elroy.Jpl.Nasa.Gov (Alan S. Mazer) writes:

> Can someone tell me how these packages might compare in an image analysis
> environment? What I really need is a nice front end that can do plots and
> simple image manipulation and interface easily with a lot of custom C code.

I don't know about IDL or Khoros, but:

PV Wave is primarily a graphics package that has math functions
on the side.

Matlab is primarily a math package (or more precisely a matrix math
package) that has graphics functions on the side.

Not suprisingly, Wave is better at graphics and Matlab is better at
math. Both packages have C interfaces, but the C interface to matlab
is easier than the one to Wave

Subject: Re: IDL, PV-Wave, Matlab, Khoros
Posted by thompson on Tue, 19 Jan 1993 21:49:00 GMT
View Forum Message <> Reply to Message

In article <1993Jan19.175815.8722@iscnvx.lmsc.lockheed.com>,
kevin@dipl.rdd.lmsc.lockheed.com writes...
> alan@elroy.Jpl.Nasa.Gov (Alan S. Mazer) writes:
>
>> Can someone tell me how these packages might compare in an image analysis
>> environment? What I really need is a nice front end that can do plots and
>> simple image manipulation and interface easily with a lot of custom C code.
>
> I don't know about IDL or Khoros, but:
>
> PV Wave is primarily a graphics package that has math functions
> on the side.

IDL and PV Wave are very similar products having a common ancestor (an older
version of IDL). The above description could apply equally well to IDL.

> Matlab is primarily a math package (or more precisely a matrix math
> package) that has graphics functions on the side.
>
> Not suprisingly, Wave is better at graphics and Matlab is better at
> math. Both packages have C interfaces, but the C interface to matlab

> is easier than the one to Wave

I don't know about PV-Wave, but IDL's C interface is *very* easy to use, IMHO.
I think that this feature is one of those that is *not* identical between the
two packages, but can't be positive.

Bill Thompson

Subject: Re: IDL, PV-Wave, Matlab, Khoros
Posted by at913 on Wed, 20 Jan 1993 14:39:15 GMT
View Forum Message <> Reply to Message

In a previous article, thompson@stars.gsfc.nasa.gov (William Thompson, code 682.1, x2040)
says:

> In article <1993Jan19.175815.8722@iscnvx.lmsc.lockheed.com>,
kevin@dipl.rdd.lmsc.lockheed.com writes...
>> alan@elroy.Jpl.Nasa.Gov (Alan S. Mazer) writes:
>>
>>> Can someone tell me how these packages might compare in an image analysis
>>> environment? What I really need is a nice front end that can do plots and
>>> simple image manipulation and interface easily with a lot of custom C code.
>>
>> I don't know about IDL or Khoros, but:
>>
>> PV Wave is primarily a graphics package that has math functions
>> on the side.
>
> IDL and PV Wave are very similar products having a common ancestor (an older
> version of IDL). The above description could apply equally well to IDL.
>
Personally, I think IDL presents an excellent mix of math capabilities and
graphics. Easy to produce numbers, easy to plot them.
Programs written in IDL can be 30-60 times slower than equivalent programs
written in FORTRAN, but if one uses extensive array manipulation facilities
execution improves tremendously (factors of 10).

> I don't know about PV-Wave, but IDL's C interface is *very* easy to use, IMHO.
> I think that this feature is one of those that is *not* identical between the
> two packages, but can't be positive.
>
I second that

> Bill Thompson
>
--

Mirko Vukovic
University of Wisconsin -- Madison
VUKOVIC@UWMFE.NEEP.WISC.EDU

Subject: Re: IDL, PV-Wave, Matlab, Khoros
Posted by thompson on Thu, 21 Jan 1993 15:59:00 GMT
View Forum Message <> Reply to Message

In article <1jjo6jINN1tn@usenet.INS.CWRU.Edu>, at913@cleveland.Freenet.Edu (Mirko Vukovic) writes...

(stuff deleted)

> Programs written in IDL can be 30-60 times slower than equivalent programs
> written in FORTRAN, but if one uses extensive array manipulation facilities
> execution improves tremendously (factors of 10).

(rest deleted)

Actually our experience has been that, depending on the problem and how the IDL code is written, IDL routines are just about as fast as an equivalent FORTRAN routine. The secret seems to be the use of loops. If you can write your code without any DO loops--and a lot of times you can--then IDL's performance is just about the same as a FORTRAN routine. If you need to use loops, then performance suffers greatly from the need to reinterpret the commands over and over again.

A lot of times I've seen less experienced IDL programs use loops when there were ways to avoid them--I've been guilty of it myself.

Bill Thompson