

---

Subject: Re: BVLS (NNLS) without loops?  
Posted by [Mike\[5\]](#) on Tue, 17 Dec 2013 17:35:53 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Juan Pablo,

From what I understand, your BVLS (Bounded-Variables Least-Squares) problems in every pixel are completely independent. They don't share the coefficients of the A array nor the B vector. The only similarity between the different problems is the size of the arrays.

If this is the case I don't see how one could speed-up the problem by some form of vectorization.

The only real solution would be for IDL to include a compiled BVLS routine in the core language. I had hoped for this to happen for some time, given that so much spectral analysis work has to solve this kind of problem.

So I take the opportunity of your message to suggest to Exelis to consider the inclusion of a compiled version of BVLS (<http://www.netlib.org/lawson-hanson/all>) in the core language.

Cheers,  
Michele

On Tuesday, December 17, 2013 6:32:00 AM UTC, JP wrote:

> Hi IDLers,  
>  
> I am currently using BVLS (<http://www-astro.physics.ox.ac.uk/~mxc/idl/bvls.pro>) for spectral unmixing. It works great, but, on one pixel at a time. I want to implement for many pixels without having to loop. Example below:

```
>  
>  
>  
> IDL> help, a  
>  
> A          FLOAT    = Array[85, 3]  
>  
> IDL> help, b  
>  
> B          FLOAT    = Array[85]  
>  
> IDL> help, bnd  
>  
> BND        FLOAT    = Array[2, 3]  
>  
> IDL> print, bnd  
>  
>    0.000000    1.00000  
>  
>    0.000000    1.00000
```

```
> 0.000000    1.00000
>
>
>
>
>
>
> bvlS, A, B, BND, X_BVLS
>
>
> IDL> help, x_bvlS
>
> X_BVLS      FLOAT   = Array[3]
>
>
>
> in my example A is a vector of "endmembers" (85 spectral bands and 3 fractions), B is a pixel
(vector of 85 bands), BND are bounds (don't go negative nor >1) and the X_BVLS are the
estimated fractions returned for that pixel.
>
> Now as you could imagine I have many many pixels (n) (my B is really a 2D array of [85, n])
and i've written a function which loops through n, but it gets very slow for large n.
>
>
>
> Any ideas?
>
>
>
> Thanks
```