
Subject: Re: Approximate convolution - for loop problem

Posted by [David Gell](#) on Fri, 02 Jan 2009 19:12:11 GMT

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

On Dec 21 2008, 11:32 am, samuel.le...@gmail.com wrote:

```
> Hello everyone, I'm trying to execute a 1-d convolution of an array,
> signal.
>
> Using an analytic approximation, obtaining the convolved bolometer
> signal, bolo_signal, at time step ii, is given by the following:
>
> nsamp=n_elements(signal)
> const1 = exp(-tsamp/taubolo)
> const2 = 1.-const1
>
> bolo_signal = const2*signal
> for ii= 1L,nsamp-1L do begin
>   bolo_signal[ii] += const1*bolo_signal[ii-1]
> endfor
>
> where tsamp and taubolo are scalars. Is there any way to avoid the for
> loop in this case? The hope is to speed up the execution.
>
> Many thanks for your help
>
> Sam Leach
```

An alternative way requires building an upper triangular matrix of powers of the second constant

```
anA=[1.,2.,3.,4]
nConst1 = 1.0
nConst2 = 0.5
```

```
nEle=n_elements(anA)
```

```
;bulid a upper diagonal matrix from the constants
anMatrix = fltarr(nEle,nEle)
for nl=0,nEle-1 do anMatrix[nl:*,nl] = nConst2^indgen(nEle-nl)
```

```
;compute result
anResult = anA ## anMatrix
help, anResult
print, anResult
end
```

```
% Compiled module: $MAIN$.
```

```
ANRESULT    FLOAT    = Array[4]
  1.00000    2.50000    4.25000    6.12500
```

The problem is building the upper triangular matrix. If that could be done efficiently, then this might speed things up.

David Gell
Southwest Research Institute
San Antonio, TX
