Subject: Re: Solving system of ODEs backwards in time?
Posted by Markus Schmassmann on Wed, 02 Aug 2017 14:58:32 GMT
View Forum Message <> Reply to Message

On 07/30/2017 08:37 PM, Barry Lesht wrote:

> I have a system of ODEs describing how a system with N state
> variables (C) evolves in time. The basic equation set is dC/dt = (W
> + A dot C) / V in which C is the state variable vector at time i, V
> is a vector of constants, W is a known vector (a function of t), and
> A is a known matrix (similarly time variable). Given an initial
> condition C[0], I've been using LSODE to solve for the successive
> time steps, updating the initial condition and values of W and A
> along the way. This has worked well.
>
> Now I'd like to reverse the problem. That is, if I know the value of
> the state vector at time i, and the values of W and A at time i-1,
> I'd like to compute the value of the state vector at time i-1. In
> essence, I want to know what the initial condition had to be to
> arrive at the current state of the system given known V, W and A.
>
> Frankly, it's been many, many years since I took an ODE class and I
> wasn't very adept then. I'd greatly appreciate any advice on how to
> approach this problem.

```
; example inputs
n=10
tt=100
  w=randomu(seed,n,tt-1,/double)
  a=randomu(seed,n,n,tt-1,/double)
  v=randomu(seed,n,/double)*100
c0=randomu(seed,n,/double)

; run it forward
ca=dblarr(n,tt)
ca[*,0]=c0
for i=0,tt-2 do ca[*,i+1]=ca[*,i]+(w[*,i]+a[*,*,i]#ca[*,i])/v

; run it back
cb=dblarr(n,tt)
cb[*,-1]=ca[*,-1]
diag_v=dblarr(n,n)
diag_v[lindgen(n),lindgen(n)]=v
for i=tt-2,0,-1 do cb[*,i]=invert(diag_v+a[*,*,i])#(cb[*,i+1]*v-w[*,i])

; compare results
print, ca[*,0]
print, cb[*,0]
```

however, if diag_v+A+a[*,*,i] can't be inverted you get nonsense as result.
So check it by running the inversion forward again

I hope this helps, Markus