R documentation

of ‘propagate_s.Rd’

December 12, 2012

`propagate_s` *Propagate stochastic model forward over a series ot timesteps*

Description

Essentially a wrapper to `propagate_1step_S`. Propagate determinstic model forward over a series ot timesteps. The numerical scheme is supplied by the model (see original codes). A routine *Npropagate_s* should be provided in a forthcoming package version, to integrate simultaneously a bunch of initial conditions

Usage

```
propagate_s(model, times, init, par, Astro, seed=01425, deltat=.1,isum= 1)
```

Arguments

`model`	One of the deterministic models out of the packaged model list (`data(models)`) or user-supplied.
`times`	Vector of times. `times[1]` at which the model trajectory is being evalueted. Is used as the initial condition
`init`	State vector of initial conditions
`par`	Parameter vector
`Astro`	Astronomical forcing list, as supplied by routine `read_astro`
`seed`	Random number seed (is converted to an integer)
`deltat`	Time step used supplied to stochsatic propagator
`isum`	Every random number is obtained as the sum of `isum` random numbers, then divided by `sqrt{isum}`. See application in `examples` below.

Value

Matrix, with columns as follows:

- `Column 1 .. n`: States at specified times (`n` is the system dimension)
- `Column n+1`: precession according to the astronomical solution
- `Column n+2`: obliquity according to the astronomical solution

Author(s)

M. Crucifix (2012)

See Also

`propagate_d` (deterministic version)

Examples

```
## check convergence of time series

require(iceages)
data(models)
Astro <- read_astro(34,34)
times=seq(-80,0,0.1)


# test brownian bridge
bb <- function(spar, title='Brownian bridge: vdp mode')
{
  bs = c(1,2,5,10,20)
  for (j in seq(along=bs))
  {
    print(bs[j])
    print(0.1/bs[j])
    STOCH <- propagate_s(models$cr12_s, init=c(0., 0.),
                         par=spar, times=times, Astro=Astro,
                         isum=bs[j], deltat=0.01*bs[j], seed=132);
    if (j==1)
    {
       plot(times, STOCH[,1], type='l', main =title)
    } else {
       lines(times, STOCH[,1], type='l', col=j)
   }
  }
  legend('topright', legend=bs, lty=1, col=seq(along=bs))
}

# van der pol mode

spar1 = c(alpha = 30., beta0 =  0.7, beta1 = 0,
         beta2 = 0., delta = 1.,
         gammapre = 0.7, gammaobl = 0.6,
         omega = 3.695, sigmax=0.3, sigmay=0.3)

bb(spar1, title='van der pol mode')

# 1-d mode
spar2 = c(alpha = 30., beta0 =  0.0, beta1 = 0.70,
         beta2 = 1.4, delta = 0.,
         gammapre = 3.696, gammaobl = 3.20,
         omega = 3.695, sigmax=0.8, sigmay=0.8)


bb(spar2, title='1-d mode')

# intermediate

spar = 0.5*(spar1 + spar2)

bb(spar=0.5*(spar1+spar2), title='intermediate')
```

Index