Subject: Re: timegen
Posted by spluque on Thu, 31 Oct 2013 02:36:44 GMT
View Forum Message <> Reply to Message

On Wed, 30 Oct 2013 15:16:57 -0700 (PDT),
Phillip Bitzer <bitzerp@uah.edu> wrote:

> This may be pertinent, from the help: Note: If the step size is not an
> integer then the last element may not be equal to the FINAL time. In
> this case, TIMEGEN will return enough elements such that the last
> element is less than or equal to FINAL.

> If you 'caldat' your final time (end_jd + 1 - (float(step_size) /
> 86400)), I suspect this is the issue.

> But, I'm not sure why this isn't what you're looking for:

> ts=timegen(start=beg_jd, final=end_jd+1,step_size=step_size,
> units='seconds') caldat, ts, mo, dd, yyyy, hh, mm, ss print, yyyy[-1],
> mo[-1], dd[-1], hh[-1], mm[-1], ss[-1] 2013 10 11 23 59 59.902636

> BTW, your original code contains the line float(step_size) which is
> superfluous - step_size is already a float, yes?

Thanks very much for these pointers. I left the float() call with
step_size by accident here. In the actual code, this is part of a
procedure that needs to be quite general for any step_size, so I'm
coercing it to float. You're quite right about the note in the help
page; that's exactly what's going on. To protect against this, the
following seems to do what I need:

```
step_d=float(step_time) / 86400
times=timegen(start=beg_jd, $
              final=end_jd + 1 - (step_d / 2), $
              step_size=step_time, units='seconds')
```

i.e. adding half a step to 'final' ensures that the last step is always
included.

Thanks,

--
Seb