Subject: Re: TRIANGULATE. Finding contiguous cells efficiently?
Posted by Wox on Wed, 28 Feb 2007 09:42:43 GMT
View Forum Message <> Reply to Message

Unique edges: nice thinking there :-). Off course, I would write some
things differently like:

```
neigh=intarr(3,ntr)
neigh(*,*)=-1
=> neigh=replicate(-1,3,ntr)

edgcount(cellhere)=edgcount(cellhere)+1L
=> edgcount[cellhere]++
```

To make unique edges, you could use hash functions like CRC32 (with
some adaption to make [a,b] the same as [b,a]), but assuming the
vertices are stored in ulong's (i.e. 32bit), this will do:
```
c=(a>b)+ishft(long64(a<b),32)
```

Who am I: http://www.ua.ac.be/wout.denolf

On 27 Feb 2007 11:44:13 -0800, "Libertan" <tbethell@umich.edu> wrote:

> Wox,
>
> Okay, here's an code which surpasses my expectations; it seems to be
> over 10,000% faster. I kid ye not. initially vectorized, ends with
> simple loop over uncostly operations. I thought the invocation of
> SORT would be slow, but imagine it's written in C. would I be right
> in thinking that the overall trick is to use 1) as few operations as
> possible, 2) use vectorized forms, and 3) use IDL instrinsics written
> in C?
>
> I'm sure it's not particularly well written, but what do you think?
> Writing fast codes in IDL is a tricky business. Wox, if you care and
> are prepared to email me your name, I'll acknowledge our discussion
> when/if I publish future results (in Astrophysical Journal).