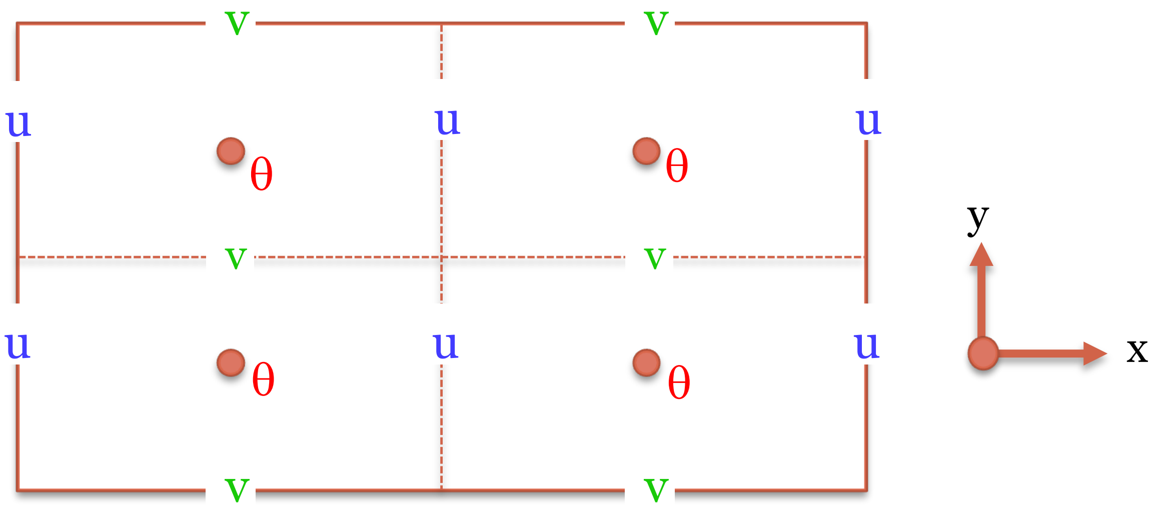
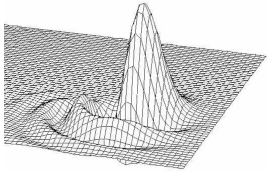
# *continued:*
# Computer Program #2

## TWO-DIMENSIONAL ADVECTION

*2-D Staggered "C-grid"*



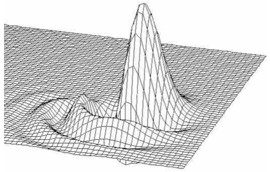
# Program 2 – Advection *routines*



- **advect1d**()
  - your old advection routine!
  - takes one scalar array…
    - integrates forward, n>n+1
    - 3 schemes for program 2
  - input *(n)*:
    - 1-D scalar field *s1d*
    - velocity component *vel1d*
      - *staggered!! (discuss)*
  - output *(n+1)*:
    - 1-D scalar field *s2*
  - *advect1d:*
    - assumes BCs already set
    - only works with 1-D data

- **advection**()
  - essentially a new routine
  - 1st double loop: X-advection
  - 2nd double loop: Y-advection
  - each advection pass:
    - copy "n" time *row or column* from 2-D s1() > 1-D *s1d()*
      - includes ghost points
    - copy *row of u() or column of v()* > 1-D *vel1d*() array
      - copy *staggered* data
    - call advect1d()  (n > n+1)
    - insert new values into s1()

# Program 2 – coding of: X advection

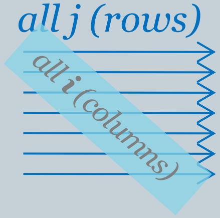


- **Advection**
  - I set up temporary 1-D arrays in my 2-D advection routine –
    - s1d(-2:nx+3), vel1d(nx+1) ***no*** *ghost points for U, V !!* until we do nonlinear advection, and u, v are evolved in time, just like s1()

      *for the scalar field* *for a velocity (u or v) field*

- Advecting *s1* rows (X)
  - copy s1(i,j) to s1d
  - copy u(i,j) to vel1d
  - pass s1d, vel1d to *advect1d()*
    - advect1d returns *s1d_out*
  - copy s1d_out(i) to s1(i,j) *not including[2] ghost points!!*

within outer *j* loop, for y



*Each* bullet is a 1-D loop; I use index ***i*** **...** for the x-direction. All are within the outer ***j*** loop. Note: in Fortran, can do this w/subscript notation!

- Coding X advection:
  - loop over all ***j*** rows
    - loop over all ***i*** columns *including[1] ghost points!!*
      - copy s1(i,j) to s1d(i)
    - loop over all ***i*** (nx+1) columns
      - copy u(i,j) to vel1d(i)
    - call *advect1d( )*
    - loop over all ***i*** (nx) columns[2]
      - copy s1d_out(i) to s1(i,j)

j i i i

[1] so advect1d() doesn't have to worry about BCs

[2] advect1d() does **not** set ghost points!