Overview of Adaptive Z-Grid (AZG) Scheme

In order to improve simulations of stratocumulus clouds, while using only a modest number of vertical layers, I have added an adaptive vertical (Z) grid to the System for Atmospheric Modeling (SAM) model version 6.10.2. The ideas behind the AZG model is to add vertical layers where and when needed rather than using fine vertical grid spacing throughout the boundary layer. By default, the AZG scheme is turned off.

As currently constructed, three criteria of can be used with AZG scheme for determining when to insert or remove vertical layers: the first criteria is based on the potential temperature, the second on the ratio of the subgrid-scale vertical water flux to the total water flux, and the third on the ratio of the SGS to total vertical water flux and SGS to total kinetic energy. A completed description and case study results can be found in the following paper (a copy of which has been put in the DOC directory).

Marchand, Roger and Thomas Ackerman (2011), A Cloud-Resolving Model with an Adaptive Vertical Grid for Boundary Layer Clouds. *J. Atmos. Sci.*, **68**, 1058–1074. doi: 10.1175/2010JAS3638.1

The AZG scheme has seen little testing outside of the cases presented in the above article. In general I do not expect the SGS-based criteria to work well if the resolution is changed much from a baseline grid with 100 m horizontal and vertical grid spacing (and a 25 m adaptive initialization) – at least not without changing the thresholds used in deciding when to add/remove layers.

In this document I describe parameters added to the SAM model to support the AZG scheme and briefly describe how to alter any existing SAM subroutines (that have not already been modified to work with the AZG scheme). Currently, the AZG scheme will work with essentially all of the code options distributed with SAM 6.10.2 -- meaning any of the three microphysics schemes (SAM1MOM, M2005, and DRIZZLE) and any of the three radiation options (SIMPLE, RAD_CAM, or RRTM), either of the two advection schemes (MPDATE or UM5) and with or without MPI.

If you are developing additional code for SAM please make it compatible with the AZG scheme. Questions and comments can be sent to Roger Marchand (rojmarch@u.washington.edu).

AZG Parameters

AZG_scheme - In order to activate the AZG scheme, the parameter *AZG_scheme* (which can be set in the "prm" file) should be set to a value of 1, 2, or 3 (depending on which criteria you wish to use). AZG_scheme set to 0, effectively turns the scheme off. Additional criteria can be easily added to the subroutine "adapt_vertical_grid". (default value is 0).

AZG_initial_adapt_dz - if >0 then layers are added at the startup where liquid water is present until all layers with liquid water have a width that is <= to the value of AZG_initial_adapt_dz. Testing showed that a value of 25 m worked well. (units = meters, default value is 0 m).

AZG_nstep – is the number of steps that must elapse before the vertical grid is altered/adapted. Testing showed a value of 5 to 10 minutes worked well and this parameter should be set in concert with the time step "dt". Note: in general "dt" should be chosen so that model will run with the thinnest layer permitted. (Default value is 300)

AZG_minimum_dz_threshold – the width of a layer must be larger than this threshold or it will not be divided (units = meters, default value is 20 m).

AZG_maximum_z_threshold - only layers whose mid-point is below this altitude can be divided (or removed). The only exception is that layers just above this threshold can be divided to ensure that neighboring layers never differ in width by more than a factor of 2.

AZG_remove_minimum_time - Minimum number of seconds that must elapse from the time a layer is created before it can be removed. Note layers are ONLY added or removed every "AZG_nsteps" steps regardless of the value of this parameter.

AZG_delta_tvz_threshold - threshold value in Δ potential used to determine whether or not to insert / divide a layer when using the potential tempurature AZG scheme #1. The thresohold to remove is 1/5 this value. (units = Kelvin, default value is 0.5 K)

AZG_sgs_to_total_water_flux_ratio_threshold - threshold value in SGS/total water flux used to determine whether or not to insert / divide a layer when using AZG scheme #2 or #3. The thresohold to remove is 1/10 this value. (default value is 0.1).

AZG_sgs_to_total_tke_ratio_threshold - threshold value in SGS/total TKE used to determine whether or not to insert / divide a layer when using AZG scheme #3. The thresohold to remove is 1/4 this value. (default value is 0.2).

Converting code to work with AZG Scheme

Converting code that works with SAM6.10.2 to work with the SAM6.10.2_AZG is generally quite easy. The primary difference is that the variables "nzm" and "nz" no longer exist. These have been replaced with "nzm_max", "nzm_used", "nz_max" and "nz_used". Here "max" references to the maximum number of layers there could be and "used" is the number of layers actually being used.

In general, you will to allocate space and initialize variables for the maximum possible number of layers (nzm_max or nz_max) but loop only over active layer (1 to nzm_used or 1 to nz_used). Note: nzm_used or nz_used is the top of the grid.

There is also a global flag called "changed_layers_flag" that will be set to .true. for the one time step afterwhich the vertical grid has been altered.

So for example,

```
Subroutine mysub(varin)
real varin(nzm)
real mystat(nz)

        if(nstep==0) then
                ! might need to do something like set ozone concentration ?
                call do_something_special_at_startup_that_depends_on_grid()
        end

        do loop=1,nzm
                mystat(loop)=variin(loop)
        end do
        mystat(nz)=1/2;         ! value of ½ at top of grid

        call hbuf_put('DumyStat',mystat,1)
end
```

becomes,

```
Subroutine mysub(varin)
real varin(nzm_max)
real mystat(nz_max)

        if(nstep==0 .or.  changed_layers_flag) then
                call do_something_special_at_startup_that_depends_on_grid()
        endif

        mystat(:)=0;    ! initialize so unused values are not random

        do loop=1,nzm_used
                mystat(loop)=variin(loop)
        end do
        mystat(nz_used)=1/2;            ! value of ½ at top of grid

        ! Note SAM hbuffer command hbuf_put is expecting an array of size nzm_max
        call hbuf_put('DumyStat',mystat,1)
end
```