From: Archie Paulson <archie.paulson@colorado.edu>
Subject: **improved climate model**
Date: February 8, 2008 5:24:29 PM MST
To: Chris Malley <cmalley@pixelzoom.com>
Cc: Wendy Kristine Adams <Wendy.Adams@colorado.edu>
3 Attachments, 91.7 KB Save Slideshow

Chris,

Based on discussion with Bob Anderson, I've created an improved climate model.

The model is detailed below. It describes how the 'temperature' and 'snowfall' sliders control the accumulation and ablation at all elevations. I've used the same numbering scheme as before.

I've attached the python code the implements it (note that this code also includes some experimental stuff in the glacier model that you should ignore). I have also attached a couple of plots of the mass balance, so you get an idea of what they look like.

Let me know if you have any questions.

-Archie

##

1. temperature control:
   (same as before)
   range from -15 to 10 (cold to warm)
   call this variable 'ref_temp' for reference temperature
   initial default ref_temp=0

2. snowfall control:
   range from -4e3 (lots of snowfall) to 3e3 (little snowfall)
   call this variable 'snow_ref_elev'
   initial default snow_ref_elev=0.0

4. accumulation:
   given
      (a) the constant snowfall_max=2.0 (m/yr)
      (b) the constant default_snow_level=4e3 (m)
      (c) the constant snow_transition_width=300 (m)
      (d) the snow_ref_elev
   the accumulation as a function of elevation (z) is
      snowfall_max* (.5+(1./pi)*
      arctan((z-snow_transition_elev)/snow_transition_width))

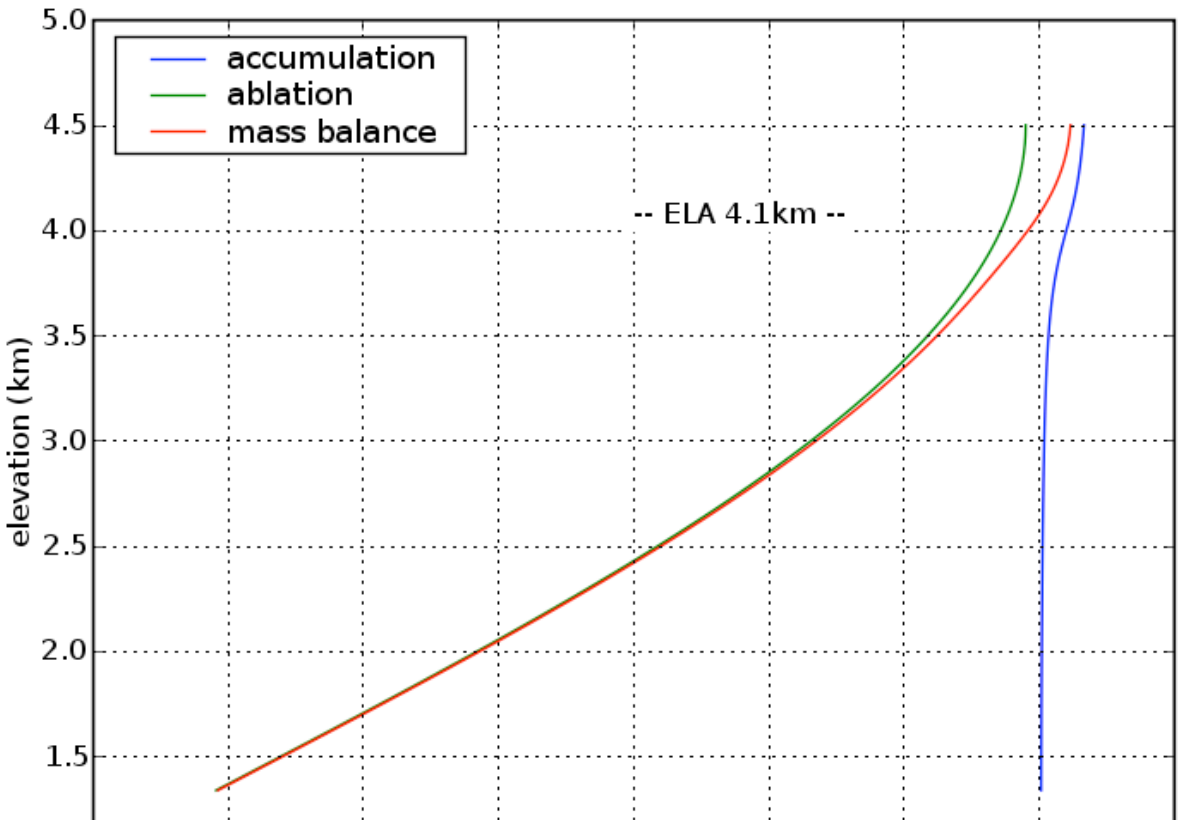
5. ablation:
   given
      (a) the constant melt_v_elev=30 (m/yr/m)
      (b) the constant melt_v_temp=200
      (c) the constants z0 and z1 which are the minimum
          and maximum elevations in the sim (m)
      (d) the ref_temp,
   first compute a temporary function of elevation (z):
      melt_v_elev* (1.-sin((z-z0-ref_temp*melt_v_temp)/((z1-z0)*2/pi)))

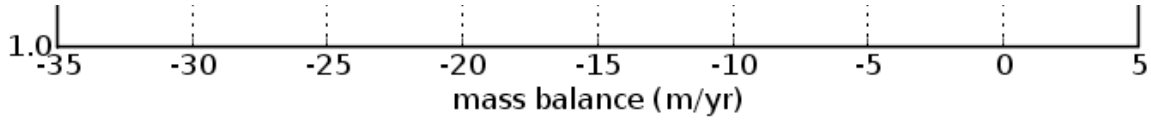
--call this function 'ablation_tmp'
--this function will have a minimum value at some high
elevation (maybe at the highest elevation); the values
of this function should be set to zero for every
elevation higher than the elevation of this minimum
--now add this offset to get the ablation function:
ablation_tmp + arctan(ref_temp/2.5)/3.+0.5

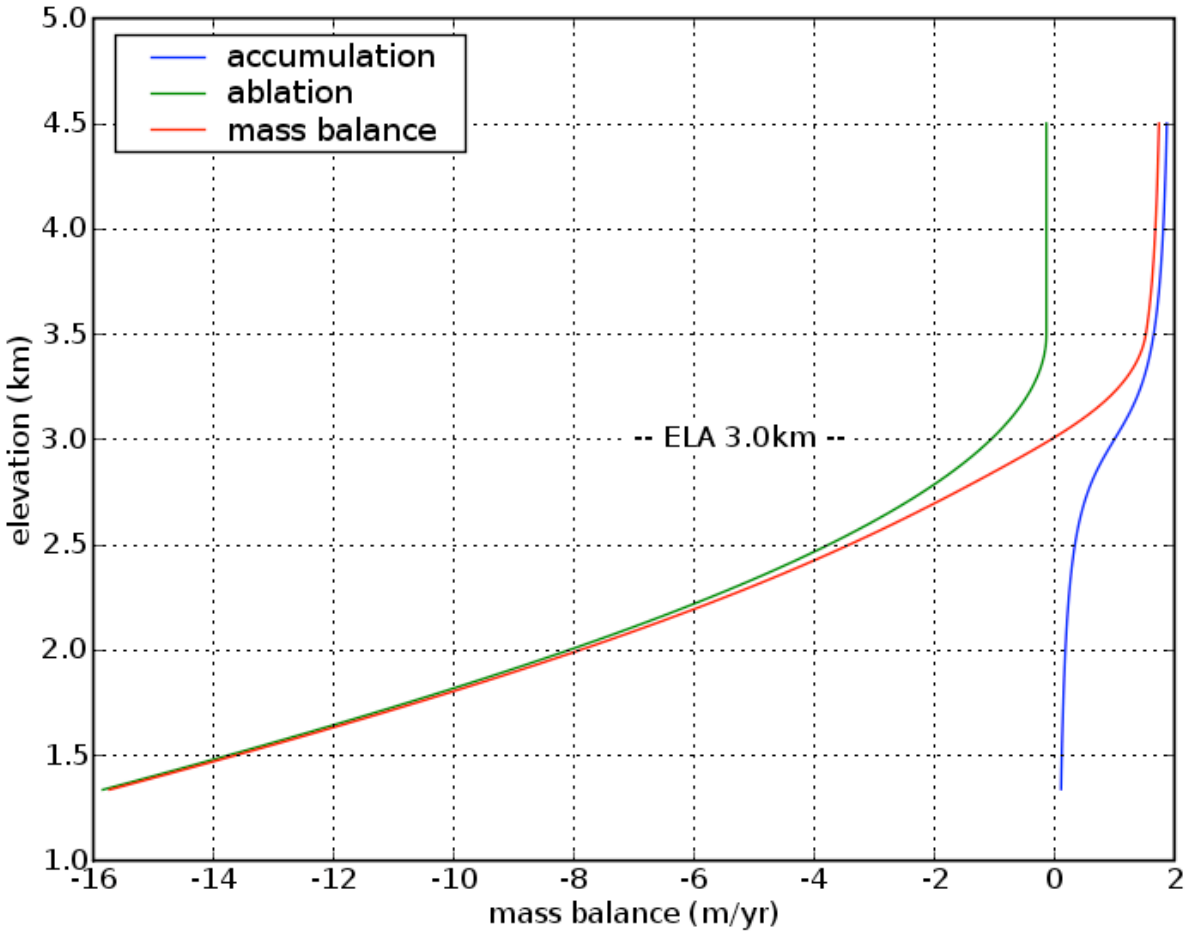
6. mass_balance:
(same as before)
given
(a) accumulation as a function of z,
(b) ablation as a function of z,
the mass_balance as a function of elevation (z) is
accumulation - ablation
note this is also called the 'glacial budget';

7. ELA:
(same as before)
given the mass_balance as a function of z,
ELA = elevation at which mass_balance=0
note: ELA stands for equilibrium line altitude; it is
the elevation of the equilibrium line

model.py (15.5 KB)




1.0
-35
-30
-25
-20
-15
-10
-5
0
5
mass balance (m/yr)


accumulation
ablation
mass balance
-- ELA 3.0km --
elevation (km)
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
-16
-14
-12
-10
-8
-6
-4
-2
0
2
mass balance (m/yr)