How to add a new variable in Hapke

# Adding a variable in the UI

1. Open the UI.html file.
2. You may want to add two kinds of variables
   1. Not dependent on the grain samples, e.g: minimization params
   2. Dependent on the grain samples. – this is more complicated. Best is to reach out to me/developer for help.
3. For non dependent variables –
   1. Find the section to add the variable
   2. Use the following template and add it to a div which has class=”param\_group”. If you want a new section altogether. Add a new <div class=”param\_group”></div> and add the below code inside your newly created div.

<div class="inp">

<label class="inplbl">

Start Points:

</label>

<input name='spts' type='number' step='1' value='5' /> </div>

* 1. Please follow the class names for the “div” and “label” tags. This is what will keep them in alignment with the existing design
  2. <input> is the actual place where you would receive the input. And the name variable is what it will be referred to as in the code.
  3. Please be aware – even though input is a number it will be read as a string in the code and we have to convert it to int or float. E.g: int(spts) or float(funtol)

1. Now you should be able to see this on the screen. Find the corresponding method – at the end of the section the above the submit button you will have something like….

<input name='section' type='hidden' value='repeat\_k' />

1. The function thus associated with this section is repeat\_k() found in prog\_state.py

# In Prog\_state.py

1. In the function definition – add the variable you added in the UI. If you add it to the definition you can access it as a variable. If not it will be available in the kwargs dictionary.
2. E.g: For spts – if you add to the definition like so

def repeat\_k(self, lowk=0, upk=0, maxfun = 1000, spts=30, diff\_step = 0.0001, funtol = 0.00000000000001, xtol= 0.00000000000001, \*\*kwargs):

* You can access it as spts.

1. If you don’t add it to the definition like so:

def repeat\_k(self, lowk=0, upk=0, maxfun = 1000, diff\_step = 0.0001, funtol = 0.00000000000001, xtol= 0.00000000000001, \*\*kwargs):

* You have to read it from kwargs like : spts = int(kwargs[‘spts’])