Implementation order of BMI methods

I've found it helpful to implement BMI methods in a certain order. In the following sections, I list my preferred order, as well as any notes on why I chose the ordering. For reference, an alphabetical list of BMI methods is given here.

Basic info methods

1. get_component_name
2. get_input_var_names
3. get_output_var_names

These methods are usually straightforward to implement. They don't require that a model be initialized before calling them. The ***names*** methods are used by the ***var*** and ***grid*** methods.

Lifecycle methods

4. initialize
5. finalize

Almost all remaining BMI methods rely on the model being initialized.

Time methods

6. get_start_time
7. get_end_time
8. get_current_time
9. get_time_step
10. get_time_units

The ***time*** methods are used by the ***update*** methods.

Update methods

11. update
12. update_frac
13. update_until

Implementation of these methods depend on how the model advances itself; e.g., *update_until* may call *update* and *update_frac*.

The get_var_grid method

14. get_var_grid

This method has its own heading because even though it's a ***var*** method, it provides the grid identifier(s) used by the ***grid*** methods.

Grid methods

1. get_grid_type
2. get_grid_rank
3. get_grid_shape
4. get_grid_size
5. get_grid_spacing
6. get_grid_origin

Which of the ***grid*** methods are implemented will depend on the grid type (unstructured, structured quads, rectilinear, uniform rectilinear).

Var methods

1. get_var_type
2. get_var_units
3. get_var_itemsize
4. get_var_nbytes

Getter methods

1. get_value
2. get_value_ref
3. get_value_at_indices

Recall that these methods should return flattened arrays.

Setter methods

1. set_value
2. set_value_at_indices

It's alright to clobber the reference to the existing values when setting new values. (Otherwise, we could implement a *set_value_ref* method.)