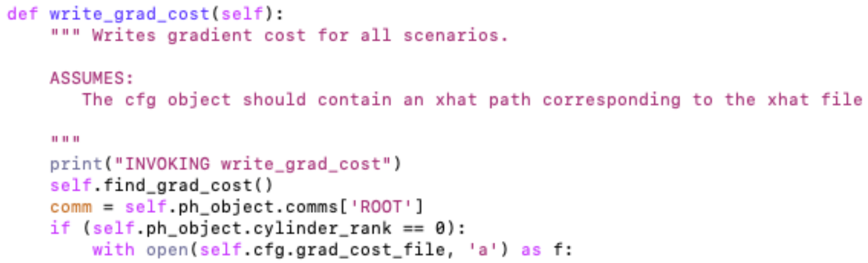
More info on grad/rho issues

Watson, Jean-Paul <watson61@llnl.gov>
To: David Woodruff <david.l.woodruff@gmail.com>

Tue, Jul 2, 2024 at 8:46 AM

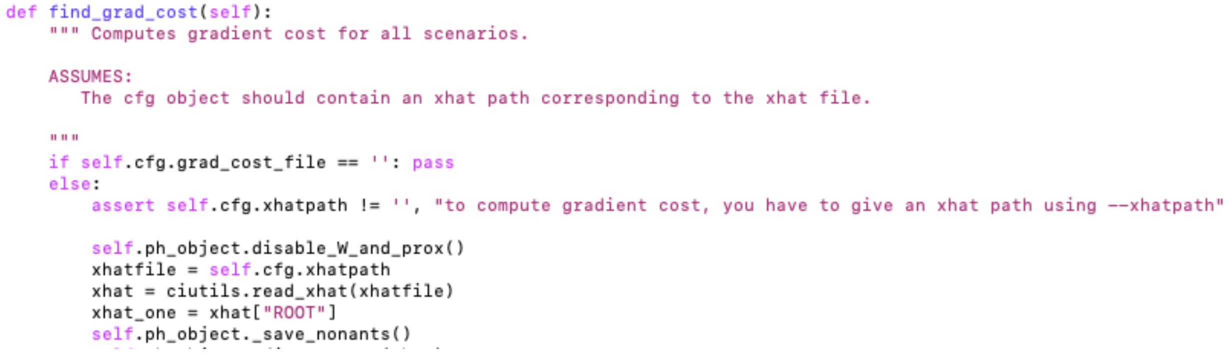
And last bit for a while…

Nit #1: I claim that "write_grad_cost" should only write, not compute:



```
def write_grad_cost(self):
    """ Writes gradient cost for all scenarios.

    ASSUMES:
       The cfg object should contain an xhat path corresponding to the xhat file

    """
    print("INVOKING write_grad_cost")
    self.find_grad_cost()
    comm = self.ph_object.comms['ROOT']
    if (self.ph_object.cylinder_rank == 0):
        with open(self.cfg.grad_cost_file, 'a') as f:
```

Nit #2: File presence should not dictate whether gradients are computed

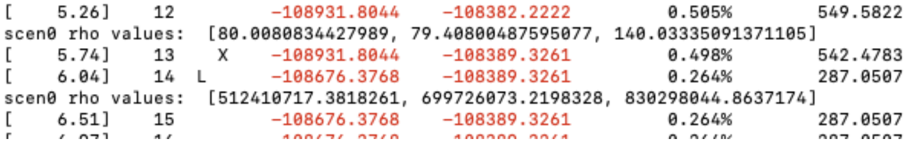


```
def find_grad_cost(self):
    """ Computes gradient cost for all scenarios.

    ASSUMES:
       The cfg object should contain an xhat path corresponding to the xhat file.

    """
    if self.cfg.grad_cost_file == '': pass
    else:
        assert self.cfg.xhatpath != '', "to compute gradient cost, you have to give an xhat path using --xhatpath"

        self.ph_object.disable_W_and_prox()
        xhatfile = self.cfg.xhatpath
        xhat = ciutils.read_xhat(xhatfile)
        xhat_one = xhat["ROOT"]
        self.ph_object._save_nonants()
```

The above could of course be why gradients are not presently (maybe) being computed.

From: Watson, Jean-Paul <watson61@llnl.gov>
Date: Tuesday, July 2, 2024 at 8:33 AM
To: David Woodruff <david.l.woodruff@gmail.com>
Subject: Re: More info on grad/rho issues

Quick follow-up – check out the following output snippet from the run:

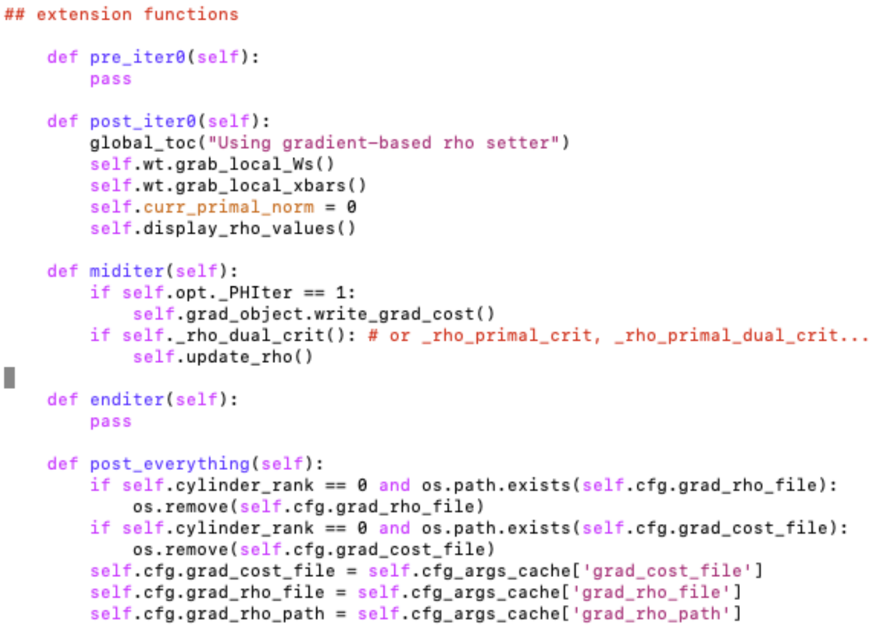


```
[     5.26]     12         -108931.8044     -108382.2222          0.505%        549.5822
scen0 rho values:  [80.0080834427989, 79.40800487595077, 140.03335091371105]
[     5.74]     13     X   -108931.8044     -108389.3261          0.498%        542.4783
[     6.04]     14   L     -108676.3768     -108389.3261          0.264%        287.0507
scen0 rho values:  [512410717.3818261, 699726073.2198328, 830298044.8637174]
[     6.51]     15         -108676.3768     -108389.3261          0.264%        287.0507
```

So at least the rhos are being updated, but I don't see how the gradients could be. Weirdness.

From: Watson, Jean-Paul <watson61@llnl.gov>
Date: Tuesday, July 2, 2024 at 8:29 AM
To: David Woodruff <david.l.woodruff@gmail.com>
Subject: More info on grad/rho issues

This is where I think things get weird, in gradient_extension.py:



```
## extension functions

    def pre_iter0(self):
        pass

    def post_iter0(self):
        global_toc("Using gradient-based rho setter")
        self.wt.grab_local_Ws()
        self.wt.grab_local_xbars()
        self.curr_primal_norm = 0
        self.display_rho_values()

    def miditer(self):
        if self.opt._PHIter == 1:
            self.grad_object.write_grad_cost()
        if self._rho_dual_crit(): # or _rho_primal_crit, _rho_primal_dual_crit...
            self.update_rho()

    def enditer(self):
        pass

    def post_everything(self):
        if self.cylinder_rank == 0 and os.path.exists(self.cfg.grad_rho_file):
            os.remove(self.cfg.grad_rho_file)
        if self.cylinder_rank == 0 and os.path.exists(self.cfg.grad_cost_file):
            os.remove(self.cfg.grad_cost_file)
        self.cfg.grad_cost_file = self.cfg_args_cache['grad_cost_file']
        self.cfg.grad_rho_file = self.cfg_args_cache['grad_rho_file']
        self.cfg.grad_rho_path = self.cfg_args_cache['grad_rho_path']
```

Specifically the "miditer" call. This is at a minimum weird, either pre- or post-hack (I'm not sure whatever I did "stuck" here). But the gradients should be computed more than once, and are – but I'm not sure how, given this logic.

Also: I was running the farmer_rho_demo command line (second one in the file). That runs "fine" (or at least executes) with what's in the repository presently. Not sure what it's doing, though…