Michael Raitor

Part 1: Algorithm Description

I based my algorithm on Nesterov's momentum approach and made 3 important modifications:

1. Adding a threshold and normalization term, where if the next step (calcualted with the following logic:

   ```
   If norm(v) > threshold
           v = v / norm(v)
   End
   Where v is the momentum/velocity term
   ```

   This addition to the algorithm prevented divergence/instability on simple1, when the gradient (and consequently the step size for this gradietn-descent method) was too large

2. I used a learning rate that decreased with step count rather than a constant learning rate. This is often sed in proofs to guarantee convergence, and here it added some stability that prevented continuous oscillations in flatter regions of the cost landscape. This decreasing learning rate, alpha, was calculated using:

```
tmp_vec = xi_vec + beta*v;
 v = beta*v - (alpha^i)*g(tmp_vec); #adjusted Nestov's alg. to incorporate
decreasing learning rate
 normThresh = 1;
 if norm(v) > normThresh
        v = v / norm(v)
end
xpi_vec = xi_vec + v;
```

I always assigned alpha as 0<alpha<1, therefore with each step (i), the effect the gradient had and consequently, the step size, decreased as iterations increased

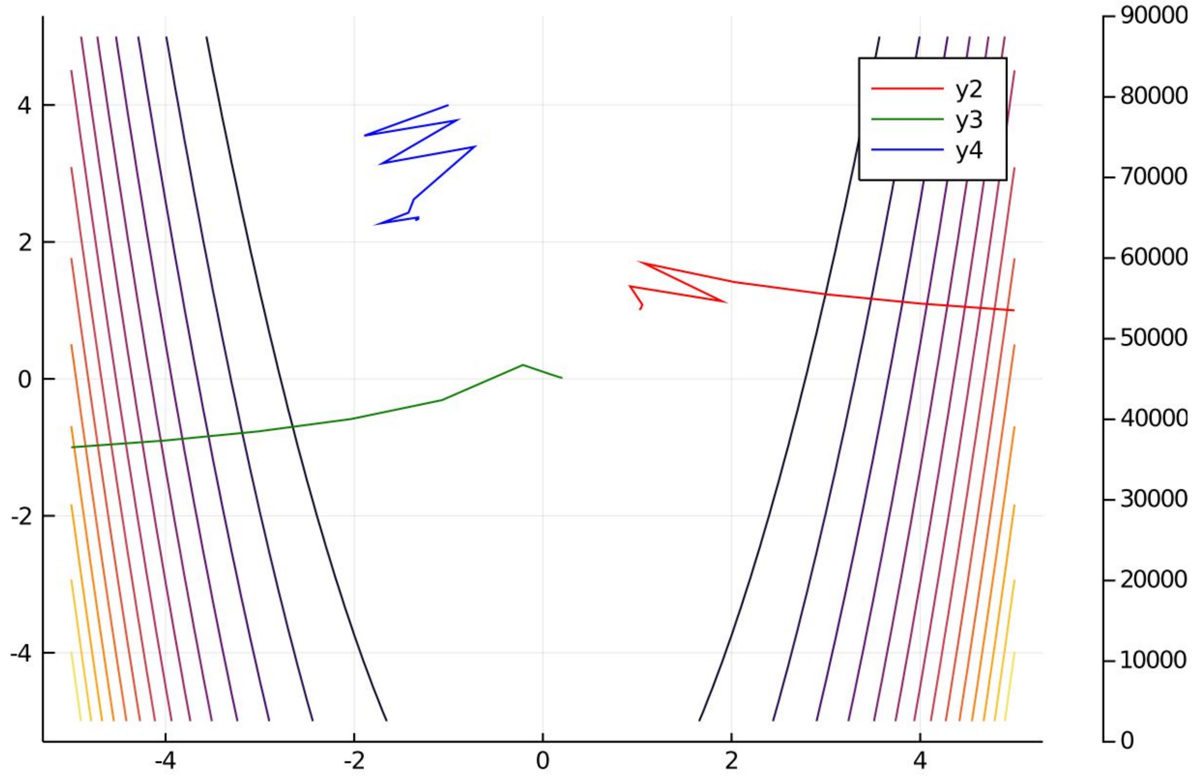
3. I added a logic step to adjust the beta (momentum decay) and (initial) alpha (learning rate) terms for simple1 problem, as the values that worked well for the other 4 problems did not converge in the steep parabola.

Michael Raitor

Part2: Contour plots and paths

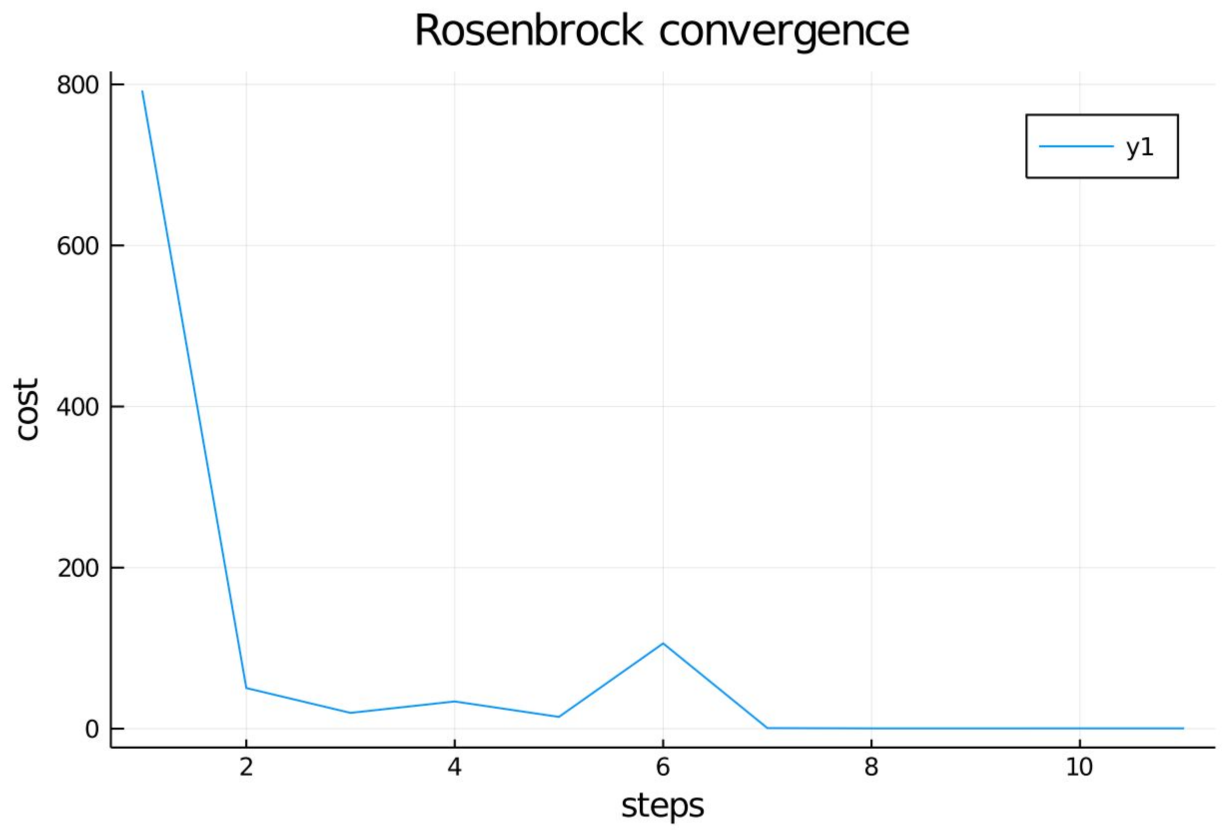
3 different initial conditions for the RosenBrock function, for y2, x0 = (5, 1), for y23 x0 = (-5, -1), for y4, x0 = (4, -1),

Clearly my algorithm found the bottom of the bowl (getting close to the optimum) when it started on the steeper gradient sections and oscillated more substantially in the flatter region.

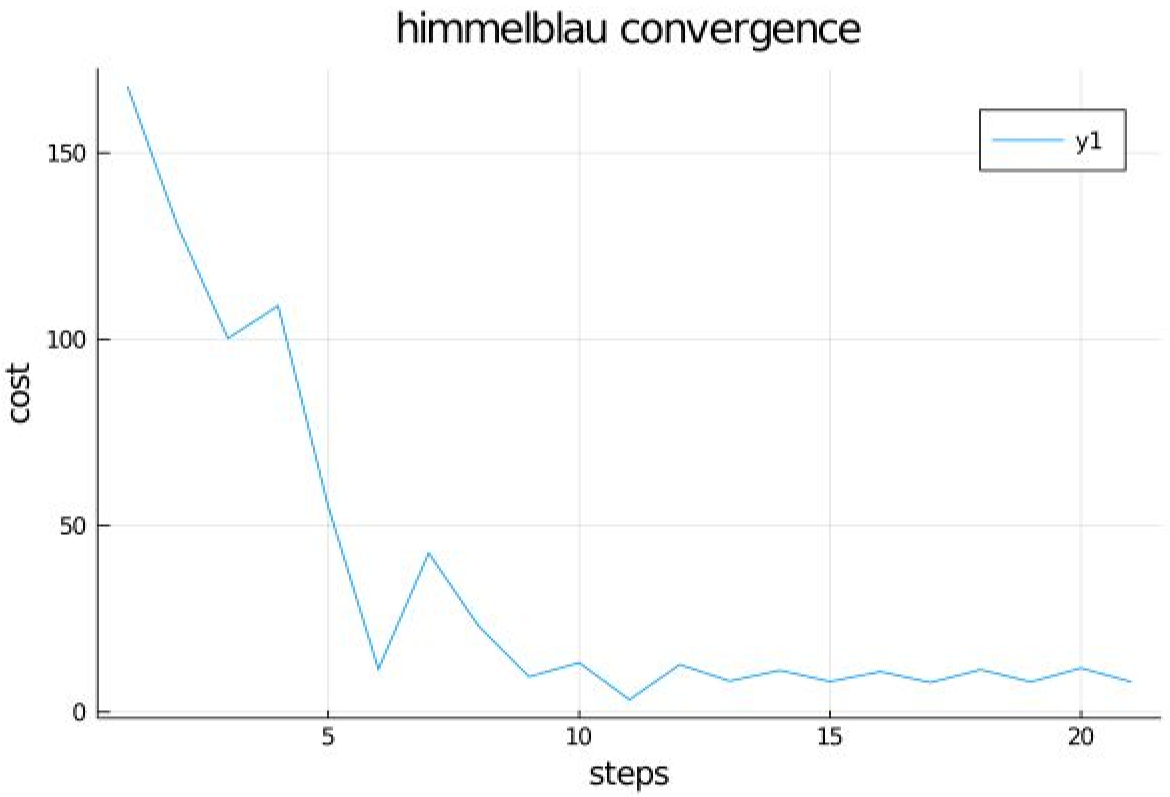


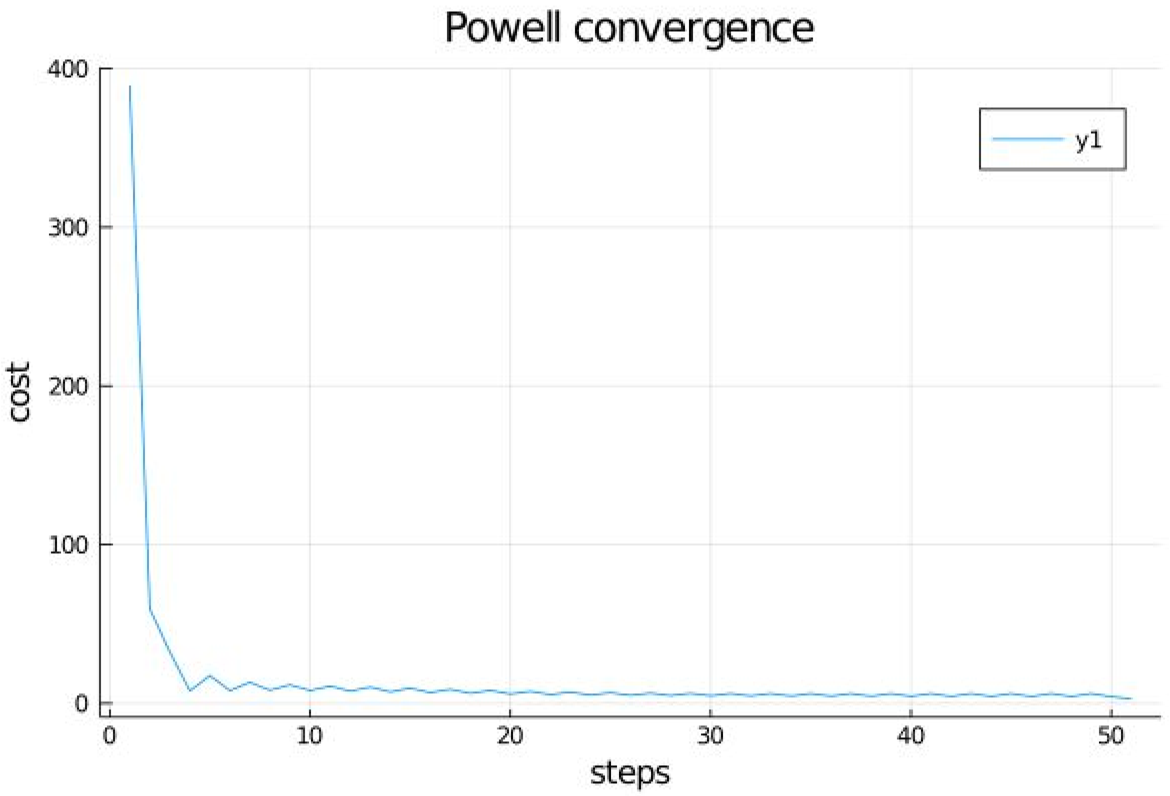
Michael Raitor

Part 3: Convergence Plots



Michael Raitor


himmelblau convergence
cost
steps
y1


Powell convergence
cost
steps
y1