Effective Programming Practices for Economists

Numerical Optimization

Derivative-Based Trust Region Algorithms

Janoś Gabler, Hans-Martin von Gaudecker, and Tim Mensinger

Basic Idea (optimagic docs)

1. Set initial trust region radius.
2. Construct a quadratic Taylor approximation of the function based on function value, gradient, and (approximation to) the Hessian.

   The Taylor approximation:

   - approximates the function well within the trust region if radius is not too large
   - is a quadratic function that it easy to optimize.
3. Minimize the Taylor approximation within the trust region.
4. Evaluate the function again at the argument that minimized the Taylor approximation.

5. Compare expected and actual improvement.
    - Expected improvement is the decrease in the criterion according to the Taylor approximation.
    - Actual improvement is the decrease in the actual function value.
6. Accept the new parameters if actual improvement is good enough.
7. Modify the trust region radius (**important and complex step**).
8. Construct a new Taylor approximation …

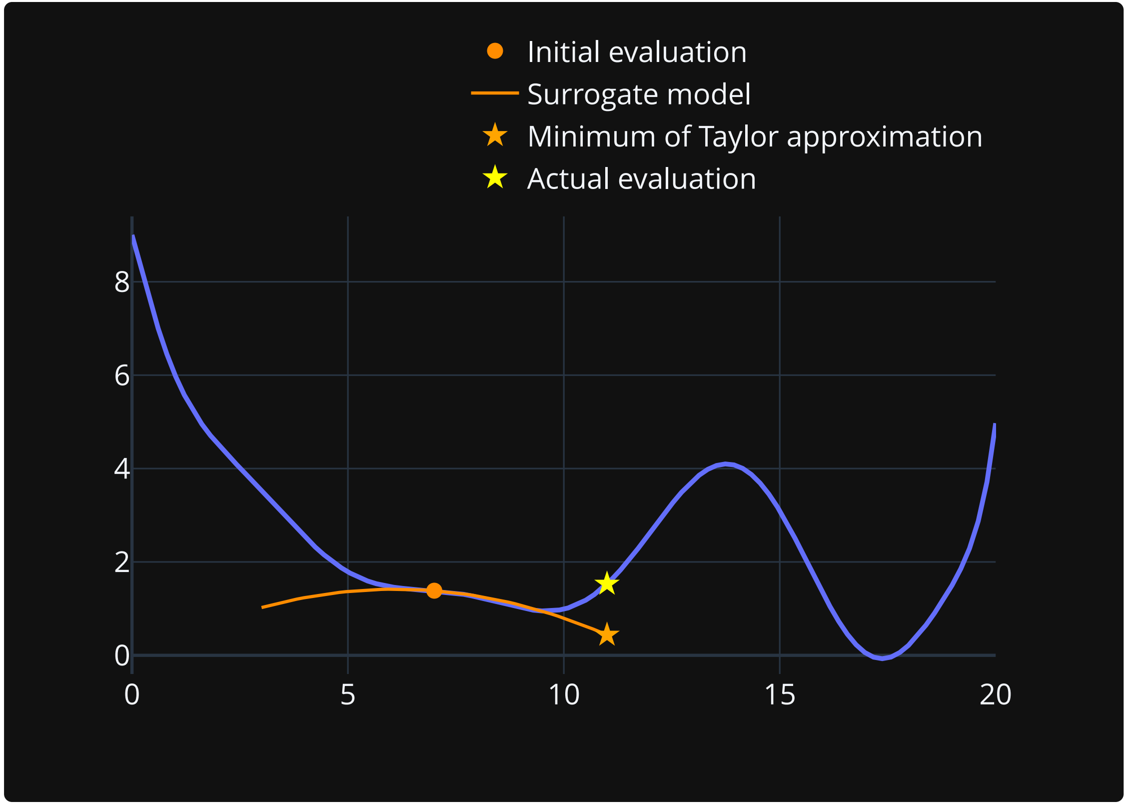
Smaller radii $\Rightarrow$ better approximations

- For a step s, the Taylor expansion of $f(x+s)$ around x satisfies:

  $$f(x+s) = f(x) + f'(x)^\top s + \tfrac{1}{2} s^\top f''(x) s + o(\|s\|^2).$$

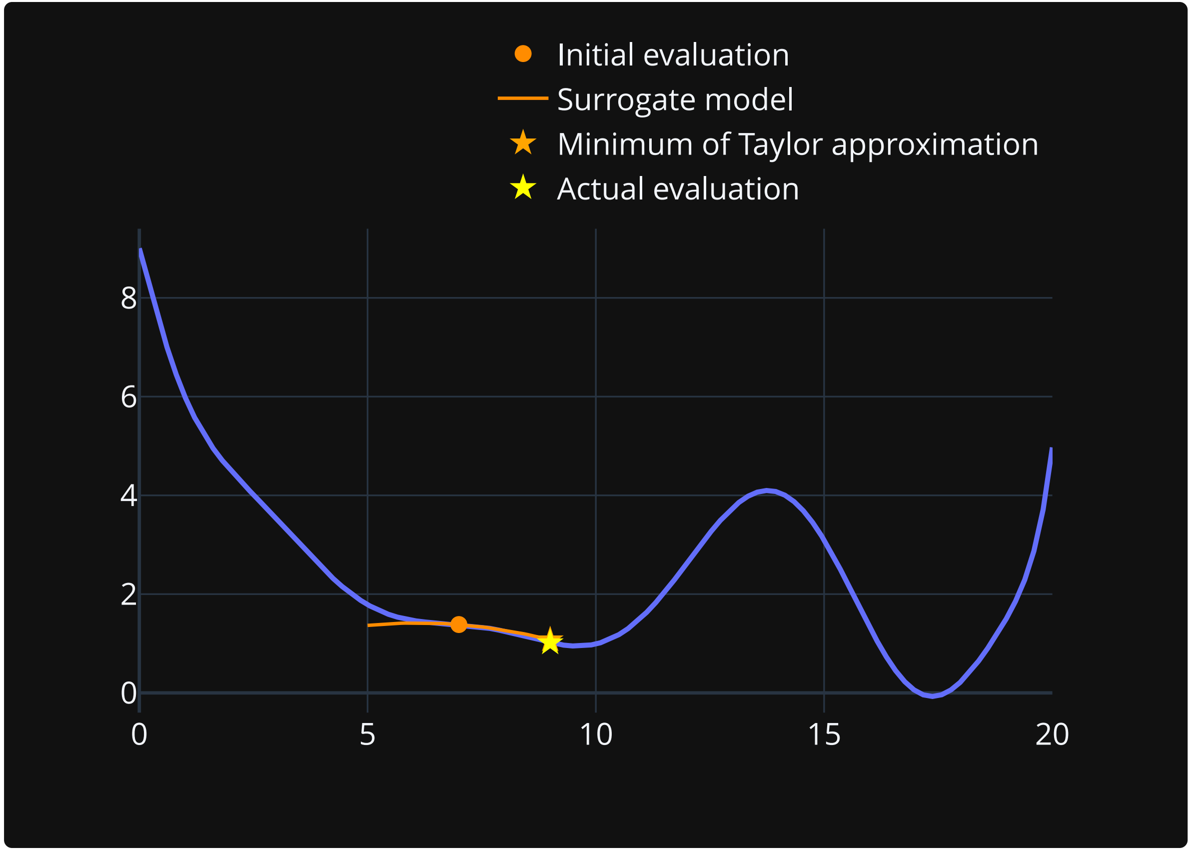
- The step s is bounded by the trust region radius Δ: $\|s\| \leq \Delta$.
- And therefore, as Δ decreases the approximation error $o(\|s\|^2)$ decreases.
- (Holds for any function f that is at least twice continuously differentiable.)

Iteration 2



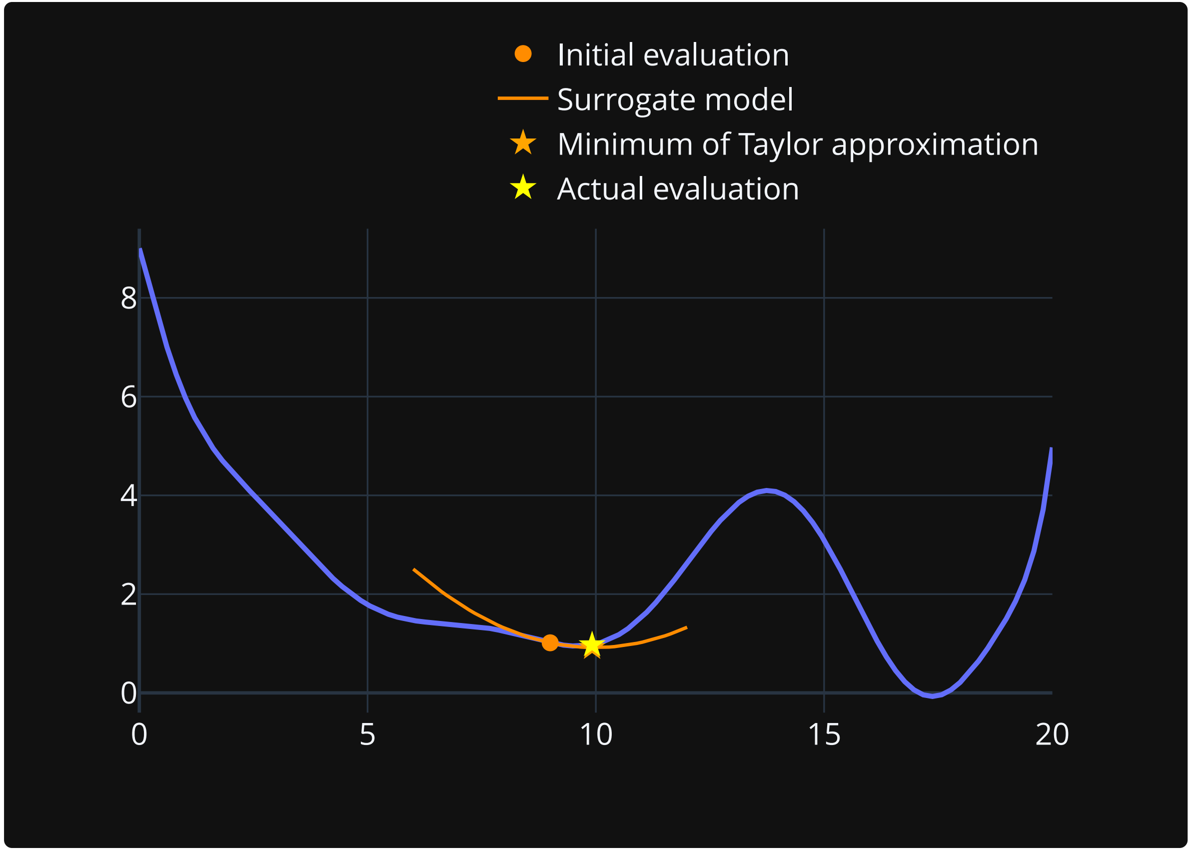
Actual improvement negative ⇒ reject, decrease radius

Iteration 3



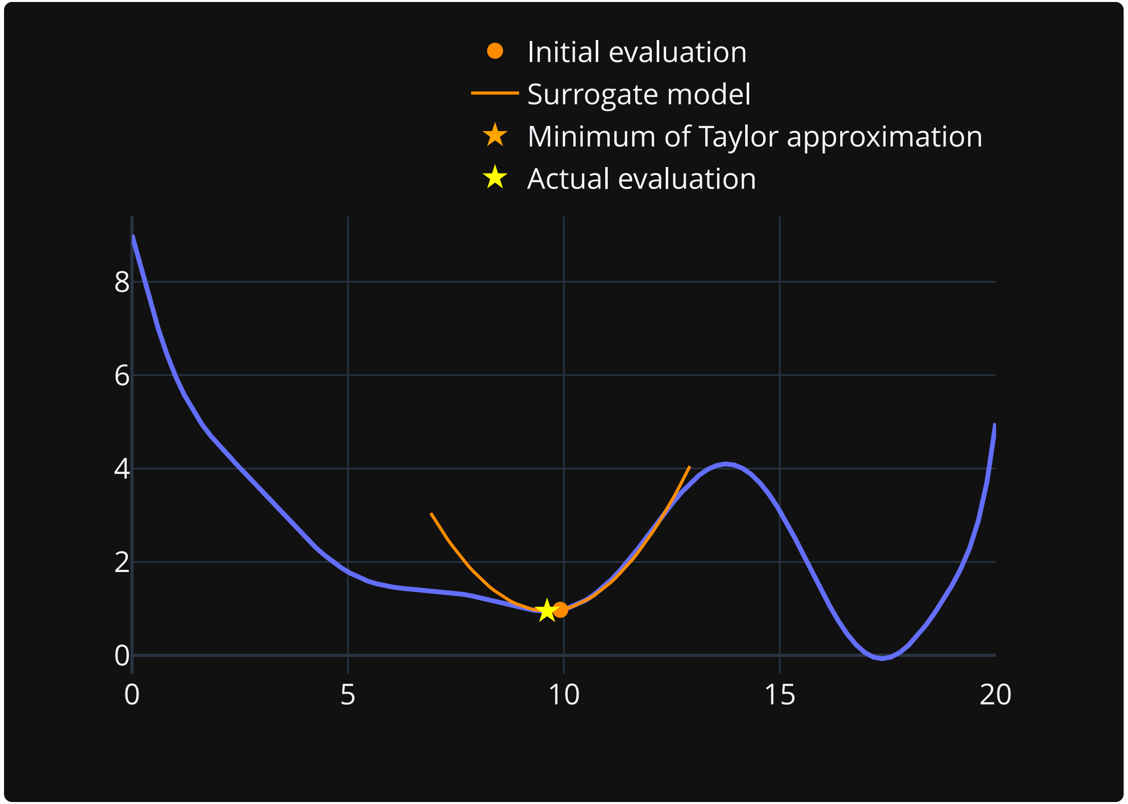
Actual improvement $\approx$ expected improvement $\Rightarrow$ accept, increase radius

Iteration 4



Actual improvement $\approx$ expected improvement $\Rightarrow$ accept, increase radius

Iteration 5



Converge around here because gradient is close to zero.

A real algorithm: fides