Using Approximations to Accelerate Engineering Design Optimization

Virginia Torczon
Department of Computer Science
College of William & Mary
and
ICASE, NASA Langley Research Center

Michael Trosset
Department of Mathematics
College of William & Mary
and
ICASE, NASA Langley Research Center

Problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in [\ell, u], \end{array}$$

where $x \in [\ell, u]$ denotes that every component of the p-dimensional vector x is greater than or equal to the corresponding component in ℓ and less than or equal to the corresponding component in u.

In other words, there are bounds on all the engineering design or decision variables x.

Operating Assumptions:

- Evaluation of the objective f depends on the output of complicated computer simulations involving large numbers of state or system variables.

  The objective f is *not* a simple algebraic expression exclusively in terms of the engineering design variables x.

- The objective is expensive to compute because the simulations upon which its value is based are expensive to compute.

  The strategy we outline is amenable to other settings where the evaluation of the objective may be expensive for other reasons.

Consequences:

- Simulations of complex physical processes can lead to:
  - nonstochastic errors in the evaluation of the objective due to the approximation, rounding, and truncation introduced by the simulations and
  - high-frequency, low-amplitude distortions in the "true" value of the objective,

  features which can cause difficulties for higher-order optimization methods.

- We must be frugal in our evaluation of the objective.

Goals:

Given a modest number of evaluations of the objective f (a "budget") produce:

1. a better "picture" of the behavior of the objective in the region(s) of interest and

2. do so in a way that is guaranteed—in theory—to produce at least one subsequence of iterates that converges to a local constrained stationary point of the function.

Fundamental Idea:

Use a *sequence* of surrogate objectives to *predict* optimal engineering design candidates.

Fundamental Perspective:

- Concentrate on the *predictive* ability of the surrogate.

- Concern ourselves with *accuracy* only in the vicinity of a minimizer.

Approximations versus Models:

Our basic strategy should work with any type of surrogate for the objective.

We distinguish between

- *models*, which are auxiliary simulations that are less physically faithful, but also less computationally expensive, and
- *approximations*, which are algebraic summaries obtained from known values of the objective.

For the purposes of this presentation, we will focus on approximations since our experience has been with this class of surrogates.

Basic Strategy:

1. Perform an initial computer experiment:
   (a) select N initial experimental design sites,
   (b) evaluate the true objective f at the initial experimental design sites, and
   (c) construct an initial approximation a_c of f based on the values of f obtained at the experimental design sites.

2. Choose:
   (a) an initial grid over the feasible region $[\ell, u]$ that signifies the degree of resolution desired (this can be refined later, as deemed appropriate) and
   (b) an initial feasible baseline engineering design x_c at which f is known.

3. Do until the "budget" of evaluations has been exhausted (or an interesting engineering design has been found):
   (a) Find a candidate x_t that *improves* (perhaps *minimizes*) a_c on the grid and treat x_t as a site at which a_c *predicts* improvement for f on the grid.
   Evaluate $f(x_t)$.
   (b) Update the approximation a_c to include the objective value $f(x_t)$.
   (c) If $f(x_t) < f(x_c)$ then let $x_c = x_t$.
   (d) Repeat Step 3.

First Example:

Test Problem:

$$\begin{array}{ll} \text{minimize} & f(x) = e^{-2x}\sin(10\pi x) \\ \text{subject to} & x \in [0, 0.5], \end{array}$$

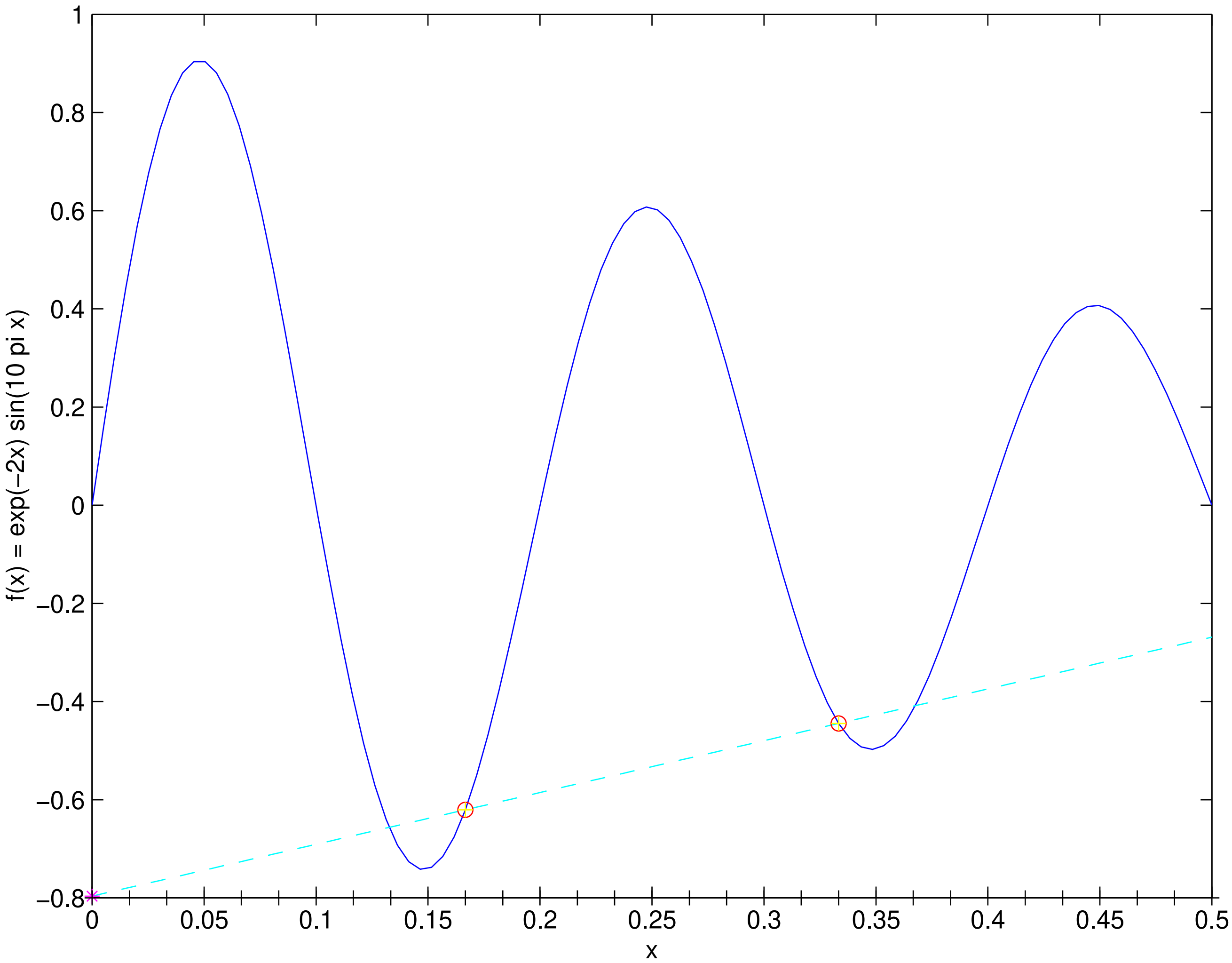
Start with *two* initial experimental design sites (N = 2), at 0.167 and 0.333, and build a *linear* interpolant a_c.

Choose whichever of the two initial experimental design sites has the lower value of the objective to be the baseline engineering design x_c.

After the first optimization step, when we have three known values of f, build a *quadratic* interpolant a_c.

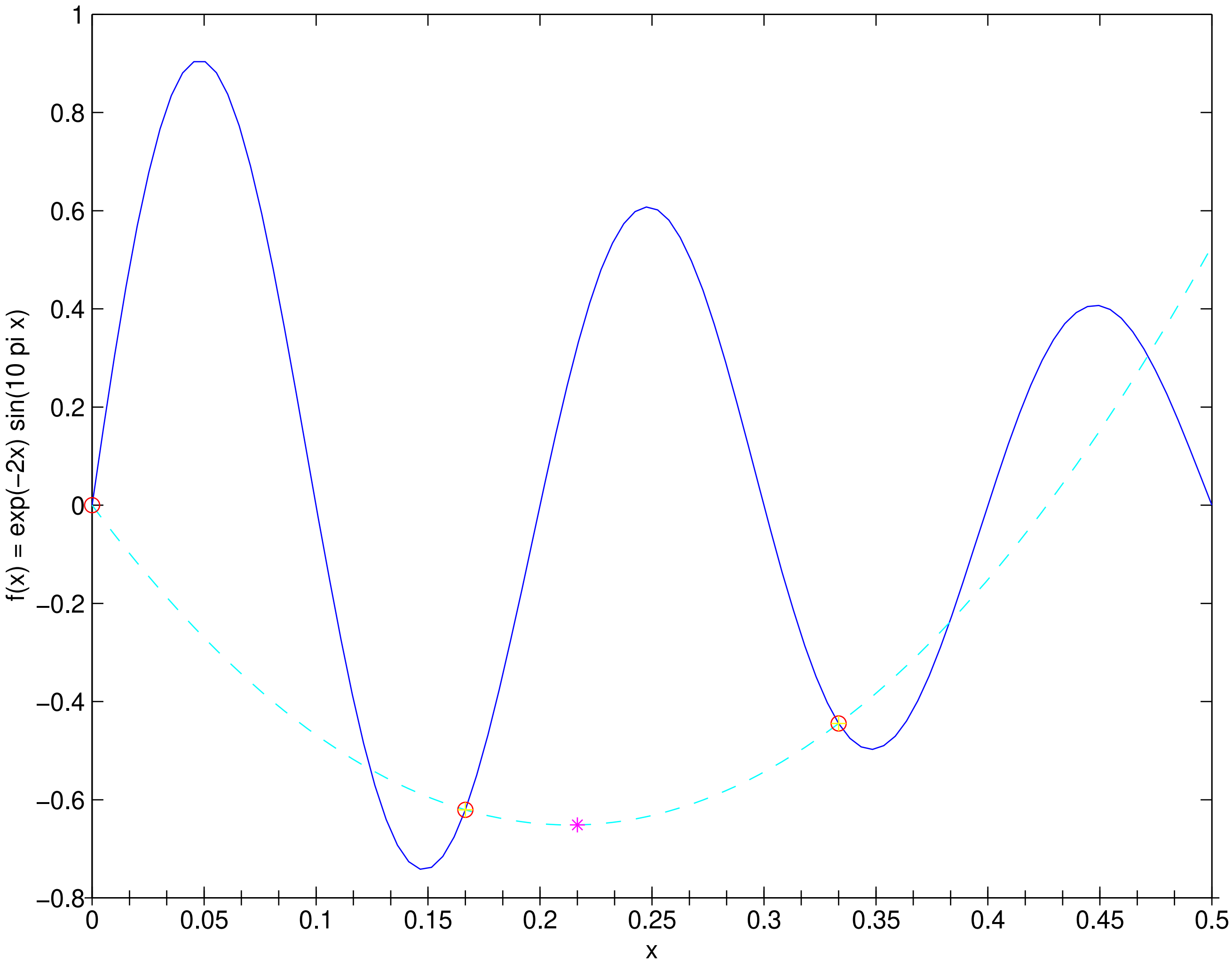
For subsequent iterations, build a *cubic spline* interpolant a_c.

Iteration 1:

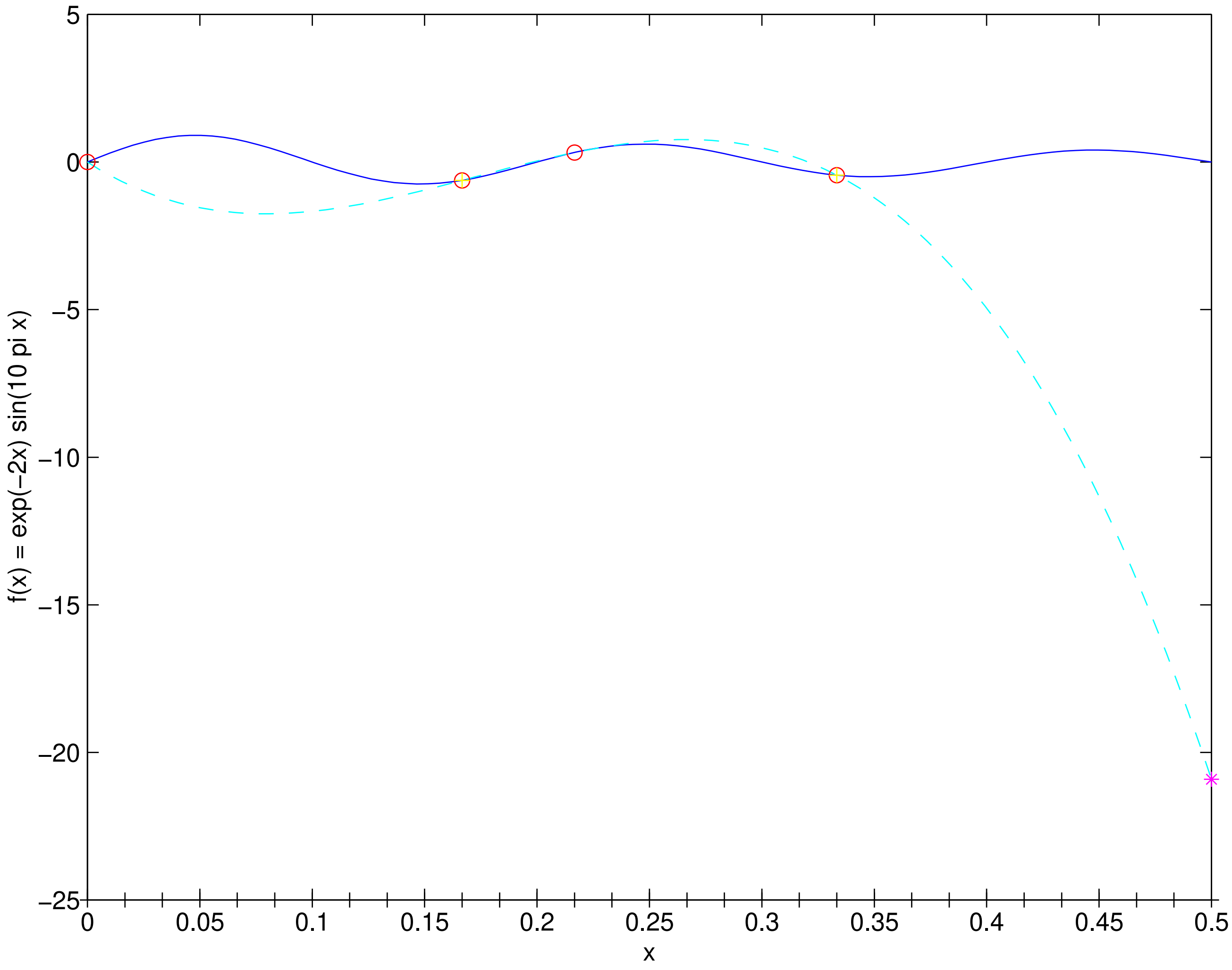


f(x) = exp(−2x) sin(10 pi x)
x
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

Iteration 2:

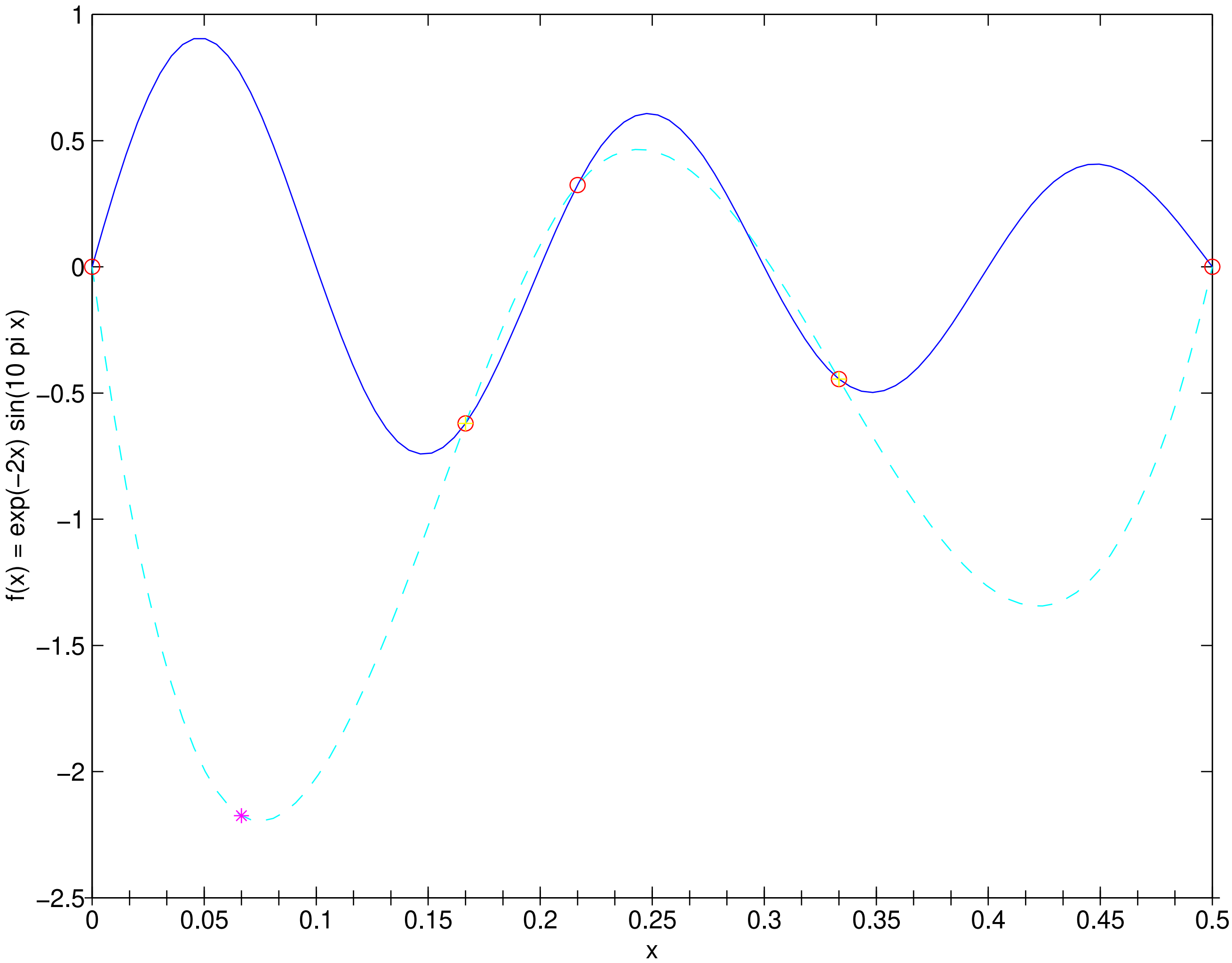

f(x) = exp(−2x) sin(10 pi x)
x
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

Iteration 3:



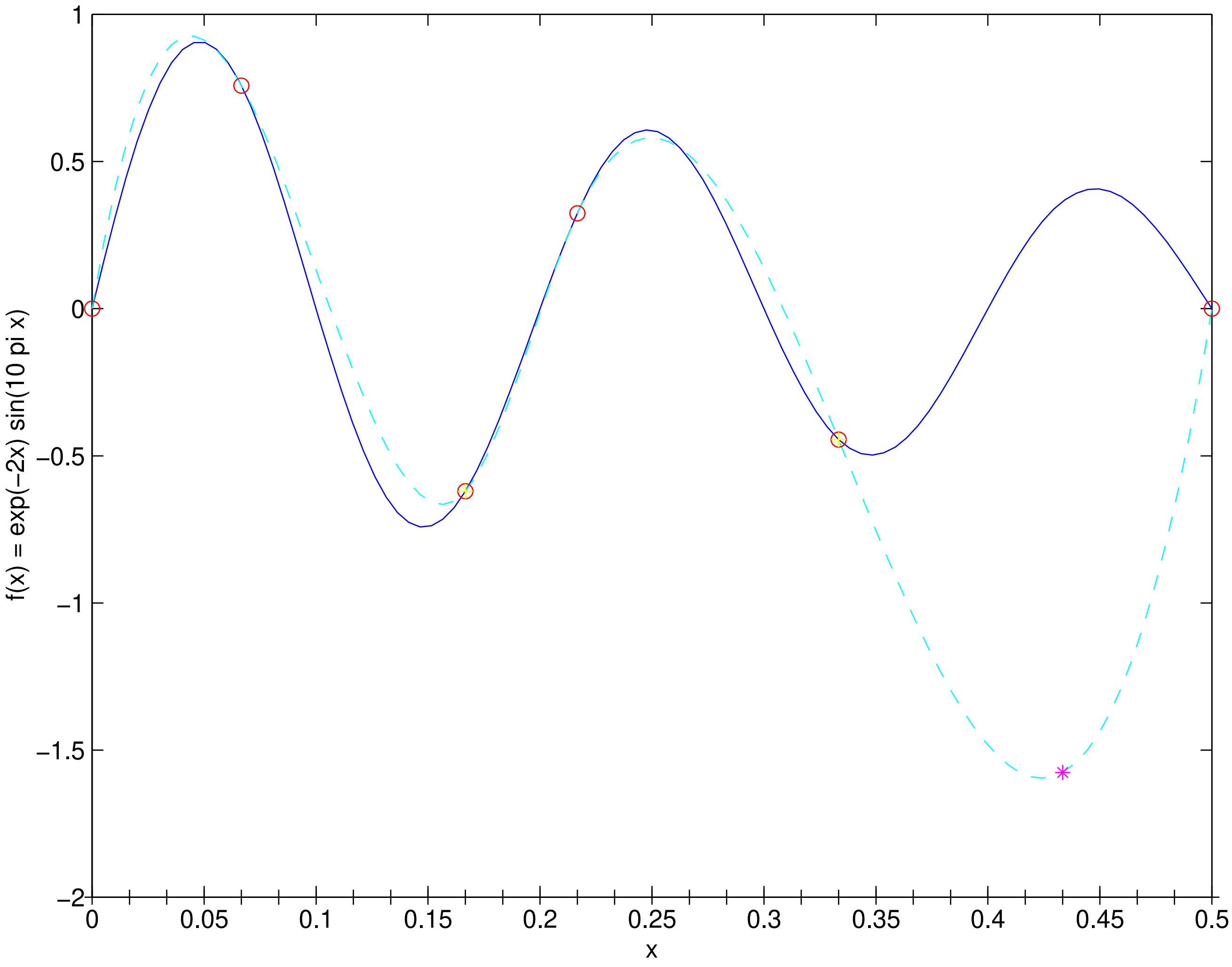
f(x) = exp(−2x) sin(10 pi x)
x
5
0
−5
−10
−15
−20
−25
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

Iteration 4:



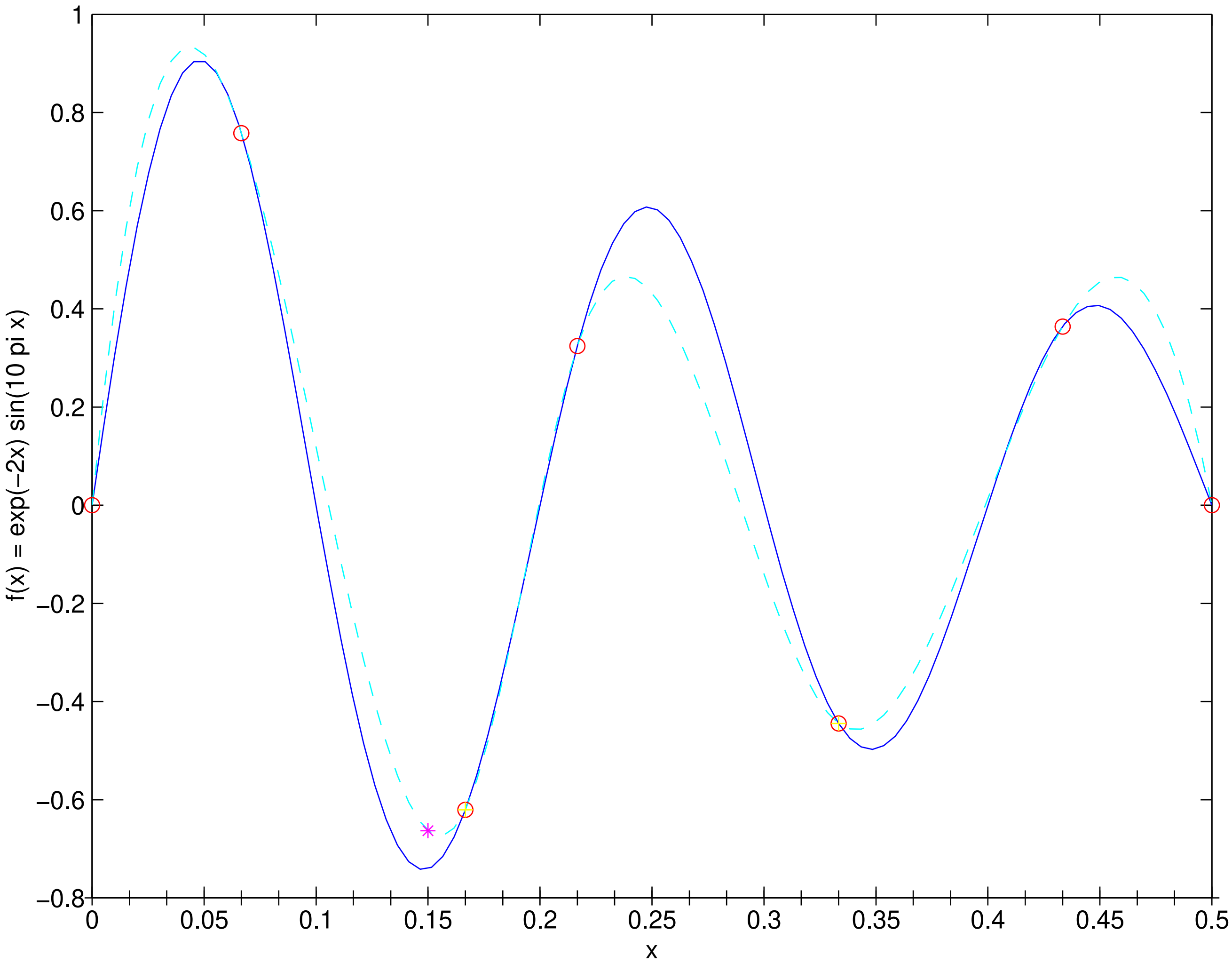
f(x) = exp(−2x) sin(10 pi x)
x
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

Iteration 5:



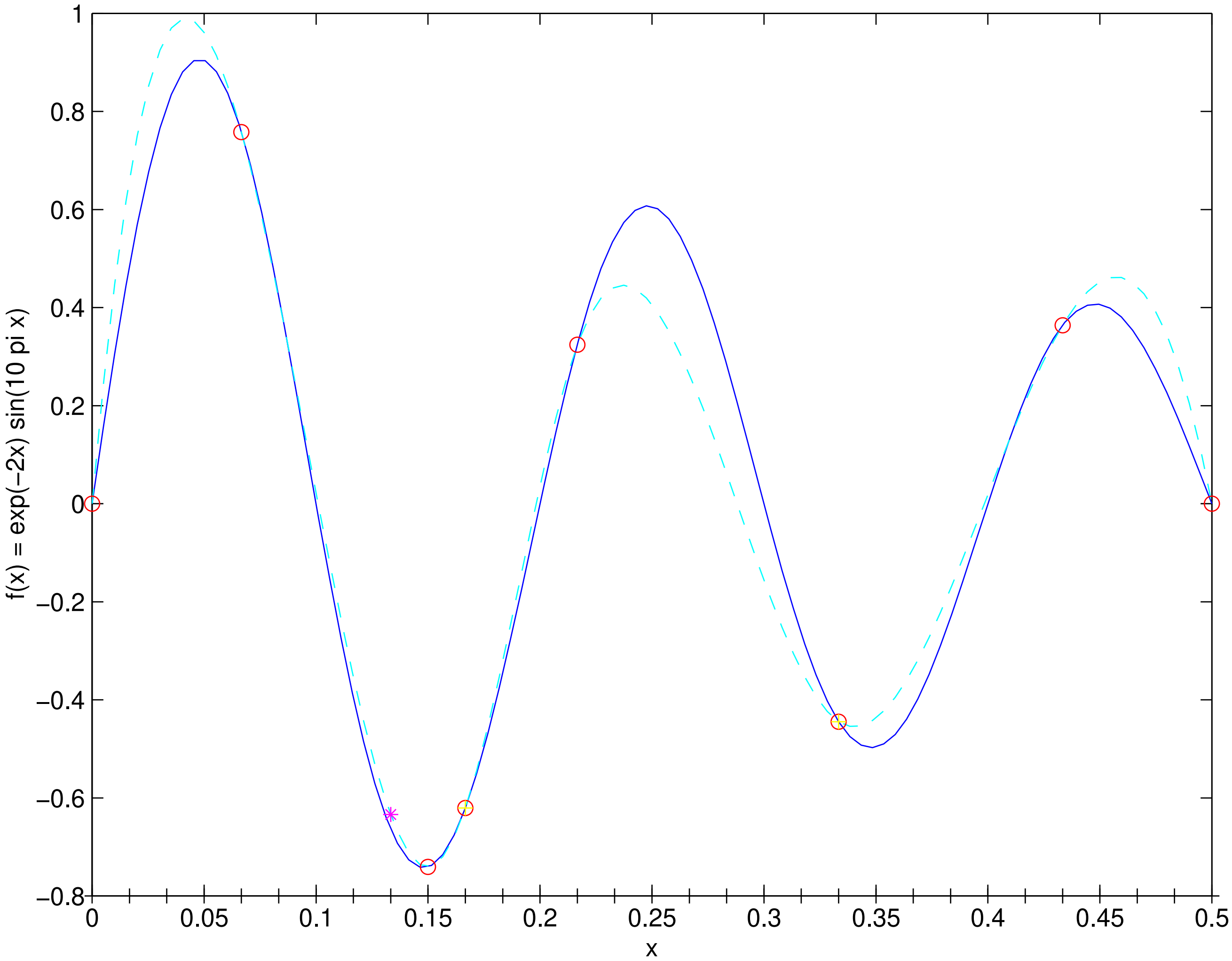
1
0.5
0
−0.5
−1
−1.5
−2
f(x) = exp(−2x) sin(10 pi x)
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
x

Iteration 6:



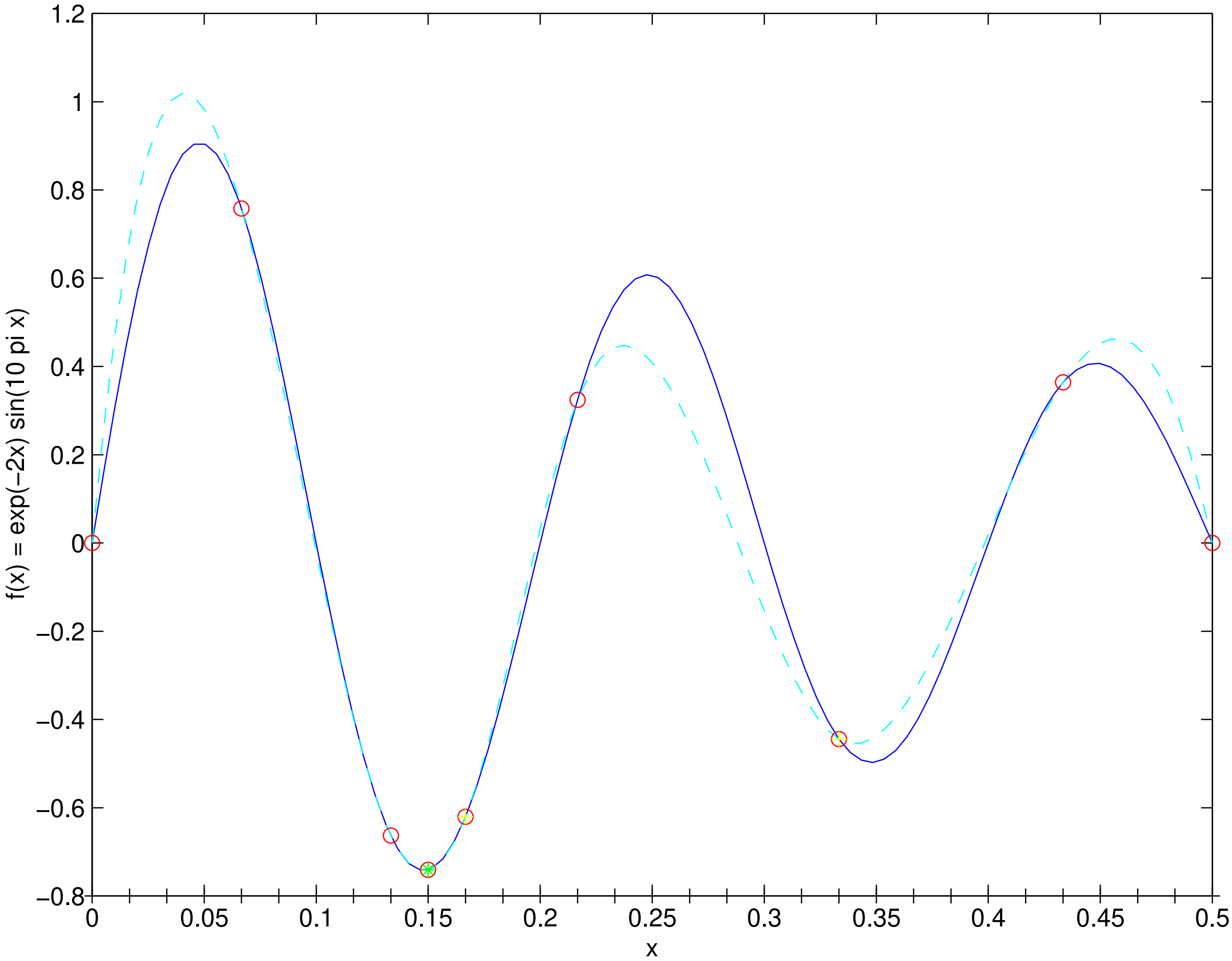
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
f(x) = exp(−2x) sin(10 pi x)
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
x

Iteration 7:



f(x) = exp(−2x) sin(10 pi x)
x
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

Iteration 8:



f(x) = exp(−2x) sin(10 pi x)
x
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

Summary for First Example:

Consider the sequence of current solutions x_c versus trial solutions x_t that we generated at each iteration:

Iteration	x_c	x_t
1	0.167	0.000
2	0.167	0.217
3	0.167	0.500
4	0.167	0.067
5	0.167	0.433
6	0.167	0.150
7	0.150	0.133
8	0.150	—

Second Example:

Let's consider a second example, which has been constructed to show that our strategy, while *provably* robust, may not always produce a *global* minimizer—which is often the preferred solution.

Test Problem:

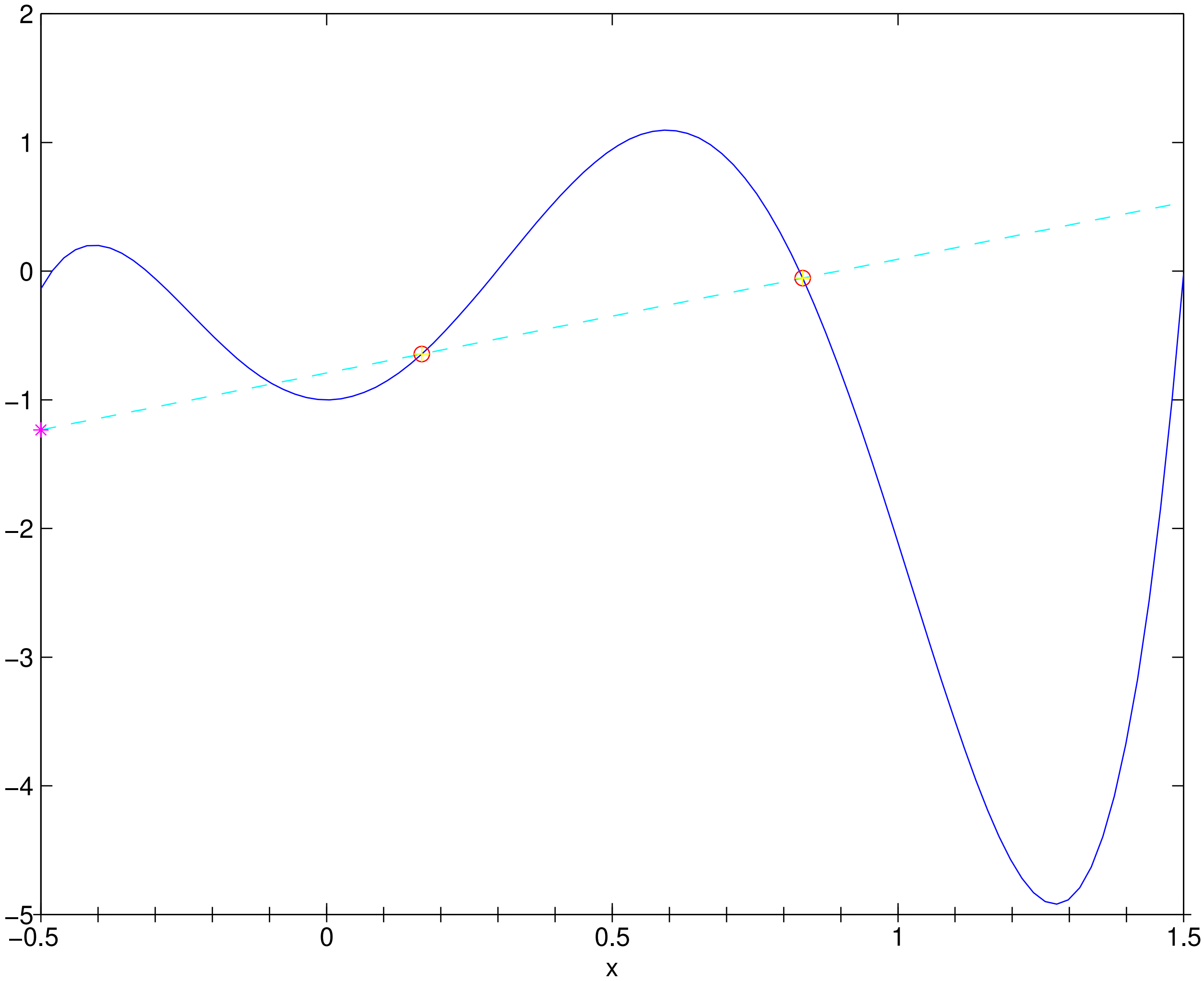
$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in [-0.5, 1.5], \end{array}$$

Again we start with *two* initial experimental design sites, at 0.167 and 0.833, and build a *linear* interpolant a_c.

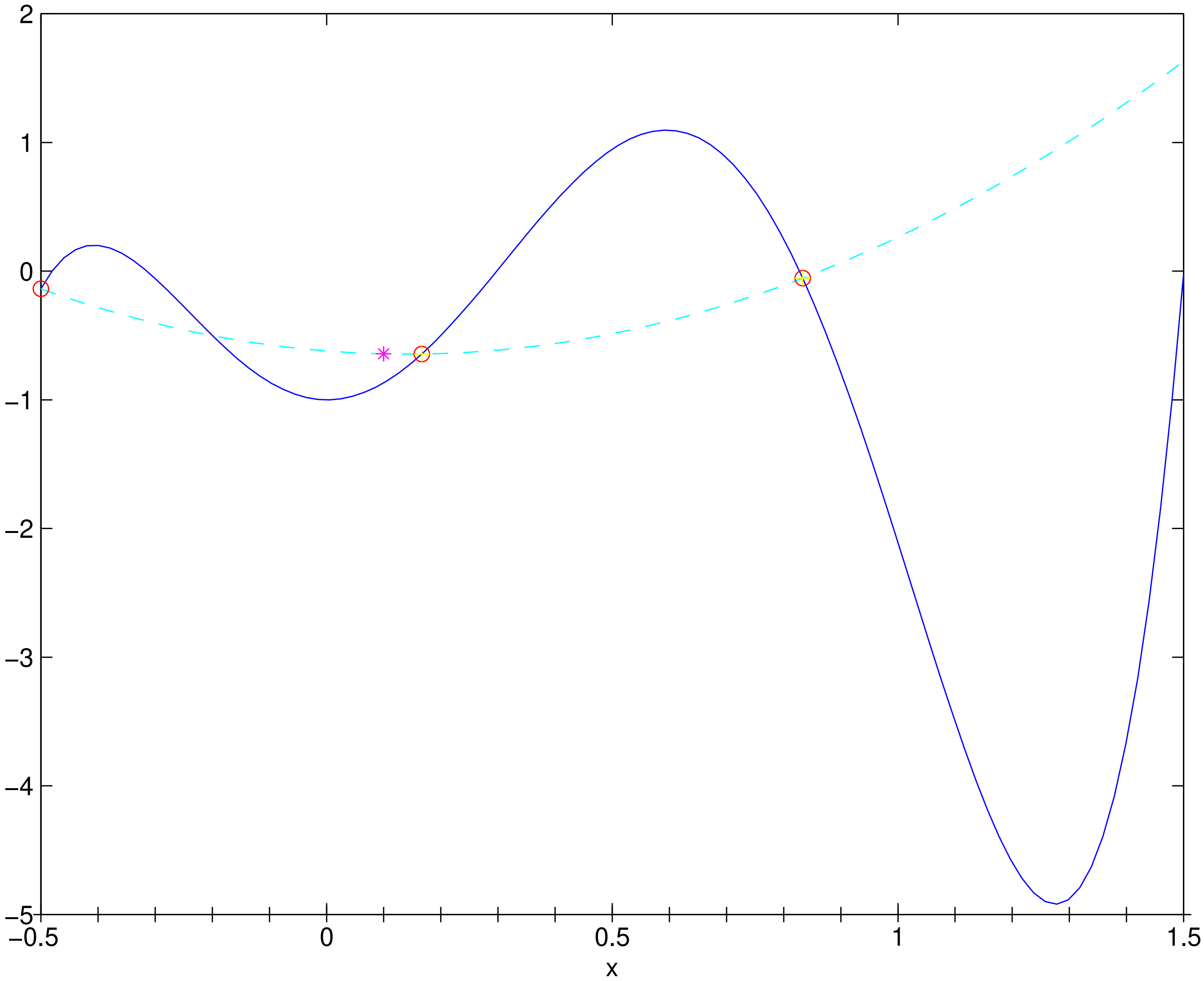
We choose whichever of the two initial experimental design sites has the lower value of the objective to be the baseline engineering design x_c.

We then proceed as before.

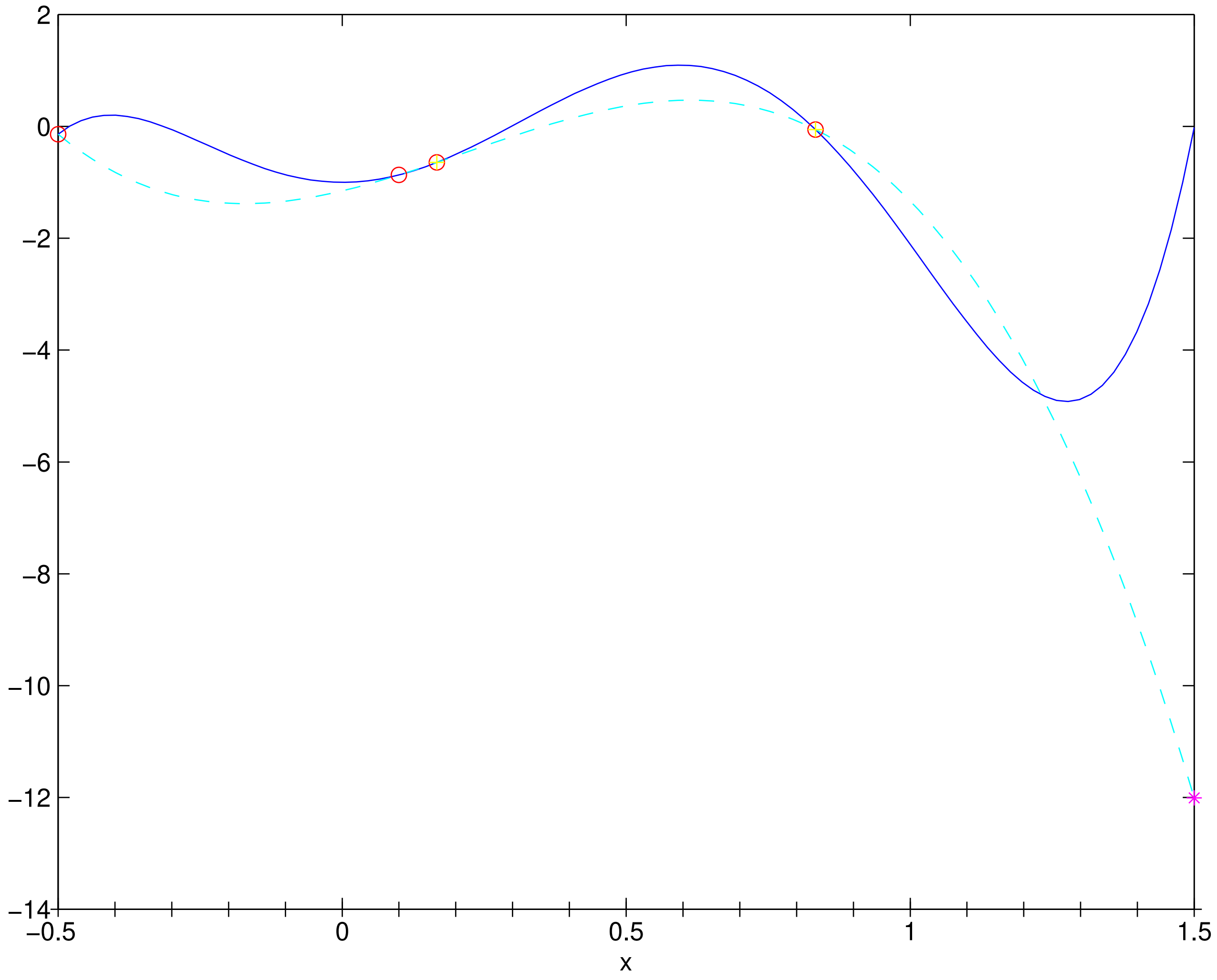
Iteration 1:



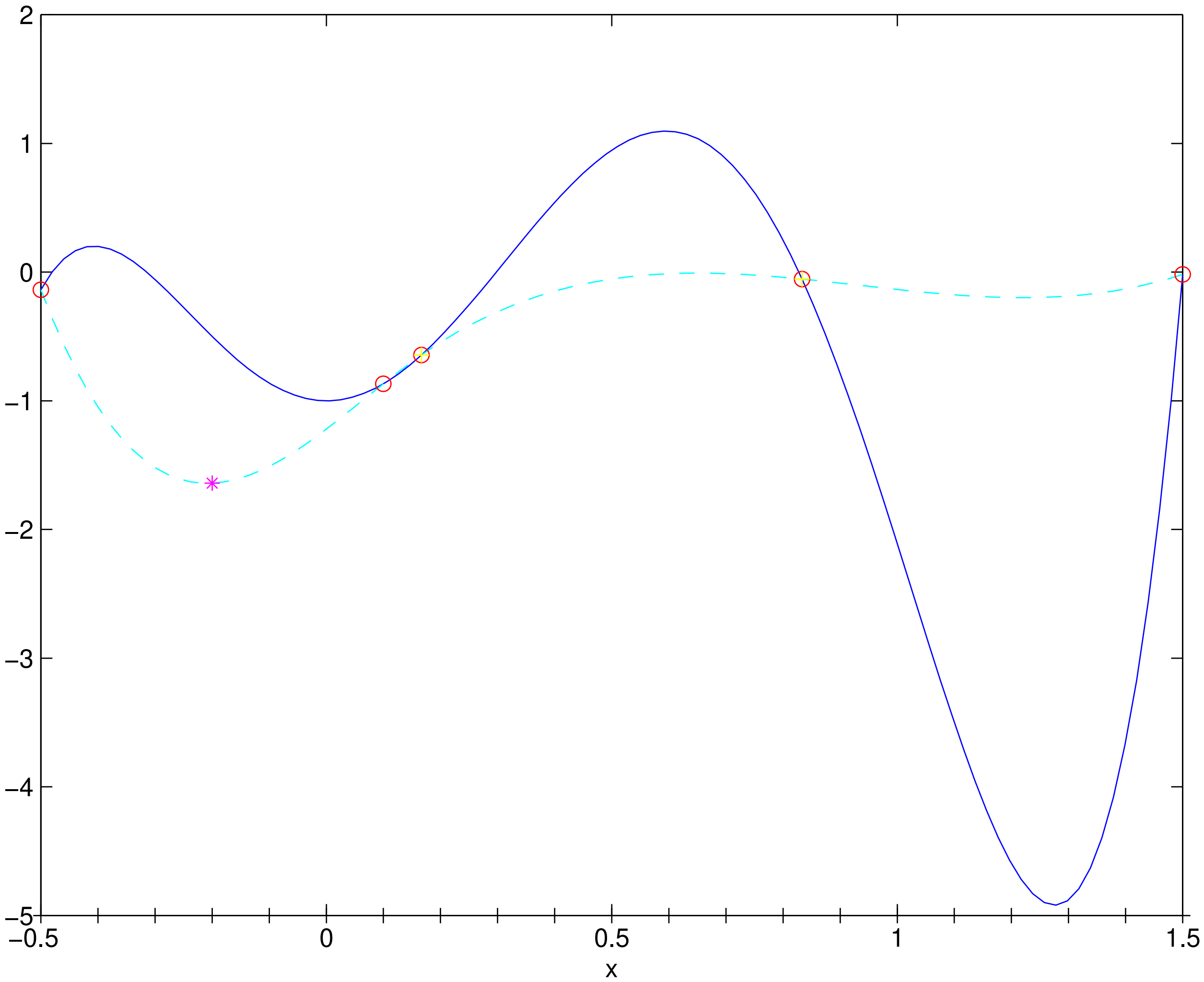
Iteration 2:



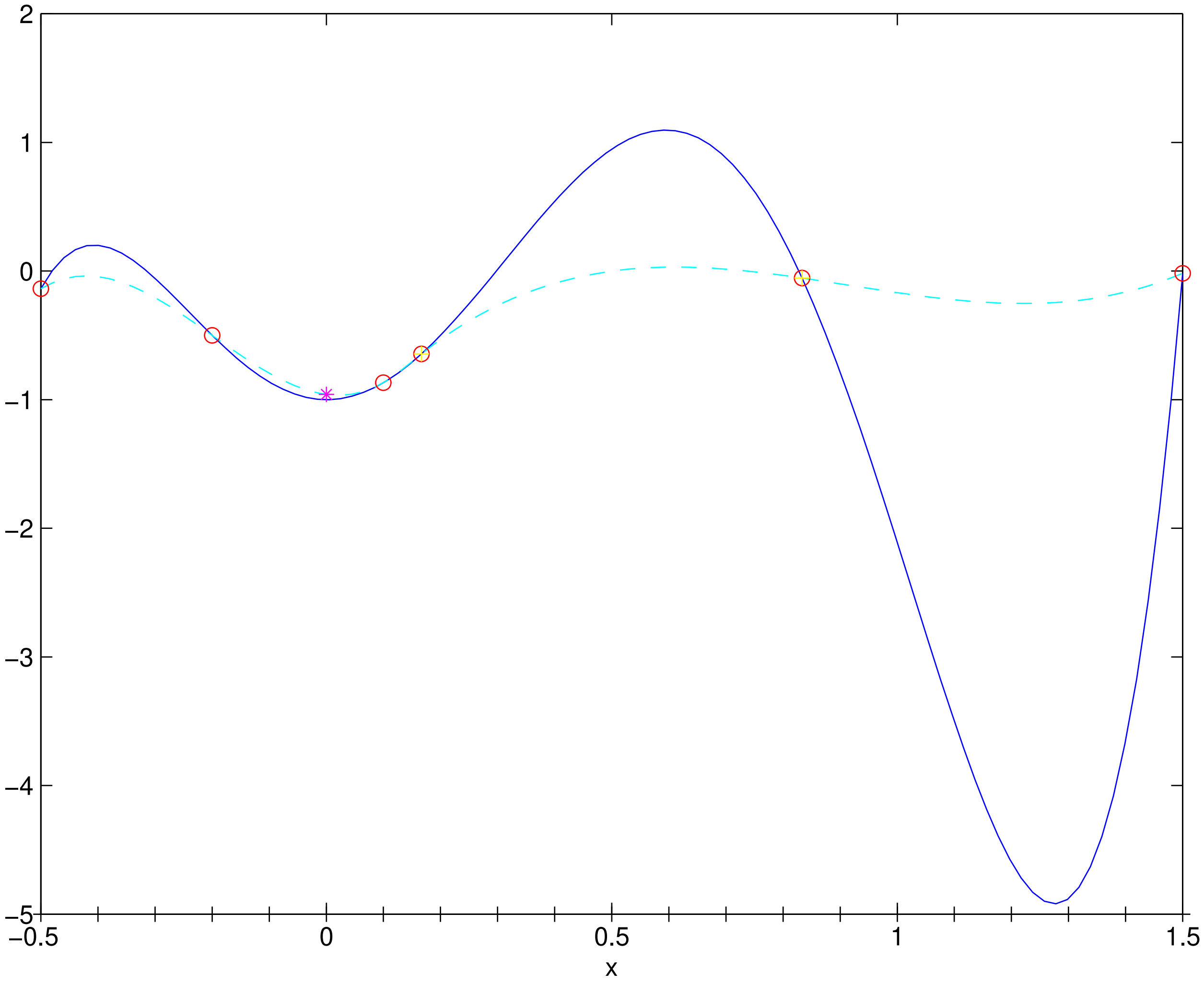
Iteration 3:



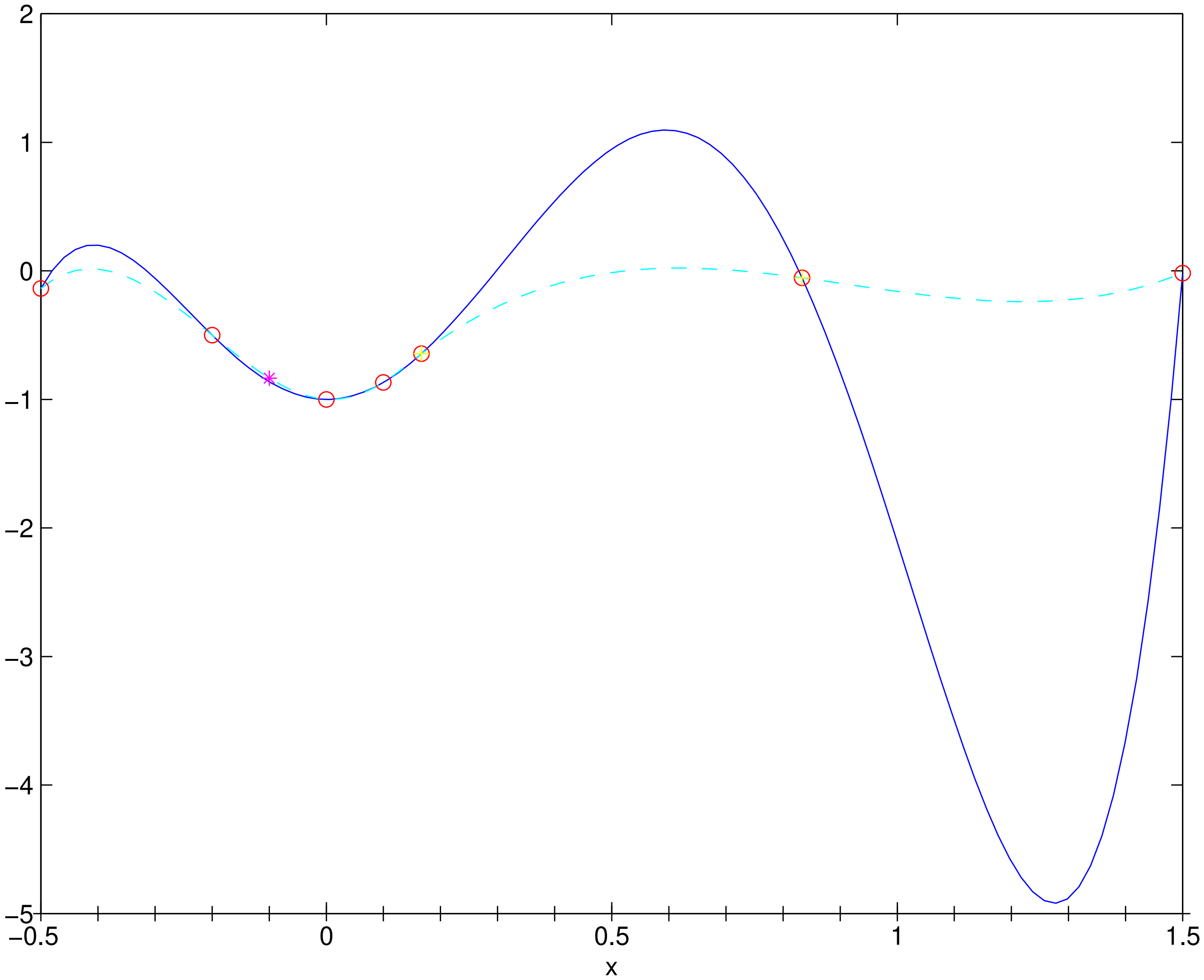
Iteration 4:



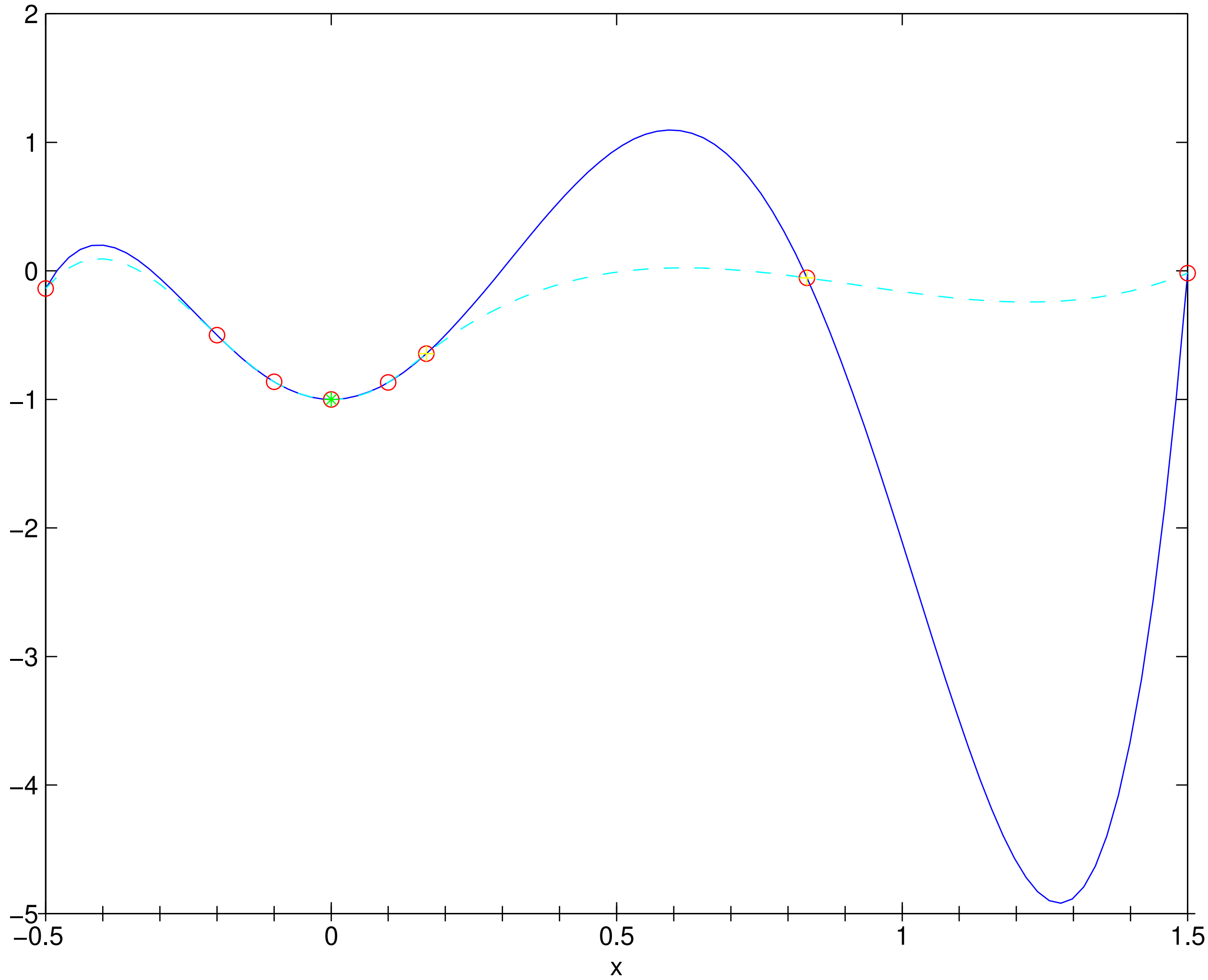
Iteration 5:



Iteration 6:



Iteration 7:



Summary for Second Example:

Consider the sequence of current solutions x_c versus trial solutions x_t that we generated at each iteration:

Iteration	x_c	x_t
1	0.167	-0.500
2	0.167	0.100
3	0.100	1.500
4	0.100	-0.200
5	0.100	0.000
6	0.000	-0.100
7	0.000	—

The optimization correctly identifies a *local* minimizer of the object f—which is all the analysis guarantees.

An Alternate Strategy:

Search in a region where the current approximation a_c indicates there might be a minimizer

and

where we realize we “know” very little about the objective.

Combine these two goals into a single objective, which we will refer to as a *merit function*.

One Possibility:

Use a merit function of the form

$$m_c(x) = a_c(x) - \rho_c d_c(x),$$

where $\rho_c \geq 0$ and

$$d_c(x) = \min \|x - x_i\|_2,$$

where the minimum in d_c is computed over all points x_i at which we know the value of the true objective. Thus, $d_c(x)$ is the distance from x to the nearest previous experimental design site.

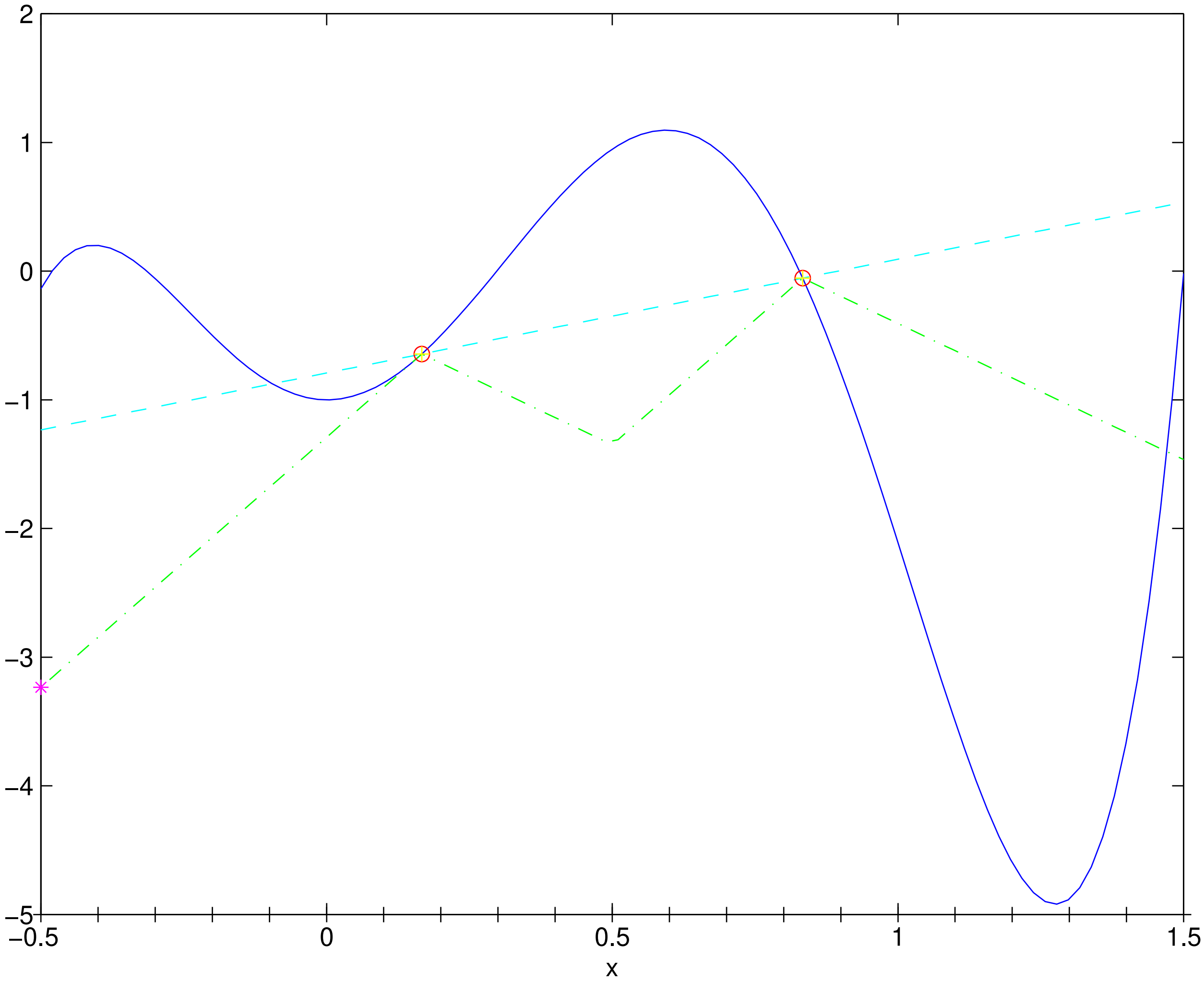
Other choices are possible.

Second Example Revisited:

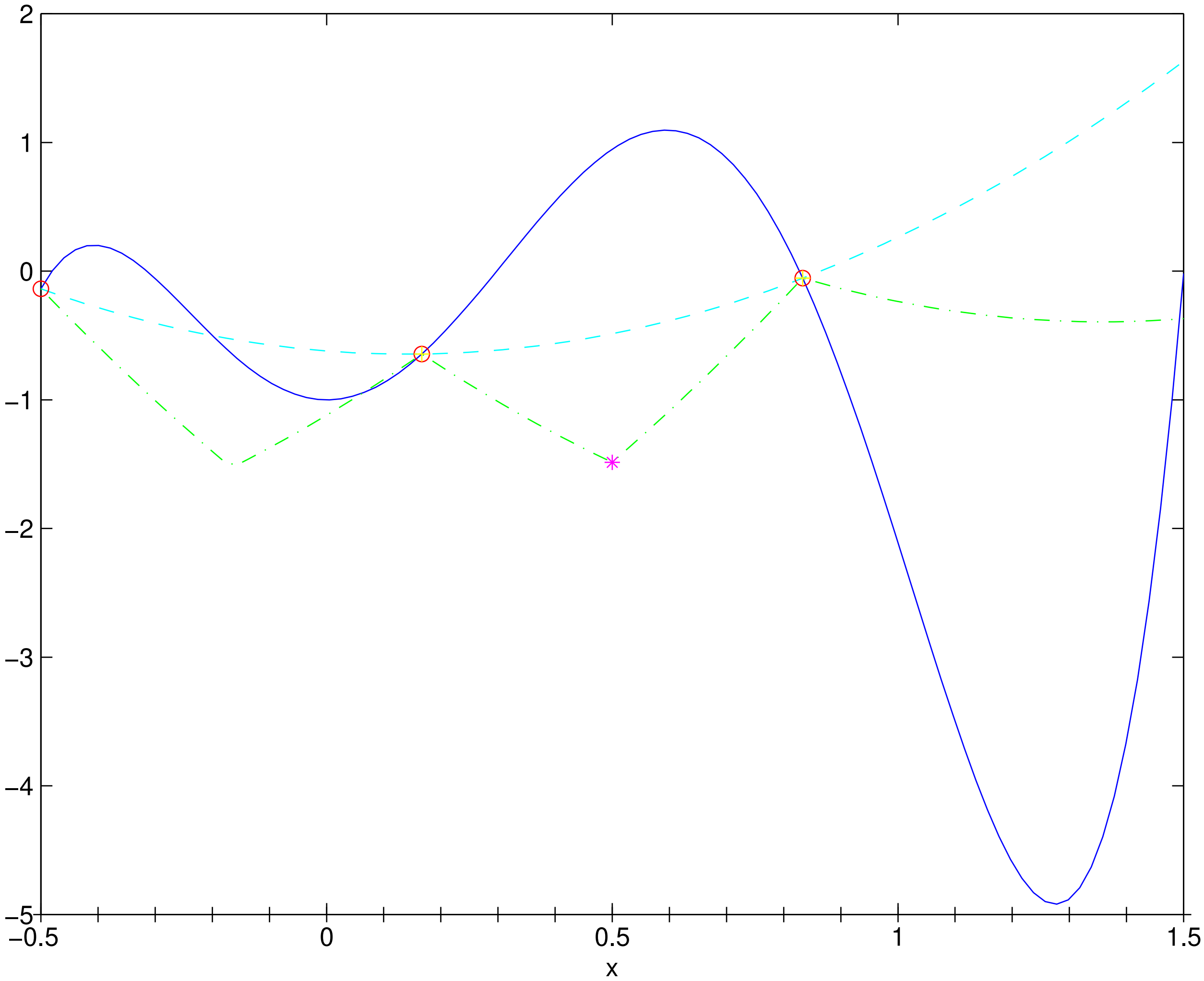
To see how well this works, we will repeat the second example, but now we will use m_c—rather than a_c—to choose our trial points x_t.

For this example, we use $\rho_c = 3$. (There is no real motivation for this choice of ρ_c; it was the first value tried and it happened to work particularly well for this example.)

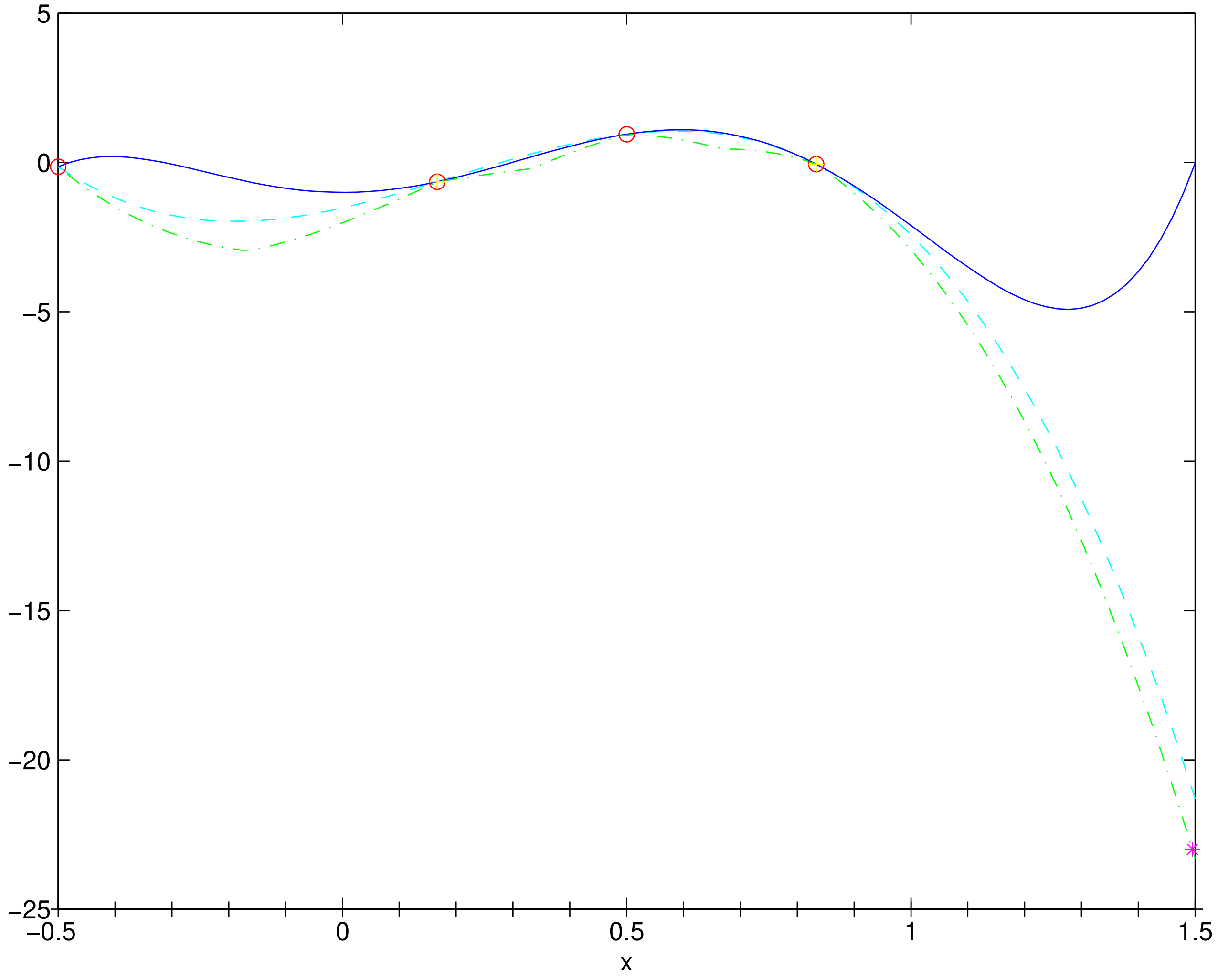
Iteration 1:



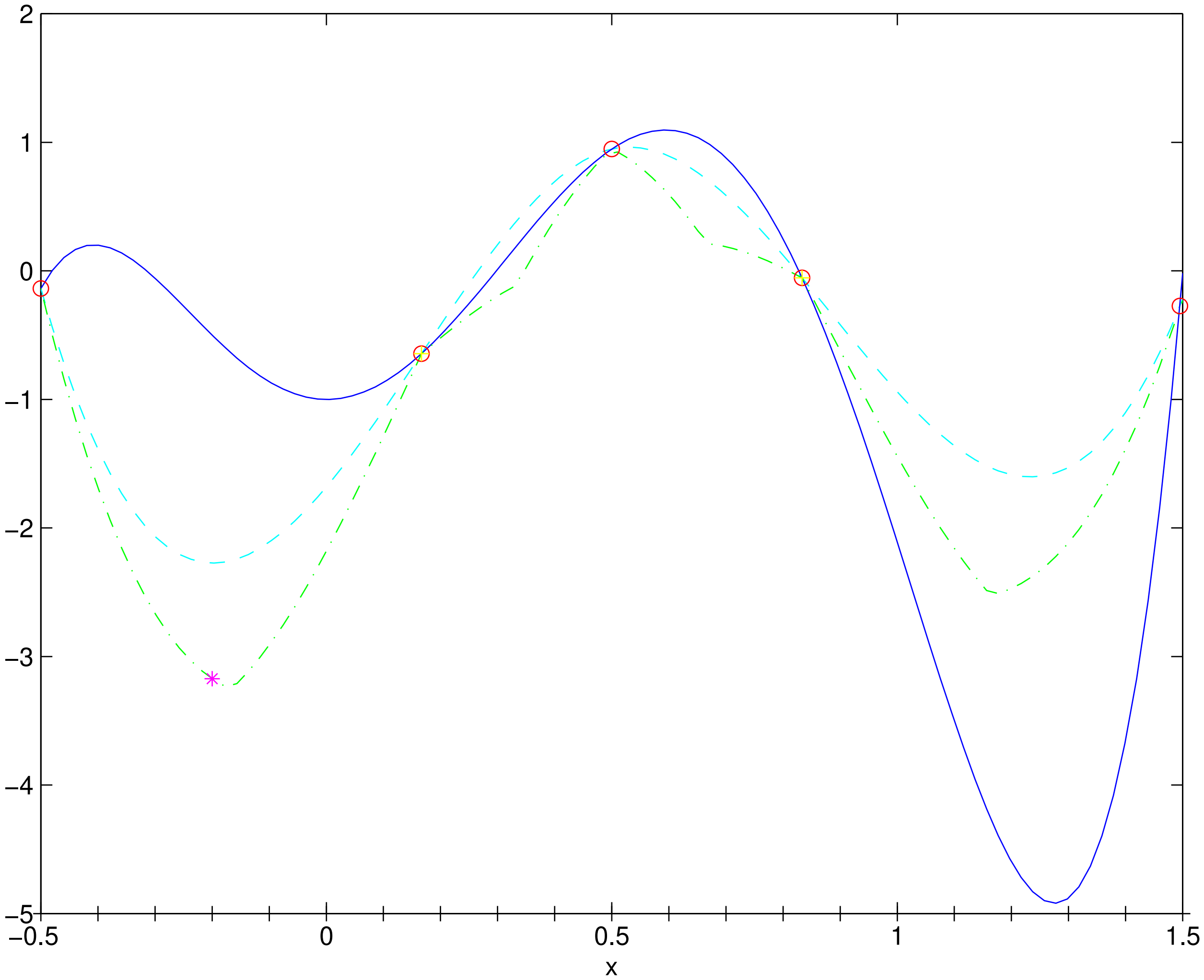
Iteration 2:



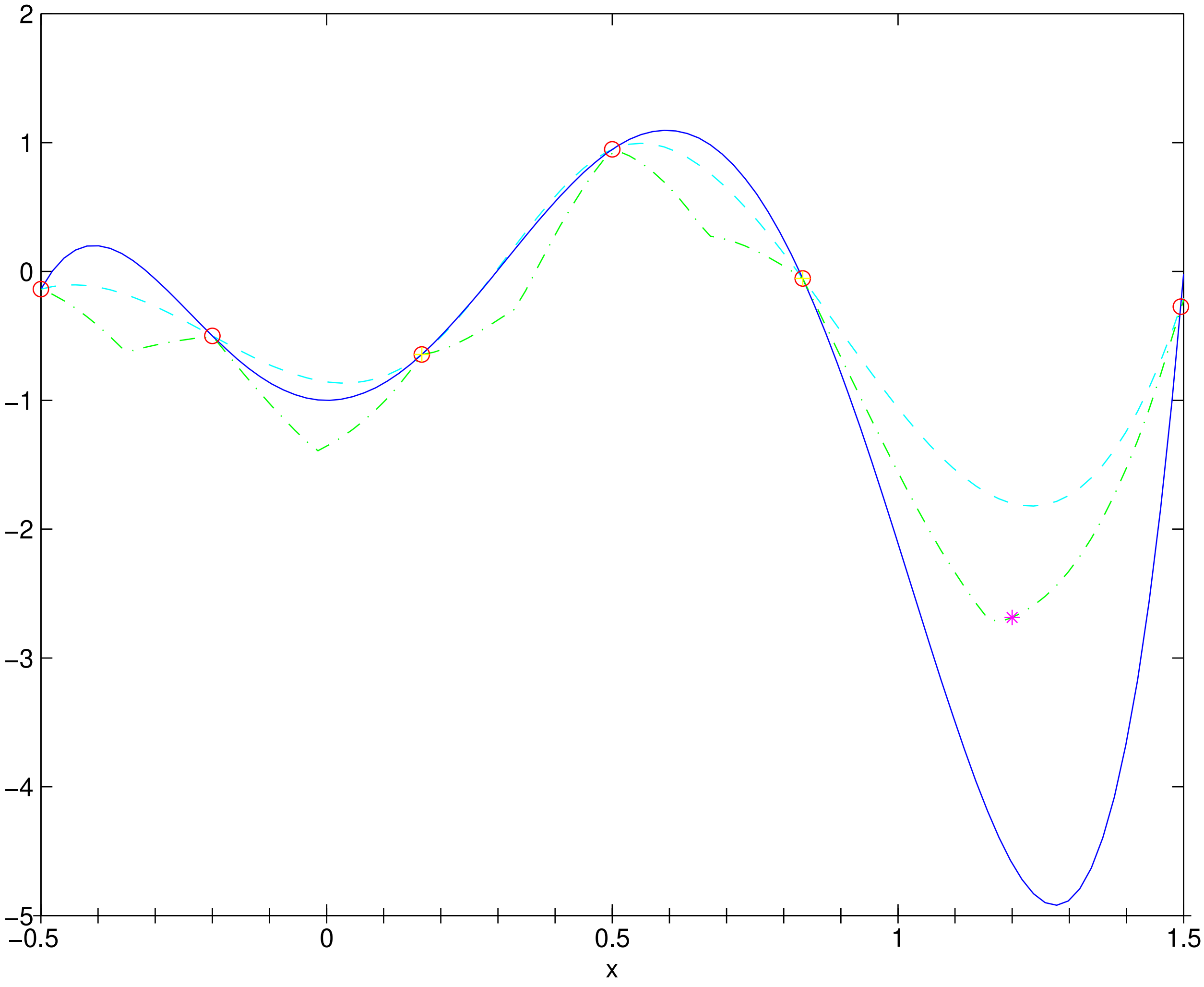
Iteration 3:



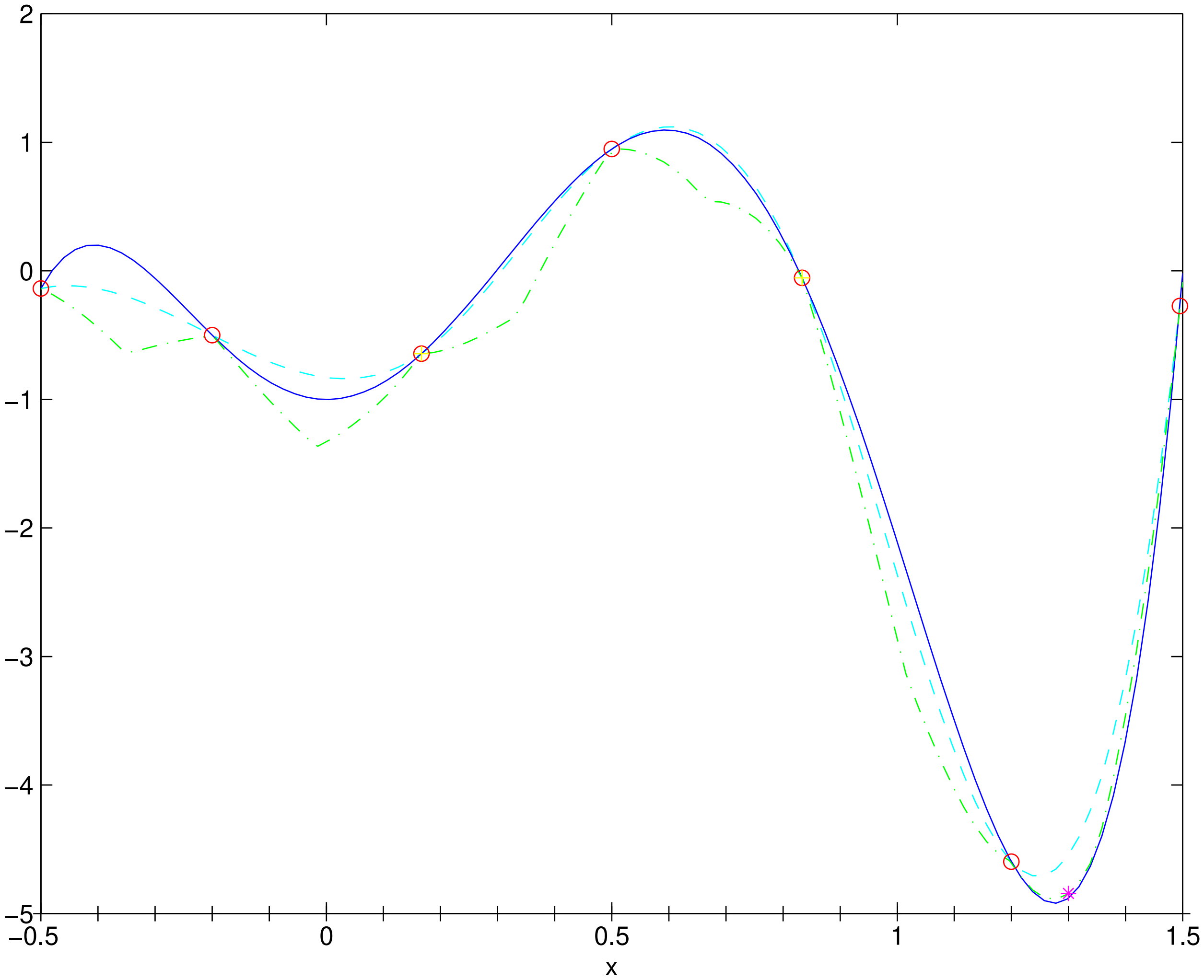
Iteration 4:



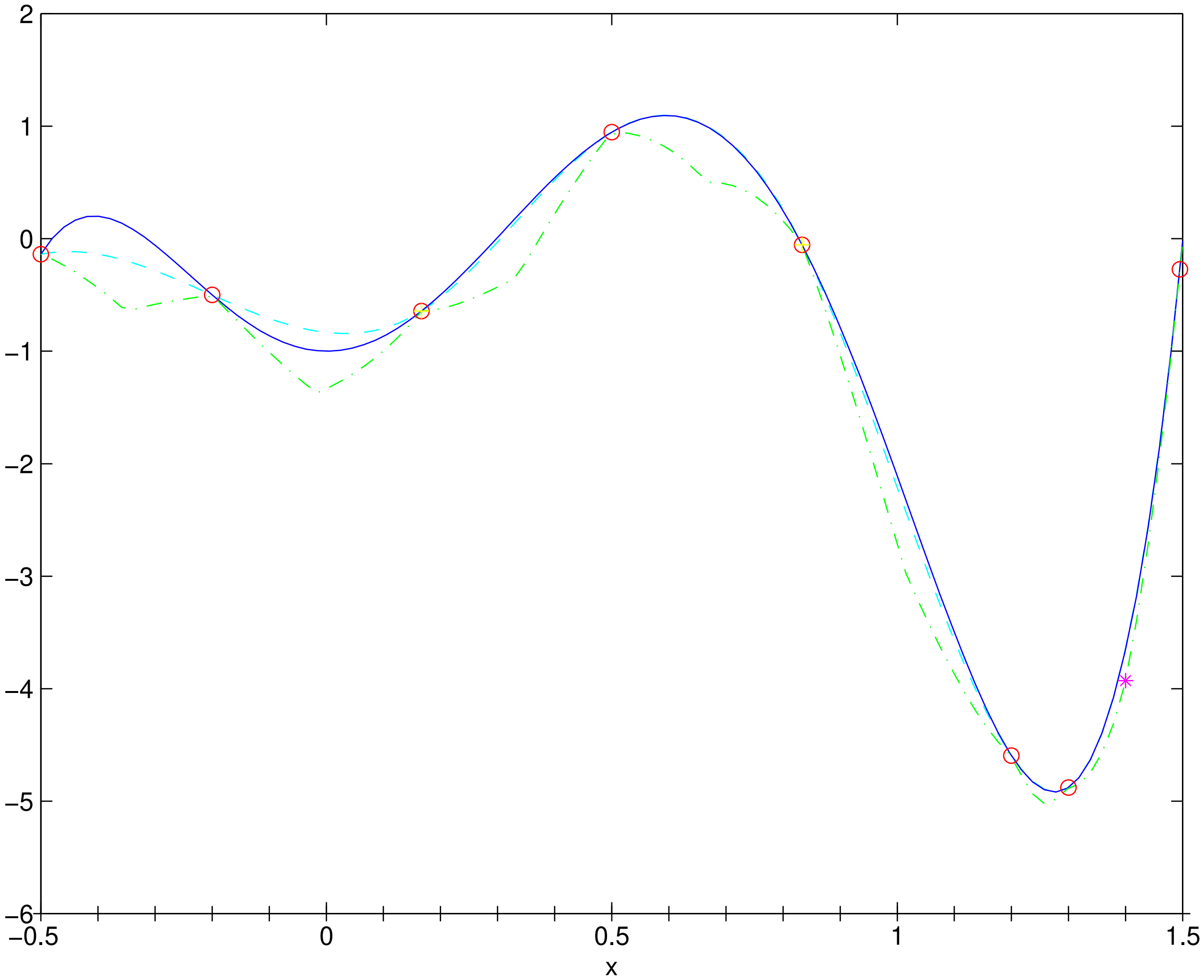
Iteration 5:



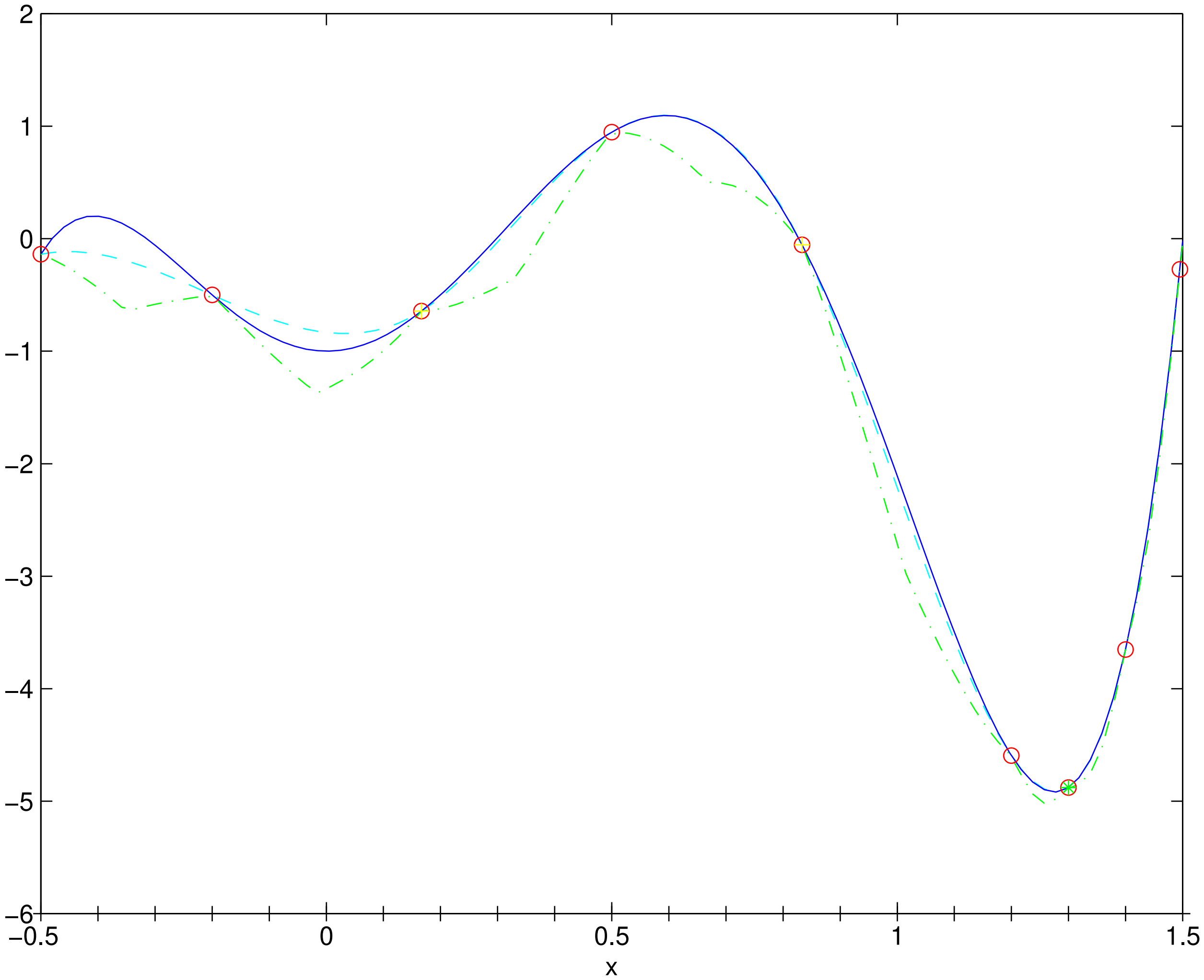
Iteration 6:



Iteration 7:



Iteration 8:



Summary for Second Example Revisited:

Consider the sequence of current solutions x_c versus trial solutions x_t that we generated at each iteration:

Iteration	x_c	x_t
1	0.167	-0.500
2	0.167	0.500
3	0.167	1.500
4	0.167	-0.200
5	0.167	1.200
6	1.200	1.300
7	1.300	1.400
8	1.300	—

Conclusions:

1. Global approximations can be helpful in facilitating optimization.

2. There is merit in balancing the goal of finding minimizers of the current approximation a_c against the goal of constructing better approximations for future iterations.

Outstanding Issues:

- How best to balance the goal of finding minimizers of the current approximation versus the goal of constructing better approximations?

  For now, we introduce the parameter ρ_c—and leave open the question of how best to choose ρ_c.

- What is the best way to "optimize" either the approximation a or the merit function m to predict a candidate minimizer for the objective f?

  The merit function m necessarily introduces multiple local minimizers.

- How well will this approach work on problems with a reasonable number of engineering design variables?

For a corrected PostScript version of this paper, and PostScript copies of other related papers, visit:

`http://www.cs.wm.edu/~va/research`

Many of these papers are also available as ICASE Technical Reports, which are distributed in both PostScript and Portable Document Format (pdf) form. Visit

`http://www.icase.edu`

and click on “publications”.