CISC 371 Class 2

Minimizing by Approximation

Texts: [1] pp. 95–98

Main Concepts:

- *Approximation by a quadratic, no derivatives*
- *Approximation by a quadratic with first derivative*
- *Approximation by a quadratic with first and second derivatives*
- *Extra notes: scalar differential calculus*

Sample Problem, Signal Processing: For the step response of a simple dynamical system that oscillates, where is the first minimum of the oscillation?

For an unknown or difficult scalar objective function, with one or more estimates of the minimizer, an *approximation* method uses a local *model* of the objective to improve the estimate of the minimizer. The idea is that the model is simple enough that its local minimum can be found easily, and the minimizer for the model can be used as a new estimate of the minimizer of the objective. An iterative implementation can have an inner loop that performs three three tasks:

- Use current estimates to create a model function $p(t)$
- Analytically use $p'(t)$ to find a local minimizer $\hat{t}$ of the model $p(t)$
- Update the current estimate with $\hat{t}$

An example of a difficult objective can come from signal processing. An example of an optimization problem is a dynamical system. A simple case is a damped-spring system. The spring causes a displaced mass to oscillate, or bounce, and friction or another source of damping causes the mass to exponentially come to rest. In combination, the mass will oscillate with a steadily decreasing amplitude. The equation of motion for this dynamical system, as a function of $t \geq 0$, is

$$f(t) = e^{-bt} \cos(2\pi\omega t) \tag{2.1}$$

For an oscillation frequency of $\omega = 2$ and a damping coefficient of $b = 1$, the motion described by Equation 2.1 is illustrated in Figure 2.1.

This function has one global minimum and multiple local minima, so we need more powerful methods for this complicated problem.

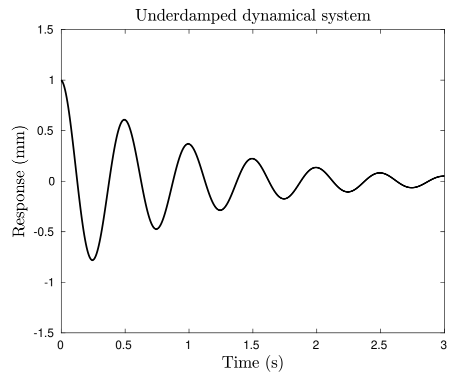

Figure 2.1: A dynamical system oscillates with a constant frequency and decreasing amplitude.

2.1 Quadratic Interpolation: 3 Points

Most approximation methods use a relatively low-order polynomial $p(t)$ as a local model for the objective function. The lowest order of a polynomial that has a minimum is second order, which is a quadratic polynomial. In `MATLAB` notation, the coefficients for a second-order polynomial model are

$$p(t) \stackrel{\text{def}}{=} a_1 t^2 + a_2 t + a_3 \tag{2.2}$$

These 3 polynomial coefficients can be determined with 3 independent pieces of information. The simplest source of information is 3 points for which the objective evaluations are non-collinear. If the objective $f(t)$ has a single local minimum – which is the usual assumption in scalar optimization – then the non-collinearity condition is always satisfied.

We can write the 3 points near the minimum of $f(t)$ as t_1, t_2, and t_3. Using the abbreviation $f_i \stackrel{\text{def}}{=} f(t_i)$, we can constrain the polynomial model $p(t)$ so that is evaluates to f_i at t_i. That is, we use the constraints $p(t_i) = f_i$. In vector notation, the three constraints are

$$\begin{bmatrix} p(t_1) \\ p(t_2) \\ p(t_3) \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{2.3}$$

Expanding the left side of Equation 2.3 produces

$$\begin{bmatrix} a_1 t_1^2 + a_2 t_1 + a_3 \\ a_1 t_2^2 + a_2 t_2 + a_3 \\ a_1 t_3^2 + a_2 t_3 + a_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{2.4}$$

The constraints of Equation 2.4 provide 3 equations in the 3 unknown coefficients a_j, which we can write as the linear equation

$$\begin{bmatrix} t_1^2 & t_1 & 1 \\ t_2^2 & t_2 & 1 \\ t_3^2 & t_3 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{2.5}$$

The matrix in Equation 2.5 is the Vandermonde matrix and the equation can be solved numerically or algebraically to find the coefficients of the second-order polynomial $p(t)$.

The minimizer of the quadratic model can be found using elementary differential calculus. The minimizer must occur at a stationary point, which is a point $\hat{t}$ for which $p'(\hat{t}) = 0$. Finding the first derivative of Equation 2.2, and constraining the derivative to be zero, gives the equation

$$p'(\hat{t}) = 2a_1\hat{t} + a_2 = 0 \tag{2.6}$$

Equation 2.6 has the solution

$$\hat{t} = \frac{-a_2}{2a_1} \tag{2.7}$$

The use of a 3-point quadratic model for optimization, using Equation 2.7 to estimate a new point for an iterative minimization process, is illustrated in Figure 2.2.

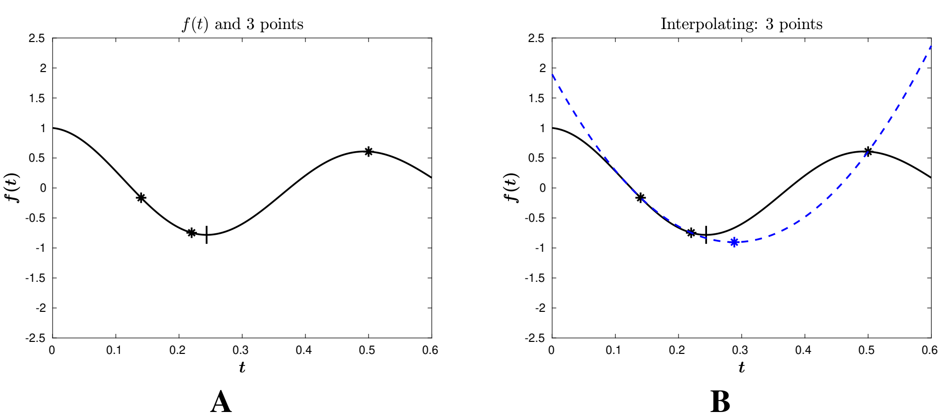

Figure 2.2: A quadratic model of a scalar objective function using 3 points. (A) 3 points near the local minimum of a smooth objective function; the minimum is indicated by the vertical bar. (B) In blue, the quadratic model that interpolates the 3 points and the analytic minimizer of the model.

2.2 Quadratic Interpolation: 2 Points, First Derivative

In some applications of scalar optimization, the first derivative of one point is known. This piece of information can be used to constrain the local polynomial model, which implies that the information from just 2 points is needed to determine the coefficients of a quadratic polynomial model.

Suppose that 2 points near a local minimum of the objective $f(t)$ are given as t_1 and t_2, with corresponding objective evaluations f_1 and f_2. Suppose that the evaluation of the first derivative at t_1 is also provided as $f'_1 \stackrel{\text{def}}{=} f'(t_1)$. How can we use this new information to determine the coefficients of the model quadratic polynomial?

Following the methods that we used for constraining a model with 3 points, we can re-write Equation 2.3 as

$$\begin{bmatrix} p(t_1) \\ p(t_2) \\ p'(t_1) \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f'_1 \end{bmatrix} \tag{2.8}$$

We know the first derivative of the polynomial model in Equation 2.2, from elementary differential calculus, as

$$p'(t) = 2a_1 t + a_2$$

Expanding the polynomial model, and collecting the polynomial coefficients into a vector of unknowns, gives the linear equation

$$\begin{bmatrix} t_1^2 & t_1 & 1 \\ t_2^2 & t_2 & 1 \\ 2t_1 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f'_1 \end{bmatrix} \tag{2.9}$$

Equation 2.9 can be solved numerically or in a closed form to find the coefficients of the quadratic polynomial model. Its use is illustrated in Figure 2.3.

2.3 Quadratic Interpolation: 1 Point, First and Second Derivative

It may be that both the first derivative and the second derivative of the objective function are known at one point. This information can also be used to constrain the local polynomial model. For a quadratic polynomial model, knowledge of the values of the first two derivatives at a point are sufficient to constrain the coefficients of the model.

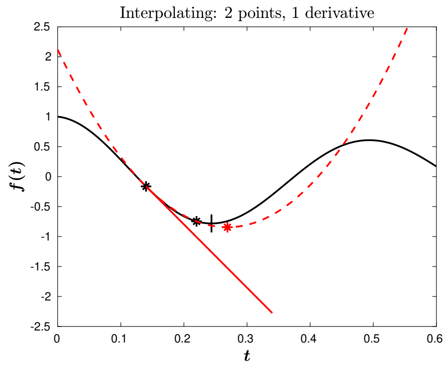

Figure 2.3: A quadratic model of a scalar objective function using 2 points and a first derivative. The 2 points near the local minimum of a smooth objective function are shown as black asterisks, as is the minimum is indicated by the vertical hash bar. The first derivative of the objective at the first given point is shown as a solid red line. The quadratic model that interpolates the 2 points is shown as a dashed red curve. The analytic minimizer of the model is shown as a red asterisk.

Suppose that 1 point near a local minimum of the objective $f(t)$ is given as t_1, with corresponding objective evaluation f_1. Suppose that the evaluation of the first derivative at t_1 is provided as $f'_1 \stackrel{\text{def}}{=} f'(t_1)$, and that the evaluation of the second derivative at t_1 is provided as $f''_1 \stackrel{\text{def}}{=} f''(t_1)$.

We know the second derivative of the polynomial model in Equation 2.2, from elementary differential calculus, as

$$p''(t) = 2a_1$$

Expanding the polynomial model, and collecting the polynomial coefficients into a vector of unknowns, gives the linear equation

$$\begin{bmatrix} t_1^2 & t_1 & 1 \\ 2t_1 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f'_1 \\ f''_1 \end{bmatrix} \tag{2.10}$$

Equation 2.10 can be solved numerically or in a closed form to find the coefficients of the quadratic polynomial model. It use is illustrated in Figure 2.4.

2.4 Higher-Order Approximation Methods

A higher-order polynomial can be used to locally model an objective function of a scalar argument. In practice, optimization algorithms use either a second-order quadratic model or a third-order cubic model. A cubic model requires 4 constraints, which in practice are usually one of:

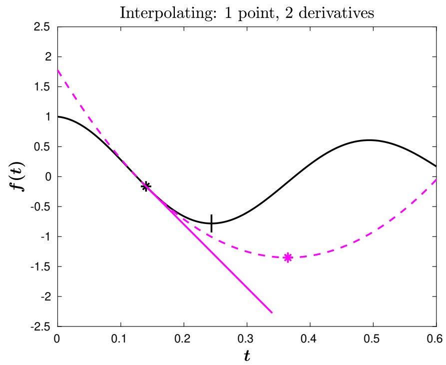

Figure 2.4: A quadratic model of a scalar objective function using 1 point, plus the first and second derivatives. The 1 point near the local minimum of a smooth objective function are shown as a black asterisk; the minimum is indicated by the vertical hash bar. The first derivative of the objective at the first given point is shown as a solid magenta line. The quadratic model that interpolates the 1 point is shown as a dashed magenta curve. The analytic minimizer of the model is shown as a magenta asterisk.

- 4 points t_1, t_2, t_3, and t_4
- 3 points t_1, t_2, and t_3; plus 1 first derivative f_1'
- 2 points t_1 and t_2; plus 2 first derivatives f_1' and f_2'

The methods described above can be used to develop a 4×4 matrix and associated linear equation that can be solved numerically or in closed form.

2.5 Computer Implementation

For each approximation method, there are subtleties associated with its computer implementation in an iterative algorithm. Good computer code will check, when possible, that the proposed minimizer $\hat{t}$ has the required property of having an objective evaluation that is strictly less than the evaluations for the given points. The new point $\hat{t}$ should replace a previous point in the most useful way possible, by providing a superior new estimate of the true minimizer t^*. If derivatives are used, the first derivative must be checked to ensure that it “descends” towards a local minimum and the second derivative must be upward-convex. Numerical concerns, such as a derivative being close to zero that results in a poorly conditioned linear equation, should also be tested.

References

[1] Antoniou A, Lu WS: Practical Optimization: Algorithms and Engineering Applications. Springer Science & Business Media, 2007