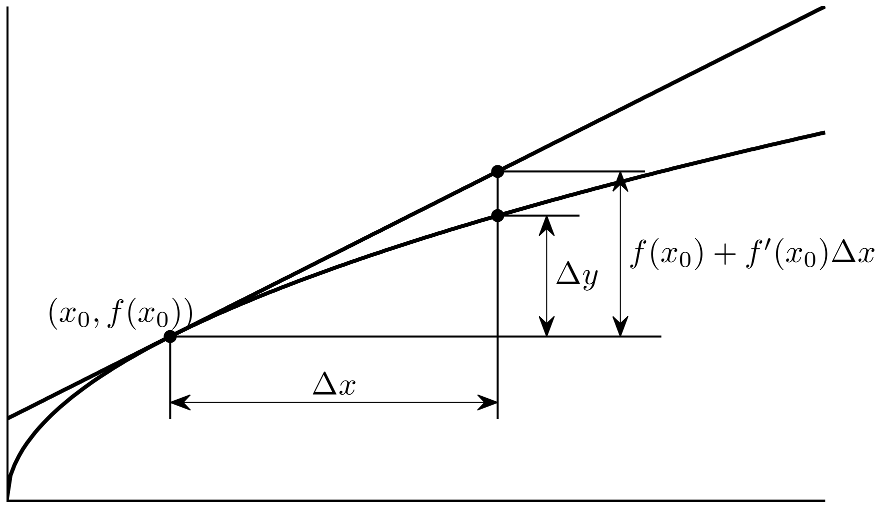


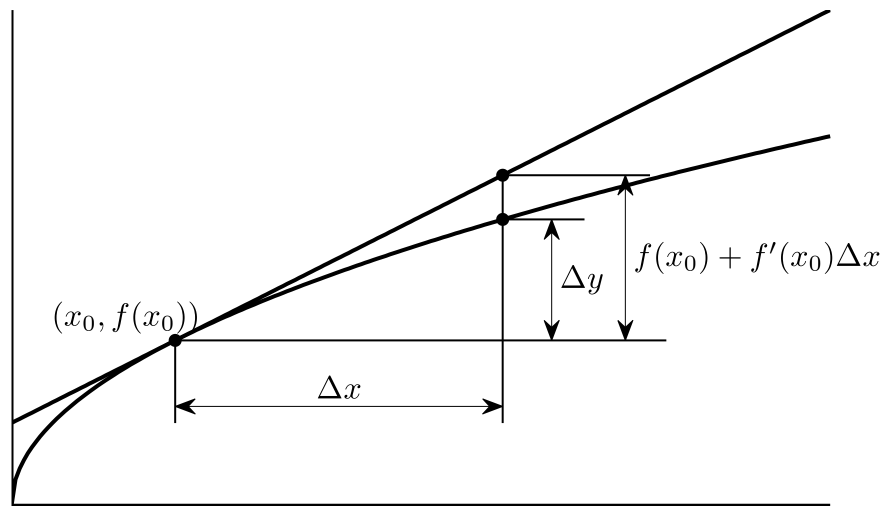
Figure 1: Let f be a differentiable function. Suppose that we know its value and also the value of its derivative at a certain point x_0. How can we estimate the value of f at a nearby point $x = x_0 + \Delta x$ using these two numbers? The figure shows that a reasonable approximation is given by the simple formula: $f(x) \approx f(x_0) + f'(x_0)\,\Delta x$. This is, of course, the familiar linear (or tangent line) approximation from Calculus I. Now suppose that we are given additional information about f at x_0, say, the value of its second derivative $f''(x_0)$. How can we improve upon the linear approximation? Think about it and try to come up with some ideas before reading on.

Linear approximation revisited

Consider the following standard exercise from Calculus I:

> Use linear approximation to estimate $\sqrt{10}$.

I will solve this problem in the next paragraph. However, before reading my solution, solve the problem by yourself and see how long it takes. Figure 1, reproduced below, is a clue.



Now the solution goes as follows. It is clear that $\sqrt{10}$ should be close to $\sqrt{9} = 3$. We therefore linearize $f(x) = \sqrt{x}$ at $x_0 = 9$ by computing the equation of the tangent line as follows:

$$y = f(x_0) + f'(x_0)\,(x - x_0) = 3 + \frac{1}{6}\,(x - 9).$$

We now use this linearization to approximate

$$f(10) \approx y(10) = 3 + \frac{1}{6}\,(10 - 9) = 3.1666\ldots$$

Spend a few minutes consciously connecting the algebra in this example to the geometry in Figure 1—this is very important! Do you see how the change in the function $\Delta y = f(x) - f(x_0)$ is dominated by the *differential* $f'(x_0)\,\Delta x = f'(x_0)\,(x - x_0)$?

All of this is well and good and, hopefully, very familiar from Calculus I. What may not be familiar is the error of linear approximation. What does this error look like? More importantly, how can we reduce it?

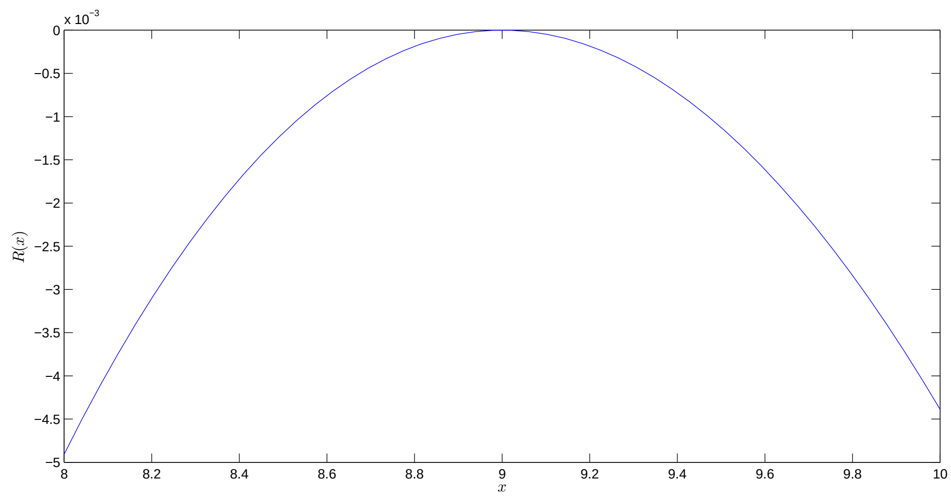


Figure 2: Plot of the remainder of linear approximation of $\sqrt{x}$ in the neighborhood of $x_0 = 9$

Let us define

$$R_1 = f(x) - f(x_0) - f'(x_0)\,(x - x_0) \tag{1}$$

as the error of linear approximation at x_0; the letter R stands for *remainder* and the subindex indicates the degree of the approximating polynomial. With x_0 and f fixed, R_1 is a function of x alone showing how much f deviates from its tangent line based at x_0 as x moves away from x_0. For our specific example

$$R_1 = \sqrt{x} - 3 - \frac{1}{6}(x-9).$$

Figure 2 shows the plot of R_1 in a small neighborhood of $x_0 = 9$; the figure was produced in `Matlab` using the following code:

```
f = @(x) sqrt(x);           % define f(x)
g = @(x) 3 + (x-9)/6;       % define linear approximation at x0=9
R = @(x) f(x)-g(x);         % remainder of linear approximation

x = linspace(8,10,50);      % generate 50 abscissas in [8,10]
y = R(x);                   % corresponding ordinates

figure                      % open a new window for the figure
plot(x,y,'b-')              % plot y against x as a solid blue line
xlabel('x')                 % label x-axis
ylabel('R')                 % label y-axis
```

Evidently the plot in Figure 2 looks a lot like a simple parabola. To confirm that, we can perform quadratic fit with the following code:

```
p = polyfit(x,y,2);         % fit quadratic model p = a*x^2+b*x+c

figure                      % open a new window for plotting
plot(x,y,'bo')              % plot data as circles
hold on                     % ensure that plots are added rather than replaced
plot(x,polyval(p,x),'r-')   % plot quadratic fit as a red line
legend('data','fit')        % add legend
disp(p)                     % display [ a b c ]

>> -0.0046    0.0838    -0.3777
```

Figure 3 shows convincingly that in the interval $[8, 10]$ the remainder R_1 is indeed very close to the quadratic:

$$R_1 \approx -0.0046\,x^2 + 0.0838\,x - 0.3777$$

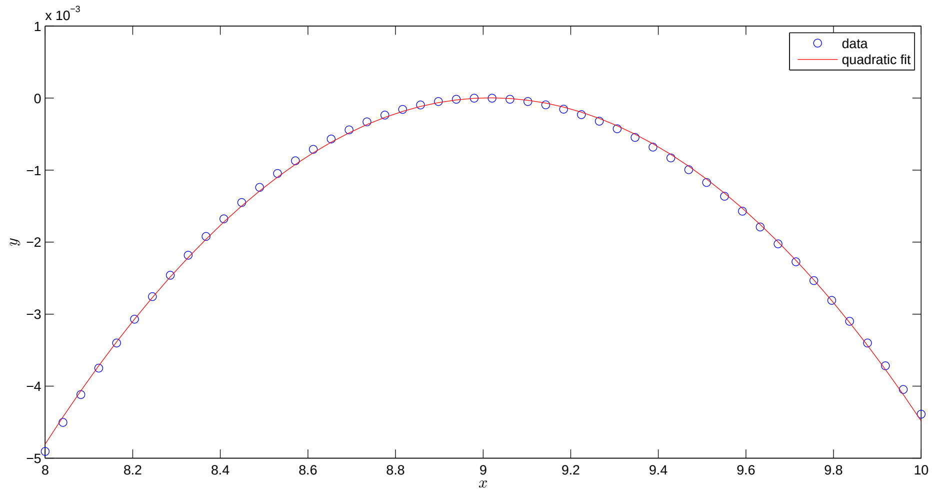


Figure 3: Plot of the remainder of linear approximation of $\sqrt{x}$ in the neighborhood of $x_0 = 9$ and the best quadratic fit

This suggests that in order to decrease the error in linear approximation, we should add a quadratic correction term. In other words, the next logical step after linear approximation is *quadratic* approximation.

Quadratic approximation

We could, with some effort, deduce the likely form of the quadratic approximation by continuing to work with the data shown in Figure 3. However there is a more expedient way leading directly to analytic formulas. The method is based on the simple observation that if a function is a quadratic to begin with, then its quadratic approximation at any point should be exact.

Consider $f = a\,x^2 + b\,x + c$ near x_0. According to Equation (1), the remainder of linear approximation is

$$\begin{aligned} R_1 &= a\,x^2 + b\,x + c - (a\,x_0^2 + b\,x_0 + c) - (2\,a\,x_0 + b)\,(x - x_0) \\ &= a\,(x - x_0)^2. \end{aligned}$$

By the way, do not take my word for the above simplification—derive it yourself! Since the second derivative of a quadratic is a constant function, we can write $a = f''(x_0)/2$. This leads to the identity

$$f = f(x_0) + f'(x_0)\,(x - x_0) + \frac{f''(x_0)}{2}\,(x - x_0)^2 \tag{2}$$

which, of course, is valid only for polynomials of degree 2 or less. If f is not a quadratic, the above formula does not hold exactly. Nevertheless, in light of Figure 3, we may expect it to hold approximately.

It is time to introduce more notation which will be used throughout the course. Let

$$T_1 = f(x_0) + f'(x_0)\,(x - x_0)$$

denote the linear approximation to f at x_0. The symbol T stands for "Taylor polynomial" while the subindex indicates the degree. The right-hand side is the familiar equation of the tangent line, so "tangent line" and "Taylor polynomial of degree 1" are synonymous. Equation (2) suggests that we define

$$T_2 = f(x_0) + f'(x_0)\,(x - x_0) + \frac{f''(x_0)}{2}\,(x - x_0)^2. \tag{3}$$

as the Taylor polynomial of degree 2, or Taylor quadratic. Notice that T_2 differs from T_1 by just a single term involving the second derivative of the function. Naturally, we expect that term to be very close to the remainder of linear interpolation R_1. Returning to the square root function $f(x) = \sqrt{x}$ at $x_0 = 9$ we use (3) to compute

$$T_2 = 3 + \frac{1}{6}\,(x - 9) - \frac{1}{216}\,(x - 9)^2$$

If we expand the last term $-\frac{1}{216}\,(x-9)^2$, we get

$$-.0046\,x^2 + .0833\,x - .3750$$

which is remarkably close to the quadratic we found experimentally by using `polyfit`. The match cannot be exact, of course, since $\sqrt{x}$ is clearly not a quadratic. Nevertheless, it is fair to say that the *dominant* term of R_1 is the $-\frac{1}{216}\,(x-9)^2$ and, indeed, its graph is almost indistinguishable from the parabolas in Figures 2 and 3.

Returning to the original question, let us evaluate T_2 at $x = 10$. This gives quadratic approximation to $\sqrt{10}$ in the form:

$$\sqrt{10} \approx 3 + \frac{1}{6} - \frac{1}{216} = 3.1620\ldots$$

Notice that this approximation is accurate to three digits—one digit better than the linear approximation 3.1667–as we should expect.

This is a good place to stop, not because we exhausted the subject of polynomial approximation but rather because it is time for you to perform some numerical experiments on your own. Reread this handout carefully and try to guess what the our next steps will be. Do you think that quadratic approximation can be further improved? Can the process of improvement continue indefinitely? If so, what might happen in the limit, as approximation is taken to extremes?

Exercises

1. Let $R_1(x) = f(x) - T_1(x)$ be the error of linear approximation. Compute R_1 for $f(x) = e^x$ at $x = 0$ and investigate its shape. That is, plot R_1 on $[-1, 1]$ and use `polyfit` to fit a quadratic as done in the handout. Check the accuracy with a plot similar to Figure 3. Compare the results of the fit with the analytic formula suggested in the handout. Finally, repeat the exercise on a smaller interval $[-.1, .1]$. Summarize your observations from both numerical experiments in a clear paragraph.

2. Let

$$R_2 = f(x) - T_2(x) = f(x) - f(x_0) - f'(x_0)\,(x - x_0) - \frac{f''(x_0)}{2}\,(x - x_0)^2$$

be the remainder of quadratic approximation. Compute R_2 for $f(x) = \sqrt{x}$ at $x_0 = 9$ and investigate it in the manner of this handout:

   (a) Plot R_2 on $[8, 10]$ and examine its shape. Does it look like a simple polynomial of low degree?

   (b) Perform polynomial regression (fit) using `polyfit` and check its accuracy with a plot similar to Figure 3.

   (c) If the results in the previous part are good, try to derive an analytical expression for R_2. If the numerical fit is not good, try shrinking the interval and see if that improves the fit (generally, it should).

(d) Provided that you got somewhere in the last part, give an improved approximation to $\sqrt{10}$. How many digits of accuracy does the improved formula provide?

3. The equation of the tangent line can be derived as follows. First construct the secant through $(x_0, f(x_0))$ and $(x_0 + h, f(x_0 + h))$:

$$y = f(x_0) + \frac{f(x_0 + h) - f(x_0)}{h}(x - x_0).$$

Next take the limit as $h \to 0$: the result is a formula for T_1. Try to derive a formula for T_2 in the same fashion. Take three points on the curve: $(x_0, f(x_0))$, $(x_0 + h, f(x_0 + h))$, and $(x_0 - h, f(x_0 - h))$. Then find the equation of the parabola

$$y = a\,x^2 + b\,x + c$$

that passes through these points. Finally, take the limit as $h \to 0$. Confirm your algebra with at least two numerical examples. Also, in this problem pay particular attention to mathematical exposition. Read your solution aloud substituting mathematical formulas and numbers with the word "blah". If all you hear is "plug blah into blah then divide by blah", add some good prose. Use the style of this handout or Stewart's Calculus as a guide to mathematical exposition.

4. `Matlab` has two random number generators: `rand` and `randn`. The first produces random numbers distributed uniformly in $[0, 1]$; the second produces normally distributed random numbers. Either command can be used to generate random polynomials which are good for numerical experimentation. For instance,

```
p = rand(3,1)
```

produces a quadratic with coefficients `p(1)`, `p(2)`, `p(3)` (in *descending* order) picked at random from $[0, 1]$. Redo the first exercise with a random polynomial of order 6 and $x_0 = 0$. You may want to read the help files for `polyval`, `polyfit`, and `polyder` if you are not familiar with `Matlab`'s commands for manipulating polynomials.