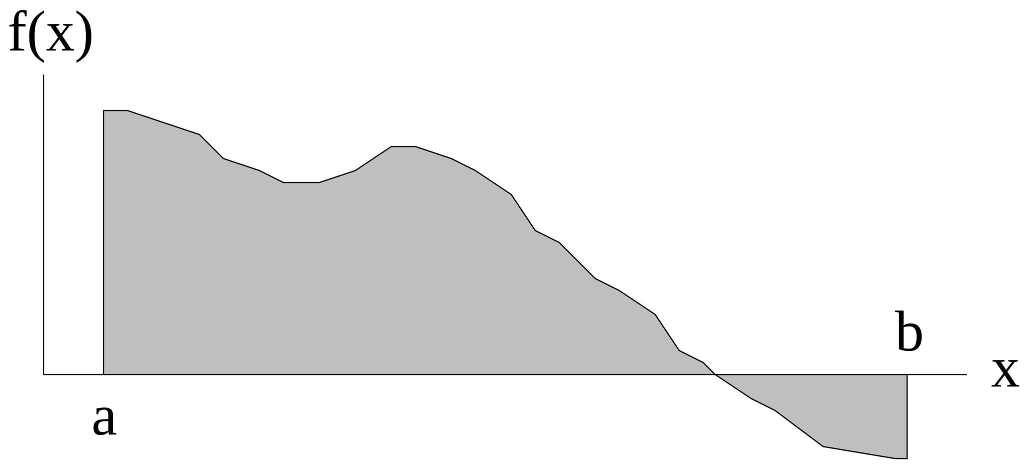
The Integration Problem

Integration is the process of finding totals, areas, volumes, etc.

The integral from a to b of a function $f(x)$, written as

$$\int_a^b f(x)\,dx$$

is the area beneath the curve defined by $f(x)$:



Where $f(x)$ is negative, the area counts as being negative.

The integral of a function of two variables is the volume under the surface it defines, and so forth in higher dimensions.

Applications of Integration

- In statistics, the probability of an event is the integral of the probability density for all the possible ways it could happen. Eg,

$$\Pr(x > 10) \;=\; \int_{10}^{\infty} p(x)\, dx$$

- The *arc length* for a curve can be defined in terms of a parametric representation, $(x(t), y(t))$, with $t_0 \le t \le t_1$, as

$$\int_{t_0}^{t_1} \sqrt{x'(t)^2 + y'(t)^2}\, dt$$

- When rendering an image with intensity function $I(x, y)$, in terms of 1×1 pixels, we should set pixel p, extending over $x \in [x_p - \frac{1}{2},\, x_p + \frac{1}{2}]$ and $y \in [y_p - \frac{1}{2},\, y_p + \frac{1}{2}]$, to have intensity

$$\int_{x_p-\frac{1}{2}}^{x_p+\frac{1}{2}} \int_{y_p-\frac{1}{2}}^{y_p+\frac{1}{2}} I(x, y)\, dx\, dy$$

Symbolic Integration

As you know from calculus, if we can find a function $F(x)$ whose derivative is $f(x)$, then we can evaluate an integral for $f(x)$ as

$$\int_a^b f(x)\,dx \;=\; F(b) - F(a)$$

Unfortunately, finding such an F (an "anti-derivative" of f) is not always easy. Maple knows many tricks for doing this, so it sometimes succeeds:

```
> int((1+x)/(1+x^2),x);
                                    2
                        1/2 ln(1 + x ) + arctan(x)

> int((1+x)/(1+x^2),x=0..1);

                            1/2 ln(2) + 1/4 Pi
```

But often it won't succeed (and neither will any other program or person).

What if Symbolic Integration Fails?

One time-honoured thing to do when you can't find an integral is to just define a new function to be the answer! For example, the "Gamma" function:

$$\Gamma(a) \;=\; \int_0^\infty x^{a-1} \operatorname{exp}(-x)\, dx$$

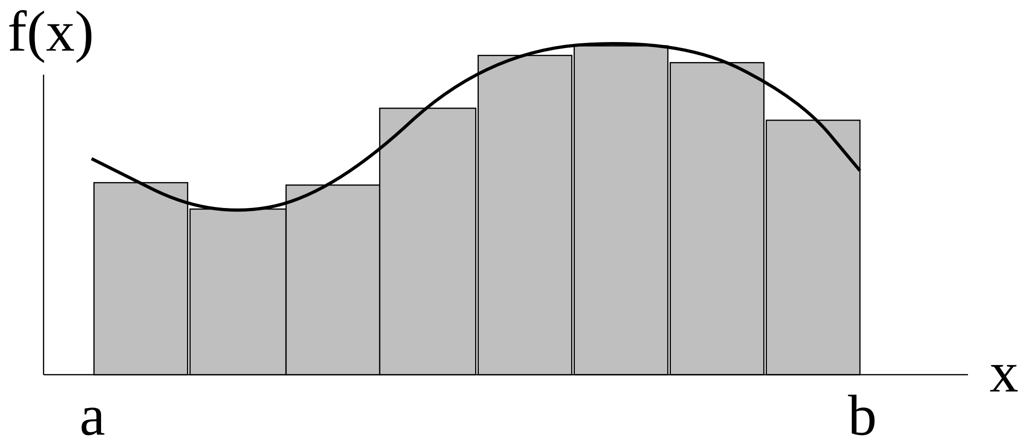
You can then try to figure out how to approximate the new function, and use it to solve other integrals. Eg,

$$\int_0^1 x^{a-1}\, (1-x)^{b-1}\, dx \;=\; \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

If we don't find this satisfying (eg, we want an actual answer!), then we can use *numerical integration*, also called *numerical quadrature*.

Numerical Integration with Rectangles

One way to approximate an integral is as the sum of the areas of rectangles:



Using n rectangles, of width $h = (b - a)/n$:

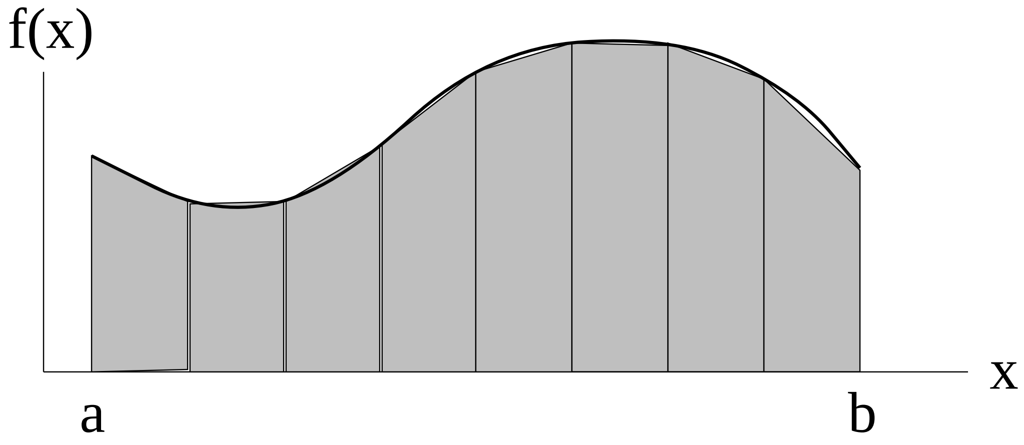
$$\int_a^b f(x)\,dx \;\approx\; \sum_{i=1}^{n} h\,f(a + (i - 1/2)h)$$

The following Maple program implements this:

```
int_rectangle := proc(fn,rng,n)
    local low, high, sum, i, h;
    low  := evalf(op(1,rng));
    high := evalf(op(2,rng));
    h := (high-low)/n;
    sum := 0;
    for i from 1 to n do
        sum := sum + h * fn(low+(i-0.5)*h);
    od;
    sum;
end:
```

Numerical Integration with Trapezoids

Rather than using a piecewise constant approximation to the function, we might use a piecewise linear approximation:



The integral is then the sum of the areas of n trapezoids.

The formula is as follows (with $h = (b - a)/n$):

$$\int_a^b f(x)\,dx \;\approx\; \sum_{i=1}^{n} h\,\frac{f(a+(i-1)h) + f(a+ih)}{2}$$

We can combine some terms, to give

$$\int_a^b f(x)\,dx \;\approx\; \frac{h}{2}(f(a) + f(b)) \;+\; \sum_{i=1}^{n-1} h\,f(a+ih)$$

Two Maple Programs for Integration Using Trapezoids

```
int_trapezoid1 := proc(fn,rng,n)
    local low, high, sum, i, h;
    low  := evalf(op(1,rng));
    high := evalf(op(2,rng));
    h := (high-low)/n;
    sum := 0;
    for i from 1 to n do
        sum := sum
          + (h/2) * (fn(low+(i-1)*h) + fn(low+i*h));
    od;
    sum;
end:
```

```
int_trapezoid2 := proc(fn,rng,n)
    local low, high, sum, i, h;
    low  := evalf(op(1,rng));
    high := evalf(op(2,rng));
    h := (high-low)/n;
    sum := (fn(low) + fn(high)) / 2;
    for i from 1 to n-1 do
        sum := sum + fn(low+i*h);
    od;
    h * sum;
end:
```

Accuracy of Integration Using Rectangles and Trapezoids

Some experimental results:

```
> int(sin(x),x=0..Pi/2);

                              1

> int_rectangle(x->sin(x),0..Pi/2,10) - 1;

                         .00102882415

> int_trapezoid1(x->sin(x),0..Pi/2,10) - 1;

                        -.002057013638

> int_rectangle(x->sin(x),0..Pi/2,100) - 1;

                         .00001028092

> int_trapezoid1(x->sin(x),0..Pi/2,100) - 1;

                        -.000020561752
```

For both methods, the error seems to be proportional to $1/n^2$. Why doesn't the better piecewise linear approximation of the trapezoid method produce a better answer?

Integration by Interpolation

The trapezoid method can be seen as first finding a piecewise linear interpolant, then computing the integral of the interpolant.

We can use any other interpolation method instead. For example:

- Piecewise cubic interpolation with Catmull-Rom constraints. The book considers this.
- Piecewise cubic interpolation with the natural spline constraints. This may be less attractive — it's harder to figure out the interpolant for a given number of data points.
- Piecewise Lagrange interpolation, of any degree.

Note: The integral of a piecewise interpolant is just the sum of the integrals of the pieces.

Integration by Piecewise Quadratic Interpolation: Simpson's Rule

Recall the Lagrange interpolation formula:

$$x(t) \;=\; \sum_{i=0}^{n} \; x_i \; \prod_{\substack{j=0 \\ j \neq i}}^{n} \; \frac{t - t_j}{t_i - t_j}$$

For $n = 3$, $t_0 = 0$, $t_1 = 1/2$, $t_2 = 1$, we get

$$\begin{aligned} x(t) \;&=\; x_0 \, \tfrac{t-1/2}{0-1/2} \, \tfrac{t-1}{0-1} \;+\; x_1 \, \tfrac{t-0}{1/2-0} \, \tfrac{t-1}{1/2-1} \;+\; x_2 \, \tfrac{t-0}{1-0} \, \tfrac{t-1/2}{1-1/2} \\ &=\; x_0 \, (2t^2 - 3t + 1) \;-\; x_1 \, (4t^2 - 4t) \;+\; x_2 \, (2t^2 - t) \end{aligned}$$

From this, we get

$$\begin{aligned} &\int_0^1 x(t) \, dt \\ &\;=\; \Big[x_0 \, (\tfrac{2}{3}t^3 - \tfrac{3}{2}t^2 + t)) \;-\; x_1 \, (\tfrac{4}{3}t^3 - \tfrac{4}{2}t^2) \;+\; x_2 \, (\tfrac{2}{3}t^3 - \tfrac{1}{2}t^2) \Big]_0^1 \\ &\;=\; \Big[x_0 \, (\tfrac{2}{3} - \tfrac{3}{2} + 1)) \;-\; x_1 \, (\tfrac{4}{3} - \tfrac{4}{2}) \;+\; x_2 \, (\tfrac{2}{3} - \tfrac{1}{2}) \Big] \;-\; 0 \\ &\;=\; \frac{1}{6} \Big[x_0 \;+\; 4 \, x_1 \;+\; x_2 \Big] \end{aligned}$$

Compound Simpson's Rule

We can integrate a function by applying Simpson's Rule to n pieces, as follows:

$$\int_a^b f(x)\,dx$$

$$= \sum_{i=0}^{n-1} \frac{h}{6} \Big[f(a+ih) + 4f(a+(i+\tfrac{1}{2})h) + f(a+(i+1)h)\Big]$$

where $h = (b-a)/n$.

We can combine terms in this sum to get

$$\int_a^b f(x)\,dx$$

$$= \frac{h}{6} \Big[f(a) + 4f(a+\tfrac{1}{2}h) + 2f(a+h) + 4f(a+\tfrac{3}{2}h) + 2f(a+2h) + \cdots + 4f(a+(n-\tfrac{1}{2})h) + f(b)\Big]$$

Simpson's Rule in Maple

```
int_simpson := proc(fn,rng,n)
    local low, high, sum, i, h;
    low  := evalf(op(1,rng));
    high := evalf(op(2,rng));
    h := (high-low)/n;
    sum := (h/6) * (fn(low) + fn(high));
    for i from 1 to n-1 do
        sum := sum + (h/3) * fn(low+i*h);
    od;
    for i from 1 to n do
        sum := sum + (2*h/3) * fn(low+(i-0.5)*h);
    od;
    sum;
end:
```

How accurate is it?

```
> Digits:=30:

> int_simpson(x->sin(x),0..Pi/2,10) - 1;
                                                   -6
                        .211546591423602078005071*10

> int_simpson(x->sin(x),0..Pi/2,100) - 1;
                                                -10
                         .2113928089365798526*10
```

The error seems to be going down as $1/n^4$.

General Newton-Cotes Rules

The rectangle, trapezoid, and Simpson's methods are examples of "Newton-Cotes" rules, in which the function is evaluated at equally-spaced points.

There are other integration methods that don't use equally-spaced points, or that adapt the number of point to the behaviour of the function.

We can distinguish two kinds of integration rules:

- *Closed* rules, such as the trapezoid method and Simpson's Rule, evaluate the function at the far left and far right endpoints.
- *Open* rules, such as the rectangle method we looked at, do not evaluate the function at the endpoints.

Integrating Functions with Singularities

Integrals can be well-defined even for functions that go to infinity at an endpoint. For example:

$$\int_0^1 x^{-1/2}\,dx \;=\; \left[2\,x^{1/2}\right]_0^1 \;=\; 2$$

How can we integrate such a function numerically?

A closed rule will not work, since the function is undefined at one of the endpoints.

Open rules will be OK, however, though because of the singularity, they may not converge as rapidly as one might hope.

We could also try a transformation to get rid of the singularity.

Integrating Over Infinite Regions

Integrals can also be well-defined when they are over a region of infinite size. For example:

$$\int_0^\infty e^{-x}\,dx \;=\; \left[-e^{-x}\right]_0^\infty \;=\; 1$$

How can we evaluate such an integral numerically?

Newton-Cotes rules, with equally-spaced points, can't work directly — we'd need an infinite number of points!

Some other schemes do work. They select points that go further out as the number of points goes up.

We can also solve these integrals by using a transformation that converts them to integrals over a finite range.

A Transformation from $[0, \infty]$ *to* $[0, 1]$

An example: We can evaluate

$$I = \int_0^\infty f(x)\, dx$$

by using the transformaton

$$y = 1 - e^{-x}, \qquad x = -\log(1 - y)$$

This takes the bounds $0 \to 0$ and $\infty \to 1$.
Also, $dx = dy \,/\, (1 - y)$.

The result is therefore:

$$I = \int_0^1 \frac{f(-\log(1 - y))}{1 - y}\, dy$$

This is not the only possible transformation from $[0, \infty]$ to $[0, 1]$. Which is best depends on what $f(x)$ is like.