

Integral Calculus with Lab4IQM

Volume 2, Chapters 1–7: integration, differential equations, series, parametric and polar curves

Lab4IQM Integral Calculus module 1.0 · lab4iqm.com

This companion follows the chapters and sections of *Calculus Volume 2* by Gilbert Strang, Edwin Herman and others (OpenStax, Rice University), a free textbook that students can read online or download. Chapter 1 of Volume 2 is the same as Chapter 5 of Volume 1, so this module continues where the Differential Calculus module stops. For each section the companion gives the Lab4IQM commands that draw the ideas of the section and compute its results: antiderivatives and exact values come from Lab4IQM's own symbolic engine, and every definite integral is also computed numerically, independently, as a check. Read the textbook for the explanations and the exercises; use Lab4IQM to see the areas and solids, test your intuition, and check your answers. This version covers all seven chapters of Volume 2.

Type the commands in the Lab4IQM console after `import intcalc` (the module needs Lab4IQM 1.6). Solids (Sections 2.2–2.4) are drawn in 3D: turn them with the mouse in Blender's view. `help(integral)` explains any command; the Examples panel has examples for every section; `view2d()` looks at a graph straight on. Every command that draws also prints its results.

Typing functions and intervals

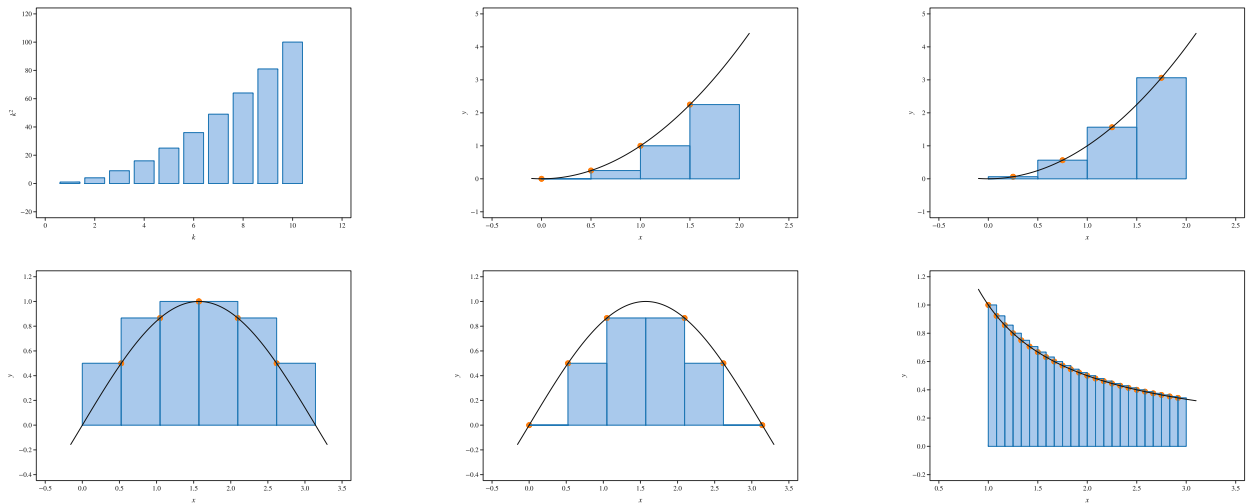
<code>f=x^3 - 3*x, f=2x^2</code>	powers with <code>^</code> ; <code>2x</code> means <code>2*x</code>
<code>sqrt cbrt, exp log, sin cos tan sec</code>	the usual functions (radians)
<code>csc cot</code>	
<code>asin acos atan asec acsc acot</code>	(or inverse trigonometric functions
<code>arcsin ...)</code>	
<code>x=0..pi, x=sqrt(2)..2</code>	an interval of integration (its ends are kept exact)
<code>f=sin(t), t=0..5</code>	the variable can be <i>t</i> (or <i>u</i> , <i>s</i> , ...)
<code>f="x for x < 1; 2 - x for x >= 1"</code>	a piecewise function (quoted)
<code>xlim=-4..4, ylim=-2..2</code>	the window

1.1 Approximating Areas

<code>sigma(f=k^2, k=1..10)</code>	385, and $\sum_{k=1}^n k^2 = n^3/3 + n^2/2 + n/6$
<code>sigma(f=2*k + 3, k=3..20)</code>	the rules of sums, $S(20) - S(2)$
<code>sigma(f=(1/2)^k, k=0..10)</code>	a geometric sum and its limit
<code>riemann(f=x^2, x=0..2, n=4)</code>	the left-endpoint sum $L_4 = 7/4$
<code>riemann(f=x^2, x=0..2, n=4, method=mid)</code>	right, mid, upper, lower, trap
<code>riemann(f=1/x, x=1..3, n=(4, 8, 16, 32, 64), method=all)</code>	

`sigma` adds the terms, draws them as bars and, for a polynomial in k , uses the rules of sums with the formulas for $\sum k$, $\sum k^2$, $\sum k^3$ to give the sum of the first n terms as a polynomial in n . `riemann` draws the rectangles (blue above the axis, red below) and the sample points, prints Δx , the sample points and the sum (exactly when it is a rational number) and compares it with the integral. It says when a left or right sum must be an underestimate or an overestimate (an increasing or decreasing

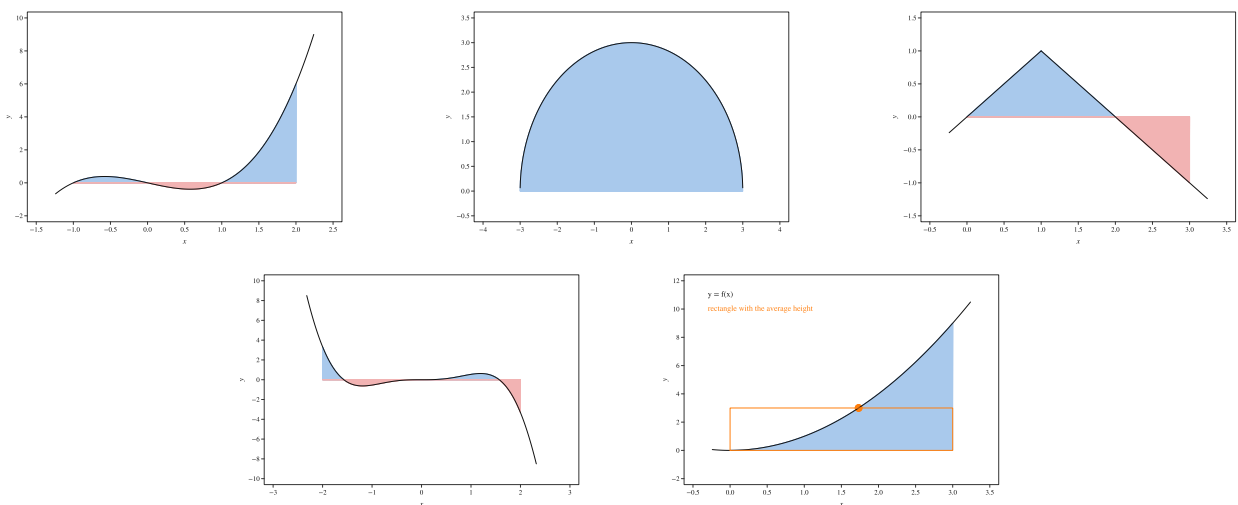
f). With several n it prints a table that shows the sums approaching the integral, and how fast the error shrinks: like $1/n$ for left and right sums, like $1/n^2$ for the midpoint sum.



1.2 The Definite Integral

```
integral(f=x^3 - x, x=-1..2)      net signed area 9/4, total area 11/4
integral(f=sqrt(9 - x^2), x=-3..3) area of a half disc: 9*pi/2
integral(f="x for x < 1; 2 - x for x >= 1", x=0..3)
integral(f=x^3*cos(x), x=-2..2)  odd function, symmetric interval: 0
integral(f=sin(x), x=pi..0)      int_a^b = -int_b^a
average(f=x^2, x=0..3)           average value 3 at c = sqrt(3)
```

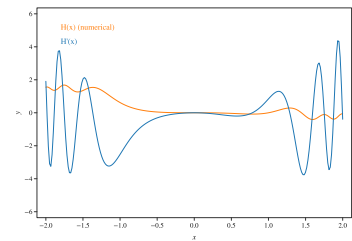
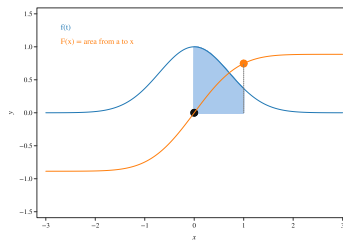
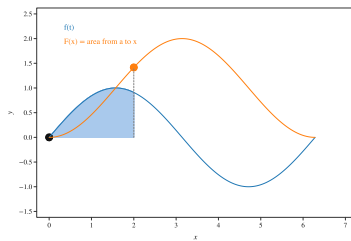
The integral is the net signed area: the region above the axis is blue, the region below is red, and the console gives both areas, the total area $\int_a^b |f(x)| dx$ and the zeros of f that split it. It names the properties that apply: $\int_a^b f = -\int_b^a f$, odd and even functions on $[-a, a]$, and the bounds $m(b-a) \leq \int_a^b f \leq M(b-a)$. When no antiderivative is found (the half disc), the value is computed numerically and recognised when it is a simple exact number. Piecewise functions are integrated piece by piece. **average** draws the rectangle with the same area, of height f_{ave} , and the points c of the mean value theorem for integrals.



1.3 The Fundamental Theorem of Calculus

<code>areafunction(f=sin(t), a=0, x=0..2*pi, at=2)</code>	$F(x) = 1 - \cos(x)$, $F' = f$
<code>areafunction(f=exp(-t^2), a=0, x=-3..3, at=1)</code>	a new function (numerical)
<code>ftc(f=sqrt(1 + t^2), lower=1, upper=x^3)</code>	$H'(x) = 3*x^2*\sqrt{x^6 + 1}$
<code>ftc(f=sin(t^2), lower=x, upper=x^2)</code>	both limits variable
<code>integral(f=sin(x), x=0..pi)</code>	part 2: $F(\pi) - F(0) = 2$
<code>integral(f=1/x^2, x=-1..1)</code>	unbounded inside: improper

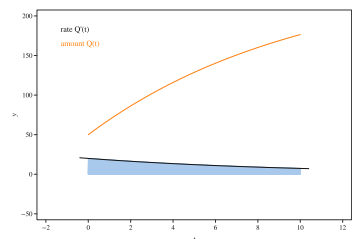
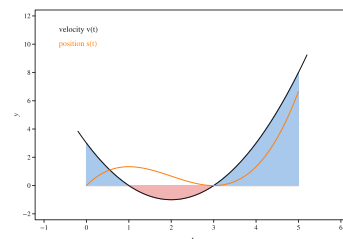
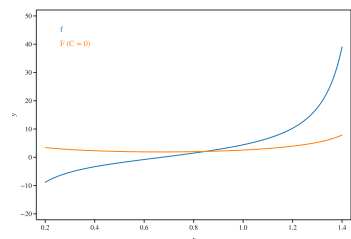
areafunction draws f and its area function $F(x) = \int_a^x f(t) dt$ together: F rises where $f > 0$, falls where $f < 0$, and turns where f changes sign (part 1 of the theorem: $F' = f$). **ftc** differentiates an integral with variable limits by the theorem and the chain rule, $\frac{d}{dx} \int_{a(x)}^{b(x)} f(t) dt = f(b(x)) b'(x) - f(a(x)) a'(x)$, which needs no antiderivative, and draws H and H' . **integral** evaluates $F(b) - F(a)$ (part 2), with every rule used to find F , and checks the result against an independent numerical value; it refuses to apply the theorem when f is unbounded inside the interval, the classic error $\int_{-1}^1 x^{-2} dx = -2$.



1.4 Integration Formulas and the Net Change Theorem

<code>integral(f=3*x^2 - 2/x + sec(x)^2)</code>	indefinite: $F(x) + C$, with the rules used
<code>netchange(v=t^2 - 4*t + 3, t=0..5)</code>	displacement $20/3$, distance $28/3$
<code>netchange(v=3*cos(t), t=0..2*pi, s0=1)</code>	position $s(t)$
<code>netchange(rate=20*exp(-0.1*t), t=0..10, initial=50)</code>	

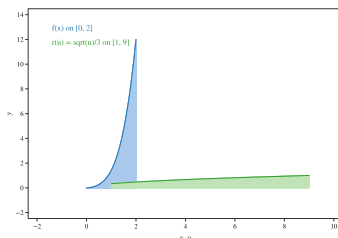
Without an interval, **integral** gives the indefinite integral, lists each rule (sum, constant multiple, power rule, the table of basic integrals) and checks the answer by differentiating. **netchange** reads a velocity: the displacement $\int v dt$ (blue minus red), the total distance $\int |v| dt$ (blue plus red), the times where the motion changes direction, and the position $s(t)$ (orange). With **rate=** it gives the amount $Q(b) = Q(a) + \int_a^b Q'(t) dt$.



1.5 Substitution

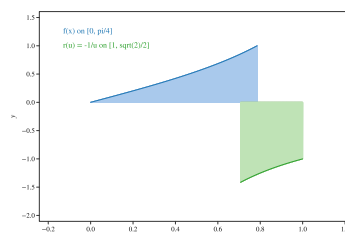
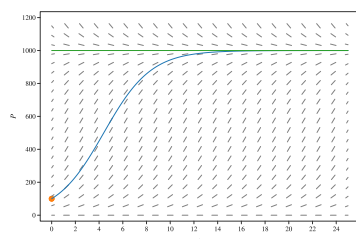
<code>substitute(f=x*(x^2 + 1)^5, u=x^2 + 1)</code>	$u, du, \int u^{5/2} du$, back to x , check
<code>substitute(f=x^2*sqrt(x^3 + 1), x=0..2)</code>	u found automatically; limits 1..9
<code>substitute(f=x*cos(x^2), u=x^3)</code>	a u that does not work, and why

`substitute` writes out the five steps: choose $u = g(x)$ and $du = g'(x) dx$; rewrite the integral in u (with no x left); integrate in u (rules shown); substitute back; check by differentiating. Without `u=` it looks for an inside function that works. For a definite integral the limits change with u ; both regions are drawn, the one in x (blue) and the one in u (green), and they have the same area. If a choice of u leaves x in the integrand, the console shows what is left so that another u can be tried.



1.6 Integrals Involving Exponential and Logarithmic Functions

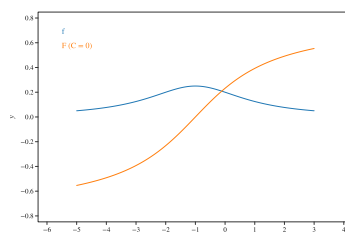
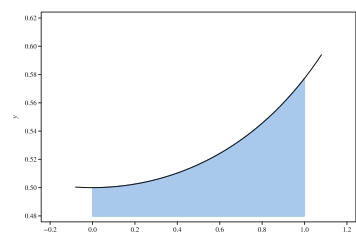
<code>integral(f=exp(x)/(1 + exp(x)))</code>	substitution $u = \exp(x) + 1$: $\ln(e^x + 1)$
<code>integral(f=3^x)</code>	$3^x / \ln(3)$
<code>integral(f=log(x)/x)</code>	substitution $u = \ln(x)$
<code>substitute(f=tan(x), x=0..pi/4)</code>	$\tan = \sin/\cos$, $u = \cos(x)$: $\ln(2)/2$
<code>integral(f=1/x, x=1..e)</code>	1



1.7 Integrals Resulting in Inverse Trigonometric Functions

<code>integral(f=1/sqrt(4 - x^2), x=0..1)</code>	$\arcsin(x/2)$: $\pi/6$
<code>integral(f=1/(1 + 9*x^2))</code>	$\arctan(3*x)/3$
<code>integral(f=1/(x*sqrt(x^2 - 1)), x=sqrt(2)..2)</code>	$\operatorname{arccsc}(x)$: $\pi/12$
<code>integral(f=1/(x^2 + 2*x + 5))</code>	complete the square: $(x + 1)^2 + 4$
<code>integral(f=(2*x + 3)/(x^2 + 4))</code>	$\ln(x^2 + 4) + (3/2) \arctan(x/2)$
<code>integral(f=exp(x)/(1 + exp(2*x)))</code>	substitution, then $\arctan$

The engine knows the three forms $\int du/\sqrt{a^2 - u^2} = \sin^{-1}(u/a)$, $\int du/(a^2 + u^2) = \frac{1}{a} \tan^{-1}(u/a)$ and $\int du/(u\sqrt{u^2 - a^2}) = \frac{1}{a} \sec^{-1}(|u|/a)$, completes the square in quadratics, splits a linear numerator into a multiple of Q' plus a constant, and gives exact values such as $\sec^{-1} 2 - \sec^{-1} \sqrt{2} = \pi/12$.

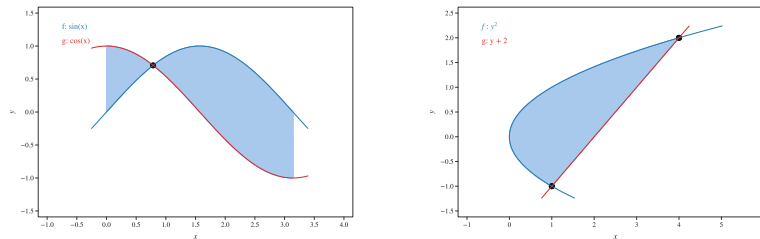


2.1 Areas between Curves

```
between(f=x + 2, g=x^2)
between(f=sin(x), g=cos(x), x=0..pi)
between(f=y^2, g=y + 2)
```

```
between the intersections x = -1 and 2: 9/2
the curves cross at pi/4: 2*sqrt(2)
sideways: int (right - left) dy
```

`between` integrates (top – bottom); where the curves cross it splits the interval and says what the unsplit integral would wrongly give. Without an interval it uses the region between the first and last intersections. Functions of y describe a region sideways ($x = f(y)$), which avoids splitting for regions such as the one between $x = y^2$ and $x = y + 2$.

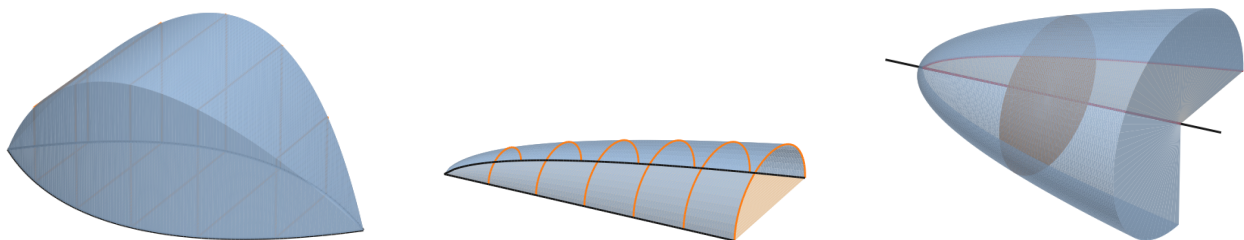


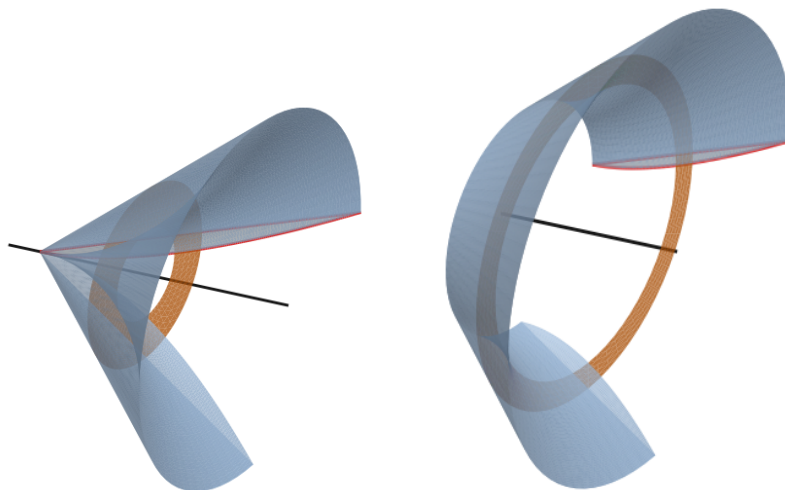
2.2 Determining Volumes by Slicing

```
slices(f=1 - x^2/4, g=x^2/4 - 1, x=-2..2, shape=square)
slices(f=sqrt(x), x=0..4, shape=semicircle)
slices(area=(4 - x)^2, x=0..4)
revolve(f=sqrt(x), x=0..4)
revolve(f=x, g=x^2, x=0..1, about=x)
revolve(f=x, g=x^2, x=0..1, about="y=-1")
```

```
square slices: 128/15
semicircles: pi
a pyramid: 64/3
disks: 8*pi
washers: 2*pi/15
about a line: 7*pi/15
```

`slices` builds a solid over a base region from its cross-sections (squares, equilateral triangles, semicircles, disks, right triangles) and integrates their area $A(x)$. `revolve` turns a region about an axis; slices perpendicular to the axis are disks or washers, $V = \int \pi (R^2 - r^2) dx$, and the console prints R and r . Three quarters of the solid are drawn, with the region (orange) in its plane and one representative washer.

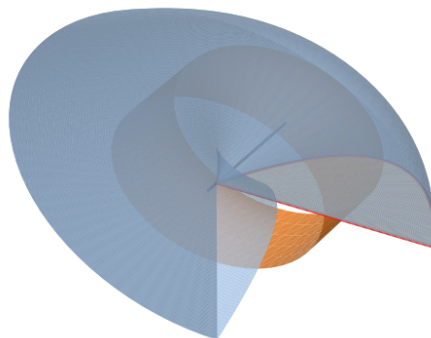




2.3 Volumes of Revolution: Cylindrical Shells

<code>revolve(f=2*x - x^2, x=0..2, about=y)</code>	<code>shells: int 2 pi x (2x - x^2) dx = 8*pi/3</code>
<code>revolve(f=x, g=x^2, x=0..1, about="x=2")</code>	<code>radius 2 - x: pi/2</code>
<code>revolve(f=sqrt(y), y=0..4, about=y)</code>	<code>the region by functions of y: disks</code>
<code>revolve(f=x, g=x^2, x=0..1, about=y, method=washer)</code>	<code>explains why not</code>

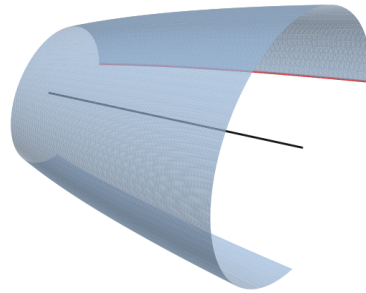
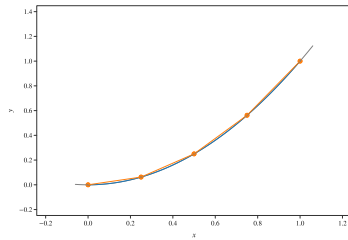
When the strips of the region run parallel to the axis, each sweeps a cylindrical shell of radius $p(x)$ and height $h(x)$: $V = \int 2\pi p(x) h(x) dx$. The method follows from the axis and the variable; describing the region by functions of the other variable switches between washers and shells for the same solid, which is how the two methods are compared.



2.4 Arc Length of a Curve and Surface Area

<code>arclength(f=x^(3/2), x=0..1)</code>	<code>(13*sqrt(13) - 8)/27</code>
<code>arclength(f=x^3/6 + 1/(2*x), x=1..2)</code>	<code>1 + f'^2 is a perfect square: 17/12</code>
<code>arclength(f=x^2, x=0..1, n=8)</code>	<code>numerical, and 8 segments</code>
<code>arclength(f=sqrt(x), x=1..4, about=x)</code>	<code>surface area: S = int 2 pi f sqrt(1 + f'^2)</code>
<code>dx</code>	

`arclength` prints f' , $1 + f'^2$ and the integrand. It recognises the perfect squares that textbook exercises are built on, and $1 + \sinh^2 = \cosh^2$, $1 + \tan^2 = \sec^2$; most other lengths have no elementary formula and are computed numerically. The graph shows the curve with n segments whose length approaches L . With `about=` it gives the surface area of revolution and draws the surface.



2.5 Physical Applications

```

mass(density=1 + x^2, x=0..2)
mass(density=sqrt(r), r=0..4, shape=disk)
work(force=3*x^2, x=0..2)
work(force=10, stretch=0.2, x=0..0.5)
pump(tank=cone, r=2, h=4, fill=3, out=1)
fluidforce(plate=triangle, W=6, H=3, top=1)
fluidforce(width=2*sqrt(d), d=0..4)

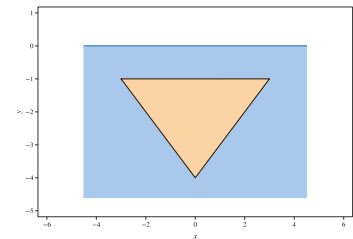
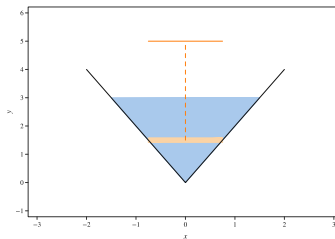
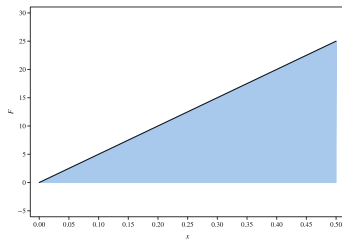
```

```

a rod: m = 14/3 (centre of mass 9/7)
rings of area 2 pi r dr: 128*pi/5
W = int F dx = 8
Hooke: k = 50, W = 25/4
slab by slab: 121275*pi/2 J
F = int rho g d w(d) dd
any plate

```

Each command prints the slice it adds up (a piece of the rod, a ring, a slab of liquid lifted $h - y$, a strip of the plate at depth d) and the integral. Water ($\rho = 1000$, $g = 9.8$) gives joules and newtons; for feet and pounds use `density=62.4`, `g=1`.



2.6 Moments and Centers of Mass

```

centroid(f=sqrt(x), x=0..4)
centroid(f=4 - x^2, x=-2..2)
centroid(f=x, g=x^2, x=0..1, about=x)
centroid(points=[(-2, 0), (1, 1), (3, -1)], masses=(2, 3, 1))
mass(density=1 + x^2, x=0..2)

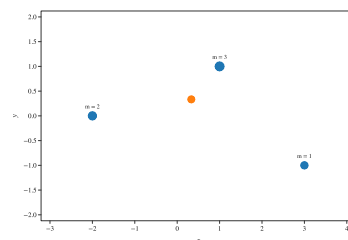
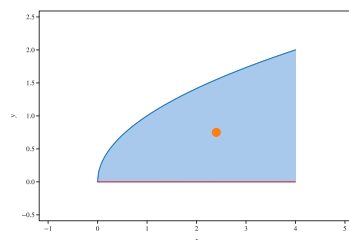
```

```

(12/5, 3/4)
symmetric: xbar = 0
Pappus: V = 2 pi d A = 2*pi/15
the centre of mass of a rod

```

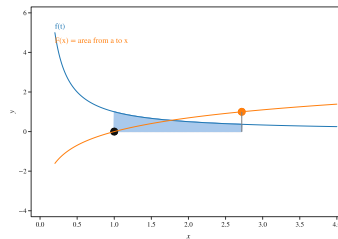
`centroid` prints the area, the moments $M_y = \int x(f - g) dx$ and $M_x = \int \frac{1}{2}(f^2 - g^2) dx$, and the centroid; with `about=` it applies the theorem of Pappus, which reproduces the volume found by `revolve`.



2.7 Integrals, Exponential Functions, and Logarithms

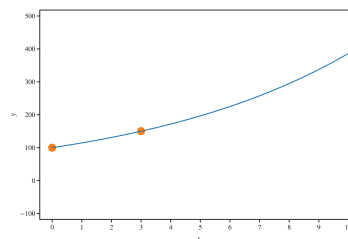
<code>areafunction(f=1/t, a=1, x=0.2..4, at=e)</code>	$\ln x$ as the area under $1/t$: $F(e) = 1$
<code>integral(f=1/x, x=1..e)</code>	1
<code>integral(f=exp(x)/(1 + exp(x)))</code>	$\ln(e^x + 1)$

The section defines $\ln x = \int_1^x dt/t$ and e as the number with $\ln e = 1$; `areafunction` shows that area function and its value 1 at e .



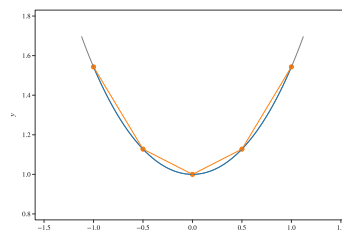
2.8 Exponential Growth and Decay

<code>expmodel(y0=200, k=0.05, t=0..40)</code>	doubling time $\ln 2 / k$
<code>expmodel(points=((0, 100), (3, 150)), t=0..10, target=400)</code>	k from two values; when $y = 400$
<code>expmodel(y0=10, halflife=5730, t=0..20000, at=1000)</code>	radiocarbon
<code>expmodel(y0=1000, doubling=7, t=0..30)</code>	



2.9 Calculus of the Hyperbolic Functions

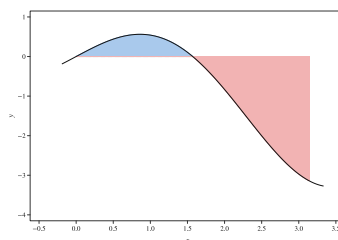
<code>arclength(f=cosh(x), x=-1..1)</code>	the catenary: $2 \cdot \sinh(1)$
<code>integral(f=sinh(x)^2)</code>	$\sinh(2x)/4 - x/2$
<code>integral(f=1/cosh(2*x)^2)</code>	$\tanh(2x)/2$
<code>integral(f=1/sqrt(x^2 + 4*x + 8))</code>	$\operatorname{asinh}((x + 2)/2)$



3.1 Integration by Parts

<code>parts(f=x*exp(x))</code>	<code>u = x (LIATE), dv = exp(x) dx</code>
<code>parts(f=x^2*sin(x), tabular=true)</code>	<code>the table method</code>
<code>parts(f=log(x))</code>	<code>u = ln x, dv = dx</code>
<code>parts(f=exp(x)*cos(x))</code>	<code>the integral comes back: solve for it</code>
<code>parts(f=x*cos(x), u=x, x=0..pi)</code>	<code>a definite integral: [u v] - int v du = -2</code>

`parts` prints u , dv , du , v and the new integral $\int v du$, then how that one is done. u is chosen by LIATE (logarithmic, inverse trigonometric, algebraic, trigonometric, exponential) unless $u=$ is given; a poor choice shows a harder integral. When the original integral reappears, as in $\int e^x \cos x dx$, the equation $I = A - I$ is solved for I . `integral` uses the same method automatically and lists it among its steps.



3.2 Trigonometric Integrals

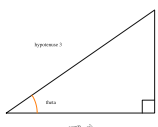
<code>integral(f=sin(x)^3*cos(x)^2)</code>	<code>odd power of sin: w = cos x</code>
<code>integral(f=sin(x)^2*cos(x)^2)</code>	<code>even powers: half-angle formulas</code>
<code>integral(f=tan(x)^2*sec(x)^4)</code>	<code>even power of sec: w = tan x</code>
<code>integral(f=tan(x)^3*sec(x))</code>	<code>odd power of tan: w = sec x</code>
<code>integral(f=sec(x)^3)</code>	<code>the reduction formula</code>
<code>integral(f=sin(3*x)*cos(2*x))</code>	<code>product-to-sum formula</code>

The steps name the strategy of the section: keep one factor and convert the rest with $\sin^2 + \cos^2 = 1$ or $1 + \tan^2 = \sec^2$, use the half-angle formulas for even powers, the reduction formulas for powers of sec and csc, and the product-to-sum formulas.

3.3 Trigonometric Substitution

<code>trigsub(f=sqrt(9 - x^2), x=0..3)</code>	<code>x = 3 sin(theta): 9*pi/4</code>
<code>trigsub(f=1/(x^2*sqrt(x^2 + 4)))</code>	<code>x = 2 tan(theta)</code>
<code>trigsub(f=sqrt(x^2 - 1)/x)</code>	<code>x = sec(theta)</code>
<code>integral(f=sqrt(x^2 + 2*x + 5))</code>	<code>complete the square first</code>

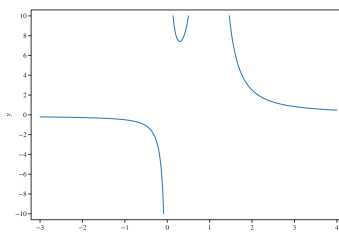
`trigsub` prints the substitution, the integral in θ with its steps, and the way back to x , and draws the reference triangle with its sides labelled.



3.4 Partial Fractions

<code>partialfractions(f=1/((x - 1)*(x + 2)))</code>	
<code>partialfractions(f=(x^2 + 1)/(x*(x - 1)^2))</code>	a repeated linear factor
<code>partialfractions(f=(3*x + 1)/(x^4 - 1))</code>	an irreducible quadratic factor
<code>partialfractions(f=x^4/(x^2 - 1))</code>	long division first

`partialfractions` factors the denominator exactly into linear and irreducible quadratic factors with their multiplicities, writes the form of the decomposition with unknown constants $A, B, C, \dots$, solves for them exactly, and integrates term by term.



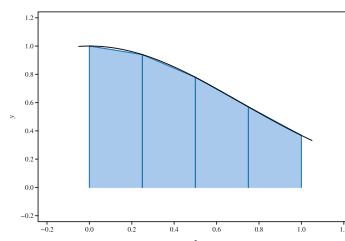
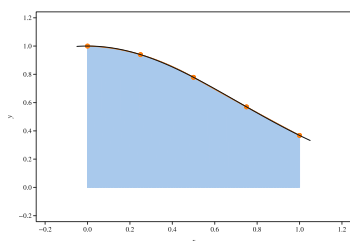
3.5 Other Strategies for Integration

<code>integral(f=sqrt(x)/(1 + x))</code>	a rationalizing substitution $u = \sqrt{x}$
<code>integral(f=1/(x^(1/3) + 1))</code>	$u = x^{1/3}$
<code>integral(f=exp(x)/(exp(2*x) - 1))</code>	substitution, then partial fractions

The section's tables of integrals and computer algebra systems are what `integral` is: it tries the basic rules, substitution (including rationalizing substitutions), parts, the trigonometric methods and partial fractions, says which one worked at every step, and checks the answer by differentiating it.

3.6 Numerical Integration

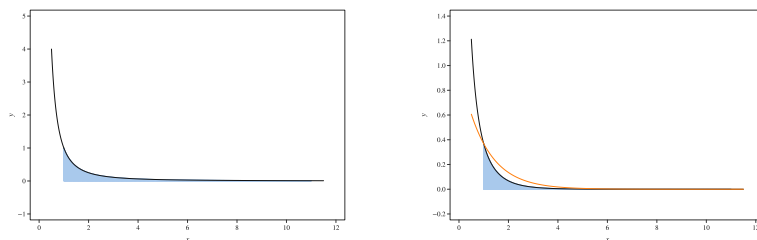
<code>numint(f=exp(-x^2), x=0..1, n=4)</code>	M_4, T_4, S_4 , their errors and the bounds
<code>numint(f=1/x, x=1..2, n=(2, 4, 8, 16))</code>	the errors shrink like $1/n^2$ and $1/n^4$
<code>numint(f=sin(x^2), x=0..1, tol=1e-6)</code>	the n each rule needs
<code>numint(f=exp(-x^2), x=0..1, n=4, method=trap)</code>	draw the trapezoids



3.7 Improper Integrals

<code>improper(f=1/x^2, x=1..inf)</code>	converges to 1
<code>improper(f=1/x, x=1..inf)</code>	diverges
<code>improper(f=1/(1 + x^2), x=-inf..inf)</code>	split at 0: π
<code>improper(f=1/sqrt(x), x=0..1)</code>	unbounded at 0: 2
<code>improper(f=1/x^2, x=-1..1)</code>	unbounded inside: diverges
<code>improper(f=exp(-x^2), x=0..inf)</code>	numerically: $\sqrt{\pi}/2$
<code>improper(f=exp(-x)/x, x=1..inf, compare=exp(-x))</code>	the comparison theorem

Each piece is written as a limit and evaluated; without an antiderivative the partial integrals are computed and watched as they settle. `integral` refuses an infinite interval and points to `improper`.



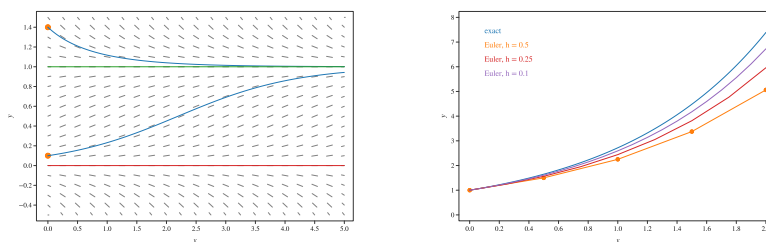
4.1 Basics of Differential Equations

```
verify(y=exp(2*x), de="y'' - 4*y = 0")           order 2: a solution
verify(y=C*exp(-x^2), de="y' + 2*x*y = 0")       a family of solutions
verify(y=3*cos(2*t) + sin(2*t), de="y'' + 4*y = 0", t0=0, y0=3)  with an initial value
```

4.2 Direction Fields and Numerical Methods

```
slopefield(f=x - y)                               y' = f(x, y): the direction
  field
slopefield(f=y*(1 - y), x=0.5, ylim=-0.5..1.5, points=((0, 0.1), (0, 1.4)))
eulermethod(f=x + y, x0=0, y0=1, h=0.1, n=10)     a table of Euler steps
eulermethod(f=y, x0=0, y0=1, h=(0.5, 0.25, 0.1), x1=2)  the error is proportional to h
```

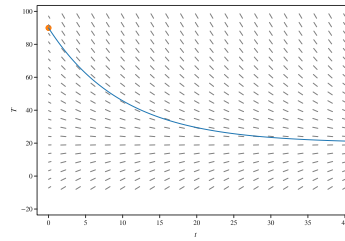
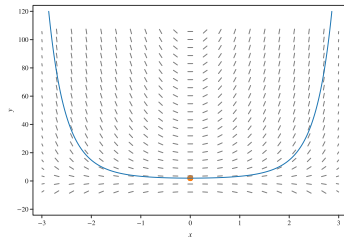
`slopefield` draws a short segment of slope $f(x, y)$ at each grid point and the solution curves through given points; for an autonomous equation $y' = f(y)$ it finds the equilibrium solutions and classifies them (green: stable, red: unstable). `eulermethod` prints the steps $y_{k+1} = y_k + h f(x_k, y_k)$ next to the exact solution.



4.3 Separable Equations

```
desolve(f=x*y, x0=0, y0=2)                       y = 2 exp(x^2/2)
desolve(dy/dx = x/y, x0=0, y0=3)                  y = sqrt(x^2 + 9)
desolve(dT/dt = -k*(T - 20), k=0.1, t0=0, T0=90, t=0..40)  Newton's law of cooling
```

`desolve` recognises $y' = g(x)h(y)$, separates, integrates both sides (each integral by the engine), solves for y , finds the constant from the initial value and checks the solution by substituting it into the equation. The zeros of h are equilibrium solutions. The direction field is drawn behind the solution.



4.4 The Logistic Equation

```
logistic(r=0.5, K=1000, P0=100, t=0..25)
      K/2
```

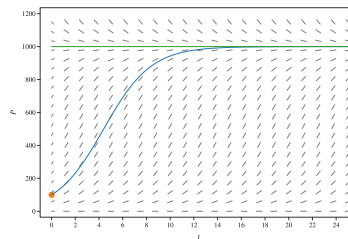
$P(t)$, carrying capacity, inflection at

```
logistic(r=0.2, K=500, P0=900, t=0..30)
```

starting above K

```
logistic(r=0.3, K=1000, T=100, P0=150, t=0..40)
```

a threshold population T



4.5 First-order Linear Equations

```
desolve(f=x - y, points=((0, 1), (0, -1), (0, 2)))
```

$y' + y = x$: $\mu = e^x$

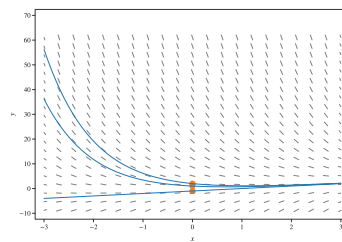
```
desolve(f=y/x + x^2, x0=1, y0=2, x=0.2..3)
```

$\mu = 1/x$

```
desolve(dI/dt = (E - R*I)/L, E=12, R=4, L=2, t0=0, I0=0)
```

an RL circuit

For $y' + p(x)y = q(x)$ the console shows the integrating factor $\mu = e^{\int p dx}$, the equation $(\mu y)' = \mu q$, its integral and the solution; several initial values give several curves.



5.1 Sequences

```
sequence(a=(2*n + 1)/(n + 3))
```

terms, $\lim a_n = 2$, increasing, bounded

```
sequence(a=(1 + 1/n)^n, N=40)
```

the limit e

```
sequence(a=factorial(n)/n^n)
```

factorials (through $\log |a_n|$): 0

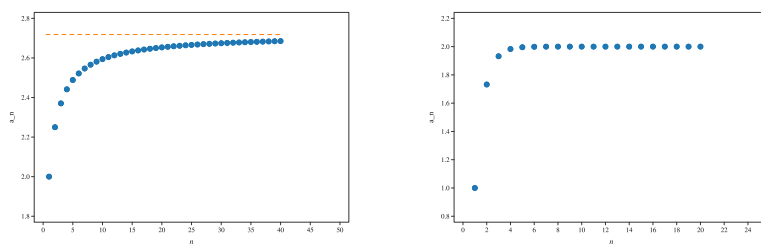
```
sequence(a=(-1)^n)
```

no limit

```
sequence(next=sqrt(2 + a), a1=1)
```

a recursive sequence: $L = g(L)$ gives $L = 2$

`sequence` lists the terms (exactly), draws them as points with the limit (dashed), finds the limit through the function $f(x) = a_x$ (algebra or L'Hôpital's rule) or along whole numbers, and says when the sequence is monotone and bounded. `factorial(n)` is $n!$.



5.2 Infinite Series

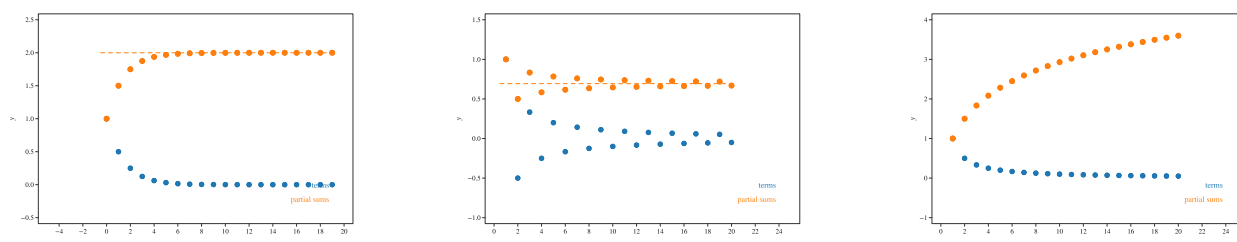
<code>series(a=(1/2)^n, n0=0)</code>	geometric: $a/(1 - r) = 2$
<code>series(a=1/(n*(n + 1)))</code>	telescoping: $1/n - 1/(n + 1)$, the sum 1
<code>series(a=1/(n^2 - 1), n0=2)</code>	telescoping with a gap: $3/4$

`series` draws the terms (blue) and the partial sums S_N (orange) that approach the sum.

5.3–5.6 The Divergence and Integral Tests, Comparison Tests, Alternating Series, Ratio and Root Tests

<code>series(a=n/(n + 1))</code>	divergence test: $a_n \rightarrow 1$
<code>series(a=1/n, test=integral)</code>	the harmonic series diverges
<code>series(a=1/n^2)</code>	a p -series, $p = 2$: converges to $\pi^2/6$
<code>series(a=1/(n*log(n)), n0=2)</code>	integral test: diverges
<code>series(a=n/(n^3 + 1), compare=1/n^2)</code>	comparison
<code>series(a=(-1)^(n + 1)/n)</code>	alternating: converges conditionally, to $\ln 2$
<code>series(a=factorial(n)/n^n)</code>	ratio test: the limit is $1/e < 1$
<code>series(a=(n/(2*n + 1))^n, test=root)</code>	root test: $1/2 < 1$

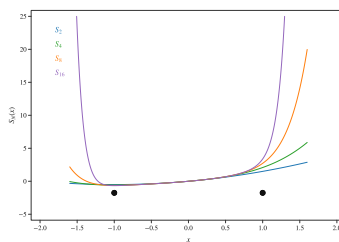
The tests are tried in a sensible order and the reasoning is written out: the divergence test, geometric and telescoping series (with their sums), the alternating series test (with absolute or conditional convergence), the ratio test, comparison with a p -series, the integral test. `test=` shows one test's computation; `compare=` uses a comparison series of your choice. The sums are exact when they are known constants ($\pi^2/6$, $\ln 2$, $\pi^4/90$), else numerical.



6.1–6.2 Power Series and Functions; Properties of Power Series

<code>powerseries(c=1/n, n0=1)</code>	$R = 1$, interval $[-1, 1)$
<code>powerseries(c=(-1)^n/(n*3^n), a=2, n0=1)</code>	centred at 2: $(-1, 5]$
<code>powerseries(c=1/factorial(n))</code>	$R = \infty$
<code>powerseries(c=factorial(n))</code>	$R = 0$
<code>powerseries(term=x^(2*n)/factorial(2*n))</code>	any term in n and x

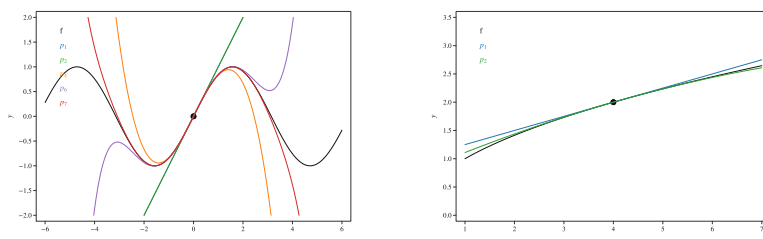
The ratio test gives the radius of convergence; both endpoints are tested as series of numbers, which gives the interval. The partial sums $S_N(x)$ settle down inside the interval and fly apart outside it.



6.3–6.4 Taylor and Maclaurin Series; Working with Taylor Series

<code>taylor(f=exp(x), n=4)</code>	<code>p_4</code> and the series $\sum x^n/n!$
<code>taylor(f=sin(x), n=7, x=-6..6)</code>	polynomials approaching $\sin x$
<code>taylor(f=sqrt(x), a=4, n=2, at=4.1)</code>	Taylor's theorem: the error bound
<code>taylor(f=sin(x^2), n=6)</code>	a series from a known one: $x^{(4n+2)}$
<code>taylor(f=(1+x)^(1/2), n=4)</code>	the binomial series
<code>taylor(f=exp(-x^2), n=8, integrate=0..1)</code>	$\int_0^1 e^{-x^2} dx$ term by term

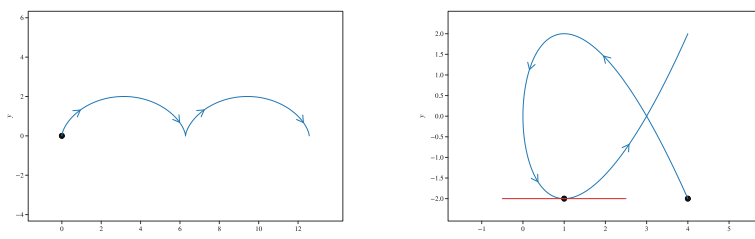
`taylor` gives the Taylor polynomial with its coefficients $f^{(k)}(a)/k!$ and draws it with f . For Maclaurin series built from the basic ones (e^u , $\sin u$, $\cos u$, $1/(1-u)$, $\ln(1+u)$, $\arctan u$, the binomial series) with $u = cx^p$, it gives the general term and the radius. `at=` bounds the error by Taylor's theorem; `integrate=` integrates the series term by term, with the alternating series error bound.



7.1–7.2 Parametric Equations; Calculus of Parametric Curves

<code>parametric(x=t^2 - 3, y=2*t + 1, t=-2..3)</code>	eliminating the parameter
<code>parametric(x=2*cos(t), y=3*sin(t), t=0..2*pi)</code>	an ellipse
<code>parametric(x=t^2, y=t^3 - 3*t, t=-2..2, at=1)</code>	dy/dx , the tangent line
<code>parametric(x=t - sin(t), y=1 - cos(t), t=0..4*pi, length=true)</code>	the cycloid: $L = 16$
<code>parametric(x=t - sin(t), y=1 - cos(t), t=0..2*pi, area=true)</code>	area 3π

Arrows show the orientation. The console gives $dy/dx = y'(t)/x'(t)$, d^2y/dx^2 , the horizontal and vertical tangents and the cusps, and with `length=` or `area=` the arc length and the area.

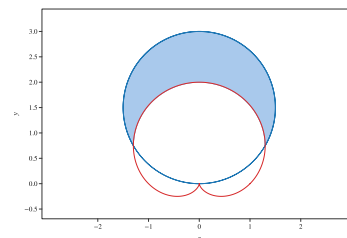
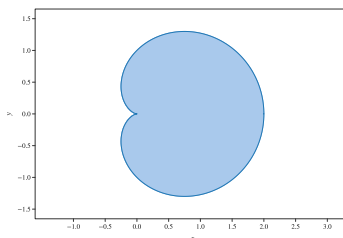
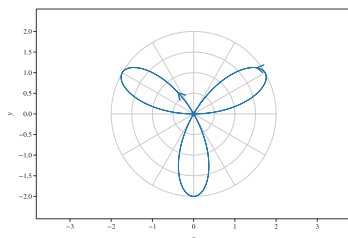


7.3–7.4 Polar Coordinates; Area and Arc Length in Polar Coordinates

```

coords(x=1, y=-1)
polar(r=2*sin(3*theta))
/3
polar(r=1 + 2*cos(theta), at=pi/3)
polar(r=1 + sin(theta), length=true)
polararea(r=1 + cos(theta))
polararea(r=sin(3*theta))
polararea(r=3*sin(theta), g=1 + sin(theta))
(sqrt(2), 7*pi/4)
a rose: symmetry, r = 0 at multiples of pi
the slope at a point
a cardioid's length: 8
(1/2) int r^2 dtheta = 3*pi/2
traced once for 0..pi: pi/4
between two curves: pi

```



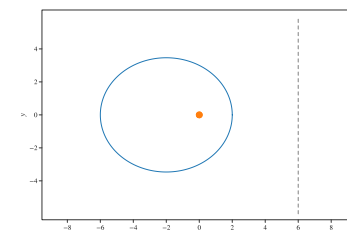
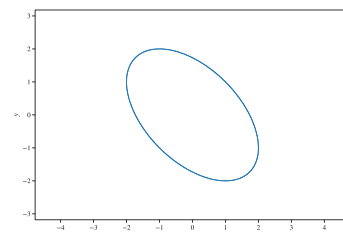
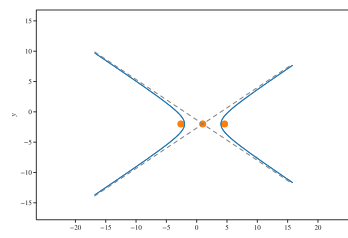
7.5 Conic Sections

```

conic(x^2/9 + y^2/4 = 1)
conic(4*x^2 - 9*y^2 - 8*x - 36*y - 68 = 0)
its asymptotes
conic(y^2 = 8*x)
conic(x^2 + x*y + y^2 = 3)
conic(e=1/2, p=6)

```

an ellipse: foci, eccentricity
 completing the squares: a hyperbola and
 a parabola: focus, directrix
 rotation of axes: $\theta = \pi/4$
 polar form: $r = e p / (1 + e \cos \theta)$



Practice and checking your answers

```

practice(substitution)
practice(definite, n=8, seed=3)
practice(riemann, answers=true)
check(antiderivative, f=x*cos(x^2), answer=sin(x^2))
Not quite ... Hint: the derivative of your answer is 2 times f: a constant factor is
off ...
check(definite, f=2*x*(x^2 + 1)^2, x=0..1, answer=1/3)
Not quite: int = 7/3. Hint: after the substitution u = x^2 + 1 the limits must change
too ...
check(area, f=x^3 - x, x=-1..2, answer=11/4)
check(average, f=x^2, x=0..3, answer=3)
check(sigma, f=k^2, k=1..10, answer=385)
check(riemann, f=x^2, x=0..2, n=4, method=mid, answer=21/8)
check(ftc, f=sin(t^2), lower=0, upper=x^2, answer=2*x*sin(x^4))
check(between, f=sin(x), g=cos(x), x=0..pi, answer=2)
Not quite: the area is 2*sqrt(2). Hint: the curves cross: split the interval ...
check(volume, f=x, g=x^2, x=0..1, about=x, answer=pi/30)

```

```

Not quite: V = 2*pi/15 (washer method). Hint: pi (R^2 - r^2), not pi (R - r)^2.
check(arclength, f=x^(3/2), x=0..1, answer=1.4397)    decimals are right to the digits
given
check(antiderivative, f=x*exp(x), answer=x*exp(x) - exp(x))
check(improper, f=1/x, x=1..inf, answer=diverges)
check(ode, f=x*y, x0=0, y0=2, answer=exp(x^2/2))
Not quite: your function solves the equation but y(0) = 1, not 2: find the constant
...
check(series, a=(1/2)^n, n0=1, answer=2)
Not quite: the sum is 1. Hint: a/(1 - r) with a the FIRST term (n = 1): a = 0.5.
check(series, a=1/n, answer=converges)
check(seqlimit, a=(1 + 1/n)^n, answer=e)
check(radius, c=1/(n*2^n), n0=1, answer=2)
check(taylorpoly, f=sqrt(x), a=4, n=2, answer=2 + (x - 4)/4 - (x - 4)^2/64)
check(polararea, r=2*cos(theta), theta=0..pi, answer=pi)

```

Topics: `sigma`, `riemann`, `definite`, `ftc`, `average`, `basic`, `netchange`, `substitution`, `expint`, `invtrigint`, `mixedint`; for Chapter 2 `between`, `volume`, `arclength`, `work`, `centroid`, `growth`; for Chapter 3 `parts`, `trigint`, `trigsub`, `partialfractions`, `numint`, `improper`; for Chapter 4 `separable`, `linearde`, `eulermethod`; for Chapter 5 `sequences`, `series`, `convergence`; for Chapter 6 `radius`, `maclaurin`, `taylorpoly`; for Chapter 7 `paramslope`, `polararea`, `conics`. `practice` and `check` are shared with the Differential Calculus module: with both imported, each answers its own topics. `check` accepts any equivalent form of an answer and names the common mistakes: a wrong sign, a missing constant factor in a substitution, $F(a) - F(b)$ instead of $F(b) - F(a)$, forgetting $F(a)$, forgetting to change the limits after a substitution, the total area instead of the net area, a missing chain-rule factor in a derivative of an integral, the wrong kind of Riemann sum, an area that was not split where the curves cross, $\pi(R - r)^2$ for $\pi(R^2 - r^2)$, a missing factor π or 2, $\sqrt{1 + f'}$ for $\sqrt{1 + f'^2}$.

How the results are checked

Every antiderivative found by the engine is differentiated and compared with the integrand at many points before it is used. Every definite integral computed by the Fundamental Theorem is compared with an independent adaptive Gauss–Kronrod quadrature, and the theorem is not applied when the function is unbounded inside the interval. The tests check the quadrature on smooth, endpoint-singular and rapidly decaying integrands, exact definite integrals (9 , 2 , $\pi/4$, $9\pi/2$, $e - 1$), improper cases (detected, convergent or divergent), total areas, all six kinds of Riemann sums, the sum formulas and geometric sums, the derivative of integrals with variable limits, that every practice problem has an answer, every hint of `check`, and the console output of every command. For Chapter 2 they check areas between curves (split and sideways), volumes by slices, disks, washers and shells about the axes and about other lines (the two methods agree with each other and with the theorem of Pappus), the meshes of the solids, arc lengths with perfect squares and hyperbolic and trigonometric identities, surface areas, mass, work, pumping, fluid force, centroids and exponential models. For Chapter 3 they check integration by parts (repeated, tabular and cyclic), the trigonometric integrals, the three trigonometric substitutions, partial fractions with repeated and quadratic factors, the numerical rules and their error bounds, and improper integrals that converge and diverge. For Chapter 4 they check separable and linear equations solved exactly (and verified by substitution), equilibria and their stability, Euler’s method, the logistic solution and `verify`. For Chapters 5–7 they check limits of sequences (with factorials and recursive sequences), the sums of geometric, telescoping, p - and alternating series, each convergence test, radii and intervals of convergence with their endpoints, Taylor polynomials, series built from the known ones, Taylor’s error bound, term-by-term integration, parametric slopes, lengths and areas, polar symmetry, slopes, areas and lengths, and conics in every form. The engine’s own tests cover 103 antiderivatives,

including completing the square, rational functions with a quadratic denominator and the inverse trigonometric forms, and the steps it reports.