

Lesson 9: Basins of attraction

Since the Plot component turned out not to be completely reliable (at least on my computer), here's another attempt using an animation that should be more reliable.

```
> restart;
with(plots):
stair:= x -> ([x,x],[x,g(x)]);
staircase:= proc(x0, a, b, n)
  local x, count, Curves, Staircase;
  uses plots;
  Curves:= plot([x,g(x)],x=a..b, colour=[red,blue]);
  x[0]:= x0;
  for count from 1 to n-1 do
    x[count]:= g(x[count-1])
  end do;
  Staircase:= plot([seq(stair(x[j]),j=0..n-1)],colour=black);
  display([Curves, Staircase,
    plot([[x0,x0]],style=point,colour=black)]);
end proc;
```

$$\text{stair} := x \rightarrow ([x, x], [x, g(x)])$$

```
staircase := proc(x0, a, b, n)
```

```
  local x, count, Curves, Staircase;
```

```
  Curves := plot([x, g(x)], x = a .. b, colour = [red, blue]);
```

```
  x[0] := x0;
```

```
  for count to n - 1 do x[count] := g(x[count - 1]) end do;
```

```
  Staircase := plot([seq(stair(x[j]), j = 0 .. n - 1)], colour = black);
```

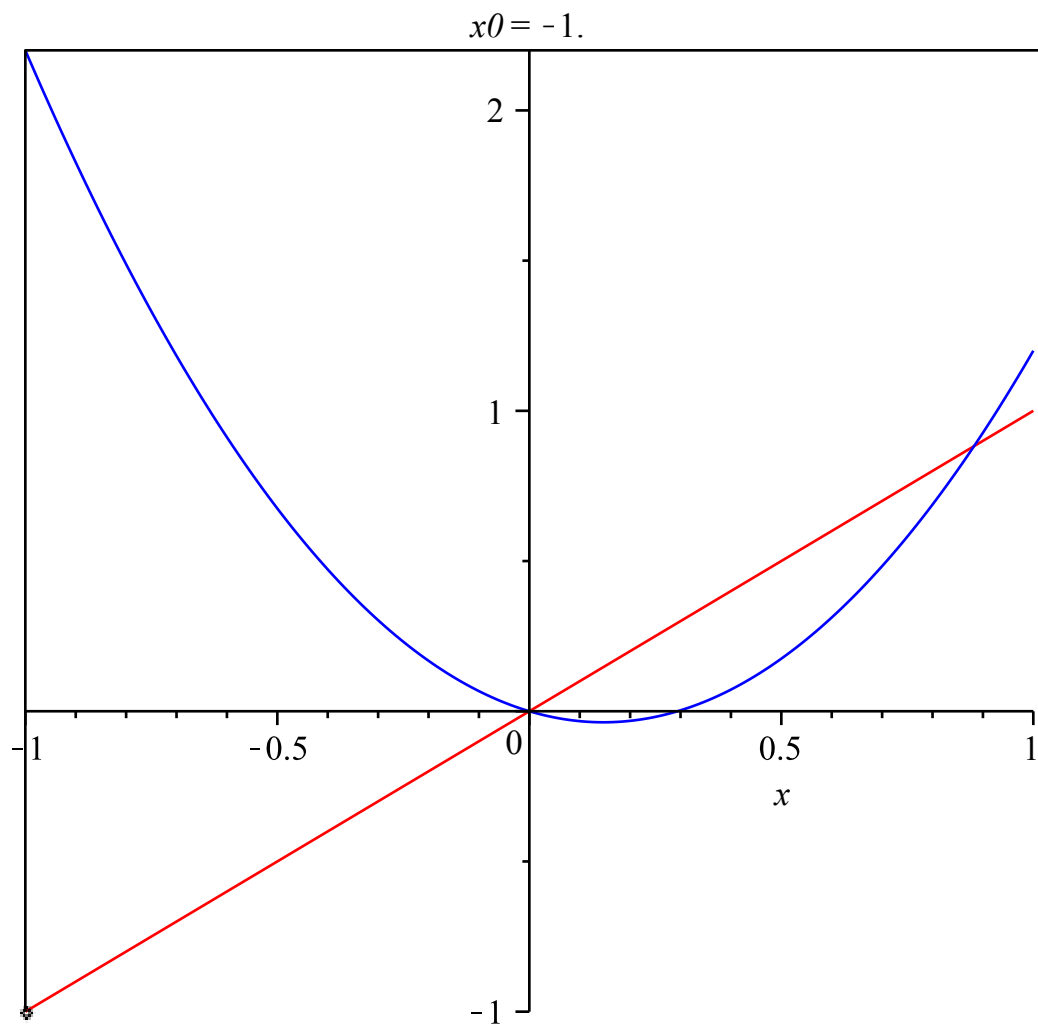
```
  plots:-display([Curves, Staircase, plot([[x0, x0]], style = point, colour = black)])
```

```
end proc
```

```
> g:= x -> 1.7*x^2 - x/2;
animate(staircase, [x0,-1,1,3],x0=-1..1,frames=100,view=[-1..1,
-1..g(-1)]);
```

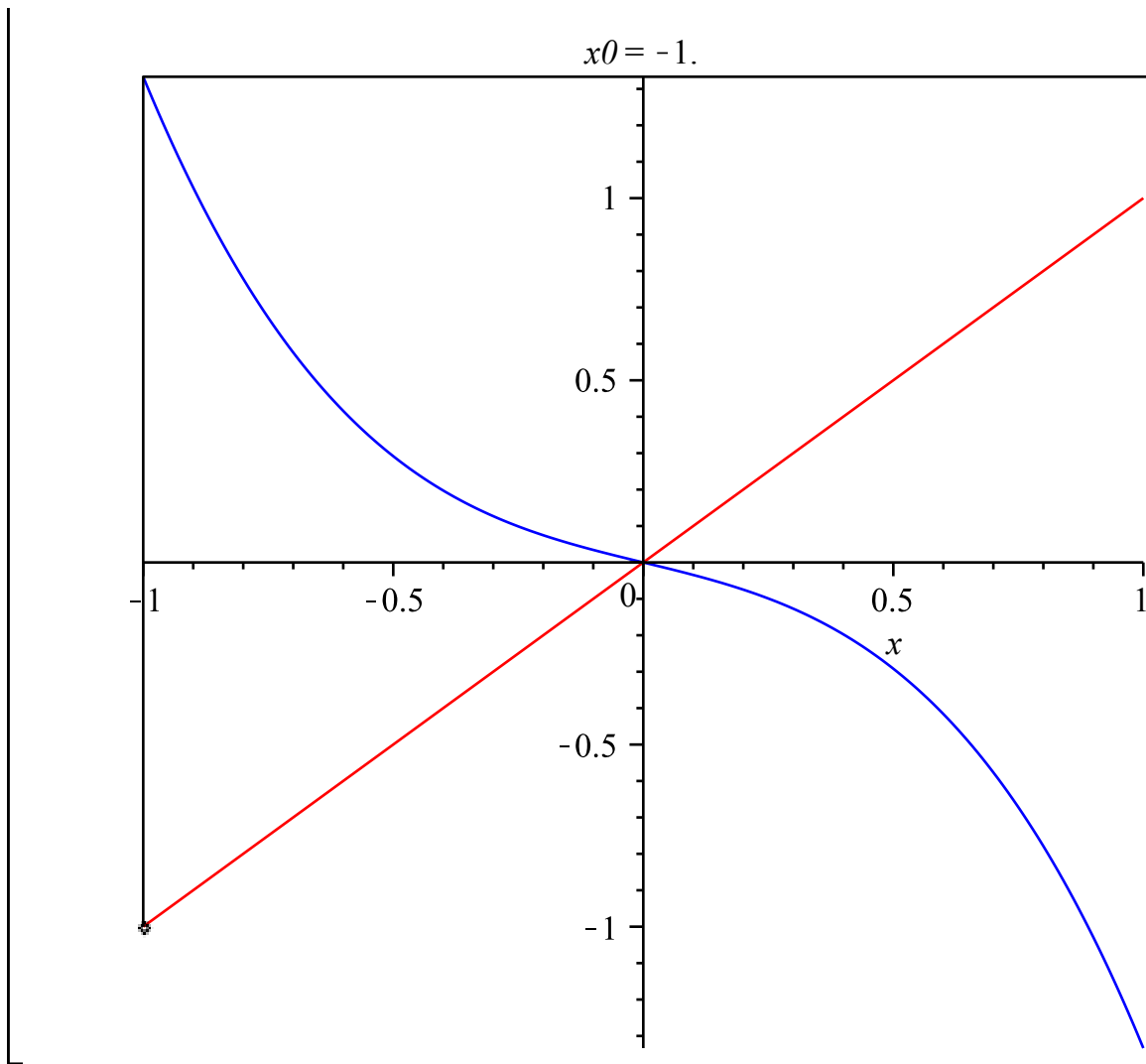
$$g := x \rightarrow 1.7x^2 - \frac{1}{2}x$$

(1)



```
> g:= x -> - x/3 - x^3;
animate(staircase, [x0,-1,1,3],x0=-1..1,frames=100,view=[-1..1,g
(-1)..g(1)]);
```

$$g:=x \rightarrow -\frac{1}{3}x - x^3$$



▼ Possibilities for the immediate basin of attraction

Back to theory:

Given a continuous function $g(x)$ with attracting fixed point p , how could we find the immediate basin of attraction (a, b) ?

What can $g(a)$ be? Only a or b . Similarly for $g(b)$. So there are four possibilities if a and b are finite:

- 1) $g(a) = a, g(b) = b$: a and b are fixed points (and can't be attractors).
- 2) $g(a) = a, g(b) = a$: a is a fixed point (not an attractor), b is a point mapped to a .
- 3) $g(a) = b, g(b) = b$: b is a fixed point (not an attractor), a is a point mapped to b .
- 4) $g(a) = b, g(b) = a$: a and b form a 2-cycle (not an attractor).

The possibilities with a or b infinite are:

- 5) $a = -\infty, b = \infty$: basin is the whole real line
- 6) $a = -\infty, g(b) = b$: b is a fixed point (not an attractor)
- 7) $g(a) = a, b = \infty$: a is a fixed point (not an attractor)

Another fixed point, or a point mapped to another fixed point, or a 2-cycle, can't be in the immediate

basin of attraction. So here's a way to find the immediate basin of attraction of an attracting fixed point p :

- 1) Find non-attracting fixed points, points mapped to those fixed points, and 2-cycles
- 2) The closest of these to p on either side of p (if any) are the boundary points of the immediate basin of attraction. If there are none on one side, the basin extends to infinity on that side.

For example: find the immediate basin of attraction of 0 for $g(x) = \frac{1}{2}x - x^3 + x^4$.

```
> g:= x -> x/2 - x^3 + x^4 ;
```

$$g := x \rightarrow \frac{1}{2}x - x^3 + x^4 \quad (1.1)$$

```
> fps:= fsolve(g(x)=x);
```

$$fps := 0., 1.297156508 \quad (1.2)$$

```
> fsolve(g(g(x))=x);
```

$$0., 1.297156508 \quad (1.3)$$

No two-cycles here, just another fixed point (which can't be an attractor).

```
> b:= fps[2];
fsolve(g(x)=b);
```

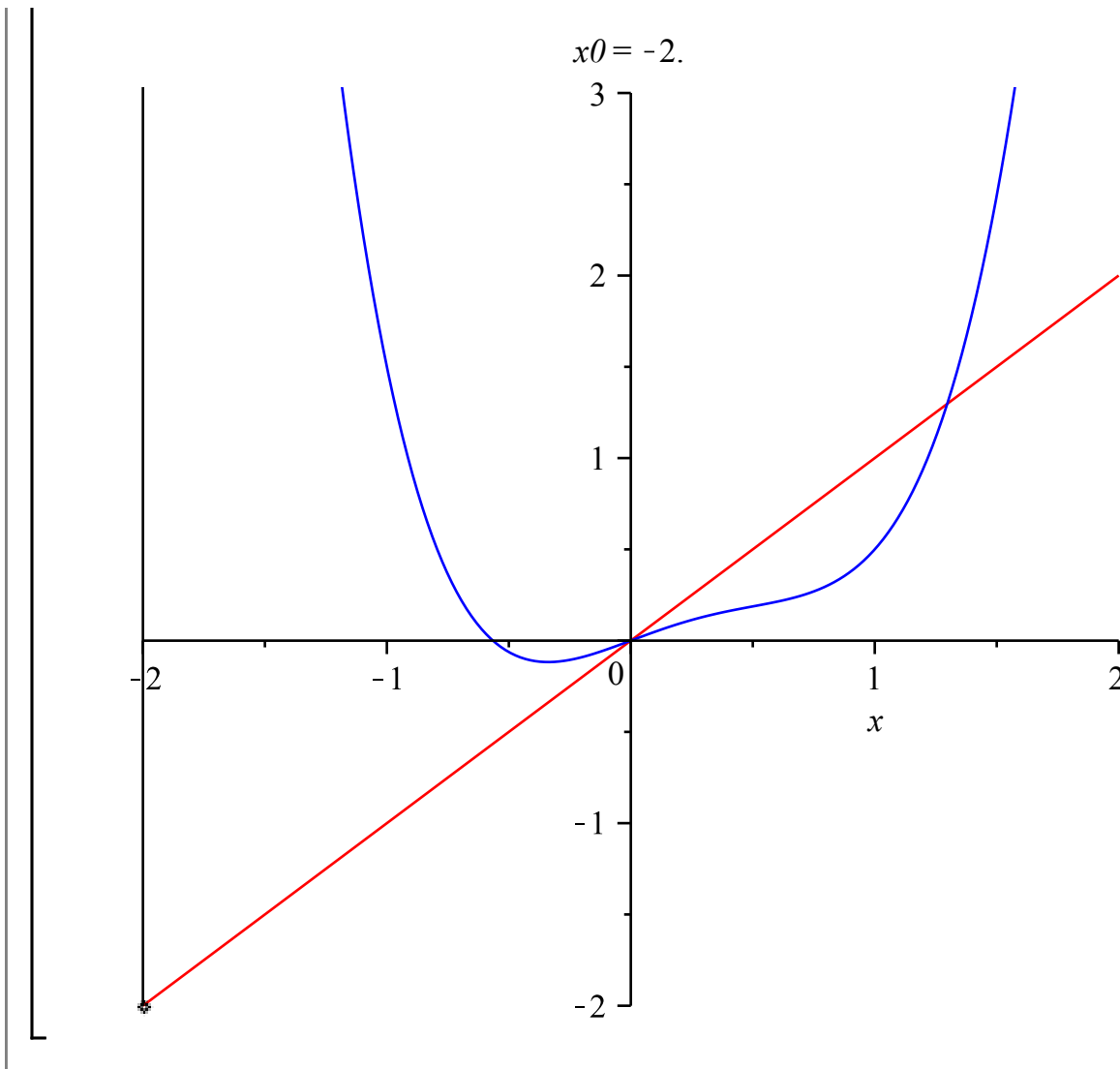
$$b := 1.297156508$$
$$-0.9673432857, 1.297156508 \quad (1.4)$$

```
> a:= %[1];
```

$$a := -0.9673432857 \quad (1.5)$$

I'll make an animation for this one, with x going from -2 to 2 and y from -2 to 3.

```
> animate(staircase,[x0,-2,2,3],x0=-2..2,frames=100,view=[-2.
.2,-2..3]);
```



An immediate basin for Newton's method

Consider Newton's method, say for this function from Lessons 5 and 6:

```
> f := x -> sin(x) - x/Pi;
```

$$f := x \rightarrow \sin(x) - \frac{x}{\pi} \quad (2.1)$$

We defined Newton's method as the iteration of the function **newt**:

```
> newt := x -> x - f(x)/D(f)(x);
g := newt;
```

$$\begin{aligned} \text{newt} &:= x \rightarrow x - \frac{f(x)}{D(f)(x)} \\ g &:= \text{newt} \end{aligned} \quad (2.2)$$

The solutions of $f(x) = 0$ are the fixed points of **newt**. In this case there were three of them, approximately -2.313734132, 0 and 2.313734132. These are all attractors for **newt**. But we saw if we started at a point that wasn't close to one of these, we could wander all over the place before eventually settling down and approaching a fixed point (and it's not at all clear that we ever would

settle down). Now I want to find the immediate basin of attraction of one of the fixed points, say 2.313734132.

A few things to notice:

- If there's more than one solution to $f(x) = 0$, the basin won't be the whole real line.
- There's also the potential complication that **newt** is not continuous: it's not defined at points where $f'(x) = 0$, and such points can exist. As x approaches such a point (which is not a solution of $f(x) = 0$) from one side or the other, $\text{newt}(x)$ will go to ∞ or $-\infty$.
- The situation where an endpoint was a repelling fixed point, or mapped into a repelling fixed point, won't happen here: every fixed point of **newt** is a solution of $f(x) = 0$, and those are attracting fixed points.
- The other situation, where the endpoints form a repelling 2-cycle, can happen.

Thus the possibilities for the immediate basin of attraction of a solution p are as follows:

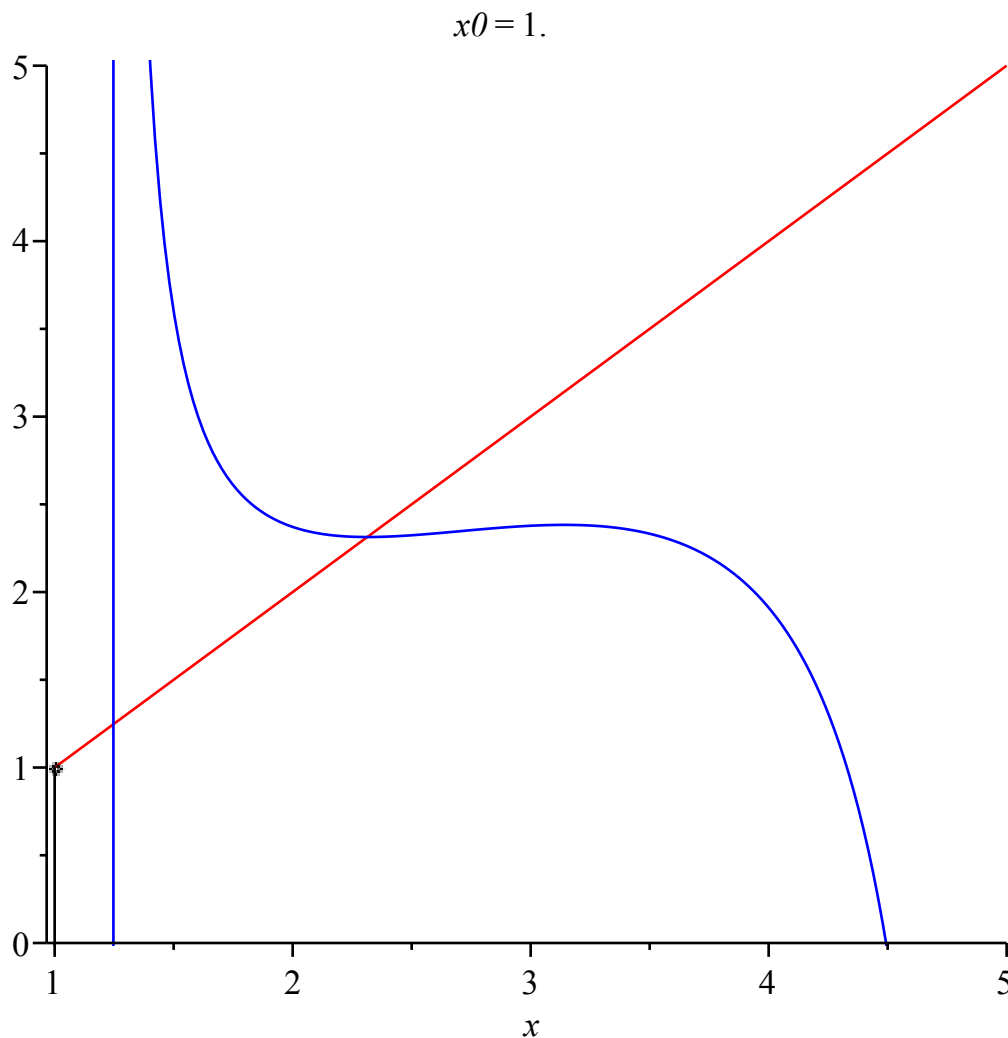
1) $a = -\infty, b = \infty$ (if p is the only real solution to $f(x) = 0$)

2) $a = -\infty, f'(b) = 0$, where $\lim_{x \rightarrow b^-} \text{newt}(x) = -\infty$

3) $b = \infty, f'(a) = 0$, where $\lim_{x \rightarrow a^+} \text{newt}(x) = \infty$

4) A 2-cycle: $\text{newt}(a) = b, \text{newt}(b) = a$.

```
> animate(staircase, [x0, 1, 5, 3], x0=1..5, frames=100, view=[1..5, 0.  
.5]);
```



We see that the endpoints of the immediate basin should be a 2-cycle, with one point approximately 1.4, the other about 4.2.

```
> a:= fsolve(newt(newt(x))=x,x=1.3 .. 1.5);
      a := 1.444746368
```

(2.3)

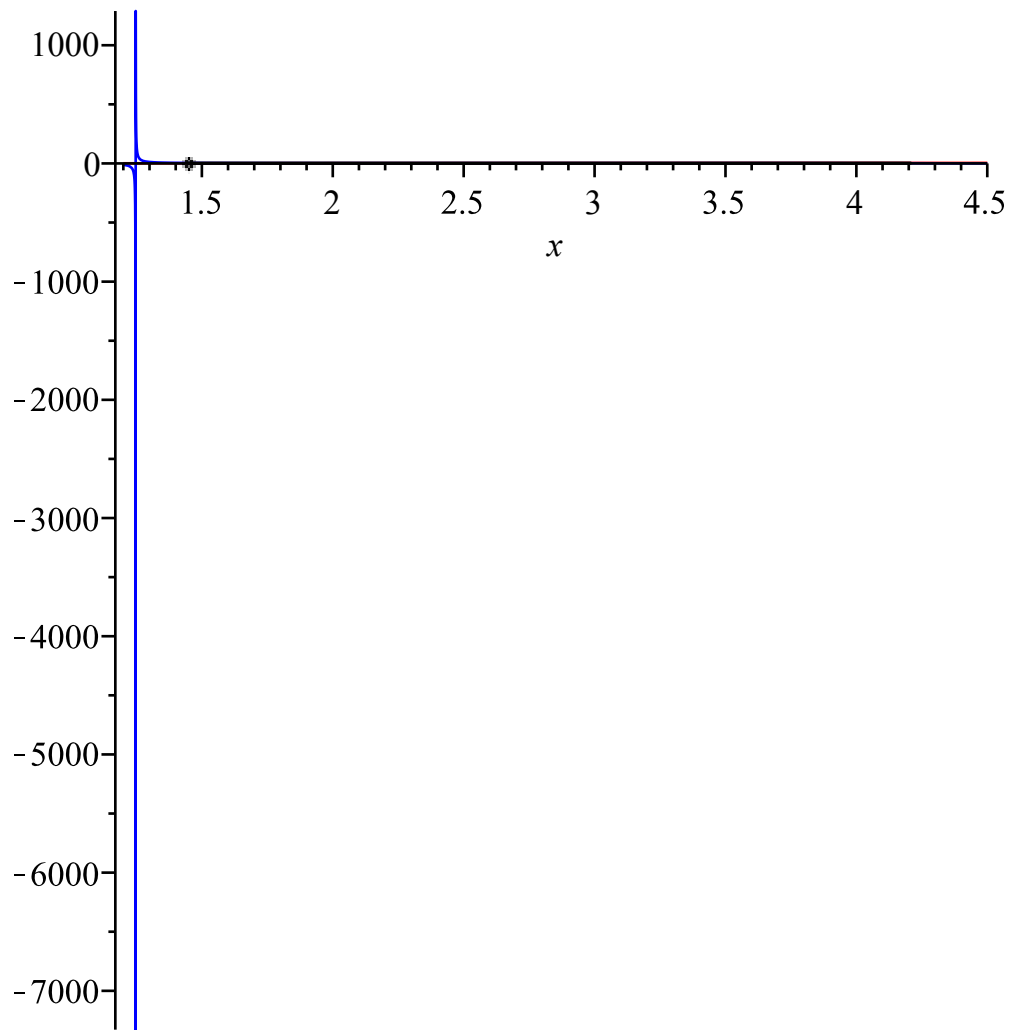
```
> b:= fsolve(newt(newt(x))=x, x=4.1 .. 4.3);
      b := 4.208023889
```

(2.4)

```
> evalf(newt(a) = b);
      evalf(newt(b) = a);
      4.208023891 = 4.208023889
      1.444746369 = 1.444746368
```

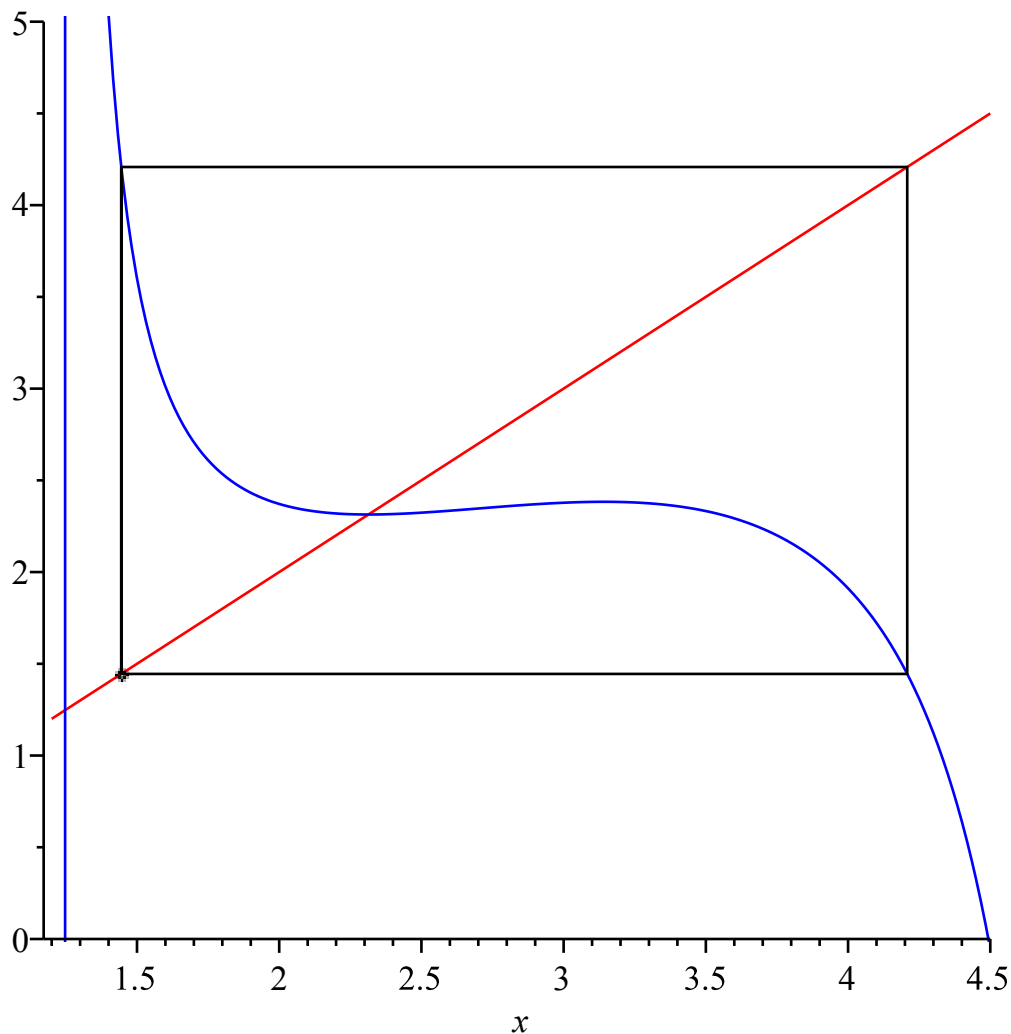
(2.5)

```
> staircase(a, 1.2, 4.5, 3);
```



I'd better use the **view** option: the graph of newt(x) has a vertical asymptote.

```
> display(%,view=[1.2 .. 4.5, 0..5]);
```



When Newton is slow

We saw that when the equation $f(x) = 0$ has a solution p with $f'(p) \neq 0$, Newton's method converges very quickly (for an appropriate starting point), because $\text{newt}'(p) = 0$. When $f'(p) = 0$, on the other hand, convergence is slower, as you will see in an example in the second homework. Let's investigate what happens in such a case.

Suppose $f(p) = 0$ and $f'(p) = 0$. For most of the functions we meet (analytic functions), unless $f(x) = 0$ in an interval around p , there will be some derivative of f that is not 0 at p . Typically we can write $f(x) = (x - p)^m h(x)$ where $m \geq 2$ and $h(p) \neq 0$, and then all derivatives up to the $(m-1)$ 'th will be 0, but $D^{(m)}(f)(p) \neq 0$. What would happen to Newton's method? For convenience I'll take $p = 0$.

```
> f := x -> x^m * h(x);
```

$$f := x \rightarrow x^m h(x)$$

(3.1)

```
> newt := x -> x - f(x)/D(f)(x);
newt(x);
```

$$\text{newt} := x \rightarrow x - \frac{f(x)}{D(f)(x)}$$

$$x - \frac{x^m h(x)}{\frac{x^m m h(x)}{x} + x^m D(h)(x)} \quad (3.2)$$

> `D(newt)(0);`

Error, (in unknown) numeric exception: division by zero

I should have expected the error: we are asking Maple to divide by 0.

> `D(newt)(x);`

$$\frac{x^m h(x) \left(\frac{x^m m^2 h(x)}{x^2} - \frac{x^m m h(x)}{x^2} + \frac{2 x^m m D(h)(x)}{x} + x^m D^{(2)}(h)(x) \right)}{\left(\frac{x^m m h(x)}{x} + x^m D(h)(x) \right)^2} \quad (3.3)$$

However, this can be simplified. The **normal** command does ordinary algebraic simplification, such as putting everything over a common denominator and cancelling common factors from numerator and denominator.

> `normal(%);`

$$\frac{h(x) (m^2 h(x) - m h(x) + 2 m D(h)(x) x + D^{(2)}(h)(x) x^2)}{(m h(x) + D(h)(x) x)^2} \quad (3.4)$$

After this simplification we shouldn't have troubles with 0 in the denominator.

> `eval(%,x=0);`

$$\frac{m^2 h(0) - m h(0)}{h(0) m^2} \quad (3.5)$$

But this can be simplified some more.

> `normal(%);`

$$\frac{m-1}{m} \quad (3.6)$$

Since $m \geq 2$, this is greater than 0 but less than 1. That means that the solution p is an attractor for Newton's method, but the convergence is only linear, not quadratic: i.e. $|x_{n+1} - p|$ is approximately

$\frac{(m-1)}{m} |x_n - p|$ rather than a constant times $|x_n - p|^2$. Moreover, if m is large the factor $\frac{(m-1)}{m}$ is close to 1, so convergence will be rather slow.

Some on-line references for discrete dynamical systems:

Notes by Sebastien Guenneau, University of Liverpool

<<http://www.maths.liv.ac.uk/~guenneau/1Ddiscrete.pdf>>

Notes by Pei Yu, University of Western Ontario

<<http://pyu1.apmaths.uwo.ca/~pyu/pub/courses/am303b/notes/ch4.ps>>

First 22 pages of A First Course in Dynamics by Boris Hasselblatt and Anatole Katok

<<http://assets.cambridge.org/052158/3047/sample/0521583047ws.pdf>>

Polynomials

Underneath all the bells and whistles, much of what Maple does is concerned with polynomials and rational functions, so I think it's a good idea to take a good look at those, both from a theoretical point of view and from the Maple point of view.

We'll start with polynomials. A polynomial in one variable (or **indeterminate**) x is an expression of

the form $\sum_{j=0}^n a_j x^j = a_0 + a_1 x + \dots + a_{n-1} x^{n-1} + a_n x^n$, where n is a nonnegative integer. The a_j are called **coefficients**. When writing it we generally leave out terms with coefficient 0. The term with the highest power of x and a nonzero coefficient is called the **leading term**: if this is $a_n x^n$, its coefficient a_n is the **leading coefficient**, and the exponent n is the **degree** of the polynomial. Here's a typical polynomial in Maple:

```
> P := 3 + x^2 + 2*x^3;
```

$$P := 3 + x^2 + 2x^3 \quad (4.1)$$

```
> degree(P);
```

$$3 \quad (4.2)$$

The coefficients in this case are integers, but they don't have to be. Usually they will be numbers of some sort, but sometimes, as we'll see, they can be more complicated things, which might depend on other variables. The most basic requirement is that you can do addition and multiplication and division in the usual way with the coefficients: technically, they are members of a **field**. If there's more than one name in the polynomial, and you want the degree, you'd need to tell **degree** what you're considering as the variable for this purpose.

```
> degree(x^2 + y*x, x);
```

$$2 \quad (4.3)$$

Constants are polynomials too: if nonzero, they have degree 0.

```
> degree(2);
```

$$0 \quad (4.4)$$

The polynomial 0 is rather special (it has no nonzero coefficients, so no leading term). We consider it to have degree $-\infty$.

```
> degree(0);
```

$$-\infty \quad (4.5)$$

In high school you learned (I hope) how to do the usual algebraic operations, addition, subtraction, multiplication, division on polynomials. It's no surprise that Maple can do these too.

```
> Q := P + (3 - x);
```

$$Q := 6 + x^2 + 2x^3 - x \quad (4.6)$$

Notice that Maple doesn't care about the order of the terms of this polynomial. If you do care, you can ask Maple to **sort** the terms in either ascending or descending order of the powers of x . The default is descending.

```
> sort(Q);
```

$$2x^3 + x^2 - x + 6 \quad (4.7)$$

```
> sort(Q, x, ascending);
```

$$6 - x + x^2 + 2x^3 \quad (4.8)$$

The **sort** command is unusual in that it actually changes its polynomial input, rather than just

returning a value.

```
> Q;
```

$$6 - x + x^2 + 2x^3 \quad (4.9)$$

Next, let's multiply.

```
> P*Q;
```

$$(3 + x^2 + 2x^3)(6 - x + x^2 + 2x^3) \quad (4.10)$$

Notice that Maple didn't automatically expand out the product. For some purposes you might want to keep a polynomial in a factored form, rather than expanded. Maple lets you have it whichever way you want. The **expand** command will expand it out.

```
> expand(P*Q);
```

$$18 - 3x + 9x^2 + 17x^3 - x^4 + 4x^5 + 4x^6 \quad (4.11)$$

Similarly, with a power:

```
> P^3;
```

$$(3 + x^2 + 2x^3)^3 \quad (4.12)$$

```
> expand(%);
```

$$27 + 27x^2 + 54x^3 + 9x^4 + 36x^5 + 37x^6 + 6x^7 + 12x^8 + 8x^9 \quad (4.13)$$

For division:

```
> %/P;
```

$$\frac{27 + 27x^2 + 54x^3 + 9x^4 + 36x^5 + 37x^6 + 6x^7 + 12x^8 + 8x^9}{3 + x^2 + 2x^3} \quad (4.14)$$

It should be possible to simplify this, since the denominator P should be a factor of the numerator. Maple doesn't try to do that unless you tell it to. The **normal** command will cancel common factors from numerator and denominator.

```
> normal(%);
```

$$9 + 6x^2 + 12x^3 + x^4 + 4x^5 + 4x^6 \quad (4.15)$$

If the numerator isn't divisible by the denominator, the result of division, even after using **normal**, will be a rational function, that is a fraction whose numerator and denominator are polynomials.

```
> normal(P/(x^2+1));
```

$$\frac{3 + x^2 + 2x^3}{x^2 + 1} \quad (4.16)$$

There's another kind of division of polynomials: division with remainder. This is done in Maple by the commands **quo** (for quotient) and **rem** (for remainder). You have to tell these the two polynomials and the name of the variable.

```
> Q:= quo(P,x^2+1,x);
```

$$Q := 2x + 1 \quad (4.17)$$

```
> R:= rem(P,x^2+1,x);
```

$$R := 2 - 2x \quad (4.18)$$

Here's what these do. Given any two polynomials A and B , where B is not the 0 polynomial, $Q = \mathbf{quo}(A,B,x)$ and $R = \mathbf{rem}(A,B,x)$ are polynomials such that $A = BQ + R$ and the degree of R is less than the degree of B .

```
> P = expand((x^2+1)*Q + R);
```

$$3 + x^2 + 2x^3 = 3 + x^2 + 2x^3 \quad (4.19)$$

Factoring is another important operation on polynomials. The **Fundamental Theorem of Algebra** says that a polynomial of degree n in one variable x (with coefficients that are complex numbers) can be written as

$a_n (x - r_1) (x - r_2) \dots (x - r_n) = a_n \prod_{j=1}^n (x - r_j)$ where the complex numbers r_1, \dots, r_n are the roots of the polynomial. Some roots may be present more than once: these are called **repeated roots**. The number of times a root appears is called its **multiplicity**.

Maple has a **factor** command to factor polynomials, but usually it doesn't do this kind of complete factorization. If the coefficients of the polynomial are rational numbers, it will just find factors with coefficients that are rational numbers. For example:

```
> P:= 4*x^4 -4*x^2 +12*x - 9;
   factor(P);
```

$$P := 4x^4 - 4x^2 + 12x - 9$$

$$(2x^2 + 2x - 3)(2x^2 - 2x + 3) \quad (4.20)$$

To factor completely into linear factors, you can say

```
> factor(P, complex);
```

$$4. (x + 1.82287565553230) (x - 0.500000000000000 + 1.11803398874989 I) (x - 0.500000000000000 - 1.11803398874989 I) (x - 0.822875655532295) \quad (4.21)$$

Notice that the roots r_j are given as (approximate) floating-point numbers. You could also ask for a factorization over the real numbers:

```
> factor(P, real);
```

$$4. (x + 1.82287565553230) (x - 0.822875655532295) (x^2 - 1. x + 1.50000000000000) \quad (4.22)$$

This gives you factors where all the coefficients are real: they may be linear factors, or quadratics whose roots are not real.

Maple objects introduced in this lesson

normal
 degree
 sort
 ascending (option to sort)
 expand
 quo
 rem
 factor