# Steffensen's Method

# Acceleration Methods

Can we speed up the convergence of the fixed point method?

**Idea:** To solve $f(x)=0$, find a clever choice of $g(x)$ that converges quickly. So far, we have seen:

① Algebraically manipulate $f(x)$ to find a $g(x)$ so that

$$\bar{x} = g(\bar{x}) \iff f(\bar{x}) = 0$$

**Example**

$$f(x) = x^3 + x^2 - 3x + 3 = 0$$

$$g(x) = \sqrt[3]{-x^2 + 3x - 3}$$

Choose $g(x)$ so that $|g'(\bar{x})| \ll 1$

② Newton's method

The choice of

$g'(\bar{x}) = 0$ $$g(x) = x - \frac{f(x)}{f'(x)}$$

led to a quadricly convergent method.
but requires a derivative of $f(x)$.

③ Can we get quadratic convergence
without requiring a derivative
$f'(x)$? Yes

Steffensen's method

Idea is to find an iteration that
"extrapolates the error".

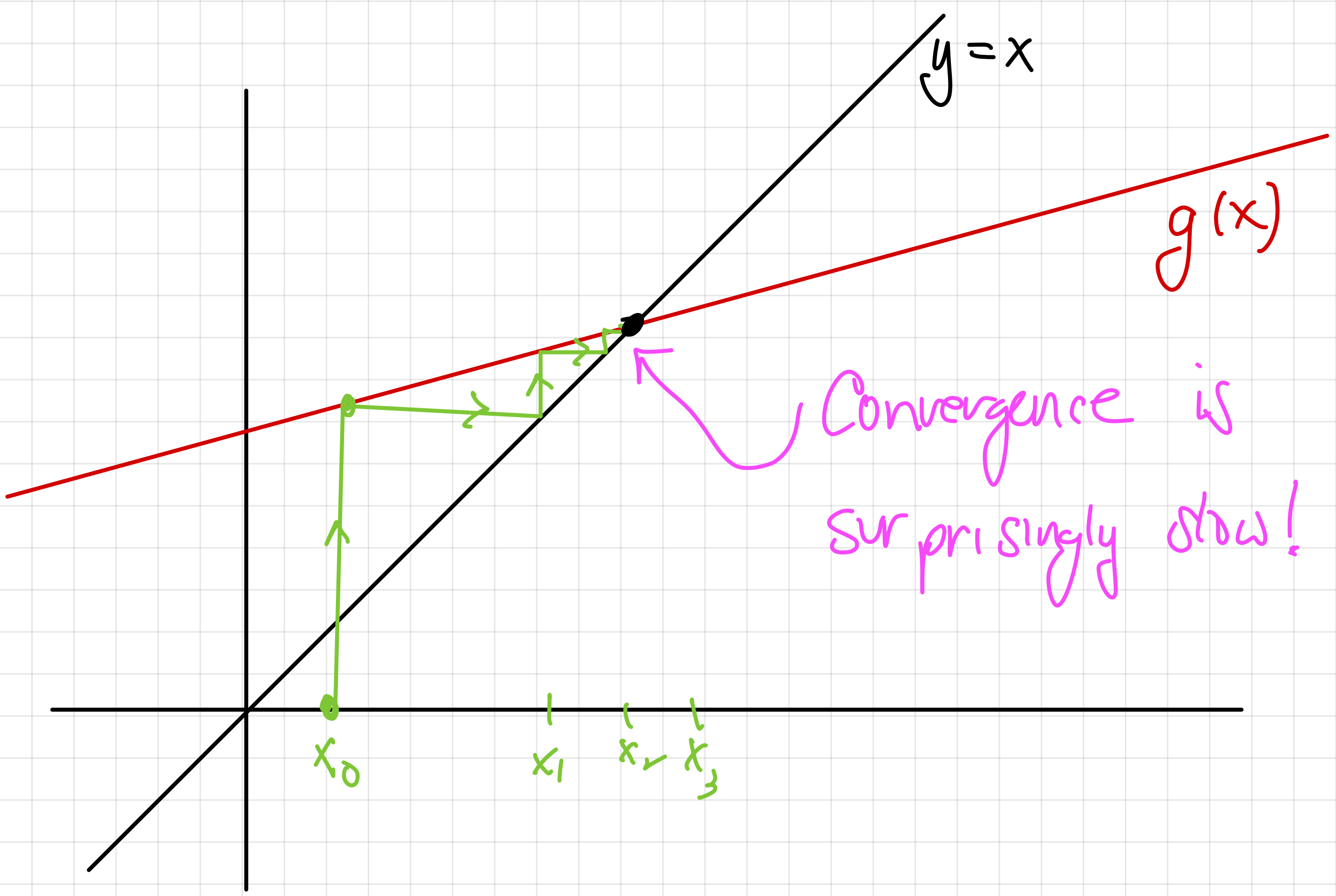
# Steffensen's Acceleration Method

Can we improve the order of convergence of the fixed point method?

**Idea:** Consider the linear equation

$$g(x) = ax + b$$

If we apply fixed point iteration to this $g(x)$, we get a sequence of iterates that converges quite slowly:



## Example:

$$g(x) = 0.1x + 1$$

Fixed point iteration:

$$x_{k+1} = g(x_k)\ ; \quad x_0 = 0$$

$x_k$ $\qquad\qquad$ $|x_{k+1} - x_k|$

```
Fixed point iteration
     1 1.000000000000000e+00   1.0000e+00
     2 1.100000000000001e+00   1.1111e-02
     3 1.110000000000001e+00   1.1111e-03
     4 1.111000000000000e+00   1.1111e-04
     5 1.111100000000000e+00   1.1111e-05
     6 1.111110000000000e+00   1.1111e-06
     7 1.111111000000000e+00   1.1111e-07
     8 1.111111100000000e+00   1.1111e-08
     9 1.111111110000000e+00   1.1111e-09
    10 1.111111111000000e+00   1.1111e-10
    11 1.111111111100000e+00   1.1111e-11
    12 1.111111111110000e+00   1.1111e-12
    13 1.111111111110999e+00   1.1108e-13
    14 1.111111111111001e+00   1.1146e-14
```

linear convergence

Solution: $\quad g(x) = 0.1x + 1 = x$

$$\rightarrow \quad x = 10/9 = 1.\overline{1111}$$

The strategy is to find a method that converges in one iteration for the problem $g(x) = ax + b$.

The idea is to extrapolate the error to find an improved iterate

Given $x_0$, define

$$e_0 = g(x_0) - x_0$$

Setting $\tilde{x}_1 = g(x_0)$, we also have

$$\tilde{e}_1 = g(\tilde{x}_1) - \tilde{x}_1$$

We now have two data points:

$$(x_0, e_0) \text{ and } (\tilde{x}_1, \tilde{e}_1)$$

What choice of $x_1$ would produce a zero error? Since this is a linear problem, $x_1$ might be the true root. Even a non-linear problem looks locally like a linear problem near a root:

$$g(x) \approx g(\bar{x}) + g'(\bar{x})(x - \bar{x}) + O(|x - \bar{x}|^2)$$

To extrapolate, we need the eqn of the line through the data points: $(x_0, e_0)$ and $(\tilde{x}_1, \tilde{e}_1)$

$$y = \frac{\tilde{e}_1 - e_0}{\tilde{x}_1 - x_0}(x - x_0) + e_0$$

To see where this crosses the x-axis we solve for $x_1$ in $(x_1, 0)$

extrapolated error.

$$0 = \frac{\tilde{e}_1 - e_0}{\tilde{x}_1 - x_0}(x_1 - x_0) + e_0$$

or

some algebra

$$x_1 = x_0 - \underbrace{\left(\frac{\tilde{x}_1 - x_0}{\tilde{e}_1 - e_0}\right) e_0}_{\approx\, x_0 - e_0 \,\equiv\, \bar{x}} \approx \bar{x}$$

this iteration can be used to get each new update.

A more convenient form is given by:

$$X_1 = X_0 - \frac{(g(x_0) - x_0)^2}{g(g(x_0)) - 2g(x_0) + X_0}$$

this requires 2 function evaluations

How does this method behave on a linear function?

$$g(x) = ax + b = x$$

$$X_1 = X_0 - \frac{(ax_0 + b - x_0)^2}{a(ax_0+b) + b - 2(ax_0+b) + x_0}$$

(some algebra ...)

$$X_1 = \frac{-b}{a-1} = \bar{x}, \quad g(\bar{x}) = \bar{x}$$

So this iteration solves the linear problem in one step. ☺

**Example:** $g(x) = 0.1x + 1$ ; $x_0 = 0$

| | $x_k$ | $\lvert x_k - x_{k-1} \rvert$ |
|---|---|---|
| Steffensen's Method | | |
| 0 | 0.0000000000000000e+00 | |
| 1 | 1.1111112345679148e+00 | 1.1111e+00 |
| 2 | 1.1111112345679151e+00 | 2.2204e-16 |

The idea is that if we apply this to nonlinear problems, we can speed up the convergence near a root.

# Steffensen's Algorithm

```
function [xroot,en] = steffensens(g,x0,tol,kmax)

xkm1 = x0;
for k = 1:kmax
    gk = g(xkm1);
    ggk = g(gk);
    D = (ggk - 2*gk + xkm1);
    if (D == 0)
        fprintf('Tolerance achieved\n');
        xroot = g(xkm1);
        break;
    else
        xk = xkm1 - (gk-xkm1)^2/D;
    end

    en(k) = abs(xk-xkm1);
    fprintf('%5d %20.16e %12.4e\n',k,xk,en(k));
    if (en(k) < tol)
        fprintf('Tolerance achieved\n');
        xroot = xk;
        break;
    end
    xkm1 = xk;
end
xroot = xk;

end
```

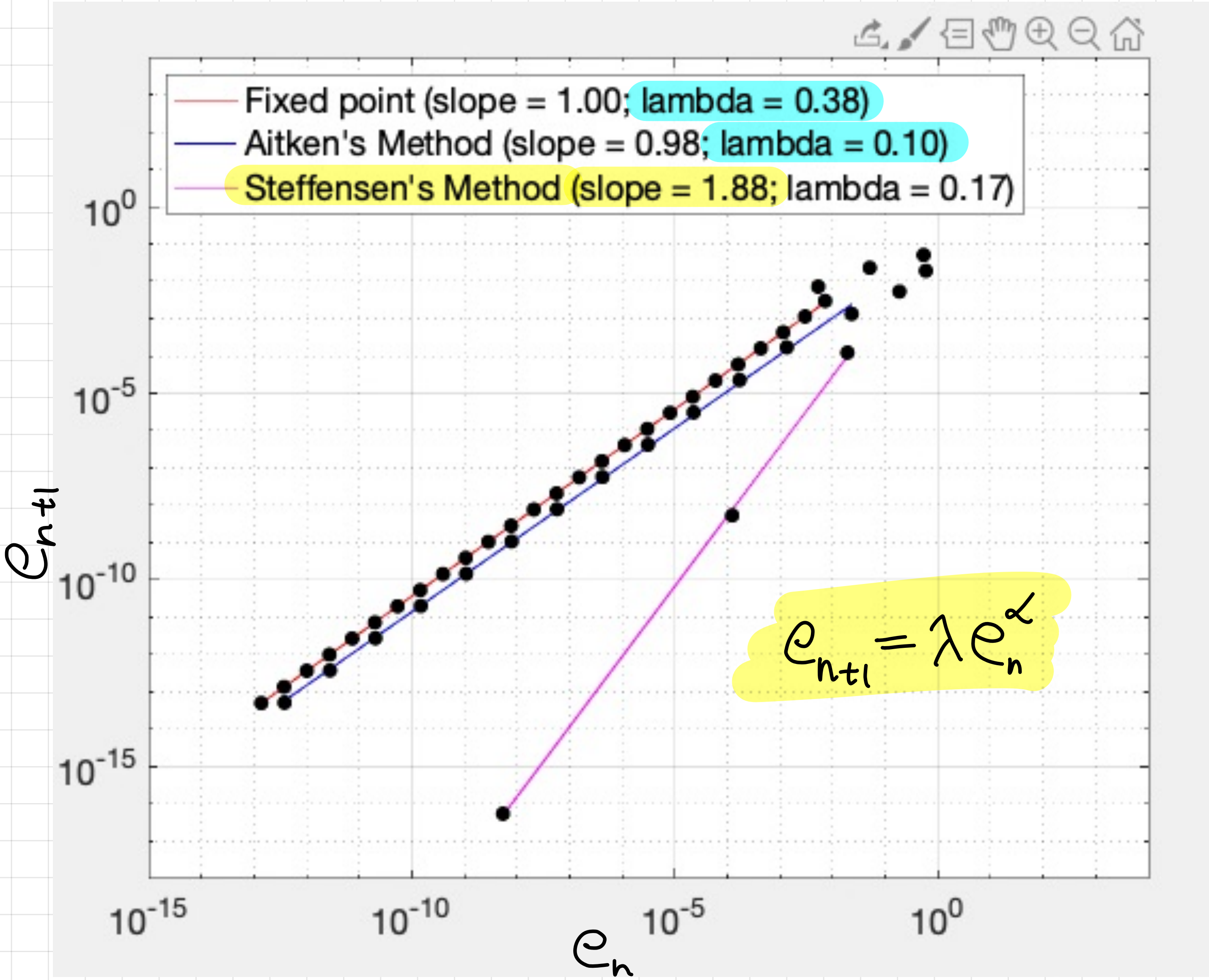
check that the denominator is not zero.

- We have to check the denominator in the update.
- Show that at a root $\bar{x}$:

$$\lim_{k \to \infty} x_k - \frac{(g(x_k) - x_k)^2}{g(g(x_k)) - 2g(x_k) + x_k} = g(\bar{x}) = \bar{x}$$

**Example:** $f(x) = \frac{1}{3}x^3 - x^2 + \frac{4}{3}\beta \quad , \beta = 0.1$

$$g(x) = \frac{1}{3}x^3 - x^2 + x + \frac{4}{3}\beta$$




- Aitken's method (not discussed) - improves asymptotic error constant $\lambda$.
- Steffensen's method is quadratically convergent. $\Rightarrow$ improves order