Maria Cameron

Contents

1. Numerical methods for solving nonlinear equations

Reading:

- A. Gil, J. Segura, N. Temme, Numerical Methods for Special Functions, SIAM 2007, Chapter 7, Sections 7.2.1 and 7.2.2.
- G. W. Stewart, Afternotes on Numerical Analysis, Lectures 1–5.

We will be discussing, analyzing, and experimenting with the following list of methods for solving $f(x) = 0$ where f is a continuous function and $x \in \mathbb{R}$:

- Bisection method
- Newton's method a.k.a. the Newton-Raphson method
- Secant method
- Hybrid method (of secant and bisection)

2. Fixed point methods for nonlinear equations

The basic idea of fixed point methods consists in finding an iteration function $T(x)$ such that (i) the zero x^* of $f(x)$ satisfies

$$T(x^*) = x^*,$$

and (ii) $T(x)$ generates successive approximations to the solution

$$x_{n+1} = T(x_n)$$

starting from the provided initial approximation x_0.

We start with considering linear maps $x_{n+1} = T(x_n)$. the are four different cases leading to four qualitatively different behaviors of sequences of iterates illustrated in Fig. 1. Such figures where the sequence of iterates on the plane $x_n x_{n+1}$ are connected with line segments and the line $x_n = x_{n+1}$ is plotted are called Lamerei's diagrams. We see that

- if $|T'| < 1$ the sequence converges to a fixed point, while if $|T'| > 1$ the sequence diverges;

- If $T' > 0$, the sequence is monotone, while if $T' < 0$ the sequence oscillates.

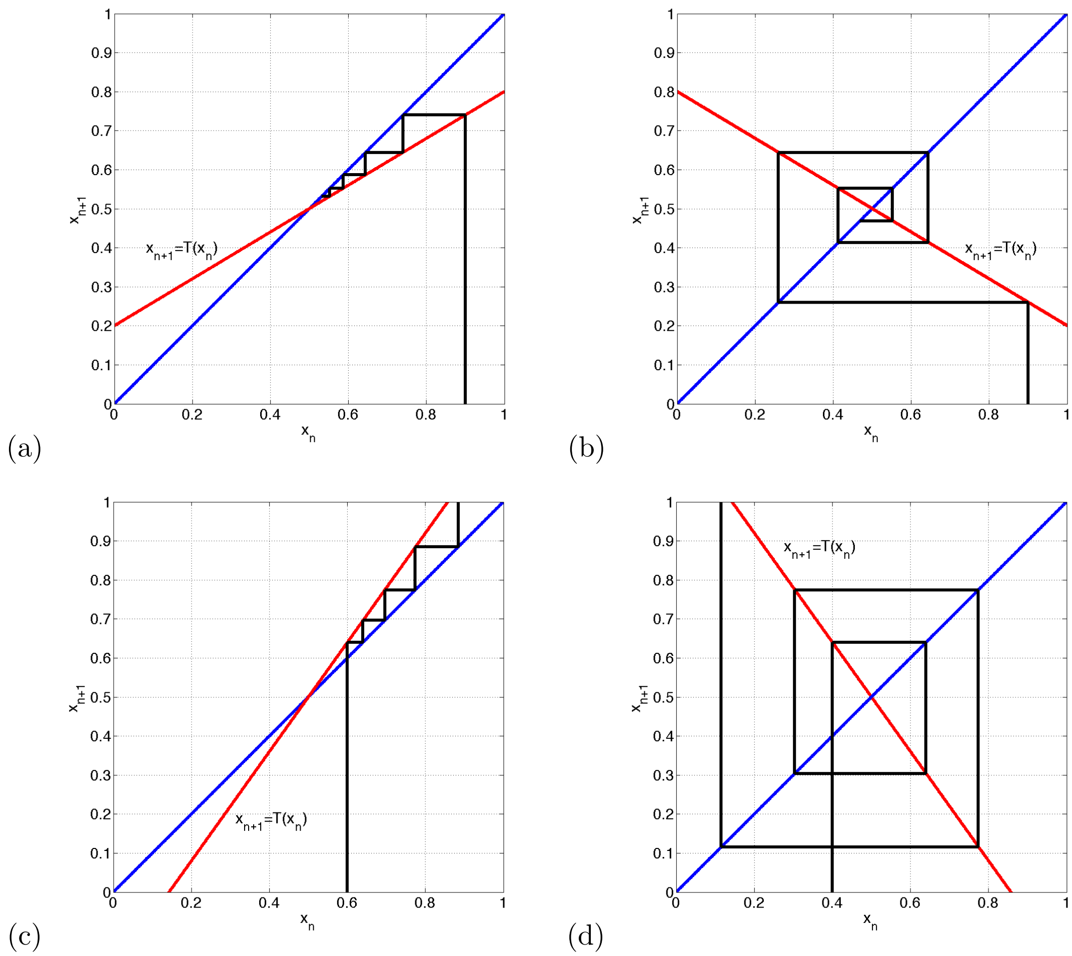

FIGURE 1. Lamerei's diagrams for linear mappings $x_{n+1} = T(x_n)$. (a): Case $0 < T' < 1$. (b): Case $-1 < T' < 0$. (c): Case $T' > 1$. (d): Case $T' < -1$.

Fig. 1 suggests that the sequence of iterates $x_{n+1} = T(x_n)$ starting from the initial approximation x_0 converges to $x^* = T(x^*)$ if the function T is a *contraction* or a *contraction mapping* on some interval containing x_0 and x^*.

Definition 1. *A function $f(x)$ is contraction on the interval $[a, b]$ if*

(1) *it maps the interval $[a, b]$ into itself, i.e., $f([a, b]) \subset [a, b]$, and*

(2) *there exists a constant* $0 \le M < 1$ *such that for any* $x, y \in [a,b]$ *we have*

$$|f(x) - f(y)| \le M|x - y|.$$

There is a remarkable fact associated with a contraction mapping on any compact set in $\mathbb{R}^n$ called Brouwer's Fixed Point Theorem. In 1D this theorem is easy-to-prove and we will do it below. In $\mathbb{R}^n$, its proof is much harder.

Theorem 1. (Brouwer's fixed point theorem in 1D) *Let* $f(x)$ *be a contraction mapping on the interval* $[a,b]$. *Then there exists a point* $c \in [a,b]$ *such that* $f(c) = c$.

Proof. Note that since $f(x)$ is a contraction on $[a,b]$ it is continuous on $[a,b]$. Since $f([a,b]) \subset [a,b]$, we have that

$$f(a) \in [a,b], \text{ i.e. } a \le f(a) \le b, \text{ hence } f(a) - a \ge 0,$$
$$f(b) \in [a,b], \text{ i.e. } a \le f(b) \le b, \text{ hence } f(b) - b \le 0.$$

Therefore, the continuous function

$$g(x) := f(x) - x$$

has different signs at the ends of the interval $[a,b]$. Hence by the intermediate value theorem there is a point $c \in [a,b]$ such that $g(c) = f(c) - c = 0$, i.e. $f(c) = c$. $\square$

The next theorem shows how the fixed point theorem applies to iterative methods for solving nonlinear equations.

Theorem 2. *Let* $T(x)$ *be a continuous and differentiable function in* $[a,b]$ *such that*

$$T([a,b]) \subset [a,b] \quad \text{and} \quad |T'(x)| \le M < 1, \quad x \in [a,b].$$

Then

- *there exists a unique point* $x^* \in [a,b]$ *such that* $T(x^*) = x^*$;
- *and for any* $x_0 \in [a,b]$ *the sequence of iterates* $x_{n+1} = T(x_n)$, $n = 0, 1, 2, \dots$, *converges to* x^*;
- *the error after the* n*-th iteration is bounded by*

$$|x_n - x^*| \le \frac{M^n}{1-M}|x_1 - x_0|.$$

Proof. (1) **T is a contraction.** The requirements

$$T([a,b]) \subset [a,b] \quad \text{and} \quad |T'(x)| \le M < 1 \quad \text{for all} \quad x \in [a,b]$$

guarantee that $T(x)$ is a contraction. Indeed, it follows from the intermediate value theorem that

$$T(x) - f(y) = T'(z)(x - y) \quad \text{for some } z \in (y, x) \subset [a,b].$$

Since $|T'(z)| \le M < 1$ we have

$$|T(x) - T(y)| \le M|x - y|.$$

Hence $T(x)$ is a contraction.

(2) **Uniqueness.** Suppose there are two fixed points: $y = T(y)$ and $z = T(z)$. Then applying the mean value theorem we obtain

$$|y - z| = |T(y) - T(z)| = |T'(\zeta)||y - z| \leq M|y - z| < |y - z|,$$

a contradiction. Hence there is at most one fixed point in $[a, b]$.

(3) **Existence.** Let us show that the sequence $x_{n+1} = T(x_n)$ is Cauchy. We have

$$|x_{n+k} - x_n| = |T(x_{n+k-1}) - T(x_{n-1})| \leq M|x_{n+k-1} - x_{n-1}| \leq \ldots \leq M^n|x_k - x_0| \leq M^n|b - a|.$$

Hence this difference can be made arbitrarily small as soon as n is large enough. Hence this sequence is Cauchy and hence it converges.

(4) **Error bound.** Let x^* be the fixed point, i.e., $T(x^*) = x^*$.

$$|x_n - x^*| = |T(x_{n-1}) - T(x^*)| \leq M|x_{n-1} - x^*| \leq \ldots \leq M^n|x_0 - x^*|.$$

On the other hand,

$$|x_0 - x^*| \leq |x_0 - x_1| + |x_1 - x^*| \leq |x_0 - x_1| + M|x_0 - x^*|.$$

Hence,

$$(1 - M)|x_0 - x^*| \leq |x_0 - x_1|.$$

Therefore we get the following error bound:

$$|x_n - x^*| \leq M^n|x_0 - x^*| \leq \frac{M^n}{1 - M}|x_0 - x_1|.$$

$\square$

3. The order of convergence

The performance of any iterative algorithm for solving nonlinear equations is characterized by

- its ability to find a solution (global convergence or local convergence), and
- how fast the convergence takes place.

An important quantitative characteristic of such an algorithm is the order of convergence. We start with the simplest definition.

Definition 2. *Let $\{x_n\}_{n=0}^{\infty}$ be a sequence which converges to x^* and such that $x_n \neq x^*$ for $n \in \mathbb{N}$. We will say that the sequence converges with order $p \geq 1$ if there exists a constant $C > 0$ such that*

$$\lim_{n \to \infty} \frac{|e_{n+1}|}{|e_n|^p} = C, \tag{1}$$

where $e_n = x_n - x^*$, $n \in \mathbb{N}$. C is called the asymptotic error constant.

Remark Definition 2 accounts for an ideal situation where the limit in (1) exists. This is often the case when we are dealing with 1D problems as in this course and often not the case for nonlinear solvers in higher dimensions. Then a relaxed version of the definition of the order of convergence is used: we say that a sequence $\{x_n\}$ converges to x^* with order p if for large enough n one can find a constant C such that

$$\|e_{n+1}\| \le C\|e_n\|^p.$$

Remark Typically, the values of p do not exceed 3.

- $p = 1$. Then we say that the *convergence is linear*. Example: the bisection method: $p = 1$, $C = 1/2$.
- $p = 2$ corresponds to the *quadratic convergence* as in Newton's method.
- $p = 3$ occurs rarely. We call such convergence *cubic*. One such method is the Rayleigh iteration for finding eigenpairs for a symmetric matrix.
- $p \in (1, 2)$. Then we say that the *convergence is superlinear*. This is the case for the secant method: $p = (1 + \sqrt{5})/2$.

Now we determine the order of convergence for a fixed point method $x_{n+1} = T(x_n)$. Let us assume that $T(x)$ is smooth, i.e., differentiable as many times as we desire. We define

$$e_n := x_n - x^*$$

and consider the Taylor expansion

$$x_{n+1} = T(x_n) = T(x^* + e_n) = T(x^*) + T'(x^*)e_n + \frac{1}{2}T''(x^*)e_n^2 + \dots. \tag{2}$$

Let $T^{(p)}(x^*)$ be the first non-vanishing derivative of T at x^*. Then we subtract x^* from both sides of (2) and get:

$$e_{n+1} = \frac{T^{(p)}(x^*)}{p!}e_n^p + O(e_n^{p+1}).$$

Hence

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^p} = \left|\frac{T^{(p)}(x^*)}{p!}\right|, \tag{3}$$

i.e., the order of convergence and the asymptotic error constant are determined by the first nonvanishing derivative of the contraction mapping $T(x)$ at the fixed point x^*.

4. Newton's method

For Newton's method (read G. W. Stewart, Afternotes on Numerical Analysis, Lecture 2),

$$T(x) = x - \frac{f(x)}{f'(x)}.$$

Differentiating, we obtain

$$T'(x) = 1 - \frac{f'(x)}{f'(x)} + \frac{f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2},$$

$$T''(x) = \frac{f''(x)}{f'(x)} + \frac{f(x)f'''(x)}{[f'(x)]^2} - \frac{2f(x)[f''(x)]^2}{[f'(x)]^3]}.$$

Since $f(x^*) = 0$, we have

$$T'(x^*) = 0, \quad T''(x^*) = \frac{f''(x^*)}{f'(x^*)}.$$

Hence, if $f(x)$ is twice continuously differentiable and $f'(x^*) \neq 0$ and $f''(x^*) \neq 0$, Newton's method converges quadratically, and, using Eq. (3), we obtain the asymptotic error constant

$$C = \left| \frac{f''(x^*)}{2f'(x^*)} \right|.$$

4.1. **Examples.**

4.1.1. *The Babylonian method for finding square roots.* Let $a > 0$ be a given number. Consider the fixed point iteration

$$x_{n+1} = T(x_n), \quad \text{where } T(x) = \frac{1}{2}\left(\frac{a}{x} + x\right). \tag{4}$$

(1) Show that $x = \sqrt{a}$ is a fixed point of this iteration (it is the Babylonian method for computing $\sqrt{a}$).
(2) What is the order of convergence to $\sqrt{a}$?
(3) Find the interval where the map $T(x)$ is a contraction.
(4) What will happen with the iterates if the starting point x_0 does not belong to the interval you have found?

Solution First, we plot the graph of the iteration function T and the line $x_{n+1} = x_n$ in the (x_n, x_{n+1})-plane (see Fig. 2). The positive branch of the graph of T has a vertical asymptote $x = 0$, a slant asymptote $x_{n+1} = {}^{x_n}\!/_2$, and a unique extremum, a minimum at $x^* = \sqrt{a}$. It is always good to take a few initial values and see how they iterate.

(1)
$$T(x) = \frac{1}{2}\left(\frac{a}{x} + x\right) = x, \quad \text{hence} \quad 2x^2 - a = 0.$$
Therefore, $x = \sqrt{a}$ and $x = -\sqrt{a}$ are the fixed points of $T(x)$.

(2) In order to determine the order of convergence, we need to find the first nonvanishing derivative of T at $x = \sqrt{a}$.
$$T'(x) = \frac{1}{2}\left(1 - \frac{a}{x^2}\right), \quad T(\sqrt{a}) = 0.$$
$$T''(x) = \frac{a}{x^3}, \quad T(\sqrt{a}) = \frac{1}{\sqrt{a}} \neq 0.$$
Therefore, the order of convergence is $p = 2$.

(3) We need to find the maximal interval where $|T'(x)| \le 1$.

$$\begin{aligned}
\frac{1}{2}\left|1-\frac{a}{x^2}\right| &\le 1 \\
-2 \le 1-\frac{a}{x^2} &\le 2 \\
-3 \le -\frac{a}{x^2} &\le 1 \\
-1 \le \frac{a}{x^2} &\le 3 \\
x^2 &\ge \frac{a}{3} \\
x &\ge \sqrt{\frac{a}{3}}, \quad \text{or} \quad x \le -\sqrt{\frac{a}{3}}.
\end{aligned}$$

Hence, $T(x)$ is a contraction on $[\sqrt{a/3},\infty)$ and on $(-\infty,-\sqrt{a/3}]$. Indeed, for any $x,y \in [\sqrt{a/3},\infty)$, $x \ne y$, we have

$$|T(x)-T(y)| = |T'(z)||x-y| < |x-y|, \text{ where } z \in (x,y).$$

(4) The iterations will converge to $\sqrt{a}$ starting from any $x_0 > 0$ (see Fig. 2). Indeed, note that x^* is the minimum of $T(x)$ on the positive semiaxis. Hence for all

$$x_0 < x^* \text{ we will have } x_1 = f(x_0) > f(x^*) = x^* > \frac{x^*}{\sqrt{3}} = \sqrt{\frac{a}{3}}.$$

Therefore, x_1 either way will fall to the zone of convergence.

We also can equate $|x_0 - x^*| = |x_1 - x^*|$ and find that if $x_0 < \frac{1}{3}\sqrt{a}$, then $|x_0-\sqrt{a}| < |x_1-\sqrt{a}|$, i.e., the error will decrease starting from the first iteration.

Remark We have just established that the iteration (4) will converge to $\sqrt{a}$ starting from any positive value x_0. Does it mean that we should not care about the choice of the initial approximation at all? The answer is: we should. Suppose that the initial guess is either large ($\gg \sqrt{a}$) or small. In the latter case, the first iterate will be large. Thus, either way, let us consider iterations starting from some large value x_0. Then $x_1 \approx {}^{x_0}/_2$, ..., $x_n \approx {}^{x_0}/_{2^n}$, i.e., the early iterations will be make as fast progress as bisection iterations until x_n will become small enough to make the contribution of ${}^{a}/_{x_n}$ substantial, i.e, until $x_n \sim 3\sqrt{a}$ (which approximately corresponds to ${}^{a}/_{x_n} = 0.1x_n$). Only after then the quadratic convergence will kick in.

4.1.2. *Finding specific roots.* Consider the problem of finding N smallest positive roots of the oscillating function

$$f(x) = \kappa x \sin(x) - \cos(x). \tag{5}$$

The graph of f for $\kappa = 0.9$ is shown in Fig. 3 (the oscillating curve). We calculate the derivative of f:

$$f'(x) = (\kappa+1)\sin(x) - \kappa x \cos(x),$$

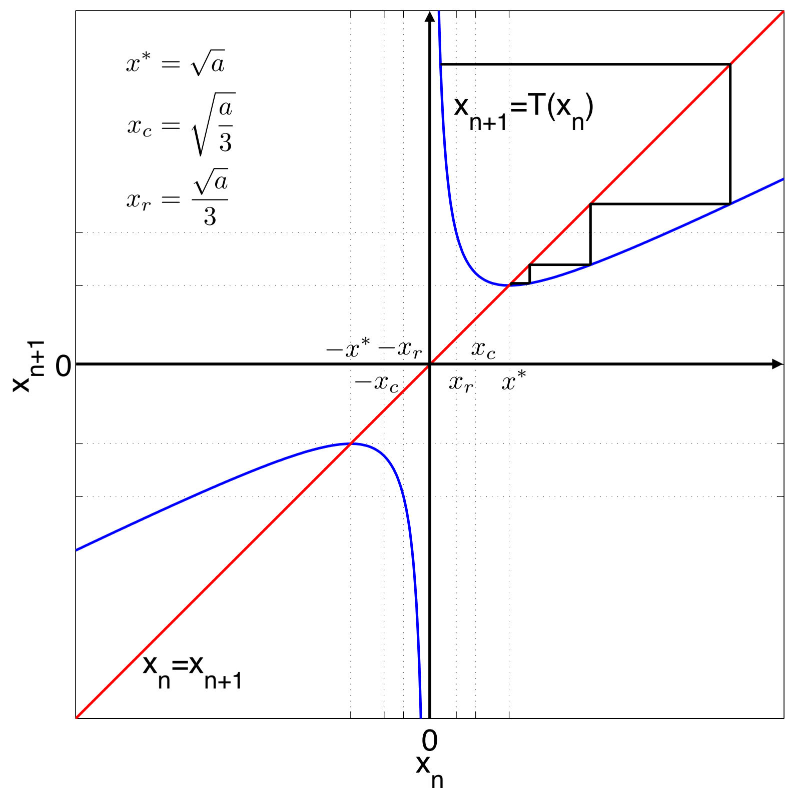

FIGURE 2. The Balylonian method for finding $\sqrt{a}$, the iteration function and some important points: $x^* = \pm\sqrt{a}$ are the fixed points where $T(x^* = x^*$; $\pm x_c$ bound the intervals where $T(x)$ is a contraction (($T(x)$ is a contraction on $(-\infty, -x_c]$ and on $[x_c, +\infty)$); $\pm x_r$ bound the intervals of monotone decay of the absolute value of the error, i.e., if $x_0 \in (x_r, +\infty)$ then $|e_{n+1}| < |e_n|$ for all n and $x_n \to \sqrt{a}$, if $x_0 \in (-\infty, -x_r)$ then $|e_{n+1}| < |e_n|$ for all n and $x_n \to -\sqrt{a}$.

and use Newton's iteration starting from an initial guess for each root. The Matlab code below sets the problem up and looks for a root of f starting with the initial guess `y0 = 7`.

```
 function MyZeroFinder()
% Task: find the 10 smallest positive roots
kap = 0.9;
fun = @(x)kap*x.*sin(x) - cos(x);
tol = 1e-14;
```

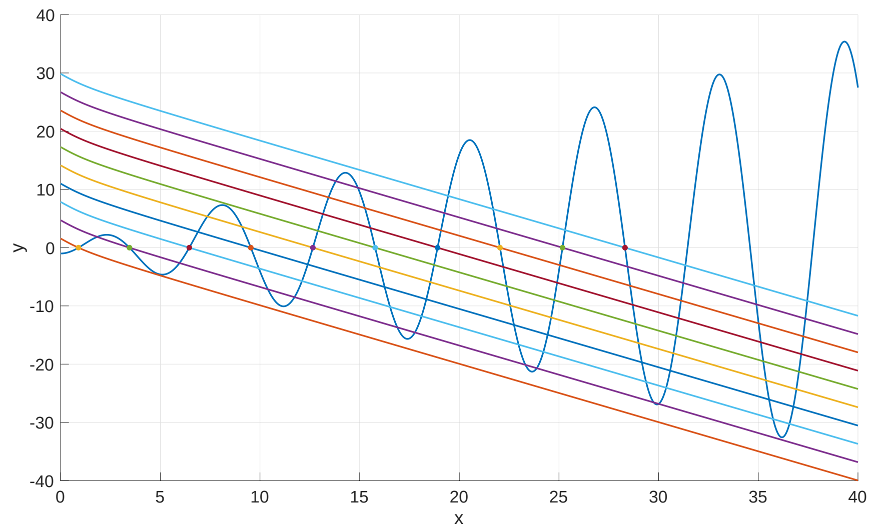

FIGURE 3. Finding the 10 smallest roots of $f(x) = 0.9x\sin(x) - \cos(x)$. The oscillatory curve is the graph of $f(x)$. The rest of the curves are the graphs of $g_m(x) = \arctan\left(\frac{1}{0.9x}\right) + \pi m - x$, $m = 0, 1, \ldots, 9$.

```
x = linspace(1e-10,40,1000);
fx = fun(x);
flag = 0;
if flag == 0
    figure(1); clf; hold on; grid;
    set(gca,'Fontsize',20);
    xlabel('x','Fontsize',20);
    ylabel('y','Fontsize',20);
end
plot(x,fx,'Linewidth',2);
%% Choose approach
approach = 1;
%% Attempt 1: use Newton's method to solve the problem as it is formulated
if approach == 1
    y = 7; % the initial guess
    fy = fun(y);
    fder = @(x) (kap + 1)*sin(x) + kap*x.*cos(x);
    iter = 0;
    while abs(fy) > tol
        y = y - fy/fder(y);
        fy = fun(y);
```

```
        iter = iter + 1;
        fprintf('iter #%d: y = %.15e, fy = %.15e\n',iter,y,fy);
    end
    plot(y,fy,'.','Markersize',20);
end
```

This approach is reasonable provided that we have the graph of $f(x)$ and can give close enough initial guesses. However, imagine that you need to provide initial guesses without being able to see the graph.

Let us try to reformulate the problem in order to isolate the roots. We observe finding positive roots of $f(x) = 0$ is equivalent to solving a series of nonlinear equations

$$g_m(x) := \arctan\left(\frac{1}{\kappa x}\right) + \pi m - x, \quad m = 0, 1, 2, \ldots. \tag{6}$$

Each of the graphs $g_m(x)$ crosses the x-axis exactly once. Moreover, the first two derivatives of $g_m(x)$ are:

$$g_m'(x) = -\frac{\kappa}{1+\kappa^2x^2} - 1,$$
$$g_m''(x) = \frac{2\kappa x^3}{(1+\kappa^2x^2)^2}.$$

We see that $g_m'(x)$ is approximately -1 for large x, while $g_m''(x)$ is positive and approaches zero for large x. Hence the initial guesses for the first 10 zeros do need to be chosen close to actual values – Fig. 3.

Here is a continuation of the Matlab code above implementing this approach.

```
%% Attempt 2: reformulate the problem and use Newton's method
if approach == 2
    tic
    for m = 0 : 9
        fprintf('Find zero #%d\n',m);
        fun = @(x)atan(1.0./(kap*x)) - x + pi*m;
        fx = fun(x);
        y = 1;
        fy = fun(y);
        plot(x,fx,'Linewidth',2);
        fder = @(x)-kap./(1 + (kap*x).^2) - 1;
        iter = 0;
        while abs(fy) > tol
            y = y - fy/fder(y);
            fy = fun(y);
            iter = iter + 1;
            fprintf('iter #%d: y = %.15e, fy = %.15e\n',iter,y,fy);
        end
        plot(y,fy,'.','Markersize',20);
```

```
    end
    toc
end
```

4.2. **Issues with multiple roots.** We have established that Newton's method converges quadratically provided that f is twice continuously differentiable in some interval surrounding x^*, $f'(x^*) \neq 0$, and the initial guess is sufficiently close to x^*. Let us understand what happens when $f'(x^*) = 0$, i.e., the root x^* is multiple.

Suppose x^* is a root of multiplicity m, i.e., $f(x^*) = 0$, $f'(x^*) = 0$, ..., $f^{(m-1)}(x^*) = 0$, while $f^{(m)}(x^*) \neq 0$. Then

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{(m-1)!f^{(m)}(x^*)(x-x^*)^m + O((x-x^*)^{m+1})}{m!f^{(m)}(x^*)(x-x^*)^{(m-1)} + O((x-x^*)^{m+1})}.$$

Hence

$$e_{n+1} = e_n - \frac{e_n}{m} + O(e_n^2) = \left(1 - \frac{1}{m}\right) e_n + O(e_n^2).$$

This shows that the convergence will be linear, and the larger m, the closer the error constant to one.

5. Need for quasinewton methods

Suppose we need to solve a nonlinear equation $f(x) = 0$ where the derivative of f is unavailable. For example, suppose that we need to find the value of the parameter μ in the Van der Pol oscillator

$$\begin{aligned} \dot{y_1} &= y_2, \\ \dot{y_2} &= \mu(1-y_1^2)y_2 - y_1. \end{aligned} \tag{7}$$

such that the maximal value of y_2 in the periodic solution is equal to 10. For each $\mu \geq 0$, ODE (7) has a stable and globally attractive periodic solution. The following Matlab function finds the maximal value of y_1 in the periodic solution for a given μ:

```
function yout = FindPeriodicSolution(mu0)
global mu
mu = mu0;
% First find the periodic solution for the given mu
%% The Van der Pol oscillator
VDPol = @(t,y)[y(2);mu*(1 - y(1).^2).*y(2) - y(1)];
options = odeset('Reltol',1e-12,'Abstol',1e-12,'Events',@events);
%%
Tmax = 100;
figure(1); hold on; grid;
y0 = [0,1];
tol = 1e-10;
d = 10;
iter = 0;
```

```
while d > tol
    [T,Y] = ode45(VDPol,[0,Tmax],y0,options);
%     plot(Y(:,1),Y(:,2),'Linewidth',2);
%     drawnow;
    y1 = Y(end,:);
    d = norm(y1 - y0);
    y0 = y1;
    iter = iter + 1;
%     fprintf('iter #%d: d = %d\n',iter,d);
    yout = y0(2);
end
[T,Y] = ode45(VDPol,[0,Tmax],y0,options);
plot(Y(:,1),Y(:,2),'Linewidth',2);

end
%%
%%
function [position,isterminal,direction] = events(t,y)
global mu
position = mu*(1 - y(1).^2).*y(2) - y(1); % The value that we want to be zero
isterminal = 1;  % Halt integration
direction = -1;   % The zero can be approached from the negative direction
end
```

Hence, the function whose root we need to find is given by:

```
fun = @(x)FindPeriodicSolution(x) - 10;
```

The Matlab code below fulfills this task in two ways: using the bisection method and using a quasinewton method where the derivative is approximated by a forward difference.

```
function SolveNonlinEq()
close all
figure(1); hold on; grid;
% Task: find the value of mu at which the maximal value of
% y(2) in the periodic solution reaches 10
fun = @(x)FindPeriodicSolution(x) - 10;
tol = 1e-12;
method = 2;

%% Use the bisection method
if method == 1
% Bracket the root
a = 1.0;
fa = fun(a);
b = 10;
```

```
fb = fun(b);
fprintf('a = %d, fa = %d; b = %d, fb = %d\n',a,fa,b,fb);
iter = 0;
tic
while abs(fa - fb) > tol
    c = 0.5*(a + b);
    fc = fun(c);
    if fa*fc <= 0
        b = c;
        fb = fc;
    else
        a = c;
        fa = fc;
    end
    iter = iter + 1;
    fprintf('iter #%d: a = %d, fa = %d; b = %d, fb = %d\n',iter,a,fa,b,fb);
end
toc
end
%% Use a quasinewton method
if method == 2
tic
h = 1e-1;
a = 1.0;
fa = fun(a);
b = a + h;
fb = fun(b);
der = (fb - fa)/h;
d = 10;
iter = 0;
while abs(fa) > tol
    a = a - fa/der;
    fa = fun(a);
    b = a + h;
    fb = fun(b);
    der = (fb - fa)/h;
    iter = iter + 1;
    fprintf('iter #%d: a = %d, fa = %d; b = %d, fb = %d\n',iter,a,fa,b,fb);
end
toc
end
end
```

To satisfy the error tolerance of 10^{-12}, the bisection method requires 44 iterations, while the quasinewton method achieves the same goal just in 5 iterations.

The drawback of the finite-difference quasinewton method used here is two-fold: First, it requires an extra function evaluation. Second, it requires the user to provide a value of h, the finite difference step. Too large or too small values of h will result in an inaccurate estimate for the derivative and slowdown of convergence.

A better approach is the secant method (read G. W. Stewart, Afternotes on Numerical Analysis, Lecture 4.

6. Secant method

The secant method is almost as fast as Newton's but has the advantage that it does not require an extra function evaluation. The derivative is approximated by the slope a secant line giving the name for this method. We start with two initial values x_0 and x_1 and iterate:

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}. \tag{8}$$

Let us establish its order of convergence. We will see that the asymptotic order of convergence is

$$\alpha = \frac{1+\sqrt{5}}{2} \approx 1.6180.$$

Subtracting x^* from both sides of (8) we get:

$$e_{n+1} = e_n - f(x_n)\frac{e_n - e_{n-1}}{f(x_n) - f(x_{n-1})} = \frac{f(x_n)e_{n-1} - f(x_{n-1})e_n}{f(x_n) - f(x_{n-1})}. \tag{9}$$

Assuming e_n and e_{n+1} are close to zero, we Taylor expand $f(x_n)$ and $f(x_{n-1})$ near x^*. Recall that $f(x^*) = 0$. Then

$$\begin{aligned} f(x_n) = f(x^* + e_n) &= e_n f'(x^*) + e_n^2\frac{f''(x^*)}{2} + O(e_n^3), \\ f(x_{n-1}) = f(x^* + e_{n-1}) &= e_{n-1} f'(x^*) + e_{n-1}^2\frac{f''(x^*)}{2} + O(e_{n-1}^3). \end{aligned}$$

Plugging these expressions into (9) we get:

$$\begin{aligned} e_{n+1} &= \frac{e_{n-1}e_n f'(x^*) + e_{n-1}e_n^2\frac{f''(x^*)}{2} - e_n e_{n-1} f'(x^*) - e_n e_{n-1}^2\frac{f''(x^*)}{2} + O(e_{n-1}e_n^3) + O(e_n e_{n-1}^3)}{e_n f'(x^*) + e_n^2\frac{f''(x^*)}{2} - e_{n-1}f'(x^*) - e_{n-1}^2\frac{f''(x^*)}{2} + O(e_n^3) + O(e_{n-1}^3)} \\ &= e_n e_{n-1}\frac{1/2 f''(x^*)(e_n - e_{n-1}) + O(e_n^2) + O(e_{n-1}^2)}{f'(x^*)(e_n - e_{n-1}) + O(e_1^2) + O(e_{n-1}^2)} \end{aligned}$$

Therefore,

$$\lim_{n\to\infty}\frac{e_{n+1}}{e_n e_{n-1}} = \frac{f''(x^*)}{2f'(x^*)}. \tag{10}$$

We have assumed that $f'(x^*) \neq 0$. Now let us find the order of convergence. We wish to find α such that $e_{n+1} \approx C e_n^\alpha$ where C is some unknown constant, i.e.,

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n^\alpha} = C.$$

Then

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n e_{n-1}} = \lim_{n\to\infty} \frac{C\left(C e_{n-1}^\alpha\right)^\alpha}{C e_{n-1}^\alpha e_{n-1}} = \lim_{n\to\infty} e_{n-1}^{\alpha^2-\alpha-1} C^\alpha = C^\alpha,$$

which happens if and only iff

$$\alpha^2 - \alpha - 1 = 0, \quad \text{i.e.} \quad \alpha = \frac{1 \pm \sqrt{5}}{2}.$$

We select the positive root that corresponds to decaying error:

$$\alpha = \frac{1+\sqrt{5}}{2}.$$

References

[1] A. Gil, J. Segura, N. Temme, Numerical Methods for Special Functions, SIAM, 2007
[2] J. Nocedal, S. Wright, Numerical Optimization, Springer, 1999
[3] C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, 1995.