Numerical Methods I – Problem Sheet 1

Matúš Harvan, Marcel Oliver

September 24, 2003

5. (a) Consider the equation $x^2 + px + 1 = 0$. We use the well known formula for roots of a quadratic equation

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2}$$

so that

$$x_1 = \frac{-p - \sqrt{p^2 - 4}}{2} \tag{1}$$

$$x_2 = \frac{-p + \sqrt{p^2 - 4}}{2} \tag{2}$$

The claim is that $x_1 \sim -p$ and $x_2 \sim -1/p$, where $x \sim y$ denotes that x *asyptotes* y, i.e. that

$$\lim_{p\to\infty} \frac{x(p)}{y(p)} = 1\,.$$

This condition is easily checked for x_1:

$$\lim_{p\to\infty} \frac{x_1}{-p} = \lim_{p\to\infty} \frac{-p - \sqrt{p^2 - 4}}{-2p} = \lim_{p\to\infty} \frac{1 + \sqrt{1 - 4/p^2}}{2} = 1\,.$$

For x_2 we write

$$\begin{aligned} x_2 &= -\frac{p - \sqrt{p^2 - 4}}{2} \\ &= -\frac{p - \sqrt{p^2 - 4}}{2} \cdot \frac{p + \sqrt{p^2 - 4}}{p + \sqrt{p^2 - 4}} \\ &= -\frac{p^2 - (p^2 - 4)}{2p + 2\sqrt{p^2 - 4}} \\ &= -\frac{4}{2p + 2\sqrt{p^2 - 4}} \\ &= -\frac{2}{p + \sqrt{p^2 - 4}}\,. \end{aligned} \tag{3}$$

Therefore,

$$\lim_{p\to\infty} \frac{x_2}{-1/p} = \lim_{p\to\infty} \frac{2p}{p + \sqrt{p^2 - 4}} = \lim_{p\to\infty} \frac{2}{1 + \sqrt{1 - 4/p^2}} = 1\,.$$

(b)
```
octave:1> format long; p=1e10;
octave:2> x1=(-p+sqrt((p^2)-4))/2
x1 = 0
octave:3> x2=(-p-sqrt((p^2)-4))/2
x2 = -10000000000
octave:4> x2better=(02/(p+sqrt(p^2-4)))
x2better =  1.00000000000000e-10
```

x1 and x2 are calculated using the standard formula, equation (2). However, the result for x_2 has an extremely large relative error.

(c) The stable way of computing x_2 is to use equation (3). In the above transcript it is denoted as `x2better` and has negligible relative error.

6. (a) The algorithm is in file `p6.m`[1]. A copy of it follows:

```
start=2;
stop=40;
z=[1:stop];
z(2)=2;
err=[1:stop];

function out = iter(z,n)
         out=(2^(n-(1/2))) * sqrt( 1-sqrt(1- (4^(1-n)) * (z^2)) );
end

for n=start:stop-1
        z(n+1)=iter1(z(n),n);
endfor

ideal=ones(1,stop).*pi;
err=abs(ideal.-z);
n=1:stop;

gset term postscript
gset output "p6_fig1.ps"
semilogy(n(start:stop),err(start:stop))
gset term x11
```

(b) The recursive formula contains $\sqrt{1-\sqrt{1-4^{1-n}\cdot\pi^2}} = \sqrt{1-\sqrt{1-2^{2-2n}\cdot\pi^2}}$. The especially bad part here is the $1-4^{1-n}\cdot\pi^2$. For $n = 17$ this gets $1-4^{-16}\pi = 1-2^{-32}\pi$, so that we subtract a very small number from 1. Although this is not yet smaller than ε_M, it already produces an error, as subtraction has rather higher relative error. This error is then propagated and further amplified as the formula is calculated (square roots, other subtractions, ...). At a certain point the $4^{1-n}\pi$ gets smaller than ε_M, so we get $2^{n-\frac{1}{2}}\sqrt{1-\sqrt{1-0}} = 0$ and there is no point in continuing the calculations further.

[1]MO: This code contains a few tricks that are specific to Octave. A more generic (and shorter)n Octave/Matlab code is in the file `piseq.m`

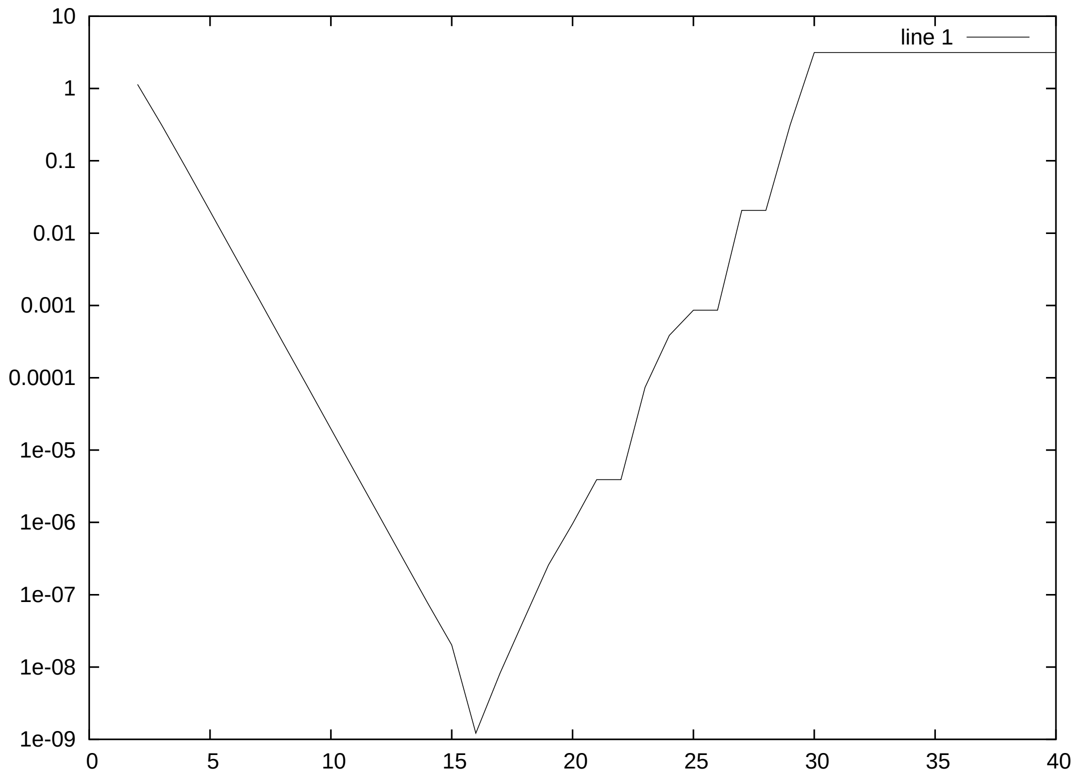
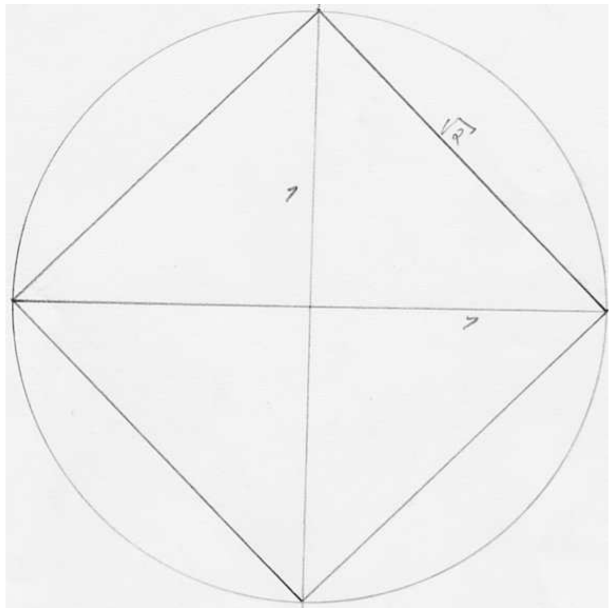


Figure 1: Output of `pt.m`: Error vs. n

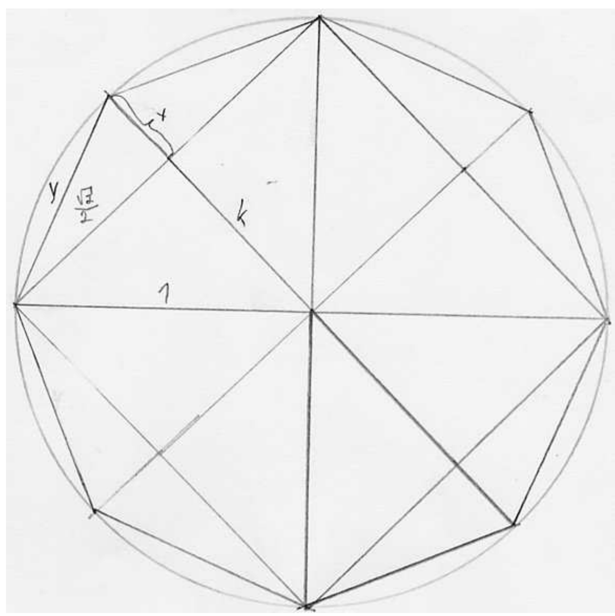
Another problem is that this is a recursive formula and even small errors at earlier steps are carried on to further steps and get amplified.

(c) The idea behind the formula is to take a unit circle and divide it into a finite number of triangles (2^n). Then areas of those triangles are added together. Their total area approximates the area of the unit circle, which is π. The more triangles the unit circle is divided into, the more accurate the approximation is.



$n = 2$

The first iteration is for $n = 2$ e.g. drawing four triangles (which together form a square) into the unit circle. This square contains four triangles and has side of length $\sqrt{1^2 + 1^2} = \sqrt{2}$ and area of 2. This is the starting point of out sequence.



$n = 3$

$n = 3$: Each of the four triangles from $n = 2$ gets divided into two triangles, so we get eight triangles. Two sides of the triangle are of length 1 (radius of the unit circle) and its height is $\frac{\sqrt{2}}{2}\frac{1}{2} \cdot 1 \cdot 8 = 2\sqrt{2} \approx 2.828 = z_3$.

(d) We can stabilize the algorithm by manipulating the formula:

$$
\begin{aligned}
z_{n+1} &= 2^{n-\frac{1}{2}}\sqrt{1-\sqrt{1-4^{1-n}z_n^2}} \\
&= \frac{2^{n-\frac{1}{2}}\sqrt{1-\sqrt{1-4^{1-n}z_n^2}}\cdot\sqrt{1+\sqrt{1-4^{1-n}z_n^2}}}{\sqrt{1+\sqrt{1-4^{1-n}z_n^2}}} \\
&= 2^{n-\frac{1}{2}}\sqrt{\frac{\sqrt{1-(1-4^{1-n}z_n^2)}}{1+\sqrt{1-4^{1-n}z_n^2}}} \\
&= 2^{n-\frac{1}{2}}\sqrt{\frac{\sqrt{4^{1-n}z_n^2}}{1+\sqrt{1-4^{1-n}z_n^2}}} \\
&= \frac{2^{n-\frac{1}{2}}z_n\sqrt{4^{1-n}}}{\sqrt{1+\sqrt{1-4^{1-n}z_n^2}}} \\
&= \frac{2^{n-\frac{1}{2}}z_n 2^{1-n}}{\sqrt{1+\sqrt{1-4^{1-n}z_n^2}}} \\
&= \frac{\sqrt{2}z_n}{\sqrt{1+\sqrt{1-4^{1-n}z_n^2}}}
\end{aligned}
$$

This gives us a more stable algorithm, as can be seen in Figure 2.
The implementation of the algorithm is in file `p6_2.m`. A copy follows:

```
start=2;
stop=40;
z2=[1:stop];
z2(2)=2;
err2=[1:stop];

function out = iter2(z,n)
        out=sqrt(2)*z/sqrt(1+sqrt(1- (4^(1-n)) * (z^2) ));
end

for n=start:stop-1
        z2(n+1)=iter2(z2(n),n);
endfor

ideal=ones(1,stop).*pi;
err2=abs(ideal.-z2);
n=1:stop;
gset term postscript
gset output "p6_fig2.ps"
semilogy(n(start:stop),err2(start:stop))
gset term x11
```

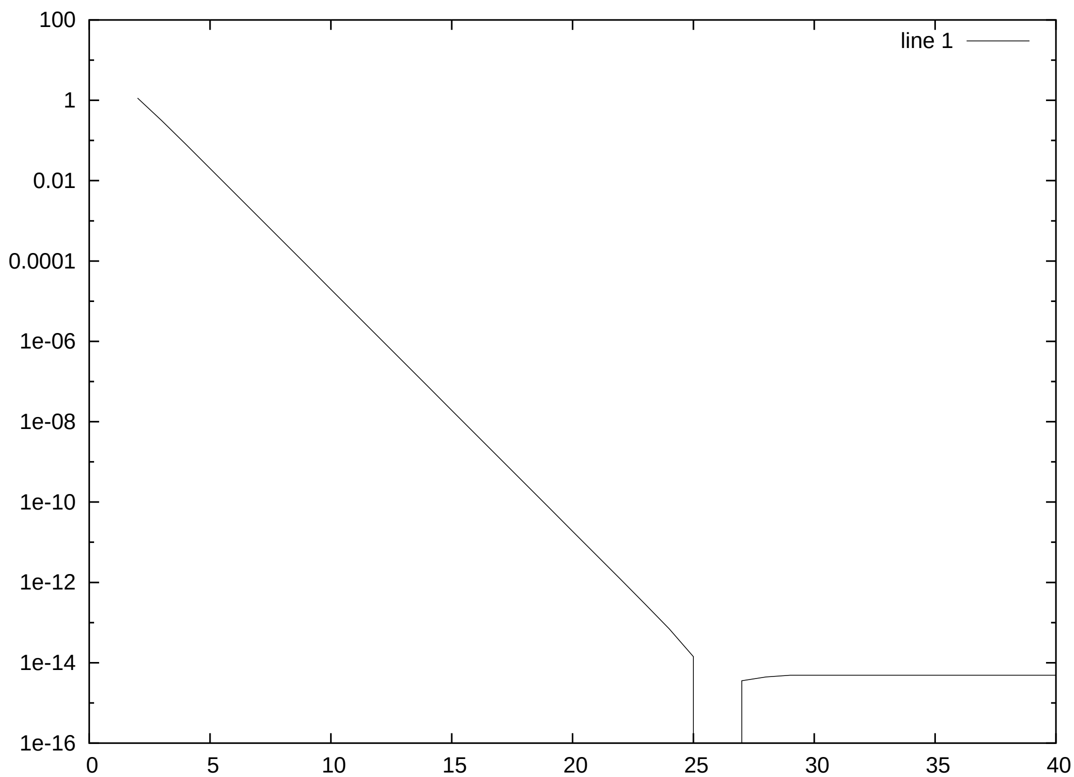


Figure 2: errors vs. n for stable method