MATH 501 HW2

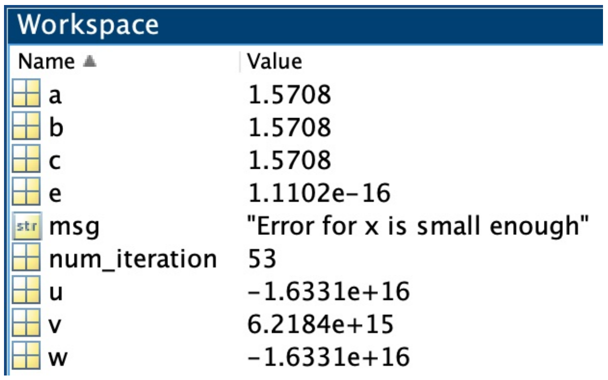
Qilin Ye

February 19, 2021

Pseudocode Programming, p.36

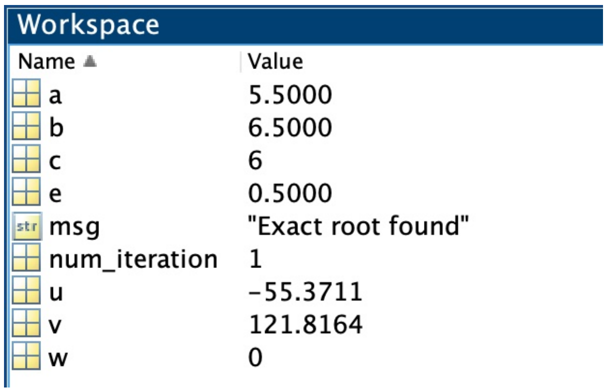
```
a = *; %left endpoint
b = *; %right endpoint

u = f(a);
v = f(b);
e = b - a;
if (sign(u)==sign(v))
    error_msg = "Same sign on endpoints";
elseif (sign(u)~=sign(v)) && (sign(u)+sign(v)~=0)
    error_msg = "Zero at an endpoint";
else
    for num_iteration = 1:100 %setting the "M"
        e = e/2; %bisecting the interval
        c = a+e;
        w = f(c);
        if w==0
            msg = "Exact root found";
            break
        elseif abs(e)<eps
            msg = "Error for x is small enough";
            break
        elseif abs(w)<eps
            msg = "Error for f is small enough";
            break
        else %continue iterating
            if (sign(w) ~= sign(u)) %taking the left-half
               b = c;
               v = w;
            else %taking the right-half
            a = c;
            u = w;
            end
        end
    end
end

function y = f(x)
    y = *; %input function here
end
```

Output for Ex.3.1.19:



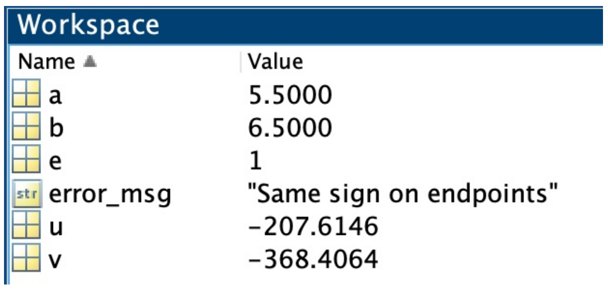
```
Workspace
Name ▲          Value
a               1.5708
b               1.5708
c               1.5708
e               1.1102e-16
msg             "Error for x is small enough"
num_iteration   53
u               -1.6331e+16
v               6.2184e+15
w               -1.6331e+16
```

Output for Ex.3.1.20(1):



```
Workspace
Name ▲          Value
a               5.5000
b               6.5000
c               6
e               0.5000
msg             "Exact root found"
num_iteration   1
u               -55.3711
v               121.8164
w               0
```

Output for Ex.3.1.20(2):



```
Workspace
Name ▲          Value
a               5.5000
b               6.5000
e               1
error_msg       "Same sign on endpoints"
u               -207.6146
v               -368.4064
```

Problems from Textbook

Ex.2.3.1 Find analytically the solution of this difference equation with the given initial values:

$$x_0 = 1, x_1 = 0.9, \text{ and } x_{n+1} = -0.2x_n + 0.99x_{n-1}.$$

Solution

This question is equivalent to solving the equation

$$\begin{bmatrix} -0.2 & 0.99 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_n \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix}.$$

This 2-by-2 matrix has eigenvalues -1.1 and 0.9. The general solution is of form $x_n = (-1.1)^n A + 0.9^n B$. Substituting the initial conditions $x_0 = 1$ and $x_1 = 0.9$ in, we have

$$\begin{cases} A + B = 1 \\ -1.1A + 0.9B = 0.9 \end{cases} \implies \begin{cases} A = 0 \\ B = 1 \end{cases} \implies x_n = 0.9^n.$$

This computation will be stable since $|0.9| < 1$ and the original error decays exponentially.

Ex.2.3.6(a)(c)(e) What are the condition numbers of (a) $(x-1)^\alpha$, (c) $\sin x$, and (e) $x^{-1}e^x$? When are they large?

Solution

Recall the definition where the red term is the conditional number:

$$\frac{f(x+h) - f(x)}{f(x)} \approx \frac{hf'(x)}{f(x)} = \frac{xf'(x)}{f(x)} \frac{h}{x}.$$

For (a), the conditional numbers is

$$\frac{x\alpha(x-1)^{\alpha-1}}{(x-1)^\alpha} = \frac{\alpha x}{x-1}.$$

We see that $\lim_{x\to\infty} \alpha x/(x-1) = \lim_{x\to-\infty} \alpha x/(x-1) = 1$. However this numbers gets large as $x \to 1$.

For (e), the conditional number is

$$\frac{x\cos(x)}{\sin(x)} = \frac{x}{\tan(x)}.$$

Notice that the conditional number does not tend to $\pm\infty$ when $|x| \to 0$. Besides this special case, the conditional number $\to \infty$ when $|x| \uparrow k\pi$ (and $\to -\infty$ when $|x| \downarrow k\pi$), where k is any positive integer.

For (e), the conditional number is

$$\frac{x \cdot e^x(x^{-1} - x^{-2})}{x^{-1}e^x} = x - 1.$$

One immediately sees that this conditional number blows up as $x \to \infty$.

Ex.2.3.7 Consider the example in the text $y_{n+1} = e - (n+1)y_n$. How many decimals of accuracy should be used in computing $y_1, y_2, \ldots, y_{20}$ if y_{20} is to be accurate to five decimals?

Solution

Suppose we start with $y0$ and the error caused by computing y_1 is δ. Recall that when computing $(n+1)y_n$, the error of y_n propagates by a factor of $n+1$. Therefore the accumulated error when we compute y_{20} is $20!\delta$. To ensure five-decimal accuracy we need to make sure the error $< 5 \cdot 10^{-6}$. Therefore $\delta < 5 \cdot 10^{-6}/20! \approx 2.06 \cdot 10^{-24} \in (2^{-78}, 2^{-78})$. Therefore we need 25 digits in a decimal system or 79 digits in a binary system to ensure y_{20} is accurate to five decimals.

Ex.2.3.9 Show that the recurrence relation

$$x_n = 2x_{n-1} + x_{n-2}$$

has a general solution of the form

$$x_n = A\lambda^n + B\mu^n.$$

Is the recurrence relation a good way to compute x_n from all initial values x_0 and x_1?

Solution

Again we write this equation in matrix form:

$$\begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{n-1} \\ x_{n-2} \end{bmatrix} = \begin{bmatrix} x_n \\ x_{n-1} \end{bmatrix}.$$

It follows that the 2-by-2 matrix has eigenvalues $1 \pm \sqrt{2}$ and so the general solution is of form

$$x_n = A(1+\sqrt{2})^n + B(1-\sqrt{2})^n.$$

This algorithm isn't always stable. For example consider $x_0 = x_1 = 2$. We see that $A = B = 1$ in this case, but then $(1+\sqrt{2})^n$ blows up, and so will the error.

Ex.3.1.3 In the bisection, an interval $[a_{n-1}, b_{n-1}]$ is divided into half, and one of these halves is chosen for the next subinterval. Define $d_n = 0$ if $[a_n, b_n]$ is the left half, and $d_n = 1$ otherwise. Express the root determined by the algorithm in terms of the sequence $d_1, d_2, \ldots$.

Solution

If an exact root $c \in [a, b]$ is found, then the binary number $0.d_1d_2\ldots$ is the binary expression of $(c-a)/(b-a)$. Otherwise, i.e., if the algorithm stops after hitting one of the bounds, the binary number $0.d_1d_2\ldots$ is an approximation of the $(c'-a)/(b-a)$ where c' is the actual root, with absolute error not exceeding 2^{-M-1}, where M is the number of iterations operated.

Ex.3.1.5 Give an example (or disprove) of a sequence $\{a_n\}$ of left endpoints from this bisection method in which $a_0 < a_1 < a_2 < \ldots$.

Solution

If $\{a_n\}$ is finite with a total of m terms, then we simply need to pick a root within $(b-k, b)$ where $k = 2^{-m}$. If $\{a_n\}$ is infinite, no such sequence exists. WLOG assume $a = 0$ and $b = 1$. Since $a_0 < a_1$ we know $a_1 = 1/2$. Similarly we know $a_2 = 3/4$ and $a_n = 1 - 1/2^n$. Since

$$\sum_{k=0}^{\infty} a_{k+1} - a_k = \sum_{k=1}^{\infty} 2^{-k} = 1,$$

we must have the root at b. But then the algorithm would have stopped before even reaching the loop, contradiction. Therefore no infinite $\{a_n\}$ can be strictly increasing.

Ex.3.1.11 Consider the bisection method starting with the interval $[1.5, 3.5]$.

(a) What is the width of the interval at the n^{th} step of this method? $\boxed{2^{-n+1}.}$

(b) What is the maximum distance possible between the root r and the midpoint of this interval? $\boxed{2^{-n}.}$ (Here I assumed that *this interval* refers to the one mentioned in the previous part. If it actually refers to the original interval $[1.5, 3.5]$ then the answer is simply 1, half the length of $[1.5, 3.5]$, obtained when $r \in \{1.5, 3.5\}$.)

Ex.3.1.12 Let the bisection method be applied to a continuous function, resulting in intervals $[a_n, b_n]$. Let $r := \lim_{n\to\infty} a_n$. Which of the following can be false?

(a) $a_0 \leqslant a_1 \leqslant \dots$. *True by definition.*

(b) $|r - 2^{-1}(a_n + b_n)| \leqslant 2^{-n}(b_0 - a_0)$. *True by the theorem in text.*

(c) $|r - 2^{-1}(a_{n+1} + b_{n+1})| \leqslant |r - 2^{-1}(a_n + b_n)|, n \geqslant 0$. **Can be false**. Consider $r = 0.6, [a, b] = [0, 1]$, and $n = 0$. Then $|r - 2^{-1}(a_n - b_n)| = 0.6 - 0.5 = 0.1$ but $|r - 2^{-1(a_{n+1}-b_{n+1})}| = 0.75 - 0.6 = 0.15$.

(d) $|r - a_n| = \mathcal{O}(2^{-n})$ as $n \to \infty$. True; the claim follows from (b).

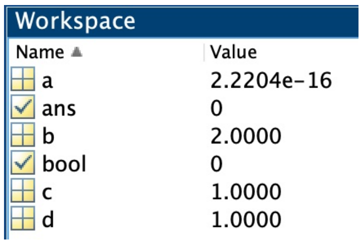
(e) $|r - c_n| < |r - c_{n-1}|, n \geqslant 1$. **Can be false**. See counterexample to (c).

Ex.3.1.16 On your computer, find numbers a and b such that $(a + b)/2$ and $a + 0.5(b - a)$ produce different results. Assume $a < b$ and don't use overflow or underflow.

Solution

Consider the following code: the result `bool=0` means false.

```
a = eps;
b = 2 + 2*eps;
c1 = (a+b) / 2;
c2 = a + 0.5 * (b-a);
bool = (c == d);
```



Workspace

Name	Value
a	2.2204e-16
ans	0
b	2.0000
bool	0
c	1.0000
d	1.0000