

UNIVERSITY OF TECHNOLOGY SYDNEY
School of Mathematical and Physical Sciences
35007 Real Analysis

COMPUTER LABORATORY

Goals: The goals of the computer laboratory are:

- Demonstrate the usefulness of numerical computer programming in solving and checking real analysis problems.
- Reinforce some key real analysis concepts via movie-like illustrations.
- Explain the Weierstrass Approximation Theorem (Theorem 5.17 in the subject notes) via movie-like illustrations. There is an assessment task associated with this part of the computer laboratory.

1. Downloading Required Files

- (a) Decide on a directory/folder in which you will store the files for this laboratory. It is best if you put them all in the same directory/folder. A simple option is to put them on the `Desktop` directory/folder, but please take note of where you store them for later.
- (b) Open a web browser and go to the web-page:
`http://matt-p-wand.net/35007.html`
- (c) Scroll down to the heading **File for the Computer Laboratory** and download the ZIP archive file named
`CompLabFiles.zip`
into a suitable directory/folder on your laptop. Take note of this location, since it is relevant for later steps involving the R language.
- (d) Open, and possibly unzip, `CompLabFiles.zip`. This should produce eleven text files with names:

<code>testRscript.R</code>	<code>Week5DerivChk.R</code>	<code>sinSeries.R</code>
<code>regionConverge1.R</code>	<code>regionConverge2.R</code>	<code>wavOrds.txt</code>
<code>RMIexamps.R</code>	<code>BernsteinPolyn.r</code>	<code>WATexamp1.R</code>
<code>WATexamp2.R</code>	<code>draw.axes.r</code>	
- (e) The remainder of this laboratory assumes that the filenames are **exactly** as listed here. If the downloading process corrupts the filenames in any way then you need to change the filenames to be these.

2. Making Sure that File Locations Match the Current Working Directory of Your R Session

If you already know about this topic then please skip this part of the computer laboratory.

Start an R session and issue the following command:

`getwd()`

If all files associated with this laboratory are stored in this directory then all should be fine.

Suppose that the files are in a different directory such as `C:\UTSwork\reaAnaWork`. The current working directory of your R session can be changed using either:

- a graphical user interface approach – described in the first three boxes below – but which are difficult to guarantee given the various types of interfaces hanging around and subject to change,
- the `setwd()` command – described in the last box below – which always works regardless of the interface to R.

Macintosh (Mac) Graphical User Interface Solution

If you are using R's graphical user interface on a Macintosh (Mac) laptop then there is a good chance that the following steps can be used to change the current working directory:

- In the menu at the top click on Misc.
- Then choose the menu item Change working directory...
- Select the directory where the computer laboratory files reside.

Windows Graphical User Interface Solution

If you are using R's graphical user interface on a Windows laptop then there is a good chance that the following steps can be used to change the current working directory:

- In the menu at the top click on File.
- Then choose the menu item Change dir...
- Select the directory where the computer laboratory files reside.

R Studio Graphical User Interface Solution

If you are accessing R via the R Studio graphical user interface then there is a good chance that the following steps can be used to change the current working directory:

- In the menu at the top click on **Session**.
- Then choose the menu item **Set Working Directory...**
- Then choose the menu item **Choose Directory...**
- Select the directory where the computer laboratory files reside.

Command Line Solution

Enter the command (**with forward slashes (/) rather than backward slashes (\)**) as illustrated by:

```
setwd("C:/UTSwork/reaAnaWork")
```

3. Running R via scripts

When several R (or, possibly, several thousand) R commands need to be run then it is better to use a *script*, which is simply a text file containing the commands.

- (a) Open an R session and make sure that the current working directory corresponds to the directory/folder in which `testRscript.R` resides (i.e. the directory/folder where you downloaded the files at the start of the laboratory).

- (b) Inside R, issue the command `source("testRscript.R")`

This will run the several R commands listed in the file `testRscript.R`

- (c) If successful, the previous step should lead to a graphic with lots of coloured circles. If this doesn't happen then please ask for help from the lecturer.

4. Command Line Use of R

- (a) Commence an R session.

- (b) Enter the command `413*126` and hit `return` or `enter`. This should lead to the output

```
[1] 52038
```

which, as expected, is equal to 413×126 .

- (c) **IMPORTANT: Quick command recall**

Suppose that, for the sake of illustration, the new multiplication 413×127 is of interest. Then rather than typing `413*127` hit the up-arrow key ($\uparrow$). This will lead to the previous command `413*126` being recalled. Change the 6 to a 7 and hit `return` or `enter`. If all is well then you will get the output:

```
[1] 52451
```

This trick is very useful when using R in command line mode.

(d) IMPORTANT: Ending (or “quitting”) an R session

To end this short R session enter the command `q()` and hit `return` or `enter`. In response to the query

```
Save workspace image? [y/n/c]:
```

type `n` and hit `return` or `enter`.

5. Computer Checks for L'Hôpital's Rule Limit Problems

This segment demonstrates how relatively simple computer checks can be done to aid L'Hôpital's Rule limit problems. The examples that we consider here are the limit problems in Question 7(c) of the assignment, Exercise H.15 and Exercise H.14:

$$[\mathbf{A}] \lim_{x \rightarrow \infty} \left(\frac{1}{x} - \frac{1}{e^x - 1} \right), \quad [\mathbf{B}] \lim_{x \rightarrow 0^+} \cot(x)^{\sin(x)} \quad \text{and} \quad [\mathbf{C}] \lim_{x \rightarrow 0} \frac{1 + \frac{1}{2}x^2 - \sqrt{1 + x^2}}{(\cos(x) - e^{x^2}) \sin(x^2)}.$$

(a) Commence an R session.

(b) Issue the following commands, noting that the command containing "l" corresponds lower case L for line-type plotting. Be sure to hit `return` or `enter` after each command.

```
xAg <- seq(1,1000,length=1001)
yAg <- 1/xAg - 1/(exp(xAg) - 1)
plot(xAg,yAg,type="l")
abline(0,0,col="red")
```

Note that `abline(0,0,col="red")` draws a red line with intercept $a = 0$ and slope $b = 0$. The resulting plot suggests that

$$\frac{1}{x} - \frac{1}{e^x - 1} \rightarrow 0 \quad \text{as} \quad x \rightarrow \infty,$$

which matches the answer for [A] from repeated application of L'Hôpital's Rule.

(c) We now turn attention to limit problem [B]. Recall that, by definition, $\cot(x) = 1/\tan(x)$. Issue the commands:

```
xBg <- seq(0.001,1,length=1001)
yBg <- (1/tan(xBg))^sin(xBg)
plot(xBg,yBg,type="l")
abline(1,0,col="red")
```

Note that `abline(1, 0, col="red")` draws a red line with intercept $a = 1$ and slope $b = 0$. The resulting plot suggests that

$$\cot(x)^{\sin(x)} \rightarrow 1 \quad \text{as } x \rightarrow 0^+,$$

which matches the answer for [B] from repeated application of L'Hôpital's Rule after playing some games with exponentials and logarithms.

- (d) We conclude this segment with some useful graphics for limit problem [C]. Issue the commands:

```
xCg <- seq(-1.5, 1.5, length=1001)
numerg <- 1 + 0.5*xCg^2 - sqrt(1 + xCg^2)
denomg <- (cos(xCg) - exp(xCg^2))*sin(xCg^2)
yCg <- numerg/denomg
plot(xCg, yCg, type="l")
print(min(yCg[!is.na(yCg)]))
```

The last command is concerned with finding the minimum value along the curve in the vertical direction. Often the simpler command `print(min(yCg))` would be sufficient. In this case the more complicated command `print(min(yCg[!is.na(yCg)]))` is appropriate since it allows for the “not a number” designation used by R when it encounters division of 0 by 0.

- (e) The number printed out by the last command is a decimal approximation of $-1/12$. Issue the command:

```
abline((-1/12), 0, col="red")
```

- (f) The final graphics is in keeping with the fact that limit problem [C] has the answer $-1/12$. As shown in the solution for Exercise H.14., getting this answer takes about $1\frac{1}{2}$ pages of calculus and algebra, including four applications of L'Hôpital's Rule. The above few lines of code and ensuing graphics can be worthwhile since it indicates that the limit exists and is finite. This implies that repeated applications of L'Hôpital's Rule will eventually lead to the limit being obtained mathematically.

Even though the above numerical computations and graphics indicate the solutions for limit problems [A], [B] and [C], they do not constitute proofs of the limit results. For example, the graphics for [C] may suggest an answer of $-1/12$, but other fractions have similar decimal forms and perhaps the actual answer is

$$-373/4482 = -0.08322\dots$$

Mathematical arguments involving L'Hôpital's Rule are still required to establish the limit results.

6. Use of Numerical Derivatives to Check Derivative Expressions

In this segment we introduce the concept of numerical derivatives and demonstrate how they can be used to check expressions for derivatives of functions of interest. Whilst this concept has been used by the lecturer for many years (going back to before existence of the Internet), it has started to be partially superseded by web-sites that support symbolic differentiation. Nevertheless, it is worth learning about this approach for situations where use of web-sites is not allowed or not available.

- (a) The first example involves differentiation of the function

$$f(x) = \frac{e^x}{(1 + e^x)^2}.$$

Suppose that a first attempt at obtaining the first derivative expression led to the wrong answer:

$$f'_{\text{wrong}}(x) = \frac{e^x(2 - e^x)}{(e^x + 1)^3}.$$

- (b) Start an R session and issue the following commands. Hit `return` or `enter` after each command.

```
ng <- 101 ; xg <- seq(-3, 3, length=ng)
eg <- exp(xg) ; fg <- eg / (1 + eg)^2
exprfdg <- eg * (2 - eg) / (eg + 1)^3
h <- xg[2] - xg[1]
numefdg <- (fg[2:ng] - fg[1:(ng-1)]) / h
plot(xg, exprfdg, col="deepskyblue", type="l", lwd=6)
lines(xg[2:ng], numefdg, col="darkblue", lwd=2)
```

Assuming correct typing, the plot shows the exact, but incorrect, derivative in lighter blue and the numerical derivative in darker blue. The numerical derivative code is based on the expression

$$\frac{f(x + h) - f(x)}{h}$$

and computes this function for h set to the gap between the plotting grid points, which is 0.06 in this case. Since h is relatively small, and noting the limit definition of a derivative, the `numefdg` vector is a good approximation to the actual derivative. These are often called *numerical derivatives* and a relatively quick way of checking derivative expressions.

In this case the curves based on the derivative expression and the numerical derivatives do not match, which indicates a calculus/algebra error.

- (c) Further calculus and algebra leads to the following updated derivative expression

$$f'(x) = \frac{e^x(1 - e^x)}{(e^x + 1)^3}.$$

Issue the following commands to achieve graphical checking verification that this is the correct answer:

```
exprfdg <- eg * (1 - eg) / (eg + 1)^3
plot(xg, exprfdg, col="deepskyblue", type="l", lwd=6)
lines(xg[2:ng], numefdg, col="darkblue", lwd=2)
```

- (d) The remaining example concerns the following function from the Week 5 slide presentation titled *Differentiation and Real Analysis*. There, the main example function is

$$f(x) = 7e^{-|x+2|} + 2xe^{-|x-5|}, \quad x \in \mathbb{R}$$

and its derivative is derived to be for $x \notin \{-2, 5\}$:

$$f'(x) = \begin{cases} 7e^{x+2} + 2(1+x)e^{x-5}, & x < -2, \\ -7e^{-x-2} + 2(1+x)e^{x-5}, & -2 < x < 5, \\ -7e^{-x-2} + 2(1-x)e^{5-x}, & x > 5 \end{cases}$$

Continue or commence an R session for which all downloaded files are in the same current working directory of the session and issue the command `source("Week5DerivChk.R")`. The graphic produced by the R script `Week5DerivChk.R` is a numerical derivatives check of the above f' expression (which was used by the lecturer to check the correctness of the Week 5 slides). Either now or later the code could be read to see how the graphic is produced. A curious part of the code is

```
ng <- 1001
xg <- seq(-6-1/sqrt(pi), 6+2/sqrt(pi), length=ng)
```

which sets up an x -axis plotting grid of length 1001 between

$$-6 - \frac{1}{\sqrt{\pi}} \quad \text{and} \quad 6 + \frac{2}{\sqrt{\pi}}.$$

The reason for these unusual (irrational number) endpoints is to avoid the points of non-differentiability at $x = -2$ and $x = 5$ where the R function evaluations crash.

7. Visualisation of Power Series Convergence and Region of Convergence

This segment reinforces the concepts of power series convergence and region of convergence via computer graphics and visualisation. It involves running some scripts, written in the R computer programming language, which produce movie-type displays of these key real analysis concepts.

- Commence an R session for which all downloaded files are in the same current working directory of the session.
- Issue the command `source("sinSeries.R")` and position the windows so that the graphics window is relatively large and the command window is relatively small and narrow.

By hitting `return` or `enter` when prompted, the partial sums

$$\sum_{k=0}^n \frac{(-1)^k x^{2k+1}}{(2k+1)!} \quad \text{for} \quad -25 \leq x \leq 25$$

are shown for increasing n . The graphics shows the power series converging to the familiar sine curve, which is in keeping with the definition

$$\sin(x) = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!}.$$

(c) One of the Week 5 lectures showed that the power series

$$\lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{\sqrt{7k^2 + 2k + 38} (x - 9)^k}{4^k}.$$

is convergent for $5 < x < 13$. Issue the command

`source("regionConverge1.R")` to run a movie which shows the sequential convergence of this power series. In particular, the fact that it is only convergent over the open interval $(5, 13)$ is apparent.

(d) The last power series example involves the power series

$$\lim_{n \rightarrow \infty} \sum_{k=0}^n (2807 + 403k - 120k^2 + 3k^3)x^k \quad (1)$$

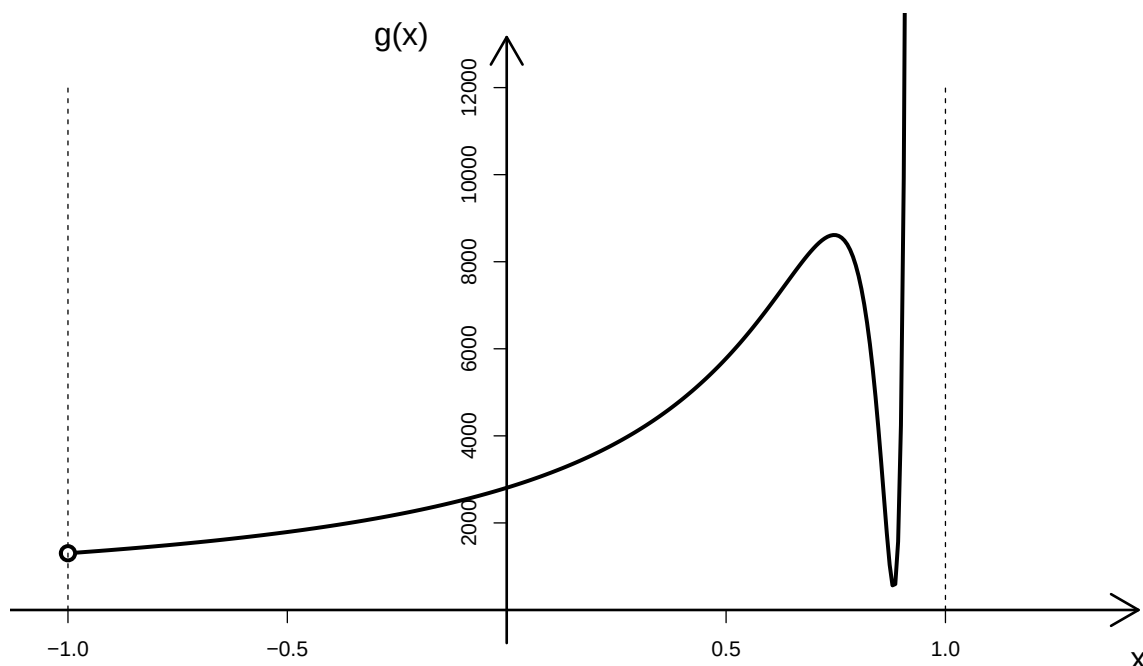
which has region of convergence $-1 < x < 1$. To run a movie which shows the sequential convergence of this power series, issue the command

`source("regionConverge2.R")`.

For this power series, steps similar to those used in Exercises I.9, I.10 and I.11 and Question 8 of the assignment can be used to show that the power series in (1) converges to the function $g : \{x \in \mathbb{R} : x \neq 1\} \rightarrow \mathbb{R}$ given by

$$g(x) = \frac{2807(1-x)^3 + 403x(x-1)^2 + 120x(x^2-1) + 3x(x^2+4x+1)}{(x-1)^4}.$$

The movie produced by `regionConverge2.R` gives the impression that the power series converges to a function with asymptotes at $x = -1$ and $x = 1$. In fact, the only asymptote is at $x = 1$. For x just to the right of -1 , that is, for $x = -1 + \xi$ for $\xi > 0$ and arbitrarily small, the series is convergent but in an oscillatory way. An appendix to this computer laboratory – probably best read later if of interest to you – explains this phenomenon in more detail. The following figure shows the power series convergent function with the asymptote at $x = 1$. Near $x = -1$ there is no asymptote and g is continuous at $x = -1$.



8. Reinforcement of Riemann Integration Principles

This segment use a set of examples and movie-type computer graphics to reinforce the principles of Riemann integration. Each example involves an integrand function f over the interval $[0, 1]$. The family arithmetic partitions of $[0, 1]$

$$\mathcal{P} = \left\{ 0, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n-1}{n}, 1 \right\}, \quad n \in \mathbb{N}$$

is used throughout this segment. The values of $L(f, \mathcal{P})$, $U(f, \mathcal{P})$ and the Riemann's Criterion quantity

$$U(f, \mathcal{P}) - L(f, \mathcal{P})$$

are highlighted.

- (a) Commence an R session for which all downloaded files are in the same current working directory of the session.
- (b) Issue the command `source("RMIexamples.R")` and position the windows so that the graphics window is relatively large and the command window is relatively small and narrow. The graphics window should be wide enough to read all lines of text in the first graphic.
- (c) Repeated hits on the `return` or `enter` key run movie-like computer graphics that show, for increasing values of

$n = \text{number of Riemann integration rectangles}$

the values of $L(f, \mathcal{P})$, $U(f, \mathcal{P})$ and their difference

$$U(f, \mathcal{P}) - L(f, \mathcal{P}).$$

In each example it is apparent that $U(f, \mathcal{P}) - L(f, \mathcal{P})$ can be made arbitrarily small by increasing the partition size.

In each case Riemann's Criterion appears to hold and the function is Riemann integrable.

9. The Weierstrass Approximation Theorem

This segment is concerned with the Weierstrass Approximation Theorem (Theorem 5.17) which states that if f is a continuous function on the bounded interval $[a, b]$ then for any given $\varepsilon > 0$ there is a polynomial P with the property that

$$\sup\{|f(x) - P(x)| : a \leq x \leq b\} < \varepsilon.$$

In other words, given $\varepsilon > 0$ a polynomial P of degree n can be found that is within ε of f uniformly over $[a, b]$.

This part of the computer laboratory uses examples and computer graphics to illustrate the Weierstrass Approximation Theorem. It involves running some scripts, written in the R computer programming language, which produce movie-type displays of this real analysis concept.

- (a) Commence an R session for which all downloaded files are in the same current working directory of the session.

- (b) Issue the command `source("WATexampl.R")` and position the windows so that the graphics window is relatively large and the command window is relatively small and narrow. The code in the R script `WATexampl.R` is concerned with polynomial approximation of the function $f_1 : [0, 1] \rightarrow \mathbb{R}$ given by

$$f_1(x) = \cos(4\pi x^3) + \sin(\pi x) + 1.5x, \quad 0 \leq x \leq 1.$$

- (c) Hold down the `return` or `enter` to see the (Bernstein) approximating polynomial of degree n for each of $n = 1, 2, \dots, 1000$. For each n the text at the top of the plot reports the value of

$$\sup\{|f_1(x) - P_n(x)| : 0 \leq x \leq 1\} = \text{"maximum absolute vertical gap"}$$

where P_n is the appropriate Bernstein polynomial of degree n . Notice that the maximum absolute vertical gap decreases as the polynomial degree increases. This is in keeping with the Weierstrass Approximation Theorem which says that, for any $\varepsilon > 0$, there is a polynomial degree n such that the maximum absolute vertical gap is below ε . **It is anticipated that it will take about 80 seconds to run the full "movie" to $n = 1000$ with the key held down. If you get the point for a lower n then you may not want to go all the way to 1000 and kill the session using e.g. `control-C` (assuming your R set-up easily allows this – otherwise you may be stuck with the full version!).**

- (d) Note that `source("WATexampl.R")` command can be repeated to run this example as many times as you would like to help see the Weierstrass Approximation Theorem in action.
- (e) A somewhat pedantic remark is that scripts for this segment are not computing the supremum quantities exactly but instead using an accurate fine-grid approximation. This minor detail will be ignored for the remainder of the segment.
- (f) When you are finished running the first example involving f_1 issue the command `source("WATexam2.R")` to run a second example, this time for the function

$$f_2(x; \nu_4) = \exp\left(-|(x/\nu_4) - 1|\right) + \cos(8\pi\nu_4 x^2) + 2, \quad 0 \leq x \leq 1. \quad (2)$$

where, to 12 decimal places, $\nu_4 = 0.226757367347$.

- (g) Copy the file `WATexam2.R` to a new file named `myWATexam2.R`. Open the file `myWATexam2.R` in an editor and replace the character string `22222222` about 10-15 lines from the top of the file by your student number, save the file and exit from the editor.
- (h) In an R session issue the command `source("myWATexam2.R")` to visualise the Weierstrass Approximation Theorem for a version of the family of functions defined by (2), but with the value of ν_4 determined by your student number.
- (i) If you are now wondering what degree of polynomial is needed to approximate your $f_2(\cdot; \nu_4)$ with a maximum vertical gap corresponding to half of the reciprocal of the treble iterated square-root of your student number then you're in luck – since this is the computer laboratory's assessment task.

The computer laboratory continues overleaf.



ASSESSMENT TASK:

- Define

$$\varepsilon_S = \frac{1}{2\sqrt{\sqrt{\sqrt{S}}}} = \frac{1}{2S^{1/8}} \quad \text{where } S \text{ is your student number.}$$

- Using the R script `myWATexamp2.R`, determine the smallest n such that

$$\sup\{|f_2(x; \nu_4) - P_n(x)| : 0 \leq x \leq 1\} < \varepsilon_S.$$

You are not required to do any R programming for this task and one or more runs of `myWATexamp2.R` is sufficient to determine the required n value.

- Between now and the assessment task deadline of

2:50pm Wednesday 14th October 2026

send an e-mail message to `matt.wand@uts.edu.au` with the subject or body of the e-mail message containing a sentence of the form

My value of `n` is 999.

where the 999, which is just a place-holder here, is replaced by your value of n .

Appendix: Some Further Details Regarding the Last Power Series Example

For each $-1 < x < 1$ let

$$g_n(x) = \lim_{n \rightarrow \infty} \sum_{k=0}^n (2807 + 403k - 120k^2 + 3k^3)x^k$$

and

$$g(x) = \frac{2807(1-x)^3 + 403x(x-1)^2 + 120x(x^2-1) + 3x(x^2+4x+1)}{(x-1)^4}.$$

Then results in the subject material allow us to assert that

$$\lim_{n \rightarrow \infty} g_n(x) = g(x) \quad \text{for each } -1 < x < 1.$$

However, for x close to -1 this convergence can be very slow. From the ε - N definition of a sequence limit with $\varepsilon = 1$, there must be $N \in \mathbb{N}$ such that

$$n \geq N \quad \text{implies} \quad |g_n(x) - g(x)| < 1.$$

Based on some computer coding, it can be shown that for $x = -0.9$ we need to have N at least equal to 142. In other words at least 143 terms are needed in the series to be within 1 of the limit.

For $x = -0.99$ we need at least 2,357 terms.

For $x = -0.999$ we need at least 31,459 terms.

For $x = -0.9999$ we need at least 389,660 terms.

The figure below shows these behaviours of the $g_n(x)$ series for $x \in \{-0.9, -0.99, -0.999, -0.9999\}$.

