Department of Mathematics and Statistics
620–295
Real analysis with applications

Before starting, copy the folder `Lab4` from the lab server `M&S Lab Materials\620-295` to `D:MATLAB` and set the path to `D:MATLAB` including subfolders.

Laboratory Class 3: Series, Taylor series and Fourier series

1 Series

Leibniz' Theorem: If (a_n) is a monotonic sequence with $a_n \to 0$, then the series $\sum_{n=1}^{\infty}(-1)^n a_n$ converges.

1.0.1 Exercise

Run `Lab4Ex1.m` . You should see a plot of the partial sums of a sequence that satisfies the conditions of Leibniz' Theorem. Notice that the even partial sums and the odd partial sums separately form monotonic, bounded (hence convergent) sequences.

This plot suggests (correctly) an proof of Leibniz' Test: (Homework — not for the Lab)

i. Show from the assumptions that the two subsequences of even and odd partial sums are monotonic and bounded (hence convergent) with limits S_1 and S_2.

ii. Since $a_n \to 0$ by assumption, prove that $S_1 = S_2$ and so the series converges.

Notice from the plot that at any stage the limit lies between the latest even partial sum and the latest odd partial sum. This suggests the *error bound for alternating series*:

Error of Alternating Sum: For a series satisfying the assumptions of Leibniz' Test, the error of the partial sum with n terms is bounded by the absolute value of the first neglected term.

$$|S_n - S| \leq |a_{n+1}| \tag{1}$$

Optional: prove the error bound in Eq. 1.

2 Taylor series

2.1 Taylor polynomials

Your demonstrator will show you the use of `taylortool.m`, a program that illustrates the approximation of functions by Taylor polynomials.

2.1.1 Exercise

Try the following examples. In each case, experiment with values of N to plot the Taylor polynomials $T_N(x)$ approximating the given function f. You can also use the GUI to change the function and the plotting interval (set to $[-\pi, \pi]$) by default).

```
taylortool('sin(x)')
taylortool('cos(x)')
```

What happens to the region where f is well-approximated by $T_N(x)$ as N increases? Can you explain why the Taylor polynomial only changes for every other value of N?

```
taylortool('1/(1+x^2)')
taylortool('log(1+x)')
```

What happens to the region where f is well-approximated by $T_N(x)$ as N increases? Why?

```
taylortool('exp(-x^2)')
```

```
taylortool('1/exp(1/x^2)')
```

What's going on?

2.2 An application to integration

2.2.1 Exercise

Using the Maclaurin series for $\exp(x)$, show that the integral $I = \int_0^1 \exp(-x^2)\ dx$ can be expressed as the alternating series

$$I = \sum_{n=0}^{\infty}(-1)^n \frac{1}{n!}\frac{1}{2n+1}$$

Modify `Lab4Ex1` to compute the partial sums S_N up to $N = 10$.

Note: see `help factorial` to learn how to compute $n!$ in Matlab.

2.2.2 Exercise

Use Eq. 1 to estimate the error in your estimated value of I. Run your modified program again with $N = 15$. The result should be consistent with your error estimate.

You can't go much further doing ordinary computing since you've reached the accuracy attainable using 'double precision numbers'. For more accuracy, you can use 'variable precision arithmetic' in a symbolic computing environment.

3 Fourier series

Your demonstrator will show you the use of `fsgui.m`, a program that illustrates the approximation of *periodic functions* by Fourier series.

3.1 Examples

3.1.1 Exercise

In the command window, enter `fsgui`. Enter the function $f(x) = \sin(x)$: (use the Function menu to enter `sin(x)`). The plot shows the partial sum S_N of the Fourier series

$$f(x) = a_0 + \sum_{k=1}^{\infty}(a_k \cos kt + b_k \sin kt) \tag{2}$$

with the top harmonic (i.e. value of N) controlled using right and left arrows or by editing the value. By editing the value, start from $N = 1$.

The default window shows the partial Fourier sum in the top pane and the error $|f(x) - S_N(x)|$ in the bottom pane. It might seem strange that the error is not zero when $N = 1$ — it's because the coefficients are computed numerically using a trapezoid approximation.

3.1.2 Exercise

Try the following functions: (use the Function menu to enter new functions — array operators are not required, so that `x^2` works fine)
$f(x) = \cos x$; $f(x) = \sin x \cos 6x$; $f(x) = \exp(\sin(x))$; $f(x) = \exp(-2x^2)$;
$f(x) = x^2$; $f(x) = |x|$; $f(x) = x$; $f(x) =$`sign(x)`
In each case, experiment with N to see how well the function is approximated and where the error is largest.

3.2 Fourier coefficients

A radio button allows you to see the Fourier coefficients (a_k, b_k) used to construct the Fourier sums.

For $f(x) = \sin(x)$, the function is a single sinusoid, so you should have $b_1 = 1$, with all other coefficients vanishing. Check whether this holds.

3.2.1 Exercise

For $f(x) = \cos x$ and $f(x) = \sin x \cos 6x$, check that the coefficients are what you expect.

In some of these cases, all the a_k or all the b_k vanish — explain why.

In some of these cases, the coefficients decay rapidly with k; in some cases, the decay is slow. Can you guess what influences the rate of decay of coefficients?

3.3 Ringing

For $f(x) = x$ and $f(x) =$ `sign(x)`, the partial sums are particularly inaccurate. What feature of f is causing this phenomenon?

For more information, look up 'Gibbs phenomenon' on Wikipedia.

3.4 Complex Fourier series

A beautifully symmetric form of the Fourier series of a function results from using complex exponentials, rather than sinusoids, as the basis functions.

3.4.1 Exercise

By using the identities

$$\cos kt = \frac{1}{2}(e^{ikt} + e^{-ikt}) \qquad \text{and} \qquad \sin kt = \frac{1}{2i}(e^{ikt} - ie^{-ikt})$$

show that Eq. 3 can be re-written as

$$f(t) = \sum_{-\infty}^{\infty} c_k e^{ikt} \tag{3}$$

with the *complex Fourier coefficients* c_k defined by

$$c_k = \begin{cases} \frac{1}{2}(a_k - ib_k), & \text{if } k \geq 1, \\ a_0, & \text{if } k = 0, \\ \frac{1}{2}(a_{-k} + ib_{-k}), & \text{if } k \leq -1, \end{cases} \qquad \text{so that} \quad c_k = \frac{1}{2\pi}\int_0^{2\pi} f(t)e^{-ikt}\, dt \tag{4}$$

Eqs. 3–4 are an alternative form for the Fourier series of f. The coefficients c_k are now complex numbers, so harder to visualize. Their modulus is given by

$$|c_k| = \frac{1}{2}(a_k^2 + b_k^2)^{1/2}, \qquad \text{for } k > 0,$$

This can be plotted in `fsgui` by using the third radio button.

3.5 Finally

When the function f is only defined at discrete values of t, a pair of equations similar to Eqs. 3–4 can be derived, and defines the *Discrete Fourier Transform (DFT)* of a discrete periodic function f. The DFT is an important tool in the fields of *signal processing* and *image processing.*