18.311 — MIT (Spring 2011)

Rodolfo R. Rosales (MIT, Math. Dept., 2-337, Cambridge, MA 02139).

April 15, 2011.

Problem Set # 05.

Due: Friday April 22.

Turn it in before 3:30 PM, in the box provided in Room 2-108.

IMPORTANT: The Regular the Special Problems must be **stapled in TWO SEPARATE packages,** each with your **FULL NAME** clearly spelled.

Contents

1 Regular Problems.

1.1 Statement: GBNS02 Scheme B. Backward differences for $u_t + u_x = 0$.

See the GBNS Good and Bad Numerical Schemes Problems.

1.2 Statement: vNSA02 Scheme B. Backward differences for $u_t + u_x = 0$.

See the vNSA von Neumann Stability Analysis Problems.

1.3 Statement: AENS02 Scheme B. Backward differences for $u_t + u_x = 0$.

See the AENS Associated Equation to a Numerical Scheme Problems.

1.4 Introduction to Computer Exercise in Fourier Series.

Generally, a 2π-periodic function $F = F(x)$ can be expressed in terms of its **Fourier Series:**

$$F(x) = \sum_{n=-\infty}^{\infty} F_n \, e^{i\,n\,x}, \tag{1.1}$$

where the n^{th} complex Fourier coefficient F_n is defined by

$$F_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} F(x)\, e^{-i\,n\,x}\, dx, \quad \text{where} \quad n = 0,\, \pm 1,\, \pm 2,\, \pm 3, \ldots \tag{1.2}$$

An alternative formulation, obtained upon using $e^{-i\,n\,x} = \cos(n\,x) - i\,\sin(n\,x)$, is given by:

$$F(x) = c_0 + \sum_{n=1}^{\infty} \left(c_n\, \cos(n\,x) + s_n\, \sin(n\,x) \right), \tag{1.3}$$

where: (i) $c_0 = F_0$ and $c_n = (F_n + F_{-n})$ are the cosine Fourier coefficients, and (ii) $s_n = i\,(F_n - F_{-n})$ are the sine Fourier coefficients. Thus

$$c_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} F(x)\, dx, \tag{1.4}$$

$$c_n = \frac{1}{\pi} \int_{-\pi}^{\pi} F(x)\, \cos(n\,x)\, dx, \quad \text{for: } n = 1,\, 2,\, 3,\, \ldots \tag{1.5}$$

$$s_n = \frac{1}{\pi} \int_{-\pi}^{\pi} F(x)\, \sin(n\,x)\, dx, \quad \text{for: } n = 1,\, 2,\, 3,\, \ldots \tag{1.6}$$

If F is real valued, then F_{-n} is the complex conjugate of F_n, so that

$$c_n = 2\,\mathsf{Re}\,(F_n) \quad \text{and} \quad s_n = -2\,\mathsf{Im}\,(F_n) \quad \text{for} \quad n > 0. \tag{1.7}$$

Generally, the issue of how well (or even in which sense), the Fourier Series in (1.1) or in (1.3) converges to the function F is a rather subtle one. *The main point of this problem is to conduct a numerical exploration of some aspects of this question.* In particular, consider the **partial sums:**

$$F_N(x) = c_0 + \sum_{n=1}^{N} \left(c_n \cos(n\,x) + s_n \sin(n\,x) \right), \tag{1.8}$$

where N is some natural number. Important questions are then: **How well does F_N approximate the function F?** and **How big is the error and how fast does it vanish as $N \to \infty$.**

Remark 1.1 *Obviously, an important element in answering the questions above is how fast the Fourier coefficients vanish as $n \to \infty$. This is determined by how fast the* **power spectrum**

$$P_n = \frac{1}{2}|F_n| = \sqrt{c_n^2 + s_n^2} \tag{1.9}$$

vanishes as $n \to \infty$ **(Assume that F is real valued, so that (1.7) applies).**
P_n gives information on "how important" the n-th mode is in the Fourier Series expansion. The name follows from the fact that (in many physical situations) one can interpret the square of the amplitude of the n-th Fourier coefficient (i.e.: P_n^2) as the amount of energy in the n-th mode of the solution (this is the case for the wave equation, for example).

1.5 Statement: Computer Exercise in Fourier Series.

This problem objective is to "experimentally" **study how Fourier series converge.** For this purpose you should use the following MatLab scripts

FouSerRedame.m   fourierSC.m   FSFun.m   FSoption.m   FSoptionP.m   heatSln.m

Put the scripts in a directory and start MatLab there. The help command will work as usual, in particular: help FouSerReadme gives a description of all the scripts. Each script has its own detailed description. The **script you need is** fourierSC. The others (except for heatSln) are helper scripts.

IMPORTANT:

- *When you start the script* fourierSC*, it will ask you first the questions:*
  A. *Do you want to use the fancy (with buttons) or the plain interface?*
  B. *Up to how many terms in the Fourier series do you want to compute?*
  C. *For which values you want to plot?*

  About **B** *and* **C***: Calculations will be done (and the results shown) for the partial sums in (1.8), for the values* N = 0, Nskip, 2*Nskip, ..., Np. *You will be asked to input* Nskip and Np.
- *After you finish answering the questions,* fourierSC *will present you with a list of options for functions whose Fourier series it can compute: "user's choice", and pre-selected.* **Check the scripts code to make sure you understand exactly which functions you are dealing with!**
- *The script* FSFun.m *is the one used to input the "user's choice" selection, whatever function you program there will be the one used when "user's choice" is selected. A trivial example is pre-programmed in* FSFun.m*, but* **you can alter it,** *and write there any function for which you want to investigate the Fourier series — this is so you can* **go beyond** *the preselected options.*
- The pre-selected options include smooth functions, as well as functions with various types of singular behaviors — discontinuities, corners and cusps. The idea is to investigate how any particular "singular" behavior in the function is related to the convergence properties of its Fourier series.

A **cusp** *is a singularity such as the one that* $\sqrt{|x|}$ *has at the origin. Other possibilities are* $|x|^\alpha$*, where* $0 < \alpha < 1$*. Investigate the effect of singularities of this type on the convergence!*

The most important singular behavior whose effect on the Fourier series you should elucidate is that of a discontinuity. How does it affect the convergence? How do the partial sums look like in this case? Is there any peculiar behavior you can observe?

Odd an even functions are also provided in the pre-selections, so that you can see what effect symmetries of the function have on its Fourier series. Can you think of other symmetries?

- *The script* fourierSC *makes lots of plots, which will be made one on top of the other. You will need to move the windows to see all the plots.* **These plots illustrate various aspects of how a Fourier series behaves, as follows** *(this is the order in which the plots are done):*
  - **Exact function whose Fourier series is being computed.**
  - **Sine Fourier coefficients s_n, as a function of n.**
  - **Cosine Fourier coefficients c_n, as a function of n.**
  - **Semi-log plot of the power spectrum $p_n = \sqrt{c_n^2 + s_n^2}$ as a function of n.**
    Exponential decay will give a straight line in this kind of plot.
  - **Log-log plot of the power spectrum as a function of n.**
    Algebraic decay will give a straight line in this kind of plot.
  - **Partial sums $F_N = F_N(x)$ — as in equation (1.8) — for** N = 0:Nskip:Np.
    All these plots will be shown in the same window,
    so you must look at them as they are done.
  - **Relative error in the approximation F_N, as a function of N.**
    Shows the error in the partial sums in (1.8), as a function of N, for N = 1:Np.
  - **Semi-log plot of the relative error, as a function of N.**
  - **Log-log plot of the relative error, as a function of N.**

This is what you should do:

Use the script fourierSC and report any "patterns" or peculiar behavior you observe in the way Fourier series converge. Experiment with the various choices. Look at the plots and think: what is happening? Many of the plots are useful in figuring out how fast things converge (i.e. how fast do the Fourier coefficients vanish as $n \to \infty$). Look at the plots, look for patterns and trends. Make hypothesis as to what is happening and **check them by further experimentation.** Use the script FSFun to produce functions where you can test your hypothesis. Write your conclusions in the answers. **Describe the evidence for your conclusions** — no proof is required, numerical evidence is enough, but **you must produce, and describe, the evidence!** Think of it in the same way that you would think in the situation of a lab experimenter trying to figure out what happens in some problem. **A few plots with your answer are fine, but please, just a few!**

IMPORTANT.
Anything smaller than about 10^{-14} is numerical error. Ignore it!

2 Special Problems.

2.1 Statement: GBNS04 Scheme D. Centered differences for $u_t = u_{xx}$.

See the GBNS Good and Bad Numerical Schemes Problems.

2.2 Statement: vNSA04 Scheme D. Centered differences for $u_t = u_{xx}$.

See the vNSA von Neumann Stability Analysis Problems.

2.3 Statement: AENS01 Scheme A. Forward differences for $u_t + u_x = 0$.

See the AENS Associated Equation to a Numerical Scheme Problems.

THE END.