Queens College, CUNY, Department of Computer Science
Numerical Methods
CSCI 361 / 761
Spring 2018
Instructor: Dr. Sateesh Mane

April 14, 2018

29 Lecture 29

- This lecture is devoted exclusively to one topic.
- The topic is a very important **highly efficient algorithm to sum a Fourier series.**
- The algorithm employs nested summation and Horner's rule (in two variables).

29.1 Motivation: magnetic field maps

- Magnetic field maps are frequently expressed using polar coordinates r and θ. (See Fig. 1.)

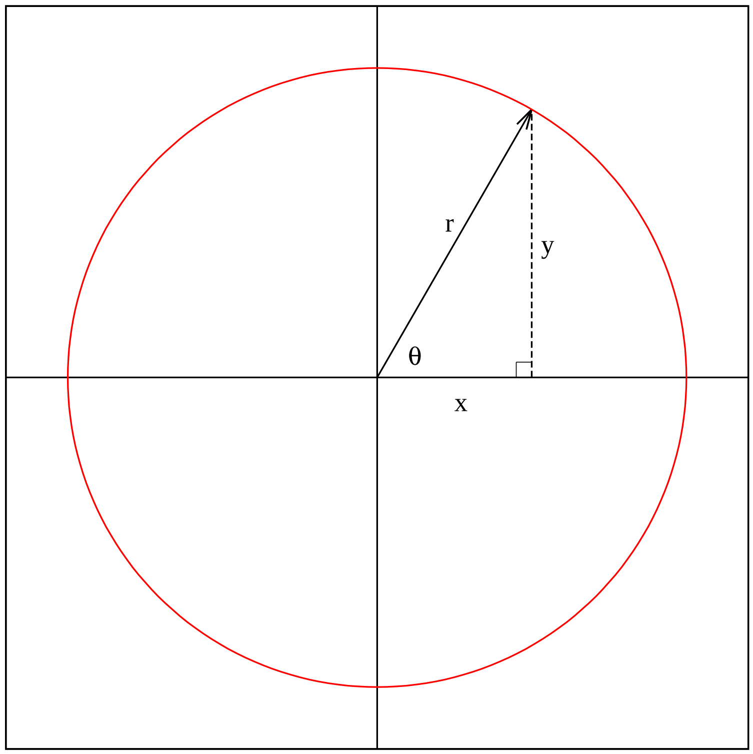

Figure 1: Sketch of polar coordinates r and θ.

- This is done, for example, for magnets in the particle accelerators at Brookhaven National Laboratory, as well as many other applications.
- *Note: there are also other ways of mapping magnetic fields.*
- Let us use polar coordinates and go around the circumference of a circle of radius r.
- From physics, we know the magnetic field is really a vector, but to keep things simple let us treat only a scalar function f.
- Suppose we measure a function $f(\theta)$ as a function of the polar angle θ (for a fixed value of r).
- If we go full circle, the function $f(\theta)$ must return to its original value, i.e.

$$f(\theta + 2\pi) = f(\theta)\,. \tag{29.1.1}$$

- In this situation, the function $f(\theta)$ can be expressed using a Fourier series.

29.2 Efficient summation of Fourier series

- For computational purposes, suppose the Fourier series is a finite sum, call it $f_n(\theta)$, given by

$$f_n(\theta) = \tfrac{1}{2}a_0 + \sum_{j=1}^{n} \bigl[\, a_j \cos(j\theta) + b_j \sin(j\theta) \,\bigr] \,. \tag{29.2.1}$$

- This is not as stupid as it sounds. The finite sum in eq. (29.2.1) appears in many applications for scientific computing, for example to approximate the magnetic fields in particle accelerators. I have written and operated such programs myself.

- However, the sine and cosine functions in eq. (29.2.1) are expensive to compute, even though they are standard mathematical library functions, available to full machine precision.

- There is an efficient extremely algorithm to to evaluate the sum in eq. (29.2.1).

- First we require the following trigonometric identities:

$$\begin{aligned} \cos((j+1)\theta) &= \cos(j\theta)\cos(\theta) - \sin(j\theta)\sin(\theta) \,. \\ \sin((j+1)\theta) &= \sin(j\theta)\cos(\theta) + \cos(j\theta)\sin(\theta) \,. \end{aligned} \tag{29.2.2}$$

- Define temporary variables $c = \cos(\theta)$ and $s = \sin(\theta)$ ***which we compute only once.***

- We employ a set of nested sums U_j and V_j, where $j = 1, \dots, n$, beginning with $U_n = a_n$ and $V_n = b_n$.

- Next note that, using eq. (29.2.2),

$$\begin{aligned} U_j \cos(j\theta) + V_j \sin(j\theta) &= a_j \cos(j\theta) + b_j \sin(j\theta) + U_{j+1}\cos((j+1)\theta) + V_{j+1}\sin((j+1)\theta) \\ &= \quad \cos(j\theta) \bigl[\, a_j + U_{j+1}\cos(\theta) + V_{j+1}\sin(\theta) \,\bigr] \\ &\qquad + \sin(j\theta) \bigl[\, b_j - U_{j+1}\sin(\theta) + V_{j+1}\cos(\theta) \,\bigr] \,. \end{aligned} \tag{29.2.3}$$

- Hence this establishes the recurrence, for $1 \le j \le n-1$:

$$\begin{aligned} U_j &= a_j + cU_{j+1} + sV_{j+1} \,, \\ V_j &= b_j - sU_{j+1} + cV_{j+1} \,. \end{aligned} \tag{29.2.4}$$

- At the end, instead of a_0 we have $\frac{1}{2}a_0$, hence the function value is

$$f_n(\theta) = \tfrac{1}{2}a_0 + cU_1 + sV_1 \,. \tag{29.2.5}$$

- **This is an extremely efficient sum. It is used in real-world applications.**

- It has been tacitly assumed that all the a_j and b_j are real numbers, hence $f(\theta)$ is also a real number. This is true in most (or almost all) practical applications.

- However, the above algorithm also works if the a_j and b_j are complex numbers.

- The loop is as follows. (It is coded for real variables.) Note that it works even if $n = 0$.

```
const double c = cos(theta);
const double s = sin(theta);
double U = 0;
double V = 0;
for (int j = n; j > 0; --j) {
  double Utmp = a[j] + c*U + s*V;
  double Vtmp = b[j] - s*U + c*V;
  U = Utmp;
  V = Vtmp;
}
double fn = 0.5*a[0] + c*U + s*V;
```

29.3 Example

- Let us try eq. (29.2.4) for a simple case.
- Let us set $n = 2$. Then

$$f_2(\theta) = \tfrac{1}{2}a_0 + a_1\cos(\theta) + a_2\cos(2\theta) + b_1\sin(\theta) + b_2\sin(2\theta)\,. \tag{29.3.6}$$

- We initialize $U_2 = a_2$ and $V_2 = b_2$.
- Next from eq. (29.2.4) we obtain

$$U_1 = a_1 + \cos(\theta)U_2 + \sin(\theta)V_2 \qquad = a_1 + \cos(\theta)a_2 + \sin(\theta)b_2\,, \tag{29.3.7a}$$

$$V_1 = b_1 - \sin(\theta)U_2 + \cos(\theta)V_2 \qquad = b_1 - \sin(\theta)a_2 + \cos(\theta)b_2\,. \tag{29.3.7b}$$

- Next we calculate f_2:

$$\begin{aligned} f_2(\theta) &= \tfrac{1}{2}a_0 + \cos(\theta)U_1 + \sin(\theta)V_1 \\ &= \tfrac{1}{2}a_0 + \cos(\theta)\Big[\,a_1 + \cos(\theta)a_2 + \sin(\theta)b_2\,\Big] \\ &\qquad + \sin(\theta)\Big[\,b_1 - \sin(\theta)a_2 + \cos(\theta)b_2\,\Big] \\ &= \tfrac{1}{2}a_0 + \cos(\theta)a_1 + [\cos^2(\theta) - \sin^2(\theta)]a_2 \\ &\qquad + \sin(\theta)b_1 + 2\sin(\theta)\cos(\theta)b_2 \\ &= \tfrac{1}{2}a_0 + \cos(\theta)a_1 + \cos^2(2\theta)a_2 \\ &\qquad + \sin(\theta)b_1 + \sin(2\theta)b_2\,. \end{aligned} \tag{29.3.8}$$

- Just for fun, what is the value of $V_0 = -\sin(\theta)U_1 + \cos(\theta)V_1$? We obtain

$$\begin{aligned} V_0 &= -\sin(\theta)U_1 + \cos(\theta)V_1 \\ &= -\sin(\theta)\Big[\,a_1 + \cos(\theta)a_2 + \sin(\theta)b_2\,\Big] \\ &\quad + \cos(\theta)\Big[\,b_1 - \sin(\theta)a_2 + \cos(\theta)b_2\,\Big] \\ &= -\sin(\theta)a_1 - \sin(\theta)\cos(\theta)a_2 - \sin^2(\theta)b_2 \\ &\quad + \cos(\theta)b_1 - \sin(\theta)\cos(\theta)a_2 + \cos^2(\theta)b_2 \\ &= -\sin(\theta)a_1 - \sin(2\theta)a_2 \\ &\quad + \cos(\theta)b_1 + \cos(2\theta)b_2\,. \end{aligned} \tag{29.3.9}$$

- If $a_0 = 0$, then notice that V_0 is a rotation of $f(\theta)$ through a right angle:

$$V_0 = f_2(\theta + \tfrac{1}{2}\pi)\,. \tag{29.3.10}$$

- This is a general relation valid for all n, if $a_0 = 0$.