Math 5620 Spring 2021

Notes of 1/10/21

• Go over syllabus

\[
\begin{array}{c}
PP \\
\rightarrow \\
MM \\
\rightarrow \\
MQ \\
\rightarrow \\
NA
\end{array}
\]

PP: Physical Problem
MM: Mathematical Models
MQ: Mathematical Question
NA: Numerical Answer

• this semester: MM = Differential Equations
• last semester: MM = everything else

• In particular, numerical PDEs is a larger subject than the rest of numerical analysis combined.

• But we will focus largely on ODEs to introduce and illustrate the relevant ideas.

• Purpose of numerical analysis: render itself obsolete by creating black box algorithms that will solve any problem of a certain kind that a user might come up with\(^1\).

\(^1\) Although, luckily for numerical analysts, users keep defeating that purpose by coming up with new and bigger problems every time there is an algorithmic breakthrough.
• The goal has been largely accomplished in particular in two areas:
  − small dense linear systems (Gaussian Elimination with partial pivoting)
  − small systems of initial value problems of ODEs
• The latter can be solved with VSVO (variable step/variable order) methods.
• Halfway through the semester we will understand VSVO methods quite thoroughly.
• We will then go on to boundary value problems (BVPs) of ODEs, and to PDEs
• The general first order ODE-IVP is

$$y' = f(x, y) \quad y(a) = y_0$$

where

$$a \leq x \leq b, \quad y \in \mathbb{R}^m \quad \text{and} \quad f(x, y) \in \mathbb{R}^m$$

• $m$ is the number of dependent variables. We have a system of $m$ ordinary differential equations.

![diagram]

An important special case is that $f$ does not depend on $y$. In that case we get an integration problem:

$$y' = f(x), \quad y(a) = 0, \quad y(x) = \int_a^x f(t)dt.$$
• Thus definite integrals can be considered the solutions of a special case of ODE-IVPs.

• The term project will consist of a friendly competition of writing the most efficient MATLAB code to compute definite integrals.

• That code should work like a VSVO code. It will be much simpler, however, because of the special structure of definite integrals. Details will be announced.

• Let’s review some relevant principles of numerical analysis which we already encountered last semester.

**Use of Structure**

- Examples:

  ODE-IVPs \[\rightarrow\] Definite Integrals
  
  sparse \[\rightarrow\] banded

  Linear Systems \[\rightarrow\]
  
  symmetric \[\rightarrow\] positive definite

• Special structure can be used to solve a problem more accurately, faster, or with less memory.

• Some problems become accessible only by using special structure.

• For example large linear systems can be solved only if they are sparse.
• Example: matlab takes about 0.3 seconds to solve a $1,000 \times 1,000$ linear system.

```matlab
>> a=rand(1000,1000);
>> b=rand(1000,1);
>> tic;
>> x=a\b;
>> toc
Elapsed time is 0.299431 seconds.
```

```matlab
>> r=norm(a*x-b)
r =
1.1345e-10
```

• We saw last semester that the effort to solve an $n \times n$ linear system by Gaussian elimination grows like $O(n^3)$

• Thus solving a one trillion by one trillion linear system by the same algorithm would take about $10^{19}$ years, a billion times the age of the Universe.

• Yet Google solves a linear system of that kind of size when computing the page rank of web pages.

\[^{−2}\text{There is a fascinating little book on this subject: Amy Langville and Carl Meyer: Google’s Page Rank and Beyond, the Science of Search Engine Rankings, Princeton University Press, 2006, ISBN 0-691-12202-4}\]
• By the same token, definite integrals are much more special than ODE-IVPs, and much easier to solve.

• In particular there are effective techniques that simply do not apply to ODEs.

• But still, definite integrals provide a good first trial problem.

Newton-Cotes Formulas

• In this context we will use another familiar principle:

• Make your method exact for specific functions.

• These functions are often polynomials.

• Other choices are possible, however.

• For definite integrals, quadrature, numerical integration, a frequently used criterion is to make the method exact if the integrand is a polynomial up to a specific degree.

• Basic idea: integrate an interpolating polynomial exactly.

• A polynomial interpolates itself, it is unique, and we get the exact integral.

• This gives rise to Newton-Cotes formulas:

\[
\int_a^b f(x) \, dx \approx \sum_{i=0}^{n} w_i f(x_i)
\]
where

\[ x_i = a + ih \quad \text{and} \quad h = \frac{b - a}{n}. \]

- The **Trapezoidal Rule** \((n = 1, \ h = b - a, \ x_0 = a, \ x_1 = b)\):

\[
\int_a^b f(x)\,dx \approx \frac{h}{2} (f(a) + f(b)).
\]

- **Simpson’s Rule** \((n = 2), \ h = \frac{b-a}{2}, \ x_0 = a, \ x_1 = \frac{a+b}{2}, \ x_2 = b\):

\[
\int_a^b f(x)\,dx \approx \frac{h}{3} (f(x_0) + 4f(x_1) + f(x_2)).
\]

- There are **open** or **closed** and **simple** or **composite** versions of Newton-Cotes Formulas.
• Example of Error Analysis (evaluations at $a$). Let $F' = f$.

$$E = \int_a^b f(x)dx - \frac{h}{2}(f(a) + f(b))$$

$$= F(a + h) - F(a) - \frac{h}{2}(f(a) + f(a + h))$$

$$= F + hf + \frac{h^2}{2}f' + \frac{h^3}{6}f'' - F$$

$$- \frac{h}{2}[f$$

$$+ f + hf' + \frac{h^2}{2}f''] + \text{HOT}$$

$$= -\frac{h^3}{12}f''(a) + \text{HOT}$$

where “HOT” stands for “higher order terms”.