# Lecture 4, Sci. Comp. for DPhil Students II

Nick Trefethen, Thursday 24.01.19

### Last lecture

- IV.8 Planetary motions
- IV.9 Chaos and Lyapunov exponents
- IV.10 The Lorenz equations
- IV.11 Sinai billiards and the SIAM 100-digit challenge

### Today

- IV.12 Time scales and stability regions
- IV.13 Stiffness
- IV.14 BVPs in Chebfun

### Handouts

- `m31_instability.m` - instability
- `m32_regions.m` - stability regions, using Chebfun
- `m33_stiffness.m` - stiffness for a scalar linear example
- `m34_stiffVDP.m` - stiffness for van der Pol equation
- `m35_ChebfunBVP.m` - Chebfun solution of BVPs ("chebop")
- Assignment 2

Assignment 2 is due Tuesday 5 February.

---

Last lecture we talked about planetary orbits (systems of ODEs) and about chaos and Lyapunov exponents.

Here's another chaotic system simulated in MATLAB – a double pendulum from Moler's online textbook: `swinger.m`. It's all done with `ode23`.

## IV.12 Time scales and stability regions

Consider the scalar, linear, autonomous equation

$$u' = au, \qquad u = Ce^{at}.$$

We sketch behavior for various choices of $a$:

$a > 0$

$a < 0$

$a \ll 0$

$a$ imaginary

$a$ complex

These different behaviours will have important effects on behaviour of a discretization. Note that the nondimensionalized quantity is $ka$, where $k$ is as always the time step:

$|ka| = O(1)$

$|ka| \ll 1$

A system of equations may mix many different values of $a$. This is analyzed normally via linearization and looking at eigenvalues of a Jacobian matrix. With these observation on the table we can turn to today's main subjects: stability regions and stiffness, which are key ideas at the heart of numerical methods for ODEs.

**ODE IVP:** $u' = f(t, u)$

Some ODE formulas are **unstable**: solutions blow up to infinity as $n \to \infty$, no matter how small $k$ is. We gave a "Brand X" example earlier:

$$v_{n+1} = -4v_n + 5v_{n-1} + k(4f_n + 2f_{n-1}) \qquad (X)$$

Even for a stable formula, solutions may blow up when $k$ is not small enough. Recall for example 2nd-order Adams-Bashforth:

$$v_{n+1} = v_n + \frac{k}{2}(3f_n - f_{n-1}) \qquad (AB2)$$

Let us demonstrate these effects. We'll keep it simple and consider the IVP

$$u' = -u, \qquad 0 \le t \le 10, \quad u(0) = 1.$$

Solution: $u(t) = \exp(-t)$

[`m31_instability.m` - both as written and with `%` character switched, `m31u`]

To analyze these phenomena, we apply the time-stepping formula to the model $u' = au$. This gives $f_n = av_n$, and the ODE formula becomes a **recurrence relation**. E.G. for AB2:

$$v_{n+1} = v_n + \frac{k}{2}(3av_n - av_{n-1}),$$

that is,

$$v_{n+1} = \left(1 + \frac{3ka}{2}\right) v_n - \frac{ka}{2} v_{n-1}.$$

$k$ and $a$ always appear together like this as the dimensionless product $ka$.

We analyze a recurrence relation by looking at roots of its **characteristic polynomial**. E.G. for AB2,

$$p(x) = x^2 - \left(1 + \frac{3ka}{2}\right) x + \frac{ka}{2}.$$

If $r$ is a root of $p$, i.e., $p(r) = 0$, then the recurrence has a solution

$$v_n = r^n$$

Blow-up: $|r| > 1$ for some root $r$ of $p$, or $|r| = 1$ for some double or multiple root.

Here's what we find:

For (X), there's a root $|r| \ge C > 1$ for all $ka$, even as $ka \to 0$. This formula is **unstable**. Indeed, consider $ka = 0$. Then (X) has characteristic polynomial

$$x^2 + 4x - 5 = (x + 5)(x - 1)$$

and obviously one root has $|r| > 1$.

For (AB2), there's a root $|r| > 1$ for $ka < -1$. For $-1 < ka < 0$, however, both roots have $|r| < 1$. And as $ka \to 0$ we get

$$x^2 - x = x(x - 1)$$

with $|r| \le 1$ for both roots. This formula is **stable**.

Given an ODE formula, we define:

**Stability region**: *set of points in complex $ka$-plane for which the recurrence relation has all roots $|r| \le 1$ and any roots with $|r| = 1$ are simple.*

A scheme is **stable** if 0 is in its stability region (necessarily on the boundary if the formula is consistent).

It tends to behave well for an ODE and step size $k$ if $ka$ is in the stability region for all values of $a$ "present in the ODE". (As mentioned before this is typically defined via linearization, eigenvalues . . . .)

Example: Euler formula

$$v_{n+1} = v_n + kf_n.$$

Apply this to the model problem $u' = au$. You get

$$v_{n+1} = (1 + ka)v_n$$

with characteristic polynomial $p(x) = x - (1 + ka)$.

Stability region = set of $ka$ such that $|1 + ka| \le 1$, i.e., disk of radius 1 centred at $ka = -1$.

[ `m32_regions.m` - sketches of stability regions ]

## IV.13 Stiffness

The classical stability/convergence theory for ODEs was established by Dahlquist in 1956.

Just a few years later it began to be widely appreciated that a major phenomenon was missing from this theory. The key mathematical paper was published by Dahlquist in 1963. With hindsight another key paper was by chemists Curtiss & Hirschfelder in 1952, who used the term "stiff", which may actually have originated with the statistician John Tukey (who also invented "FFT" and "bit" and was one of the first to speak of "software").

A **stiff ODE** is one with widely varying time scales.

More precisely, an ODE with solution of interest $u(t)$ is stiff when there are time scales present in the equation that are much shorter than that of $u(t)$ itself.

*Why it matters.* Regardless of stiffness, you'll get convergence in theory as $k \to 0$ (Dahlquist). For a stiff problem and an ordinary ODE formula, however, the result might be garbage except when $k$ is very small, because of modes $ka$ that aren't in the stability region.

Example:

$$u' = -100(u - \cos(t)) - \sin(t), \qquad u(0) = 1$$

Solution: $u(t) = \cos(t)$.

But what if we linearize about this solution, considering $u(t) = \cos(t) + w(t)$? We find

$$-\sin(t) + w' = -100w' - \sin(t), \quad \text{i.e.,} \quad w' = -100w.$$

So there's a time scale present in the equations that's 100 times shorter than that of the solution.

Cure for stiffness: **backward-differentiation formulas**. (At least, this is the starting point for a cure.) These are **implicit**, involving $f_{n+1}$ and hence requiring a equation solve at each step.

For example, here's the 2nd-order BD formula:

$$v_{n+1} - \frac{4}{3}v_n + \frac{1}{3}v_{n-1} = \frac{2k}{3}f_{n+1}.$$

Illustration of stiffness for this example ODE: `m33_stiffness.m`

Illustration of stiffness for the van der Pol equation: `m34_stiffVDP.m`

Section 10.3 of the *Chebfun Guide* describes how to invoke stiff solvers in Chebfun.

## IV.14 BVPs in Chebfun

We've been talking about IVPs, not BVPs., i.e., initial- rather than boundary-value problems. But BVPs are important too, and they appear in the assignment.

In ending this lecture let us briefly take a look at the "Chebop" part of Chebfun system, which solves BVPs using a `A\b` type syntax.

Reference: chapters 7 and 10 of the *Chebfun Guide* and maybe best of all Appendix A of *Exploring ODEs*, "Chebfun and its ODE algorithms". We won't talk now about how this works. Some of the underlying methods will appear in the later lectures of the course in the context of Chebyshev spectral methods.

Chebfun is the only software system I am aware of that solves IVPs and BVPs with the same syntax, namely backslash.

[ m35_ChebfunBVP.m ]

```
L = chebop(-1,1);                              % domain [-1,1]
L.op = @(x,u) .001*diff(u,2)+2*x*diff(u);   % u -> ep*u" + 2xu'
L.lbc = -1; L.rbc = 1;                         % left and right BCs
u = L\0;                                       % solve Lu = 0 for u
plot(u), grid on
length(u)          % For smaller ep, set domain with chebop([-1 0 1])
```