# Lecture 4. ODEs and Chebfun2

## 1. *Exploring ODEs*

T, Toby Driscoll (Delaware), and Ásgeir Birkisson (Squarepoint Capital). Chebfun-enabled, but not a book about algorithms. Published by SIAM last week; freely available at `https://people.maths.ox.ac.uk/trefethen/ExplODE`. Appendix B, "100 more examples," offers short Chebfun templates for many ODE problems.

## 2. Marching vs. global (e.g. Chebyshev) methods for ODEs: IVPs vs. BVPs

IVPs are usually solved by marching, notably Adams and other multistep methods or Runge-Kutta one-step methods. Global methods can work, if the problem is linear, but have few advantages. IVPs are fundamentally easy and standard software is routinely used, such as the Matlab suite due to Shampine and Reichelt: ODE23, ODE45, ODE113,…. Chebfun solves IVPs by ODE113 and converts the results to chebfuns.

BVPs are solved both by marching ("shooting") and global methods, though the former idea is perhaps more famous than it deserves to be. There are more challenging problems here, and the software situation is less standardized. Chebfun solves BVPs by global (or piecewise global) rectangular spectral collocation, as described below.

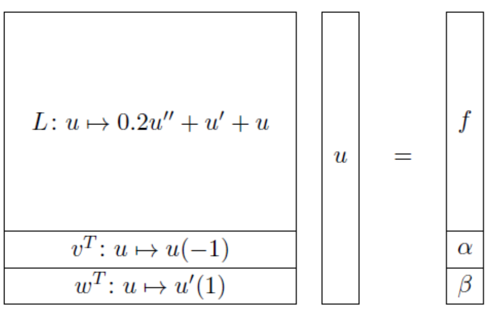
## 3. Block operators and spectral discretizations

[Refs: Driscoll and Hale, "Rectangular spectral collocation," *IMAJNA* 2016; Aurentz and T, "Block operators and spectral discretizations," *SIREV* 2017.]

Chebfun aimed to solve arbitrary BVPs, including systems, but finding a general procedure for discretizing boundary conditions proved problematic. Switching to rectangular spectral discretizations solved the problem. Idea: interpolate data from one Chebyshev grid; apply the differential operator; sample the result on a possibly different Chebyshev grid.

Stepping back from this process reveals it as a continuous analogue of the block matrices familiar in linear algebra. A typical 2nd-order block formulation of an ODE BVP has a rectangular block, wider than it is tall, together with two row blocks corresponding to boundary conditions. The operator without BCs has *index* 2 (nullity minus deficiency) and is discretized by matrices with two more columns than rows. Including rows to discretize the BCs makes the matrix square.

Example: $0.2u'' + u' + u = f$, $u(-1) = \alpha$, $u'(1) = \beta$.




Chebfun solves ODE BVPs by discretizations like this with $n$ chosen adaptively.

*Nonlinear* problems are represented with *shaded* blocks (possibly including shaded rows and/or columns). At each step of a Newton iteration, blocks are unshaded to guide discretization.

```
f = @(x) exp(x); alpha = 2; beta = 0; n = 18;
L = 0.2*diffmat([n n+2],2) + diffmat([n n+2],1) + diffmat([n n+2],0);
vT = diffrow(n+2,0,-1); wT = diffrow(n+2,1,1);
A = [L; vT; wT];
rhs = [gridsample(f,n); alpha; beta];
u = A\rhs;
```

## 4. Chebfun2 (two dimensions)

"Chebfun can do almost anything in 1D, and many things in 2D." Method: bivariate Chebyshev expansions compressed to low rank (especially good for functions aligned with axes). [Ref: Townsend & T, "An extension of Chebfun to 2D," *SISC* 2013]

## 5. Gaussian elimination as an iterative algorithm [Ref: Townsend & T, *SIAM News*, 2013]

CG = conjugate gradients: known in the 1950s as a direct algorithm, in the 1970s became the archetypical iterative algorithm
GE = Gaussian elimination: archetypical direct algorithm yet also the basic iterative algorithm for low-rank approximation