Nick Trefethen, 18 Sept. 2018

Lecture 3. Block operators and spectral discretizations

Refs: Driscoll and Hale, Rectangular spectral collocation, *IMA J. Numer. Anal.*, 2016
Aurentz and T, "Block operators and spectral discretizations," *SIAM Review*, 2017.
Software: Chebfun commands `diffmat, diffrow, gridsample, intmat, introw`

Computational science is built on linear algebra. Block matrices are a standard tool for exploiting structure.
This structure is usually in the continuous problem too, before discretization. That's the idea of block operators.
This was crucial for designing a black box ODE BVP solver for Chebfun. Without block operators, BCs are a mystery.

The mathematical fun comes from the fact that the blocks should be thought of as *rectangular*.
A 1st-order differential operator is " $\infty \times (\infty + 1)$ ", a 2nd order operator is " $\infty \times (\infty + 2)$ ", and so on.
This is made precise by the *index* of an operator: ind(L) = nullity − deficiency = dim(nullspace) − codim(range).
Index of a nonlinear operator = index of its (linear) Fréchet derivatives (assuming this is constant).

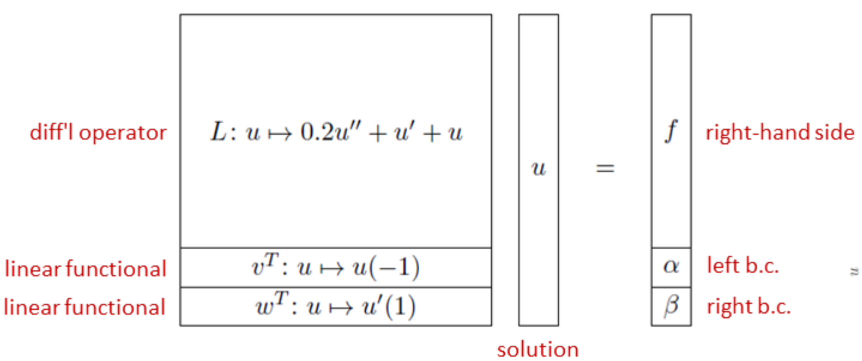
Differential or integral operator: rectangular block.
Function: column.
Functional: row.
Scalar: entry.

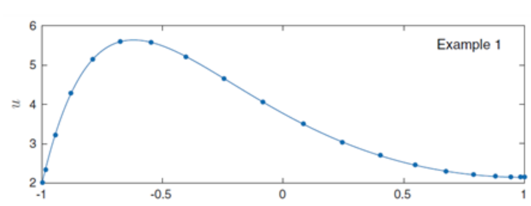
Rectangular operators then get discretized by rectangular matrices.
E.G., rectangular differentiation: interpolate on the $(n+1)$–point Chebyshev grid; differentiate;
sample on the n–point Chebyshev grid. This yields an $n \times (n+1)$ matrix.
Boundary conditions get added as additional rows, leading eventually to a square structure $Ax = b$ or $Ax = \lambda Bx$ or....

A "rectangular identity" (a dense matrix) makes good sense and in fact is needed for eigenvalue problems.

2. Linear Constant-Coefficient BVP (Example 1). As our first example, consider the advection-diffusion boundary-value problem (BVP)

$$0.2u''(x) + u'(x) + u(x) = f(x), \quad u(-1) = \alpha, \quad u'(1) = \beta, \tag{2.1}$$






```
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];
f = @(x) exp(x); alpha = 2; beta = 0;
rhs = [gridsample(f,n); alpha; beta];
u = A\rhs; plot(chebfun(u),'.-')
```

← Chebfun combines discretizations like this with automated selection of n based on rate of decay of Chebyshev series