Methods for general matrix functions

We now explore methods for matrix functions in general (not restricting to specific choices of f). [Higham book, Ch. 4]

Simple strategy: diagonalize $A = V\Lambda V^{-1}$, then compute

$$f(A) = Vf(\Lambda)V^{-1} = V \begin{bmatrix} f(\lambda_1) & & \\ & \ddots & \\ & & f(\lambda_m) \end{bmatrix} V^{-1}.$$

Works fine if A is symmetric/Hermitian/normal (and Q orthogonal). Otherwise, errors on $f(\lambda_i)$ (or in the diagonalization itself) are amplified by a factor $\kappa(V)$ — possibly much higher than the conditioning of the problem.

Example: sqrt of $\begin{bmatrix} 3 & -1 \\ 1 & 1 \end{bmatrix}$.

Alternative: do 'matrix algebra' directly, e.g., evaluate polynomials in matrix arguments.

Se $K(V) \gg 1$, un errore nel calcolo di $f(\lambda_1)$ viene amplificato di un fattore $K(V)$:

$$\left\| f(A) - V \begin{bmatrix} f(\lambda_1)+\varepsilon & & & \\ & f(\lambda_2) & & \\ & & \ddots & \\ & & & f(\lambda_m) \end{bmatrix} V^{-1} \right\| =$$

$$= \left\| V \begin{bmatrix} \varepsilon & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} V^{-1} \right\| \leq \|V\| \cdot \varepsilon \cdot \|V\|^{-1} = K(V)\,\varepsilon$$

Polynomial evaluation

How to evaluate polynomials in a matrix argument?

Unlike scalar polynomials, Horner method (i.e., $(\ldots((p_d A + p_{d-1})A + p_{d-2})A + \ldots)$ for matrix arguments is no better than 'direct' evaluation (build powers of A incrementally and sum them).

Even better: divide the terms into 'chunks' of size $\sqrt{d}$, e.g.,

$$(p_8 A^2 + p_7 A + p_6)(A^3)^2 + (p_5 A^2 + p_4 A + p_3)A^3 + (p_2 A^2 + p_1 A_1 + p_0).$$

(Paterson-Stockmayer method. — requires more storage though.)

Padé approximations

Variant: Padé approximations, i.e., rational approximations.

Padé approximant (at $x = 0$)

For every f analytic at 0 and for every choice of degrees $\deg p, \deg q$, one can find a rational function $\frac{p(x)}{q(x)}$ such that

$$f(x) - \frac{p(x)}{q(x)} = \mathcal{O}(x^{\deg p + \deg q + 1}).$$

i.e., "matches first $\deg p + \deg q$ terms of the MacLaurin series". (Count degrees of freedom to get a hint of why it works.)

For many functions, they have better approximation properties than Taylor series.

We will examine them for specific functions, e.g. the square root.

Matrix approximants

Good approximation of a scalar function is not good enough: even if $|f(x) - p(x)| < \varepsilon$ for each x, this only implies

$$\|f(A) - p(A)\| = \|V(f(\Lambda) - p(\Lambda))V^{-1}\| \leq \kappa(V)\varepsilon.$$

One needs to study approximation properties directly "at the matrix level".

Convergence of Taylor series

Theorem [Higham book Thm. 4.7]

Suppose $f = \sum_{k=0}^{\infty} a_k (x-\alpha)^k$, with $a_k = \frac{f^{(k)}(\alpha)}{k!}$, is a Taylor series with convergence radius r.
Then,

$$\lim_{d\to\infty} \sum_{k=0}^{d} a_k (A-\alpha I)^k = f(A)$$

for each A whose eigenvalues satisfy $|\lambda_i - \alpha| < r$.

Proof (sketch):

- It is enough to work on Jordan blocks.
- If $p_d(x)$ is the polynomial obtained by truncating the series to degre-d, then $p_d(\lambda I + N) = \sum_{k=0}^{d} p_d^{(k)}(\lambda) N^k$.
- $p_d^{(k)}$ is the truncated Taylor series of $f^{(k)}$, which has the same radius of convergence as that of f. So $p_d^{(k)}(\lambda) \to f^{(k)}(\lambda)$.
- The sum has at most *size*(N) terms (all zero afterwards).

Parlett recurrence

Can one compute matrix functions using the Schur form of A?

Example

$$A = \begin{bmatrix} t_{11} & t_{12} \\ 0 & t_{22} \end{bmatrix}, \quad f(A) = \begin{bmatrix} s_{11} & s_{12} \\ 0 & s_{22} \end{bmatrix}.$$

Clearly, $s_{11} = f(t_{11})$, $s_{22} = f(t_{22})$.

Trick: expanding $Af(A) = f(A)A$, one gets an equation for s_{12}:

$$t_{11}s_{12} + t_{12}s_{22} = s_{11}t_{12} + s_{12}t_{22} \implies s_{12} = t_{12}\frac{s_{11} - s_{22}}{t_{11} - t_{22}}.$$

(If $t_{11} = t_{22}$, the equation is not solvable and we already know that the finite difference becomes a derivative).

Parlett recurrence — II

The same idea works for larger blocks (provided we compute things in the correct order):

$$A = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ & t_{22} & t_{23} \\ & & t_{33} \end{bmatrix}, \quad f(A) = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ & s_{22} & s_{23} \\ & & s_{33} \end{bmatrix},$$

$$t_{11} s_{13} + t_{12} s_{23} + t_{13} s_{33} = s_{11} t_{13} + s_{12} t_{23} + s_{13} t_{33}.$$

Very similar to the algorithm we used to solve Sylvester equations. In some sense, we are solving the (singular) Sylvester equation $AX - XA = 0$, after setting specific elements on its diagonal.

The same idea works blockwise — the quotients become Sylvester equations.

Parlett recurrence — III

Algorithm (Schur–Parlett method)

1. Compute Schur form $A = QTQ^*$;
2. Partition T into blocks with 'well-separated eigenvalues';
3. Compute $f(T_{ii})$ (e.g., with Taylor series in the centroid of its eigenvalues);
4. Use recurrences to compute off-diagonal blocks of $f(T)$;
5. Return $f(A) = Qf(T)Q^*$.

Tries to get 'best of both worlds': uses Taylor expansion when the eigenvalues are close, recurrences when they are distant.

Parlett recurrence and block diagonalization

The Parlett recurrence is 'almost the same thing' as block diagonalization. Consider the case of 2 blocks for simplicity. T can be block-diagonalized via

$$W^{-1}TW = \begin{bmatrix} I & -X \\ 0 & I \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} I & X \\ 0 & I \end{bmatrix} = \begin{bmatrix} T_{11} & \\ & T_{22} \end{bmatrix}$$

where X solves $T_{11}X - XT_{22} + T_{12} = 0$ (Sylvester equation). Then

$$f(T) = W \begin{bmatrix} f(T_{11}) & \\ & f(T_{22}) \end{bmatrix} W^{-1} = \begin{bmatrix} f(T_{11}) & Xf(T_{22}) - f(T_{11})X \\ & f(T_{22}) \end{bmatrix}.$$

(Note indeed that $S = Xf(T_{22}) - f(T_{11})X$ solves the Sylvester equation appearing in the Parlett recurrence.)

So both methods solve a Sylvester equation with operator $Z \mapsto T_{11}Z - ZT_{22}$.