The Faddeev–LeVerrier algorithm

The Faddeev–LeVerrier algorithm may be used to determine the inverse, determinant, and characteristic polynomial of an $n \times n$ matrix. The algorithm terminates in n steps, where each step involves a single matrix multiplication and only integer division. It works like magic!

FADDEEV-LEVERRIER ALGORITHM

given an $n \times n$ matrix A
$c_n := 1$
$N \leftarrow \mathbb{0}$
for $k \in (n-1, \dots, 1, 0)$
    $N \leftarrow N + c_{k+1}\mathbb{I}$
    $c_k := \frac{1}{k-n} \operatorname{tr}(AN)$
return
    $A^{-1} = -N/c_0$
    $\det(A) = (-1)^n c_0$
    $\chi(\lambda) = \sum_{k=0}^{n} c_k \lambda^k$

Derivation

Start with the characteristic polynomial of A.

$$\chi(\lambda) = \det(\lambda\mathbb{I} - A) = \sum_{k=0}^{n} c_k \lambda^k$$

> **Useful fact.**
>
> The *adjunct* of a matrix, $\operatorname{adj}(X)$, satisfies $\det(X)\mathbb{I} = A \operatorname{adj}(X)$.
>
> If A is $n \times n$, then $\det(X)$, and hence the entries of $X \operatorname{adj}(X)$, are degree n polynomials in the entries of A. Hence, the entries of $\operatorname{adj}(X)$ are degree $n-1$ polynomials.

The entries of $N(\lambda) := \operatorname{adj}(\lambda\mathbb{I} - A)$ are λ-polynomals of order $n-1$, so $N(\lambda) = \sum_{k=0}^{n-1} N_k \lambda^k$ where N_k are matrices. From $\det(\lambda\mathbb{I} - A)\mathbb{I} = (\lambda\mathbb{I} - A)N(\lambda)$,

$$\begin{aligned} \det(\lambda\mathbb{I} - A)\mathbb{I} &= (\lambda\mathbb{I} - A)\sum_{k=0}^{n-1} N_k \lambda^k \\ &= -AN_0 + \sum_{k=1}^{n-1}(N_{k-1} - AN_k)\lambda^k + N_{n-1}\lambda^n \end{aligned}$$

Equating coefficients of λ with $\chi(\lambda)\mathbb{I}$, we obtain:

$$\begin{aligned} c_0\mathbb{I} &= \phantom{N_{k-1}} - AN_0 \\ c_k\mathbb{I} &= N_{k-1} - AN_k \\ c_n\mathbb{I} &= N_{n-1} \end{aligned}$$

To remember these, just write $c_k\mathbb{I} = N_{k-1} - AN_k$ for all $0 \le k \le n$ with the understanding that N_k vanishes outside the range $0 \le k \le n-1$. Equivalently,

$$N_k = N_{k+1} - c_{k+1}\mathbb{I}$$

gives a descending recurrence relation for N_k in terms of the coefficients c_k.

Finding c_k in terms of A and N_k

This stroke of genious is due to (Hou, 1998).

> **Useful fact:** Laplace transform of derivative.
>
> $$\begin{aligned} \mathcal{L}\{f'(t)\}(s) &= \int_0^\infty f'(t)e^{-st}\,\mathrm{d}t \\ &= f(t)e^{-st}\Big|_{t=0}^\infty + s\int_0^\infty f(t)e^{-st}\,\mathrm{d}t \\ &= -f(0) + s\mathcal{L}\{f(t)\}(s) \end{aligned}$$

Consider

$$\frac{\mathrm{d}e^{At}}{\mathrm{d}t} = Ae^{At}$$

and perform the Laplace transform to obtain

$$-\mathbb{I} + s\mathcal{L}\{e^{At}\} = A\mathcal{L}\{e^{At}\}$$

and finally take the trace:

$$s \operatorname{tr}\mathcal{L}\{e^{At}\} - n = \operatorname{tr}(A\mathcal{L}\{e^{At}\}) \tag{1}$$

> **Useful fact:** the trace of a matrix exponential in terms of eigenvalues.
>
> If λ_i are the eigenvalues of A then $\operatorname{tr}(A) = \sum_i \lambda_i$. Also, A can be put in Jordan normal form $A = PJP^{-1}$ where J is triangular with $\operatorname{diag}(J) = (\lambda_1, \dots, \lambda_n)$. Since it is triangular, $\operatorname{diag}(J^k) = (\lambda_1^k, \dots, \lambda_n^k)$.
>
> Therefore, $\operatorname{tr}(A^k) = \operatorname{tr}(PJ^kP^{-1}) = \operatorname{tr}(J^kP^{-1}P) = \operatorname{tr}(J^k) = \sum_{i=1}^n \lambda_i^k$.
>
> Consequently, $\operatorname{tr}(e^{At}) = \sum_{k=0}^\infty \frac{t^k}{k!}\operatorname{tr}(A^k) = \sum_{k=0}^\infty \frac{t^k}{k!}\sum_{i=1}^n \lambda_i^k = \sum_{i=1}^n e^{\lambda_i t}$.

We now compute the terms in Equation 1.

$$\mathcal{L}\{e^{At}\} = \int_0^\infty e^{(A-s\mathbb{I})t}\,\mathrm{d}t = (A - s\mathbb{I})^{-1}e^{(A-s\mathbb{I})t}\Big|_{t=0}^\infty = (s\mathbb{I} - A)^{-1}$$

> **Caution.**
>
> I'm uncomfortable with these indefinite integrals. Why should $\lim_{t\to\infty} e^{(A-s\mathbb{I})t}$ converge?

Note that from $\chi(\lambda) = \det(\lambda\mathbb{I} - A) = (\lambda\mathbb{I} - A)N(\lambda)$ we have

$$(\lambda\mathbb{I} - A)^{-1} = \frac{N(\lambda)}{\chi(\lambda)} \tag{2}$$

Let $(\lambda_1, \dots, \lambda_n)$ be the eigenvalues of A. Then $A - s\mathbb{I}$ has eigenvalues $\lambda_i - s$.

$$\operatorname{tr}\mathcal{L}\{e^{At}\} = \int_0^\infty \operatorname{tr} e^{(A-s\mathbb{I})t}\,\mathrm{d}t = \sum_{i=1}^n \int_0^\infty e^{(\lambda_i - s)t}\,\mathrm{d}t = \sum_{i=1}^n \frac{1}{s - \lambda_i}$$

Recall that the roots of the characteristic polynomial of A are its eigenvalues, so $\chi(s) = \prod_{i=1}^n (s - \lambda_i)$.

$$\operatorname{tr}\mathcal{L}\{e^{At}\} = \sum_{i=1}^n \frac{\mathrm{d}}{\mathrm{d}s}\ln(s - \lambda_i) = \frac{\mathrm{d}}{\mathrm{d}s}\ln\left(\prod_{i=1}^n (s - \lambda_i)\right) = \frac{\mathrm{d}}{\mathrm{d}s}\ln\chi(s) = \frac{\chi'(s)}{\chi(s)} \tag{3}$$

Substituting Equation 2 and Equation 3 into Equation 1, we have

$$\begin{aligned} s\chi'(s) - n\chi(s) &= \operatorname{tr}(AN(\lambda)) \\ \sum_{k=0}^n (k-n)c_k s^k &= \sum_{k=0}^{n-1}\operatorname{tr}(AN_k)s^k \end{aligned}$$

which, expanding and equating powers of λ,

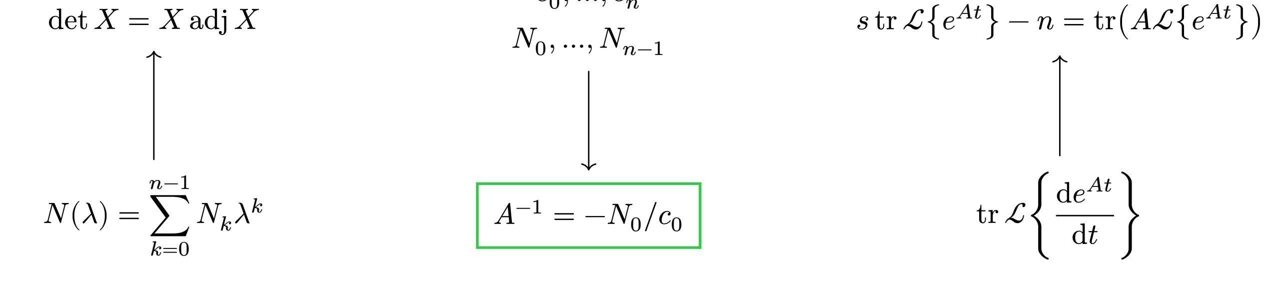
$$c_k = \frac{\operatorname{tr}(AN_k)}{k-n}$$

for all $0 \le k \le n$ where we define $N_n = 0$.

Final algorithm

> **Useful fact.**
>
> $$\begin{aligned} \chi(\lambda) &= c_0 + \dots + \phantom{\operatorname{tr}(-A)} c_{n-1}\lambda^{n-1} + c_n\lambda^n \\ &= \det(-A) + \dots + \operatorname{tr}(-A)\lambda^{n-1} + \lambda^n \end{aligned}$$
>
> $$c_0 = (-1)^n\det(A), \quad c_{n-1} = -\operatorname{tr}(A), \quad c_n = 1$$

Visual summary



References

Hou, S.-H. (1998). Classroom Note: A Simple Proof of the Leverrier–Faddeev Characteristic Polynomial Algorithm. *SIAM Rev.*, *40*(3), 706–709. https://doi.org/10.1137/S003614459732076X