Background - uniform circuit families and parallel complexity.

__Definition 1__: a circuit family $\{C_n\}_{n\geq 0}$ is said to be log-space uniform if there is a log-space transducer which on input 1^n outputs the description of C_n.

__Definition 2__: For $i \in \mathbb{N}$, let NC^i denote the class of languages $A \subseteq \{0,1\}^*$ that can be decided by a log-space uniform family $\{C_n\}_{n\geq 0}$ of circuits with $SIZE(C_n) = O(n^c)$ and $DEPTH(C_n) = O(\log^i n)$ for some constant $c \in \mathbb{N}$.
We denote $NC^i/_{poly}$ the class of languages decided by a non-uniform family satisfying the same size and depth constraints.
Finally :

$$NC := \bigcup_{i \in \mathbb{N}} NC^i$$

$$NC/_{poly} := \bigcup_{i \in \mathbb{N}} NC^i/_{poly}$$

__Remark 1__: we can define similar complexity classes for general functions $f: \{0,1\}^* \to \{0,1\}^*$, which yields the classes FNC^i ($FNC^i/_{poly}$) and FNC ($FNC/_{poly}$).

Parallel & Space complexity of linear algebra

- Fundamental tasks of linear algebra:
  - invert a matrix
  - solve system of linear equations
  - compute the determinant of a matrix
  - compute the characteristic polynomial of a matrix

Csanky 1976

Let $\mathbb{F}$ be a field with $\mathrm{char}(\mathbb{F}) = 0$.

Theorem: given a matrix $A \in \mathbb{F}^{n\times n}$, can compute $\det(A)$, A^{-1}, $p_A(t)$ in $O(\log^2 n)$ parallel time with polynomially many processors. That is, there exist circuit families $\{C_{bn}\}_{n\geq 1}$ where $b \in [3]$ s.t. $C_{1n}(A) = \det(A)$, $C_{2n}(A) = A^{-1}$, $C_{3n}(A,t) = p_A(t)$ where for each $b \in [3]$, $\{C_{b,n}\}_{n\geq 1} \in FNC^2$.

Proof: let $\lambda_1(A), \dots, \lambda_n(A)$ be the eigenvalues of A (with multiplicities). Let $c_k := (-1)^{k-1} \cdot e_k(\lambda_1(A), \dots, \lambda_n(A))$ and $s_k := \sum_{i=1}^{n} \lambda_i(A)^k = tr(A^k)$.

By the Girard-Newton formulae, we get

$$\underbrace{\begin{pmatrix} 1 & 0 & & & & \\ s_1 & 2 & 0 & & 0 & \\ s_2 & s_1 & 3 & 0 & & \\ s_3 & s_2 & s_1 & \ddots & \ddots & \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ s_{n-1} & s_{n-2} & s_{n-3} & \cdots & s_1 & n \end{pmatrix}}_{S} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_n \end{pmatrix}$$

For $k \in [n]$, let $D^{(k)}$ be the $n \times n$ matrix given by
$$D^{(k)}_{ij} = \begin{cases} 1/k & \text{if } i=j=k \\ 1 & \text{if } i=j\neq k \\ 0 & \text{o.w.} \end{cases}$$

and let
$$M^{(k)} = \begin{pmatrix} I_{k-1} & \begin{matrix} 0 \end{matrix} \\ 0 & \begin{array}{c|c} 1 & 0 \\ \hline -u_k & I_{n-k} \end{array} \end{pmatrix} \quad \text{where } u_k = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{n-k} \end{pmatrix}.$$

By induction, we see that
$$\prod_{j=k}^{1} \left(M^{(j)} D^{(j)} \right) \cdot S = \begin{pmatrix} I_k & 0 \\ 0 & \begin{matrix} k+1 & & \\ s_1 & k+2 & 0 \\ \vdots & \ddots & \ddots \\ s_{n-k-1} & \cdots\ s_1 & n \end{matrix} \end{pmatrix}$$
note the order of the indices! (it matters since matrices do not commute!)

Thus $\prod_{j=n}^{1} (M^{(j)} D^{(j)}) \cdot S = I \quad \therefore$

$S^{-1} = \prod_{j=n}^{1} (M^{(j)} D^{(j)}) \Rightarrow$ can compute S^{-1} in NC^2. (because we can compute $M^{(j)}, D^{(j)}$ in parallel)

Hence, can compute all s_k, c_k in FNC^2.
In particular, we have computed $\det(A) = (-1)^{n-1} c_n$ and $p_A(t) = t^n - \sum_{i=1}^{n} c_i t^{n-i}$.

By Cayley-Hamilton, we have

$$p_A(A) = 0 \iff A^n - c_1 A^{n-1} - c_2 A^{n-2} - \dots - c_n I = 0$$
$$\Rightarrow A \cdot (A^{n-1} - c_1 A^{n-2} - \dots - c_{n-1} I) = c_n I$$

$$\Rightarrow A^{-1} = \frac{1}{c_n} (A^{n-1} - c_1 A^{n-2} - \dots - c_{n-1} I)$$

Since we can compute all powers A^k in FNC^2 (simultaneously) and we have computed the c_k in NC^2, we can compute A^{-1} in FNC^2.

Note that the above circuit can be computed