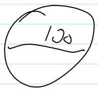
14-1100 Nov 20, hour 29: The JCF

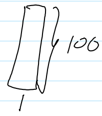
November-15-14 5:54 PM

Plan. JCF abstractly & in practice.

HW4 due, HW5 on web.

Riddle. 1. A spherical loaf of bread goes into a bread cutting machine. Which slice has the most crust?

2. Can you cover 

with 99 × 

?

Corollary 2. *Over an algebraically closed field* $\mathbb{F}$*, every square matrix* A *is conjugate to a block diagonal matrix* $B = \begin{pmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_n \end{pmatrix}$*, where each* B_i *is either a* 1×1 *matrix* (λ_1) *for some* $\lambda_i \in \mathbb{F}$*, or an* $s_i \times s_i$ *matrix with* λ_i*'s on the diagonals, 1's right below the diagonal, and 0's elsewhere,*

$$\begin{pmatrix} \lambda_i & 0 & \cdots & \cdots & 0 & 0 \\ 1 & \lambda_i & \ddots & & & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & & \ddots & \ddots & \lambda_i & 0 \\ 0 & 0 & \cdots & 0 & 1 & \lambda_i \end{pmatrix},$$

for some $\lambda_i \in \mathbb{F}$ *and for some* $s_i \geq 2$*. Furthermore,* B *is unique up to a permutation of its blocks* B_i*.*
(Corollary: good old diagonalization.)

on projector screen

JCF. V a f.d.v.s, $A: V \to V$ linear, makes V a module over $R := F[x]$ via $xu = Au$. Then $V \cong \bigoplus F[x]/(x-\lambda_i)^{s_i}$. What's $F[x]/(x-\lambda_i)^{s_i}$?

Basis: $1, x-\lambda, (x-\lambda)^2, \ldots, (x-\lambda)^{s-1}$

$A-\lambda$ acts by "shift to the right" $\begin{pmatrix} 0 & 0 & \\ 1 & 0 & \\ & 1 & \ddots \end{pmatrix}$

So A acts by $\begin{pmatrix} \lambda & & \\ 1 & \lambda & \\ & 1 & \lambda \end{pmatrix}$

Now lets do that in practice ...

step 1. Find a presentation matrix for $V \in R\text{-mod}$.

w.l.o.g $V = F^n$ and $A \in M_{n\times n}(F)$. $\ker \pi = ?$

$r_i = xe_i - Ae_i \in \ker \pi$

$R^n \xrightarrow{xI-A} R^n \xrightarrow{\pi} F^n$

$e_i \longrightarrow e_i$

$x^k e_i \longrightarrow A^k e_i$

<u>claim</u> $\langle r_i \rangle = \ker \pi$

<u>pf</u> Consider

$$\begin{tikzcd} F^n \arrow[r, "\text{onto?}", "\beta \text{ v.s. map}"'] \arrow[rrr, bend right, "I"'] & R^n/\langle r_i\rangle \arrow[r, "1-1?", "\alpha \text{ well-def'd}"'] & R^n/\ker\Pi \arrow[r, "\sim"] & F^n \end{tikzcd}$$

We want to know if α is 1-1; it is enough to show that β is onto; i.e., that any $x^k e_i$ can be written, modulo $\langle r_i\rangle$, as a combination of e_i's. Indeed,

$$x^k e_i = x^{k-1}(x e_i) = x^{k-1} A e_i = \dots = A^k e_i$$

Go over handout along with "run 1"

Dror Bar-Natan: Classes: 2014-15: Math 1100 Algebra I:

JCF Tricks and Programs

Row and Column Operations

Row operations are performed by left-multiplying N by some properly-positioned 2×2 matrix and at the same time left-multiplying the "tracking matrix" P by the same 2×2 matrix. Column operations are similar, with left replaced by right and P by Q.

```
RowOp[i_, j_, mat_] := Module[{TT = II},
   TT[[{i, j}, {i, j}]] = mat;
   NN = Simplify[TT.NN]; PP = Simplify[TT.PP];
  ];
ColOp[i_, j_, mat_] := Module[{TT = II},
   TT[[{i, j}, {i, j}]] = mat;
   NN = Simplify[NN.TT]; QQ = Simplify[QQ.TT];
  ];
```

Swapping Rows and Columns

```
SwapRows[i_, j_] := RowOp[i, j, {{0, 1}, {1, 0}}];
SwapColumns[i_, j_] := ColOp[i, j, {{0, 1}, {1, 0}}];
SwapBoth[i_, j_] := (SwapRows[i, j]; SwapColumns[i, j];)
```

The "GCD" Trick

If $q=\gcd(a,b)=sa+tb$, the equality $\begin{pmatrix} s & t \\ -b/q & a/q\end{pmatrix}\begin{pmatrix} a \\ b\end{pmatrix}=\begin{pmatrix} q \\ 0\end{pmatrix}$ allows us to replace pairs of entries in the same column by their greatest common divisor (and a zero!), using invertible row operations. A similar trick works for rows.

```
? PolynomialExtendedGCD
```

PolynomialExtendedGCD[$poly_1$, $poly_2$, x] gives the extended GCD of $poly_1$ and $poly_2$ treated as univariate polynomials in x.
PolynomialExtendedGCD[$poly_1$, $poly_2$, x, Modulus → p] gives the extended GCD over the integers mod prime p. >>

```
GCDTrick[{i_, j_}, k_] := Module[{a, b, q, s, t},
   {q, {s, t}} = PolynomialExtendedGCD[a = NN[[i, k]],
     b = NN[[j, k]], x];
   RowOp[i, j, {{s, t}, {-b/q, a/q}}]
  ];
GCDTrick[k_, {i_, j_}] := Module[{a, b, q, s, t},
   {q, {s, t}} = PolynomialExtendedGCD[a = NN[[k, i]],
     b = NN[[k, j]], x];
   ColOp[i, j, {{s, -b/q}, {t, a/q}}]
  ];
```

delayed

Factoring Diagonal Entries

If $1=\gcd(a,b)=sa+tb$, the equality $\begin{pmatrix} sa & 1 \\ -tb & 1\end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & ab\end{pmatrix}\begin{pmatrix} a & -b \\ t & s\end{pmatrix}=\begin{pmatrix} a & 0 \\ 0 & b\end{pmatrix}$ is an invertible row-column-operations proof of the isomorphism $\frac{R}{\langle a\rangle}\oplus\frac{R}{\langle b\rangle}\simeq\frac{R}{\langle ab\rangle}$.

```
SplitToSum[i_, j_, a_, b_] := Module[
   {q, s, t, T1, T2},
   {q, {s, t}} = PolynomialExtendedGCD[a, b, x];
   If[q == 1,
    RowOp[i, j, {{s a, 1}, {-t b, 1}}]; ColOp[i, j, {{a, -b}, {t, s}}];
   ]
  ];
```

The Jordan Trick

A repeated application of the identity

$$\begin{pmatrix} p^{k-1} & -1 \\ 1 & 0\end{pmatrix}\cdot\begin{pmatrix} 1 & 0 \\ 0 & p^k\end{pmatrix}\cdot\begin{pmatrix} 1 & p \\ 0 & 1\end{pmatrix}=\begin{pmatrix} p^{-1+k} & 0 \\ 1 & p\end{pmatrix}$$

will bring a matrix like $\begin{pmatrix} 1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&p^4\end{pmatrix}$ to the "Jordan" form of $\begin{pmatrix} p&0&0&0\\1&p&0&0\\0&1&p&0\\0&0&1&p\end{pmatrix}$, using invertible row and column operations.

```
JordanTrick[i_, j_, p_, s_] :=
  (RowOp[i, j, {{p^(s-1), -1}, {1, 0}}]; ColOp[i, j, {{1, p}, {0, 1}}]);
```

along with:

Running the JCF Programs

```
In[2]:= SetDirectory["C:\\drorbn\\AcademicPensieve\\Classes\\14-1100"];
        << JCF-Program.m
```

Matrix 1 - 3x3, 3 eigenvalues.

In[4]:= n = 3; AA = $\begin{pmatrix} 3 & 0 & 0 \\ 4 & -2 & -6 \\ -2 & 0 & 1 \end{pmatrix}$;

```
PP = QQ = II = IdentityMatrix[n];
MM = x II - AA;
NN = PP.MM.QQ;
```

done to $\begin{pmatrix} 1 & & \\ & 1 & \\ & & ()() \end{pmatrix}$

Recovering C from P?

$$\begin{array}{ccccc} R^n & \xrightarrow[M]{Ix-A} & R^n & \xrightarrow{\pi_A} & F^n \\ Q\uparrow & & \downarrow P & & \downarrow C \\ R^n & \xrightarrow[N]{Ix-B} & R^n & \xrightarrow{\pi_B} & F^n \end{array}$$

$$\begin{aligned} Ce_i &= \pi_B(Pe_i) \\ &= \pi_B\left(\sum x^k P_k e_i\right) \\ &= \sum x^k \pi_B(P_k e_i) \\ &= \sum B^k P_k e_i \end{aligned}$$

$\Longrightarrow C = \sum B^k P_k$ ... complete run 1

Go through run 2 until stuck, then

The "Jordan Trick": $R/\langle p^s\rangle = \langle x\rangle / p^s x = 0$

$x_0 = x$

$x_1 = -px$

$x_2 = p^2 x$

$= \langle x_0 \dots x_{s-1}\rangle \Big/ \begin{matrix} px_i + x_{i+1} = 0 \\ px_{s-1} = 0 \end{matrix}$

so $(p^s) \sim \begin{pmatrix} p & & & \\ 1 & p & & \\ & 1 & p & \\ & & 1 & p \end{pmatrix}$

more precisely: $\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & p^s \end{pmatrix} \sim \begin{pmatrix} p & & & & \\ 1 & p & & & \\ & 1 & p & & \\ & & 1 & p & \\ & & & 1 & p \end{pmatrix}$

Explicitly:

A repeated application of the identity $\begin{pmatrix} p^{k-1} & -1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & p^k \end{pmatrix} \cdot \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} p^{-1+k} & 0 \\ 1 & p \end{pmatrix}$ will bring a matrix like $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & p^4 \end{pmatrix}$ to the "Jordan" form of $\begin{pmatrix} p & 0 & 0 & 0 \\ 1 & p & 0 & 0 \\ 0 & 1 & p & 0 \\ 0 & 0 & 1 & p \end{pmatrix}$, using invertible row and column operations.

JordanTrick[i_, j_, p_, s_] := (RowOp[i, j, $\begin{pmatrix} p^{s-1} & -1 \\ 1 & 0 \end{pmatrix}$]; ColOp[i, j, $\begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix}$]);

Then go through the rest of run 2 & through run 3 . . . . .