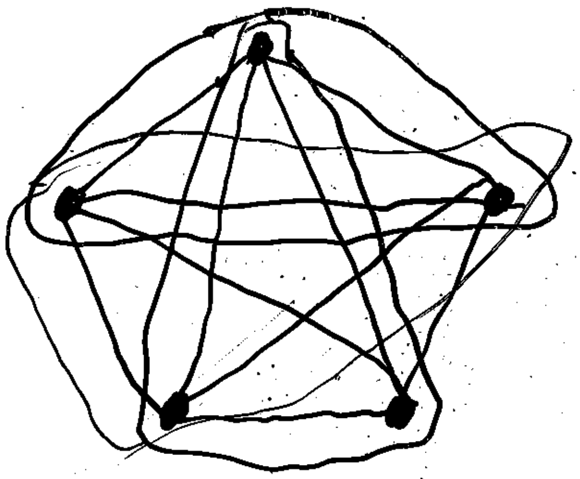
Covering designs

$C(n, k, r)$
A smallest collection of sets of size k such that every set of size r is a subset of one of them

ex



$n = 5$

$r = 2$

$k = 3$

sets
4 ~~number~~
~~istället för 10~~
instead of 10

Covering designs can be built inductively for larger n

I have been constructing a database with <u>all</u> optimal designs for small n, k, r

used close to ~~[illegible]~~ 600 over yr

widely downloaded

Row reduction

$$\begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} \begin{matrix} \\ \tfrac{1}{2} \end{matrix} \sim \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

For real and complex matrices this requires n^3 operations

What about $GF(q)$? $(= \mathbb{Z}_p$ p prime$)$

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Striped matrix elimination
(Andrén, Hellström, M)

$$\leq \frac{n^3}{\log_q n}(1+\varepsilon)$$

Lower bounds?

There are $q(n-1)n \cdot \overset{\text{const.}}{A_{n,q}}$ row operations

With k operations we can reach at most

$$\left(q(n-1)n A_{n,q}\right)^k$$

matrices

This gives a lower bound of $\frac{n^2}{2\log_q n} \leq$

factor of 2 from algorithm

The exact values?

4 The optimal results for small n and q

While our bounds for $\mathcal{D}(n,q)$ for a fixed q are quite good they are still far from giving us the exact values of $\mathcal{D}(n,q)$, and the bounds for growing q even more so. In order to determine the exact values for small n and q and get a more detailed picture of how the Cayley graph develops we have performed a computational investigation as well.

4.1 Experimental set-up

We wrote a C program which performs a breadth first search from the identity matrix. In the standard way the program keeps track of a 'state' of each vertex—whether the vertex is in the current level, the next level, some earlier level, or has not yet been seen—whereas there is no explicit representation of edges. In order to manage the larger graphs the vertex states were coded using only 2 bits of storage per vertex. This was done by allocating a large bit vector and then assigning to each matrix with nonzero columns a pair of consecutive bits within this vector; in Table 1 we can see the memory requirements for small n and q. As there are $q^n - 1$ different values for a nonzero column, a unique position for a matrix could be computed simply by interpreting its n columns as n digits in base $q^n - 1$. This use of column vectors also simplified implementing the row operations, as we could tabulate the function saying "row operation i transforms the column with number j to the column with number k" and thus compute the positions of all neighbours of a given vertex through elementary arithmetic and table look-ups.

	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$
$q=2$	<1	<1	<1	<1	15	$1.2\cdot10^5$
$q=3$	<1	<1	<1	194	$3.4\cdot10^7$	
$q=4$	<1	<1	<1	$2.6\cdot10^5$		
$q=5$	<1	<1	36			
$q=7$	<1	<1	7725			
$q=8$	<1	<1				
$q=9$	<1	<1				
$q=11$	<1	<1				
$q=13$	<1	3				
$q=16$	<1	16				
$q=17$	<1	28				
$q=19$	<1	76				
$q=23$	<1	420				
$q=25$	<1	889				

Table 1: Memory requirement for the Cayley graph in gigabytes.

As the search progresses the program outputs the number of vertices that belong to each distance class in the graph; these data are shown in tables 3–7. An unusual feature is that some graphs have one very large distance class (see e.g. Table 4) that can account for well over half the vertices of that graph. In these cases we could save a large amount of work by first searching forward and then backward. During the first phase (forward search) the program constructs the next distance class by applying all generators in our set S to each element in the current distance class. During the second phase (backward search), which starts when a predetermined distance class K is reached, the program instead applies the generators to all matrices not yet encountered in order to see if they have a neighbour in the previously constructed distance class; in this phase the processing of a vertex stops as soon as a neighbour in the lower distance class is found.

Our program was also parallelised using OpenMP. The computations were performed on three different SGI Origin machines. The largest case was the computation for GL(3, 23) which in total used over 430 gigabytes of RAM, running on over 400 processors for several days and accumulating a run time of 4.1 CPUyears.

The main obstacle to proceeding to even larger graphs was the amount of RAM available. Our program will access different parts of the RAM in a very unpredictable way so a fast shared RAM is essential for this search, and few current computers have shared RAMs larger than 512 Gb.

4.2 Results

In Table 2 we have listed the diameters for all Cayley graphs which could be reached with the computational resources available to us. In tables 3–7 we have listed the sizes of the distance classes in the Cayley graphs, i.e., the number of vertices at a given distance from the identity matrix.

	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
$q=2$	2	4	7	10	13
$q=3$	3	6	9	12	
$q=4$	4	7	11		
$q=5$	4	7	11		
$7 \leqslant q \leqslant 23$	4	8			

Table 2: Diameter of the Cayley graph

As we can see from Table 2 the diameter is monotone in both n and q, as we expected, and we state this as a conjecture.

Conjecture 4.1. $\mathcal{D}(n,q)$ is monotone in both n and q.

If we look at the diameters of the graphs for $q=2$ in Table 2 we find that the main term of our asymptotic upper bound $\frac{n^2}{\log_2 n}$ in fact agrees remarkably well with the exact values for small n, predicting $4, 5, 8, 10, 13$ as the first few diameters. The sizes of the distance classes also agrees well with our concentration result. For $n=4,5$, tables 5 and 6, we see an exponential like drop in the size of the distance class as we move away from the largest one.

If we assume that Conjecture 4.1 is true, Table 3 gives us a complete display of the phenomena expected to appear as q increases for a fixed n. For $q=2,3$ the diameter of $\mathcal{D}(2,q)$ is still less than 4. For $q=4$ we find the first matrices requiring 4 row operations, and we have $q_a(2)=4$. As q continues to increase the last distance class continues to grow and for $q=13$ it contains more than half of the vertices, giving us $q_s(2)=13$. We have performed computations for larger q than those shown in this table as well, and as q continues to increase

$n=2$										
Level	$q=2$	$q=3$	$q=4$	$q=5$	$q=7$	$q=8$	$q=9$	$q=11$	$q=13$	$q=16$
0	1	1	1	1	1	1	1	1	1	1
1	3	7	11	15	23	27	31	39	47	59
2	2	23	54	103	239	326	431	679	983	1542
3		17	110	313	1249	2034	3161	6385	11257	22106
4			4	48	504	1140	2136	6096	13920	37492

Table 3: Size of the distance classes for $n = 2$

a larger and larger proportion of the vertices belongs to the last distance class, just as expected.

$n=3$							
Level	$q=2$	$q=3$	$q=4$	$q=5$	$q=7$	$q=8$	$q=9$
0	1	1	1	1	1	1	1
1	9	18	27	36	54	63	72
2	38	182	404	728	1634	2216	2900
3	78	1156	3968	9894	33968	53772	81058
4	42	4287	26046	93813	512307	952710	1671753
5		5130	92950	545628	5245120	11675814	24409482
6		458	57846	802306	21204546	63663690	171433796
7			198	35594	6785764	39018738	141869106
8					734	12708	187512

$n=3$						
Level	$q=11$	$q=13$	$q=16$	$q=17$	$q=19$	$q=23$
0	1	1	1	1	1	1
1	90	108	135	144	162	198
2	4526	6512	10160	11564	14630	21842
3	158844	275006	536516	653178	930548	1700256
4	4152327	8702397	21299670	27845457	44776143	100461507
5	78654196	202545280	629821474	888997108	1623054880	4509563860
6	798714832	2706903968	11370766138	17699483800	37684754564	134745058836
7	1232098786	6628001012	47971230510	83878534274	228871419044	1215099678186
8	10492398	179983508	4163519396	8707753322	36587912588	364987070066

Table 4: Size of the distance classes for $n = 3$

So far the exact values all agreed well with our expectations, however the data for $n = 3$ came as a surprise to us. As q increases from 2 to 7 the diameter of the Cayley graph rapidly grows from 4 to 8, however once that diameter has been reached the graphs seem very reluctant to rise any further. The computation for $n = 3$ was pushed to higher and higher q in the hope of being able to find the value of $q_a(3)$, probably the last $q_a(n)$ for which this is computationally feasible, but as the table shows we have not succeeded. When q was increased a larger and larger part of the vertices was found in the last three, and later the last two, distance classes, but not a single vertex appeared at distance 9. It is quite possible that there is some underlying algebraic property preventing matrices at large distance when q is small, relative n, but so far we have not found one. It would be interesting to find shaprper bounds for $q_a(n)$. By Theorem 2.4 we

Level	$n = 6$ $q = 2$
0	1
1	45
2	1075
3	18195
4	240934
5	2589042
6	22779975
7	161946260
8	893603745
9	3517544498
10	8207684400
11	6816796888
12	535485765
13	18937

Table 7: Size of the distance classes for $n = 6$

infinite sequence where $M_n \in \mathrm{GL}(n, q)$ is chosen independently and uniformly at random.

We have not been able to construct a family of the kind described, but for small n and q our program can produce the full set of extremal matrices, i.e. matrices M such that $\mathcal{D}(M) = \mathcal{D}(n, q)$. In Table 8 we have listed a few examples for $q = 2$. For each size we have picked matrices with minimal and maximal number of non-zero entries. For both $n = 3$ and $n = 4$, but not for $n = 5$, we also found that $J - I$ is extremal, where J is the all ones matrix. Note that since $\mathcal{D}(M) = \mathcal{D}(M^{-1})$ the inverses of all these matrices are also extremal matrices.

So, two natural problems still remain

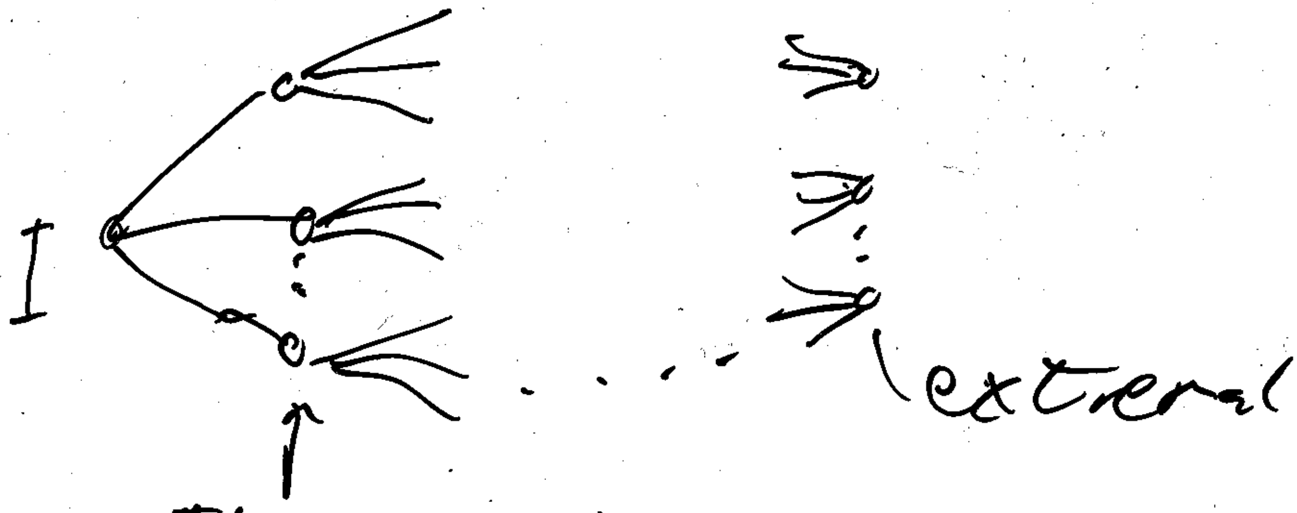
Problem 4.2. *Find an explicit construction for a sequence of matrices $\{M_n\}$, where M_n has side n, such that $n = o(\mathcal{D}(M_n))$*

Problem 4.3. *What is the complexity of computing $\mathcal{D}(M)$?*

5 Matrices over semifields

As we have seen, one impediment to the experimental side of our work has been the lack of computers with enough RAM to handle the very large graphs that are involved when q grows, so we have examined also other ways to vary the base field. One is to consider other algebraic structures than fields as domains from which to fetch the matrix elements. Such a change of domain also helps elucidate to which extent properties of the elimination problem depend on the algebraic structure of the chosen domain, rather than just its size.

Row operations are defined in terms of addition and multiplication, so those two operations are indispensable, which means the thing replacing the field will at least have to be some kind of ring. Furthermore the Gauss–Jordan algorithm, which provides the constant upper bound on the graph diameter, requires that there exists multiplicative inverses, so the thing replacing $\mathbb{F}_q$ should still be some kind of field. The first generalisation of the (commutative) field concept



the elemetary matrices

Why bother ??

integer factoring — encryption
RSA

$$\frac{n^2}{\log_2 n} \leq n^w \text{ for "small" } n$$

$n \leq 2 \cdot 10^7$

Dense

Lower bounds: Light version of "P = NP?"

Find an explicit sequence of matrices which require more than $c \cdot n$ operations

Matrix multiplication

Current project using computers to find algorithms

- Have found all Strassen-type algorithms for $n=2$
- Have proved that known methods are optimal for upper triangular 3×3 matrices
- Now working on other 3×3