CS 487/687 CM 730: Introduction to Symbolic Computation

Winter 2023

Instructor: Arne Storjohann <astorjoh@uwaterloo.ca>

Introduction

This course is an introduction to the use of computers for symbolic mathematical computation,

commonly called computer algebra. The types of mathematical computations of interest include

- simplifying rational expressions
- factoring polynomials
- solving linear equations
- solving polynomial equations
- integration of functions
- solving recurrences
- factoring integers and proving primality
- ...

The course is designed to expose students to:
1. the design and analysis of algorithms for algebraic problems;
2. the concepts from modern algebra which are applied to the development of algorithms for symbolic computation
3. the programming languages and data structures used for symbolic computation; and various applications of symbolic computation.

Next

Symbolic Mathematical Computation

In other words: How do we do mathematics on a computer?

```
> rnd:=rand(0..10^19):
> a:=rnd(); b:=rnd();
```

$$a := 2342493223442167775$$

(2.1)

$$b := 1799302827895858725 \tag{2.1}$$

```
> a+b;
```

$$41417960513380265 00 \tag{2.2}$$

```
> a*b;
```

$$42148546812663781410398404591475868 75 \tag{2.3}$$

```
> a^100;
```

$$\begin{aligned}
&928612060569280744940691709654158803204895271145968587319247508847519680 76\backslash \\
&537991655667946118309337465025877361302672233554414143061522213323249 70\backslash \\
&196593386509767322451244085325156475603679409413401147360111833371183 92\backslash \\
&606597476397712405777931923003743852913115447731285415715798769313137 43\backslash \\
&951300342262492151845805761929091462321368709981692362782681859985427 65\backslash \\
&962721748804468459362708063032497906332108928868105741862820698033723 28\backslash \\
&929490106358823158755472676141056002463000327502464606984032671355107 93\backslash \\
&039192469975383201095054624901468907347349861768285116849631410407628 90\backslash \\
&563774406465694003744896651542571300057300859569479743769329654189445 07\backslash \\
&926628703142184424592594823337023181758016777892137548011529063228648 78\backslash \\
&091707923854607500427421419479419635725598038165719268832329831070925 28\backslash \\
&333999171581360599393372009568843245361413793030523076428434819605001 70\backslash \\
&270108018811988823472372526665540432704326151167550860378037349699239 89\backslash \\
&388108894719956330351591815684596220574952829538149754296599948103677 06\backslash \\
&033778408077553863021449054552094333240712752123507988163724251569159 50\backslash \\
&654562233373055251760241051057703285031177617654404171278591310707066 02\backslash \\
&685282481427540270340152605039284088331782004074846358154957174687110 87\backslash \\
&962447117281733744056200005593368683023993859639771166335208471341421 00\backslash \\
&694646278928737326713443877517955529022704148255279432079947205333639 70\backslash \\
&659002022766117569666513355387536857019794165075484781250333035625771 30\backslash \\
&945606791417604098498952185618700477087067765460811785020019629176224 07\backslash \\
&820118998294066349110146411505820041770272997842797734473599759810931 72\backslash \\
&668115199788498639522189998615867752618078813439968412778426590859930 19\backslash \\
&017117473641458999779243330086629727639010321735150420797929538175757 84\backslash \\
&417249211127346961244012755923110635921275368087001675435880273007802 49\backslash \\
&853834221659276563012874738944901764625683426856994628906 25
\end{aligned} \tag{2.4}$$

Next

Solving Linear Systems of Equations Exactly

How do we solve systems of equations **exactly**?

```
> with(LinearAlgebra);
```

[*&x*, *Add*, *Adjoint*, *BackwardSubstitute*, *BandMatrix*, *Basis*, *BezoutMatrix*, *BidiagonalForm*, *BilinearForm*, *CARE*, *CharacteristicMatrix*, *CharacteristicPolynomial*, *Column*, *ColumnDimension*, *ColumnOperation*, *ColumnSpace*, *CompanionMatrix*, *CompressedSparseForm*, *ConditionNumber*, *ConstantMatrix*, *ConstantVector*, *Copy*, *CreatePermutation*, *CrossProduct*, *DARE*, *DeleteColumn*, *DeleteRow*, *Determinant*, *Diagonal*, *DiagonalMatrix*, *Dimension*, *Dimensions*, *DotProduct*, *EigenConditionNumbers*, *Eigenvalues*, *Eigenvectors*, *Equal*, *ForwardSubstitute*, *FrobeniusForm*, *FromCompressedSparseForm*, *FromSplitForm*, *GaussianElimination*, *GenerateEquations*, *GenerateMatrix*, *Generic*, *GetResultDataType*, *GetResultShape*, *GivensRotationMatrix*, *GramSchmidt*, *HankelMatrix*, *HermiteForm*, *HermitianTranspose*, *HessenbergForm*, *HilbertMatrix*, *HouseholderMatrix*, *IdentityMatrix*, *IntersectionBasis*, *IsDefinite*, *IsOrthogonal*, *IsSimilar*, *IsUnitary*, *JordanBlockMatrix*, *JordanForm*, *KroneckerProduct*, *LA_Main*, *LUDecomposition*, *LeastSquares*, *LinearSolve*, *LyapunovSolve*, *Map*, *Map2*, *MatrixAdd*, *MatrixExponential*, *MatrixFunction*, *MatrixInverse*, *MatrixMatrixMultiply*, *MatrixNorm*, *MatrixPower*, *MatrixScalarMultiply*, *MatrixVectorMultiply*, *MinimalPolynomial*, *Minor*, *Modular*, *Multiply*, *NoUserValue*, *Norm*, *Normalize*, *NullSpace*, *OuterProductMatrix*, *Permanent*, *Pivot*, *PopovForm*, *ProjectionMatrix*, *QRDecomposition*, *RandomMatrix*, *RandomVector*, *Rank*, *RationalCanonicalForm*, *ReducedRowEchelonForm*, *Row*, *RowDimension*, *RowOperation*, *RowSpace*, *ScalarMatrix*, *ScalarMultiply*, *ScalarVector*, *SchurForm*, *SingularValues*, *SmithForm*, *SplitForm*, *StronglyConnectedBlocks*, *SubMatrix*, *SubVector*, *SumBasis*, *SylvesterMatrix*, *SylvesterSolve*, *ToeplitzMatrix*, *Trace*, *Transpose*, *TridiagonalForm*, *UnitVector*, *VandermondeMatrix*, *VectorAdd*, *VectorAngle*, *VectorMatrixMultiply*, *VectorNorm*, *VectorScalarMultiply*, *ZeroMatrix*, *ZeroVector*, *Zip*] **(3.1)**

```
> n:=5;
```

$$n := 5 \quad \textbf{(3.2)}$$

```
> A:=RandomMatrix(n,n);
```

$$A := \begin{bmatrix} 43 & 50 & -22 & 87 & 27 \\ 25 & 10 & 45 & 33 & -93 \\ 94 & -16 & -81 & -98 & -76 \\ 12 & -9 & -38 & -77 & -72 \\ -2 & -50 & -18 & 57 & -2 \end{bmatrix} \quad \textbf{(3.3)}$$

```
> b:=RandomVector(n);
```

(3.4)

$$b := \begin{bmatrix} 77 \\ 9 \\ 31 \\ -50 \\ -80 \end{bmatrix} \tag{3.4}$$

```
>
```

```
> LinearSolve(A,b);
```

$$\begin{bmatrix} \frac{7115560949}{4814061204} \\ \frac{999581227}{1203515301} \\ \frac{2139870431}{2407030602} \\ -\frac{382156441}{1203515301} \\ \frac{1135087493}{1604687068} \end{bmatrix} \tag{3.5}$$

```
> B:=A[1..n,1..n]: B[1..n,1]:= b: Determinant(B)/Determinant(A);
```

$$\frac{7115560949}{4814061204} \tag{3.6}$$

What if A[3,3]=x and A[4,5]=y?

```
> A[3,3]:=x;
```

$$A_{3,\,3} := x \tag{3.7}$$

```
> A[4,5]:=y;
```

$$A_{4,\,5} := y \tag{3.8}$$

```
> A;
```

$$\begin{bmatrix} 43 & 50 & -22 & 87 & 27 \\ 25 & 10 & 45 & 33 & -93 \\ 94 & -16 & x & -98 & -76 \\ 12 & -9 & -38 & -77 & y \\ -2 & -50 & -18 & 57 & -2 \end{bmatrix} \tag{3.9}$$

```
> LinearSolve(A,b);
```

$$\begin{bmatrix} -\frac{43740\,x\,y+27917872\,x+19800140\,y-3683694917}{2\,(43050\,x\,y+14209873\,x-25851342\,y+1445666091)} \\ \frac{151662\,x\,y+55354336\,x-74721805\,y+2217563380}{2\,(43050\,x\,y+14209873\,x-25851342\,y+1445666091)} \\ \frac{5125966\,y+2508939983}{43050\,x\,y+14209873\,x-25851342\,y+1445666091} \\ \frac{5330\,x\,y+3816778\,x+4781219\,y-141990656}{43050\,x\,y+14209873\,x-25851342\,y+1445666091} \\ -\frac{1594342\,x-3276120777}{2\,(43050\,x\,y+14209873\,x-25851342\,y+1445666091)} \end{bmatrix} \quad \textbf{(3.10)}$$

```
> map(normal,%);
```

$$\begin{bmatrix} -\frac{43740\,x\,y+27917872\,x+19800140\,y-3683694917}{2\,(43050\,x\,y+14209873\,x-25851342\,y+1445666091)} \\ \frac{151662\,x\,y+55354336\,x-74721805\,y+2217563380}{2\,(43050\,x\,y+14209873\,x-25851342\,y+1445666091)} \\ \frac{5125966\,y+2508939983}{43050\,x\,y+14209873\,x-25851342\,y+1445666091} \\ \frac{5330\,x\,y+3816778\,x+4781219\,y-141990656}{43050\,x\,y+14209873\,x-25851342\,y+1445666091} \\ -\frac{1594342\,x-3276120777}{2\,(43050\,x\,y+14209873\,x-25851342\,y+1445666091)} \end{bmatrix} \quad \textbf{(3.11)}$$

What if n=100?

```
> A:=RandomMatrix(100,100);
```

$$A := \begin{bmatrix} 95 & 23 & -28 & -45 & -90 & -19 & 70 & 62 & -30 & -75 & \dots \\ 29 & 45 & -12 & -48 & 83 & -97 & 7 & -99 & 68 & -97 & \dots \\ 9 & 3 & -69 & -65 & 65 & -10 & 20 & -38 & -23 & -84 & \dots \\ 71 & 67 & 18 & 7 & 33 & 66 & -66 & 24 & -57 & -22 & \dots \\ -59 & -37 & 86 & 21 & 65 & -61 & 96 & 99 & 5 & 3 & \dots \\ 57 & -66 & -2 & -82 & -95 & 18 & 15 & 76 & 58 & 32 & \dots \\ 64 & -43 & -99 & -44 & -68 & 59 & 59 & 50 & -10 & 33 & \dots \\ -74 & -57 & -64 & 43 & 13 & -84 & -87 & 64 & -56 & 48 & \dots \\ -91 & -62 & -8 & 18 & -2 & -92 & -61 & -77 & 50 & 73 & \dots \\ -30 & -31 & 66 & 72 & -86 & 49 & -40 & 62 & 70 & 47 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix} \quad \textbf{(3.12)}$$

100 × 100 Matrix

```
> b:=RandomVector(100);
```

$$b := \begin{bmatrix} -16 \\ -66 \\ 25 \\ -49 \\ 31 \\ -27 \\ -23 \\ 56 \\ -37 \\ 13 \\ \vdots \end{bmatrix} \qquad \textbf{(3.13)}$$

100 element Vector[column]

```
> LinearSolve(A,b);
```

$$\left[-\frac{217500887986519882647330458360199119515155749136107390062369960236775955250844318502}{323446851452223842505177090037004212099078909195372065201032036143407795021685726627}, \right.$$

$$\frac{814708172801443124683091664181054034068072309046582283710193992443650438991257158652}{323446851452223842505177090037004212099078909195372065201032036143407795021685726627},$$

$$\frac{643773972034212020799068469880044078603498678628798651017828131297195043796419638315}{107815617150741280835059030012334737366359636398457355067010678714469265007228575542},$$

$$\frac{322897704143466763399420677537615388555194916911017110582815305198073801694975440415}{107815617150741280835059030012334737366359636398457355067010678714469265007228575542},$$

$$-\frac{332649699166465559471639429608856777828963179793230440353098983962850821735922830 15}{107815617150741280835059030012334737366359636398457355067010678714469265007228575542},$$

$$\frac{120300439564512452008290277697801154387037079665960909093166247481354982109338376211}{107815617150741280835059030012334737366359636398457355067010678714469265007228575542},$$

$$\frac{500107454200150396209521269522326767266223921072605917327731932932164660641036541050}{359385390502470936116863433374449124554532121328191183556702262381564216690761918475},$$

$$-\frac{239053598158634617657798731298274372711179686804285252345307753740852623020781082018}{323446851452223842505177090037004212099078909195372065201032036143407795021685726627},$$

$$-\frac{705694303009353821162497398202343580067585468957797936521854124624579660087119658934}{323446851452223842505177090037004212099078909195372065201032036143407795021685726627},$$

$$\left. \frac{166711151185230929650737129512828041302805662627900101689926307792671892588297382062}{107815617150741280835059030012334737366359636398457355067010678714469265007228575542} \right]$$

```
> %[1];
```

(3.15)

$$-21750088798651988264733045836019911951515574913610739006236996023677595525\backslash$$
$$08443185025364885555498581217681900220908043272040853805511493912325050\backslash$$
$$25574194196159968577585795842530882671848403558592505067082854890352069\backslash$$
$$85602091807017047265467947348239641434 4 \Big/$$
$$32344685145222384250517709003700421209907890919537206520103203614340779\backslash$$
$$50216857266277045308172758580938119190727855471486624590805865198949380\backslash$$
$$02369299263516968804180975162082333994892205112073937595367718238177909\backslash$$
$$91545350297199910365391056728337236337323 \quad (3.15)$$

Next

Factoring Polynomials

Consider the problem of factoring polynomials
• This is pretty fundamental: it takes a large problem and breaks it into smaller ones

```
> f:=x^23-x^16-2*x^13+3*x^20-2*x^10+2*x^11-2*x^4+2*x+5*x^3-5;
```

$$f := x^{23} + 3\,x^{20} - x^{16} - 2\,x^{13} + 2\,x^{11} - 2\,x^{10} - 2\,x^4 + 5\,x^3 + 2\,x - 5 \quad (4.1)$$

```
> factor(f);
```

$$\left(x^{10} - x^3 + 1\right)\left(x^{13} + 3\,x^{10} + 2\,x - 5\right) \quad (4.2)$$

• How do we go about performing this operation?
• How do we represent the polynomials in question?
• How much does this cost?

Sometimes it is easier to use a "modular algorithm"
• Compute modulo small primes, then "reconstruct" the integer or rational solution
• This will be a standard approach.
• For example, start by factoring **f** modulo a small prime **p**.

```
> p:=13;
```

$$p := 13 \quad (4.3)$$

```
> Factor(f) mod p;
```

$$\left(x^9 + 7\,x^8 + 10\,x^7 + 5\,x^6 + 9\,x^5 + 11\,x^4 + 12\,x^3 + 5\,x^2 + 9\,x + 11\right)\left(x^8 + 7\,x^6 + 10\,x^5 + 4\,x^4 + 11\,x^3 + 4\,x^2 + 4\,x + 2\right)\left(x + 10\right)\left(x^4 + 3\,x^3 + 2\,x^2 + 12\,x + 3\right)\left(x + 6\right) \quad (4.4)$$

• Looks a bit weird (where did all those factors come from), but it will help.

What about a "fancier" f:

```
> f:=x^3*y^2+x^6-3*x*y^3-3*x^4*y+15*x*y+5*y^2-25;
```

$$f := x^6 - 3\,x^4\,y + x^3\,y^2 - 3\,x\,y^3 + 15\,x\,y + 5\,y^2 - 25 \quad (4.5)$$

```
> factor(f);
```

$$\left(x^3 - 3\,x\,y + 5\right)\left(x^3 + y^2 - 5\right) \quad (4.6)$$

Next

Solving Polynomial Equations

In many applications, we often have a collection of multivariate polynomials

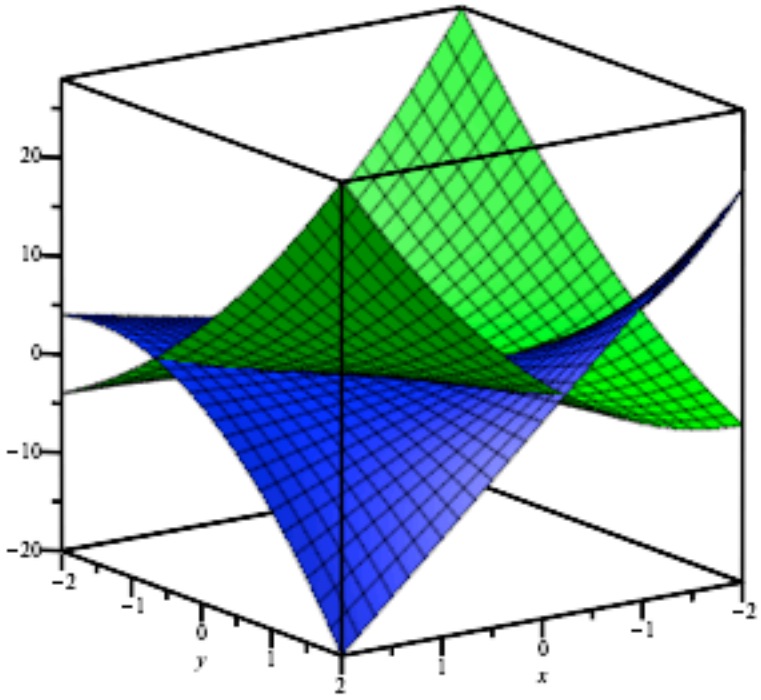
```
> f:=-x*y^2-3*x*y;
```

$$f := -x\,y^2 - 3\,x\,y \tag{5.1}$$

```
> g:=2*x^2+4*x*y+y^2;
```

$$g := 2\,x^2 + 4\,x\,y + y^2 \tag{5.2}$$

```
> plot3d([f,g],x=-2..2,y=-2..2,color=[blue,green]);
```




```
> solve(f=g);
```

$$\left\{x = \left(-\frac{y}{4} - \frac{7}{4} + \frac{\sqrt{y^2+14\,y+41}}{4}\right) y,\ y = y\right\},\ \left\{x = \left(-\frac{y}{4} - \frac{7}{4} - \frac{\sqrt{y^2+14\,y+41}}{4}\right) y,\ y = y\right\} \tag{5.3}$$

```
> solve({f=0,g=0});
```

$$\{x = 0, y = 0\},\ \{x = 3\ RootOf\left(2\ \_Z^2 - 4\ \_Z + 1\right), y = -3\} \tag{5.4}$$

We will (start to) explore the techniques use to solve these systems of polynomial equations.

Next

Computational Number Theory

Some important operations with integers are "hard" (at least in one direction) which makes them very useful.

• Consider the problem of multiplying and factoring integers.

• Multiplying integers is pretty straightforward (though there are ways to do better)

```
> a:=nextprime(2^85); b:=nextprime(a); n:=a*b;
```

$$a := 38685626227668133590597803$$

$$b := 38685626227668133590597869$$

$$n := 1496577676626844588240589052436974700726179887881807 \quad \textbf{(6.1)}$$

We can check if `n;` is prime (it better not be):

```
> isprime(n);
```

$$false \quad \textbf{(6.2)}$$

We can also attempt to factor n:

```
> ifactor(n);
```

$$(38685626227668133590597803)\,(38685626227668133590597869) \quad \textbf{(6.3)}$$

Hmmm: that seems to take a bit longer...

The difficulty of algorithms for factorization and related operations is the foundation on which much of modern cryptography rests.

How hard are these problems and why?

Next

Focus of CS 487/687

Algorithm design and analysis

Computer science

- fundamental algorithm design for algebraic problems
- careful analysis of these algorithms, counting "real" computational cost, and also more abstract notions (like counting field and ring operations)
- efficient implementation (sequentially here)
- computer algebra system use and design
- applications

Mathematics

- More algebra than analysis (at least in this course)
- Mathematics not terribly difficult, but we may use it in unusual ways
- Brush up on your Math 135 (Algebra),, Math 136 (Linear Algebra) and Math 239 (Combinatorics)

We will also get familiar with some modern computer algebra systems (mostly Maple)

Next

Course operations

Take a look at the course website: http://www.student.cs.uwaterloo.ca/~cs487
- This is your primary source of information.
- You are responsible for keeping up-to-date with current course events through this site.
- I will keep it updated with the latest information.

Let's go there now.

We will use Piazza. Not been set up yet... maybe in a week.
- Ask general questions, and please be careful not to post solutions!

Handouts
- Handouts will be posted on the website as a combination of typeset notes, slides, Maple worksheets and other code
- You are encouraged to execute and experiment with all examples and code
- Some material will only be given in class

Marking Scheme CS487
- Final Exam 40%
- Midterm Exam 25%
- Assignments 35%

Marking Scheme CS687
- Final Exam 30%
- Midterm Exam 20%
- Assignments 30%
- Project 20%

Midterm Exam
Proposed date: TBA, in class

Assignments
- Around 5 Assignments, combination of algorithm development, analysis, and coding
- This is not a course in Maple, Sage, C++, Python, etc. You will be expected to pick up the necessary coding skills in appropriate languages

CS687 Projects
- A list of possible CS 687 potential will be posted in February
- Students are encouraged to come up with their own project (related to the course topics)

Next

Topics

Some topics we will hopefully cover in CS487/CS687 are the following:
- Representation and basic arithmetic in algebraic domains (extended integers, polynomials)
- Division with remainder.
- GCD and the extended euclidean algorithm.
- Solving integer linear equations.

- Modular computation.
- Evaluation/Interpolation and the Chinese remainder algorithm.
- Rational system solving (Gaussian elimination vs. p-adic lifting)
- Fast arithmetic: multiplication and division.
- Newton iteration in algebraic domains.
- Factoring polynomials over finite fields.

Next

Representation of multi-precision integers

Will follow von zur Gathen & Gerhard text: Chapters 2 and 3, and sections of Chapters 4 and 5.

Current computers are based on architecture with 32- or 64-bit bus and words
- let's assume the word size is 64 bits
- e.g. "unsigned long" in C can represent integers exactly in range $[0 \ldots 2^{64} - 1]$
- any integer a (and I mean ***any***) can be written as

$a = (-1)^s \sum a_i \; 2^{64\,(i-1)}$ where $a_1, \ldots, a_d$ are the base-2^{64} digits of a, $0 \le a_i < 2^{64}$.

- For example

```
> a:=7^1234-1;
```

$a :=$ **(10.1)**

```
709547342150280112401045951414989290704202892003652548648831002244590615\
310959645359614247521839189833898347143458830857225946301043275704843\
813454950959834721299254363358098967399048133489131401375954827777487\
406893326596849496141162311973467935106954023011568404580579255041439\
679213040737846060014251249690928324441226158708348216074516368113070\
867549448732771436782166293866108044083497082337665021608188274414316\
929628234788516218347261590078027886099815418189756002705037098111458\
971874252832721423453899821462438727851069669041332053765106227148570\
187299800867412862748598106982453334317600731675149728049405799520143\
980455756911948283115787637760244671160282277683668276830047909392655\
398432923069244698164607404132404210543306184313774389935234011995625\
061172722638875502809299661809066729592269849797551584071055050025857\
377500767193512061381492022804530342327575834054744001006721917003325\
347912741669538744924043142095605796598736664375513698661239721232174\
765476802494126679637033993978064101736963494084 8
```

```
> A:=convert(a,base,2^64);
```

$A :=$ [383791937716500267 2, 1582321422791912747 7, 1557595545663285976, **(10.2)**
1490141813151677427 2, 1484048580927846581 8, 649051941957981856 4,
1790386243688822517 7, 958101414644813190, 1732793315817394557 7,
125560357773759646 62, 997387260917339658 5, 410984068493362727,
156366926828773746 36, 180811511442761525 85, 169894687698552816 25,
604810671906273889 6, 110478423253080621 08, 457586066138932667 3,

8812381446037964, 852542775141951 4158, 3730562961967692288,
1393564268353096 0953, 708005582523357863, 4741827300631391529,
7106182707189131278, 8670310217081208842, 10336838047443121359,
14248549772469055719, 7678259036781962664, 13517997778174320394,
3612882182354918784, 5154029295038924509, 107813727573049100,
16226912028583090698, 3488717966611906294, 15535020941492499797,
12916136377218426903, 695241936856213589, 13978353063432943831,
12757692631210613235, 17326290772922491249, 16757614229303912379,
11418648298192846008, 8218818804729114829, 3875046642394532808,
14208010283725884564, 13721441723524650979, 2057749185813976201,
12070457865020334715, 7168507067317368568, 7421895996676334322,
4129021570641193567, 5115705272123923548, 17849888354679997456, 309]

```
> b:=add(A[i]*2^(64*(i-1)),i=1..nops(A));
```

$b :=$ **(10.3)**

709547342150280112401045951414989290704202892003652548648831002244590 61\
531095964535961424752183918983389834714345883085722594630104327570484 38\
281345495095983472129925436335809896739904813348913140137595482777748 72\
406893326596849496141162311973467935106954023011568404580579255041439 47\
679213040737846060014251249690928324441226158708348216074516368113070 76\
867549448732771436782166293866108044083497082337665021608188274414316 15\
929628234788516218347261590078027886099815418189756002705037098111458 69\
971874252832721423453899821462438727851069669041332053765106227148570 29\
187299800867412862748598106982453334317600731675149728049405799520143 21\
980455756911948283115787637760244671160282277683668276830047909392655 68\
398432923069244698164607404132404210543306184313774389935234011995625 09\
061172722638875502809299661809066729592269849797551584071055050025857 54\
377500767193512061381492022804530342327575834054744001006721917003325 60\
347912741669538744924043142095605796598736664375513698661239721232174 69\
765476802494126679637033993978064101736963494 0848

```
> a-b;
```

0 **(10.4)**

The de-facto standard for integer arithmetic is GMP (GNU Multi-Precision).

This is an open-source C library (with assembly language kernels) which is used by most computer algebra and computational number theory systems.

• See http://gmplib.org/

Next

Asymptotic notation

We will use big-Oh (O(..)), big-Omega (Ω(...)) and big-Theta (Θ(...)).

Be familiar with it...

Next

Addition of multiprecision numbers (radix B)

Input: $a_0 + a_1 B + a_2 B^2 + \ldots + a_m B^m$

$+ b_0 + b_1 B + b_2 B^2 + \ldots + b_m B^m$

$= (a_0 + b_0) + (a_1 + b_1) B + (a_2 + b_2) B^2 + \ldots + (a_m + b_m) B^m$

Note that $a_0 + b_0 = c_0 + \gamma_0 B$ (where $\gamma_i \in \{0, 1\}$).

Furthermore $a_1 + b_1 + \gamma_0 = c_1 + \gamma_1 B$ (where $\gamma_1 \in \{0, 1\}$).

So the carry is either 0 or 1.

Cost of adding two integers of length m is bounded by the cost of adding m digits together (plus the carries).

Assuming the digits are of constant size (which is reasonable), the cost is $O(m)$.

Next

Representation and addition of polynomials

Polynomials have many similar properties to integers, but also many differences.

```
> f:=3*x^5+12*x+2;
```

$$f := 3x^5 + 12x + 2 \quad \textbf{(13.1)}$$

```
> g:=2*x^6+4*x;
```

$$g := 2x^6 + 4x \quad \textbf{(13.2)}$$

```
> f+g;
```

$$2x^6 + 3x^5 + 16x + 2 \quad \textbf{(13.3)}$$

```
> sort(%);
```

$$2x^6 + 3x^5 + 16x + 2 \quad \textbf{(13.4)}$$

Coefficients 3, 12, 2, etc. come from the ring of integers
(in this course a ring is always a commutative ring with identity).

We will see other rings later (e.g. integers modulo a composite number N)

A polynomial $f \in R[x]$, over a ring R, is a finite sequence $f_0, f_1, \ldots, f_n$ such that

$f = f_0 + f_1 x + \ldots + f_n x^n$.

If $f \neq 0$ the we assume that $f_n \neq 0$ and the degree of f is n.

Dense representation

- The dense representation of a polynomial is simply a list or array of all its coefficients, in order
- For example $f = f_0 + f_1 x + \ldots + f_n x^n$ is represented as $[f_0, f_1, f_2, \ldots, f_n]$, an array of $n + 1$ elements of R.

```
> f;
```

$$3 x^5 + 12 x + 2 \qquad \textbf{(13.1.1)}$$

```
> PolynomialTools[CoefficientVector](f,x);
```

$$\begin{bmatrix} 2 \\ 12 \\ 0 \\ 0 \\ 0 \\ 3 \end{bmatrix} \qquad \textbf{(13.1.2)}$$

- This uses the function (CoefficientVector) in the PolynomialTools package

Sparse representation

- The sparse representation of a polynomial is a list of coefficient/exponent pairs
- So $\sum_{i=1}^{t} a_i x^{e_i}$ would be represented as $(a_1, e_1), (a_2, e_2), \ldots, (a_t, e_t)$

```
> f;
```

$$3 x^5 + 12 x + 2 \qquad \textbf{(13.2.1)}$$

```
> C:=[coeffs(f,x,'V')];
```

$$C := [3, 12, 2] \qquad \textbf{(13.2.2)}$$

```
> V;
```

$$x^5, x, 1 \qquad \textbf{(13.2.3)}$$

```
> seq([C[i],degree(V[i])],i=1..nops(C));
```

$$[3, 5], [12, 1], [2, 0] \qquad \textbf{(13.2.4)}$$

Addition

In symbolic computation we often use other representations, some of which we will see here

- Black box representation – representing symbolic objects as functions which produce values
- Straight-line programs or algebraic circuits – representing symbolic objects as programs (which produce values and can be manipulated as programs)

Addition of two polynomials with degree bounded by n costs at most $n+1$ additions in R.
- We will usually analyze algorithms over a ring R in terms of the number of operations from R.
- I.e., we will write $O(n)$ operations in R.

What is the cost in word operations if $R = \mathbb{Z}$?
- In this case we count machine operations and must take into account the size of the coefficients in $\mathbb{Z}$.
- Assume that the absolute values of all coefficients of f are bounded by B (i.e., $|f_i| \le B$)
- Additions with integers of absolute value $\le B$ take $O(\log(B))$ operations.
- Cost two add two polynomials of degree $\le n$ with coefficients of absolute value $\le B$ is $O(n \log(B))$ word operations.

Note: similarities and differences between $\mathbb{Z}$ and R[x].
- no carries to worry about in R[x]
- many algorithms for R[x] carry over to $\mathbb{Z}$.
- not all techniques in R[x] work in $\mathbb{Z}$.

Polynomial Reversion

$$f = f_0 + f_1 x + \ldots + f_n x^n \Rightarrow reversal(f) = f_n + f_{n-1} x + \ldots + f_0 x^n = f\left(\frac{1}{x}\right) \cdot x^n$$

Note that $reversal(reversal(f)) = f$. No analogue for integers.
This operation turns out to be very useful.

```
> f;
```

$$3x^5 + 12x + 2 \qquad \textbf{(13.4.1)}$$

```
> eval(f,x=3);
```

$$767 \qquad \textbf{(13.4.2)}$$

```
> frev:=expand(x^degree(f,x)*eval(f,x=1/x));
```

$$frev := 2x^5 + 12x^4 + 3 \qquad \textbf{(13.4.3)}$$

Next

Polynomial Multiplication

First consider polynomials in $R[x]$.

$f = f_0 + f_1 x + f_2 x^2 + \ldots + f_n x^n$ and $g = g_0 + g_1 x + g_2 x^2 + \ldots + g_m x^m$

The $f \cdot g = h_0 + h_1 x + h_2 x^2 + \ldots + h_{n+m} x^{n+m}$ where $h_k = \sum_{i+j=k} f_i \cdot g_j$

The cost is $(n+1)(m+1)$ multiplications and $n \cdot m$ additions (exactly).

$\Theta(nm)$ operations in R.

Integers are similar to above.

- We can multiply $a \cdot b$ with $O(\log(a)\log(b))$ machine operations.
- Often we will call these ***bit operations*** - the difference between a 32-bit word or a 64-bit word (or even a 1-bit word) only matters "up to a constant" (which is hidden in the big-Oh).
- Of course, these constants matter a lot in practice!

```
> f:=randpoly(x,degree=20);
```

$$f := -52\,x^{18} - 41\,x^{15} - 65\,x^{14} + 94\,x^{11} - 42\,x^{10} + x^{2} \tag{14.1}$$

```
> g:=randpoly(x,degree=30);
```

$$g := -49\,x^{23} + 14\,x^{21} + 9\,x^{15} - 36\,x^{12} + 70\,x^{10} - 60\,x^{3} \tag{14.2}$$

```
> h:=f*g;
```

$$h := \left(-52\,x^{18} - 41\,x^{15} - 65\,x^{14} + 94\,x^{11} - 42\,x^{10} + x^{2}\right)\left(-49\,x^{23} + 14\,x^{21} + 9\,x^{15} - 36\,x^{12} + 70\,x^{10} - 60\,x^{3}\right) \tag{14.3}$$

```
> h:=expand(h);
```

$$\begin{aligned} h := {} & 2548\,x^{41} - 728\,x^{39} + 2009\,x^{38} + 3185\,x^{37} - 574\,x^{36} - 910\,x^{35} - 4606\,x^{34} \\ & + 1590\,x^{33} + 1316\,x^{32} - 588\,x^{31} + 1503\,x^{30} - 585\,x^{29} - 3640\,x^{28} + 1476\,x^{27} \\ & + 3186\,x^{26} - 3297\,x^{25} - 4550\,x^{24} - 3370\,x^{23} + 1512\,x^{22} + 9700\,x^{21} - 2940\,x^{20} \\ & + 2460\,x^{18} + 3909\,x^{17} - 5676\,x^{14} + 2520\,x^{13} + 70\,x^{12} - 60\,x^{5} \end{aligned} \tag{14.4}$$

Note the use of **expand**. Why do we need this?

```
> (x-1)^1000;
```

$$(x-1)^{1000} \tag{14.5}$$

```
> w:=expand((x-1)^1000):
> coeff(w,x,500);
```

$$\begin{aligned} & 270288240945436569515614693625975275496152008446548287007392875106625428701 \\ & 5522193898612483924502370165362606085021546104802209750050679917549894219699518475423665484263751733356162464079737887344364574161119497604571044985756287880514600994219426752366915856603136862602484428109296905863799821216320 \end{aligned} \tag{14.6}$$

Multivariate polynomials

- Polynomials often have many variables
- Mathematically we write $R[x_1, x_2, \ldots, x_m]$ for the ring of polynomials in variables $x_1, x_2, \ldots, x_m$

Recursive representation of multivariate polynomials

- $R[x_1, x_2, \ldots, x_m] = R[x_1, x_2, \ldots, x_{m-1}][x_m]$

```
> f:=x^3*y^2+2*x^2*y^2+3*x*y-4*y;
```

$$f := x^3 y^2 + 2\,x^2 y^2 + 3\,x\,y - 4\,y \tag{14.1.1}$$

```
> collect(f,y);
```

$$\left(x^3 + 2\,x^2\right) y^2 + (3\,x - 4)\,y \qquad \textbf{(14.1.2)}$$

- We will see other representations as the course progresses

Next

Division with remainder

Given $f, g \in R[x]$ with $g \neq 0$, find $q, h \in R[x]$ so that

$$f = q\,g + h$$

and $deg\,h < deg\,g$.

Not always possible for all rings R.

- What if $f = x^2$ and $g = 2\,x + 1$ and R=$\mathbb{Z}$

```
> rem(x^2,-x+1,x);
```

$$1 \qquad \textbf{(15.1)}$$

The division algorithm:
Assume that the leading coefficient of f is a unit in R (i.e., it has an inverse).

Classical school method: cancel the first coefficient of f and continue.

$f = f_0 + f_1 x + f_2 x^2 + \ldots + f_n x^n$ and
$g = g_0 + g_1 x + g_2\,x^2 + \ldots + g_m x^m$

with $n \geq m$.

Look at $f^{(n-1)} = f - \left(\frac{f_n}{g_m}\right) x^{n-m}\,g$.

- $deg\left(f^{(n-1)}\right) < n$ since the highest degree term is cancelled.
- $rem(f\,,\,g) = rem\left(f^{(n-1)}, g\right)$.
- $quo(f, g) = \left(\frac{f_n}{g_m}\right) x^{n-m} + quo\left(f^{(n-1)}, g\right)$.

Here it is in Maple:

```
> f:=x^5-3*x^4+2*x^3-2*x^2+5*x+1;
```

$$f := x^5 - 3\,x^4 + 2\,x^3 - 2\,x^2 + 5\,x + 1 \qquad \textbf{(15.2)}$$

```
> g:=x^3+2*x^2-5*x+4;
```

$$g := x^3 + 2\,x^2 - 5\,x + 4 \qquad \textbf{(15.3)}$$

```
> c:=lcoeff(f,x)/lcoeff(g,x)*x^(degree(f,x)-degree(g,x));
```

$$c := x^2 \qquad \textbf{(15.4)}$$

```
> ff4:=expand(f-c*g);
```

$$ff4 := -5x^4 + 7x^3 - 6x^2 + 5x + 1 \quad \textbf{(15.5)}$$

```
> rem(f,g,x);
```

$$-65x^2 + 110x - 67 \quad \textbf{(15.6)}$$

```
> rem(ff4,g,x);
```

$$-65x^2 + 110x - 67 \quad \textbf{(15.7)}$$

```
> quo(f,g,x);
```

$$x^2 - 5x + 17 \quad \textbf{(15.8)}$$

```
> quo(ff4,g,x);
```

$$-5x + 17 \quad \textbf{(15.9)}$$

```
>
```

Try it when the leading coefficient of g is not ± 1.

What does division with remainder mean for multivariate polynomials?