Experimental Mathematics 2020: Lab sheets

Alex Ghitza

Version of Fri 29th May, 2020 at 09:47

1 Introduction to Mathematica (13 March)

Open a new Mathematica notebook and follow along.

Exercise 1.1. Work through

https://www.wolfram.com/language/fast-introduction-for-math-students/en/entering-input/

Note the differences in syntax between Mathematica lists and Sage (Python, really) lists.

Also note the different behaviour of `Range`.

Exercise 1.2. Work through

https://www.wolfram.com/language/fast-introduction-for-math-students/en/fractions-and-decimals/

What is the 5-th digit after the decimal point of the number π^3?

What is the millionth digit after the decimal point of the number π^3?

Exercise 1.3. Work through

https://www.wolfram.com/language/fast-introduction-for-math-students/en/variables-and-functions/

On that page, click on the link Defining Variables and Functions to see a more clear explanation of the difference between `=` and `:=`.

Use the function definition syntax to define the two recursive sequences a and b that converge to the agM of $\sqrt{2}$ and 1.

Compute the first few terms in each sequence.

How many terms do you need in order to get 100 correct digits of the limit?

How many terms do you need in order to get ten thousand correct digits of the limit?

Exercise 1.4. Work through

https://www.wolfram.com/language/fast-introduction-for-math-students/en/sequences-sums-and-series/

Use syntax from that page to define the recursive sequence giving the number of regions cut by n lines in the plane:

$$R(0) = 1, \qquad R(n) = R(n-1) + n.$$

Compute the first few values. Get Mathematica to tell you a formula for the general term of the sequence.

Exercise 1.5. Work through

https://www.wolfram.com/language/fast-introduction-for-math-students/en/plots-in-2d/

https://www.wolfram.com/language/fast-introduction-for-math-students/en/more-plots-in-2d/

https://www.wolfram.com/language/fast-introduction-for-math-students/en/plots-in-3d/

Search the Mathematica documentation for the function `ListPointPlot3D`.

Consider the recursive formula

$$a_0 = x,$$
$$a_n = \frac{1}{2}\left(a_{n-1}^2 + y^2\right)$$

where x and y are fixed parameters.

Compute the first few values a_n for various x and y.

Use `ListPointPlot3D` to get a scatter plot of these values.

Based on the plot, formulate a conjecture regarding the region in the x-y plane where the recursion converges to a finite limit.

Exercise 1.6. Work through

https://www.wolfram.com/language/fast-introduction-for-math-students/en/matrices-and-linear-algebra/

Define the matrix

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

We want to describe all 3×3 matrices B that commute with A.

Work through

https://www.wolfram.com/language/fast-introduction-for-math-students/en/algebra/

Define the matrix of unknowns

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

and ask Mathematica to solve the equation $AB - BA = 0$.

Did you get some conditions on the coefficients b_{ij}?

Good. Now we push this a bit further. I claim that there exist α, β, γ (depending on B) such that

$$B = \alpha A^2 + \beta A + \gamma I.$$

(In other words, that B must be a quadratic polynomial in A.)

Use Mathematica's equation solving capabilities to find α, β, γ in terms of the b_{ij}'s.

Exercise 1.7. Browse through the Mathematica demonstrations at

https://demonstrations.wolfram.com/topic.html?topic=Experimental+Mathematics

2 Constant recognition (20 March)

2.1 Recognising integers

Exercise 2.1. Consider the real number

$$\alpha = \frac{\pi^{10}}{\zeta(10)}$$

Compute α to default precision and guess what the exac value of α might be.

Increase the precision a few times to see if your guess persists.

Exercise 2.2. Consider the real number

$$\alpha = \left(e^{\pi\sqrt{163}} - 744\right)^{1/3}$$

Compute α to default precision and venture a guess about the value of α.

Increase the working precision and recompute α. Do you need to adjust your guess? If not, maybe increase the precision some more.

Think about what strategies you might employ to check whether what you are seeing is just a numerical glitch.

A good approach to this type of question relies on interval arithmetic.

There is some information about interval arithmetic in Mathematica at

https://reference.wolfram.com/language/tutorial/Numbers.html

For Sage, see

https://doc.sagemath.org/html/en/prep/Quickstarts/NumAnalysis.html

which also has other useful hints for working with real numbers.

2.2 Recognising rational numbers

Exercise 2.3. Compute the real number

$$\alpha = \frac{\zeta(20)}{\pi^{20}}$$

and determine experimentally whether it is a rational number. (Use continued fractions.)

Exercise 2.4. Same as above, with

$$\alpha = \frac{\zeta(3)}{\pi^3}$$

Even today, there is much that we (the human race) do not know about the values of ζ at odd integers.

2.3 Lattices

Exercise 2.5. Write a Sage function `lattice2d` with signature

```
def lattice2d(v1, v2, range1, range2):
```

that returns a list of the points in the lattice with basis {`v1, v2`} whose first coefficient is in the range `range1` and the second is in the range `range2`.

Experiment with your function and the square lattice (basis $(1, 0)$ and $(0, 1)$). Use `list_plot` or `scatter_plot` or `points` to visualise the lattice.

Try a couple more lattices.

Exercise 2.6. Repeat the previous exercise, this time in three-dimensional space.

Exercise 2.7. Compare the plots for `lattice2d` for the lattices given by

- $(1, 0)$ and $(0, 1)$, both ranges `range(-3, 4)`
- $(2, 3)$ and $(3, 5)$, both ranges `range(-30, 31)`

You may want to restrict the viewing window for the second plot:

```
points(lst, xmin=-3, xmax=3, ymin=-3, ymax=3)
```

Exercise 2.8. This does not really involve computation, it's more of a pen and paper thing.

What kind of matrices give a valid change of basis for the vector space $\mathbb{R}^2$?

What kind of matrices give a valid change of basis for a lattice L inside the vector space $\mathbb{R}^2$?

3 More constant recognition (3 April)

Get Sage started.

3.1 Some lattice work

Exercise 3.1. Reproduce the LLL calculation from Example 3.4 in the lecture notes.

Exercise 3.2. In Example 3.4, check that the first column $\mathbf{b}_1$ is indeed a shortest nonzero vector in the lattice.

Now check that the second column $\mathbf{b}_2$ is a shortest lattice element not in $\mathrm{Span}_{\mathbb{Z}}(\mathbf{b}_1)$.

Is the third column $\mathbf{b}_3$ a shortest lattice element not in $\mathrm{Span}_{\mathbb{Z}}(\mathbf{b}_1, \mathbf{b}_2)$?

3.2 Discovering integer relations

Exercise 3.3. Reproduce the calculations from Example 3.2 in the lecture notes.

It would be nice to experiment with changing the value of the multiplier A, but it's a pain to have to run the same commands again and again. So start by writing a function with signature

```
def intrel(alst, A):
```

that takes as input a list `alst` of real numbers and an integer multiplier `A` and returns the coefficients of the approximate integer relation given by LLL.

Test your function on Example 3.2 with $A = 10^6$ to see if it works.

Now experiment with different values of A (say $A = 10^1, 10^2, 10^7, 10^8$).

Exercise 3.4. Run an experiment to see if there is an integer relation between the real numbers 1, e, and π.

3.3 Recognising algebraic numbers

An *algebraic integer* is a root α of a monic polynomial with integer coefficients:

$$f(\alpha) = 0, \qquad f(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0, \quad a_i \in \mathbb{Z}.$$

Given an algebraic integer α, its *minimal polynomial* is the monic integer polynomial f of smallest degree d such that $f(\alpha) = 0$. We refer to d as the *degree of* α.

For instance, $\alpha = \sqrt{2}$ has minimal polynomial $x^2 - 2$.

Exercise 3.5. Suppose you have a real number α and you think it might be an algebraic integer of degree d. Reduce the problem of finding the minimal polynomial of α to an instance of the integer relation problem.

Exercise 3.6. Apply your strategy from the previous exercise to find the minimal polynomial for

$$\alpha = 1 + \sqrt{1 + \sqrt{1 + \sqrt{2}}},$$

which seems like it might have degree $\leqslant 8$, doesn't it?

Exercise 3.7. Let $\mathcal{H} = \{z \in \mathbb{C} \mid \operatorname{Im}(z) > 0\}$. The *Dedekind eta function* $\eta\colon \mathcal{H} \to \mathbb{C}$ is defined by

$$\eta(z) = q^{1/24} \prod_{n=1}^{\infty} (1 - q^n), \qquad q = e^{2\pi i z}$$

It is implemented in Sage as `eta`.

Consider the number

$$\alpha = \frac{\eta^2\left(\sqrt{-5}/2\right)}{2\eta^2\left(2\sqrt{-5}\right)}$$

Figure out whether α may be algebraic. If yes, what is its minimal polynomial (and degree)?

Exercise 3.8. Figure out whether π (yes, **the** π) is algebraic. If yes, what is its minimal polynomial (and degree)?

4 Rings and ideals (24 April)

Get Sage started.

Exercise 4.1. (You may want to do this in conjunction with the next Exercise, which has explicit polynomials you can try your hand on.)

Let A be an $m \times n$ matrix with entries in $\mathbb{Q}$. Let $f_1, \dots, f_m \in \mathbb{Q}[x_1, \dots, x_n]$ be the linear polynomials corresponding to the rows of A. Let B be a row echelon form of the matrix A and let $g_1, \dots, g_r$ be the linear polynomials corresponding to the nonzero rows of B.

Prove that $\langle f_1, \dots, f_m \rangle = \langle g_1, \dots, g_r \rangle$.

Exercise 4.2. Apply the approach in the previous Exercise to the polynomials

$$\begin{aligned} f_1 &= x - 2y + z + t \\ f_2 &= x + y + 3z + t \\ f_3 &= 2x - y - z - t \\ f_4 &= 4x - 8y - z + t \end{aligned}$$

Optional: Write a Sage function that implements this "reduction" algorithm for `flst` a list of linear polynomials $f_1, \dots, f_m \in \mathbb{Q}[x_1, \dots, x_n]$:

```
def reduce_gens(flst):
    """Return reduced list of generators glst given by Gauss-Jordan elimination"""
```

Exercise 4.3. Before we get to higher degree polynomials, let's have a look at a (hopefully) more familiar setting: the good-old integers.

Apply long division repeatedly to get the greatest common divisor d of $n_1 = 52500$ and $n_2 = 10725$.

Find integers u_1 and u_2 such that

$$d = u_1 n_1 + u_2 n_2.$$

Exercise 4.4. Let $f_1 = 3x^4 - x^3 + x^2 - x - 2 \in \mathbb{Q}[x]$ and $f_2 = 3x^4 + 2x^3 + 6x + 4 \in \mathbb{Q}[x]$.

Apply polynomial long division repeatedly to get the greatest common divisor d of f_1 and f_2.

Find polynomials u_1 and u_2 in $\mathbb{Q}[x]$ such that

$$d = u_1 f_1 + u_2 f_2.$$

Let $f_3 = x^4 - x^3 + 2x - 2$. How can you find the greatest common divisor g of $\{f_1, f_2, f_3\}$? How can you find polynomials $v_1, v_2, v_3 \in \mathbb{Q}[x]$ such that

$$g = v_1 f_1 + v_2 f_2 + v_3 f_3?$$

Exercise 4.5. Something we observed in the previous exercises is that in both $\mathbb{Z}$ and $\mathbb{Q}[x]$, every ideal is *principal*, that is can be generated by a single element. In fact, in both of these rings we have a good notion of greatest common divisor, and for any elements $a_1, \dots, a_n$ we have

$$\langle a_1, \dots, a_n \rangle = \langle \gcd(a_1, \dots, a_n) \rangle.$$

There are arithmetic systems where this does not hold, for instance

$$\mathbb{Z}[\sqrt{-5}] = \left\{a + b\sqrt{-5} \mid a, b \in \mathbb{Z}\right\}.$$

You can define this ring in Sage by

```
sage: K.<alpha> = QuadraticField(-5)
sage: R = K.ring_of_integers()
```

Here `alpha` is $\sqrt{-5}$.

Once you have the ring `R`, you can define the ideal generated by elements $r_1, \ldots, r_n$ by `I = R.ideal([r1, ..., rn])`.

And once you have an ideal `I` you can ask Sage whether it is principal (i.e. generated by a single element) with `I.is_principal()`.

Try this with the ideal $\langle 2, 1 + \sqrt{-5}\rangle$ in $\mathbb{Z}[\sqrt{-5}]$, as well as the ideals from Exercises 4.3 and 4.4.

Can you find a non-principal ideal in the polynomial ring $\mathbb{Q}[x, y]$? (Interestingly, verifying this in Sage via the method described above does not work.)

Exercise 4.6. Get ready for GitHub, which we will be using for our assignments and take-home exam.

- Get a quick introduction to Git version control with the first three videos at

  https://git-scm.com/videos

  and the guide at

  https://guides.github.com/introduction/git-handbook/

- If you do not have a GitHub account, go to

  https://education.github.com

  and sign up for a student account. (If you already have a normal GitHub account, that should be sufficient and you can skip this step.)

- If you are running Sage and Mathematica on your own laptop or desktop computer, download Git from

  https://git-scm.com/downloads

  If you are using the Melbourne Uni server or Cocalc for Sage, you don't need to download Git, but you're at the mercy of the server gods.

5 Gröbner bases and some applications (1 May)

Get Sage started.

5.1 The mechanics of computing Gröbner bases

Recall that we define our polynomial rings and ideals in Sage using something like

```
R.<x, y, z> = QQ[]
I = R.ideal([x^2-y*z, z^3+x*y*z])
```

But how do we specify the monomial order? We need to use the extended syntax

```
R.<x, y, z> = PolynomialRing(QQ, order="lex")
```

For more details see `PolynomialRing?`.

Exercise 5.1. Given the system

$$x^2 + y^2 - 1 = 0$$
$$x^2 - y = 0,$$

define the appropriate ideal and find the Groebner basis with respect to some of the monomial orders.

What does the Groebner basis tell you about the solutions of the system?

Exercise 5.2. Same questions as above, for the system

$$6x^2y - x + 4y^3 - 1 = 0$$
$$2xy + y^3 = 0.$$

5.2 Graph colourings

For our purposes, a graph will be simple and undirected. (So it consists of a finite set $V = \{j = 0, 1, \dots, n-1\}$ of vertices and a finite subset E of edges of the form $\{j, k\}$.)

Given a graph $\mathcal{G}$, we want to decide whether it is 3-colourable or not: can we assign one of three available colours to each vertex of $\mathcal{G}$ in such a way that any two vertices joined by an edge have distinct colours?

The first step will be to restate this combinatorial question in algebraic terms. Let $\zeta = e^{2\pi i/3} \in \mathbb{C}$ be a third root of unity. We will think of our three colours as being $1, \zeta, \zeta^2$. The vertices of $\mathcal{G}$ will be represented by variables $x_0, x_1, \dots, x_{n-1}$. Colouring vertex j a certain colour corresponds to assigning one of the values $1, \zeta, \zeta^2$ to the variable x_j. We represent the fact that it can only be one of these three values by the equation

$$x_j^3 - 1 = 0.$$

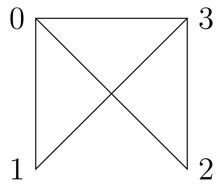
This gives us a system of n cubic equations representing the colouring. But there is an additional constraint: adjacent vertices j and k must not have the same colour. We have $x_j^3 - x_k^3 = 1 - 1 = 0$, and the factorisation $x_j^3 - x_k^3 = (x_j - x_k)(x_j^2 + x_j x_k + x_k^2) = 0$, so the constraint can be expressed as

$$x_j^2 + x_j x_k + x_k^2 = 0,$$

which adds a number of quadratic equations equal to the number of edges in the graph.

The question "is $\mathcal{G}$ 3-colourable?" then becomes "does the associated system of polynomial equations have any solutions?", or letting I be the ideal generated by the left hand sides of the equations, "is $V(I) \neq \varnothing$?" I mentioned before that $V(I) = \varnothing$ if and only if $1 \in I$, so this gives us a way to decide 3-colorability using Gröbner bases.

Exercise 5.3. Take the following graph and figure out by hand whether it's 3-colourable. (No algebra, just try assigning colours.)



Now express the question as an algebra problem in the polynomial ring $\mathbb{Q}[x_0, x_1, x_2, x_3]$ as described above.

Use Gröbner bases to answer the question.

Exercise 5.4. Can you think of a graph that is **not** 3-colourable?

Use Gröbner bases to prove that you are right.

Exercise 5.5 (Optional)**.** Write a function that takes as input a graph $\mathcal{G}$ and a natural number n and uses Gröbner bases to decide whether $\mathcal{G}$ is n-colourable.

A Sage version could have a signature like

```
def is_colourable(G, n):
    """Return True if the graph G is n-colourable, False otherwise."""
```

5.3 Minimal polynomials in field extensions

Suppose α is a real number that is algebraic (over $\mathbb{Q}$) with minimal polynomial $p \in \mathbb{Q}[x]$ (so $p(\alpha) = 0$ and p has smallest possible degree among all polynomials with this property).

Consider now a nonzero real number β of the form

$$\beta = \frac{f(\alpha)}{g(\alpha)},$$

where $f, g \in \mathbb{Q}[x]$.

What is the minimal polynomial of β?

Theorem 5.6. *Consider the ideal $I = \langle p, gy - f \rangle \subset \mathbb{Q}[x, y]$. Take the (reduced) Gröbner basis G of I with respect to the lex order with $x > y$. Then the minimal polynomial of β is the element of G that is a polynomial in y alone.*

Let's test this out!

Exercise 5.7. Let α be a root of $x^5 - x - 2$ and let

$$\beta = \frac{1 - \alpha - 2\alpha^3}{\alpha}$$

Use Gröbner bases as in the above Theorem to find the minimal polynomial of β.

6 Hypergeometric functions and Celine Fasenmyer's algorithm (8 May)

Exercise 6.1. By popular demand, I'll do a quick demo of assignment submission via GitHub.

Exercise 6.2. Given a general term t_k, figure out whether it is a hypergeometric term. (So check whether the ratio t_{k+1}/t_k is a rational function of k.)

Try your approach on

(a) $t_k = k!$

(b) $t_k = \binom{n}{k}$

(c) $t_k = (-1)^k \dfrac{x^{2k+1}}{(2k+1)!}$

Write a function `is_hypergeometric_term(f, k)` that returns True if and only if f is a hypergeometric term with respect to the variable k.

Exercise 6.3. Express the following functions using generalised hypergeometric functions of the form

$$ {}_pF_q\left[\begin{matrix}\alpha_1, & \dots, & \alpha_p \\ \beta_1, & \dots, & \beta_q\end{matrix}; x\right] = \sum_{k=0}^{\infty} \frac{(\alpha_1)_k \dots (\alpha_p)_k}{(\beta_1)_k \dots (\beta_q)_k} \frac{x^k}{k!} $$

Recall that here the ratio of consecutive terms is

$$ \frac{t_{k+1}}{t_k} = \frac{(k+\alpha_1)\dots(k+\alpha_p)}{(k+\beta_1)\dots(k+\beta_q)} \frac{x}{(k+1)} $$

(a) $\sin(x) = \sum_{k=0}^{\infty} (-1)^k \dfrac{x^{2k+1}}{(2k+1)!}$

(b) $\sum_k \dfrac{1}{(2k+1)(2k+3)!}$

Exercise 6.4. Work through the example of Celine Fasenmyer's algorithm in Section 5.2 in the lecture notes.

The idea is to do each step "manually" using the computer for the calculations. As you do this, think about what could be challenging in trying to make the process more automatic.

Exercise 6.5. Automate the process! (To some extent.) Write a function

```
def hypergeometric_term_recurrence(f, n, k, I=1, J=1):
    """
    Implement Celine Fasenmyer's algorithm for computing a k-free
    recurrence relation for the hypergeometric summand f(n, k).

    If a relation does not exist with the given box parameters I and J,
    raise an error.
    """
```

Run your function on the example from Exercise 6.4 as a test.
Now try also:

(a) $f(n,k) = \binom{n}{k}\binom{-n-1}{k}\left(\frac{1-x}{2}\right)^k$

(b) $f(n,k) = \binom{n}{k}\frac{k!\,3^k}{(3k)!}\,x^{n-k}\,y^k$

7 Indefinite hypergeometric summation (15 May)

The aim is to implement Gosper's algorithm for hypergeometric summation.

7.1 Finding suitable polynomials p, q, r (Lemma 5.5)

Exercise 7.1. Write a function `irreducible_dispersion(s, t, k)` that takes two irreducible polynomials s and t in the variable k and determines if they are the same up to a shift $k + j$, using the following algorithm:

- if s and t have different degrees, return the empty list
- let $n = \deg(s) = \deg(t)$; if n is zero, return the empty list
- let a be the coefficient of k^n in s, b the coefficient of k^{n-1} in s, c the coefficient of k^n in t, and d the coefficient of k^{n-1} in t
- let $j = (bc - ad)/(acn)$; if $j \notin \mathbb{Z}_{\geqslant 0}$, return the empty list
- if $cs(k) - at(k + j) = 0$, return the list containing the single element j

Try your function on

- $s = k$, $t = k - 97$, should get `[97]`
- $s = k^2 + 5$, $t = k^2 + 2$, should get `[]`

Exercise 7.2. Write a function `dispersion_set(q, r, k)` that takes two polynomials q and r in k and returns the dispersion set

$$J = \{j \in \mathbb{Z}_{\geqslant 0} \mid \deg \gcd(q(k), r(k + j)) > 0\}$$

Use the following algorithm:

- set $J = \{\}$
- loop through all polynomial factors s of q and all polynomial factors t of r and let D be the result of the function `irreducible_dispersion(s, t, k)`; set $J = J \cup D$

Try your function on

- $q = (k + 5)^2(2k + 7)$, $r = k^2(2k + 1)$, should get `[3, 5]`
- $q = (k + 2)(k + 3)$, $r = k^2 + 1$, should get `[]`

Exercise 7.3. Write a function `find_polys(n, d, k)` that starts with a rational function n/d in k and produces polynomials p, q, r satisfying the conditions of Lemma 5.5, using the dispersion set function from the previous exercise and the rewriting procedure following the statement of Lemma 5.5.

Try your function on

- $n = (k+1)^3$, $d = k^3$, should get $p = k^3$, $q = r = 1$;
- $n = k$, $d = k+1$, should get $p = 1$, $q = k-1$, $r = k$;
- $n = x-k$, $d = k+1$, should get $p = 1$, $q = x-k+1$, $r = k$.

7.2 Bound on the degree of the auxiliary polynomial (Lemma 5.6)

Exercise 7.4. Implement the conditions in the statement of Lemma 5.6 in the notes to obtain a function `degree_bound(p, q, r, k)` that computes an upper bound N on the auxiliary polynomial f.

Try it on:

- $p = k^3$, $q = r = 1$, should get $N = 4$;
- $p = 1$, $q = x-k+1$, $r = k$, should get $N = -1$;
- $p = 1$, $q = k-x-1$, $r = k$, should get $N = 0$.

7.3 Putting it all together: Gosper's algorithm

Exercise 7.5. For the last preparatory step, there is an operation that we can isolate from the context of Gosper's algorithm (and that also appeared in Fasenmyer's algorithm last week): given a symbolic expression f, a distinguished variable k and a set of variables $\{c_0, c_1, \ldots, c_N\}$ such that f is polynomial in k and has degree at most 1 in each of the c_j, solve for $c_0, c_1, \ldots, c_N$ in the equation

$$f = 0$$

Implement this as a function `solve_for_coefficients(f, k, coeff)`.

Exercise 7.6. Write a function `gosper_sum(a, k)` that implements Gosper's algorithm as outlined on page 30 of the lecture notes, building on the other functions from these lab sheets.

Try it on:

- $a = k^3$, should get antidifference

$$A = \frac{k^2(k-1)^2}{4}$$

- $a = \frac{1}{k}$, should get that a is not Gosper-summable
- $a = \frac{1}{k} - \frac{1}{k+1}$, should get

$$A = -\frac{1}{k}$$

- $a = \frac{1}{k(k+6)}$, should get

$$A = -\frac{(3k^4 + 30k^3 + 95k^2 + 100k + 24)(2k+5)}{6(k+5)(k+4)(k+3)(k+2)(k+1)k}$$

8 Wilf–Zeilberger method (22 May)

You may want to pull the latest version of the lab code from

https://github.com/aghitza/mast90053

8.1 Verification of Gosper's algorithm using the rational certificate

Exercise 8.1. Write a function `gosper_verify(a, R, k)` that takes a hypergeometric term a_k and a rational function R and checks that R is indeed the rational certificate for Gosper's algorithm applied to a_k.

Test your function on:

- $a_k = k$, $R = (k-1)/2$
- $a_k = k^2$, $R = k+1$
- $a_k = \frac{1}{k} - \frac{1}{k+1}$
- $a_k = (-1)^k \binom{n}{k}$
- $a_k = k\,(k!)$

8.2 Wilf–Zeilberger

Exercise 8.2. Work through the details of Example 5.10 in the lecture notes, i.e. apply the Wilf–Zeilberger method to the proof of the identity

$$\sum_{k=0}^{n} \frac{1}{2^n}\binom{n}{k} = 1$$

Exercise 8.3. Write a function `wz_verify(a, R, n, k)` that takes a hypergeometric term $a(n,k)$ and a rational function $R(n,k)$ and checks that R is indeed the Wilf–Zeilberger certificate for $a(n,k)$.

Test your function on:

$$a(n,k) = \frac{1}{2^n}\binom{n}{k}$$

with the WZ certificate you can deduce from the previous exercise.

Exercise 8.4. Let's automate the Wilf–Zeilberger method.

Write a function `wz_certificate(a, n, k)` that takes a hypergeometric term and applies the Wilf–Zeilberger method to it. If unsuccessful, it should raise an error. If successful, it should return the WZ certificate.

Test your function on:

- $a(n,k) = \dfrac{1}{2^n}\dbinom{n}{k}$
- $a(n,k) = \dbinom{n}{k}$; interpret the outcome
- $a(n,k) = \dbinom{n}{k}\dfrac{x^k}{(1+x)^n}$; interpret the outcome

9 Zeilberger's algorithm (29 May)

You may want to pull the latest version of the lab code from

https://github.com/aghitza/mast90053

Note in particular that I gathered all the code relating to hypergeometric terms and summation and put it into `hypergeometric/hypergeometric.sage`

In fact, it's high time we learned a better way to deal with a larger codebase. Instead of copying-and-pasting the code from `hypergeometric.sage`, we tell Sage where to find it. How? It depends a bit on how you use Sage:

- if you use Jupyter notebooks, use

  ```
  load('path-to-the-file-on-your-computer/hypergeometric.sage')
  ```

  For bonus `1337 h4x0r` street cred, load the file directly from where it sits on GitHub.[1]

- if you use command-line interface (from a terminal window), then

  ```
  sage: %attach 'path-to-the-file-on-your-computer/hypergeometric.sage'
  ```

  which has the added feature that it reloads if the file changes.

  (The `load` command from the previous bullet point can also be used in the terminal.)

9.1 Applying Zeilberger's method

Exercise 9.1. Work through Example 5.12 in the lecture notes, using the computer to work out the details, e.g. use `find_polys` to get p, q, r, `degree_bound` to get N, then `solve_for_coefficients` to get b_0 and σ_1.

Exercise 9.2. Write a function `zeilberger(f, n, k, J)` that implements Zeilberger's algorithm.

Test your function on:

- $f(n,k) = \binom{n}{k}$
- $f(n,k) = \binom{n}{k}^2$. What identity do you get from this?
- $f(n,k) = (-1)^{n+k}\binom{n}{k}2^k$. What identity do you get from this? Can you use a different algorithm for the same problem?

[1] Yes, it's convenient and impresses people at parties. But I should advise you against loading random code off places on the internet. You don't know where that's been. And malicious code can do lots of bad stuff that you don't even want to think about.

9.2 Timing is everything

Exercise 9.3. Apply Fasenmyer's algorithm, implemented in `fasenmyer_kfree`, to

$$f_1(n,k) = \binom{n}{k} x^{n-k} y^k$$

Now try

$$f_2(n,k) = \binom{n}{k} \frac{k! 2^k}{(2k)!} x^{n-k} y^k$$

A bit slow, isn't it?

Replacing 2 by 3 will be even slower (several minutes).

Look at the documentation for `%time`, `%timeit`, and `%prun`, and try these commands on the Fasenmyer algorithm for f_2.

Exercise 9.4. It seems that linear algebra over the symbolic ring `SR` is slow. Maybe we can do it over an appropriate ring of polynomials?

Modify the code for `solve_for_coefficients_homog` to use a polynomial ring instead of `SR`.

Run `_test_all()` to make sure you haven't broken anything.

Then run the timings on the examples from the previous exercise to see if this approach is indeed faster.

We can apply the same optimisation to the code for `solve_for_coefficients` and thereby speed up Gosper, Wilf–Zeilberger, and Zeilberger.