Proseminar on computer-assisted mathematics

```lean
import data.real.basic

def sequence_tendsto (u : ℕ → ℝ) (l : ℝ) :=
∀ ε > 0, ∃ N : ℕ, ∀ n > N, |u n - l| < ε

def continuous_function_at (f : ℝ → ℝ) (x0 : ℝ) :=
∀ ε > 0, ∃ δ > 0, ∀ x, |x - x0| < δ → |f(x) - f(x0)| < ε

example (f : ℝ → ℝ) (u : ℕ → ℝ) (x0 : ℝ )
(hu : sequence_tendsto u x0)
(hf : continuous_function_at f x0) :
sequence_tendsto (f ∘ u) (f x0) :=

begin
show (∀ ε > 0, ∃ N : ℕ, ∀ n > N, |f(u n) - f(x0)| < ε),
assume (ε ε_pos),
have h1 : (∃ δ>0, ∀ x : ℝ, |x - x0| < δ → |f x - f x0| < ε) :=
  by {apply hf, exact ε_pos},
rcases h1 with ⟨δ, δ_pos, h2⟩,
have h3 : (∃ N : ℕ, ∀ n> N, |u n - x0| < δ) :=
  by {apply hu, exact δ_pos},
rcases h3 with ⟨N, h4⟩,
use N,
intros n n_large,
apply h2,
apply h4,
exact n_large,
end
```

▼ 02bis_limit_of_f(u n).lean:27:14

▼ Tactic state

goals accomplished 🎉

▶ All Messages (0)

Matrices in *Sage*

When we define a matrix in *Sage*, we can specify the ring or field in which we take the entries.

Let us for instance consider the matrix

$$\begin{pmatrix} 2 & 4 & 6 \\ 4 & 5 & 6 \\ 3 & 1 & 2 \end{pmatrix}$$

and declare it first as a matrix A with entries in $\mathbb{Q}$, then as a matrix B with entries the field with seven elements $\mathbb{F}_7$.

```python
A = matrix( QQ, [[2,4,6],[4,5,6],[3,1,2]] )
show(A)
```


Florent Schaffhauser

Heidelberg University, Summer semester 2023

Two main objectives:

- Linear algebra, using Sagemath

Matrices in *Sage*

When we define a matrix in *Sage*, we can specify the ring or field in which we take the entries.

Let us for instance consider the matrix

$$\begin{pmatrix} 2 & 4 & 6 \\ 4 & 5 & 6 \\ 3 & 1 & 2 \end{pmatrix}$$

and declare it first as a matrix A with entries in $\mathbb{Q}$, then as a matrix B with entries the field with seven elements $\mathbb{F}_7$.

```
A = matrix( QQ, [[2,4,6],[4,5,6],[3,1,2]] )
show(A)
```

```
show(A.inverse())
```

$$\begin{pmatrix} -\frac{2}{9} & \frac{1}{9} & \frac{1}{3} \\ -\frac{5}{9} & \frac{7}{9} & -\frac{2}{3} \\ \frac{11}{18} & -\frac{5}{9} & \frac{1}{3} \end{pmatrix}$$

- Proof assistants, using Lean

```
def sequence_tendsto (u : ℕ → ℝ) (l : ℝ) :=
∀ ε > 0, ∃ N : ℕ, ∀ n ≥ N, |u n - l| < ε

def continuous_function_at (f : ℝ → ℝ) (x0 : ℝ) :=
∀ ε > 0, ∃ δ > 0, ∀ x, |x - x0| < δ → |f(x) - f(x0)| < ε

example (f : ℝ → ℝ) (u : ℕ → ℝ) (x0 : ℝ )
(hu : sequence_tendsto u x0)
(hf : continuous_function_at f x0) :
sequence_tendsto (f ∘ u) (f x0) :=
```

Programme

Topics of linear algebra to be covered in the seminar:

- Computer algebra systems. Representations of vectors and matrices.
- Row operations. Gaussian elimination. Row-reduced echelon form of a matrix.
- Invertible matrices. Elementary matrices. Determinant.
- Linear independence. Bases for the kernel and the image of a linear transformation.
- Rank-nullity theorem and the row space of matrix. Basis for the row space.
- Base change. Coordinates of a vector, matrix of a linear transformation.
- Eigenvalues and the characteristic polynomial. Diagonalisation.
- The Gram-Schmidt process. Least-square approximation.

Aspects of the *Lean* programming language to be covered in the seminar:

- Installation of a proof assistant. Familiarisation with the interface.
- The *Natural Number Game* (Peano axioms and the induction principle).
- Equality and computations (tactics to prove algebraic identities).
- Implications and equivalences (propositional logic).
- Predicates and quantifiers (first-order logic).
- Contraposition and proof by contradiction (proof tactics).
- Formalisation of basic mathematical statements.

1. The seminar is collaborative and project-based.

2. To pass, attendance is mandatory (unless excused in advance).

Schedule

#	Date	Topic	Speaker	Slides	Code
1	19/04	Introductory meeting	Florent Schaffhauser		
2	26/04	Matrices in Sage (introduction)	Florent Schaffhauser		
3	03/05	Working in pairs	N/A		
4	17/05	Kernels, images, bases and diagonalisation	Florent Schaffhauser		
5	24/05	Project prepararation	N/A		
6	31/05	Project presentation	In pairs		
7	07/06	Introduction to Lean	Florent Schaffhauser		
8	14/06	Natural Number Game	N/A		
9	28/06	Working in pairs	N/A		
10	05/07	Formalising basic mathematical objects	Florent Schaffhauser		
11	19/07	Project preparation	N/A		
12	26/07	Project presentation	In pairs		

→ Team up soon!

→ Choose a project (later).

Practical organization

Minimal requirements

- At least one computer / tablet per team.

Before 26.04

1. Create individual GitHub accounts.
   https://github.com
2. Join the seminar's Zulip channel.
   https://matematiflo.zulipchat.com

Advanced option (facultative)

- Install Sagemath (requires Python).
  https://doc.sagemath.org/html/en/installation/index.html

- Install JupyterLab. https://jupyter.org/install