

# Introduction

---

IFT503/711 – Théorie du calcul

Hiver 2026



UNIVERSITÉ DE  
SHERBROOKE

- Plan de cours
- Introduction à la théorie du calcul

# **Plan de cours**

## Michael Blondin

@ michael.blondin@usherbrooke.ca

🌐 michaelblondin.com

📍 bureau D4-1024-1 au 1<sup>er</sup> étage

## Manuel Lafond

@ manuel.lafond@usherbrooke.ca

🌐 mlafond.recherche.usherbrooke.ca

📍 bureau D4-2010 au 2<sup>ème</sup> étage

## Dave Touchette

@ dave.touchette@usherbrooke.ca

📍 bureau D4-1018-2 au 1<sup>er</sup> étage

## Groupe (en date du 5 jan.)

	B.Sc.	M.Sc.	Ph.D.	Total
Informatique	5	1	2	8
Info. quantique	13	0	0	13
<b>Total</b>	18	1	2	21

---

<b>Cours</b>	Lundi	13h30 – 15h20
	Vendredi	08h30 – 10h20

---

<b>Exercices</b>	Selon la semaine (souvent quatrième heure)	
------------------	---	--

---

## **Préalables (au 1<sup>er</sup> cycle):**

- IFT313 – Introduction aux langages formels  $\vee$
- MAT120 – Mathématiques discrètes  $\wedge$
- IFT436 – Algorithmes et structures de données

## **Atouts:**

- IFT436 – Algorithmes et structures de données
- IFT339 – Structures de données

## **Utile pour:**

- tout cours avec des problèmes algorithmiques

- Qu'est-ce qu'un ordinateur et un algorithme?
- L'ordinateur peut-il répondre à toutes nos questions?
- Peut-on toujours rendre un algo. efficace?
- Peut-on surmonter les problèmes difficiles?
- Plusieurs processeurs = toujours plus rapide?
- Qu'est-ce que l'ordinateur quantique?



## Grandes questions du cours

- Qu'est-ce qu'un ordinateur et un algorithme?
- L'ordinateur peut-il répondre à toutes nos questions?
- Peut-on toujours rendre un algo. efficace?
- Peut-on surmonter les problèmes difficiles?
- Plusieurs processeurs = toujours plus rapide?
- Qu'est-ce que l'ordinateur quantique?

## Grandes questions du cours

- Qu'est-ce qu'un ordinateur et un algorithme?
- L'ordinateur peut-il répondre à toutes nos questions?
- Peut-on toujours rendre un algo. efficace?
- Peut-on surmonter les problèmes difficiles?
- Plusieurs processeurs = toujours plus rapide?
- Qu'est-ce que l'ordinateur quantique?

## Grandes questions du cours

- Qu'est-ce qu'un ordinateur et un algorithme?
- L'ordinateur peut-il répondre à toutes nos questions?
- Peut-on toujours rendre un algo. efficace?
- Peut-on surmonter les problèmes difficiles?
- Plusieurs processeurs = toujours plus rapide?
- Qu'est-ce que l'ordinateur quantique?

## Grandes questions du cours

- Qu'est-ce qu'un ordinateur et un algorithme?
- L'ordinateur peut-il répondre à toutes nos questions?
- Peut-on toujours rendre un algo. efficace?
- Peut-on surmonter les problèmes difficiles?
- Plusieurs processeurs = toujours plus rapide?
- Qu'est-ce que l'ordinateur quantique?

## Grandes questions du cours

- Qu'est-ce qu'un ordinateur et un algorithme?
- L'ordinateur peut-il répondre à toutes nos questions?
- Peut-on toujours rendre un algo. efficace?
- Peut-on surmonter les problèmes difficiles?
- Plusieurs processeurs = toujours plus rapide?
- Qu'est-ce que l'ordinateur quantique?

## Fondements

1. Calculabilité
2. Décidabilité
3. P vs. NP

## Sujets avancés

4. Logique
5. Calcul parallèle (sous P)
6. Complexité (au-delà de NP)
7. Informatique quantique

## 1. Calculabilité

**8h** (Blondin)

- Ordinateur et algorithmes: formalisation
- Machine de Turing
- Hypothèse de Church-Turing
- Temps et mémoire

## 2. Décidabilité

**8h** (Touchette)

- Problèmes décidables et non décidables
- Réductions entre problèmes
- Problème de l'arrêt
- Théorème de récursion



## 3. P vs. NP

8h (Lafond)

- Temps polynomial déterministe et non déterministe
- Réductions polynomiales
- NP-complétude

## 4. Logique

**4h** (Blondin)

- Logique du premier ordre
- Arithmétique: décidabilité et indécidabilité

## 5. Calcul parallèle

**4h** (Blondin)

- Classes à l'intérieur de P
- Circuits booléens
- P-complétude

## 6. Informatique quantique

**8h** (Touchette)

- Circuits quantiques
- Algorithmes quantiques
- Classes de complexité (comme BQP et QMA)

## 7. Complexité du calcul (au-delà de NP)

8h (Lafond)

- Complexité en espace
- Complexité exponentielle

## **Cours théorique** (pas de programmation)

- Enseignement au tableau
- Certaines notes de cours en ligne
- Références:
  - M. SIPSER: *Introduction to the Theory of Computation*, 2<sup>ème</sup> édition
  - S. ARORA, B. BARAK: *Computational Complexity: A Modern Approach*
  - I. CHUANG, M. NIELSEN: *Quantum Computation and  
Quantum Information*

- 6 devoirs
- 1<sup>er</sup> cycle: à deux ou individuel
- 2/3<sup>ème</sup> cycles: individuel + questions avancées



---

Devoirs	40%
Examen périodique	30%
Examen final	30%


---

Sujets	Devoirs
1: Calculabilité	Devoir 1 2 semaines
2: Calculabilité	
3: Décidabilité	Devoir 2 2 semaines
4: Décidabilité	
5: P vs. NP	Devoir 3 2 semaines
6: P vs. NP	
7: Logique	Devoir 4 4 semaines
8: Examen périodique	
9: Relâche	

Sujets	Devoirs
10: Calcul parallèle	Devoir 4
11: Info. quantique	Devoir 5 2 semaines
12: Info. quantique	
13: Complexité	Devoir 6 ~ 2½ semaines
14: Complexité	
15: Complexité	
16: Examens	—
17: Examens	—

Si porte entrouverte ou  
sur rendez-vous

(en priorité avec le prof.  
qui a enseigné la matière)

 [grif-usherbrooke.github.io/calcul](https://grif-usherbrooke.github.io/calcul)

# Calculabilité