Université de Fribourg

– Laboratoire de Didactique –

Diplôme d'Enseignement aux Écoles de Maturité

Outils d'Adaptativité

Utiliser avec souplesse l'outillage mathématique

Auteurs

Mathias Blaise
Alexandros Rispo Constantinou

9 mai 2024

UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

Table des matières

Table des figures

Table des codes

Abstrait

DANS LE CADRE du Laboratoire didactique de leur formation à l'enseignement en secondaire II, les soussignés ont créé le projet *Outils d'Adaptativité* visant à répondre aux exigences suivantes : (i) développer une ou plus des compétences basales en mathématiques définies dans le Plan d'études cadre (PEC) national (CDIP 2016) ; (ii) dont notamment celle consistant à « utiliser avec souplesse l'outillage mathématique » ; (iii) à l'aide d'une ressource numérique générant automatiquement des exercices autocorrigés, pouvant être proposés durant les cours de soutien ; (iv) ainsi qu'une réflexion sur l'évaluation des compétences basales par les enseignant·e·s dans leurs cours, avec une proposition concrète à utiliser en classe.

Le résultat final de l'activité consiste en deux documents à compiler avec le moteur LuaLaTeX, rendus disponibles à travers un dépôt GitHub.[1] Ceux-ci comportent chacun une liste d'exercices — de résolution d'équations du second degré pour l'un, de résolution de systèmes d'équations linéaires à deux inconnues pour l'autre — accompagnée de leurs solutions. Pour pouvoir répondre aux exercices de façon efficace, les élèves devront déterminer la méthode la plus appropriée à la résolution de chaque équation, *utilisant* ainsi *avec souplesse l'outillage mathématique*. Aux solutions est jointe pour chaque équation une méthode jugée par nous optimale, décidée algorithmiquement. Dans la confrontation à — et l'éventuel désaccord avec — ces méthodes, les élèves pourront développer une réflexion critique sur la pertinence de leurs divers outils mathématiques face à des situations données.

Enfin, le rapport ci-présent a été écrit pour répondre au point (iv) susdit. Nous y résumons le travail fait jusqu'à présent dans le canton de Fribourg autour des compétences basales en mathématiques, puis présentons la ressource par nous créée, avec son manuel d'utilisation et une proposition concrète pour son intégration en classe. Nous analysons également les résultats d'un sondage mené auprès des enseignant·e·s autour de ce projet.

[1] Alexandros RISPO CONSTANTINOU et Mathias BLAISE (2024). *Outils d'Adaptativité*. URL : `https://github.com/tytyvillus/laboratoire-didactique`.

— 1 —

Introduction

DEPUIS DES ANNÉES, différents groupes de travail se succèdent dans le canton de Fribourg pour discuter de la notion de compétences basales en mathématiques (GCM 2017, 2018, 2020, 2021). En effet, une maturité gymnasiale se doit d'ouvrir les portes à toute filière scientifique. C'est à cette fin que les compétences basales sont définies et redéfinies : elles consistent en ce qui doit être au minimum acquis par un·e étudiant·e voulant se lancer à l'Université, peu importe son choix disciplinaire. Ainsi, la problématique est de taille : comment définir au mieux les compétences nécessaires et prendre en compte tous les domaines d'études supérieures ? Et plus important encore, comment faire pour que ces compétences basales soient acquises par les élèves ? Nous débuterons notre parcours en décrivant brièvement la cristallisation de cette notion, et survolerons ensuite les tentatives successives de concrétiser les compétences admises comme basales, ainsi que les réflexions quant à l'implémentation d'un système favorisant l'acquisition par les élèves. Pour finir, nous discuterons de notre apport dans ces réflexions d'implémentation. Celui-ci réside dans la préparation d'un document LaTeX permettant d'auto-générer des exercices et des corrigés pour deux thèmes précis : les équations quadratiques et les systèmes d'équations linéaires.

1.1 Les compétences basales

Les compétences de base sont une notion introduite par les Plans d'études cadre (PEC). On entend généralement par « compétence basale » en mathématiques toute compétence nécessaire dans des chapitres donnés pour permettre la poursuite d'études scientifiques après le gymnase.[2] Elles ne suffisent toutefois pas, à elles seules, à garantir l'aptitude générale aux études supérieures.

1.1.1 Définition générale d'une compétence basale

La Conférence des directrices et directeurs cantonaux de l'instruction publique (CDIP) est venue formaliser cette notion du Plan d'études cadre de 1994 (CDIP 1994),

[2] Notamment, une compétence basale n'est *pas* une compétence « minimale » ou « de base » en mathématiques, suffisante à un·e adulte moyen·ne dans la vie quotidienne.

jusqu'alors floue, dans son annexe du 17 mars 2016 (CDIP 2016). On en tire que la compétence de base en mathématiques, constitutive de l'aptitude générale aux études supérieures, est « une maîtrise tout en souplesse, adaptative des thèmes mathématiques de base du Plan d'études ».[3,4]

Dans les faits, cette notion de la CDIP est-elle reprise par les acteurs cantonaux ? Nous apercevons dès le troisième rapport (GCM 2018, p. 6) une définition « usuelle » des compétences basales en mathématiques, reportée dans l'encadré ci-dessous.

Définition : Une compétence basale est une compétence technique élémentaire régulièrement présente dans le cursus de mathématiques et dont la maîtrise est importante à la résolution d'exercices simples ou à la bonne compréhension de différentes branches dispensées au gymnase.

. . . le tout dans l'optique de poursuivre des études supérieures. Cette définition consolide et précise celle de la CDIP, trop vaste pour pragmatiquement être appliquée. Néanmoins, même dans cette définition bien opérationnelle, il y a deux pièges dans lesquels il faut veiller à ne pas tomber :

(i) Une compétence nécessaire à atteindre une certain seuil de suffisance peut parfaitement ne pas être considérée comme basale.

(ii) Une compétence basale n'est pas nécessairement facile à acquérir.

Munis à présent d'une notion de compétence basale soigneusement établie, voyons ensuite comment celle-ci a été concrétisée dans le canton de Fribourg en une liste de quarante-trois compétences précises.

1.1.2 Un catalogue des compétences basales : les évolutions

La définition précise des compétences à retenir comme « basales » a fait l'objet de différents mandats et groupes de travaux au fil des années, et ce depuis 2005 déjà. En effet, de 2005 à 2008, une évaluation de la réforme de la maturité (EVAMAR II ; Eberle, Gehrer et al. 2008) a été conduite. Il en est ressorti (p. 16) que les connaissances et compétences en mathématiques des gymnasien·ne·s testé·e·s étaient insuffisantes. C'est pourquoi quand, en 2012, la CDIP lançait un projet visant à garantir un accès sans examen aux Hautes Écoles pour toute personne en possession d'une maturité

[3]Les compétences basales telles qu'elles sont en vigueur dans le canton de Fribourg depuis l'année scolaire 2022 ont été créées à partir de cette définition actualisée de l'annexe au PEC. Elles sont décrites dans le *Plan des études gymnasiales* fribourgeois (Domaines des Mathématiques) (S2 2020).

[4]Les compétences basales en mathématiques sont d'ailleurs un cas spécial de la « compétence disciplinaire de base constitutive de l'aptitude générale aux études supérieures », définie dans l'annexe au PEC comme étant « la somme des savoirs et savoir-faire dans les disciplines concernées dont l'acquisition est prérequise pour étudier un grand nombre de branches universitaires, et non pas certaines d'entre elles seulement ».

gymnasiale (CDIP 2013, p. 3), l'un des sous-projets concernait les compétences disciplinaires de base requises pour les études universitaires, dans le but de les uniformiser à l'échelle nationale.

Quelques années plus tard, en 2015 et en 2016, avec l'introduction du bilinguisme dès la première année de gymnase, l'accès des élèves à l'enseignement des mathématiques dans la langue partenaire devait être facilité. Un groupe de travail, formé de deux enseignant·e·s par gymnase du canton, est alors mandaté (CDIP 2015) par le Service de l'enseignement secondaire du deuxième degré (S2) et la Conférence des recteurs (CORECOFR). En bref, le mandat du groupe de travail était de définir des buts communs à la première année de gymnase des deux communautés linguistiques, à l'aide d'un catalogue de compétences disciplinaires de base en lien avec l'aptitude générale aux études supérieures, ainsi que de développer des idées sur les manières de promouvoir la branche des mathématiques. À cette fin, un questionnaire d'exercices types (GCM 2016, reproduit en annexe B, p. 77) a été distribué aux enseignant·e·s de mathématiques des gymnases du canton, à partir duquel fut écrite une liste articulant

(i) les thèmes ayant été retenus par les deux sections linguistiques,

(ii) les thèmes ayant été rejetés par les deux sections linguistiques,

(iii) ainsi que les quatorze thèmes retenus par l'une des sections et rejetés par l'autre.

Cette liste est reproduite en fig. 1 telle qu'elle apparaît dans GCM (2017).

En parallèle de ce premier mandat, la CDIP a recommandé aux cantons d'émettre des directives pour l'acquisition de compétences de base en mathématiques, ce qui a conduit à l'annexe au PEC discutée auparavant (CDIP 2016).

Depuis ce premier mandat, trois autres du même type ont été conduits en 2017, 2018 puis 2020, avec dans chacun des cas un objectif spécifique. Le mandat de 2017 (S2 2017) se trouve être le troisième mandat portant sur les compétences basales, le premier mandat ayant disposé d'un prolongement (deuxième mandat) pour finaliser les listes en fig. 1. Ce troisième mandat avait pour but de clarifier la manière d'identifier les compétences basales et de proposer des mesures concrètes pour les améliorer. Le rapport résultant (GCM 2018) énonce quarante-trois compétences convenues basales, piochées dans les différents thèmes de la première année gymnasiale.[5]

En 2018, le quatrième mandat consiste en la mise en œuvre des mandats précédents. Cette fois-ci, le groupe de travail s'est réparti l'effort par établissement, et le rapport (GCM 2020) présente les points de vue de chaque gymnase du canton. Il propose également une adaptation de la liste des quarante-trois compétences retenues basales, mais par des modifications quasi négligeables dans la mise en pratique : le plus grand changement est l'ajout conseillé d'une nouvelle compétence n.º 7 par le Collège du Sud, amenant le total à quarante-quatre compétences basales et entraînant la renumérotation

[5]Un exemple particulier est présenté en fig. 2 ; on y remarque en effet qu'il est précisé que la factorisation par trinôme n'est pas considérée comme compétence basale.

Thèmes retenus par les deux sections linguistiques

Algèbre
- Bases de la théorie des ensembles (appartenance, intersection, réunion, différence, diagramme de Venn)
- Ensembles $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$
- Intervalles
- Opérations sur les nombres rationnels sans calculatrice, y compris puissances avec exposant rationnel
- Notation scientifique (Wissenschaftliche Schreibweise)
- Gestion des parenthèses
- Développement et réduction de polynômes
- Calcul algébrique avec puissances et racines (y compris travail avec exposants rationnels)
- Identités remarquables de degré 2 (Binomische Formeln)
- Factorisations simples (mise en évidence et polynômes du deuxième degré)
- Résolution d'équations de degré 1, quel que soit l'ensemble des solutions
- Résolution d'équations quadratiques à l'aide de la méthode du discriminant
- Résolution d'équations de degré supérieur à 2 réductibles à la forme $x^n = a$
- Résolution d'équations rationnelles réductibles au premier degré, y compris avec mise à l'écart d'une solution étrangère (Bruchgleichungen mit Lösungen, welche nicht im Definitionsbereich liegen)
- Isoler une lettre dans une formule, sans qu'une mise en évidence ne soit nécessaire
- Résolution de systèmes linéaires 2×2

Géométrie
- Notion de triangle rectangle et d'hypoténuse
- Théorème de Pythagore
- Calculs de longueurs et d'angles à l'aide de la trigonométrie du triangle rectangle
- Notion de vecteur
- Notion de colinéarité
- Représentations graphiques de différents multiples d'un vecteur donné par un de ses représentants
- Représentations graphiques des additions et des soustractions de vecteurs donnés par des représentants
- Opérations sur les composantes des vecteurs
- In geometrischen Figuren, deren Eckpunkte bekannt sind, Terme mit Vektoren vereinfachen und Berechnungen durchführen

Fonctions
- Erkennen ob eine als Pfeildiagramm oder Graph gegebene Relation eine Funktion ist
- Zu linearen und quadratischen Funktionstermen den dazugehörigen Graphen erkennen
- Erstellen einer Wertetabelle und zeichnen des Graphen
- Erstellen einer Wertetabelle mit Hilfe eines Taschenrechners
- Lösen von linearen Ungleichungen
- Premier degré : expression fonctionnelle de la droite, représentation, pente, intersections, domaine
- Premier degré : résolution d'équations graphiquement en cas de solution unique
- Deuxième degré : expression fonctionnelle de la courbe, représentation, sommet, intersections, domaine
- Premier et deuxième degré : connaissances générales au sujet des représentations graphiques

Thèmes rejetés par les deux sections linguistiques

Algèbre
- Rendre rationnel le dénominateur d'une fraction littérale en amplifiant par le conjugué
- Résolution d'équations cubiques factorisables par division polynomiale
- Résolution d'équations avec racine dont l'élévation au carré nécessite l'utilisation du carré d'une somme ou d'une différence
- Discussion du nombre de solutions d'une équation quadratique avec paramètres

Géométrie
- Constructions géométrique faisant intervenir le cercle de Thalès
- Constructions de triangles rectangles étant donné un rapport trigonométrique

Fonctions
- Zeichnerisches Verschieben und Strecken eines Funktionsgraphen in Richtung der x-Achse
- Verkettung von Funktionen und Bestimmung des Definitionsbereichs der verketteten Funktionen
- Bestimmung mit Hilfe des Taschenrechners in welchen Intervallen eine Funktion fallend ist
- Ungleichungen mit Bruchtermen, bei denen die Unbekannte im Zähler und Nenner vorkommt
- Ungleichungen mit Bruchtermen mit Hilfe des Taschenrechners lösen
- Vorzeichen von gebrochen rationalen Funktionen und Wurzelfunktionen
- Bestimmung von Gebieten in der Ebene, die durch Funktionsgraphen begrenzt sind
- Résolution algébrique d'une équation avec valeur absolue
- Donner une définition par morceaux d'une ligne brisée, éventuellement en utilisant une valeur absolue
- Établir l'équation d'une parabole de sommet connu et de droite tangente connue
- Résolution d'inéquation entre valeur absolue d'une fonction du premier degré d'une part et fonction du deuxième degré d'autre part, le graphique ne suffisant pas à lire les solutions
- Exercices relatifs à une fonction du deuxième degré dont les coefficients sont liés à un paramètre

Thèmes retenus par la section mentionnée et rejetés par l'autre

Section alémanique
- Berechnung von $(a + b)^n$ mit Hilfe des Pascal-Dreieckes (mais résultats inhomogènes entre les sections)
- Résolution d'équations avec racines simples sans solution étrangère
- Géométrie vectorielle de l'espace

Section romande
- Extraction des carrés parfaits d'une racine
- Rendre rationnel le dénominateur d'une fraction numérique comportant une racine
- Identités remarquables de degré 3
- Identification des composantes des vecteurs du plan dans des bases quelconques
- Utilisation du terme de norme et de la notation $||\vec{v}||$ pour désigner la longueur d'un représentant d'un vecteur $\vec{v}$
- Lösen von quadratischen Ungleichungen
- Ungleichungen dritten Grades, bei denen eine Lösung Null ist
- Bestimmung von Punkten im Graphen einer Funktion, die mehrere Bedingungen gleichzeitig erfüllen
- Bestimmen der Definitionsmenge und Nullstellen von gebrochen rationalen Funktionen
- Résolution graphique d'une équation dont les solutions sont multiples (réunion d'un intervalle et de singletons)
- Présentation de l'étude du signe d'une fonction sous la forme d'un tableau de signes

FIG. 1 : Liste des différents thèmes retenus et rejetés suite au deuxième mandat (GCM 2017).

$n \mapsto n + 1$ des anciennes compétences $n \in \{7, \dots, 42\}$. C'est dès ce mandat que l'importance du choix de la méthode de résolution dans un exercice commence à être citée en dehors de l'annexe au PEC (CDIP 2016). Le rapport fit remarquer qu'aucun effet positif de l'entraînement des compétences basales n'avait pu être observé, et plusieurs raisons furent amenées : la difficulté d'entraîner les compétences basales sans prétériter l'atteinte des objectifs du plan cantonal ; la courte durée de l'expérience et son interruption par le confinement COVID-19 ; et l'insuffisance de la simple responsabilisation des élèves à l'acquisition des compétences basales sans contrôle de la part de l'enseignant·e.

Le mandat le plus récent, dont le rapport date de 2021 (GCM 2021), s'est concentré sur deux aspects : l'établissement d'une liste de compétences basales pour la deuxième année gymnasiale, dans la même idée que celle qui avait été proposée par le rapport du troisième mandat, et sur la mise en place de leçons de soutien axées sur les compétences basales. Ces leçons de soutien sont encore en vigueur dans les gymnases du canton, avec notamment au collège Saint-Michel une heure par semaine ; selon GCM (2021), ces leçons peuvent entre autres s'appuyer sur :

- un recueil cantonal d'exercices, encore à rédiger (c'est là que le présent travail s'instaure) ;
- un recueil d'exercices déjà existant (EBERLE, BRÜGGENBROC et al. 2014, chap. 6.5, p. 82 *seqq.*), dit de Christof WEBER ;
- le recueil de compétences basales du canton de Berne (BKD 2019) ;
- les manuels de mathématiques issus du cycle d'orientation ;
- ainsi que sur les précédents rapports des mandats susmentionnés.

Le rapport en question (GCM 2021) contient même un chapitre résumant les outils élaborés dans les divers cantons, et leur taux de succès auprès du corps enseignant.[6]

1.2 Modalités du présent travail

Dans ce laboratoire de didactique, l'accent est mis sur la réalisation d'un projet concret permettant de pousser à la progression d'une des compétences basales en mathématiques que l'on peut retrouver dans l'annexe au PEC de la maturité gymnasiale national (CDIP 2016). Nous avons ainsi développé une ressource pour générer des exercices et corrigés centrés sur l'une des compétences basales.

Globalement, la compétence choisie est celle décrite dans l'annexe au PEC comme « utiliser avec souplesse l'outillage mathématique ». C'est-à-dire que les élèves doivent

[6]Un exemple très développer est la ressource Lernnavi (`https://lernnavi.ch/`), qui promet d'aider à développer les compétences en allemand et en mathématiques, tout en identifiant les points faibles de ses utilisat·eu·rice·s afin de les entraîner. Les soussignés n'ont toutefois pas réussi à faire fonctionner le site depuis leurs appareils.

Compétence 10
Factoriser une expression polynomiale à coefficients entiers à l'aide d'une mise en évidence d'un monôme ou d'un binôme ou à l'aide d'un produit remarquable du second degré.

La factorisation d'un trinôme du second degré n'est pas considérée comme une compétence basale.

Exemples 10.1
Factoriser les expressions suivantes le plus possible.

(a) $15x^3 + 10x$
(b) $3x(x + 1) + 5(x + 1)$
(c) $(x - 2)(x + 8) + (x - 2)(x - 5)$
(d) $a^2 + 8a + 16$
(e) $16y^2 - 9$
(f) $36x^2 - 84x + 49$

Contre-exemples 10.2
Factoriser les expressions suivantes le plus possible.

(a) $x^2 - x - 12$
(b) $2x^2 - 8x - 10$
(c) $(2x - 5)(4x - 7) - 3(5 - 2x)$
(d) $x^5 - x^3 + x^2 - 1$
(e) $x^4 - 4$
(f) $x^3 - 16x^2 - 64x$

FIG. 2 : La compétence 10 présentée dans le rapport du troisième mandat (GCM 2018).

Compétence 14
Résoudre une équation polynomiale du second degré, sans paramètre, à coefficients entiers et dont les membres gauches et droits peuvent être simplifiés selon les critères de la compétence 9.

Compétence 17
Résoudre un système régulier de deux équations linéaires.

FIG. 3 : Les compétences basales 14 et 17, correspondant à celles choisies pour le présent travail, telles qu'elles sont décrites dans GCM (2018). *La compétence 9 citée dans l'intitulé de la compétence 14 concerne la réduction d'une expression polynomiale à au plus trois termes.*

être capables de choisir, parmi leurs connaissances mathématiques, la ou les méthodes les plus appropriées à l'exercice ou au problème auquel iëls sont confronté·e·s. L'approfondissement de cette compétence dans tous les domaines parcourus en mathématiques au gymnase serait un travail titanesque, mais nécessaire dans la continuité des mandats présentés en section 1.1.2. Ainsi, les outils développés par les auteurs s'inscrivent chacun dans un thème bien précis, respectivement les équations quadratiques et les systèmes d'équations linéaires à deux inconnues. Par conséquent, si l'on reprend la liste de quarante-trois compétences données dans GCM (2018), les compétences 14 et 17 sont le sujet du présent travail (v. fig. 3). Une discussion autour du choix des thèmes est l'objet de la section 1.2.2.

1.2.1 Notre but

Le choix de compétences étant fait, comment les entraîner dans la pratique ? Une situation concrète illustrant le propos est donnée dans l'encadré.

> **Exemple :** Supposons l'élève confronté·e à l'équation $x^2 + 5x - 14 = 0$. Le but est qu'iël se rende compte que le polynôme peut être factorisé pour en soutirer les racines : $(x-2)(x+7) = 0$. Cette méthode est bien plus efficace que d'appliquer sans réfléchir la formule de Viète (même si elle marche également).

Comment faire maintenant pour que les exercices poussent à la compétence visée, qui n'est pas uniquement la résolution des équations ? Évidemment, il ne suffira pas de donner aux élèves la solution des équations pour les pousser à en accélérer la résolution. Une autre approche s'impose.

Nous proposons, dans notre projet, de créer un document LaTeX qui, à la compilation, génère un `.pdf` d'exercices de résolutions d'équations avec corrigés mettant l'accent sur la méthode et non la solution.[7] Pour reprendre l'exemple ci-dessus, le corrigé de la version compilée donnerait en bas de page

Exercice 1 : par factorisation du trinôme, $S = \{2; -7\}$.

Ce faisant, les élèves seront d'office confronté·e·s à une méthode de résolution potentiellement différente de la leur. En la comparant à celle qu'iëls auront empruntée, les élèves seront amené·e·s à remettre en cause leur *modus operandī*, et encouragé·e·s par la même occasion à l'optimiser, ayant goûté au gain de temps envisageable. Une consigne adaptée devra évidemment accompagner l'exercice.

[7] En effet, LaTeX (dans sa variante LuaLaTeX) permet d'implémenter directement dans son code le langage de programmation Lua, qui se compile en même temps que le fichier `.tex`. Ceci nous donne accès à toute la richesse d'un langage de programmation à part entière, permettant ainsi le traitement de logique plus complexe. C'est cette implémentation qui a été choisie pour la suite du projet.

1.2.2 Notre choix

En parcourant les compétences basales en mathématiques dans l'annexe au PEC (CDIP 2016), on remarque qu'elles sont divisées en deux : d'une part les thèmes de base (savoirs) et d'autre part les exigences de base (savoir-faire). En l'espèce, nous avons chacun choisi un thème qui nous semblait adapté à une automatisation informatique et facilement autocorrectif : les équations du second degré pour l'un et les systèmes d'équations linéaires (précisés dans l'annexe : à deux équations et deux inconnues) pour l'autre.

À ces thèmes il fallait maintenant ajouter une exigence. Nous avions le choix entre les trois présentées dans l'annexe :

- « utiliser avec souplesse l'outillage mathématique », *i. e.* faire preuve d'une approche flexible des outils et d'une utilisation appropriée à la spécificité de l'exercice ou problème donné ;

- « manier de manière adaptative graphiques, représentations tridimensionnelles, formules et statistiques », *i. e.* verbaliser un graphique ou une formule ou à l'inverse, formaliser un texte ou une représentation graphique ;

- « établir des liens entre les concepts mathématiques ».

Le deuxième point a été vite écarté lors de nos réflexions pour deux raisons principales. Premièrement, nos connaissances informatiques ne permettaient pas de prendre en compte la difficulté algorithmique d'un tel travail pour des graphiques tri- ou même bidimensionnels dans la génération automatique d'exercices et de corrigés. Deuxièmement, dans le cas spécifique des thèmes choisis, bien que comportant un important bagage géométrique (résoudre un système d'équations revient à calculer le point d'intersection entre deux droites, une équation de second degré à chercher les zéros d'une parabole), il nous semblait nécessaire de se concentrer sur le développement d'une seule exigence, afin de ne pas surcharger les élèves avec des informations qu'iëls ne retiendraient sûrement pas. Il serait néanmoins intéressant de compléter ce présent travail d'une approche géométrique pour lier les chapitres qui, aux yeux des étudiant·e·s, sont souvent peu rattachés.

Le troisième point a similairement été évincé pour des raisons de difficulté de mise en pratique : une exigence si vaste est difficilement soumise à une approche algorithmique.

L'exigence choisie a donc été l'utilisation avec souplesse de l'outillage mathématique. Dans notre cas cela signifiait : choisir la *bonne* méthode de résolution (combinaison linéaire ou substitution pour les systèmes linéaires, factorisation, isolation directe, Viète et autres pour les équations du second degré). Par rapport à la fig. 2, nous avons décidé de tout de même présenter des équations quadratiques présentant des factorisations par trinôme pour une raison bien spécifique : savoir reconnaître différentes méthodes de résolution fait partie du domaine des compétences transversales. En effet, il nous semblait plus important qu'un·e élève sache reconnaître qu'un polynôme donné ne soit *pas* issu d'une identité remarquable plutôt que de réussir à le factoriser par trinôme. Si, de surcroît, les élèves peuvent repartir avec la capacité de factoriser rapidement tout polynôme de second degré s'y prêtant, celle-ci ne sera guère perdue.[8]

1.3 Produit final

Ainsi avons-nous visé la création de deux fiches — l'une pour les systèmes linéaires d'équations à deux inconnues, l'autre pour les équations de second degré — posant une série d'équations à résoudre, et proposant pour chacune de celles-ci ses solutions *ainsi que la méthode jugée optimale pour y parvenir*. Chacune des fiches doit être générée algorithmiquement à partir de nombres pseudoaléatoires.

Deux exemplaires de fiches obtenues ainsi sont donnés en figg. 4 et 5.

1.4 Usage prévu (exemple)

Les enseignant·e·s sont bien évidemment libres de faire de cette ressource l'usage qui leur convient. Cependant, pour aiguiller la pensée, nous donnons ici un exemple de l'utilisation qui pourrait en être faite, et pour laquelle la mise en page des figg. 4 et 5 a été conçue.

Étape 1 : Préparation enseignante

L'enseignant·e prépare à l'avance un nombre de fiches égal au nombre d'élèves dans la classe. Chaque fiche est unique. L'enseignant·e coupe ensuite la pile de fiches le long des traits tillés, créant ainsi un lot de fiches d'exercices chacune avec sa fiche de solutions.

Étape 2 : Travail individuel

L'enseignant·e distribue ensuite les fiches d'exercices, une par élève. Les élèves ont ensuite la tâche de résoudre chacune des vingt-quatre (ou dix-huit, ou douze, ou $n \in \mathbb{N}$) équations. Une fois toutes les équations résolues, l'élève demande sa feuille de réponse (identifiée par la date et l'heure de création), et est tâché·e de s'autocorriger. Pour chaque équation, l'élève doit indiquer s'iël a suivi ou non la même méthode de résolution que celle indiquée dans la feuille de réponse.

Étape 3 : Mise en commun

L'enseignant·e demande à chaque élève de citer une seule équation pour laquelle la méthode optimale de résolution lui est parue difficile à repérer. Une fois toutes ces équations (et méthodes de résolution) récoltées, l'enseignant·e entame une discussion avec la classe pour en déterminer les points communs.

[8] Nous donnons toutefois aux enseignant·e·s la possibilité d'éliminer entièrement la factorisation par trinôme : voir code 3.4, p. 19.

Résolvez ces équations en utilisant à chaque fois la méthode la plus rapide possible. (N'oubliez pas d'écrire l'équation sous la forme standard si elle ne l'est pas déjà.) Une fois toutes les équations résolues, vérifiez vos réponses dans le corrigé. Indiquez pour chaque question si vous avez utilisé la même méthode que le corrigé ou non. Vous n'avez pas droit à la calculatrice.

Exercices

(générés le 2024-05-03 à 10h49m25s)

01. $-5x^2 - 75x - 280 = 0$
02. $7x^2 + 105x + 350 = 0$
03. $x^2 - 10x + 25 = 0$
04. $-3x^2 + 6x + 72 = 0$
05. $x^2 + 4x - 32 = 0$
06. $2x^2 = 4x + 16$
07. $3x^2 - 18x + 3 = 0$
08. $2x^2 + 16x + 53 = 0$
09. $-4x^2 + 9x - 52 = 0$
10. $5x^2 + 14x - 38 = 0$
11. $3x^2 + 2x - 21 = 0$
12. $5x^2 - 40x - 165 = 0$
13. $-4x^2 + 52x - 48 = 0$
14. $-8x^2 + 48x - 40 = 0$
15. $-3x^2 + 9x = 0$
16. $x = \frac{1}{9}x^2 + 2$
17. $-3x^2 = 1$
18. $-2x^2 = -8x - 64$
19. $-8x^2 + 20x - 25 = 0$
20. $4x^2 - 12x - 160 = 0$
21. $-x^2 = -16x + 60$
22. $-5x^2 - 45x + 110 = 0$
23. $-3x^2 + 18x + 216 = 0$
24. $8x^2 - 24x - 320 = 0$

....

Réponses

(générées le 2024-05-03 à 10h49m25s)

01. diviser par -5, puis par factorisation du trinôme, $S = \{-8; -7\}$
02. diviser par 7, puis par factorisation du trinôme, $S = \{-5; -10\}$
03. par identité remarquable (carré parfait), $S = \{5\}$
04. diviser par -3, puis par factorisation du trinôme, $S = \{6; -4\}$
05. par factorisation du trinôme, $S = \{4; -8\}$
06. diviser par 2, puis par factorisation du trinôme, $S = \{-2; 4\}$
07. formule quadr., $S = \left\{3 + 2\sqrt{2}; 3 - 2\sqrt{2}\right\} \approx \{0{,}172; 5{,}83\}$
08. formule quadr. / en calculant le discriminant (Δ), $S = \varnothing$
09. formule quadr. / en calculant le discriminant (Δ), $S = \varnothing$
10. formule quadr., $S = \left\{-\frac{7}{5} + \frac{\sqrt{239}}{5}; -\frac{7}{5} - \frac{\sqrt{239}}{5}\right\} \approx \{-4{,}49; 1{,}69\}$
11. formule quadr., $S = \left\{-3; \frac{7}{3}\right\} \approx \{-3{,}0; 2{,}33\}$
12. diviser par 5, puis par factorisation du trinôme, $S = \{11; -3\}$
13. diviser par -4, puis par factorisation du trinôme, $S = \{1; 12\}$
14. diviser par -8, puis par factorisation du trinôme, $S = \{1; 5\}$
15. par mise en évidence de $-3x$, $S = \{0; 3\}$
16. multiplier par 9, puis par factorisation du trinôme, $S = \{3; 6\}$
17. somme de nombres du même signe ne fait jamais zéro, $S = \varnothing$
18. diviser par -2, puis par factorisation du trinôme, $S = \{-4; 8\}$
19. formule quadr. / en calculant le discriminant (Δ), $S = \varnothing$
20. diviser par 4, puis par factorisation du trinôme, $S = \{8; -5\}$
21. diviser par -1, puis par factorisation du trinôme, $S = \{6; 10\}$
22. diviser par -5, puis par factorisation du trinôme, $S = \{2; -11\}$
23. diviser par -3, puis par factorisation du trinôme, $S = \{-6; 12\}$
24. diviser par 8, puis par factorisation du trinôme, $S = \{-5; 8\}$

FIG. 4 : Exemple d'une fiche pour la résolution efficace d'équations de second degré.

Résolvez ces systèmes d'équations en utilisant à chaque fois la méthode la plus rapide possible. (N'oubliez pas d'écrire l'équation sous la forme standard si elle ne l'est pas déjà.) Une fois toutes les équations résolues, vérifiez vos réponses dans le corrigé. Indiquez pour chaque question si vous avez utilisé la même méthode que le corrigé ou non. Vous n'avez pas droit à la calculatrice.

Exercices

(générés le 2024-05-02 à 16h27m14s)

01. $\begin{cases} 9x-7y=0 \\ 9x-6y=7 \end{cases}$

02. $\begin{cases} -2x+3y=-12 \\ 4x=15 \end{cases}$

03. $\begin{cases} 5x+y=-8 \\ -10x+9y=-10 \end{cases}$

04. $\begin{cases} -9x-y=-9 \\ 54x+6y=54 \end{cases}$

05. $\begin{cases} 3x+4y=-1 \\ 18x+24y=-6 \end{cases}$

06. $\begin{cases} -8y=11 \\ -32y=11 \end{cases}$

07. $\begin{cases} 8x-8y=1 \\ -3y=3 \end{cases}$

08. $\begin{cases} x+2y=0 \\ -6x-12y=0 \end{cases}$

09. $\begin{cases} -3x+10y=-11 \\ -x+y=-12 \end{cases}$

10. $\begin{cases} 8x+7y=-13 \\ 2x-2y=13 \end{cases}$

11. $\begin{cases} -8x+2y=9 \\ 4x+8y=-2 \end{cases}$

12. $\begin{cases} -x-7y=-13 \\ -5x-4y=-2 \end{cases}$

13. $\begin{cases} -10x+10y=-12 \\ -7x+5y=9 \end{cases}$

14. $\begin{cases} 8x-9y=1 \\ -5x-7y=10 \end{cases}$

15. $\begin{cases} -4x-6y=-4 \\ 5x+6y=-9 \end{cases}$

16. $\begin{cases} -10x+10y=8 \\ 8x+y=4 \end{cases}$

17. $\begin{cases} -2x+8y=3 \\ -4x+16y=6 \end{cases}$

18. $\begin{cases} x-10y=-5 \\ 10x+y=-15 \end{cases}$

....

Réponses

(générées le 2024-05-02 à 16h27m14s)

01. soustraction directe des deux équations, $S = \{(\frac{49}{9}, 7)\}$
02. isoler x dans la 2e équation, substituer ensuite, $S = \{(\frac{15}{4}, -\frac{3}{2})\}$
03. isoler y dans la 1re équation, substituer ensuite, $S = \{(-\frac{62}{55}, -\frac{26}{11})\}$
04. les deux équations sont dépendantes, $S = \{(x, 9-9x) \mid x \in \mathbb{R}\}$
05. les deux équations sont dépendantes, $S = \{(x, -\frac{1}{4} - \frac{3x}{4}) \mid x \in \mathbb{R}\}$
06. surdéfini, $S = \varnothing$
07. isoler y dans la 2e équation, substituer ensuite, $S = \{(-\frac{7}{8}, -1)\}$
08. sans solutions, $S = \varnothing$
09. isoler y dans la 2e équation, substituer ensuite, $S = \{(\frac{109}{7}, \frac{25}{7})\}$
10. diviser par -2 puis isoler y dans la 2e équation, substituer ensuite, $S = \{(\frac{13}{6}, -\frac{13}{3})\}$
11. diviser par 2 puis isoler y dans la 1re équation, substituer ensuite, $S = \{(-\frac{19}{18}, \frac{5}{18})\}$
12. diviser par -1 puis isoler x dans la 1re équation, substituer ensuite, $S = \{(-\frac{38}{31}, \frac{63}{31})\}$
13. diviser par 10 puis isoler y dans la 1re équation, substituer ensuite, $S = \{(-\frac{15}{2}, -\frac{87}{10})\}$
14. par combinaison linéaire, $S = \{(-\frac{83}{101}, -\frac{85}{101})\}$
15. addition directe des deux équations, $S = \{(-13, \frac{28}{3})\}$
16. isoler y dans la 2e équation, substituer ensuite, $S = \{(\frac{16}{45}, \frac{52}{45})\}$
17. les deux équations sont dépendantes, $S = \{(x, \frac{3}{8} + \frac{x}{4}) \mid x \in \mathbb{R}\}$
18. isoler x dans la 1re équation, substituer ensuite, $S = \{(-\frac{155}{101}, \frac{35}{101})\}$

FIG. 5 : Exemple d'une fiche pour la résolution efficace d'un système d'équations.

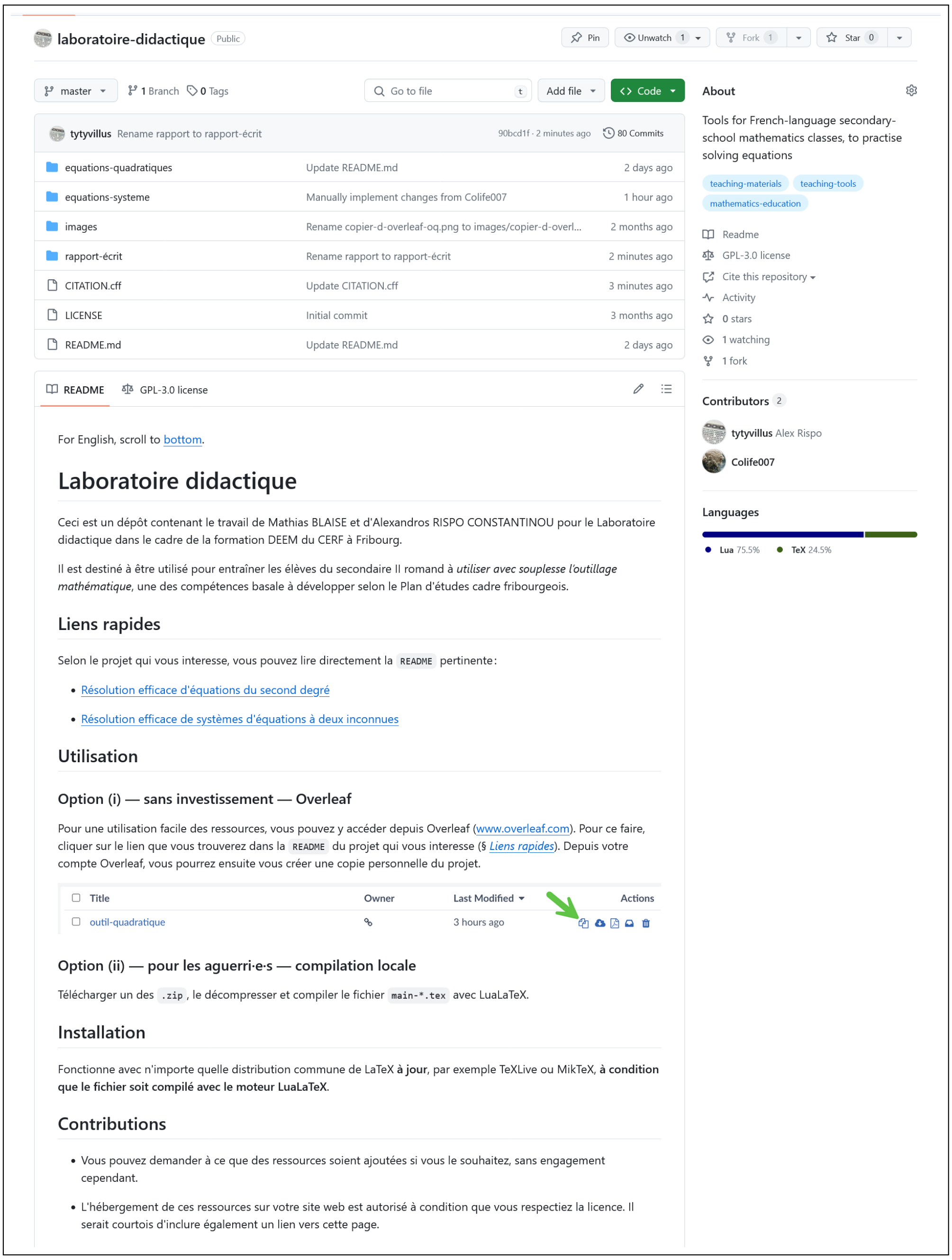


FIG. 6 : Le dépôt GitHub du travail. *Capture d'écran depuis la page d'accueil, prise en avril 2024.*

— 2 —

Mise à disposition du travail

POUR QUE LA COMMUNAUTÉ des enseignant·e·s de mathématiques puisse accéder à notre travail, nous avons décidé de lui créer un dépôt GitHub (RISPO CONSTANTINOU et BLAISE 2024). Ainsi, l'accès au projet est garanti en tout temps, et la possibilité reste ouverte pour que celui-ci évolue ; que ce soit l'un des auteurs à vouloir y apporter des modifications, ou encore une tierce personne, le système des *forks* et des *pull requests* le leur permettra aisément.

2.1 Le dépôt GitHub

Une capture d'écran du dépôt, prise depuis la page d'accueil, est reproduite en fig. 6. L'on y voit notamment la structure du dépôt, qui est divisé en quatre dossiers :

- `equations-quadratiques`, contenant la ressource visée à la résolution d'équations quadratiques ;
- `equations-systeme`, avec la ressource pour la résolution de systèmes à deux équations linéaires en deux inconnues ;
- `images`, contenant les fichiers `.png` nécessaires au bon affichage des `README` ;
- et `rapport-écrit`, avec le fichier `.pdf` du présent rapport.

Il y a également dans la page d'accueil une `README.md` globale pour le dépôt, un fichier `LICENSE` avec la licence choisie pour la diffusion du travail, et un fichier `CITATION.cff`, permettant la juste attribution bibliographique de l'œuvre. Nous avons choisi de distribuer le projet sous la *GNU General Public License* (version 3, abrégée « GPLv3 »), une licence *copyleft*, afin qu'il puisse facilement être adapté aux besoins de ses utilisat·eur·ice·s. Pour ce qui est de la mention des auteurs, la GPLv3 interdit explicitement d'imposer la citation systématique des auteurs d'origine ; nous nous bornons à rappeler — dans la `README` principale — que la juste attribution du travail d'autrui relève de la politesse et de la bonne éducation.

La `README` principale comporte une brève description du projet, des liens vers les deux sous-projets, une section « Démarrage rapide » (« *Quickstart* »), la licence, et une notice sur les possibilités de contributions ou d'hébergement externe. Ce document est également traduit en anglais pour prévenir une potentielle future traduction des ressources composant le projet.

2.2 Utilisation de la ressource

Dans chacun des deux dossiers spécifiques, c'est-à-dire `equations-quadratiques` et `equations-systeme`, sont présents sept fichiers (ici indiqués pour le dossier quadratique, pour l'autre le mot `quadratique` est à remplacer par `systeme`) :

(i) `README.md`, spécifique à l'outil ;

(ii) `QUICKSTART.md`, également spécifique à l'outil, donnant toutes les informations nécessaire à un démarrage tout en vitesse (utilisation + modification simple) ;

(iii) `codeQuadratique.lua`, contenant l'algorithme de génération des équations ainsi que de leur solution ;

(iv) `preamble-quadratique.tex`, le préambule du fichier `.tex` ;

(v) `main-quadratique.tex`, le fichier principal à ouvrir et compiler en LuaLaTeX avec une distribution LaTeX à jour ;

(vi) `main-quadratique.pdf`, un exemple d'une feuille de travail générée par l'outil en question ;

(vii) `DOWNLOADME-outil-quadratique.zip`, un fichier condensé, facile à télécharger, contenant les fichiers (i–v).

Pour utiliser la ressource, deux possibilités existent, décrites clairement et abordablement dans les `QUICKSTART`. La première, d'utilisation facile, est de suivre un lien depuis la `QUICKSTART` vers un projet Overleaf, d'en faire une copie, puis de laisser Overleaf compiler les documents. Les liens Overleaf sont les suivants :

- pour les équations de second degré,

  `https://www.overleaf.com/read/tsmdmkhnrzjq#7569b4` ;

- et pour les systèmes d'équations à deux inconnues,

  `https://www.overleaf.com/read/hmxjgqgkkmcq#9ccf30`.

La deuxième possibilité, plus technique, mais pas pour autant beaucoup plus difficile, consiste à télécharger le `DOWNLOADME-outil-*.zip`, de le dézipper, et de compiler le fichier `main-*.tex` avec une installation LuaLaTeX locale. Ici, « `*` » tient lieu de `quadratique` ou `systeme`.

Les `QUICKSTART` contiennent, en outre, des consignes au sujet des commandes mises à disposition depuis le document `main-*.tex` pour mettre en page les fiches, comme par exemple chez les équations de second degré la commande `\aprintqna`, qui imprime l'équation sur une ligne, suivie de la réponse.

Les `README`, elles, donnent des explications plus détaillées sur le fonctionnement interne des deux outils, et rajoutent également quelques conseils ou fonctionnalités additionnelles. Par exemple, pour la compilation locale, la `README` explique comment générer avec une seule commande plusieurs fiches d'un coup. Pour un système d'exploitation basé sur Linux, on peut simplement taper le code suivant dans Bash depuis le dossier contenant `main-*.tex` :

```
for i in {1..20}; do
   lualatex -jobname feuille-$i main-*.tex;
done
```

Notez que le code est ici séparé en lignes différentes pour qu'il rentre sur la page ; dans Bash, il est recommandé de tout taper sur la même ligne. Quelques adaptations pourraient être nécessaires, par exemple du nom du moteur LuaLaTeX (`lualatex`, `lualatex.exe` ou autre). Sachez également que les commandes sont à adapter à la syntaxe particulière du terminal du système d'exploitation ; nous donnons dans les `README` aussi l'exemple pour Windows PowerShell.

Elles expliquent de plus comment accéder aux fonctions internes du code Lua et, véritable apogée de notre projet, comment combiner les fonctionnements des deux outils. Un document Overleaf (encore expérimental !) illustrant cette fonctionnalité est disponible au lien suivant :

`https://www.overleaf.com/read/wzdcckddkjzy#f3d012`.

— 3 —

Structure des algorithmes

Les algorithmes sont, comme cité plus haut, principalement implémentés en langage Lua dans un fichier `.lua` appelé par le document `.tex` (Pégourié-Gonnard 2012). Les calculs sont donc effectués par Lua, et LaTeX se charge ensuite de la mise en page. Un troisième logiciel, le paquet `luacas` (Cochrane et All 2023) disponible sur CTAN, se charge pour certaines expressions complexes de leur simplification algébrique.

Dans cette section, nous passons les détails de la conversation entre Lua, LaTeX et `luacas`, et nous focalisons sur le fonctionnement général des algorithmes. Pour les enseignant·e·s désirant adapter le code à leur usage spécifique, différentes options de personnalisation sont mises à disposition depuis le document `.tex` ; pour la modification des algorithmes, certains paramètres sont facilement modifiables dans une section dédiée, au début du fichier `.lua`. Préparer un manuel détaillé documentant toutes les modifications de ce genre possibles aurait été une tâche gargantuesque ; pour les utilisat·eur·ice·s souhaitant mettre les mains à la pâte, les `QUICKSTART` et `README` sont disponibles, et les divers éléments du fichier `.lua` sont équipés de commentaires fréquents visés à élucubrer leur fonctionnalité. Nous espérons que ces commentaires suffiront.

3.1 Équations de second degré

Le code complet et détaillé de cet algorithme-ci est à trouver dans le code A.9 (p. 58), mais n'y est pas particulièrement adapté à la lecture non-initiée. Nous donnons donc ici une version simplifiée en pseudocode.

L'algorithme est assez complexe, et se découpe en quatre grandes parties : la génération des équations de type $ax^2 + bx + c = 0$ à résoudre (polynomial()), leur impression (printEquation()), le choix de la méthode la plus efficace avec résolution de l'équation (method() et solution()) et enfin l'impression de la solution (answer()). Cette logique est celle donnée en pseudocode dans le code 3.1.

Code 3.1 : Algorithme global pour les équations du second degré.

```
fonction  fullRoutine ()
    début
        tableau  polynomialCoeffs  != polynomial()
```

```
        // génère les coefficients a, b, c de ax^2 + bx + c

    chaîne eqnString != printEquation(polynomialCoeffs)
        // crée une chaîne avec "ax^2 + bx + c = 0" (ou autre équation)

    chaîne ansString != answer(
            method(polynomialCoeffs),
            solution (polynomialCoeffs)
        )
        // crée une chaîne avec "[méthode], S = {x_1, x_2}"

    toTEX(eqnString ++ ansString)
        // envoie les réponses à TeX pour qu'il les imprime
fin
```

Prenons donc ces composantes une à une, en commençant avec la génération des coefficients de polynôme.

3.1.1 Coefficients de polynôme

Pour s'assurer que les polynômes générés soient intéressants à résoudre d'un point de vue technique, et non uniquement des polynômes à résoudre avec la formule de Viète, nous avons utilisé deux méthodes différentes s'alternant de façon aléatoire. La première génère d'abord les solutions $x_1, x_2 \in [\![-15, 15]\!]$,[9] et en déduit les coefficients du polynôme; la deuxième choisit directement trois coefficients $a \in [\![-10, 10]\!]$, $b \in [\![-20, 20]\!]$ et $c \in [\![-60, 60]\!]$. La première méthode est choisie dans sept cas sur dix. Le principe est illustré dans le code 3.2.

Code 3.2 : Génération de coefficients pour équations de second degré.

```
fonction polynomial()
    début
        booléen fromSols != aléatoirement(vrai, 7/10 | faux, 3/10)

        si fromSols = vrai,
            alors // définit aléatoirement les solutions
                x1, x2 ← [[−12, 12]]
                si aléatoirement(vrai, 1/10 | faux, 9, 10)
                    x2 != x1 // insiste que ~10\% des cas fromSols...
                finsi // ... soient des carrés parfaits
                a ← [[−10, 10]] ∖ {0}
                b != − a ∗ (x1 + x2)
                c != a ∗ x1 ∗ x2
            sinon // définit aléatoirement les coefficients a, b, c
                a ← [[−10, 10]]
                b ← [[−20, 20]]
```

[9]La notation $[\![m, n]\!]$ est ici utilisée pour signifier l'ensemble $\{m, m + 1, \ldots, n - 1, n\}$, où $m < n \in \mathbb{Z}$.

```
            c ← ⟦−60, 60⟧
            x1 != (− b − sqrt(b^2 − 4*a*c)) / (2*a)
            x2 != (− b + sqrt(b^2 − 4*a*c)) / (2*a)
            // il y des protections pour rendre nil si la solution
            // n'existe ou si l'équation est linéaire/autre
        finsi

        n != longueur( filtre ( {x1, x2} | # ≠ nil ) )
        // n est défini comme le nombre de solutions
        // peut être égal à 0, 1, 2 ou infini

        rendre {a, b, c, n, x1, x2}
    fin
```

Une fois l'équation générée, il faut choisir la méthode « la plus efficace » pour la résoudre. L'algorithme consiste essentiellement à vérifier toutes les situations possible (code 3.3), avec une nuance : la fonction easyFactor() (code 3.4).

Code 3.3 : Choix de la méthode « la plus efficace ».

```
// n = nombre de solutions, équation est ax²+bx+c=0
chaîne méthode :=
    si n = ∞ alors "par évidence"
        // c'est le cas 0 = 0
    sinonsi n ≠ 0 alors
        si a = 0 alors "résoudre algébriquement"
            // équation de 1er degré
        sinonsi b = 0 alors "par réarrangement"
            // le cas ax² + c = 0
        sinonsi c = 0 alors "par mise en évidence"
            // le cas ax²+bx=0
        sinonsi easyFactor(a,x1,x2) alors
            si n = 1 alors "par identité remarquable (carré parfait)"
            sinon "par factorisation du trinôme" finsi
            // si différent de 1, rappeler de factoriser par a :
            si a ≠ 1 alors préfixer ("diviser par a, puis [...] ") finsi
        sinon "avec la formule de Viète"
        finsi
    sinonsi n = 0 alors
        si a = 0 et b = 0 et c ≠ 0 alors "évidemment"
            // c'est le cas 1 = 0
        sinonsi b = 0 "la somme de nombres positifs ne fait jamais 0"
            // c'est le cas ax²+c=0, sgn(a*c)>0
        sinon "en calculant le discriminant"
            // car le manque de solutions n'est pas immédiatement évident
        finsi
    finsi
```

La fonction easyFactor() est la méthode utilisée pour décider si un polynôme donné est facile à factoriser « à l'œil » ou non. Nous avons choisi de dire qu'un trinôme est facilement factorisable si les facteurs sont suffisamment petits, dans un sens formalisé dans le code 3.4. La fonction est volontairement séparée du reste du programme, afin que les utilisateurs du document puissent, au besoin, facilement la modifier.

Par exemple, si l'enseignant·e ne souhaite pas encourager la factorisation par trinôme, il suffit de poser

```
local function easy_factor(a, x1, x2) return false end
```

dans le code Lua, c'est-à-dire d'étiqueter tous les polynômes comme difficiles à factoriser.[10]

Code 3.4 : Décision de facilité à factoriser.

```
fonction booléenne easyFactor(a, x1, x2)
 // détermine si un polynôme est facile à factoriser
 // à partir de a et de ses racines
   début
      c != a∗x1∗x2
      rendre vrai si
         (    (max(|x1|,|x2|) ≤ 12 ou min(|x1|,|x2|) ≤ 3) et
              (|a| ≤ 5 ou a = 10 ou mod(c,10) = 0)    )
         ou |c| ≤ 144
      rendre faux sinon
   fin
```

Un commentaire, enfin, sur l'impression de l'équation : la forme de l'équation est aléatoirement choisie, pour avoir quatre formes possibles. Celles-ci sont les suivantes, avec la probabilité de leur apparition donnée entre parenthèses.

$$ax^2 + bx + c = 0 \qquad (80\,\%)$$

$$x = -\frac{c}{b} - \frac{a}{b}x^2 \qquad (6{,}66\,\%)$$

$$\frac{c}{a} = -x\left(x + \frac{b}{a}\right) \qquad (6{,}66\,\%)$$

$$ax^2 = -bx - c \qquad (6{,}66\,\%)$$

Ces trois types d'équation permettent de varier un peu la forme à résoudre et entraîner le réarrangement de l'équation sous la forme standard.

[10]Relisant aujourd'hui les détails de l'implémentation, je me rends compte que prendre cette mesure désactivera regrettablement aussi les cas d'identité remarquable de type $(A \pm B)^2$. Ce défaut pourrait être corrigé dans une version améliorée de l'outil en extirpant la condition `if num_sols == 1` de la vérification `elseif easy_factor(a, x1, x2)` (vers la ligne 247 du code A.9 ; voir sinon la condition $n = 1$ dans le code 3.3).

Une fois l'équation, les solutions et la méthode de résolution décidées, elles sont passées à LuaLaTeX, qui les met en page et les imprime. Pour le détails de ce processus, la ou le lect·eur·ice est invité·e à consulter les appendices, p. 31 *seqq*.

3.2 Systèmes d'équations

Le code complet de cet algorithme-ci est à trouver dans le code A.10 (p. 68). Son fonctionnement suit la logique reproduite dans le code 3.5.

Code 3.5 : Algorithme pour les systèmes d'équations

```
fonction  fullRoutine ()
    début
        tableau  systemCoeffs != generate_system()
        // génère les  coefficients  a1, a2, b1, b2, c1, c2 du système d'équations

        chaîne  eqnString  != printEquation(systemCoeffs)
        // crée une chaîne avec "{ a1x + b1y = c1 ; a2x + b2y = c2 " (ou autre équation)

        chaîne  ansString  != answer(
            method(systemCoeffs),
            solution (systemCoeffs)
        ) // crée une chaîne avec "[méthode], S = {(x, y)}"

        toTEX(eqnString ++ ansString)
        // envoie les réponses à TeX pour qu'il les imprime
    fin
```

Tout comme dans la précédente section, analysons en détails les composantes du pseudo-code ci-dessus.

3.2.1 Impression du système d'équation

Les systèmes d'équations traités étant uniquement linéaires et à deux inconnues, il y a peu de valeur à chercher à ce que les élèves doivent les mettre sous la forme standard. Tous les systèmes sont donc mis en page sous la forme

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases} .$$

3.2.2 Coefficients du système d'équations

Contrairement à la génération des coefficients dans les polynômes quadratiques, les coefficients sont ici générés purement aléatoirement, c'est pourquoi la plupart des solutions, lorsqu'il y en a, ne seront pas entières. Cependant, différentes méthodes de génération sont utilisées. La première méthode, qui intervient dans 10 % des cas, génère tout d'abord les coefficients de la première équation de manière aléatoire, tous compris dans l'intervalle $[\![-12, 12]\!]$, puis un coefficient $k \in [\![-7, 7]\!]$ qui viendra multiplier les derniers coefficients afin de générer la deuxième équation (ce qui nous donne une infinité de solutions, les deux équations étant linéairement dépendantes). La deuxième méthode, qui intervient dans 80 % des cas, génère de manière aléatoire tous les coefficients du système et vérifie que le système généré soit régulier

$$\left(\text{en calculant le déterminant de la matrice } \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}\right)$$

et possède ainsi une unique solution. La troisième méthode, qui intervient dans le reste des cas, est similaire à la première, en faisant attention à ce que $c_1 \neq k \cdot c_2$, pour n'avoir aucune solution (les droites sont strictement parallèles). Le principe est illustré dans le code 3.6.

Code 3.6 : Génération de coefficients pour équations de second degré.

```
fonction system(proba_singular)
    // proba_singular est donné par l'utilisateur et définit la probabilité
    // que le système généré n'ait aucune ou une infinité de solutions
    début
        booléen singular != aléatoirement(vrai, proba_singular |
                                          faux, 1−proba_singular)

        si singular = vrai,
            alors
                a1, b1, c1 ← ⟦−12, 12⟧,
                k ← ⟦−7, 7⟧,
                a2, b2, c2 != k∗(a1, b1, c1)
                // dans 50% des cas
            ou
                a1, b1, c1 ← ⟦−12, 12⟧,
                k ← ⟦−7, 7⟧,
                a2, b2 != k∗(a1, b1)
                c2 != c1
                // dans les 50% des cas restants
            systeme != {a1, b1, c1, a2, b2, c2}
            rendre systeme
        sinon
            a1, b1, a2, b2 ← ⟦−10, 10⟧
            c1, c2 ← ⟦−15, 15⟧
            systeme != {a1, b1, c1, a2, b2, c2}
            det != a1 ∗ b2 − b1 ∗ a1

            si det = 0
                rendre generate_system(proba_singular)
                // vérifie si le système est effectivement régulier ,
```

```
            // si ce n'est pas le cas, on regénère
        sinon
            rendre systeme

    n, x, y != solutions(a1, b1, c1, a2, b2, c2)
    // n est défini comme le nombre de solutions, une fonction
    // détermine ce nombre puis calcule les solutions x, y
    // peut être égal à 0, 1, ou infini

    rendre {systeme, n, x, y}
fin
```

Une fois le système généré et résolu par la règle de Cramer (plus simple à implémenter dans le code Lua) , il faut choisir la méthode « la plus efficace » pour le résoudre. Le principe est très similaire au code code 3.3 ci-dessus, et il n'y a rien de plus intelligent à faire que simplement de l'énumération de cas. Les cas sont les suivants.

Code 3.7 : Choix de la méthode « la plus efficace ».

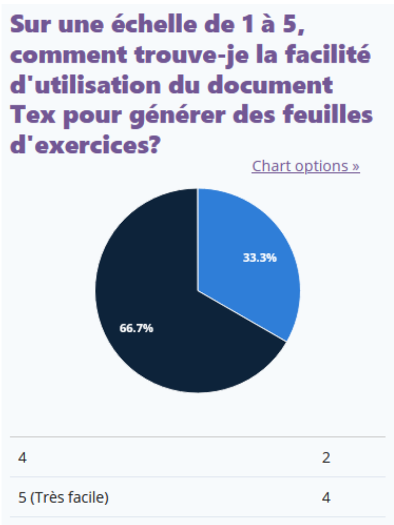
```
// n = nombre de solutions, systeme est le système
chaîne méthode :=
    si n = ∞ alors "les deux équations sont dépendantes"
    sinonsi n = 1 alors
        si a1=0 ou b1=1 alors "isoler y dans la première équation,
            substituer ensuite"
        sinonsi b1=0 ou a1=1 alors "isoler x dans la première équation,
            substituer ensuite"
        sinonsi a2=0 ou b2=1 alors "isoler y dans la deuxième équation,
            substituer ensuite"
        sinonsi b2=0 ou a2=1 alors "isoler x dans la deuxième équation,
            substituer ensuite"
        sinonsi a1=a2 ou b1=b2 alors "soustraction directe
            des deux équations"
        sinonsi a1=-a2 ou b1=-b2 alors "addition directe
            des deux équations"
        sinonsi a1 mod b1 = 0 alors "diviser par b1 puis isoler y dans
            la première équation, substituer ensuite"
        sinonsi b1 mod a1 = 0 alors "diviser par a1 puis isoler x dans
            la première équation, substituer ensuite"
        sinonsi a2 mod b2 = 0 alors "diviser par b2 puis isoler y dans
            la deuxième équation, substituer ensuite"
        sinonsi b2 mod a2 = 0 alors "diviser par a2 puis isoler x dans
            la deuxième équation, substituer ensuite"
        sinon "par combinaison linéaire "
        finsi
    sinon "sans solution (comb. lin. donne 1 = 0)"
    finsi
```

— 4 —

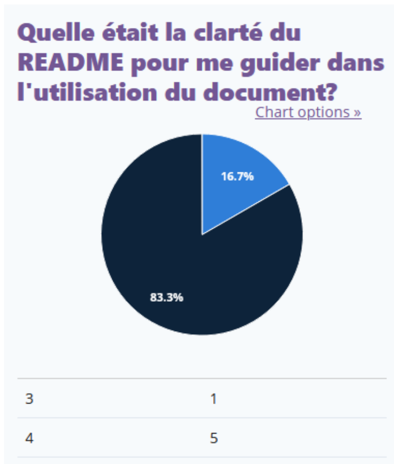
Rétroactions

4.1 Résultats du contact avec le corps enseignant

Avant de finaliser nos outils, nous avons souhaité obtenir un retour de la part du corps enseignant au sujet de son utilité et facilité d'utilisation perçues. Afin de récolter des ressentis concernant de tels documents permettant de générer des exercices autocorrigés, un lien Framaforms — pour éviter d'utiliser Google Forms — fut envoyé à différent·e·s enseignant·e·s de mathématiques du canton de Fribourg. Nous avons reçu six réponses (anonymes) à notre questionnaire. Les résultats sont reportés ci-dessous dans les figg. 7 et 8. Il est à noter que toutes les personnes ayant répondu au questionnaire ont utilisé la version en ligne (Overleaf) de l'outil et que les questions offraient toutes comme réponse une échelle allant de 1 (ne correspond pas) à 5 (correspond tout à fait).

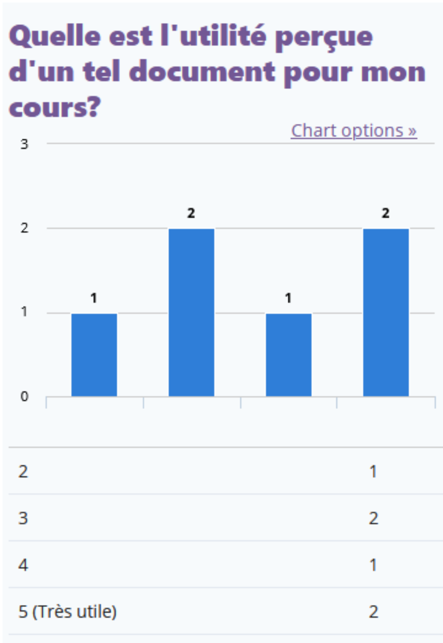


(a) La facilité d'utilisation de l'outil est mise en avant.

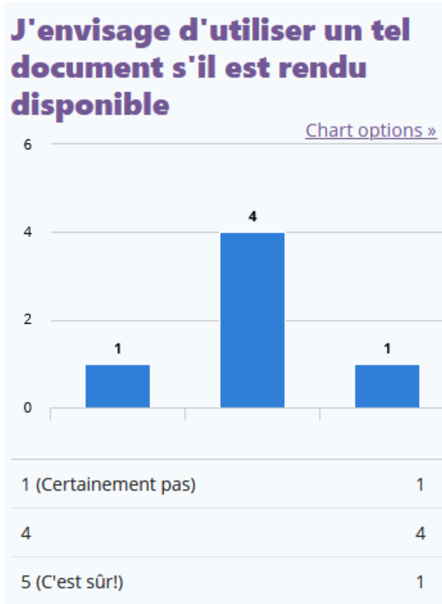


(b) La README semble bien éclaircir l'utilisation de l'outil.

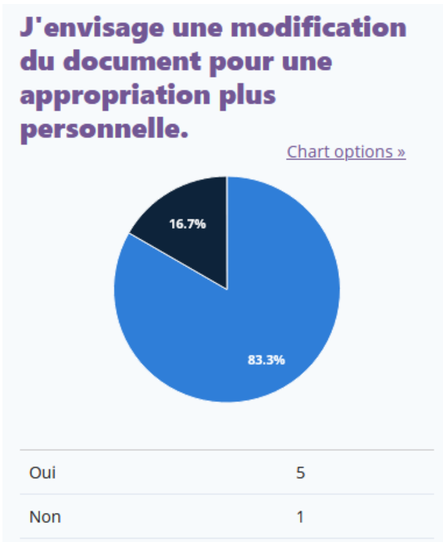
FIG. 7 : Les deux premières questions du questionnaire et leurs réponses associées.



(a) L'utilité perçue de l'outil divise.



(b) Si le document était mis à disposition, la grande majorité serait encline à l'utiliser. *La réponse « certainement pas » correspond à l'utilité perçue de 1 sur 5.*



(c) La grande majorité souhaiterait modifier le document pour le personnaliser.

(d) La modification semble tendanciellement atteignable.

FIG. 8 : Les questions successives du questionnaire et leurs réponses associées.

Plusieurs commentaires sont à faire sur ces résultats. En premier lieu, l'on remarque que l'utilité perçue de cet outil par les enseignant·e·s questionné·e·s (fig. 8a) est très divisée ; en effet, la moitié des réponses renvoient à une utilité moyenne (3) à faible (2). Les causes peuvent être multiples ; nous en citerons une ici, et parlerons d'améliorations possibles dans la section 4.2.

Une cause pourrait être que, bien qu'il soit utile pour les élèves de savoir faire usage de différentes méthodes afin de choisir la plus efficace dans un contexte donné, la majorité des élèves se satisfasse de l'étape *trouver un résultat*, peu importe la méthode. La compétence serait donc bien plus terre-à-terre, à décrire comme « résoudre des équations quadratiques » ou « résoudre des systèmes d'équations linéaires à deux inconnues ». La compétence naturellement visée par les élèves étant différente, nous comprenons l'écart qu'il existe entre cette dernière et celle choisie dans ce travail : l'une semble presque utopique. Cependant, l'outil élaboré permet de travailler les deux ; il suffit d'acquiescer auprès des élèves que la méthode donnée dans le corrigé est indicative.

Le deuxième commentaire concerne la modification du document (fig. 8c). Malgré une écrasante majorité souhaitant modifier le document pour une appropriation plus personnelle, l'espace « Remarques, suggestions et commentaires » à la fin du questionnaire ne permettait pas aux enseignant·e·s de laisser exprimer tout ce qu'iëls souhaitaient nous dire. À moins de retrouver ces personnes, nous ne pouvons que supputer des raisons poussant à vouloir modifier le document. En observant les résultats de la question suivante (fig. 8d), on peut imaginer que les modifications soient d'ordre esthétique, et que plus d'informations dédiées au fichier `preamble-*.tex` permettraient potentiellement de répondre assez facilement aux demandes d'adaptabilité.[11] Cependant, il serait très intéressant de pouvoir avoir une discussion plus élaborée sur les apports potentiels des enseignant·e·s ; nous tenterons de les contacter.

Dernier aspect du questionnaire : le champ des remarques, suggestions et commentaires. Ci-dessous, la liste exhaustive des commentaires reçus concernant ce travail.

(i) *12 équations devraient suffire — Pourquoi les élèves n'ont pas le droit à la calculatrice ?*

(ii) *Le terme formule de Viète est utilisé au CO notamment dans l'aide mémoire, mais n'est pas correct.*

(iii) *Bravo pour votre travail, il permet de gagner un temps considérable pour la création d'items si les étudiants en demandent plus.*

(iv) *Pas assez de place ici !!! Je ne pourrais faire que de petites modifications facilement (consigne, etc.), pas de grosse modif.*

(v) *Pas assez de place ici !!!*

[11] Notamment, nous avons pu mettre en place un ajout de ce type ; voir plus bas.

Certaines de ces remarques sont plus énigmatiques que d'autres. Avant de les commenter dans l'ordre afin d'apporter de potentielles réponses aux questions posées, il faut remarquer que l'une des personnes n'a pas laissé de remarque (malheureusement cette personne est celle ayant répondu que ce travail n'avait aucun utilité ; nous aurions bien aimé en discuter avec elle afin d'envisager diverses pistes d'améliorations).

Nos réponses sont les suivantes :

(i) Le nombre d'équations est, comme décrit dans la `README`, très facile à changer. Le nombre de 24 a été choisi simplement pour remplir une demi-page servant d'exemple. Le non-droit à la calculatrice était pour nous un choix pour pousser les élèves à réfléchir à la méthode à utiliser plutôt que de foncer, tête baissée, dans le calcul du discriminant. Néanmoins, il est vrai que ce choix est discutable.

(ii) Merci pour la remarque ; la formule de Viète correspond en effet à tout autre chose. Nous avons, depuis, réécrit l'outil pour qu'il emploie le terme de « formule quadratique », abrégé en « formule quadr. ».

(iii) Cette remarque nous touche beaucoup et nous permet de croire en notre travail, prenant en compte les différents retours ainsi que les améliorations discutées dans la section 4.2. Elle contraste particulièrement avec l'utilité nulle perçue par l'un·e de nos collègues enseignant·e·s.

Suite à l'obtention de ces retours, et à une discussion plus approfondie avec une enseignante de l'ECGF ayant testé ces outils, la structure de la `README.md` fut modifiée et un fichier `QUICKSTART.md` ajouté au dépôt GitHub, ceci afin de faciliter l'appropriation des documents par les enseignant·e·s. Nous espérons ainsi rendre plus abordable la modification et personnalisation de ces outils.

4.2 Améliorations et extensions possibles

4.2.1 Limitations du format par fiches

Le choix de mener à terme le projet à l'intérieur de LaTeX est relativement controversé. Pour que l'implémentation fonctionne, nous dûmes jongler entre trois « logiciels » différents (LaTeX, Lua et `luacas`), augmentant ainsi la complexité finale. Le projet est non-trivial à adapter, et rendre sa modification plus accessible serait une amélioration bien appréciée.

Cependant, les alternatives ne sont pas forcément meilleures : nous avions, à l'origine, pensé créer une interface Moodle proposant les équations directement aux élèves de façon interactive ; seulement une fois que celles- et ceux-ci auraient tapé leur solution putative dans un champ textuel, la réelle solution ainsi que la méthode conseillée auraient été affichées. L'avantage de cette implémentation aurait également été, dans le cas où l'élève se serait trompé·e dans sa réponse, que les différentes étapes de la résolution auraient pu être affichées, afin d'aider l'élève à comprendre son erreur. Malgré les bienfaits d'une telle approche, toutefois, nous avons vite été contraints à l'abandonner. Nous reproduisons ici un extrait de notre journal de bord :

```
## 28.11.2023

- Décision d'utiliser un script `lua` dans LuaLaTeX
  plutôt que le plugin Moodle «STACK»:

    1. STACK n'est utilisé ni à St-Michel, ni à l'ECGF;

    2. Il nous est peu clair si nous pouvons facilement
       avoir accès à STACK, et c'est un accès qu'on
       risque de perdre dès l'obtention du DEEM;

    3. STACK semble très puissant pour faire des
       calculs et manipulations en arrière-plan - grâce
       à l'intégration du système d'algèbre numérique
       (SAN) *Maxima* - mais ne semble pas
       particulièrement plus propice à la création de
       questions favorisant les compétences; ce que
       nous avions en tête est parfaitement réalisable
       au sein d'un langage de code classique non SAN.

    Nous avons donc opté pour un script `lua`, plus
       facilement accessible aux enseignants (même s'il
       faudra à l'enseignant compiler le document pour
       avoir les questions, sans que les élèves
       puissent de façon autonome aller sur Moodle et
       remplir les questions intéractivement).
```

L'option plus extrême, celle de coder *ā nihilō* un site web avec l'implémentation susdite, était très attirante, notamment pour sa liberté d'accès depuis la Toile, mais comportait hélas pour les auteurs — non rodés à la création de sites web — une trop forte charge de travail pour être mise en marche dans les délais de leur formation.

4.2.2 Intégration dans la pratique

Nous nous sommes beaucoup concentrés sur la production du travail en question, quelque peu au détriment de sa diffusion. Le temps investi — nécessairement ! — pour produire une ressource fiable était, par la force des choses, du temps qui n'a pu être consacré à la planification de l'intégration de notre ressource dans la pratique enseignante. Une extension possible à notre projet serait donc de contacter un plus grand nombre d'enseignant·e·s en leur demandant de tester nos outils au sein de leurs classes. Ceci permettrait de maximiser l'utilité du travail que nous avons fourni, ainsi que de générer plus de retours pouvant — qui sait ? — mener à d'ultérieures améliorations.

Il faut cependant souligner que cette étape de diffusion est difficile pour tout nouveau projet, et que nous n'eussions pu fournir des outils en si bon état d'achèvement si nous nous y fussions trop penchés.

4.2.3 Ajout d'autres outils

L'on pourrait également concevoir que l'un des auteurs — ou peut-être même une tierce personne — décide de reprendre la structure de base des outils que nous fournissons ici pour entraîner encore un autre type de calcul où un choix de méthode est à faire. Par exemple, le calcul des déterminants dans l'algèbre matricielle. Le mode de diffusion (GitHub) choisi pour notre ressource nous permettra facilement d'intégrer des extensions de ce type.

4.3 Conclusion

Le laboratoire de didactique est une opportunité intéressante offerte par la formation fribourgeoise à l'enseignement en écoles de maturité pour développer une approche qui n'aurait jamais sinon été suivie. Cette année, la contrainte était de produire des questions algorithmiquement générées. L'un des auteurs, en tout cas, a pris un malin plaisir à entraîner ses capacités de codage : c'est une activité agréablement différente de l'habituelle préparation de leçon. L'autre auteur, quant à lui, a pris un plaisir quelque peu moins malin ; habitué du langage Python et de Stack-Exchange, le langage Lua et ses subtilités lui ont causé quelques longues soirées. Néanmoins, l'expérience retirée d'un tel projet, tant du point de vue du codage que de la lecture des différentes mandats et rapports sur les compétences basales, est immense. Le bagage construit par ces mois de travail permettra, *quizás*, de créer un nouvel outil qui sera utilisé dès les premières années d'engagement des futurs enseignant·e·s.

Le succès du projet fut mitigé, bien que seule une minorité des personnes interrogées ne vît en ce projet aucune utilité : si la ressource développée fonctionne surprenamment bien, son adoption laisse quelque chose à désirer. Néanmoins, les auteurs souhaitent remercier leur superviseur pour ce projet, et seront par la suite d'autant plus reconnaissants des ressources enseignantes à leur disposition, dont la création ils savent à présent fatalement chronophage.

* * *

Achevé le 9 mai 2024 à Fribourg par
Alexandros Rispo Constantinou
et Mathias Blaise.

Références

BILDUNGS- UND KULTURDIREKTION DES KANTONS BERN Abteilung Mittelschulen [BKD] (22 fév. 2019). *Beispielaufgaben Mathematik Basale fachliche Studierkompetenzen*. Allemand. gym|MATUR. URL : `https://www.bkd.be.ch/content/dam/bkd/dokumente/de/themen/bildung/mittelschulen/gymnasium/ams-gym-bk-beispielaufgaben-mathematik.pdf` (visité le 05/05/2024).

COCHRANE Evan et ALL Timothy (26 mai 2023). *The `luacas` package*. Anglais. Version 1.0.2. URL : `https://ctan.org/pkg/luacas` (visité le 11/01/2024).

CONFÉRENCE SUISSE DES DIRECTEURS CANTONAUX DE L'INSTRUCTION PUBLIQUE [CDIP] (9 juin 1994). *Plan d'études cadre pour les écoles de maturité · du 9 juin 1994*. Berne. Chap. 5.2. 139 pp. URL : `https://www.cdip.ch/fr/documentation/textes-juridiques/recueil-des-bases-legales` (visité le 01/12/2023).

— (mars 2013). *Rapport annuel 2012*. Berne. 44 pp. URL : `https://edudoc.ch/nanna/record/107403/files/Jb_2012_f_web.pdf` (visité le 09/05/2024).

— (17 mars 2016). *Annexe au plan d'études cadre du 9 juin 1994 pour les écoles de maturité : Compétences de base en mathématiques et en langue première constitutives de l'aptitude générale aux études supérieures · du 17 mars 2016*. Berne. Chap. 5.2.2. 9 pp. URL : `https://www.cdip.ch/fr/documentation/textes-juridiques/recueil-des-bases-legales` (visité le 01/12/2023).

DIRECTION DE L'INSTRUCTION PUBLIQUE, DE LA CULTURE ET DU SPORT [DICS] (23 fév. 2015). *Mandat du Service de l'enseignement secondaire du 2e degré (S2) et de la Conférence des recteurs (CORECOFR) · Groupe de travail de « mathématiques »*. 3 pp.

EBERLE Franz, BRÜGGENBROC Christel, RÜEDE Christian et al. (15 oct. 2014). *Basale fachliche Kompetezen für allgemeine Studierfähigkeit in Mathematik und Erstsprache · Schlussbericht zuhanden der EDK*. Allemand. Éd. révisée du 12 janvier 2015. 284 pp. URL : `https://www.ife.uzh.ch/research/lehrstuhleberle/forschung/bfkfas/downloads/Schlussbericht_final_V7.pdf`.

EBERLE Franz, GEHRER Karin, JAGGI Beat et al. (31 oct. 2008). *Evaluation des Maturitätsreform 1995 (EVAMAR) · Schlussbericht zur Phase II*. Allemand. 418 pp. URL : `https://www.sbfi.admin.ch/sbfi/de/home/bildung/maturitaet/gymnasiale-maturitaet/evamar.html/`.

FREE SOFTWARE FOUNDATION (29 juin 2007). *GNU General Public License · version 3*. Anglais. URL : https://www.gnu.org/licenses/gpl-3.0.en.html (visité le 21/04/2024).

GROUPE CANTONAL DE MATHÉMATIQUES [GCM] (2016). *Analyse der zu erwartenden Kompetenzen · Teil 1 : Algebra im ersten Jahr*. Français et allemand. 14 pp.

— (juin 2017). *Schlussbericht der Arbeitsgruppe · 2. Mandat*. Français et allemand. 7 pp.

— (15 nov. 2018). *Compétences basales en première année gymnasiale dans le canton de Fribourg*. Français et allemand. 29 pp.

— (30 juin 2020). *Mise en œuvre des compétences basales en première année gymnasiale dans le canton de Fribourg durant l'année 2019–2020*. Français et allemand. 49 pp.

— (31 août 2021). *Rapport suite au cinquième mandat*. Français et allemand. 19 pp.

PÉGOURIÉ-GONNARD Manuel (23 jan. 2012). *The luacode package*. Anglais. Version 1.2a. URL : https://ctan.org/pkg/luacode (visité le 11/01/2024).

RISPO CONSTANTINOU Alexandros et BLAISE Mathias (2024). *Outils d'Adaptativité*. URL : https://github.com/tytyvillus/laboratoire-didactique.

SERVICE DE L'ENSEIGNEMENT SECONDAIRE DU DEUXIÈME DEGRÉ [S2] (2 oct. 2017). *3e mandat du Service de l'enseignement secondaire du 2e degré (S2) et de la Conférence des recteurs (CORECOFR) · Groupe de travail « mathématiques »*. 3 pp.

— (nov. 2020). *Plan des études gymnasiales · Domaines des Mathématiques*. Fribourg. 25 pp. URL : https://www.fr.ch/sites/default/files/2021-08/plan-d-etudes-s2mathematiquesgym2021_0.pdf (visité le 27/03/2024).

WEBER Christof (août 2008). "»Umfallen und Wegrutschen ist gleich« — mit mathematischen Vorstellungsubungen in den Dialog gehen". Allemand. In : *Besser lernen im Dialog · Dialogisches Lernen in der Unterrichtspraxis*. Sous la dir. d'Urs RUF, Stefan KELLER et Felix WINTER. Klett | Kallmeyer, p. 142–161. ISBN : 978-3-7800-4913-1.

— A —

Code source

A.1 Manuels d'utilisation (QUICKSTART et README)

A.1.1 Équations du second degré

Code A.1 : QUICKSTART pour la ressource sur les équations de second degré

```
# Guide de démarrage rapide

##  « Je ne gère pas l'informatique » : version en ligne

Si vous voulez rapidement accéder au projet sans trop avoir à mettre les mains à la pâte, vous pouvez accéder à une [version en ligne](https://www.overleaf.com/read/tsmdmkhnrzjq#7569b4). Il faut en créer une copie. Pour cela, le plus simple est de vous créer un compte sur Overleaf, 'douvrir le lien ci-dessus puis de créer une copie du projet en cliquant sur le bouton  «copie»:

![Bouton  «copie» dans Overleaf, mis en évidence avec une grande flèche verte.](../images/copier-d-overleaf-oq.png)

- Assurez-vous, si le document refuse de compiler, que le compilateur soit bien sélectionné comme LuaLaTeX et non pas pdfTeX. Vous accéder à ce paramètre depuis le menu en haut à gauche d'Overleaf.

Vous pouvez aussi télécharger les fichiers et les compiler avec **une version à jour** de LaTeX sur votre ordinateur en choisissant LuaLaTeX :

![Bouton  «téléchargement» dans Overleaf, mis en évidence avec une grande flèche verte.](../images/telecharger-d-overleaf-oq.png)

Pour une présentation plus complète, vous êtes invité·e à lire la README, notamment la [section sur l'utilisation de l'outil](./README.md#utilisation-de-loutil).

***

## Personnalisation de l'outil

Deux aspects de l'outil sont  modifiables: la fiche visible (`.tex`) et le code de génération des exercices (`.lua`). Le premier est particulièrement adapté à la modification par utilisat·eu·rice·s non techniques.

### Modifier les `.tex`

L'outil contient deux documents `.tex `: le fichier principal `main-quadratique.tex` et son préambule `preamble-quadratique.tex`.

#### Fichier principal
```

````markdown

La plupart des modifications sont apportable dans le fichier `main-quadratique.tex`. En l'ouvrant (sur Overleaf, il suffit de cliquer dessus pour l'ouvrir), vous pourrez personnaliser les aspects suivants.

##### Consigne

Cherchez le bloc de texte suivant, et modifiez-y la consigne comme bon vous semble.

```tex
%%%%%%%%%%%%%% CONSIGNE %%%%%%%%%%%%%%%

{\small\em Résolvez ces équations en utilisant à chaque fois la méthode la plus rapide possible. (N'oubliez pas d'écrire l'équation sous la forme standard si elle ne l'est pas déjà.) Une fois toutes les équations résolues, vérifiez vos réponses dans le corrigé. Indiquez pour chaque question si vous avez utilisé la même méthode que le corrigé ou non. Vous n'avez pas droit à la calculatrice.}

```

##### Nombre de questions

Pour modifier le nombre de questions imprimées, cherchez le bloc suivant et remplacez `24` par le nombre qui vous arrange.

```tex
%%%%%%% GÉNÈRE LES QUESTIONS DANS UNE BOÎTE VIRTUELLE %%%%%%

\raggedright
\foreach \n in {1,2,...,24}{\afullroutine{\n}}

```

##### Horodatages

Pour facilement pouvoir retrouver la fiche de réponses pour chaque fiche d'exercices, un horodatage (*timestamp* en anglais) a été fourni. Ceci peut-être utile dans le cas où les exercices sont séparés de leurs solutions en découpant le long des traits tillés.

Si cette fonctionnalité ne vous sert pas, effacez dans les deux sections intitulées `[...] HORODATAGE [...]` les caractères suivants (ni plus ni moins).

```tex
\\{\tiny\sffamily (générés le \DTMtoday\ à \DTMcurrenttime s)}
```

##### Refaire la mise en page

Si vous voulez refaire depuis le début la mise en page, sachez que le document minimal qui fonctionne encore selon le même modèle est le suivant. Tout le reste du `main-quadratique.tex` est cosmétique. C'est à vous de voir ce que vous voulez retirer, ajouter ou retenir.

```tex
\documentclass[a4paper, 11pt]{article}
\input{preamble-quadratique.tex}

\begin{document}
````
````

````

  \foreach \n in {1,2,...,24}{\afullroutine{\n}} % prépare les questions/réponses

  \showallquestions % imprime la liste des questions

  \showallanswers % imprime la liste des réponses

\end{document}

```

##### Mises en page alternatives

Si vous voulez *vraiment* complètement refaire la mise en page, quelques commandes additionnelles vous sont fournies, à utiliser depuis le fichier `main-*.tex`:

- `\aprintqna` : imprime l'équation sur une ligne, suivie de la réponse.

- `\amakequestion` : à utiliser en mode mathématique (*mathmode*, c.-à-d. `$\amakequestion$`), génère une équation à résoudre.

- `\amakeanswer` : à utiliser en mode texte, donne la solution recommandée à la dernière équation générée par `\amakequestion`.

Les deux dernières sont plus modulables, pouvant être utilisées pour placer l'équation et la réponse plus librement, par exemple au sein d'une plus grande feuille de travail avec d'autres types de questions.

##### Autres modifications

Si vous le souhaitez, vous pouvez également modifier quelques réglages dans l'algorithme qui décide des polynômes et des méthodes pour les résoudre.  Ces réglages sont tous dans la section intitulée `(EDITABLE) PREAMBLE` du fichier `.lua`. En particulier, vous avez accès aux fonctionnalités suivantes :

- choisir la graine pour la générations de nombres pseudoaléatoires, en modifiant le paramètre de la fonction `math.randomseed()` - *si vous voulez pouvoir garder vos feuilles d'une fois à une autre, il vous faut fixer ce paramètre **avant** la compilation*;

- choisir la terminologie pour la formule de Viète (de base `formule = [[formule quad.\@]]`, que vous pouvez remplacer par `formule = [[formule de Viète]]` par exemple) ;

- déterminer la probabilité que les solutions d'un exercice donné soient des nombres entiers, en modifiant la fonction booléenne `whether_from_factored_form()`;

- fixer, dans le cas de solutions entières, la proportion avec laquelle un carré parfait sera imposée, avec la fonction booléenne `enforce_perfect_square()`;

- décider quels polynômes sont estimés faciles à factoriser par trinôme, en modifiant la fonction booléenne `easy_factor()` [pour utilisat·eu·rice·s avancé·e·s].

***

## Plus d'informations

````
````

```
D'autres conseils et modes d'utilisation sont donnés dans la
    [README](./README.md), notamment pour le traitement par lots (génération
    automatique de plusieurs fiches d'un coup avec une version locale de
    LuaLaTeX). Il y a aussi une version combinant cet outil avec son jumeau sur
    les systèmes d'équations, disponible [sur
    Overleaf](https://www.overleaf.com/read/wzdcckddkjzy#f3d012). Pour son mode
    d'emploi, consulter la README.
```

Code A.2: README pour la ressource sur les équations de second degré

```
# Démarrage rapide

Pour démarrer rapidement l'utilisation de cet outil sans vous soucier de tous les
    aboutissants de sa création, vous pouvez consulter le fichier
    [`QUICKSTART.md`](./QUICKSTART.md) ci-joint. Il contient également des
    conseils sur la modification facile de l'outil.

Si vous lisez ce document depuis un autre chose que la page GitHub, sachez qu'en
    suivant le lien [pour le dépôt du
    projet](https://github.com/tytyvillus/laboratoire-didactique) vous pouvez lire
    des versions de ce guide d'un abord facile, avec une mise en page élégante.

# Résoudre efficacement une équation de second degré

## Présentation

Le présent outil est conçu pour permettre à des élèves d'entraîner la résolution
    d'équations de deuxième degré $a x^2 + b x + c = 0$ en utilisant la méthode la
    plus appropriée, en reconnaîssant les suivants  cas:

- quand $b = 0$, on peut résoudre par réarrangement $x ^2= - \frac c  a$;

- quand $c = 0$, on peut factoriser $ax$ pour avoir $a x \left( \frac b a x +
    \frac c a \right) = 0 $;

- si c'est un carré parfait, utiliser les identités remarquables $(A \pm B)^2$
    pour  factoriser;

- si les nombres sont petits, factoriser par  trinôme;

- et sinon, utiliser la formule de Viète / formule quadratique.

(Certains cas spéciaux sont aussi  inclus: l'équation de premier degré, ou de type
    $0 = 1$, ou $0 = 0 $; ceux-ci apparaissent avec une faible probabilité.)

Pour ce faire, l'outil permet de produire des fiches d'exercices avec des
    équations du second degré à résoudre le plus efficacement possible par les
    élèves.  Avec les exercices sont générées des solutions recommandées, choisies
    parmi les méthodes énumérées ci-dessus.  L'utilisation recommandée de l'outil
    est après avoir discuter et entraîné les différentes méthodes indépendamment -
    ayant expliqué l'utilité de chercher une méthode efficace - de distribuer une
    feuille par élève, chacune générée séparément, et de les laisser travailler à
    les résoudre.  Une fois une proportion suffisante des exercices résolus,
    l'enseignant·e peut procéder à une mise en commun lors de laquelle chaque
    élève - par exemple - expose l'équation dont la méthode de résolution a été
    perçue comme la moins évidente.

Ce projet entre dans le contexte d'entraîner la compétence de base  «utiliser avec
    souplesse l'outillage  mathématique» donnée dans l'*Annexe au plan d'études
    cadre du 9 juin 1994 pour les écoles de maturité : Compétences de base en
    mathématiques et en langue première constitutives de l'aptitude générale aux
    études supérieures, du 17 mars 2016*.
```

````markdown

## Structure de l'outil

L'outil se présente comme cinq fichiers dans une archive `.zip `: le QUICKSTART, la README, un fichier `codeQuadratique.lua`, un fichier `main-quadratique.tex` et un fichier `preamble-quadratique.tex`.

- Le `*.lua` contient la majorité de l'algorithme servant à produire les questions.

- Le `preamble-*.tex` enclenche un petit nombre de paquets, librement distribués dans TEXLive, MikTEX ou sur www.ctan.org.  Ceux-ci comprennent `luacode` (automatique avec LuaLaTeX) et `luacas`, utilisé pour simplifier algébriquement certaines des équations-réponses. Dans le préambule, plusieurs commandes sont définies qui se réfèrent au `*.lua`.

- Le `main-*.tex` enclenche le préambule puis se charge de la mise en page du document.

## Utilisation de l'outil

### Démarrage rapide

L'outil est disposé afin que, après avoir décompressé l'archive `.zip`, il suffise d'ouvrir le fichier `main-*.pdf` dans votre éditeur TeX favori et de le compiler pour obtenir une fiche complète et utilisable.  C'est un outil «prêt à l' emploi» ( «*plug and play *»).

Il vous faut seulement vous assurer que vous ayez une distribution LaTeX à jour, et **compiler le document avec le moteur LuaLaTeX** (et non pdfLaTeX ou XeLaTeX).

#### Version Overleaf

Une version en ligne est aussi mise à disposition si vous ne voulez pas devoir installer LaTeX localement. Le lien est [celui-ci](https://www.overleaf.com/read/tsmdmkhnrzjq#7569b4). Pour pouvoir modifier le document, il faudra vous créer une copie personnelle: le plus simple est de vous créer un compte sur Overleaf, ouvrir le lien donné puis de créer une copie du projet en cliquant sur le bouton  «copie»:

![Bouton  «copie» dans Overleaf, mis en évidence avec une grande flèche verte.](../images/copier-d-overleaf-oq.png)

Assurez-vous, si le document refuse de compiler, que le compilateur soit bien sélectionné comme LuaLaTeX et non pas pdfTeX. Vous accéder à ce paramètre depuis le menu en haut à gauche d'Overleaf.

### Modifier le  document: mains à la pâte

#### Depuis le `.tex`

Le document `main-*.tex` consiste en une série de 24 exercices, mis en page algorithmiquement avec les commandes
```tex
\foreach \n in {1,2,...,24}{\afullroutine{\n}}
```
et `\showallquestions` et `\showallanswers`.

Quelques commandes additionnelles sont également proposées directement dans le `main-*.tex` pour une mise en page alternative :
````

````
- `\aprintqna` : imprime l'équation sur une ligne, suivie de la réponse.

- `\amakequestion` : à utiliser en mode mathématique (*mathmode*), génère une équation à résoudre.

- `\amakeanswer` : à utiliser en mode texte, donne la solution recommandée à la dernière équation générée par `\amakequestion`.

Les deux dernières sont plus modulables, pouvant être utilisées pour placer l'équation et la réponse plus librement, par exemple au sein d'une plus grande feuille de travail avec d'autres types de questions.

#### Depuis le code `.lua`

Si vous le souhaitez, vous pouvez modifier quelques réglages dans l'algorithme qui décide des polynômes et des méthodes pour les résoudre.  Ces réglages sont tous dans la section intitulée `(EDITABLE) PREAMBLE` du fichier `.lua`. En particulier, vous avez accès aux fonctionnalités suivantes :

- choisir la graine pour la générations de nombres pseudoaléatoires, en modifiant le paramètre de la fonction `math.randomseed()` - *si vous voulez pouvoir garder vos feuilles d'une fois à une autre, il vous faut fixer ce paramètre **avant** la compilation*;

- choisir la terminologie pour la formule de Viète (de base `formule = [[formule quadr.\@]]`, que vous pouvez remplacer par `formule = [[formule de Viète]]` par exemple) ;

- déterminer la probabilité que les solutions d'un exercice donné soient des nombres entiers, en modifiant la fonction booléenne `whether_from_factored_form()`;

- fixer, dans le cas de solutions entières, la proportion avec laquelle un carré parfait sera imposée, avec la fonction booléenne `enforce_perfect_square()`;

- décider quels polynômes sont estimés faciles à factoriser par trinôme, en modifiant la fonction booléenne `easy_factor()` [pour utilisat·eu·rice·s avancé·e·s].

#### Accès à des fonctions additionnelles

Pour les utilisat·eu·rice·s les plus téméraires, quelques dernières fonctions sont également fournies à LaTeX depuis le code Lua. C'est la dernière section du code `mainQuadratique.tex`, qui ressemble à ceci:

```lua
-- INTERFACE WITH LUALATEX ENGINE --

-- Export user-accessible functions (renamed using syntax `new = old`):
return {
    polynomial = generate_polynomial, -- returns {a, b, c, num_sols, x1, x2, rat}
    methodString = pick_method, -- provides recommended method
    printEquation = cas_equation, -- question preprinted for LaTeX
    answer = answer_line, -- answer preprinted for LaTeX
    fullRoutine = full_routine, -- whole shebang
    printQnA = print_questions_and_answers, -- alternative, single-equation formatting style
}
```

````

````markdown
Pour accéder à ces fonctions, il faut les appeler depuis un document `.tex`. C'est ce qui est déjà fait dans `preamble-*.tex` pour certaines de celles-ci, avec les commandes suivantes.

```tex
% Activer les fonctions dans le code / Load lua code:
\directlua{codeA = require "codeQuadratique"}

% Convertir fonctions de code.lua en commandes LaTeX / Extract useful functions as LaTeX commands:
\newcommand{\afullroutine}[1]{\directlua{codeA.fullRoutine(#1)}} % format par défaut / default style

%[...]
```

La syntaxe de ces fonctions est documentée dans le fichier `mainQuadratique.lua` directement.

### Traitement par lots

Pour produire plusieurs feuilles différentes d'un coup, afin de pouvoir par exemple les distribuer individuellement à une classe de 20 élèves, nous n'avons malheureusement pas trouvé d'autre solution que d'appeler plusieurs fois le moteur LuaLaTeX depuis un programme externe. Par exemple, sur Linux, vous pouvez taper le code suivant dans Bash depuis le dossier contenant `main-*.tex` :

```bash
for i in {1..20}; do lualatex -jobname feuille-$i main-quadratique.tex; done
```
Il faudra cependant adapter ces commandes à la syntaxe particulière de votre système opératoire. Un autre exemple, pour Windows : ouvrez le dossier contenant `main-*.tex`, effectuez un clic-droit et sélectionner *Ouvrir dans le Terminal*. (Vérifiez que celui-ci soit bien la Windows PowerShell.) Dans le terminal, vous pouvez ensuite taper :
```powershell
for ($var = 1; $var -le 20; $var++) {lualatex.exe -jobname feuille-$var main-quadratique.tex}
```
Il vous faudra donc vous familiariser avec la variante qui vous conviendra. *N.b.* - il vous faudra aussi adapter le nom de votre moteur LuaLaTeX (`lualatex`, `lualatex.exe` ou autre).

### Intégration avec l'autre outil du même projet

Dans le cadre de ce laboratoire didactique, deux outils ont été créés: l'un sur les systèmes d'équations et l'autre sur les équations du second degré. Si vous voulez pouvoir combiner les deux types de questions dans un seul et même document, créez un dossier contenant les fichiers `codeQuadratique.lua` et `codeSysteme.lua`. Ensuite, créez un fichier `preamble.tex` et copiez-y le code suivant (n'oubliez pas de sauvegarder).

```tex
%!TeX root = main.tex

% --- PRÉAMBULE ---

% langue et police:
\usepackage[quiet]{fontspec}
\usepackage{polyglossia} \setmainlanguage[variant=swiss]{french}
````
````

```
% calculs internes:
\usepackage{luacas}

% maths:
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage[warnings-off={mathtools-colon,mathtools-overbracket},
    math-style=ISO,]{unicode-math}
\usepackage[locale = FR, round-precision = 3, round-mode = figures, round-pad =
    false,]{siunitx}
% vvvvvvvvvvvvvvvvvvvvv
% \newcommand{\num}[1]{#1}
%%  ^ à décommenter si vous voulez vous débarasser de <>siunitx
%%  ^ uncomment if you want to get rid of <>siunitx

% choix esthétiques, facultatifs:
\let\oldemptyset\emptyset
\let\emptyset\varnothing

% permettent mise en page:
\usepackage{multicol}
\usepackage{pgffor}

% date et heure
\usepackage[timesep=., showzone=false, hourminsep=h, minsecsep=m,]{datetime2} %
    permet les horodatages


% ---- ANSWER-PRINTING STYLE FROM https://tex.stackexchange.com/a/15354 ----

% Define answer environment

\newbox\allanswers
\setbox\allanswers=\vbox{}

\newenvironment{customanswer}
{\global\setbox\allanswers=\vbox\bgroup
  \unvbox\allanswers
  \vspace{-4pt}
}
{\bigbreak\egroup}

\newcommand{\showallanswers}{\par\unvbox\allanswers}

% Define question environment

\newbox\allquestions
\setbox\allquestions=\vbox{}

\newenvironment{customquestion}
{\global\setbox\allquestions=\vbox\bgroup
  \unvbox\allquestions
}
{\bigbreak\egroup}

\newcommand{\showallquestions}{\par\unvbox\allquestions}


%% ---------- PROVIDE \timestamp COMMAND -------------
% -- from https://flaterco.com/util/timestamp.sty --
```

```
\makeatletter

\newcount\@DT@modctr
\newcount\@dtctr

\def\@modulo#1#2{%
  \@DT@modctr=#1\relax
  \divide \@DT@modctr by #2\relax
  \multiply \@DT@modctr by #2\relax
  \advance #1 by -\@DT@modctr}

\newcommand{\xxivtime}{%
  \@dtctr=\time%
  \divide\@dtctr by 60
  \ifnum\@dtctr<10 0\fi\the\@dtctr.%
  \@dtctr=\time%
  \@modulo{\@dtctr}{60}%
  \ifnum\@dtctr<10 0\fi\the\@dtctr%
}

\newcommand{\timestamp}{\the\year-%
  \ifnum\month<10 0\fi\the\month-%
  \ifnum\day<10 0\fi\the\day\ \xxivtime}

\makeatother


% ---------------- ACCÈS AU CODE LUA: -----------------

% Aide-mémoire Lua: https://devhints.io/lua

%% OUTIL SECOND DEGRÉ

% Activer les fonctions dans le code:
% Load lua code:
\directlua{codeA = require "codeQuadratique"}

% Pour le format par défaut:
\newcommand{\afullroutine}[1]{\directlua{codeA.fullRoutine(#1)}}

% Pour imprimer question immédiatement suivie de réponse:
\newcommand{\aprintqna}{\directlua{codeA.printQnA()}}

% Pour librement produire question, et séparément la réponse
\newcommand{\amakequestion}{%
  % MUST BE INSERTED IN MATHMODE: $\amakequestion$
    \directlua{%
      polynomialcoeffs = codeA.polynomial()
      eqn = codeA.printEquation(table.unpack(polynomialcoeffs,1,3))
      tex.print(eqn)
    }
  }
\newcommand{\amakeanswer}{%
  % IN TEXTMODE
    \directlua{%
      tex.print(codeA.answer(table.unpack(polynomialcoeffs)))
    }
  }

%% OUTIL SYSTÈME D'ÉQUATIONS
```

````
% Activer les fonctions dans le code:
% Load lua code:
\directlua{codeB = require "codeSysteme"}

% Pour le format par défaut:
\newcommand{\bfullroutine}[1]{\directlua{codeB.fullRoutine(#1)}}

% Pour imprimer question immédiatement suivie de réponse:
\newcommand{\bprintqna}{\directlua{codeB.printQnA()}}

% Pour librement produire question, et séparément la réponse
\newcommand{\bmakequestion}{%
  % MUST BE INSERTED IN MATHMODE: $\bmakequestion$
    \directlua{%
      coeffs, num_sols, x, y = codeB.polynomial(0.2)
      eqn = codeB.printEquation(coeffs)
      tex.print(eqn)
    }
  }
\newcommand{\bmakeanswer}{%
  % IN TEXTMODE
    \directlua{%
      tex.print(codeB.answer(coeffs, num_sols, x, y))
    }
  }

```

Une fois que c'est fait, vous pouvez créer un fichier `main.tex` (toujours dans le même dossier) et le remplir comme vous convient, tant que vous n'oubliez pas d'y inclure le préambule avec la commande `\input{preamble.tex}`. Une idée de base depuis laquelle travailler est la suivante.

```tex
\documentclass[a4paper, 11pt]{article}
\usepackage[margin=2cm]{geometry}
\input{preamble.tex}

\begin{document} \thispagestyle{empty}

  \foreach \n in {1,2,...,12}{\afullroutine{\n}} % prépare les questions/réponses
  \begin{multicols}{3} \showallquestions \end{multicols} % imprime la liste des questions
  ~\medskip \showallanswers   % espace vertical, puis imprime la liste des réponses

  \medskip %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  \hbox to \linewidth{\leaders\hbox to 4pt{\hss · \hss}\hfil} % séparation entre les parties

  \setbox\allanswers=\vbox{}  % vider la boîte des réponses
  \setbox\allquestions=\vbox{}  % vider la boîte des questions

  \medskip %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  \foreach \n in {1,2,...,9}{\bfullroutine{\n}} % prépare les questions/réponses
  \begin{multicols}{3} \showallquestions \end{multicols} % imprime la liste des questions
````

````
  ~\medskip \showallanswers   % espace vertical, puis imprime la liste des réponses

\end{document}
```

Un exemple de fiche créée suivant cette procédure est disponible [sur Overleaf](https://www.overleaf.com/read/wzdcckddkjzy#f3d012).

***

## Licence et attributions

Cet outil est distribué avec la licence [GPL-3.0-or-later](https://www.gnu.org/licenses/gpl-3.0.html) par ses auteurs :

- Alexandros Rispo Constantinou

Une [page GitHub pour cet outil](https://github.com/tytyvillus/laboratoire-didactique) est également disponible.
````

A.1.2 Systèmes de deux équations

Code A.3 : QUICKSTART pour la ressource sur les systèmes d'équations

````
# Guide de démarrage rapide

##  « Je ne gère pas l'informatique » : version en ligne

Si vous voulez rapidement accéder au projet sans trop avoir à mettre les mains à la pâte, vous pouvez accéder à une [version en ligne](https://www.overleaf.com/read/hmxjgqgkkmcq#9ccf30). Il faut en créer une copie. Pour cela, le plus simple est de vous créer un compte sur Overleaf, 'douvrir le lien ci-dessus puis de créer une copie du projet en cliquant sur le bouton  «copie»:

![Bouton  «copie» dans Overleaf, mis en évidence avec une grande flèche verte.](../images/copier-d-overleaf-oq.png)

- Assurez-vous, si le document refuse de compiler, que le compilateur soit bien sélectionné comme LuaLaTeX et non pas pdfTeX. Vous accéder à ce paramètre depuis le menu en haut à gauche d'Overleaf.

Vous pouvez aussi télécharger les fichiers et les compiler avec **une version à jour** de LaTeX sur votre ordinateur en choisissant LuaLaTeX :

![Bouton  «téléchargement» dans Overleaf, mis en évidence avec une grande flèche verte.](../images/telecharger-d-overleaf-oq.png)

Pour une présentation plus complète, vous êtes invité·e à lire la README, notamment la [section sur l'utilisation de l'outil](./README.md#utilisation-de-loutil).

***

## Personnalisation de l'outil

Deux aspects de l'outil sont  modifiables: la fiche visible (`.tex`) et le code de génération des exercices (`.lua`). Le premier est particulièrement adapté à la modification par utilisat·eu·rice·s non techniques.
````
````

````markdown

### Modifier les `.tex`

L'outil contient deux documents `.tex `: le fichier principal `main-systeme.tex` et son préambule `preamble-systeme.tex`.

#### Fichier principal

La plupart des modifications sont apportable dans le fichier  `main-systeme.tex`. En l'ouvrant (sur Overleaf, il suffit de cliquer dessus pour l'ouvrir), vous pourrez personnaliser les aspects suivants.

##### Consigne

Cherchez le bloc de texte suivant, et modifiez-y la consigne comme bon vous semble.

```tex
%%%%%%%%%%%%%% CONSIGNE %%%%%%%%%%%%%%%

{\small\em Résolvez ces équations en utilisant à chaque fois la méthode la plus rapide possible. (N'oubliez pas d'écrire l'équation sous la forme standard si elle ne l'est pas déjà.) Une fois toutes les équations résolues, vérifiez vos réponses dans le corrigé. Indiquez pour chaque question si vous avez utilisé la même méthode que le corrigé ou non. Vous n'avez pas droit à la calculatrice.}

```

##### Nombre de questions

Pour modifier le nombre de questions imprimées, cherchez le bloc suivant et remplacez `18` par le nombre qui vous arrange.

```tex
%%%%%%% GÉNÈRE LES QUESTIONS DANS UNE BOÎTE VIRTUELLE %%%%%%

\raggedright
\foreach \n in {1, 2, ..., 18}{\bfullroutine{\n}}

```

##### Horodatages

Pour facilement pouvoir retrouver la fiche de réponses pour chaque fiche d'exercices, un horodatage (*timestamp* en anglais) a été fourni. Ceci peut-être utile dans le cas où les exercices sont séparés de leurs solutions en découpant le long des traits tillés.

Si cette fonctionnalité ne vous sert pas, effacez dans les deux sections intitulées `[...] HORODATAGE [...]` les caractères suivants (ni plus ni moins).

```tex
\\{\tiny\sffamily (générés le \DTMtoday\ à \DTMcurrenttime s)}
```

##### Refaire la mise en page

Si vous voulez refaire depuis le début la mise en page, sachez que le document minimal qui fonctionne encore selon le même modèle est le suivant. Tout le
````

````
reste du `main-systeme.tex` est cosmétique. C'est à vous de voir ce que vous voulez retirer, ajouter ou retenir.

```tex
\documentclass[a4paper, 11pt]{article}
\input{preamble-systeme.tex}

\begin{document}

  \foreach \n in {1, 2, ..., 18}{\bfullroutine{\n}} % prépare les questions/réponses

  \showallquestions % imprime la liste des questions

  \showallanswers % imprime la liste des réponses

\end{document}
```

##### Mises en page alternatives

Si vous voulez *vraiment* complètement refaire la mise en page, quelques commandes additionnelles vous sont fournies, à utiliser depuis le fichier `main-*.tex`:

- `\bprintqna` : imprime l'équation sur une ligne, suivie de la réponse.

- `\bmakequestion` : à utiliser en mode mathématique (*mathmode*, c.-à-d. `$\bmakequestion$`), génère une équation à résoudre.

- `\bmakeanswer` : à utiliser en mode texte, donne la solution recommandée à la dernière équation générée par `\bmakequestion`.

Les deux dernières sont plus modulables, pouvant être utilisées pour placer l'équation et la réponse plus librement, par exemple au sein d'une plus grande feuille de travail avec d'autres types de questions.

##### Autres modifications

Si vous le souhaitez, vous pouvez modifier quelques réglages dans l'algorithme qui décide des polynômes et des méthodes pour les résoudre.  Ces réglages sont tous dans la section intitulée `(EDITABLE) PREAMBLE` du fichier `.lua`. En particulier, vous avez accès aux fonctionnalités suivantes :

- choisir la graine pour la générations de nombres pseudoaléatoires, en modifiant le paramètre de la fonction `math.randomseed()` - *si vous voulez pouvoir garder vos feuilles d'une fois à une autre, il vous faut fixer ce paramètre **avant** la compilation*;

- déterminer la probabilité que le système généré soit singulier, en modifiant la variable `probability_singular`.

***

## Plus d'informations

D'autres conseils et modes d'utilisation sont donnés dans la [README](./README.md), notamment pour le traitement par lots (génération automatique de plusieurs fiches d'un coup avec une version locale de LuaLaTeX). Il y a aussi une version combinant cet outil avec son jumeau sur les équations du second degré, disponible [sur
````

```
    Overleaf](https://www.overleaf.com/read/wzdcckddkjzy#f3d012). Pour son mode
    d'emploi, consulter la README.
```

Code A.4: README pour la ressource sur les systèmes d'équations

```
# Démarrage rapide

Pour démarrer rapidement l'utilisation de cet outil sans vous soucier de tous les
    aboutissants de sa création, vous pouvez consulter le fichier
    [`QUICKSTART.md`](./QUICKSTART.md) ci-joint. Il contient également des
    conseils sur la modification facile de l'outil.

Si vous lisez ce document depuis un autre chose que la page GitHub, sachez qu'en
    suivant le lien [pour le dépôt du
    projet](https://github.com/tytyvillus/laboratoire-didactique) vous pouvez lire
    des versions de ce guide d'un abord facile et de mise en page élégante.

# Résoudre efficacement un système de deux équations linéaires

## Présentation

Le présent outil est conçu pour permettre à des élèves d'entraîner la résolution
    de systèmes d'équations linéaires de la forme
```math
\left\{\begin{array}{c} ax + by + c = 0 \\ dx + ey + f = 0 \end{array} \right.
```
en utilisant la méthode la plus appropriée, en reconnaîssant les cas suivants:

- quand $a \in \lbrace 0, 1\rbrace$ ou $b   \in \lbrace 0, 1\rbrace$, on peut
    isoler directement l'une des deux inconnues dans la première équation ;

- quand $d = 0$ ou $e=0$, on peut isoler directement l'une des deux inconnues
    dans la deuxième équation;

- si l'une des équations est un multiple de l'autre, le nombre de solutions est
    infini (droites confondues) ;

- si le déterminant de la matrice des coefficients est nul et que les deux
    équations ne sont pas multiples l'une de l'autre, alors les droites sont
    parallèles;

- si $a \equiv 0 \pmod b$ ou inversément, on divise toute l' équationpar b,
    respectivement a et on retombe dans le premier cas abordé ci-dessus; idem si
    $d \equiv 0 \pmod e$ ou inversément;

- et sinon, faire de la combinaison linéaire (via règle de Cramer si elle est
    connue).

Pour ce faire, l'outil permet de produire des fiches d'exercices avec des
    systèmes de deux équations linéaires à résoudre le plus efficacement possible
    par les élèves. Avec les exercices sont générées des solutions recommandées,
    choisies parmi les méthodes énumérées ci-dessus. L'utilisation recommandée de
    l'outil est à faire après avoir discuté et entraîné les différentes méthodes
    indépendamment, en ayant expliqué l'utilité de chercher une méthode efficace.
    Ensuite, il conviendra de distribuer une feuille par élève, chacune générée
    séparément, et de les laisser travailler à les résoudre. Une fois une
    proportion suffisante des exercices résolus, l'enseignant·e peut procéder à
    une mise en commun lors de laquelle chaque élève - par exemple - expose
    l'équation dont la méthode de résolution a été perçue comme la moins évidente.

```

````
Ce projet entre dans le contexte d'entraîner la compétence de base «utiliser avec souplesse l'outillage mathématique» donnée dans l'*Annexe au plan d'études cadre du 9 juin 1994 pour les écoles de maturité : Compétences de base en mathématiques et en langue première constitutives de l'aptitude générale aux études supérieures, du 17 mars 2016*.

Structure de l'outil

L'outil se présente comme cinq fichiers dans une archive `.zip `: le QUICKSTART, la README, un fichier `codeSysteme.lua`, un fichier `main-systeme.tex` et un fichier `preamble-systeme.tex`.

- Le `*.lua` contient la majorité de l'algorithme servant à produire les questions.

- Le `preamble-*.tex` enclenche un petit nombre de paquets, librement distribués dans TEXLive, MikTEX ou sur www.ctan.org. Ceux-ci comprennent `luacode` (automatique avec LuaLaTeX) et `luacas`, utilisé pour simplifier algébriquement certaines des équations-réponses. Dans le préambule, plusieurs commandes sont définies qui se réfèrent au `*.lua`.

- Le `main-*.tex` enclenche le préambule puis se charge de la mise en page du document.

Utilisation de l'outil

L'outil est conçue pour que, après avoir décompressé l'archive `.zip`, il suffise d'ouvrir le fichier `main-*.pdf` dans votre éditeur TeX favori et de le compiler pour obtenir une fiche complète et utilisable. C'est un outil «prêt à l' emploi» («*plug and play*»).

Il vous faut seulement vous assurer que vous ayez une distribution LaTeX à jour, et **compiler le document avec le moteur LuaLaTeX** (et non pdfLaTeX ou XeLaTeX).

***

Modifier le document: mains à la pâte

Depuis le `.tex`

Le document `main-*.tex` consiste en une série de 18 exercices, mis en page algorithmiquement avec les commandes
```tex
\foreach \n in {1,2,...,18}{\bfullroutine{\n}}
```
et `\showallquestions` et `\showallanswers`.

Quelques commandes additionnelles sont également proposées directement dans le `main-*.tex` pour une mise en page alternative :

- `\bprintqna` : imprime l'équation sur une ligne, suivie de la réponse.

- `\bmakequestion` : à utiliser en mode mathématique (*mathmode*), génère une équation à résoudre.

- `\bmakeanswer` : à utiliser en mode texte, donne la solution recommandée à la dernière équation générée par `\bmakequestion`.

````

````markdown
Les deux dernières sont plus modulables, pouvant être utilisées pour placer l'équation et la réponse plus librement, par exemple au sein d'une plus grande feuille de travail avec d'autres types de questions.

#### Depuis le code `.lua`

Si vous le souhaitez, vous pouvez modifier quelques réglages dans l'algorithme qui décide des polynômes et des méthodes pour les résoudre.  Ces réglages sont tous dans la section intitulée `(EDITABLE) PREAMBLE` du fichier `.lua`. En particulier, vous avez accès aux fonctionnalités suivantes :

- choisir la graine pour la générations de nombres pseudoaléatoires, en modifiant le paramètre de la fonction `math.randomseed()` - *si vous voulez pouvoir garder vos feuilles d'une fois à une autre, il vous faut fixer ce paramètre **avant** la compilation*;
- choisir la probabilité d'avoir une matrice de coefficients qui soit singulière, en modifiant le paramètre `probability_singular`.

#### Accès à des fonctions additionnelles

Pour les utilisat·eu·rice·s les plus téméraires, quelques dernières fonctions sont également fournies à LaTeX depuis le code Lua. C'est la dernière section du code `mainSysteme.tex`, qui ressemble à ceci:

```lua
-- INTERFACE WITH LUALATEX ENGINE --

-- Export user-accessible functions (renamed using syntax `new = old`):
return {
    polynomial = generate_exercise, -- returns {coefficients, num_sols, x, y}
    methodString = pick_method_case, -- provides recommended method
    printEquation = cas_equation, -- question preprinted for LaTeX
    answer = answer_line, -- answer preprinted for LaTeX
    fullRoutine = full_routine, -- whole shebang
    printQnA = print_questions_and_answers, -- alternative, single-equation formatting style
}
```

Pour accéder à ces fonctions, il faut les appeler depuis un document `.tex`. C'est ce qui est déjà fait dans `preamble-*.tex` pour certaines de celles-ci, avec les commandes suivantes.

```tex
% Activer les fonctions dans le code / Load lua code:
\directlua{codeB = require "codeSysteme"}

% Convertir fonctions de code.lua en commandes LaTeX / Extract useful functions as LaTeX commands:
\newcommand{\bfullroutine}[1]{\directlua{codeB.fullRoutine(#1)}} % format par défaut / default style

%[...]
```

La syntaxe de ces fonctions est documentée dans le fichier `mainSysteme.lua` directement.

### Traitement par lots

````

````markdown
Pour produire plusieurs feuilles différentes d'un coup, afin de pouvoir par exemple les distribuer individuellement à une classe de 20 élèves, nous n'avons malheureusement pas trouvé d'autre solution que d'appeler plusieurs fois le moteur LuaLaTeX depuis un programme externe. Par exemple, sur Linux, vous pouvez taper le code suivant dans Bash depuis le dossier contenant `main-*.tex` :

```bash
for i in {1..20}; do lualatex -jobname feuille-$i main-systeme.tex; done
```
Il faudra cependant adapter ces commandes à la syntaxe particulière de votre système opératoire. Un autre exemple, pour Windows : ouvrez le dossier contenant `main-*.tex`, effectuez un clic-droit et sélectionner *Ouvrir dans le Terminal*. (Vérifiez que celui-ci soit bien la Windows PowerShell.) Dans le terminal, vous pouvez ensuite taper :
```powershell
for ($var = 1; $var -le 20; $var++) {lualatex.exe -jobname feuille-$var main-systeme.tex}
```
Il vous faudra donc vous familiariser avec la variante qui vous conviendra. *N.b.* - il vous faudra aussi adapter le nom de votre moteur LuaLaTeX (`lualatex`, `lualatex.exe` ou autre).

### Intégration avec l'autre outil du même projet

Dans le cadre de ce laboratoire didactique, deux outils ont été créés: l'un sur les systèmes d'équations et l'autre sur les équations du second degré. Si vous voulez pouvoir combiner les deux types de questions dans un seul et même document, créez un dossier contenant les fichiers `codeQuadratique.lua` et `codeSysteme.lua`. Ensuite, créez un fichier `preamble.tex` et copiez-y le code suivant (n'oubliez pas de sauvegarder).

```tex
%!TeX root = main.tex

% --- PRÉAMBULE ---

% langue et police:
\usepackage[quiet]{fontspec}
\usepackage{polyglossia} \setmainlanguage[variant=swiss]{french}

% calculs internes:
\usepackage{luacas}

% maths:
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage[warnings-off={mathtools-colon,mathtools-overbracket}, math-style=ISO,]{unicode-math}
\usepackage[locale = FR, round-precision = 3, round-mode = figures, round-pad = false,]{siunitx}
% vvvvvvvvvvvvvvvvvvvv
% \newcommand{\num}[1]{#1}
%%  ^ à décommenter si vous voulez vous débarasser de <>siunitx
%%  ^ uncomment if you want to get rid of <>siunitx

% choix esthétiques, facultatifs:
\let\oldemptyset\emptyset
\let\emptyset\varnothing

% permettent mise en page:
````
````

```
\usepackage{multicol}
\usepackage{pgffor}

% date et heure
\usepackage[timesep=., showzone=false, hourminsep=h, minsecsep=m,]{datetime2} % permet les horodatages


% ---- ANSWER-PRINTING STYLE FROM https://tex.stackexchange.com/a/15354 ----

% Define answer environment

\newbox\allanswers
\setbox\allanswers=\vbox{}

\newenvironment{customanswer}
{\global\setbox\allanswers=\vbox\bgroup
  \unvbox\allanswers
  \vspace{-4pt}
}
{\bigbreak\egroup}

\newcommand{\showallanswers}{\par\unvbox\allanswers}

% Define question environment

\newbox\allquestions
\setbox\allquestions=\vbox{}

\newenvironment{customquestion}
{\global\setbox\allquestions=\vbox\bgroup
  \unvbox\allquestions
}
{\bigbreak\egroup}

\newcommand{\showallquestions}{\par\unvbox\allquestions}


%% ---------- PROVIDE \timestamp COMMAND -------------
% -- from https://flaterco.com/util/timestamp.sty --

\makeatletter

\newcount\@DT@modctr
\newcount\@dtctr

\def\@modulo#1#2{%
  \@DT@modctr=#1\relax
  \divide \@DT@modctr by #2\relax
  \multiply \@DT@modctr by #2\relax
  \advance #1 by -\@DT@modctr}

\newcommand{\xxivtime}{%
  \@dtctr=\time%
  \divide\@dtctr by 60
  \ifnum\@dtctr<10 0\fi\the\@dtctr.%
  \@dtctr=\time%
  \@modulo{\@dtctr}{60}%
  \ifnum\@dtctr<10 0\fi\the\@dtctr%
}

```

```
\newcommand{\timestamp}{\the\year-%
  \ifnum\month<10 0\fi\the\month-%
  \ifnum\day<10 0\fi\the\day\ \xxivtime}

\makeatother


% ---------------- ACCÈS AU CODE LUA: -----------------

% Aide-mémoire Lua: https://devhints.io/lua

%% OUTIL SECOND DEGRÉ

% Activer les fonctions dans le code:
% Load lua code:
\directlua{codeA = require "codeQuadratique"}

% Pour le format par défaut:
\newcommand{\afullroutine}[1]{\directlua{codeA.fullRoutine(#1)}}

% Pour imprimer question immédiatement suivie de réponse:
\newcommand{\aprintqna}{\directlua{codeA.printQnA()}}

% Pour librement produire question, et séparément la réponse
\newcommand{\amakequestion}{%
  % MUST BE INSERTED IN MATHMODE: \(\amakequestion\)
    \directlua{%
      polynomialcoeffs = codeA.polynomial()
      eqn = codeA.printEquation(table.unpack(polynomialcoeffs,1,3))
      tex.print(eqn)
    }
  }
\newcommand{\amakeanswer}{%
  % IN TEXTMODE
    \directlua{%
      tex.print(codeA.answer(table.unpack(polynomialcoeffs)))
    }
  }

%% OUTIL SYSTÈME D'ÉQUATIONS

% Activer les fonctions dans le code:
% Load lua code:
\directlua{codeB = require "codeSysteme"}

% Pour le format par défaut:
\newcommand{\bfullroutine}[1]{\directlua{codeB.fullRoutine(#1)}}

% Pour imprimer question immédiatement suivie de réponse:
\newcommand{\bprintqna}{\directlua{codeB.printQnA()}}

% Pour librement produire question, et séparément la réponse
\newcommand{\bmakequestion}{%
  % MUST BE INSERTED IN MATHMODE: \(\bmakequestion\)
    \directlua{%
      coeffs, num_sols, x, y = codeB.polynomial(0.2)
      eqn = codeB.printEquation(coeffs)
      tex.print(eqn)
    }
  }
\newcommand{\bmakeanswer}{%
```

````
  % IN TEXTMODE
    \directlua{%
      tex.print(codeB.answer(coeffs, num_sols, x, y))
    }
  }

```

Une fois que c'est fait, vous pouvez créer un fichier `main.tex` (toujours dans le même dossier) et le remplir comme vous convient, tant que vous n'oubliez pas d'y inclure le préambule avec la commande `\input{preamble.tex}`. Une idée de base depuis laquelle travailler est la suivante.

```tex
\documentclass[a4paper, 11pt]{article}
\usepackage[margin=2cm]{geometry}
\input{preamble.tex}

\begin{document} \thispagestyle{empty}

  \foreach \n in {1,2,...,12}{\afullroutine{\n}} % prépare les questions/réponses
  \begin{multicols}{3} \showallquestions \end{multicols} % imprime la liste des questions
  ~\medskip \showallanswers   % espace vertical, puis imprime la liste des réponses

  \medskip %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  \hbox to \linewidth{\leaders\hbox to 4pt{\hss · \hss}\hfil} % séparation entre les parties

  \setbox\allanswers=\vbox{}  % vider la boîte des réponses
  \setbox\allquestions=\vbox{}  % vider la boîte des questions

  \medskip %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  \foreach \n in {1,2,...,9}{\bfullroutine{\n}} % prépare les questions/réponses
  \begin{multicols}{3} \showallquestions \end{multicols} % imprime la liste des questions
  ~\medskip \showallanswers   % espace vertical, puis imprime la liste des réponses

\end{document}
```

Un exemple de fiche créée suivant cette procédure est disponible [sur Overleaf](https://www.overleaf.com/read/wzdcckddkjzy#f3d012).

***

## Licence et attributions

Cet outil est distribué avec la licence [GPL-3.0-or-later](https://www.gnu.org/licenses/gpl-3.0.html) par ses auteurs :

- Alexandros Rispo Constantinou
- Mathias Blaise

Une [page GitHub pour cet outil](https://github.com/tytyvillus/laboratoire-didactique) est également disponible.
````
````

A.2 Documents LuaLaTeX

A.2.1 Équations du second degré

Code A.5 : Préambule LuaLaTeX pour les exercices de résolution d'équations de second degré

```
%!TeX root = main-quadratique.tex


% --- PRÉAMBULE  ---

% langue et police:
\usepackage[quiet]{fontspec}
\usepackage{polyglossia} \setmainlanguage[variant=swiss]{french}

% calculs internes:
\usepackage{luacas}
%\usepackage[pl,import]{penlight} % désuet

% maths:
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage[ % choix esthétique, facultatif
  warnings-off={mathtools-colon,mathtools-overbracket},
  math-style=ISO,
  ]{unicode-math}
\usepackage[ % à régler selon préférences / change as preferred
  % utilisé ici pour formatter les nombres à virgule avec \num{}
  % used here to format decimal numbers via \num{}
  locale = FR,
  round-precision = 3,
  round-mode = figures,
  round-pad = false,
  %zero-decimal-as-symbol = true, % choix personnel / personal choice
  ]{siunitx}
  % vvvvvvvvvvvvvvvvvvvvv
% \newcommand{\num}[1]{#1}
%%  ^ à décommenter si vous voulez vous débarasser de <>siunitx
%%  ^ uncomment if you want to get rid of <>siunitx

% choix esthétiques, facultatifs:
\let\oldemptyset\emptyset
\let\emptyset\varnothing

% permettent mise en page:
\usepackage{multicol}
\usepackage{pgffor}

% date et heure
\usepackage[ % permet les horodatages (timestamps)
  timesep=.,
  showzone=false,
  hourminsep=h,
  minsecsep=m
  ]{datetime2}


% ---- ANSWER-PRINTING STYLE FROM https://tex.stackexchange.com/a/15354 ----

% Define answer environment

```

```latex
\newbox\allanswers
\setbox\allanswers=\vbox{}

\newenvironment{customanswer}
{\global\setbox\allanswers=\vbox\bgroup
  \unvbox\allanswers
  \vspace{-4pt}
}
{\bigbreak\egroup}

\newcommand{\showallanswers}{\par\unvbox\allanswers}

% Define question environment

\newbox\allquestions
\setbox\allquestions=\vbox{}

\newenvironment{customquestion}
{\global\setbox\allquestions=\vbox\bgroup
  \unvbox\allquestions
}
{\bigbreak\egroup}

\newcommand{\showallquestions}{\par\unvbox\allquestions}

% ---------------------------------------------------------------------------



%% ---------- PROVIDE \timestamp COMMAND -------------
% -- from https://flaterco.com/util/timestamp.sty --

\makeatletter

\newcount\@DT@modctr
\newcount\@dtctr

\def\@modulo#1#2{%
  \@DT@modctr=#1\relax
  \divide \@DT@modctr by #2\relax
  \multiply \@DT@modctr by #2\relax
  \advance #1 by -\@DT@modctr}

\newcommand{\xxivtime}{%
  \@dtctr=\time%
  \divide\@dtctr by 60
  \ifnum\@dtctr<10 0\fi\the\@dtctr.%
  \@dtctr=\time%
  \@modulo{\@dtctr}{60}%
  \ifnum\@dtctr<10 0\fi\the\@dtctr%
}

\newcommand{\timestamp}{\the\year-%
  \ifnum\month<10 0\fi\the\month-%
  \ifnum\day<10 0\fi\the\day\ \xxivtime}

\makeatother

% -----------------------------------------------------
```

```

% ---------------- ACCÈS AU CODE LUA: -----------------

% Aide-mémoire Lua: https://devhints.io/lua
% (= Lua cheat-sheet)

% Activer les fonctions dans le code:
% Load lua code:
\directlua{codeA = require "codeQuadratique"}

% Convertir fonctions de code.lua en commandes LaTeX:
% Extract useful functions as LaTeX commands:

% Pour le format par défaut / for default style:
\newcommand{\afullroutine}[1]{\directlua{codeA.fullRoutine(#1)}}

% Pour imprimer question immédiatement suivie de réponse:
% To print question immediately followed by answer:
\newcommand{\aprintqna}{\directlua{codeA.printQnA()}}

% Pour librement produire question, et séparément la réponse
% To freely produce question, and separately answer
\newcommand{\amakequestion}{%
  % MUST BE INSERTED IN MATHMODE: \(\amakequestion\)
    \directlua{%
      polynomialcoeffs = codeA.polynomial()
      eqn = codeA.printEquation(table.unpack(polynomialcoeffs,1,3))
      tex.print(eqn)
    }
  }
\newcommand{\amakeanswer}{%
  % IN TEXTMODE
    \directlua{%
      tex.print(codeA.answer(table.unpack(polynomialcoeffs)))
    }
  }
```

Code A.6: Document LuaLaTeX pour lesdits exercices de second degré

```
%!TeX TS-program = lualatex
%!TeX spellcheck = fr

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% AUTHORSHIP & LICENCE %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This tool is distributed with the GPL-3.0-or-later
% (https://www.gnu.org/licenses/gpl-3.0.html) licence
% by its author, Alexandros Rispo Constantinou, from
% its GitHub page tytyvillus/laboratoire-didactique.
%
% https://github.com/tytyvillus/laboratoire-didactique
%
% Cet outil est distribué avec la licence GPL-3.0-or-later
% (https://www.gnu.org/licenses/gpl-3.0.html) par son auteur
% Alexandros Rispo Constantinou et est mis à disposition
% depuis sa page GitHub tytyvillus/laboratoire-didactique.

\documentclass[a4paper, 11pt]{article}
\usepackage[margin=2cm]{geometry}

```

```latex
\input{preamble-quadratique.tex}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%    DÉBUT DU DOCUMENT    %%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\setlength\parindent{0pt} % pour rendre les lignes plus compactes
\baselineskip=0.9\baselineskip

\begin{document} \thispagestyle{empty}

%%%%%%%%%%%%%%% CONSIGNE %%%%%%%%%%%%%%%

{\small\em Résolvez ces équations en utilisant à chaque fois la méthode la plus
   rapide possible. (N'oubliez pas d'écrire l'équation sous la forme standard si
   elle ne l'est pas déjà.) Une fois toutes les équations résolues, vérifiez vos
   réponses dans le corrigé. Indiquez pour chaque question si vous avez utilisé
   la même méthode que le corrigé ou non. Vous n'avez pas droit à la
   calculatrice.}

%%%%%%% GÉNÈRE LES QUESTIONS DANS UNE BOÎTE VIRTUELLE %%%%%%%

\raggedright
\foreach \n in {1,2,...,24}{\afullroutine{\n}}

%%%%%% MET TITRE ET HORODATAGE, PUIS IMPRIME EXERCICES %%%%%%

% "horodatage" = timestamp

\centering\vspace{-12pt}

Exercices \\{\tiny\sffamily (générés le \DTMtoday\ à \DTMcurrenttime s)} \\

\begin{multicols}{3}
  \showallquestions
\end{multicols}

\medskip

%%%%%%%% SÉPARATION ENTRE EXERCICES ET RÉPONSES %%%%%%%%

\hbox to \linewidth{\leaders\hbox to 4pt{\hss · \hss}\hfil}
% --> traits tillés (voir https://tex.stackexchange.com/a/474310)

% modifiable, par exemple remplaçant par nouvelle page \newpage

%%%%%% MET TITRE ET HORODATAGE, PUIS IMPRIME EXERCICES %%%%%%

\bigskip

Réponses \\{\tiny\sffamily (générées le \DTMtoday\ à \DTMcurrenttime s)} \\~\\

\linespread{0.5}\selectfont
\showallanswers

\end{document}
```

A.2.2 Systèmes de deux équations

Code A.7 : Préambule LuaLaTeX pour les exercices de résolution de systèmes d'équations

```
%!TeX root = main-systeme.tex


% --- PRÉAMBULE  ---

% langue et police:
\usepackage[quiet]{fontspec}
\usepackage{polyglossia} \setmainlanguage[variant=swiss]{french}

% calculs internes:
\usepackage{luacas}
%\usepackage[pl,import]{penlight} % désuet

% maths:
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage[ % choix esthétique, facultatif
warnings-off={mathtools-colon,mathtools-overbracket},
math-style=ISO,
]{unicode-math}
\usepackage[ % à régler selon préférences / change as preferred
% utilisé ici pour formatter les nombres à virgule avec \num{}
% used here to format decimal numbers via \num{}
locale = FR,
round-precision = 3,
round-mode = figures,
round-pad = false,
%zero-decimal-as-symbol = true, % choix personnel / personal choice
]{siunitx}
% vvvvvvvvvvvvvvvvvvvvv
% \newcommand{\num}[1]{#1}
%%  ^ à décommenter si vous voulez vous débarasser de <>siunitx
%%  ^ uncomment if you want to get rid of <>siunitx

% choix esthétiques, facultatifs:
\let\oldemptyset\emptyset
\let\emptyset\varnothing

% permettent mise en page:
\usepackage{multicol}
\usepackage{pgffor}

% date et heure
\usepackage[ % permet les horodatages (timestamps)
timesep=.,
showzone=false,
hourminsep=h,
minsecsep=m
]{datetime2}


% ---- ANSWER-PRINTING STYLE FROM https://tex.stackexchange.com/a/15354 ----

% Define answer environment

\newbox\allanswers
\setbox\allanswers=\vbox{}

\newenvironment{customanswer}
{\global\setbox\allanswers=\vbox\bgroup
```

```
  \unvbox\allanswers
  \vspace{-4pt}
}
{\bigbreak\egroup}

\newcommand{\showallanswers}{\par\unvbox\allanswers}

% Define question environment

\newbox\allquestions
\setbox\allquestions=\vbox{}

\newenvironment{customquestion}
{\global\setbox\allquestions=\vbox\bgroup
  \unvbox\allquestions
}
{\bigbreak\egroup}

\newcommand{\showallquestions}{\par\unvbox\allquestions}

% ----------------------------------------------------------------------------



%% ---------- PROVIDE \timestamp COMMAND -------------
% -- from https://flaterco.com/util/timestamp.sty --

\makeatletter

\newcount\@DT@modctr
\newcount\@dtctr

\def\@modulo#1#2{%
  \@DT@modctr=#1\relax
  \divide \@DT@modctr by #2\relax
  \multiply \@DT@modctr by #2\relax
  \advance #1 by -\@DT@modctr}

\newcommand{\xxivtime}{%
  \@dtctr=\time%
  \divide\@dtctr by 60
  \ifnum\@dtctr<10 0\fi\the\@dtctr.%
  \@dtctr=\time%
  \@modulo{\@dtctr}{60}%
  \ifnum\@dtctr<10 0\fi\the\@dtctr%
}

\newcommand{\timestamp}{\the\year-%
  \ifnum\month<10 0\fi\the\month-%
  \ifnum\day<10 0\fi\the\day\ \xxivtime}

\makeatother

% -----------------------------------------------------



% ---------------- ACCÈS AU CODE LUA: -----------------

% Aide-mémoire Lua: https://devhints.io/lua

```

```
% Activer les fonctions dans le code:
% Load lua code:
\directlua{codeB = require "codeSysteme"}

% Convertir fonctions de code.lua en commandes LaTeX:
% Extract useful functions as LaTeX commands:

% Pour le format par défaut / for default style:
\newcommand{\bfullroutine}[1]{\directlua{codeB.fullRoutine(#1)}}

% Pour imprimer question immédiatement suivie de réponse:
% To print question immediately followed by answer:
\newcommand{\bprintqna}{\directlua{codeB.printQnA()}}

% Pour librement produire question, et séparément la réponse
% To freely produce question, and separately answer
\newcommand{\bmakequestion}{%
  % MUST BE INSERTED IN MATHMODE: \(\amakequestion\)
    \directlua{%
      coeffs, num_sols, x, y = codeB.polynomial(0.2)
      eqn = codeB.printEquation(coeffs)
      tex.print(eqn)
    }
  }
\newcommand{\bmakeanswer}{%
  % IN TEXTMODE
    \directlua{%
      tex.print(codeB.answer(coeffs, num_sols, x, y))
    }
  }
```

Code A.8 : Document LuaLaTeX pour lesdits exercices de systèmes d'équations

```
%!TeX TS-program = lualatex
%!TeX spellcheck = fr

\documentclass[a4paper, 11pt]{article}
\usepackage[margin=2cm]{geometry}
\input{preamble-systeme.tex}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%     DÉBUT DU DOCUMENT     %%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\setlength\parindent{0pt} % pour rendre les lignes plus compactes
\baselineskip=0.9\baselineskip

\begin{document} \thispagestyle{empty}

%%%%%%%%%%%%%%% CONSIGNE %%%%%%%%%%%%%%%

{\small\em Résolvez ces systèmes d'équations en utilisant à chaque fois la
    méthode la plus rapide possible. (N'oubliez pas d'écrire l'équation sous la
    forme standard si elle ne l'est pas déjà.) Une fois toutes les équations
    résolues, vérifiez vos réponses dans le corrigé. Indiquez pour chaque question
    si vous avez utilisé la même méthode que le corrigé ou non. Vous n'avez pas
    droit à la calculatrice.}

%%%%%%% GÉNÈRE LES QUESTIONS DANS UNE BOÎTE VIRTUELLE %%%%%%

\raggedright
```

```
\foreach \n in {1, 2, ..., 18}{\bfullroutine{\n}}

%%%%%% MET TITRE ET HORODATAGE, PUIS IMPRIME EXERCICES %%%%%%

% "horodatage" = timestamp

\centering\vspace{-12pt}

Exercices \\{\tiny\sffamily (générés le \DTMtoday\ à \DTMcurrenttime s)}\\

\begin{multicols}{3}
  \showallquestions
\end{multicols}

\medskip

%%%%%%%% SÉPARATION ENTRE EXERCICES ET RÉPONSES %%%%%%%%

\hbox to \linewidth{\leaders\hbox to 4pt{\hss · \hss}\hfil}
% --> traits tillés (voir https://tex.stackexchange.com/a/474310)

% modifiable, par exemple remplaçant par nouvelle page \newpage

%%%%%% MET TITRE ET HORODATAGE, PUIS IMPRIME EXERCICES %%%%%%

\bigskip

Réponses \\{\tiny\sffamily (générées le \DTMtoday\ à \DTMcurrenttime s)}\\~\\

\linespread{0.5}\selectfont
\showallanswers

\end{document}
```

A.3 Algorithmes Lua

A.3.1 Équations du second degré

Code A.9 : Code Lua pour générer les équations du second degré avec réponses

```
-- All functions are defined locally, and only user-accessible functions are
   exported at the end.
-- The practices followed are given by [this stackexchange
   answer](https://tex.stackexchange.com/a/464049).

-- IMPORTANT NOTE: algorithm assumes a, b, c \in \mathbb{Z}
--> generalising the algorithm to non-integer a, b, c will be a bit of work,
--  because luacas isn't very happy with non-integer coefficients



----------------------------------------
--------  (EDITABLE) PREAMBLE  --------
----------------------------------------

-- SET GLOBAL SEED
--> set manually if you want a reproducible sheet:
math.randomseed(os.time()) -- e.g. 2 or os.time()


```

```
-- GENERAL PARAMETERS --> modify at will but at your own peril

-- Choose your preferred terminology for the quadratic formula:
local formule = [[formule quadr.\@]]
--> typical choices are [[formule de Viète]] or [[formule quadratique]]

local function whether_from_factored_form ()
    -- randomly decides whether choose [integer] solutions first (return true)
    -- or generate a, b, c at random (return false)
    if math.random(10) <= 7 --> weightings, feel free to change
        then return true
        else return false
    end
end

local function enforce_perfect_square ()
    -- randomly chooses to enforce a perfect-square polynomial (return true)
    -- or just leave things alone (return false)
    if math.random(10) == 1 --> weightings, feel free to change
        then return true
        else return false
    end
end

local function easy_factor(a, x1, x2)
    -- boolean test; decides whether a*x^2 + b*x + c is easy to factorise or not
    -- where x1, x2 are the solutions s.t. a*x^2 + b*x + c = a*(x-x1)*(x-x2)
    -- n.b. a::Int, x1::Float, x2::Float|Nil

    local decision = true -- by default, if x2 is nil, poly is easy to factor

    if x2 ~= nil then

        decision = (x1 % 1 == 0) and (x2 % 1 == 0) and --> check integer solutions
            (
                (
                    (
                        math.max(math.abs(x1),math.abs(x2)) <= 12
                        or math.min(math.abs(x1),math.abs(x2)) <= 3
                    )
                and
                    (math.abs(a) <= 5 or a == 10 or math.abs(a*x1*x2) % 10 == 0)
                )
                or (math.abs(a*x1*x2) <= 144)
            )
            --> these choices were made using gut instinct and trial-and-error
            -- feel free to change

    end

    return decision

end



--------------------------------------
---------  HELPER FUNCTIONS  ----------
--------------------------------------

local function is_one_of (value, table)
```

```
    -- checks whether value is one of the elements of table
    for idx, val in ipairs(table) do
        if value == val then return true end
    end
    return false --> this only gets triggered if all previous checks fail
end

local function gcd (a, b)
    if b == 0 then
        return a
    end
    return gcd(b, a % b)
end

local function lcm (a, b)
    return (a * b) / gcd(a, b)
end


local function map (tbl, f)
    -- applies function to table
    local t = {}
    for k,v in pairs(tbl) do
        t[k] = f(v)
    end
    return t
end

local function range (n)
    -- generates table from 1 to n
    assert(n % 1 == 0 and n > 0, "argument of range should be a positive integer")
    local list = {}
    for i = 1, n, 1 do
        list[i] = i
    end
    return list
end

local function approx (x, y)
    -- checks approximate equality, within 10^(-6)
    if math.abs(x-y) < 1e-6 then return true else return false end
end

local function table_concat(t1,t2)
    for i=1,#t2 do
        t1[#t1+1] = t2[i]
    end
    return t1
end


local function table_flatten_1 (matrix)
  local output = {}
  for i, list in ipairs(matrix) do
    local ouput = table_concat(output, list)
  end
  return output
end
```

```lua
---------------------------------------
----------  MAIN  ALGORITHM  ----------
---------------------------------------

-- ALGORITHM PART 1: POLYNOMIAL GENERATOR --

-- warning: have a lot of checks of type a==0

-- Generate quadratic polynomials:
local function generate_polynomial()
    -- returns {a, b, c, num_sols, x1, x2, rat}

    -- initialise generate_polynomial() outputs, fixes scope:
    local a, b, c, num_sols, x1, x2, delta

    -- decide whether integer solutions or not:
    local rat = whether_from_factored_form()

    if rat == true then -- integer solutions
        local sol_bdy = 12 -- boundary of solution range

        -- generate two solutions:
        x1 = math.random(-sol_bdy, sol_bdy)
        if enforce_perfect_square() then x2 = x1 else -- artificially bump up number of perfect squares
        x2 = math.random(-sol_bdy, sol_bdy) end

        -- also generate leading coefficient:
        a = (-1)^math.random(2) * math.random(10) -- avoid zero
        -- now compute other coefficients:
        b = - a * (x1 + x2)
        c = a * x1 * x2

        -- set number of and order solutions
        if x1 == x2 then num_sols, x2 = 1, nil -- set number of solutions to one and clear x2
        else num_sols = 2 -- set number of solutions to two
            if x1 > x2 then x1, x2 = x2, x1 end -- order solutions correctly
        end

    else -- totally random solutions
        a = math.random(-10,10) -- zero (linear equation) allowed (with low probability, cf rat == true)
        b = math.random(-20,20)
        c = math.random(-60,60)

        -- Compute solutions:
        if a == 0 then-- if linear then either
            if b == 0 then -- no x-dependence
                if c == 0 then num_sols, x1, x2 = math.huge, nil, nil -- "0 = 0"
                    --> might have to hardcode this special case later
                    --> essentially, here reserve num_sols = math.huge for S = \R
                else num_sols, x1, x2 = 0, nil, nil end -- "0 = 1"
            else -- or linear
                num_sols, x1, x2 = 1, -c / b, nil -- (and delta = nil)
        end
        else delta = b^2 - 4*a*c -- elseif quadratic
            -- if quadratic, do the three cases
            if delta == 0 then -- one solution with multiplicity two
                num_sols = 1
                x1 = - b / (2*a)
```

```
                x2 = nil
            elseif delta < 0 then -- no solutions
                num_sols = 0
                x1 = nil
                x2 = nil
            elseif delta > 0 then -- two separate solutions
                num_sols = 2
                x1 = (- b - math.sqrt(delta)) / (2*a)
                x2 = (- b + math.sqrt(delta)) / (2*a)
            else assert(false,"You broke the law of excluded middle, wtf dude.")
            end
        end
    end

    return {a, b, c, num_sols, x1, x2, rat}
end




-- ALGORITHM PART 2: OPTIMAL METHOD DETECTOR --

local function pick_method(polynomial_info, override_method)
    -- polynomial_info is the list a, b, c, num_sols, x1, x2, rat
        -- override_method is an optional input, of the form {bool: overwrite
            method?, string: new method}

    local a, b, c, num_sols, x1, x2, rat = table.unpack(polynomial_info)
    -- a, b, c are the polynomial coefficients (number type)
    -- num_sols is the number of solutions (number type, values in {0, 1, 2})
    -- x1, x2 are the solutions of the equation a x^2 + b x + c = 0 (number type)
    -- rat is a flag for which version of the generation algorithm was used
        (boolean type)

    --assert(num_sols == math.huge or (num_sols >= 0 and num_sols <= 2), --
        remove if works
    assert( is_one_of(num_sols, {0, 1, 2, math.huge}),
        "Woah, something went wrong -- I didn't receive a sensible number of
            solutions.")

    local method = "" -- initialise string containing the answer method

    -- run through different preset methods when problem has solution:
    if num_sols == math.huge then method = [[par évidence]] --> S = \R case

    elseif num_sols ~= 0 then
        if a==0 then method = [[équation de premier degré, à résoudre
            algébriquement]]
        elseif b==0 then method = [[résolution algébrique classique
            \(\left(\text{attention à }\pm\sqrt{\square}\right)\)]]
        elseif c==0 then method = [[par mise en évidence de ]] -- then by x or
            a*x, depending
            if a==1 then method = method..[[\(x\)]]
            else method =  method..string.format([[\(%d x\)]], a)
            end
        elseif -- (rat == true) or -- "rat==true" commented out because should
            already be covered by easy_factor
            easy_factor(a, x1, x2) -- test whether easily factorised by hand
            then
                if num_sols == 1 then method = [[par identité remarquable (carré
                    parfait)]] -- (x \pm x1)^2 = 0
```

```
                    else method = [[par factorisation du trinôme]]
                end
                if a ~= 1 -- then, if a =/= 1, remind of necessity of bringing $a$ out the front
                and not (override_method and not override_method[1]) -- check whether already being told about premultiplying
                then method = string.format([[diviser par \(%d\), puis ]], a)..method end
        else method = formule -- [[formule de Viète]]
        end

    -- give instead methods when problem has no solution:
    else method = [[sans solutions]] -- initalise as no solutions, just in case
        if a==0 and b==0 and c~=0 then method = [[évidemment]]
        elseif b==0 then method = [[somme de nombres du même signe ne fait jamais zéro]]
        else method = formule..[[ / en calculant le discriminant \((\Delta)\)]]
        end
    end

    -- override method if equation was weirdly printed
    if override_method then --> checks whether nil
        if override_method[1] then method = override_method[2] --> check whether need to overwrite
        else method = override_method[2]..method --> else just prepend
        end
    end

    return method
end


---------------------------------------
----------- PREPARE OUTPUTS -----------
---------------------------------------

-- PRINT LUALATEX-FORMATTED EQUATION --

local function cas_equation (a, b, c)
    -- prints equation using luacas
    -- outputs string AND (optionally!) method override

    local tex_string = "" -- initialise output equation string

    local override_method = nil -- initialise optional method override
        --> this is of the form {Bool, String},
        --  where Bool is whether to overwrite
        --  (true = overwrite, false = prepend)
        --  and String is the method to use

    -- randomly choose form, then create with luacas:
    if math.random(10) > 2 then --> normal in 8/10 cases
        -- f(x) = a*x^2 + b*x + c
        tex_string = string.format(
            [[
                \begin{CAS}
                    vars('x')
                    f = %d * x^2 + %d * x + %d
                    f = topoly(f)
                \end{CAS}
                \print{f} = 0
```

```
            ]], -- work out in luacas, then print
            a, b, c
        )
    elseif a*b ~= 0 and math.random(3) == 1 then
        -- x = - c/b - a/b x^2
            --> n.b. a must be non-zero otherwise get x = -c/b
        tex_string = string.format(
            [[
                \begin{CAS}
                    vars('x')
                    a = %d
                    b = %d
                    c = %d
                    f = topoly(- c / b - a / b * x^2)
                \end{CAS}
                x = \print{f}
            ]],
            a, b, c
        )
        if c == 0 then --> easy and obvious, so override predicted method
            override_method = {true, string.format(
                [[par réarrangement et mise en évidence de \(\begin{CAS}
                temp = (%d) / (%d) \end{CAS}\print{temp} x\)]],
                a, b)
            }
        elseif (c/b % 1) ~= 0 or (a/b % 1) ~= 0 then --> if non-integer, tell them to multiply to kill fractions
            override_method = {false, string.format([[multiplier par %d, puis ]],
                math.abs(lcm(b / gcd(b, c), b / gcd(a, b)))) --> gets lcm of denominators
            }
        else override_method = {false, ""} --> to kill the 'multiplier par a', without changing method
        end
    elseif a*b ~= 0 and math.random(2) == 1 then
        -- c/a = -x(x + b/a)
            --> n.b. b must be non-zero otherwise get c/a = -x(x)
        tex_string = string.format(
            [[
                \begin{CAS}
                    vars('x')
                    a = %d
                    b = %d
                    c = %d
                    f = c/a
                    g = topoly(-x(x + b/a))
                \end{CAS}
                \print{f} = \print{g}
            ]],
            a, b, c
        )
        if c == 0 then override_method = {true, [[par le principe du produit nul]]} --> easy so override
        elseif (c/a % 1) ~= 0 or (b/a % 1) ~= 0 then --> if non-integer, tell them to multiply to kill fractions
            override_method = {false, string.format([[multiplier par %d, puis ]],
                math.abs(lcm(a / gcd(a, c), a / gcd(a, b)))) --> gets lcm of denominators
            }
        else override_method = {false, ""} --> to kill the 'multiplier par a', without changing method
```

```
        end
    else -- a x^2 = -b x - c
        tex_string = string.format(
            [[
                \begin{CAS}
                    vars('x')
                    a = %d
                    b = %d
                    c = %d
                    f = topoly(a*x^2)
                    g = topoly(-b*x-c)
                \end{CAS}
                \print{f} = \print{g}
            ]],
            a, b, c
        )
    end

    return tex_string, override_method
end

-- PRINT LUALATEX-FORMATTED ANSWER --

local function cas_sol_set (polynomial_info)
    -- gets luacas to calculate and simplify solutions
    -- outputs string to tex.print in math environment

    local a, b, c, num_sols, x1, x2, rat = table.unpack(polynomial_info)

    local S -- initialise output string

    -- cases:

    if num_sols == math.huge then S = [[\mathbb{R}]] -- "0 = 0"

    elseif num_sols == 0 then S = [[\emptyset]] -- no sols

    elseif num_sols == 1 then
        if a == 0 then S = string.format( -- one sol to linear eqn
            [[
            \begin{CAS}
            x1 = - %d / %d
            \end{CAS}
            \left\{ \print*{x1} \right\}
            ]],
            c, b
        ) --> x = - c / b
        else S = string.format( -- one sol to quadratic eqn
            [[
            \begin{CAS}
            x1 = - %d / (2*%d)
            \end{CAS}
            \left\{ \print*{x1} \right\}
            ]],
            b, a
        ) --> x1 = b / (2*a)
        end

    elseif num_sols == 2 then S = string.format( -- two sols
        [[
        \begin{CAS}
```

```
        vars('x')
        f = %d*x^2 + %d*x + %d
        S = roots(f)
        \end{CAS}
        \left\{ \print*{S[1]}; \print*{S[2]} \right\}
        ]],
        a, b, c
    ) --> roots of a*x^2 + b*x + c

    else assert(false,"There is deep error -- number of solutions not being tracked correctly.")
    end -- debugging

    return S
end

local function num_sol_set (polynomial_info)
    -- create numerical solution set as string
    -- CURRENTLY UNDER DEBUGGING

    local a, b, c, num_sols, x1, x2, rat = table.unpack(polynomial_info)

    local sol_set_list = {} -- initialise solution set
    local S = "" -- initialise output string

    -- since appending nil to an empty table does nothing, append both solutions
    table.insert(sol_set_list, x1) -- first small
    table.insert(sol_set_list, x2) -- then big

    -- preprocess formatting: turn entries into strings with \num
    for idx, val in ipairs(sol_set_list) do --> cycle through elements
        sol_set_list[idx] = string.format([[\num{%s}]], val)
    end

    -- preprocess formatting: combine into output string with set notation
    --> if empty, replaces with empty set symbol; else just fills braces
    if next(sol_set_list) == nil --> check table empty
        then S = [[\emptyset]]
        else S = string.format([[\left\{%s\right\}]], table.concat(sol_set_list, "; "))
    end

    return S
end

local function answer_line (polynomial_info, --[[optional]]override_method)
    -- print the answer in the form "par factorisation, $S = \{1; 2\}$" :

    local a, b, c, num_sols, x1, x2, rat = table.unpack(polynomial_info)

    assert(not override_method or type(override_method) == "table", "override_method is not a table...")

    local output_string = string.format(
        [[{%s}, \(S = %s\)]],
        pick_method(polynomial_info, override_method),
        cas_sol_set(polynomial_info)
    )

    -- if solutions aren't integers, then also give numerical values
    if is_one_of(num_sols, {1, 2})
```

```
    and ( (x1 % 1 ~= 0) or (x2 ~= nil and x2 % 1 ~= 0) ) --> if there are non-integer solutions
        then
            output_string = output_string..string.format(
                [[\( \approx %s \)]],
                num_sol_set(polynomial_info) -- nb: fn only actually uses x1, x2
            )
        --> formats and appends numerical solutions x1, x2 to output string
    end

    return output_string
end



-- FULL ROUTINE

local function print_questions_and_answers()
    -- compute some coefficients
    local polynomial_coeffs = generate_polynomial()

    local equation_string, override_method = cas_equation(table.unpack(polynomial_coeffs,1,3))
        --> cas provides equation and optional override for the method string
        --> n.b. override_method is an optional output, so might well be nil

    -- print the equation
    tex.print([[\begin{equation}]]..equation_string..[[\end{equation}]])
    --> take printed equation string, enclose in $$.$$ and write to tex

    -- print the solution
    tex.print(
        answer_line(polynomial_coeffs, override_method)
            --> = "avec [méthode], S = {x1;x2}"
    )

end

local function full_routine(n)
    -- takes an integer and outputs question + answer text numbered with that integer

    -- compute some coefficients
    local polynomial_coeffs = generate_polynomial()

    local unwrapped_equation, override_method = cas_equation(table.unpack(polynomial_coeffs,1,3))
    --> cas provides equation and optional override for the method string

    -- wrap equation string
    local eqn_string = [[\(]]..unwrapped_equation..[[\)]] --> enclose in \(...\)

    -- create answer string
    local ans_string = answer_line(polynomial_coeffs, override_method)
        --> = "avec [méthode], S = {x1;x2}"

    -- produce output tex with properly formatted answer key
    local output_string = string.format([[
        \begin{customquestion} %02d.~~\ %s \end{customquestion}
    \begin{customanswer}   %02d.~~  \ %s \end{customanswer}
        ]], n, eqn_string, n, ans_string)
```

```

    tex.print(output_string)
end


-- INTERFACE WITH LUALATEX ENGINE --

-- Export user-accessible functions (renamed using syntax `new = old`):
return {
    polynomial = generate_polynomial, -- returns {a, b, c, num_sols, x1, x2, rat}
    methodString = pick_method, -- provides recommended method
    printEquation = cas_equation, -- question preprinted for LaTeX
    answer = answer_line, -- answer preprinted for LaTeX
    fullRoutine = full_routine, -- whole shebang
    printQnA = print_questions_and_answers, -- alternative, single-equation
        formatting style
}
```

A.3.2 Systèmes de deux équations

Code A.10 : Code Lua pour générer les systèmes de deux équations avec réponses

```
-- All functions are defined locally, and only user-accessible functions are
    exported at the end.
-- The practices followed are given by [this stackexchange
    answer](https://tex.stackexchange.com/a/464049).

-- IMPORTANT NOTE: algorithm assumes a, b, c \in \mathbb{Z}
--> generalising the algorithm to non-integer a, b, c will be a bit of work,
--  because luacas isn't very happy with non-integer coefficients


----------------------------------------
--------  (EDITABLE) PREAMBLE  --------
----------------------------------------

-- SET GLOBAL SEED
--> set manually if you want a reproducible sheet:
math.randomseed(os.time()) -- e.g. 2 or os.time()

-- Set probability to generate singular matrices (number between 0 and 1)
probability_singular = 0.2 -- here 20% of the generated singular equations
    systems will be kept (the real proportion is way less, since the probability
    of generating a singular matrix fully at random approaches 0)

local function whether_singular ()
    -- randomly decides whether choose top have a unique solution first (return
        true)
    -- or generate a singular system (return false)
    if math.random(10)/10 <= probability_singular --> weightings, feel free to
        change
         then return true
         else return false
    end
end

-- HELPER FUNCTIONS --

local function is_one_of (value, table)
    -- checks whether value is one of the elements of table
```

```
    for idx, val in ipairs(table) do
        if value == val then return true end
    end
    return false
end

local function map (tbl, f)
    -- applies function to table
    local t = {}
    for k,v in pairs(tbl) do
        t[k] = f(v)
    end
    return t
end

local function range (n)
    -- generates table from 1 to n
    assert(n % 1 == 0 and n > 0, "argument of range should be a positive integer")
    local list = {}
    for i = 1, n, 1 do
        list[i] = i
    end
    return list
end

local function approx (x, y)
    -- checks approximate equality, within 10^(-6)
    if math.abs(x-y) < 1e-6 then return true else return false end
end

function flatten(v)
    local res = {}
    local function flatten(v)
        if type(v) ~= "table" then
            table.insert(res, v)
            return
        end
        for _, v in ipairs(v) do
            flatten(v)
        end
    end
    flatten(v)
    return res
end


----------------------------------------
----------  MAIN  ALGORITHM  ----------
----------------------------------------

-- ALOGRITHM PART 0: EQUATION SOLVER --

-- Function to solve a system of linear equations by linear combination (Cramer's rule)
-- If the system is of the form a1x + b1y = c1 , a2x + b2y = c2

function solveLinearEquations(system)
    local a1, b1, c1 = system[1], system[2], system[3]
    local a2, b2, c2 = system[4], system[5], system[6]
    -- Calculate determinant
    local determinant = a1 * b2 - b1 * a2
```

```
    -- Check if the determinant is zero
    if determinant == 0 then
        print("The lines are either overlapping or parallel.")
        return nil
    else
        -- Calculate x and y using Cramer's rule
        local x = (b2 * c1 - b1 * c2) / determinant
        local y = (a1 * c2 - a2 * c1) / determinant

        -- local x_num, x_den = decimalToFraction(x)
        -- local y_num, y_den = decimalToFraction(y)
        return x, y
    end
end


-- ALGORITHM PART 1: POLYNOMIAL GENERATOR --

-- warning: have a lot of checks of type a==0

-- Generate two factors linear polynomials:
 -- generate 2x2 matrices to have 0, 1 or infinitely many solutions depending on
    the random number given (if the matrix is singular, then it will be wiped
    away 80% of the time)
local function generate_system(probability_singular)

    local rat = whether_singular()

    if rat == true then -- integer solutions
        if math.random()<= 0.5 then
            local coef_bdy = 12 -- boundary of coefficient range

            -- generate first equation
            local a1 = math.random(-coef_bdy, coef_bdy)
            local b1 = math.random(-coef_bdy, coef_bdy)
            local c1 = math.random(-coef_bdy, coef_bdy)

            -- also generate proportionality coefficient:
            local k = math.random(-7, 7)
            -- now compute other coefficients:
            local a2, b2, c2 = k*a1, k*b1, k*c1
            system = {a1, b1, c1, a2, b2, c2}
            return system
        else
            local coef_bdy = 12 -- boundary of coefficient range

            -- generate first equation
            local a1 = math.random(-coef_bdy, coef_bdy)
            local b1 = math.random(-coef_bdy, coef_bdy)
            local c1 = math.random(-coef_bdy, coef_bdy)

            -- also generate proportionality coefficient:
            local k = math.random(-7, 7)
            -- now compute other coefficients:
            local a2, b2, c2 = k*a1, k*b1, c1

            system = {a1, b1, c1, a2, b2, c2}

            return system
        end
```

```lua

    else -- Generate random values for the matrix
        local a1 = math.random(-10, 10)
        local b1 = math.random(-10, 10)
        local a2 = math.random(-10, 10)
        local b2 = math.random(-10, 10)

        -- Choose arbitrary constants for the linear equations
        local c1 = math.random(-15, 15)
        local c2 = math.random(-15, 15)
        local system = {a1, b1, c1, a2, b2, c2}
        -- Check if the determinant is non-zero
        local determinant = a1 * b2 - b1 * a1

        if determinant==0 then
            return generate_system(probability_singular)
        else
            -- Return the system
            return system
        end
    end
end

    -- Function to generate a system of linear equations using the coefficients
       the generated matrix
local function generate_linear_system(matrix)
    -- Choose arbitrary constants for the linear equations
    local c1 = math.random(-15, 15)
    local c2 = math.random(-15, 15)

    -- Extract coefficients from the matrix
    local a1, b1, a2, b2 = matrix[1][1], matrix[1][2], matrix[2][1], matrix[2][2]

    -- Generate the system of linear equations
    local system = {a1, b1, c1, a2, b2, c2}
    return system
end

    -- function that determines the number of solutions of the system
  -- function that determines the number of solutions of the system
local function number_solutions_two(eqs)
    -- Coefficients from equations
    local a1, b1, c1 = eqs[1], eqs[2], eqs[3]
    local a2, b2, c2 = eqs[4], eqs[5], eqs[6]
    local det = a1 * b2 - a2 * b1

    if det ~= 0 then
        -- unique solution
        return 1
    else
        -- verifiy if the lines coincide or if they are parallel
        local ratio_a = a1 / a2
        local ratio_b = b1 / b2
        local ratio_c = c1 / c2

        if (a1 == 0 and a2 == 0) and (b1 == 0 and b2 == 0) then
            if c1 == c2 then
                return math.huge  -- zero system
            else
                return 0  -- Impossible if c1 != c2
            end
```

```
        elseif (a1 == 0 and a2 == 0) or (b1 == 0 and b2 == 0) then
            -- one variable is totally missing
            if ratio_c == (b1 == 0 and b2 == 0 and ratio_a or ratio_b) then
                return math.huge
            else
                return 0
            end
        else
            if ratio_a == ratio_b and ratio_b == ratio_c then
                -- lines coincide
                return math.huge
            else
                -- parallel lines
                return 0
            end
        end
    end
end

local function generate_exercise(probability_singular)
    local system = generate_system(probability_singular)
    local num_sols = number_solutions_two(system)
    local x, y = solveLinearEquations(system)

    return system, num_sols, x, y
end

-- ALGORITHM PART 2: OPTIMAL METHOD DETECTOR --

local function pick_method_case(system, num_sols)
    -- system is the matrix of coefficients of the system
    -- num_sols is the number of solutions (number type, values in {0, 1, math.huge})
    a, b, d, e = system[1], system[2], system[4], system[5]
    assert( is_one_of(num_sols, {0, 1, math.huge}),
        "Woah, something went wrong -- I didn't receive a sensible number of solutions.")

    local method = "" -- initialise string containing the answer method

    -- run through different preset methods when problem has solution:
    if num_sols == math.huge then method = [[les deux équations sont dépendantes]] --> S = line case, i.e. (a&d==0 and b&e==0 and c&f==0) or (d&a==0 and e&b==0 and f&c==0)

    elseif num_sols == 1 then
        if a==0 or b==1 then method = [[isoler \(y\) dans la 1\textsuperscript{re} équation, substituer ensuite]]
        elseif b==0 or a==1 then method = [[isoler \(x\) dans la 1\textsuperscript{re} équation, substituer ensuite]]
        elseif d==0 or e==1 then method = [[isoler \(y\) dans la 2\textsuperscript{e} équation, substituer ensuite]]
        elseif e==0 or d==1 then method = [[isoler \(x\) dans la 2\textsuperscript{e} équation, substituer ensuite]]
        elseif a==d or b==e then method = [[soustraction directe des deux équations]]
        elseif a==-d or b==-e then method = [[addition directe des deux équations]]
        elseif a%b==0 then method = string.format([[diviser par \(%d\) puis isoler \(y\) dans la 1\textsuperscript{re} équation, substituer ensuite]], b)
```

```
        elseif b%a==0 then method = string.format([[diviser par \(%d\) puis isoler \(x\) dans la 1\textsuperscript{re} équation, substituer ensuite]], a)
        elseif d%e==0 then method = string.format([[diviser par \(%d\) puis isoler \(y\) dans la 2\textsuperscript{e} équation, substituer ensuite]], e)
        elseif e%d==0 then method = string.format([[diviser par \(%d\) puis isoler \(x\) dans la 2\textsuperscript{e} équation, substituer ensuite]], d)
        else method = [[par combinaison linéaire]]
        end

    -- give instead methods when problem has no solution:
    else method = [[sans solutions (combinaison linéaire donne $1=0$)]]
    end
    return method
end

---------------------------------------
----------- PREPARE OUTPUTS -----------
---------------------------------------

-- PRINT LUALATEX-FORMATTED EQUATION --

local function cas_equation (system)
    -- prints equation using luacas

    tex_string = string.format(
            [[
                \begin{CAS}
                    vars('x', 'y')
                    f = Equation(%d * x + %d * y, %d)
                    g = Equation(%d * x + %d * y, %d)
                \end{CAS}
                \left\{\begin{array}{c} \print*{f} \\ \print*{g}\end{array}\right.
            ]], -- work out in luacas, then print
            table.unpack(system)
        )
    return tex_string
end

-- PRINT LUALATEX-FORMATTED ANSWER --

local function cas_sol_set (system, num_sols, x, y)
    -- outputs string to tex.print in math environment

    local S -- initialise output string

    -- cases:

    if num_sols == math.huge then
        S = string.format(
        [[
            \begin{CAS}
            vars('x', 'y')
            f= Equation(%d * x + %d * y, %d):autosimplify()
            eqx = f:solvefor(y)
            \end{CAS}
            \left\{
                \left(x, \print{eqx.rhs}\right) \; \middle| \; x \in \mathbb{R}
            \right\}
```

```
        ]], -- droites confondues
        table.unpack(system)
    )

    elseif num_sols == 0 then S = [[\emptyset]] -- no sols

    --  A MODIFIER ABSOLUMENT POUR AVOIR LES FRACTIONS EXACTES DES SOLUTIONS
    elseif num_sols == 1 then
        S = string.format(
        [[
            \begin{CAS}
            determinant = %d * %d - %d * %d
            x = (%d * %d - %d * %d) / determinant
            y = (%d * %d - %d * %d) / determinant
            \end{CAS}
            \left\{\left(\print*{x}, \print*{y} \right)\right\}
        ]], -- droites sécantes
        system[1], system[5], system[4], system[2], system[5], system[3], system[2], system[6], system[1], system[6], system[4], system[3]
    )
    else assert(false,"There is a deep error -- number of solutions not being tracked correctly.")
    end
    return S
end

local function num_sol_set (system, num_sols, x, y)
    -- create numerical solution set as string
    -- CURRENTLY UNDER DEBUGGING

    local sol_set_list = {} -- initialise solution set
    local S = "" -- initialise output string

    -- since appending nil to an empty table does nothing, append both solutions
    table.insert(sol_set_list, x) -- first small
    table.insert(sol_set_list, y) -- then big

    -- preprocess formatting: turn entries into strings with \num
    for idx, val in ipairs(sol_set_list) do --> cycle through elements
        sol_set_list[idx] = string.format([[\num{%s}]], val)
    end

    -- preprocess formatting: combine into output string with set notation
    --> if empty, replaces with empty set symbol; else just fills braces
    if next(sol_set_list) == nil --> check table empty
        then S = [[\emptyset]]
        else S = string.format([[\left\{%s\right\}]], table.concat(sol_set_list, "; "))
    end

    return S
end

local function answer_line (system, num_sols, x, y)
    -- print the answer in the correct form:

    local output_string = string.format(
        [[{%s}, \(S = %s\)]],
        pick_method_case(system, num_sols),
        cas_sol_set(system, num_sols, x, y)
    )
```

```lua

    return output_string
end


-- FULL ROUTINE

local function print_questions_and_answers()
    -- compute some coefficients and the associated equations
    local coeffs, num_sols, x, y = generate_exercise(probability_singular)

    -- print the equation
    tex.print(
        [[\begin{equation}]]..
        cas_equation(coeffs) --> cas provides equation
        ..[[\end{equation}]] --> then, enclose in $$.$$
    ) --> write to tex

    -- print the solution
    tex.print(
        answer_line(coeffs, num_sols, x, y) --> = "avec [méthode], S = ..."
    )

end

local function full_routine(n)
    -- takes an integer and outputs question + answer text numbered with that integer
    -- compute coefficients and the associated equations
    local coeffs, num_sols, x, y = generate_exercise(probability_singular)

    -- create equation string
    local eqn_string = [[\(]]..
        cas_equation(coeffs) --> cas provides equation
        ..[[\)]] --> then, enclose in \(...\)

    -- create answer string
    local ans_string = answer_line(coeffs, num_sols, x, y) --> = "avec [méthode], S = {x1;x2}"

    -- produce output tex with properly formatted answer key
    local output_string = string.format([[
        \begin{customquestion} %02d.~~\ %s \end{customquestion}
    \begin{customanswer}   %02d.~~   \ %s \end{customanswer}
        ]], n, eqn_string, n, ans_string)

    tex.print(output_string)
end

-- INTERFACE WITH LUALATEX ENGINE --

-- Export user-accessible functions (renamed using syntax `new = old`):
return {
    polynomial = generate_exercise, -- returns {coefficients, num_sols, x, y}
    methodString = pick_method_case, -- provides recommended method
    printEquation = cas_equation, -- question preprinted for LaTeX
    answer = answer_line, -- answer preprinted for LaTeX
    fullRoutine = full_routine, -- whole shebang
    printQnA = print_questions_and_answers, -- alternative, single-equation formatting style
}
```

— B —

Questionnaire du GCM aux enseignant·e·s de mathématiques fribourgeois·e·s

Groupe Cantonal de Mathématiques 2016 *Kantonale Arbeitsgruppe Mathematik*

Analyse der zu erwartenden Kompetenzen

Teil 1 : Algebra im ersten Jahr

Bien que les collèges du canton de Fribourg disposent d'un plan d'étude commun pour l'enseignement des mathématiques, certaines différences entre écoles et entre sections linguistiques sont fréquemment évoquées. Monsieur Piccand et la CORECOF nous ont demandé de faire un état des lieux aussi précis que possible, en commençant par une analyse des compétences visées en fin de première année.

Nous aimerions que chaque enseignant de mathématiques nous renseigne sur ses attentes. Aussi, nous vous demandons de compléter ce questionnaire jusqu'au **vendredi 15 avril** en prenant connaissance de la liste d'exercices ci-dessous et cochant des cases selon la convention suivante :

- $\overset{\forall}{\boxtimes} \equiv$ contenu devant être enseigné avec un objectif d'acquisition par tous les élèves ;
- $\overset{\exists}{\boxtimes} \equiv$ contenu devant être enseigné avec un objectif d'acquisition par un maximum d'élèves ;
- $\overset{\varnothing}{\boxtimes} \equiv$ contenu ne devant pas être enseigné ou devant être traité sans objectif d'acquisition par les élèves.

Nous sommes persuadés que la liberté que le plan d'étude actuel nous laisse est bénéfique à l'enseignement des mathématiques et nous souhaitons la préserver. L'objectif de ce questionnaire est d'identifier nos différences afin de mieux pouvoir les gérer en cas de changement de section linguistique voire d'école.

Bien que nous ayons essayé de composer cette liste du mieux possible, nous vous invitons à nous communiquer tous les manquements éventuels, tant au niveau du contenu que de la forme. Les personnes de contact sont Fabien Augsburger et Ole Raemy pour le collège de Gambach, Gisela Bissig et Dominique Murith pour le collège Sainte-Croix, Marius Fux et Yves Roisin pour le collège Saint-Michel et Jérôme Charrière et Laurent Karth pour le collège du Sud.

En vous souhaitant une agréable lecture et en vous remerciant d'avance de votre précieuse collaboration, nous vous adressons nos salutations les meilleures.

Gisela Bissig et Yves Roisin

Ensembles, nombres et opérations

1

Représenter les ensembles

$$A = \{\alpha;\beta;\gamma;\mu\} \qquad B = \{\beta;\delta;\lambda;\mu;\psi\} \qquad C = \{\alpha;\beta;\gamma;\delta;\mu;\psi;\varphi;\omega\} \qquad D = \{\sigma;\epsilon\}$$

à l'aide d'un diagramme de Venn et répondre aux questions suivantes :

(1) Compléter avec les symboles $\in$, $\notin$, $\subset$ ou $\not\subset$.

(a) $\alpha \quad A$

(b) $\alpha \quad B$

(c) $\{\delta\} \quad A$

(d) $\{\delta\} \quad B$

(e) $\{\alpha;\sigma\} \quad C$

(f) $\varnothing \quad D$

(2) Donner les ensembles suivants par énumération.

(a) $A \cup B$

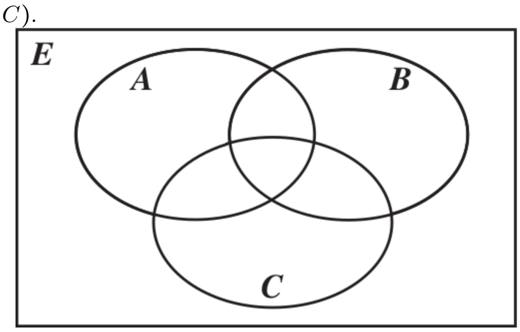
(b) $A \cap B$

(c) $C \cap D$

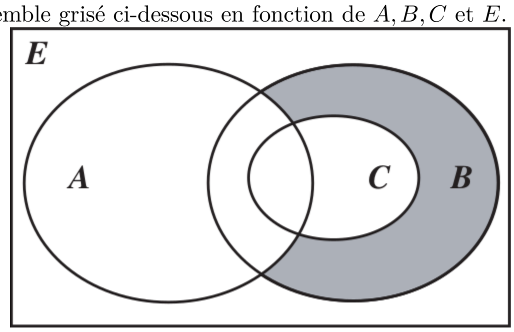
(d) $C \setminus B$

2

(1) Dans le diagramme de Venn ci-dessous, colorier le sous-ensemble correspondant à l'expression $[A \setminus (B \cup C)] \cup (B \cap C)$.



(2) Exprimer le sous-ensemble grisé ci-dessous en fonction de A, B, C et E.



3

Le dépouillement d'un sondage effectué sur 180 personnes a montré que ...

- ... 75 d'entre elles regardent des matchs de tennis à la télévision ;
- ... 60 font du tennis ;
- ... 40 regardent des matchs de tennis à la télévision et pratiquent elles-mêmes ce sport.

Considérons les ensembles

$$E = \{x \mid x \text{ est une personne interrogée lors du sondage}\}$$

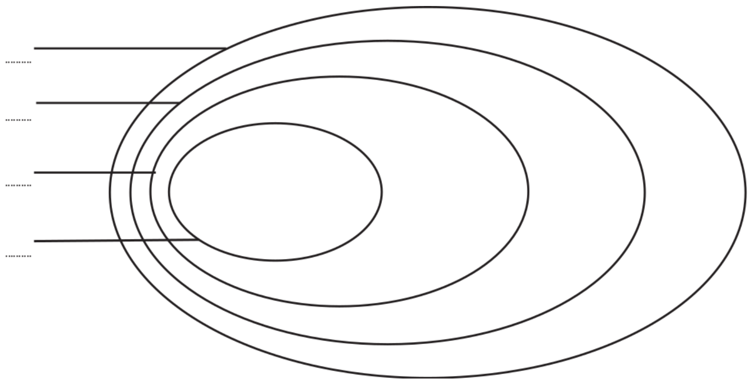
$$A = \{x \in E \mid x \text{ regarde des matchs de tennis à la TV}\}$$

$$B = \{x \in E \mid x \text{ pratique le tennis}\}$$

Représenter dans un diagramme de Venn les ensembles E, A et B de telle sorte qu'aucun sous-ensemble ne soit vide, puis indiquer dans chaque sous-ensemble du diagramme le nombre d'éléments qu'il contient.

4

(1) Désigner les ensembles de la figure ci-dessous par leur symbole usuel $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ ou $\mathbb{R}$.



(2) Placer les nombres suivants dans le bon sous-ensemble de la figure ci-dessus.

(a) 0

(b) 4

(c) -2

(d) π

(e) 0.75

(f) $\frac{\sqrt{2}}{5}$

(g) $4.0\bar{1}$

(h) $0.\bar{9}$

(i) $\frac{\sqrt{48}}{\sqrt{3}}$

5

Compléter les affirmations suivantes avec le symbole d'appartenance $\in$ ou le symbole de non-appartenance $\notin$ sans utiliser la calculatrice.

∀ ∃ ∅ ☐ ☐ ☐ (1) $\dfrac{62}{2} \quad \mathbb{Q}_+$

∀ ∃ ∅ ☐ ☐ ☐ (2) $8 \quad \mathbb{Z}_-$

∀ ∃ ∅ ☐ ☐ ☐ (3) $0 \quad \mathbb{R}^*$

∀ ∃ ∅ ☐ ☐ ☐ (4) $-\pi \quad \mathbb{R}^*_-$

∀ ∃ ∅ ☐ ☐ ☐ (5) $0.42 \quad]-0.42;0.42]$

∀ ∃ ∅ ☐ ☐ ☐ (6) $\dfrac{3}{7} \quad]-0.42;0.42]$

∀ ∃ ∅ ☐ ☐ ☐ (7) $-\dfrac{3}{7} \quad]-0.42;0.42]$

∀ ∃ ∅ ☐ ☐ ☐ (8) $-0.41 \quad]-0.42;0.42]$

∀ ∃ ∅ ☐ ☐ ☐ (9) $|-4| \quad \mathbb{N}$

6

On considère les ensembles suivants :

- $A = \{x \in \mathbb{R} \,|\, x = n^2 \text{ pour } n \in \mathbb{Z}^* \text{ et } -2 \leqslant n < 4\}$
- $B = \{x \in \mathbb{R} \,|\, -2 \leqslant x < 4\}$
- $C =]-\infty;3[$

Si c'est possible, écrire les ensembles suivants sous forme d'intervalle. Si ce n'est pas possible, les donner par énumération.

Seules les réponses sont demandées.

∀ ∃ ∅ ☐ ☐ ☐ (1) A

∀ ∃ ∅ ☐ ☐ ☐ (2) B

∀ ∃ ∅ ☐ ☐ ☐ (3) $B \cup C$

∀ ∃ ∅ ☐ ☐ ☐ (4) $C \cap B$

∀ ∃ ∅ ☐ ☐ ☐ (5) $B \setminus C$

7

Sans utiliser la calculatrice, effectuer et simplifier au maximum les calculs suivants.

∀ ∃ ∅ ☐ ☐ ☐ (1) $\dfrac{1}{2} + 0.\bar{3}$

∀ ∃ ∅ ☐ ☐ ☐ (2) $\dfrac{1}{3} - \dfrac{1}{6} - \dfrac{1}{15}$

∀ ∃ ∅ ☐ ☐ ☐ (3) $\dfrac{1}{24} + \dfrac{1}{36}$

∀ ∃ ∅ ☐ ☐ ☐ (4) $\dfrac{5}{120} + \dfrac{1}{36}$

∀ ∃ ∅ ☐ ☐ ☐ (5) $\dfrac{1}{252} - \dfrac{1}{360}$

(6) $0.17 - \dfrac{3}{-25}$

(7) $\dfrac{8}{36} + 0.\overline{48}$

(8) $\dfrac{35}{6} \cdot \dfrac{4}{5}$

(9) $\dfrac{4}{5} \div \dfrac{-7}{8}$

(10) $\dfrac{\frac{3}{14}}{\frac{5}{21}}$

8

Sans utiliser la calculatrice, effectuer et simplifier au maximum les calculs suivants.

Donner des réponses sans exposant négatif ou fractionnaire.

(1) $36^{\frac{3}{2}}$

(2) 4^{-2}

(3) $(0.00032)^{-\frac{1}{5}}$

(4) $\dfrac{(3^2 \cdot 2^4)^2}{(2^2)^3}$

(5) $(-\frac{5}{4})^{-3}$

(6) $\dfrac{1}{(81 \cdot 27^{-2})^2}$

9

Sans utiliser la calculatrice, écrire les nombres suivants en notation scientifique.

(1) $0.000\,274$

(2) $\dfrac{(3 \cdot 10^5) \cdot (2000)^2}{6 \cdot 10^{-5}}$

10

In ein zylinderförmiges Glase werden 0.2 ml der Flüssigkeit A, 2 cm^3 der Flüssigkeit B und 4 μl der Flüssigkeit C gegossen. Der Tachenrechner ist erlaubt.

(1) Wie viel ml Flüssigkeit hat es im Glas ?

(2) Wie viel μl Flüssigkeit hat es im Glas ?

(3) Wie viel cm^3 Flüssigkeit hat es im Glas ?

(4) Wie hoch steht die Flüssigkeit im Glas, wenn es einen Durchmesser von 2 cm hat ?

11

Die absolute Masse eines Protons beträgt $1.67 \cdot 10^{-27}$ kg, die eines Electrons $9.11 \cdot 10^{-31}$ kg. Wie viel mal ist das Proton schwerer als das Elektron ? Der Tachenrechner ist erlaubt.

12

Calculer et simplifier les expressions suivantes sans utiliser la calculatrice.

Des réponses exactes sont exigées.

(1) $\sqrt{50} - 2\sqrt{8} + 3\sqrt{18} - 7\sqrt{2}$

(2) $-\sqrt{\dfrac{28}{5}} \cdot \dfrac{\sqrt{35}}{2}$

(3) $\sqrt[10]{25}$

(4) $\sqrt[5]{8} \cdot \sqrt[5]{4}$

(5) $\dfrac{8 - \sqrt{88}}{12}$

13

Sans utiliser la calculatrice, rendre le dénominateur des expressions suivantes rationnel et simplifier le résulat si possible.

(1) $\dfrac{3}{\sqrt{2}}$

(2) $\dfrac{2\sqrt{2}}{\sqrt{10}}$

(3) $\dfrac{4}{3 - \sqrt{3}}$

(4) $\dfrac{1}{\sqrt[3]{7}}$

14

Sans utiliser la calculatrice, effectuer et simplifier les expressions suivantes au maximum.

(1) $-4^2 + 8^{\frac{4}{3}} \div 4 \cdot 2^2 + 27^{-\frac{2}{3}}$

(2) $0.375 + \left(\dfrac{30}{345 \cdot 2^{-2}}\right)^{-1}$

(3) $\dfrac{25^{-6} \cdot 5^{11}}{(-5^2 \cdot 7)^{-1}}$

(4) $216^3 \cdot (16 \cdot 81)^{-2}$

(5) $\sqrt{24} - \sqrt{15} \cdot \sqrt{10} - \dfrac{18}{-\sqrt{6}}$

(6) $\dfrac{1}{\sqrt{15} - 2} - \dfrac{\sqrt{60}}{22} + 1.\overline{81}$

(7) $\dfrac{\left(1 - \frac{1}{4}\right)^2}{\frac{1}{3} + \frac{1}{60}}$

15

Sans utiliser la calculatrice, déterminer si chacune des affirmations suivantes est vraie ou fausse. Il est possible que une/plusieurs/aucune/toutes les affirmations soient vraies. Aucune justification n'est exigée.

(1) L'inverse du carré de l'opposé de 4 vaut $\frac{1}{16}$.

(2) L'opposé de -2 appartient à $\mathbb{Z}_+ \cap]-3;5[$.

(3) $\sqrt{16} = \pm 4$

(4) $-\frac{5}{51} > -\frac{5}{53}$

(5) $-(-4^{-2}) > 0$

(6) $-2.5 \in]-3;1] \setminus \{-2;-1\}$

(7) Pour tout nombre réel x, on a $\sqrt{x^2} = x$.

Calcul littéral

16

Supprimer les parenthèses inutiles (sans changer les opérations). *Entfernen Sie die überflüssigen Klammern (ohne die Struktur des Terms zu verändern).*

(1) $(4 \cdot x) + 5 \cdot (2 + x)$

(2) $((x+2) \cdot (x-1)) + (3 \cdot x) - (x+2)$

(3) $((x^2) \div (1-x) + (3-x)) - (2-(-x))$

17

Développer puis simplifier les expressions suivantes le plus possible. *Rechnen Sie die folgenden Terme aus, und vereinfachen Sie dann so weit wie möglich.*

(1) $-[4-3(2-c)] - (3c-1)$

(2) $((x^2+1) - 3x(x+2))(x-(x^2+1))$

(3) $a^3 + ax^2 - (x^3 - (a+x)(x^2+a^2))$

18

Ecrire les expressions suivantes à l'aide de radicaux et/ou de puissances entières positives. *Schreiben Sie die folgenden Ausdrücke mit Hilfe von Wurzeln und/oder positiven Exponenten.*

(1) x^{-3}

(2) $x^{\frac{1}{3}}$

(3) $x^{\frac{5}{6}}$

(4) $x^{-\frac{1}{4}}$

19

Ecrire les expressions suivantes à l'aide d'exposants rationnels mais sans radicaux. *Schreiben Sie die folgenden Ausdrücke mit Hilfe von rationalen Exponenten (aber ohne Wurzel).*

(1) $\sqrt[8]{x}$

(2) $\sqrt[3]{x^5}$

∀☐ ∃☐ ∅☐ (3) $\frac{1}{\sqrt{x}}$

∀☐ ∃☐ ∅☐ (4) $\frac{1}{\sqrt[3]{x^5}}$

20

Effectuer puis simplifier les expressions suivantes le plus possible. *Berechnen und vereinfachen Sie die folgenden Terme so weit wie möglich.*

∀☐ ∃☐ ∅☐ (1) $(3y^3)^4 \cdot (4y^2)^{-3}$

∀☐ ∃☐ ∅☐ (2) $\dfrac{\sqrt{x^3}}{\sqrt[3]{x^2}}$

∀☐ ∃☐ ∅☐ (3) $\sqrt{\sqrt[3]{x^6}}$

∀☐ ∃☐ ∅☐ (4) $\left(\dfrac{c^{-5}}{a^{-3}b^9}\right)^{-3} \cdot \left(\dfrac{a^{-2}c^3}{b^{-5}}\right)^{-5}$

∀☐ ∃☐ ∅☐ (5) $\dfrac{9\sqrt[3]{x^2} \cdot \sqrt{x}}{(3x \cdot x^3)^3}$

21

∀☐ ∃☐ ∅☐ Rendre le dénominateur rationnel et simplifier. *Ändern Sie die folgenden Terme so, dass im Nenner keine Wurzelausdrücke vorhandeln sind. Vereinfachen Sie den Term so weit wie möglich.*

$$\frac{8a-4}{\sqrt{4a+7}+3}$$

22

Développer puis simplifier les expressions suivantes le plus possible. *Rechnen Sie die folgenden Terme aus, und vereinfachen Sie dann so weit wie möglich.*

∀☐ ∃☐ ∅☐ (1) $(3a+5bc^2)^2$

∀☐ ∃☐ ∅☐ (2) $(2x-3yz^2)^3$

∀☐ ∃☐ ∅☐ (3) $(x-1)^2-(2-x-2)^2$

∀☐ ∃☐ ∅☐ (4) $(16-4y+y^2)(4+y)$

∀☐ ∃☐ ∅☐ (5) $(2x-1)(1+2x)-(4x-1)^3$

23

Développer les expressions suivantes à l'aide du triangle de Pascal. *Rechnen Sie mit Hilfe des Pascal'schen Dreieicks die folgenden Terme aus.*

∀☐ ∃☐ ∅☐ (1) $(a+b)^5$

∀☐ ∃☐ ∅☐ (2) $(2-3a)^4$

24

Factoriser si possible les expressions suivantes le plus possible. *Faktorisieren Sie falls möglich die folgenden Terme so weit wie möglich.*

∀☐ ∃☐ ∅☐ (1) $15a^2x^3-21ax^5$

∀☐ ∃☐ ∅☐ (2) $(y-3)^2-(3-y)$

∀☐ ∃☐ ∅☐ (3) $2x^3+6x^2-2x-6$

∀☐ ∃☐ ∅☐ (4) $1+x^6-2x^3$

(5) $ab^2 + 12abc + 36ac^2$

(6) $8v^3 + 48v^2w + 96vw^2 + 64w^3$

(7) $8a^3 + 27$

(8) $a^2 - 4a - 5$

(9) $x^2 + 3x + 4$

(10) $6x^2 - x - 35$

(11) $r^3 + 6r^2 + 9r$

(12) $x^3 - 6x^2y + 12xy^2 - 8y^3 - (x - 2y)^2$

25

Simplifier les fractions littérales suivantes le plus possible. *Vereinfachen Sie die folgenden Brüche so weit wie möglich.*

(1) $\dfrac{x^2 - 4x + 4}{16 - 8x}$

(2) $\dfrac{y^2 - 9}{y^3 - 27}$

(3) $\dfrac{6x^2 - 7x - 5}{4x^2 + 4x + 1}$

(4) $\dfrac{3x^2 - 4x + 1}{x^3 - x^2 + 4x - 4}$

(5) $\dfrac{(x - 2)^2 + 3(2 - x)}{x^3 - 8}$

26

Effectuer puis simplifier les fractions littérales suivantes le plus possible. *Rechnen Sie die folgenden Brüche aus, und vereinfachen Sie dann so weit wie möglich.*

(1) $\dfrac{2}{3x^2} + \dfrac{y}{xy} - \dfrac{2}{3y}$

(2) $\dfrac{3a - 5}{a^2 - 1} - \dfrac{4}{a + 1}$

(3) $\dfrac{p^4 - q^4}{q^2 + p^2} \cdot \dfrac{(p - q)^2}{(p - q)^3}$

(4) $\dfrac{x^2 - 1}{x + 2} \div \dfrac{x + 1}{3x + 6}$

(5) $\dfrac{4a}{x^2 - 16} + \dfrac{a}{x^2 - 4x}$

(6) $\dfrac{1}{5a - 5} - \dfrac{6a - 4}{5a^2 - 5} + \dfrac{1}{a + 1}$

(7) $\dfrac{\frac{1}{x+h} - \frac{1}{x}}{h}$

(8) $\left(2x - \dfrac{3}{x}\right)^2 - 3 \cdot \dfrac{6x^2 + 3}{x^2}$

∀ ∃ ∅ ☐ ☐ ☐ (9) $2x\sqrt{x-1} - \dfrac{x^2+3}{\sqrt{x-1}}$

27

∀ ∃ ∅ ☐ ☐ ☐ Effectuer la division euclidienne et donner le reste. *Führen Sie die Polynomdivision mit Rest aus.*

$$\frac{6x^3 - 1 + 5x^2}{2x - 1}$$

28

Welche der folgenden Aussagen sind richtig/falsch ?

∀ ∃ ∅ ☐ ☐ ☐ (1) Beim Addieren und Subtrahieren von Bruchtermen müssen die Nenner zuerst gleichnamig gemacht werden.

∀ ∃ ∅ ☐ ☐ ☐ (2) Bruchterme werden addiert, indem man die beiden Zähler und die beiden Nenner addiert.

∀ ∃ ∅ ☐ ☐ ☐ (3) Beim Dividieren von Bruchtermen wird der Kehrwert des Zählerbruchs mit dem Nennerbruch multipliziert.

∀ ∃ ∅ ☐ ☐ ☐ (4) Die Vereinfachung bei der Multiplikation und Division von Bruchtermen findet häufig durch Kürzen statt, deshalb sollte man Summen in Produkte mit möglichst viele Faktoren verwandeln.

29

Déterminer si les énoncés suivants sont *vrai* ou *faux* tout en donnant une justification.

∀ ∃ ∅ ☐ ☐ ☐ (1) Une puissance de base positive et d'exposant négatif est négative.

∀ ∃ ∅ ☐ ☐ ☐ (2) Diviser deux puissances de même base revient à soustraire les exposants.

∀ ∃ ∅ ☐ ☐ ☐ (3) L'expression $\sqrt{x^2+y^2}$ se simplifie et vaut $x+y$.

∀ ∃ ∅ ☐ ☐ ☐ (4) Un trinôme du second degré ax^2+bx+c se factorise toujours en un produit de deux parenthèses dépendant de x.

∀ ∃ ∅ ☐ ☐ ☐ (5) La mise au même dénominateur lors de l'addition de deux fractions consiste à trouver le plus petit multiple commun des dénominateurs des deux fractions à additionner.

30

Gegeben sind die folgenden Terme.

① $\dfrac{x}{x+y}$	④ $\dfrac{x^2}{x^2-4y^2}$	⑦ $\dfrac{x^2-4xy+3y^2}{x^2-9y^2}$
② $\dfrac{x^2-y^2}{x+y}$	⑤ $\dfrac{x+2y}{x^2+xy-6y^2}$	⑧ $\dfrac{x^2-4y^2}{x^2+2x}$
③ $\dfrac{x^2-4y^2}{x+2y}$	⑥ $\dfrac{x^2}{x+2y}$	⑨ $\dfrac{x^2+4x+4}{x^2+5xy+6y^2}$

∀ ∃ ∅ ☐ ☐ ☐ (1) Wählen Sie zwei Terme aus, so dass die Addtion der Terme möglichst einfach ist. Begründen Sie Ihre Wahl.

∀ ∃ ∅ ☐ ☐ ☐ (2) Wählen Sie zwei Terme aus, so dass die Addtion der Terme möglichst aufwändig ist. Begründen Sie Ihre Wahl.

∀ ∃ ∅ ☐ ☐ ☐ (3) Wählen Sie zwei Terme aus, so dass das Endresultat der Multiplikation möglichst einfach ist. Begründen Sie Ihre Wahl.

Équations et systèmes d'équations

31

Lösen Sie folgende Gleichungen und geben Sie die Lösungsmenge an.

$\forall$ $\exists$ $\varnothing$ (1) $2x - 3 = x - 1$

$\forall$ $\exists$ $\varnothing$ (2) $2x - 3 = 2(x - 1) - 1$

$\forall$ $\exists$ $\varnothing$ (3) $2x - 3 = 2(x - 1) - 2$

32

Lösen Sie folgende Gleichungen und geben Sie die Lösungsmenge an.

$\forall$ $\exists$ $\varnothing$ (1) $3x^2 - x - 2 = 0$

$\forall$ $\exists$ $\varnothing$ (2) $2x^2 - x + 4 = 0$

$\forall$ $\exists$ $\varnothing$ (3) $5x^2 + 20 = 20x$

33

Lösen Sie folgende Gleichungen und geben Sie die Lösungsmenge an.

$\forall$ $\exists$ $\varnothing$ (1) $2x^3 + 54 = 0$

$\forall$ $\exists$ $\varnothing$ (2) $x^6 + 64 = 0$

$\forall$ $\exists$ $\varnothing$ (3) $x^4 - 13x^2 + 36 = 0$

$\forall$ $\exists$ $\varnothing$ (4) $48x^8 + 29x^4 = 2$

$\forall$ $\exists$ $\varnothing$ (5) $7x^3 + 8 = x^6$

$\forall$ $\exists$ $\varnothing$ (6) $x^3 + x^2 = 4x + 4$

$\forall$ $\exists$ $\varnothing$ (7) $x^3 - 6x^2 + 12x = 8$

34

Résoudre dans $\mathbb{R}$ les équations suivantes.

Le domaine de définition et l'ensemble des solutions de l'équation sont exigés.

$\forall$ $\exists$ $\varnothing$ (1) $\dfrac{x-4}{x-3} = \dfrac{2x-4}{6-2x}$

$\forall$ $\exists$ $\varnothing$ (2) $\dfrac{2}{x} + \dfrac{3}{x-2} = \dfrac{x+1}{x^2-2x}$

$\forall$ $\exists$ $\varnothing$ (3) $\dfrac{2}{x-7} + \dfrac{5}{x^2-5x} = \dfrac{x-3}{(x-5)(x-7)}$

$\forall$ $\exists$ $\varnothing$ (4) $4 - \dfrac{8}{x^2+1} = x^2 - 1$

$\forall$ $\exists$ $\varnothing$ (5) $\dfrac{5x^4 - 18x^3 - 8x^2}{x^2 - 5x + 4} = \dfrac{x^3}{x-1}$

35

Résoudre dans $\mathbb{R}$ les équations suivantes.

Le domaine de définition et l'ensemble des solutions de l'équation sont exigés.

$\forall$ $\exists$ $\varnothing$ (1) $\sqrt{6-3x} + x = 2$

$\forall$ $\exists$ $\varnothing$ (2) $\sqrt{6-3x} + 2 = x$

∀ ∃ ∅ ☐ ☐ ☐ (3) $\sqrt{x+2} - \sqrt{x-1} = 1$

36

∀ ∃ ∅ ☐ ☐ ☐ Bestimmen Sie die Lösungsmenge bezüglich x in Abhängigkeit des Parameters k der folgenden Gleichung : $kx = 2x$.

37

∀ ∃ ∅ ☐ ☐ ☐ Déterminer le nombre de solutions de l'équation paramétrique $(m-1)x^2 - 2mx + m + 3 = 0$ en fonction de la valeur de $m \in \mathbb{R}$.

38

Pour chacune des formules suivantes, isoler la lettre indiquée :

∀ ∃ ∅ ☐ ☐ ☐ (1) t dans $v = at + v_0$

∀ ∃ ∅ ☐ ☐ ☐ (2) z dans $g = G\dfrac{m}{(R+z)^2}$ $\quad (R+z>0)$

∀ ∃ ∅ ☐ ☐ ☐ (3) m dans $E = \frac{1}{2}mv^2 + mgh$

∀ ∃ ∅ ☐ ☐ ☐ (4) V_1 dans $\dfrac{V_2 - V_1}{T_2 - T_1} = \gamma V_1$

∀ ∃ ∅ ☐ ☐ ☐ (5) p dans $\dfrac{1}{f} = \dfrac{1}{p} + \dfrac{1}{p'}$

39

Welche der folgenden Aussagen sind falsch ?

∀ ∃ ∅ ☐ ☐ ☐ (1) Addiert man zur rechten Seite einer Gleichung den Term x, dann ist dies eine Äquivalenzumformung.

∀ ∃ ∅ ☐ ☐ ☐ (2) Löst man nur auf der linken Seite einer Gleichung die Klammern auf, dann verändert sich die Lösungsmenge der Gleichung.

∀ ∃ ∅ ☐ ☐ ☐ (3) Quadrieren beider Seiten einer Gleichung ist eine Äquivalenzumformung.

∀ ∃ ∅ ☐ ☐ ☐ (4) Zwei Gleichungen heissen äquivalent, wenn sie die gleiche Lösungsmenge haben.

40

Welche der folgenden Aussagen sind richtig ?

∀ ∃ ∅ ☐ ☐ ☐ (1) Eine reinquadratische Gleichung hat die Form $ax^2 + bx = 0$.

∀ ∃ ∅ ☐ ☐ ☐ (2) Eine quadratische Gleichung heisst auch Gleichung zweiten Grades.

∀ ∃ ∅ ☐ ☐ ☐ (3) Eine quadratische Gleichung hat maximal zwei Lösungen.

∀ ∃ ∅ ☐ ☐ ☐ (4) Bei einer gemischtquadratischen Gleichungen ohne Konstante ist eine Lösung immer gleich Null.

41

Lösen Sie folgende Gleichungssysteme. Geben Sie die Lösungsmenge an.

∀ ∃ ∅ ☐ ☐ ☐ (1)

$$\begin{cases} 3x - 5y &= 27 \\ -x + 2y &= -10 \end{cases}$$

∀ ∃ ∅ ☐ ☐ ☐ (2)

$$\begin{cases} \dfrac{y}{x+1} &= 3 \\[2ex] \dfrac{y+1}{x} &= 5 \end{cases}$$

(3)
$$\begin{cases} xy = 4 \\ 81y^2 - 72 = -x^2 \end{cases}$$

(4)
$$\begin{cases} x - 2y + z = 1 \\ 2x + y + z = 7 \\ -x - 8y + 5z = 9 \end{cases}$$

42

Lösen Sie das Gleichungssystem
$$\begin{cases} 3x - 2y = 17 \\ 6x + 3y = 6 \end{cases}$$
mit Hilfe der ...

(1) ... Einsetzungsmethode.

(2) ... Additionsmethode.

43

Bestimmen Sie, ob das folgende Gleichungssystem genau eine Lösung, unendlich viele Lösungen oder keine Lösung hat. Geben Sie für den Fall, dass das Gleichungssystem genau eine Lösung hat, die Lösungsmenge an.

(1)
$$\begin{cases} 3x - 5y = 27 \\ -15x + 25y = -135 \end{cases}$$

(2)
$$\begin{cases} 3x - 5y = 27 \\ -15x + 25y = 120 \end{cases}$$

(3)
$$\begin{cases} 3x - 5y = 27 \\ 6x + 3y = 6 \end{cases}$$

(4)
$$\begin{cases} x - 2y + z = 1 \\ 2x + y + z = 7 \\ -x - 8y + 5z = 9 \end{cases}$$

(5)
$$\begin{cases} x - 2y + z = 1 \\ 2x + y + z = 7 \\ 7x + y + 4z = 22 \end{cases}$$

(6)
$$\begin{cases} x - 2y + z = 1 \\ 2x + y + z = 7 \\ x - 7y + 2z = -5 \end{cases}$$

44

Stellen Sie ein lineares Gleichungssystem mit zwei Gleichungen und zwei Unbekannten auf, welches unendlich viele Lösungen hat.

45

Stellen Sie ein Gleichungssystem mit den Unbekannten x, y und z auf, das die Lösung $x = 2$, $y = -1$ und $z = 4$ hat. Wie viele solcher Gleichunssysteme gibt es?

Problèmes

46 ∀ ∃ ∅

Die Fruchtsäfte A und B sollen gemischt werden. A enthält einen Fruchtanteil von 25% und B einen von 60%. Durch Mischung dieser Fruchtsäfte sollen 20 Liter Fruchtsaft entstehen, dessen Fruchtanteil 50% beträgt. Wie viel Liter muss man von A und B nehmen ?

47 ∀ ∃ ∅

Une dame commence son jogging à 15 heures et court plein nord à l'allure de 16 km/h. Plus tard, elle change de direction et court plein sud à l'allure de 14 km/h. Calculer la distance totale parcourue par cette dame sachant qu'elle revient à son point de départ à 15h45.

48 ∀ ∃ ∅

Ein gleichschenkliges Trapez mit der Höhe $h = 4\,\text{cm}$ hat ein Umfang von $U = 28\,\text{cm}$ und eine Fläche von $A = 36\,\text{cm}^2$. Berechnen Sie die untere und obere Seite des Trapezes.

49 ∀ ∃ ∅

Bei einem Rechteck ergeben die Breite b und die Diagonale d zusammen 41 cm. Die Länge l des Rechtecks beträgt 27.6 cm. Berechnen Sie den Flächeninhalt A des Rechtecks.

50 ∀ ∃ ∅

Une somme de 24000 francs doit être distribuée à parts égales entre un certain nombre de personnes. Au moment du partage, cinq personnes se retirent, ce qui augmente la part de chacun des autres de 240 francs. Combien de personnes devaient participer initialement au partage ?

51 ∀ ∃ ∅

Une place de sport rectangulaire de 448 m^2 est bordée d'un champ sur trois côtés et d'une rivière sur le quatrième. Pour entourer la place de sport par un grillage, on a utilisé un grillage d'une hauteur de 2.5 mètres à 10 francs le mètre courant [1] pour les trois côtés bordant le champ et un grillage d'une hauteur de 4 mètres à 20 francs le mètre courant pour le côté bordant la rivière. En sachant que le grillage a coûté au total 1160 francs, quelle est la longueur du côté de la place bordant la rivière ?

La solution contiendra une définition claire et précise de chaque inconnue utilisée ainsi que le domaine de définition relatif à chaque inconnue. Les réponses données doivent être exactes et formulées à l'aide d'une phrase.

52 ∀ ∃ ∅

Une entreprise a fabriqué 89 jouets en bois, dont certains sont des camions, d'autres des pantins et le reste des puzzles. Chaque camion a nécéssité 2 kg de bois et trois heures de travail, chaque pantin 500 g de bois et quatre heures de travail et chaque puzzle 800 g de bois et trois heures et trente minutes de travail. Sachant que l'ensemble de la production des jouets a duré 313 heures et que 91 kg de bois ont été utilisés, déterminer le nombre de jouets de chaque type fabriqués.

1. Le mètre courant est la mesure de quelque chose par mètres, en longueur, sans avoir égard à la hauteur.