Bataille navale linéaire

Source: `Bataille-navale-lineaire.tex`.

Version imprimable: `Bataille-navale-lineaire.pdf`

Algèbre linéaire. DEUG première année. Angle pédagogique : *Ludique.*

Objectifs et commentaires. *Dans cette activité, on utilise un jeu de type "bataille navale" pour amener les étudiants à se poser des questions sur la géométrie des sous-espaces vectoriels, et plus particulièrement sur la dimension.*

Réalisation pratique *Notons d'abord que contrairement à ce qu'on peut penser au premier abord, le jeu ne nécessite pas beaucoup de calculs, puisqu'on peut répondre à la plupart des questions à l'aide d'arguments de dimension : évidemment, l'un des buts de l'exercice est d'inciter les étudiants à utiliser ces raisonnements.*

On suggère de grouper les étudiants par quatre, et de les faire jouer deux contre deux (et éventuellement plus tard quatre contre quatre). Par rapport à un affrontement "un contre un", ces configuration ont l'avantage de favoriser la confrontation des idées : le fait de ne pas être d'accord sur le coup à jouer devrait obliger les étudiants à justifier leurs propositions, donc à jouer plus rationnellement, voire à chercher une bonne stratégie. Et aussi, ceci diminue la fréquence des erreurs...

Il vaut probablement mieux commencer par jouer dans $\mathbb{R}^3$, où l'intuition aide beaucoup. Mais il serait dommage de ne pas jouer aussi dans $\mathbb{R}^4$, où l'étudiant doit justement remplacer son intuition défaillante par les raisonnements sur la dimension.

On peut préciser les règles au fur et à mesure des parties ; on suggère les règles additionnelles suivantes :

- *un joueur qui détecte une erreur dans la réponse de son adversaire à l'une de ses questions gagne la partie (y compris si la détection intervient longtemps après l'erreur) ;*
- *le joueur qui se trompe en annonçant la dimension perd la partie ;*
- *un joueur est autorisé à modifier son sous-espace vectoriel secret en cours de partie (y compris après que l'autre ait proposé une dimension) à condition toutefois que son nouveau sous-espace secret reste compatible avec ses réponses précédentes. L'intérêt de cette règle est d'obliger le joueur qui prétend avoir trouvé à être sûr de lui (et donc à chercher à démontrer que la dimension est ce qu'il pense). On pourrait aussi demander explicitement au gagnant, à la fin de la partie, de fournir une preuve que la dimension est ce qu'il affirme (mais ceci risque de rendre l'activité plus scolaire et moins ludique, ce qui est dommage).*

Complexité algorithmique *Après un peu de pratique, on peut suggérer aux étudiants de chercher un algorithme, par exemple en leur demandant comment on pourrait programmer un ordinateur pour qu'il trouve la réponse en temps fini dans tous les cas. Ceci est probablement assez difficile en DEUG, sans parler de la recherche d'une stratégie optimale : on peut ainsi transformer le jeu en un exercice de niveau licence... Voici des éléments de réponses (en note de bas de page pour ne pas contrarier le lecteur qui voudrait y réfléchir tout seul !).* [1]

[1] **Quelques remarques sur la complexité** Pour un champ de bataille E de dimension n, quel est le nombre de coups nécessaires pour répondre dans tous les cas ? Voici une stratégie qui marche toujours en moins de n coups.

Ce jeu est inspiré d'une bataille navale géométrique (non linéaire) proposée par Nicolas Bouleau dans le texte de sa conférence au colloque EM2000 sur l'enseignement des maths, Y a-t-il lieu d'envisager des mathématiques post-modernes ? *(http://em2000.imag.fr/Actes/). Citons le texte d'origine, qui peut inspirer d'autres exercices :*

"... Chaque élève d'un binôme définit une figure géométrique (par exemple constituée d'un cercle et de deux droites) dans un système de coordonnées. Pour deviner la figure de son adversaire, il tire des droites et non des points. Si à son tour de jouer, il propose ainsi la droite $y = 2x + 1$, son adversaire lui indique tous les points d'intersection de sa figure avec $y = 2x + 1$, etc. Il y a de multiples variantes suivant les figures cachées et les objets qu'on tire, la déduction et la combinatoire ne sont pas absentes de ce jeu qui peut s'organiser en tournoi comme les échecs."

La bataille navale linéaire est un jeu qui se joue à deux joueurs, dans lequel chacun doit découvrir la dimension d'un sous-espace vectoriel caché par l'autre.

Expliquons les règles en détail. On décide d'abord du champ de bataille : il s'agit d'un espace vectoriel E de dimension finie (par exemple $\mathbb{R}^3$ ou $\mathbb{R}^4$). Chaque joueur choisit en secret un sous-espace vectoriel de E (on notera E_1 et E_2 ces deux sous-espaces). Commence alors la phase de jeu proprement dite : le premier joueur suggère un sous-espace vectoriel F de E ; le second lui répond

- "touché" si le sous-espace proposé a une intersection non triviale avec le sous-espace caché (autrement dit, si $F \cap E_2$ n'est pas réduit à $\{0\}$) ;
- "dans l'eau" sinon.

C'est alors au tour du second de faire une proposition, et le jeu continue ainsi jusqu'à ce que l'un des deux joueurs trouve la **dimension** du sous-espace choisi par l'autre.

On précise que chaque joueur est libre de choisir son sous-espace secret et les sous-espaces qu'il propose de la manière qui lui convient (par exemple, à l'aide d'un système d'équations cartésiennes ou bien d'une base).

On choisit une base $B = \{e_1, .., e_n\}$. On regarde les parties F contenues dans B telles que $\text{vect}(F) \cap E_1 = \{0\}$ (où E_1 est l'espace inconnu). On note $\mathcal{F}$ l'ensemble de ces parties.

Lemme *Une partie maximale dans $\mathcal{F}$ est de dimension $n - \dim(E_1)$.*

Donc on cherche une partie maximale. L'algorithme est le suivant. On pose la question "$F = \text{Vect}(e_1)$?" Si la réponse est "à l'eau", on prend $F_1 = \{e_1\}$; si c'est "touché", on prend $F_1 = \emptyset$. Puis on continue : à l'etape i, on pose la question "$F = vect(F_{i-1} \cup \{e_i\})$?", et on prend $F_i = F$ si "à l'eau", $F_i = F_{i-1}$ si "touché". Le n-ième ensemble F_n est maximal : en effet, le i-ième ensemble F_i est maximal pour l'ajout d'un des i premiers vecteurs (*i. e.* si on rajoute un des i premiers vecteurs à F_i, on touche E_1).

Remarquons aussi la différence entre le "presque sûr" et le "sûr" : si on fait une dichotomie sur la dimension (en proposant d'abord un espace de dimension $n/2$, *etc.*), on va trouver la dimension en temps $\log(n)$ pour presque tout sous-espace ; la complexité presque sure est en $\log(n)$, alors que c'est beaucoup moins évident pour la complexité "au pire".

Question *Y a-t-il un algorithme meilleur que celui proposé au-dessus ? Autrement dit, quelle est la complexité au pire ?*