

# LABORATOIRE D'INFORMATIQUE FONDAMENTALE

PROJET TER  
MASTER 1 INFORMATIQUE  
Rapport

---

## Polynôme de Tutte

---

**RÉALISÉ PAR :**  
M.AMARA Sofiane

**ÉNCADRÉ PAR :**  
M. PERROT Kevin

# Sommaire

INTRODUCTION.....	3
PRÉSENTATION DU SUJET.....	3
DÉFINITION PRÉLIMINAIRE.....	4
Graphe.....	4
CONNEXITÉ SUR LES GRAPHS .....	4
POLYNÔME CHROMATIQUE .....	5
POLYNÔME DE TUTTE .....	6
DESCRIPTION DU TRAVAIL RÉALISÉ .....	7
INTRODUCTION.....	7
Langages et choix des Framework utilisés .....	7
Architecture de l'éditeur de graphes .....	8
Module Traitement Graphe.....	8
Applications .....	9
Choix des sous-ensembles.....	9
Calcule des composants connexes.....	10
Traitement et création des Polynôme .....	10
Stockage .....	11
Lecture et affichage Polynôme de tutte.....	12
Interface Graphique .....	12
Gestion du projet .....	14
Méthodes Scrum .....	14
CONCLUSION .....	14
Bibliographie.....	15

## INTRODUCTION

Les graphes sont considérés comme un outil très puissant de modélisation des problèmes complexes qui trouvent leurs applications dans plusieurs domaines, tels que l'informatique, le traitement d'images, la recherche opérationnelle, la biologie, la chimie, l'écologie et autres.

La théorie des graphes est la branche des mathématiques et d'informatique théorique qui, en développement permanent, permet d'une part, l'étude d'un graphe en tant qu'une structure mathématique afin de connaître ses propriétés et d'autre part, de proposer de nouveaux algorithmes pour la résolution des problèmes réels.

De ce fait, un environnement informatique dédié aux graphes peut s'avérer d'une très grande utilité pour les chercheurs. Un tel outil, nommé éditeur de graphes ou graph software, permet non seulement la création et la visualisation des graphes, mais aussi leurs traitement en un temps beaucoup plus réduit, ce qui permettra aux spécialistes de confirmer ou d'infirmer des conjectures ou encore de vérifier des propriétés sur des graphes de plus grand ordre.

## PRÉSENTATION DU SUJET

Ce logiciel représente un utilitaire pour les traitements des graphes non orientés, ainsi l'interprétation graphique en 3D de polynôme de toute obtenue suite à une série de calculs de données du graphe introduit.

Pour entamer la description en détail de l'application nous remarquons qu'un scénario type d'exécution du logiciel se décompose en quatre phases :

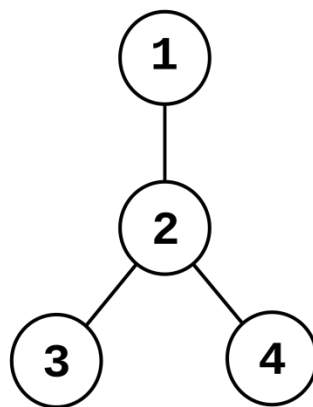
- Une phase de définition d'un graphe en format dot.
- Une phase de visualisation de polynôme, (analyse syntaxique).
- Une phase d'interprétation de définition récursive de polynôme sur un graphe (calcul).
- Une phase d'analyse et de manipulation de polynôme obtenu.

## DÉFINITION PRÉLIMINAIRE

### Graphe

Un **graphe non orienté** est un couple  $(V, E)$  où :

- $V$  est appelé l'ensemble des **sommets** de  $G$ , et
- $E$  appartient à  $\{\{a, b\} \mid a, b \in V\}$  est un ensemble de paires d'éléments de  $V$  appelé l'ensemble des **arêtes**.



Exemple : graphe simple

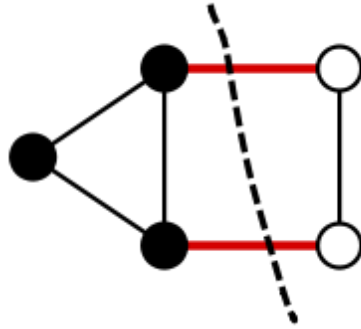
Un tel graphe est aussi appelé **simple** pour le distinguer des multi graphes, construction où il peut exister plusieurs arêtes pour une même paire de sommets. La notation  $E$  est fréquente par emprunt à l'anglais, où elle dénote l'ensemble des « edges ».

Le nombre d'arêtes issues d'un sommet est le **degré** de ce sommet.

## CONNEXITÉ SUR LES GRAPHS

Un graphe non orienté  $G=(S, A)$  est dit *connexe* si quels que soient les sommets  $u$  et  $v$  de  $S$ , il existe un chemin reliant  $u$  à  $v$ .

-Un sous-graphe connexe maximal d'un graphe non orienté quelconque est une *composante connexe* de ce graphe.



## Sous-graphe

En théorie des graphes, un **sous-graphe** est un graphe contenu dans un autre graphe. Formellement, un graphe  $H = (V_H, E_H)$  est un sous-graphe de  $G = (V_G, E_G)$  si

$$V_H \subseteq V_G \text{ et } E_H \subseteq \{(x, y) \in E_G \mid x \in V_H \wedge y \in V_H\}$$

L'ensemble des sommets du sous-graphe **H** est un sous-ensemble de l'ensemble des sommets de **G** et l'ensemble des arcs de **H** est un sous-ensemble de l'ensemble des arcs de **G** ayant leur origine et leur extrémité parmi les sommets de **H**.

## POLYNÔME CHROMATIQUE

Pour  $\lambda=0$ , le polynôme de Tutte se spécialise en le polynôme chromatique :

$$\chi_G(\lambda) = (-1)^{|V|-k(G)} \lambda^{k(G)} T_G(1 - \lambda, 0),$$

Où  $k(G)$  est le nombre de composantes connexes de  $G$

Pour des valeurs entières de  $\lambda$ , la valeur du polynôme chromatique  $\chi_G(\lambda)$  est égale au nombre de colorations du graphe  $G$  en utilisant  $\lambda$  couleurs. Clairement, le  $\chi_G(\lambda)$  ne dépend pas de l'ensemble de couleurs. Ce qui est moins clair, c'est que c'est la valeur en  $\lambda$  d'un polynôme à coefficients entiers. Pour voir cela, on observe que : résidence

1. si  $G$  a  $n$  sommets et pas d'arêtes, alors  $\chi_G(\lambda) = \lambda^n$ .
2. si  $G$  contient une boucle, alors  $\chi_G(\lambda)=0$ .
3. si  $e$  est une arête qui n'est pas une boucle, alors

$$\chi_G(\lambda) = \chi_{G \setminus e}(\lambda) - \chi_{G/e}(\lambda).$$

Ces trois conditions permettent de calculer  $\chi_G(\lambda)$  en appliquant une suite de contractions-suppressions, mais elles ne garantissent pas que des séquences différentes de suppressions et contractions conduisent au même résultat ; ceci provient du fait que  $\chi_G(\lambda)$  compte des propriétés combinatoires, indépendamment de la récurrence. En particulier,

$$T_G(2, 0) = (-1)^{|V|} \chi_G(-1)$$

Donne le nombre d'orientations acycliques du graphe.

## POLYNÔME DE TUTTE

Le polynôme de Tutte, aussi appelé polynôme dichromatique ou polynôme de Tutte–Whitney, est un polynôme invariant de graphes dont les valeurs expriment des propriétés d'un graphe. C'est un polynôme en deux variables qui joue un rôle important en théorie des graphes et en combinatoire. Il est défini pour tout graphe non orienté  $G$  et contient des informations liées à ses propriétés de connexité.

**Définition [Whitney - Tutte] :** Le Polynôme de Tutte est défini par une sommation sur les sous-graphes couvrants

-Le polynôme de Tutte peut s'exprimer grâce à la fonction suivante:

$$T_G(x, y) = \sum_{A \subseteq E} (x - 1)^{k(A) - k(E)} (y - 1)^{k(A) + |A| - |V|},$$

$K(A)$  : le nombre de composantes connexes de  $A$ .

- $K(E)$  : le nombre de composantes connexes de  $E$ .
- $|A|$  : nombre d'arêtes de sous-ensemble  $E$
- $|V|$  : nombre de sommets

# DESCRIPTION DU TRAVAIL RÉALISÉ

## INTRODUCTION

Après avoir présenté, dans le chapitre précédent, les différents concepts nécessaires pour la bonne compréhension et la réalisation de notre travail, nous passons à l'étape de conception du noyau de l'éditeur et du module Application définies sur les arêtes qui fait l'objet de notre étude.

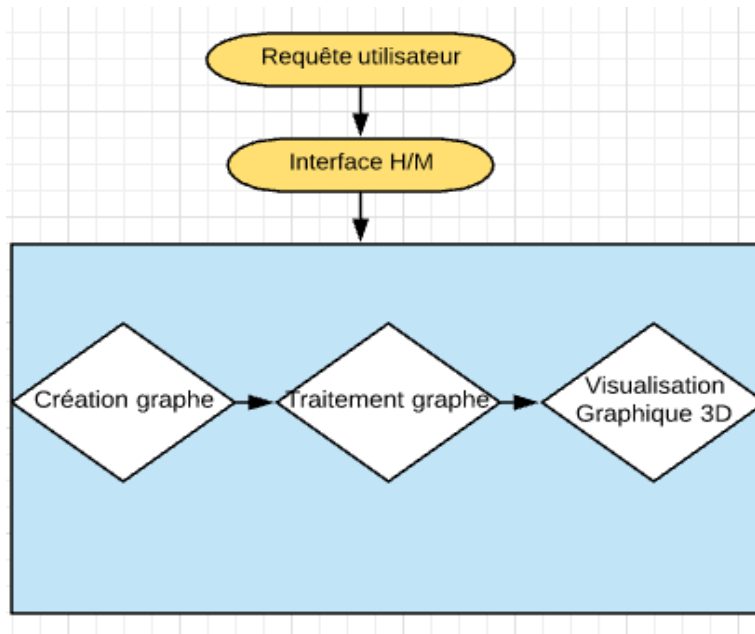
## Langages et choix des Framework utilisés

- ❖ **HTML/CSS** Le HTML et le CSS sont des langages informatiques complémentaires. Alors que le HTML structure la page, le CSS vas la mettre en forme en y apportant du style.
- ❖ **JavaScript** est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation (par exemple) de Node.js.
- ❖ **Le langage DOT** est un langage de description de graphe dans un format texte. Il fait partie de l'ensemble d'outils open source Graphviz créés par les laboratoires de recherche d'AT&T.
- ❖ **Graphviz (diminutif de *Graph Visualization Software*)** est un ensemble d'outils *open source* créés par les laboratoires de recherche d'AT&T qui manipulent des graphes définis à l'aide de scripts suivant le langage DOT. Cet ensemble fournit aussi des bibliothèques permettant l'intégration de ces outils dans diverses applications logicielles.
- ❖ **D3.js (ou D3 pour Data-Driven Documents)** est une bibliothèque graphique JavaScript qui permet l'affichage de données numériques sous une forme graphique et dynamique. Il s'agit d'un outil important pour la conformation aux normes W3C qui utilise les technologies courantes SVG, JavaScript et CSS pour la visualisation des données. D3 est le successeur officiel du précédent Framework Protovis.

## Architecture de l'éditeur de graphes

Comme nous l'avons déjà mentionné, nous présentons la conception générale de l'éditeur de graphes.

L'architecture du système est représentée comme suite :



**FIGURE 01 : ARCHITECTURE LOGICIEL**

## Module Traitement Graphe

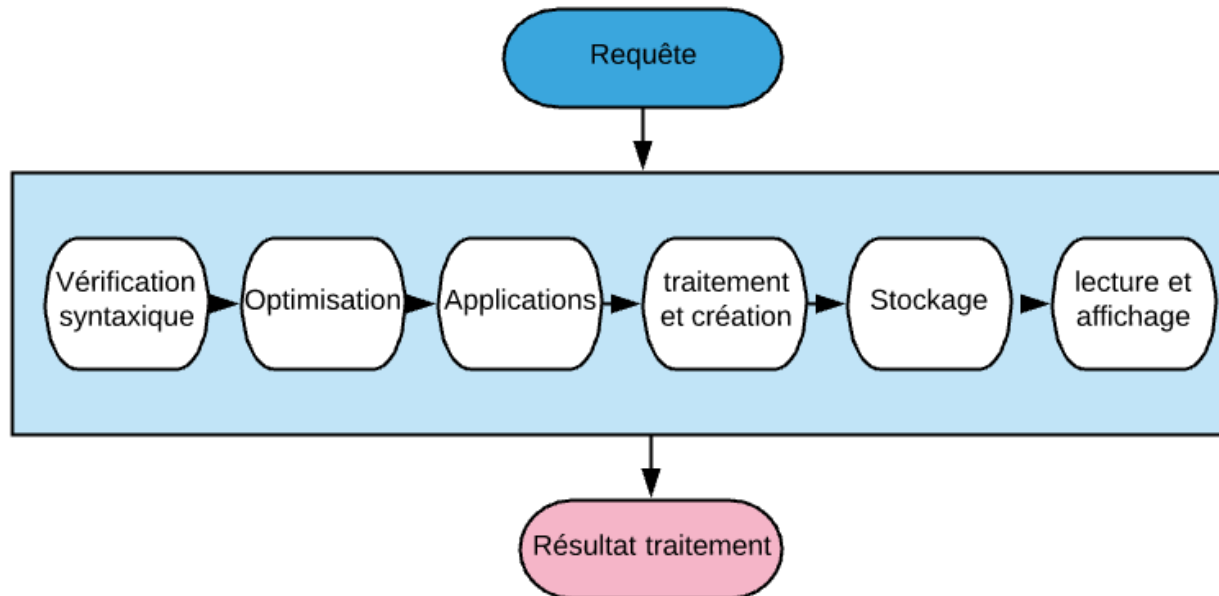
Ce module permet de manipuler les graphes en effectuant les différentes opérations et les traitements qui permettent de les étudier en offrant une palette de fonctionnalités telles que, la vérification des propriétés Syntaxique du format DOT introduit, le calcul des propriétés des sous ensembles d'arêtes, la recherche des composants connexes pour chaque s-ensemble, et bien d'autres fonctionnalités qu'on explicitera tout au long de ce chapitre, les objets manipulables dans notre application sont les ensemble d'arêtes ou d'arcs.

Les composants modulaires de traitement de graphe sont:

- Module Vérification propriétés syntaxique.
- Module Optimisation des données.
- Module Applications
- Module de traitement et création.



- Module Stockage.
- Module lecture et affichage finale.



**-Figure 02: Module Traitement-**

Pour notre partie de travail, nous nous intéressons de manière particulière à tous les modules sauf La vérification et optimisation.

## Applications

### Choix des sous-ensembles

Pour un ensemble d'arêtes  $E$  de graphe  $G$  de sommet  $V$ , il existe ( $2^{\text{nombre d'arêtes}}$ ) graphe partiel  $A$  de  $G$  qui partageant les mêmes sommets  $V$  et des arêtes différentes, ces graph partiel on les nomme sous graph couvrant de  $G$  et leur sommation en théorie définit la fonction de polynôme tutte.

La technique de programmation utiliser pour l'extraction des sous ensemble existant ce base sur les représentations binaire des arêtes, le mécanisme s'annonce a convertir tout les entier naturelle (décimal) appartenant a l'intervalle  $[0 : 2^{\text{nombre d'arêtes}}]$  en binaire et pour chaque étude octale de la suite binaire le cas ou l'octet vaut « 1 » l'arête correspondante sera ajouté sur le s-e sinon on rajoute rien.

## Calcule des composants connexes

Pour chaque sous-ensemble on extrait les sommets responsable des arêtes du s-e et on utilisant des technique de propagation sur le graphe nous conduit à connaître et a stocker le nombre de composante connexe.

## Traitement et création des Polynôme

**Création** : chaque sous-ensemble prêtant avoir une fonction de format suivant, on prenant compte que les puissances de cette formule sont déjà acquis par le module précédent

$$\bar{v}(x-1)^{k(A)-k(E)}(y-1)^{k(A)+|A|-|V|}$$

Afin de simplifier on s'engage à calculer sa notation abrégée standard pour chaque formule séparément on respectant la règle suivant :

$$(a+b)^n = \sum_{k=0}^n \binom{n}{k} \cdot a^{n-k} \cdot b^k$$

La signification du coefficient monotone est le suivant :

$$\binom{n}{k} = C_n^k = \frac{n!}{k! (n-k)!}$$

Au final on aura le résultat suivant :

$$(a+b)^n = a^n + C_n^1 a^{n-1} \cdot b + C_n^2 a^{n-2} \cdot b^2 + \dots \\ + C_n^{n-2} a^2 \cdot b^{n-2} + C_n^{n-1} a \cdot b^{n-1} + b^n$$

## Stockage

Dans cette partie on s'intéresse à stocker tous les coefficients binomiaux précédents sur une unique matrice, dans le but de combler et additionner la totalité des données, pour pouvoir calculer le polynôme final de tout à partir de cette matrice.

Information générale de la matrice : la 1<sup>ère</sup> ligne contient les puissances de x variant de 0 à n et la 1<sup>ère</sup> colonne contient les puissances de y variant de 0 à n et le contenu de la cellule représente la valeur du coefficient binomial d'une clause.

Exemple pour mieux comprendre le principe de stockage :

On va essayer de stocker la fonction suivante dans la matrice

$$F(x,y) = x^8 + 4xy^{(7,6)} - 35xy^{(3,2)} + 5y^4 - 1$$

-1	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0
0	0	0	-35	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	4	0	0
0	0	0	0	0	0	0	0	0	0

On se procède avec le même mécanisme sur toutes les fonctions, additionnant les sur la même matrice jusqu'à leurs terminaisons, la dernière matrice obtenue représente les données du polynôme de tout

## Lecture et affichage Polynôme de tutte

La lecture ça se fait sur la dernière version de la matrice, l'extraction du polynôme se fait avec le même mécanisme de stockage sous traitement syntaxique.

## Interface Graphique

Dans cette partie, nous allons parler de notre interface graphique, de ses particularités et de la façon qu'on l'utilise.

Nous avons essayé de faire quelque chose de simple et efficace, pour permettre à tous les utilisateurs potentiels de bien l'appréhender.

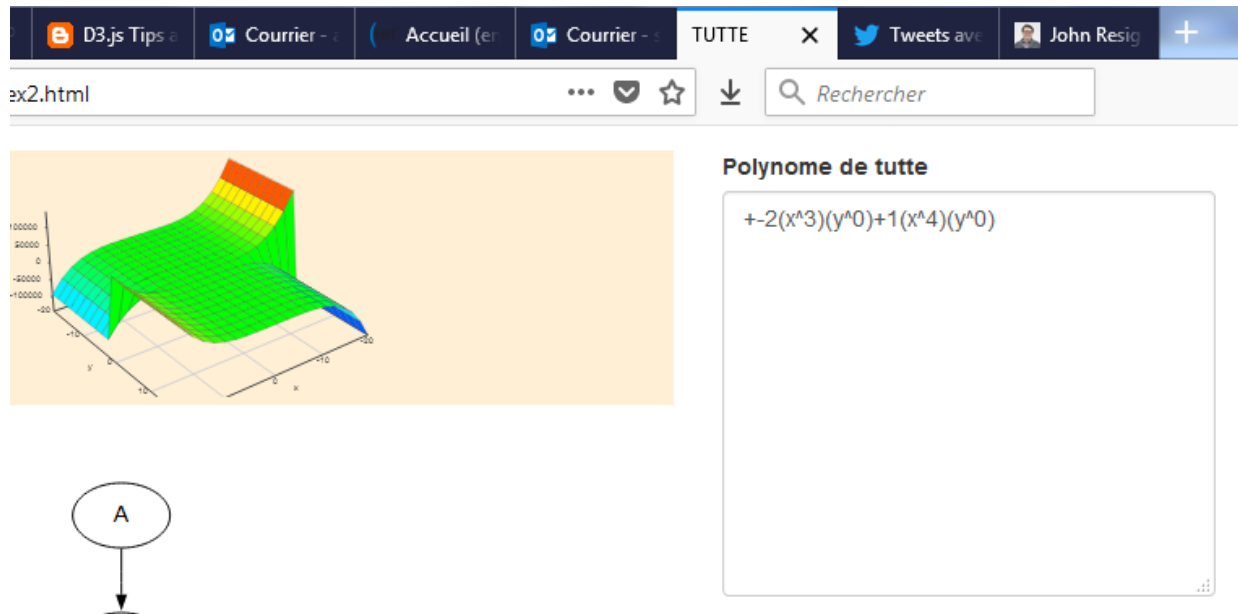
Nous allons commencer par l'éditeur de graphe, la lecture se fait par une syntaxe texte précise un modèle est visible sur la zone de texte en cas d'erreur le parseur indique et identifie l'erreur on l'affichant sous forme d'une alerte. Un bouton de validation sert à dessiner le graphe correspondant et compatible.

The screenshot shows a web browser window with the address bar displaying `file:///C:/wamp/www/test/liviz/index2.html`. The page has a dark header with social media links (Facebook, traducteur, D3.js Anim, CSS backg, D3.js the P, D3.js Tips, and a chat icon). The main content area is titled "Introduisez votre graphe" with the subtitle "Au format dot". Below this is a text input field containing the following DOT syntax:

```
digraph testgraph{  
  A -> B;  
  B -> D;  
  D -> C;  
}
```

Below the input field are two buttons: "Poly Tutte" and "graph schema". To the right of the input field is a large green rectangular area with a black grid pattern. Below the grid is a vertical flow diagram with four nodes labeled A, B, D, and C, connected by downward arrows in that sequence.

La génération graphique sur dimension 3 du polynôme de tutte résultant du graphe introduit est prit en compte par le Framework D3.js que on a expliqué auparavant, l'affichage se fait on cliquant sur le buttons « poly tutte » en parallèle la fonction tutte apparait sur la gauche de l'écran.



Ainsi que nous avons implémenté d'autre fonctionnalité qui sont utiles pour des clarifications concernant la représentation graphique :

- Possibilité de draguer la vue du graphe sur les 4 sens de l'écran.
- Limité l'intervalle de la rangé **x** et **y** sur le graphe 3D.
- Possibilité de gérer les points à tracé sur ce graphe.

Interval de X



Interval de Y



Nombre de points à tracer

Graph 3D

# Gestion du projet

## *Méthodes Scrum*

Dans cette partie je vais aborder les méthodologies de travail mis en place lors de mon projet encadrer TER.

La méthode Scrum fait partie de la famille des méthodes dites «agile». L'approche agile consiste à se donner des objectifs à courts termes, une fois l'objectif terminé nous faisons le point et suivant le résultat nous adapte les nouveaux objectifs en fonctions du résultat obtenue précédemment et ainsi de suite jusqu'à atteindre le résultat final. Le commanditaire est impliqué dans le projet du début à la fin ce qui lui donne de la visibilité.

Cela permet en outre d'éviter «l'effet tunnel», c'est-à-dire se lancer sur un projet et arriver à terme avec un produit qui ne correspond pas aux attentes des clients ou ne pas mener le projet à terme. La méthodologie Scrum est l'approche agile la plus utilisée des approches agiles existantes et est simple à comprendre.

## CONCLUSION

Pour conclure, ce projet nous a permis de réaliser un certain nombre de services, d'acquérir des compétences en développement web (JavaScript, CSS, Framework D3 et graphviz et HTML), mais aussi des compétences en théorie des graphes et de la recherche en informatique, et surtout de regrouper ces deux compétences en un seul projet. Ça n'a pas été tout le temps simple, le projet a été très dur par moment surtout durant le calcul de l'équation finale de tout où l'on a dû beaucoup travailler durant notre temps libre, pour essayer de rendre un bon projet, ce que nous pensons être le cas.

## Bibliographie

- [1] \_ <http://villemin.gerard.free.fr/Wwwgvm/Identite/IdentAut.htm>
- [2] \_ <http://people.brandeis.edu/~bernardi/slides/transparents-plongement-Tutte-LaCIM.pdf>
- [3] \_ <http://bl.ocks.org/rkirsling/5001347>
- [4] \_ <http://www.toutjavascript.com/savoir/savoir08.php3>
- [5] \_ <https://github.com/davidpiegza/Graph-Visualization>
- [6] \_ <https://github.com/dzigerlig/CDAR>
- [7] \_ [https://fr.wikipedia.org/wiki/Polyn%C3%B4me\\_de\\_Tutte](https://fr.wikipedia.org/wiki/Polyn%C3%B4me_de_Tutte)
- [8] \_ <https://github.com/AlexOuyang/3DGrapher>
- [9] \_ <https://www.apprendre-en-ligne.net/graphes/graphes.pdf>
- [10] \_ <https://visionscarto.net/ode-aux-projections-de-d3-js>
- [11] \_ <https://stackoverflow.com/questions/6344318/pure-javascript-graphviz-equivalent>