

Projet Semestre Atelier G- Code

Collaboration avec C-
S Communication

Introduction

➤ Contexte du projet :

Il s'agit de rendre les procédures d'usinage d'une pièce avec des machines à commande numérique et plus particulièrement la réalisation du programme de commande (dit "G-code") accessibles au plus grand nombre.

➤ Mission du projet :

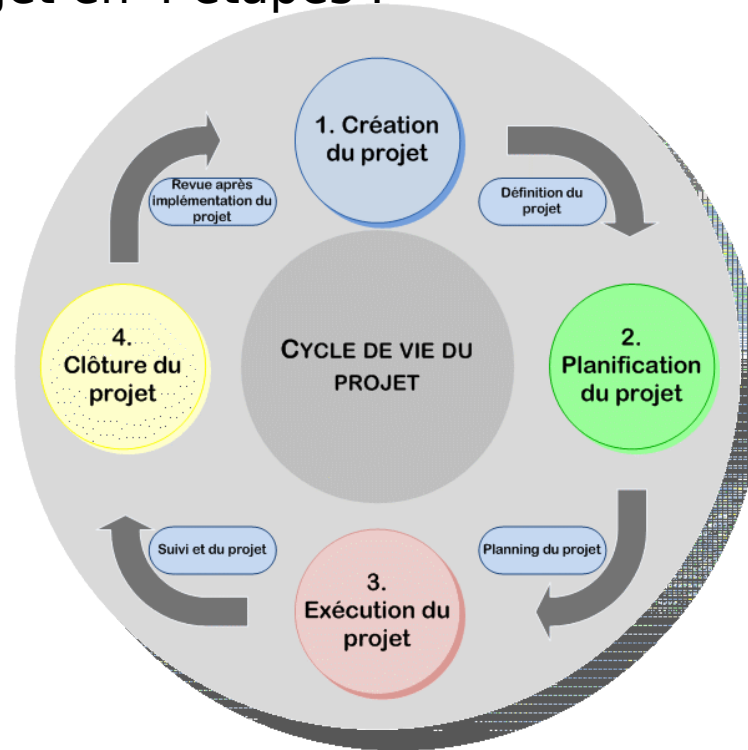
*Concevoir un **éditeur** permettant de vérifier et corriger un programme G-code donné, ainsi qu'une **aide contextuelle** se basant sur les **connaissances en usinage** afin d'aider le programmeur à éditer/écrire son programme en restant cohérent avec le modèle de conception voulu et les propriétés mécaniques du matériau utilisé.*

Plan de la présentation

- I. Project Management Lifecycle
- II. Programmation de commande numérique
- III. Base de connaissances en usinage matériau/outil
- IV. Environnement de programmation
- V. Perspectives d'évolution du projet (simulation, pilotage par rétroaction)

I. Project Management Lifecycle

- Gestion du projet en 4 étapes :



I. Project Management Lifecycle

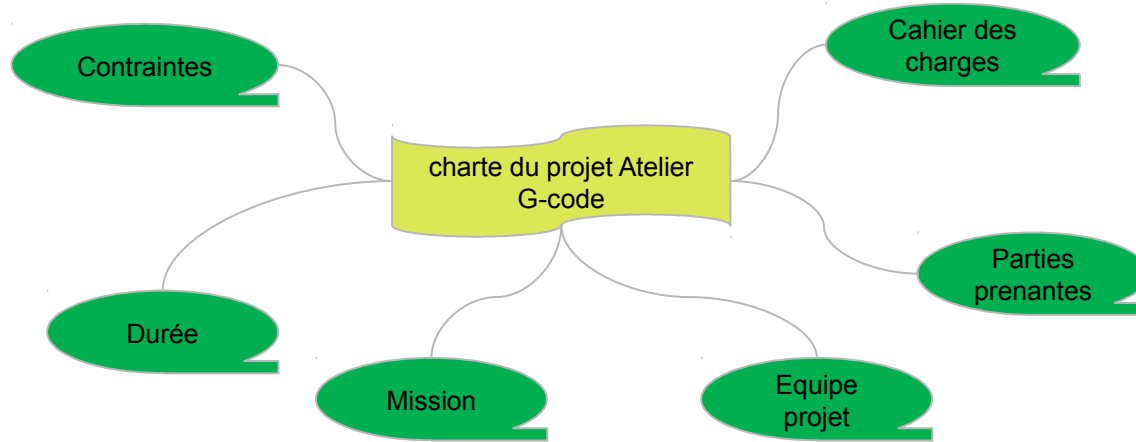
1) Création du projet :

➤ Charte du projet :

- Différentes variantes de G-Code
- Programme Open Source
- Difficile d'être compatible avec tous les interpréteurs qui existent
- Connaissances d'usinage difficiles à trouver

Du 23-Octobre-2017

Au 15 Avril



Créer un environnement de développement qui aide à la programmation du G-Code

Tétar Adrien :

Chef de projet/Responsable développement informatique

Pliya Tatiana & Sbitri Youssra : Co-Responsables gestion de projet/Consultants en usinage

- Editeur
- Coloration syntaxique
- Base de connaissance usinage
- Aide contextuelle
- Interpréteur de langage G-Code

Expert Cs Com :

M. Felecan Peter

Superviseur ESIEE :

M.Hadj Daoud Mousselmal

Equipe projet :

Étudiants de l'ESIEE

I. Project Management Lifecycle

2) Planification du projet :

➤ Gantt :

tâche 1 : Réunion d'introduction
tâche 2 : Réunion avec l'entreprise et le partage
des rôles dans l'équipe
tâche 3 : Documentation (G-code, machines
cartésiennes...)
tâche 4 : Rédaction du cahier des charges
tâche 5 : Rédaction de la charte de projet
tâche 6 : Création d'un environnement de
développement(éditeur+parseur)
tâche 7 : Création d'une BD(matériaux,
propriétés, outils, machines ..)
tâche 8 : Création d'un projet PLM
tâche 9 : Création d'une aide contextuelle en
se basant sur une BD d'usinage
tâche 10 : Réunion de mise au point
équipe/tuteur/industriel
tâche 11 : Rédaction du rapport projet
tâche 12 : Finalisation des livrables
tâche 13 : Réunion finale équipe / industriel
tâche 14 : Finalisation du PLM
tâche 15 : Préparation de la présentation



I. Project Management Lifecycle



2) Planification du projet :

➤ Partage des tâches :

Tétar Adrien	Documentation + Rédaction du cahier des charges + Création d'un environnement de développement + Aide contextuelle + Rédaction rapport
Sbitri Youssra	Documentation + Rédaction du cahier des charges + Rédaction de la charte + Planning et partage des tâches + Développement de la BD usinage + Préparation de la présentation
Pliya Tatiana	Documentation + Rédaction du cahier des charges + Création de la BD usinage + PLM de projet + Rédaction rapport + Préparation de la présentation

I. Project Management Lifecycle

➤ Etat d'avancement du projet :

☒ **Phase de création du projet**

☒ **Phase de planification**

☒ **Phase d'exécution am**

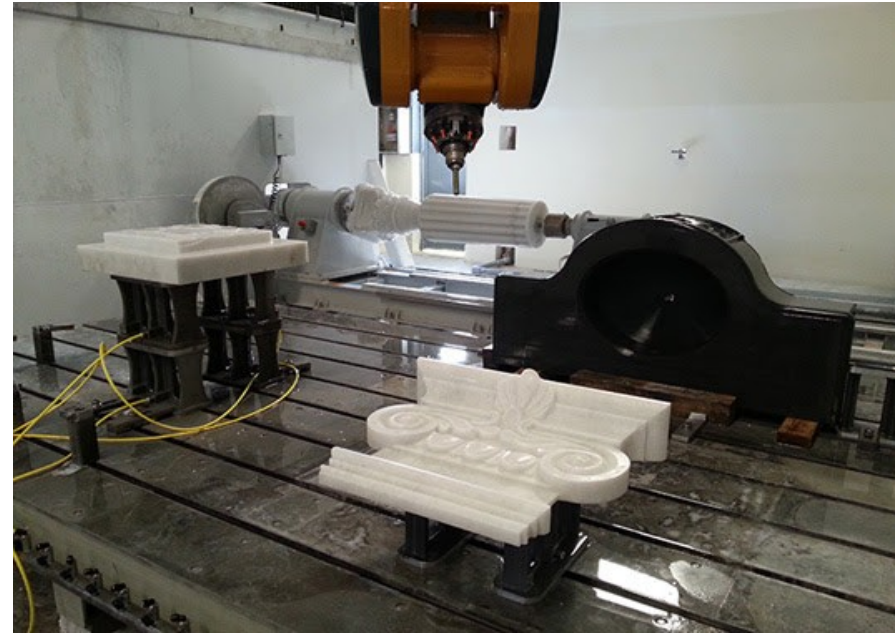
☐ **Livrables non terminés**

☐ **Phase de tests**

☐ **Validation et livraison**

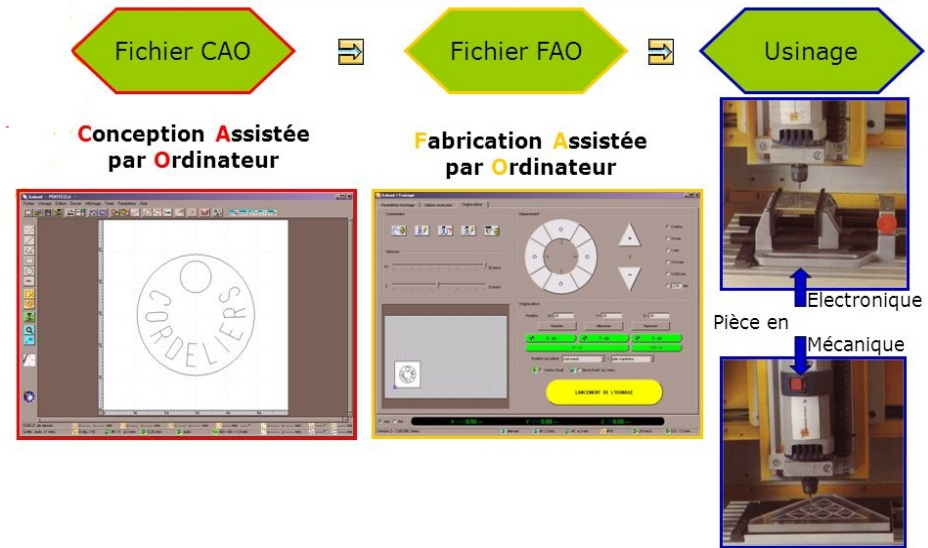
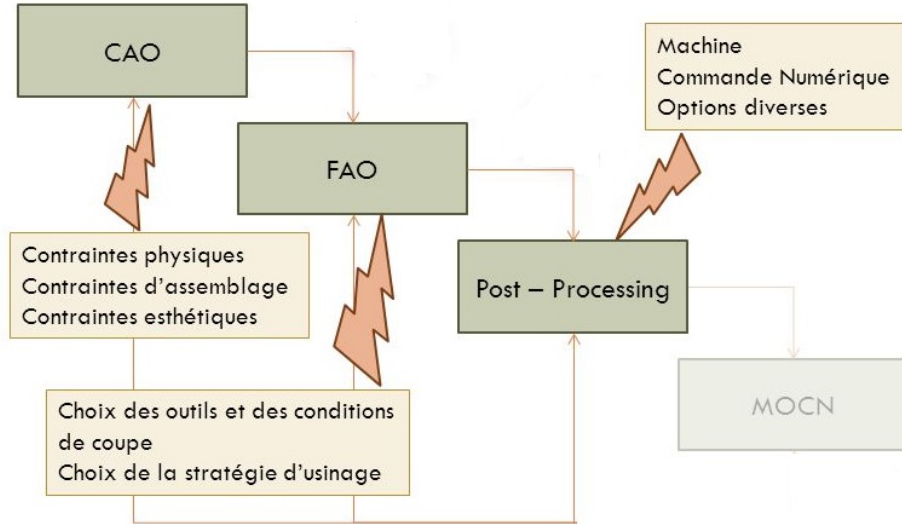


II. Programmation de commande numérique



- Usage par enlèvement de matière : tournage, fraisage, perçage, gravure, chariotage, alésage, défonçage
- Découpe avec : couteau, laser, jet d'eau, plasma, flamme ou oxydation
- poinçonnage
- impression 3D : par dépôt de matière ("fabrication additive"), durcissement d'une résine

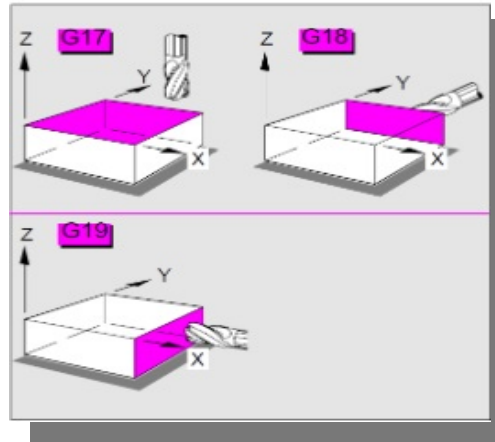
II. Programmation de commande numérique



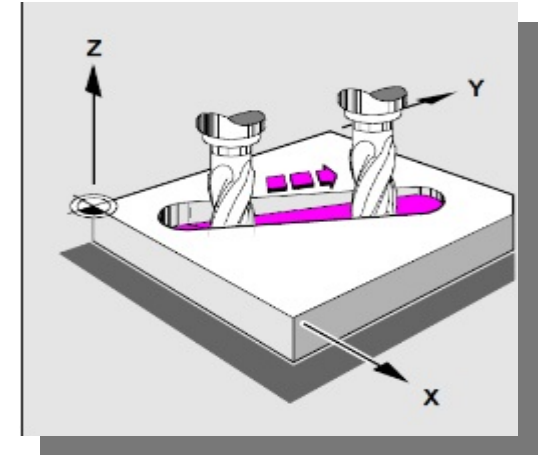
II. Programmation de commande numérique

Fraisage CNC

1. O100;
2. M6 T1;
3. G0 G90 G40 G54 X0 Y-5;
4. M13 S2500;
5. G0 G43 H1 Z-4;
6. G1 F150 Y20;
7. X40;
8. Y0;
9. X-5;
10. M9;
11. M5;
12. G0 G53 Z0;
13. G0 G53 Y0;
14. M30;
15. %

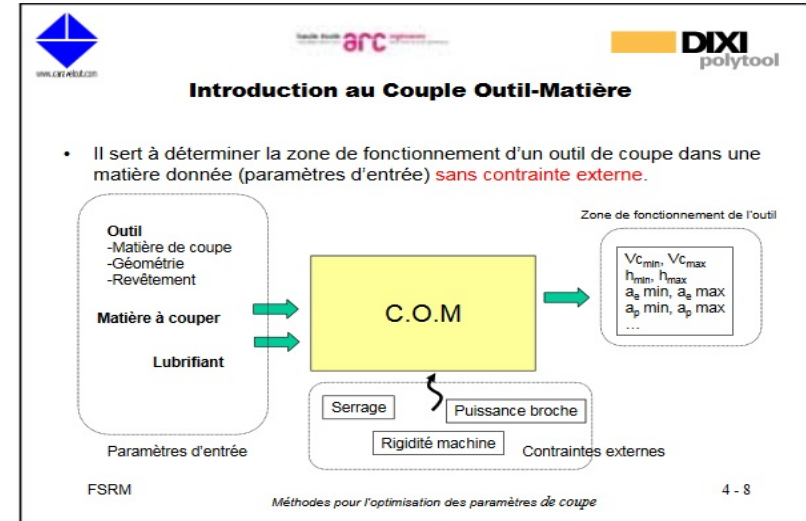
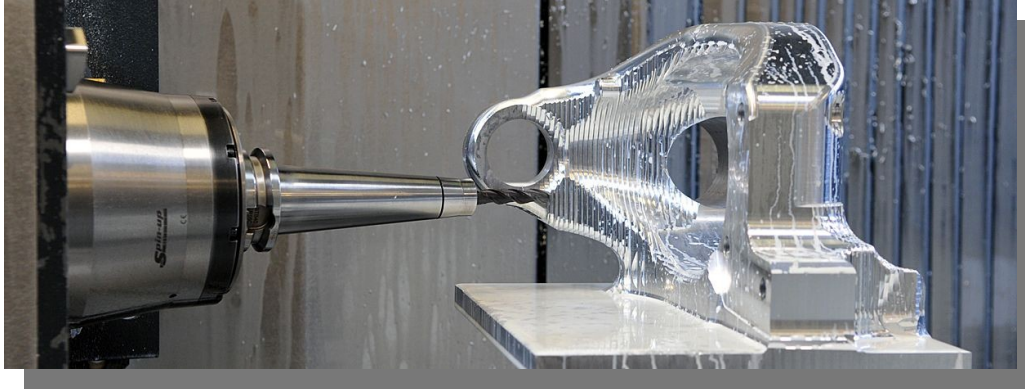


Sélection du plan
de l'usinage



Coordonnées du point de
destination avec avance
par commande F

III. Base de connaissance en usinage



III. Base de connaissance en usinage

PROBLEMES	SOLUTIONS									
	Réduire la vitesse de coupe	Augmenter la vitesse de coupe	Réduire l' avance	Augmenter l' avance	Réduire la profondeur de coupe	Augmenter la profondeur de coupe	Choisir une nuance plus résistance à l' usure	Choisir une nuance plus tenace	Choisir un petit rayon de bec	Choisir une géométrie positive
Usure en dépouille rapide	X						X			
Usure en entaille	X						X			
Usure en cratère rapide	X		X				X			X
Déformation plastique	X		X				X			
Formation d'arête rapportée		X								X
Petites fissures perpendiculaires à l'arête de coupe								X		
Petites fractures de l'arête (écaillage)		X						X		X
Rupture de plaquette			X		X			X		
Copeaux longs enchevêtrés				X		X			X	
Vibration	X			X	X				X	X

III. Base de connaissance en usinage

	A	B	C	D	E	F
1	Matériaux		Propriétés physiques			Propriétés mécaniques
2			Masse volumique (g/cm³)	Dureté (Mohs)	Module d'Young (Gpa)	Résistance à la traction (Mpa)
3	Céramiques					
4	Carbure de silicium		3,2	9		610
5	Nitride d'aluminium		3,26	7	318	300
6	Sable					
7	Béton		2450			
8	Bois/Pierre		400-1000		0,01	2,5
9	Marbre					
10	Nylons					
11	Nylon 618					
12	Nylon 645					
13	Filament PETT					
14	Organiques					
15	Tissus biologiques					
16	Cires					
17	Matières alimentaires					
18	Métaux					
19	Acier inoxydable		7800		210	440-640
20	Acier maraging		8,1			
21	Aluminium		2700		67,5	30
22	Bronze		8800		106	126
23	Fonte		7250			
24	Plexiglass					
25	Cuivre		8930		100	40
26	Alliages					
27	Inconel					
28	Monel					
29	Uranus					
30	Hastelloy					
31	Plastiques					
32	PLA					
33	ABS					
34	PET					

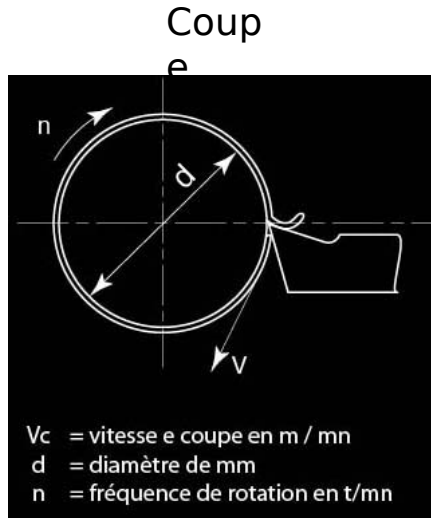
III. Base de connaissance en usi

A	B	C	D	E
19	Acier inoxydable	7800		210
20	Acier maraging	8,1		
21	Aluminium	2700		67,5
22	Bronze	8800		106
23	Fonte	7250		
24	Plexiglass			
25	Cuivre	8930		100
26	Alliages			
27	Inconel			
28	Monel			
29	Uranus			
30	Hastelloy			
31	Plastiques			
32	PLA			
33	ABS			
34	PET			
35	TPU			
36	PC			
37				
38				
39	Matériaux	Dureté (indicatif) Brinell	Machines de tournage (turning)	
40	Aciers au carbone		vitesse de coupe (matériau de coupe : HSS)	vitesse d'avar
41	1212,1213,1215	100-150/150-200	150-160	
42	1108,1109,1115, 1117, 1118, 1120, 1126, 1211	100-150/150-200	130-120	
43	1132, 1137, 1139, 1140, 1144, 1146, 1151	175-225/275-325/325-375/375-425	120/75/50/40	
44	11L17, 11L18, 12L13, 12L14	100-150/150-200/200-250	140/145/110	
45	1006, 1008, 1009, 1010, 1012,	100-125/125-175/175-225/225-275	120/110/90/70	
46	1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022,			
47	1023, 1024, 1025, 1026, 1513, 1514			
48	1027, 1030, 1033,	125-175/175-225/225-275/275-2		
49	1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042,			
50	1043, 1045, 1046, 1048, 1049, 1050, 1052, 1524,			
51	1526, 1527, 1541			
52				
53				
54				

Free-Machining Grades					
AISI No.	Composition *, %				SAE No.
	C	Mn	P	S	
Resulfurized					
1108	0.08 to 0.13	0.50 to 0.80	0.040 max	0.08 to 0.13	1108
1109	0.08 to 0.13	0.60 to 0.90	0.040 max	0.08 to 0.13	1109
1110	0.08 to 0.13	0.30 to 0.60	0.040 max	0.08 to 0.13	1110
1116	0.14 to 0.20	1.10 to 1.40	0.040 max	0.16 to 0.23	1116
1117	0.14 to 0.20	1.00 to 1.30	0.040 max	0.08 to 0.13	1117
1118	0.14 to 0.20	1.30 to 1.60	0.040 max	0.08 to 0.13	1118
1119	0.14 to 0.20	1.0 to 1.30	0.040 max	0.24 to 0.33	1119
1132	0.27 to 0.34	1.35 to 1.65	0.040 max	0.08 to 0.13	1132
1137	0.32 to 0.39	1.35 to 1.65	0.040 max	0.08 to 0.13	1137
1139	0.35 to 0.43	1.35 to 1.65	0.040 max	0.13 to 0.20	1139
1140	0.37 to 0.44	0.70 to 1.00	0.040 max	0.08 to 0.13	1140
1141	0.37 to 0.45	1.35 to 1.65	0.040 max	0.08 to 0.13	1141
1144	0.40 to 0.48	1.35 to 1.65	0.040 max	0.24 to 0.33	1144
1145	0.42 to 0.49	0.70 to 1.00	0.040 max	0.04 to 0.07	1145
1146	0.42 to 0.49	0.70 to 1.00	0.040 max	0.08 to 0.13	1146
1151	0.48 to 0.55	0.70 to 1.00	0.040 max	0.08 to 0.13	1151
Resulfurized and rephosphorized					
1211	0.13 max	0.60 to 0.90	0.07 to 0.12	0.10 to 0.15	1211
1212	0.13 max	0.70 to 1.00	0.07 to 0.12	0.16 to 0.23	1212
1213	0.13 max	0.70 to 1.00	0.07 to 0.12	0.24 to 0.33	1213
1215	0.09 max	0.75 to 1.05	0.04 to 0.09	0.26 to 0.35	1215
12L 14++	0.15 max	0.85 to 1.15	0.04 to 0.09	0.26 to 0.35	12L14

Brinell, Vickers,
Rockwell
High Speed Steel
Carbure
Revêtement dur,
nitrure de chrome

III. Base de connaissance en usinage



$$n \neq \frac{V_c}{\pi * d}$$

Vitesse d'avance

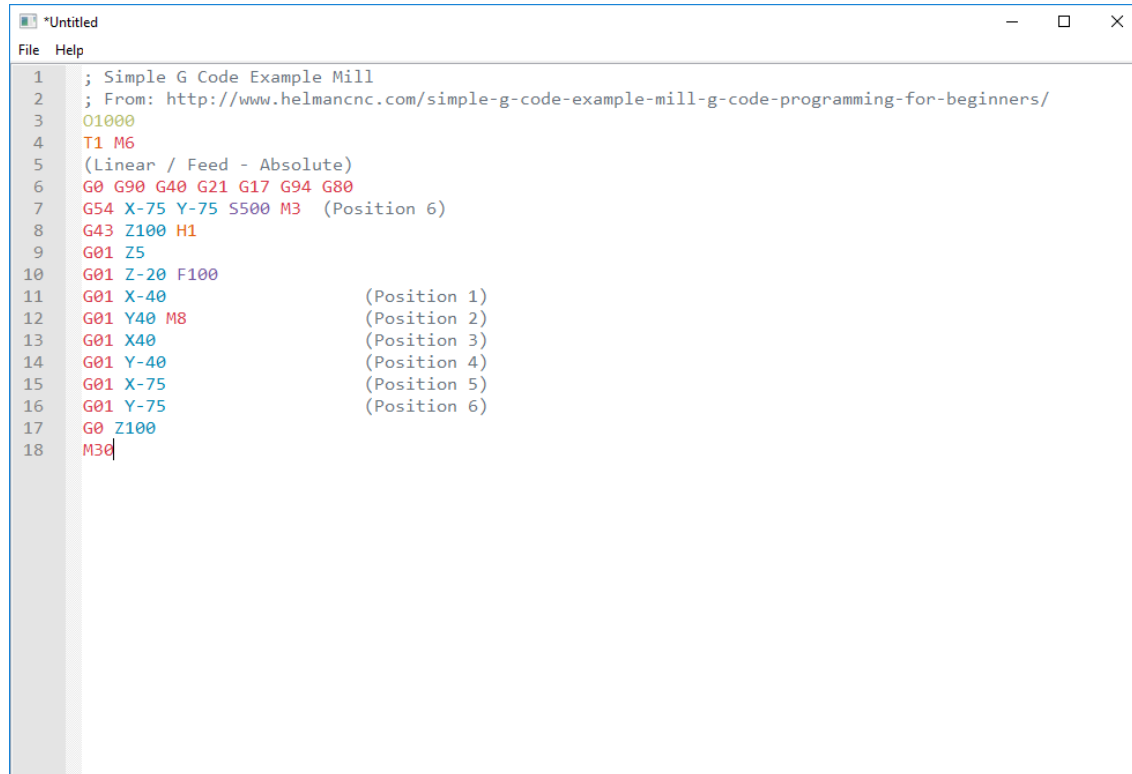
$$V_f = f_z \times z \times N$$



Tournage
($z=1$)

$$V_f = \cancel{f_z} * N$$

IV. Environnement de programmation



The screenshot shows a window titled '*Untitled' with a menu bar containing 'File' and 'Help'. The main text area contains 18 lines of G-code. The code is color-coded: line numbers are grey, semicolons are black, and G, T, M, X, Y, Z, F, S, and H codes are in various colors (green, red, blue, black). Comments are in black text. The code defines a simple mill operation with positions 1 through 6.

```
1 ; Simple G Code Example Mill
2 ; From: http://www.helmacnc.com/simple-g-code-example-mill-g-code-programming-for-beginners/
3 O1000
4 T1 M6
5 (Linear / Feed - Absolute)
6 G0 G90 G40 G21 G17 G94 G80
7 G54 X-75 Y-75 S500 M3 (Position 6)
8 G43 Z100 H1
9 G01 Z5
10 G01 Z-20 F100
11 G01 X-40 (Position 1)
12 G01 Y40 M8 (Position 2)
13 G01 X40 (Position 3)
14 G01 Y-40 (Position 4)
15 G01 X-75 (Position 5)
16 G01 Y-75 (Position 6)
17 G0 Z100
18 M30
```

IV. Environnement de programmation

```
class Lexer {  
public:  
    Lexer(const std::wstring& text);  
    Token::Token next();  
  
private:  
    void scan_integer_();  
    Token::Token tokenize_alpha_();  
    Token::Token tokenize_comment_();  
    Token::Token tokenize_number_();  
  
    unsigned pos_;  
    const std::wstring& text_;  
    size_t text_length_;  
};
```

IV. Environnement de programmation

```
class Parser {  
public:  
    Parser(const std::wstring& text);  
    ~Parser();  
    Program parse();  
  
private:  
    void add_word_no_dupl_(std::vector<Word>& words, WordSet& rec_words);  
    void add_word_no_type_dupl_(std::vector<Word>& words, TokenSet& rec_types);  
    void advance_lexer_();  
    Block fetch_block_();  
    std::wstring fetch_comment_();  
    Header fetch_header_();  
    unsigned fetch_unsigned_();  
    Word fetch_word_();  
  
    Lexer* lexer_;  
    Token::Token cur_token_;  
    Token::Token next_token_;  
    const std::wstring& text_;  
};
```

IV. Environnement de programmation

```
PROGRAM = HEADER (BLOCK)*  
HEADER = PGM_START (IDENT)? END_BLOCK  
BLOCK = (SEQ_NUMBER)? DATA_WORDS END_BLOCK  
DATA_WORDS = (G_WORD)* ( (DIM_WORD)+ (IERP_WORD)* (ADV_WORD)* )? (SPIN_WORD)* (TOOL_WORD)* (AUX_WORD)*
```

```
G_WORD = "G" PARAM  
DIM_WORD = ("X" | "Y" | "Z" | "U" | "V" | "W" | "P" | "Q" | "R" | "A" | "B" | "C") PARAM  
IERP_WORD = ("I" | "J" | "K") PARAM  
ADV_WORD = ("E" | "F") PARAM  
SPIN_WORD = "S" PARAM  
TOOL_WORD = ("D" | "T") PARAM  
AUX_WORD = "M" PARAM
```

```
PGM_START = "%"   
IDENT = DIGITS | COMMENT  
COMMENT = "(" [ ^%:]* ")"  
SEQ_NUMBER = "N" 3 * DIGIT  
END_BLOCK = "\n"
```

```
PARAM = (DIGITS "=")? NUMBER  
NUMBER = ("+" | "-") ( (DIGIT)+ ("." (DIGIT)*)? | "." (DIGIT)+ )
```

IV. Environnement de programmation

```
Program { header: Header, blocks: std::vector<Block> }  
Header { identifier: std::optional<std::variant<unsigned, wstring>> }  
Block { number: std::optional<BlockNumber>, data_words: std::vector<Word> }  
BlockNumber { value: unsigned }  
Word { type: Token::Type, value: float }
```

IV. Environnement de programmation

```
class Visitor;
```

```
class BaseNode {  
public:  
    virtual void accept(class Visitor*) = 0;  
};
```

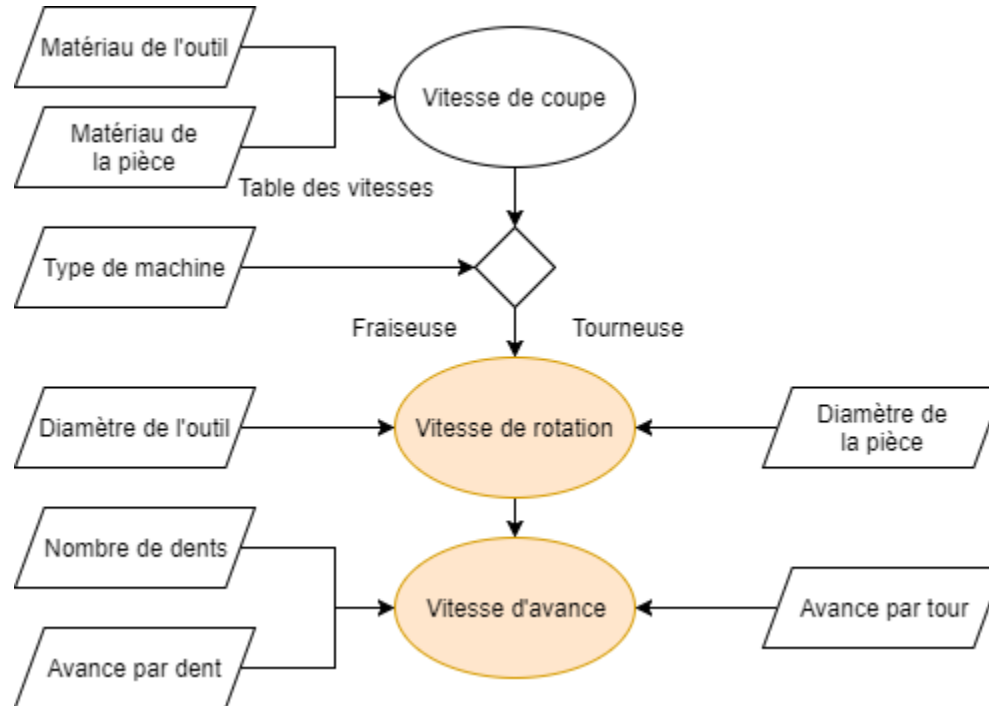
```
/* */
```

```
class Program : public BaseNode {  
public:  
    Program() { }  
    void accept(Visitor* v);  
    Header header;  
    std::vector<Block> blocks;  
};
```

```
class Visitor {  
public:  
    virtual void visit(Program* p) { };  
};
```

IV. Environnement de programmation

- Structure du programme :



IV. Environnement de programmation

```
class SpeedVisitor : public Visitor {
    SpeedVisitor() { }
    void visit(Word* w) {
        if (w->kind == Token::G) {
            if (w->value == 70) {
                units_ = SpeedVisitor::Imperial;
            } else if (w->value == 71) {
                units_ = SpeedVisitor::Metrics;
            } else if (w->value == 96) {
                speed_kind_ = SpeedVisitor::ConstantSurfaceSpeed;
            } else if (w->value == 97) {
                speed_kind_ = SpeedVisitor::RevPerMinute;
            }
        } else if (w->kind == Token::S) {
            speed_records_.emplace_back(0, w->value, calcSpindleSpeed());
        }
    }
}

struct SpeedRecord { unsigned line; float value; float calculatedValue; }

};
```


IV. Environnement de programmation

➤ Paramètres machine-outil :

si G96 :

// le Δd est la différence entre le diamètre initial de la pièce et celui où on se

// trouve en fin de mouvement

si !G50 et $\Delta d \geq \text{threshold}$:

warn

si G70 :

// on convertit les mètres par minute en pouces par minute

$$SS = \frac{V_c}{39,37}$$

sinon : // G71

$$SS = V_c$$

sinon : // G97

$$SS = \frac{1000 \times V_c}{\pi \times d}$$

sachant V_c la vitesse de coupe (m/min) et d le diamètre de l'objet en rotation (mm)

IV. Environnement de programmation

si fraiseuse :

si G95 :

$$FR = f_z \times Z$$

sinon : // G94

$$FR = f_z \times Z \times N$$

sinon : // tourneuse

si G95 :

$$FR = f$$

sinon : // G94

$$FR = f \times N$$

sachant pour la fraiseuse f_z l'avance par dent par tour et Z le nombre de dents et pour la tourneuse f l'avance par tour

V. Perspectives d'évolution du projet

- Simulation sur machine cartésienne :
Transition du code vers la réalité
- Pilotage avec rétroaction : Vérification et retour d'expérience pour une amélioration

Merci de votre
attention 😊

Shruti
Yousra
Pliya Tatiana
Tétar Adrien