

Discrete 3D surfaces of revolution

Final presentation

Zied BEN OTHMANE

Thomas BENOIST

Adrien BISUTTI

Lydie RICHAUME

University of Poitiers

March 2nd, 2016

Outline

- 1 Introduction
- 2 Work achieved
- 3 Project management
- 4 Conclusion

Outline

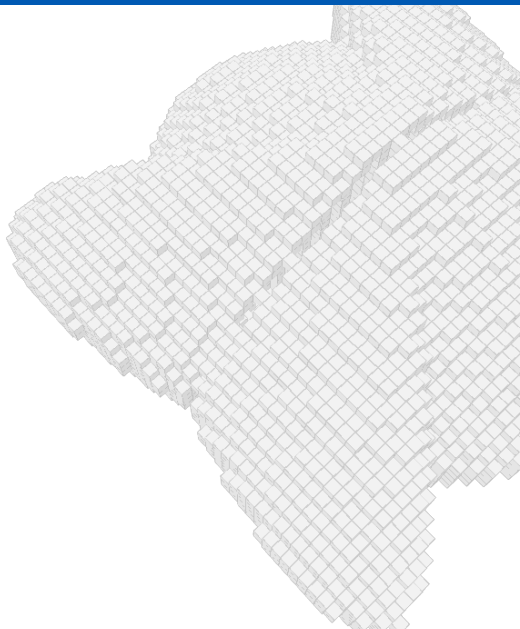
1 Introduction

- Collaborators and clients
- Roles
- Context
- Objectives

2 Work achieved

3 Project management

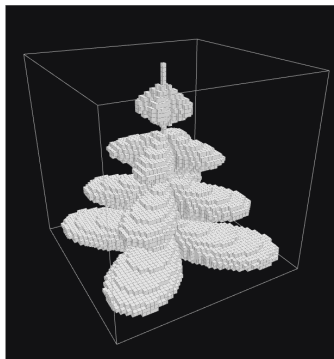
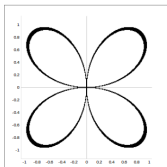
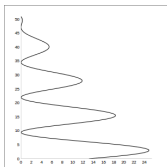
4 Conclusion



- Clients :
 - Éric ANDRES (Professor and former director of XLIM-SIC department)
 - Gaëlle LARGETEAU-SKAPIN (University lecturer, Discrete geometry)
- Exemple of final user :
 - Aurélie MOURIER (Artist)
- Pedagogic Supervisor :
 - Philippe MESEURE (Professor, Computer Graphics)

- Team composition :
 - Thomas BENOIST - Project manager
 - Zied BEN OTHMANE - Quality manager
 - Adrien BISUTTI - Risks manager
 - Lydie RICHAUME - Tasks manager

- Éric ANDRES and Gaëlle LARGETEAU-SKAPIN developed a new algorithm to model discrete surfaces of revolution.
- Display the result with Mathematica



- Need a usable tool by everyone and everywhere

- Surfaces visualization tool
 - 3D, slices visualization
 - Choose the generatrix and directrix
 - Export the results
- Algorithm to generate surfaces of revolution
 - Provided by the clients
 - Possible evolution of the algorithm

Outline

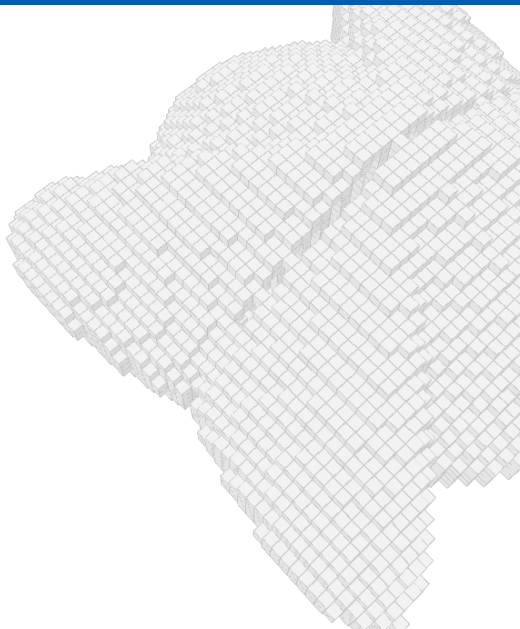
1 Introduction

2 Work achieved

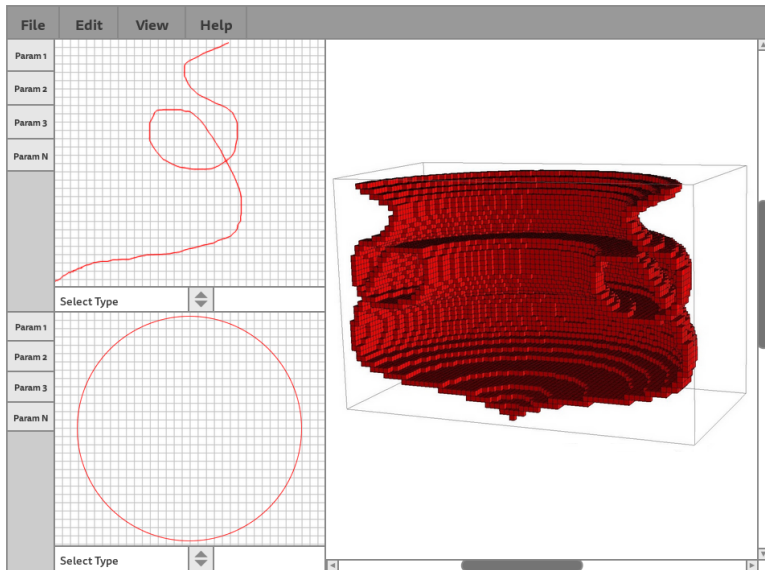
- Prototype
- Demonstration
- Technical aspect

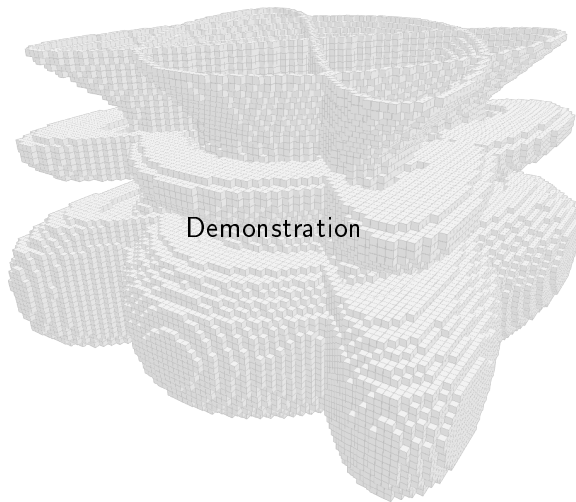
3 Project management

4 Conclusion



Prototype





- Connexity
 - Just what you want
 - All in one pass
- Rendering
 - Computing only for rendering
 - Pre-computing during generation
 - Ignored → let the computation to the GPU
- Implicit curve display
 - Discretisation of the curve
 - Use a library

Outline

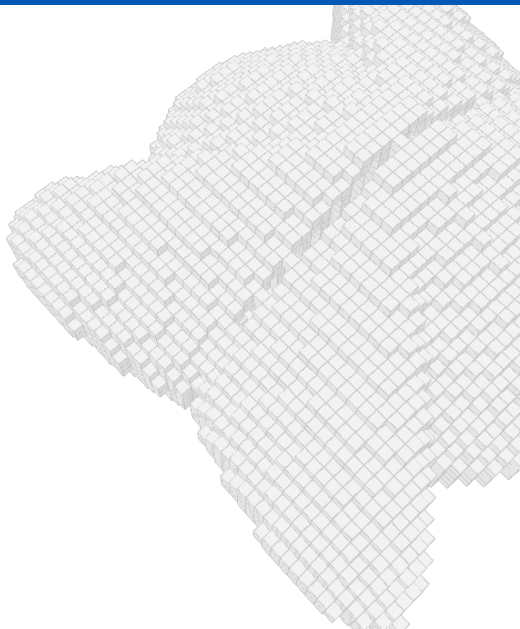
1 Introduction

2 Work achieved

3 Project management

- Task list
- Gantt diagram
- Progress
- Deliverables
- Risks
- Risk evolution
- Quality insurance plan
- Costs

4 Conclusion



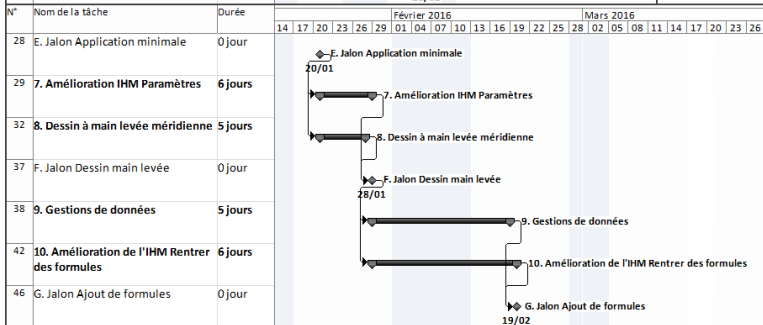
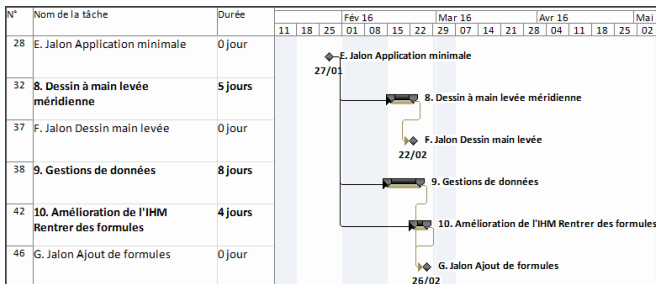
Task list

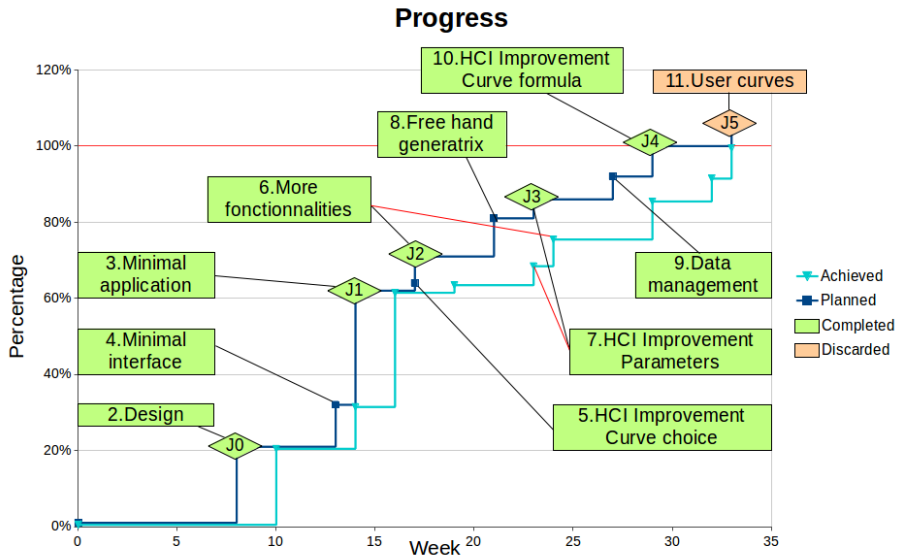
1 - Documentation, test and users help		✓	
2 - Design		✓	
3 - Functional kernel	✓	4 - Minimal interface	✓
6 - Fonctionnalités adding	✓	5 - Interface enhancement Curve choice	✓
8 - Free hand generatrix	✓	7 - Interface enhancement Parameters	✓
9 - Data management	✓	10 - Interface enhancement Formula input	✓
11 - User's curve			✗
12 - Technical report			✓

Planned diagram

Revised diagram

Gantt diagram





Deliverables

N°	Deliverable	Tasks	Planned date	Actual date
1	Interface and algorithm result	2, 3, 4	Dec. 23 rd	Jan. 18 th
2	Minimal application	5, 6	Jan. 21 st	Jan. 25 th
2 ^{bis}	Multi-slice and parameters	7	—	Jan. 29 th
3	Free hand drawing and curves with editable parameters	7, 8	Jan. 29 th	Feb. 24 th
4	Equations and export	9, 10	Feb. 19 th	Feb. 24 th
5	Final application and documentation	1 to 11	Mar. 2 nd	Mar. 2 nd
5 ^{bis}	Final documentation	1	Mar. 11 th	Mar. 14 th

List of risks

Risk	Gravity	Probability	Criticity	
Server linked problems	1	0	0	*
Equipment/device dysfunction	1	1	1	+ ✓
New clients	1	2	1	* ✓
Validation reveals serious technical problem	2	1	1	*
3D rendering needs too much ressources	2	1	1	* ✓
Evolution of the generation algorithm	1	3	2	*

Risk evolution

- Server linked problems

Gravity	0	1	2	3
Delay	●			
Costs	●			
Receipts	●			
Performance	●			
Other				
Global	●			

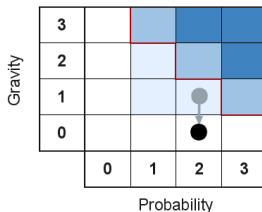
Gravity	3				
	2				
	1				
	0	●			
		0	1	2	3
		Probability			

Level	Gravity	Probability	Criticality
0	None	< 1%	No critical
1	Low	de 1% à 5%	
2	Important	de 5% à 20 %	Critical
3	Dangerous	> 20%	

Risk evolution

- New clients

Gravity	0	1	2	3
Delay	● ← ●			
Costs	●			
Receipts	●			
Performance	● ← ●			
Other				
Global	● ← ●			

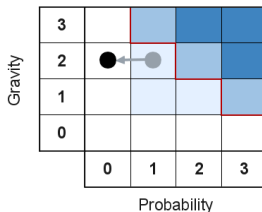


Level	Gravity	Probability	Criticality
0	None	< 1%	No critical
1	Low	de 1% à 5%	
2	Important	de 5% à 20 %	Critical
3	Dangerous	> 20%	

Risk evolution

- Slow rendering

Gravity	0	1	2	3
Delay			●	
Costs	●			
Receipts	●			
Performance			●	
Other				
Global			●	

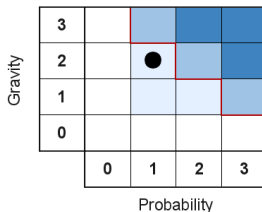


Level	Gravity	Probability	Criticality
0	None	< 1%	No critical
1	Low	de 1% à 5%	
2	Important	de 5% à 20 %	Critical
3	Dangerous	> 20%	

Risk evolution

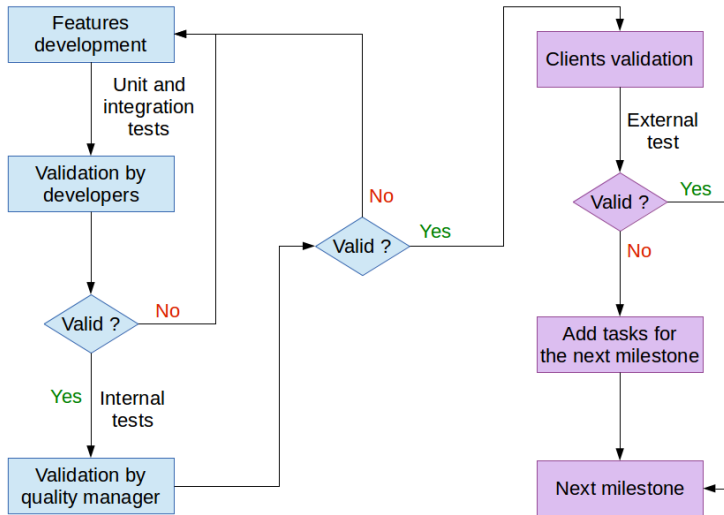
- Evolution of the generation algorithm

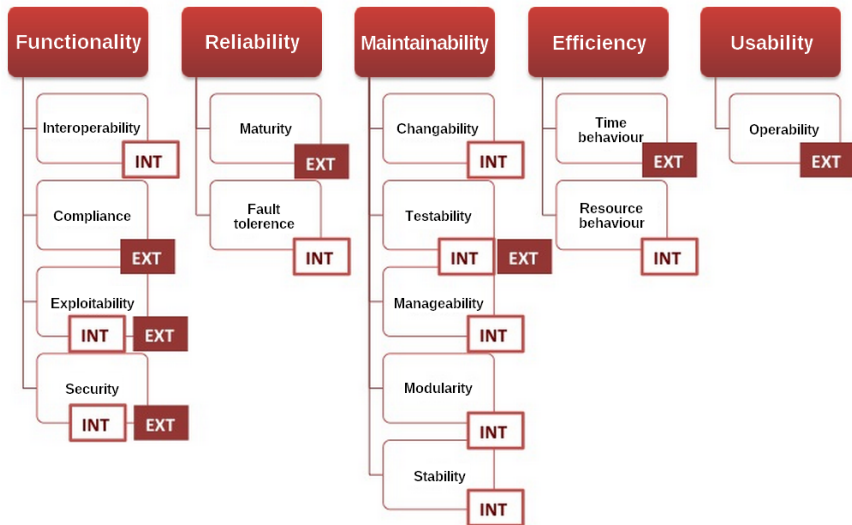
Gravity	0	1	2	3
Delay	●			
Costs	●			
Receipts	●			
Performance			●	
Other				
Global			●	



Level	Gravity	Probability	Criticality
0	None	< 1%	No critical
1	Low	de 1% à 5%	
2	Important	de 5% à 20 %	Critical
3	Dangerous	> 20%	

Quality insurance plan





Software quality measurement

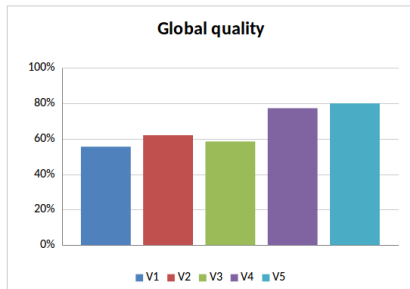
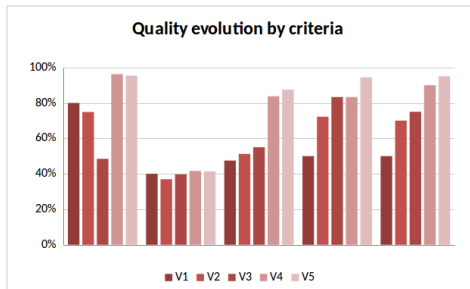
1	Question	Version 1	Version 2	Version 3	Version 4	Version 5
1	Overall vision	1	1	0.5	1	1
2	The ease to find the information	0.5	0.5	0.5	0.5	1
3	Response speed	0.5	0.5	0.5	1	1
4	Utility of the information	0	0.5	0.5	1	1
5	The choice of title and heading and their meanings	0.5	1	1	1	1
6	The completeness of the information found against the need	1	0.5	1	1	1
7	Rapidité d'exécution	0	0.5	1	1	1
8	Errors rate	0.5	0.5	0.5	1	1
9	Handling the use	1	1	1	0.5	0.5
10	The reliability of the application	0	1	1	1	1
	Total	50%	70%	75%	90%	95%

Standard divisions

- ① Quality model
- ② External metrics
- ③ Internal metrics
- ④ Quality in use metrics

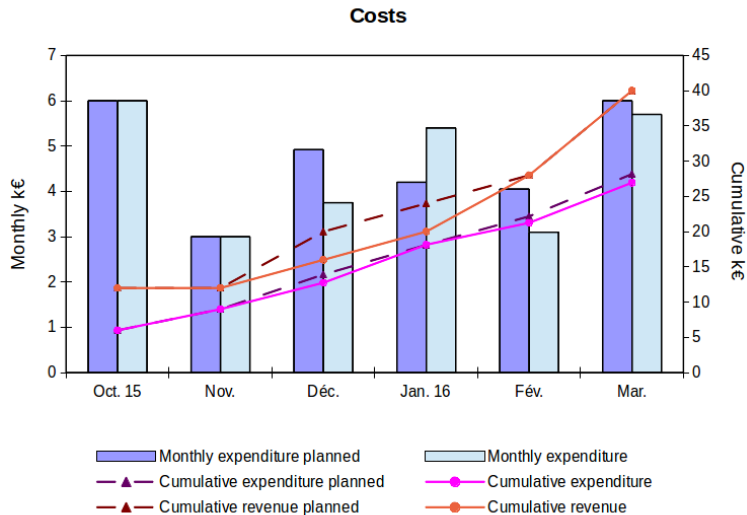
1	Functionality	Level 1		Level 2		Level 3		Level 4		Level 5	
		INT	EXT	INT	EXT	INT	Ext	INT	Ext	INT	Ext
1	Interoperability										
Goal	Ability to interact with one or more systems										
Question	Is the application uses norms and technical standards?										
	Evaluation	90%		75%		85%		100%		95.83%	
1	Adequacy										
Goal	Checking the adequacy of spots against the needs										
Question	Does each function is adequate to the customer need?										
	Evaluation		100%		80%		25%		85%		90%
0.3	Operability										
Goal	The ability to properly use the software system										
Question	At what level the software is usable?										
	Evaluation	25 %	25 %		32.14%	35.71%	35.71%		100%		100%
	Note I/E	76.66 %	83.33%	75%	74.76%	60.35%	30.35%	100%	92.5%	95.83%	95%
	Fonctionnalité	79.99 %		74.88 %		45.35 %		96.25%		95.41%	

Software quality evaluation



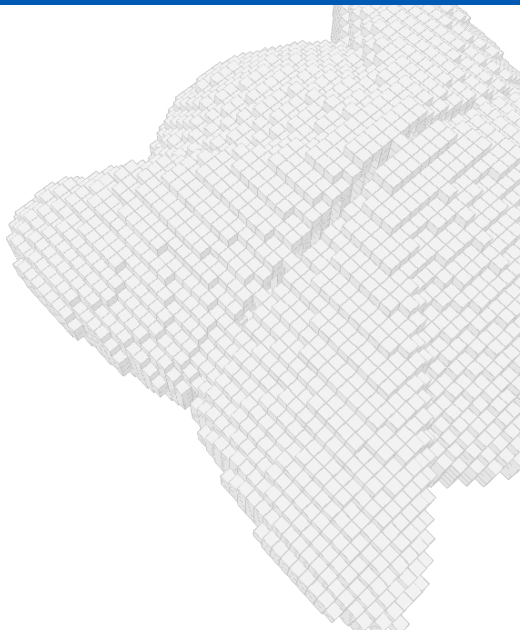
Q.I.P Reviews

- 1 Such techniques to analyze the quality during the requirements phases.
- 2 Well-differentiated characteristics of software quality has been developed.
- 3 A large number of software quality-evaluation metrics have been defined.
- 4 Quality can lead to significant savings in software life-cycle costs.



Outline

- 1 Introduction
- 2 Work achieved
- 3 Project management
- 4 Conclusion**



- Technical Javascript improvement (classes, worker, blob, webgl, etc.)
- Final deliverable in two step
- Perspectives
 - Réutilisation dans quelques semaines
 - Ajout de nouveau(x) algo

- Javascript improvement (classes, worker, blob, etc.)
- WebGL improvement
- Résolution de problème mathématique (matrice de changement de repère, tracer de courbe implicite)

Discrete 3D surfaces of revolution

Final presentation

Thanks for your attention.

Are there any questions ?