

Discrete 3D surfaces of revolution

Final presentation

Zied BEN OTHMANE

Thomas BENOIST

Adrien BISUTTI

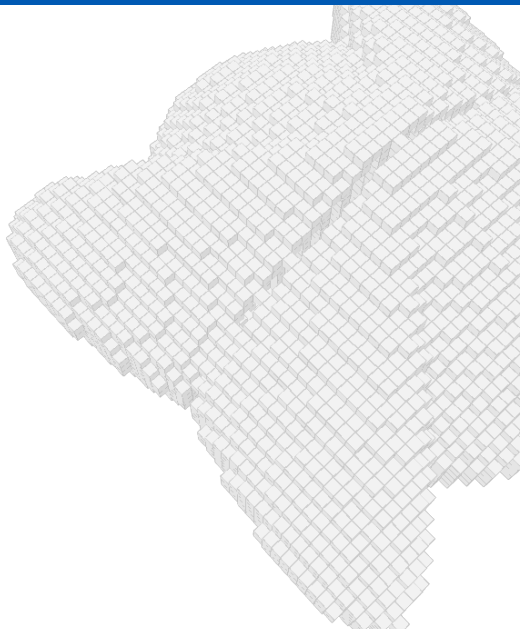
Lydie RICHAUME

University of Poitiers

March 21st, 2016

Outline

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



Outline

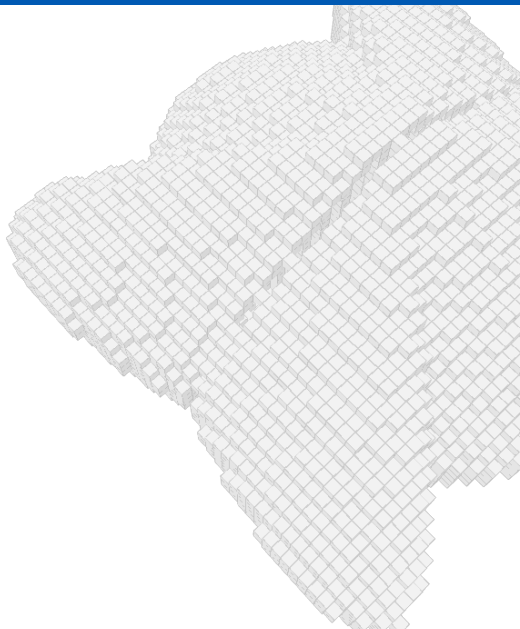
1 Introduction

- Collaborators and clients
- Context
- Objectives

2 Work achieved

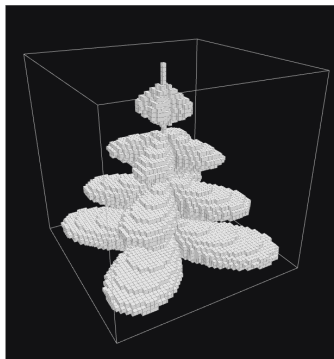
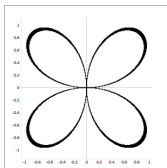
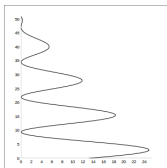
3 Project management

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)

- Éric ANDRES and Gaëlle LARGETEAU-SKAPIN developed a new algorithm to model discrete surfaces of revolution.



- Display the result with Mathematica
- Need a tool usable 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

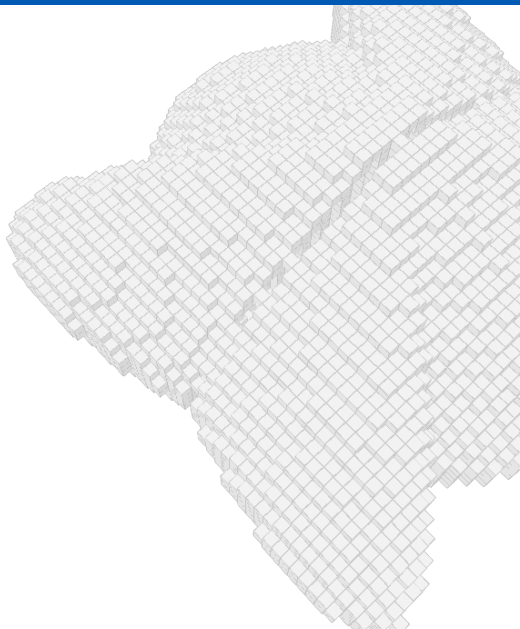
1 Introduction

2 Work achieved

- Technological choices
- Prototype
- Demonstration
- Technical aspect

3 Project management

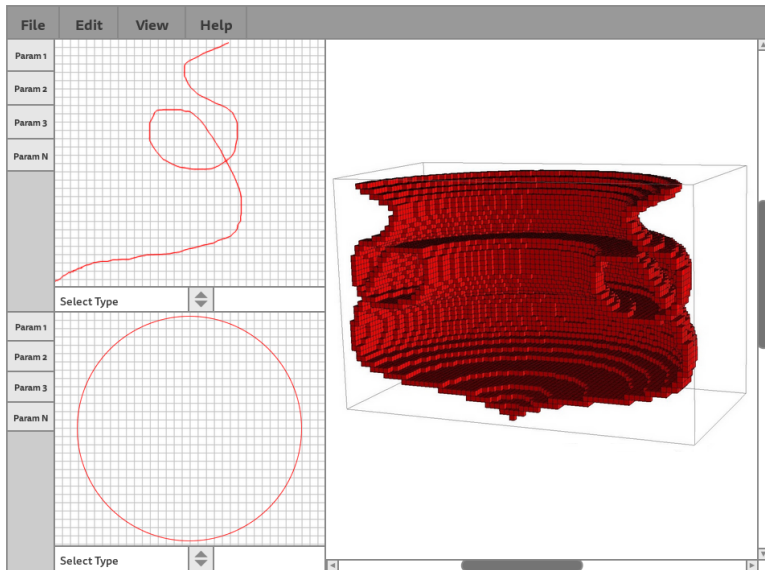
4 Conclusion

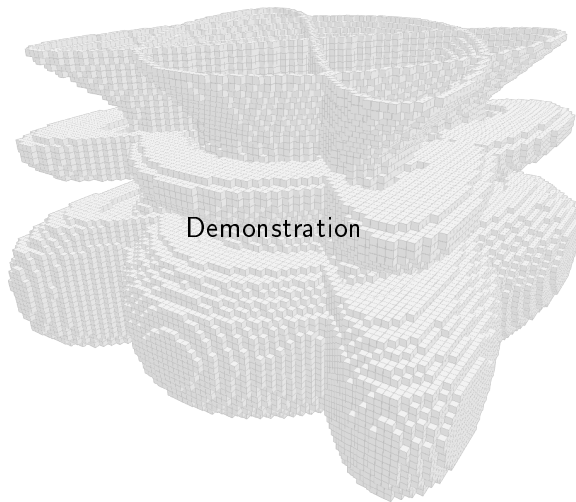


Tool usable by everyone and everywhere → web application

- Mathematica ?
 - Algorithm already implemented for Mathematica
 - Server application : difficult to set up a server from the university
 - Client application : user would have to install Mathematica (not free)
- Without Mathematica
 - Server application : security problems, data transfert
 - Client application : large quantity of computations → slow for computers with low capacities
- Choice : client application → HTML5/CSS, Javascript, WebGL

Prototype





- Curves
 - Use MathJS
 - Two kind of definition : formula or freehand drawing
 - Formula : parse string \rightarrow equation
 - Freehand drawing : retrieve points from HTML5 canvas
- Display
 - Explicit curve : easy to display
 - How to display implicit curves ?
 - Use functionPlot and HTML5 canvas
 - Formula curves : functionPlot \rightarrow SVG
 - Freehand drawing : 2D rendering context on HTML5 canvas

- Generation
 - Use MathJS
 - Two versions : graph search and brute-force
 - Interactivity → generation with worker(s)
- Rendering
 - Problem : detection of the outer faces
 - First version : computation during drawing
 - Following versions : computation during generation
 - WebGL : limited buffer size → multiple buffers

- Save & load curves
 - XML format
 - Formula curves : stores the string of the equation
 - Freehand drawn curves : stores the list of points
- PNG export
 - Formula curves : saveSvgAsPng library
 - Drawn curves : FileSaver library and HTML5 canvas functionalities
- 3D export
 - X3D : transform each voxel into boxes
 - Excessive amount of boxes → slow to access
 - STL : binary file for 3D printer

Outline

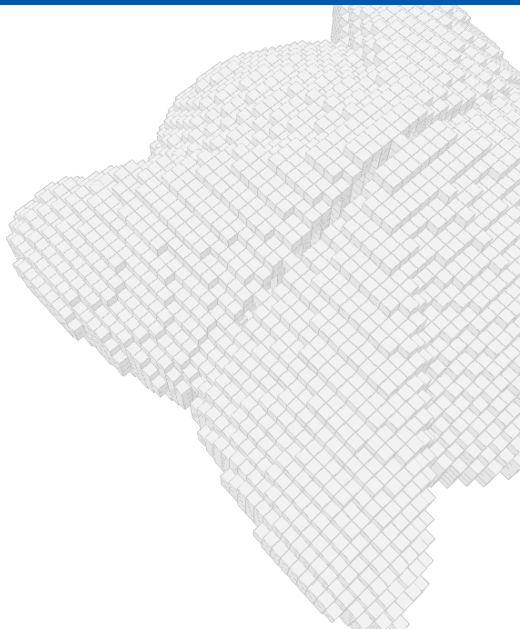
1 Introduction

2 Work achieved

3 Project management

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

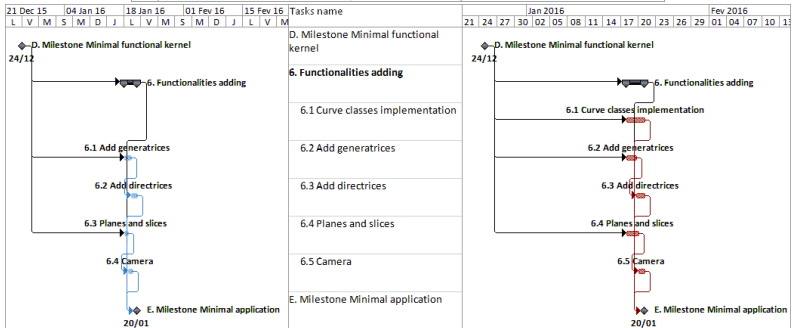
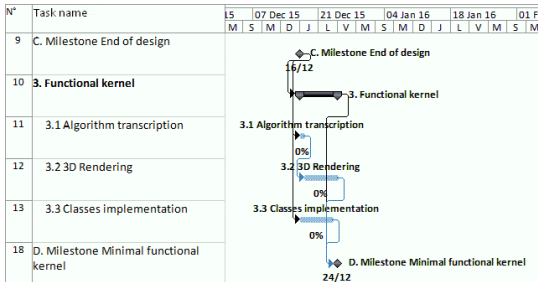
4 Conclusion



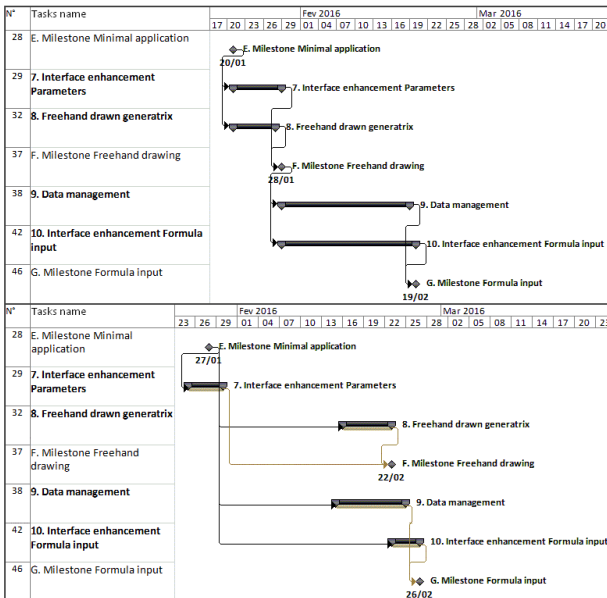
Task list

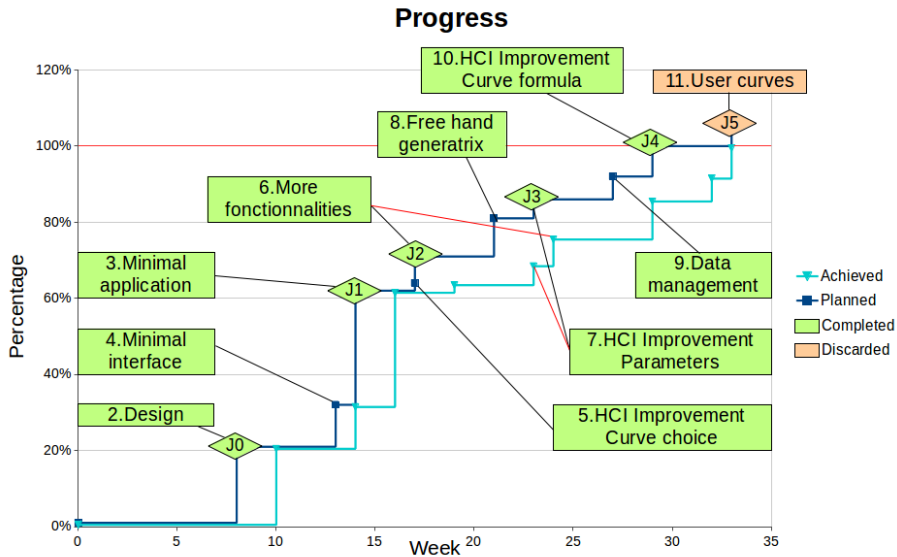
1 - Documentation, test and users help		✓	
2 - Design		✓	
3 - Functional kernel	✓	4 - Minimal interface	✓
6 - Functionalities adding	✓	5 - Interface enhancement Curve choice	✓
8 - Free hand drawn generatrix	✓	7 - Interface enhancement Parameters	✓
9 - Data management	✓	10 - Interface enhancement Formula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

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

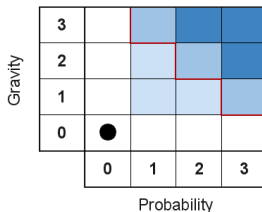
* Initial

+ Added

✓ Encountered

- Server linked problems

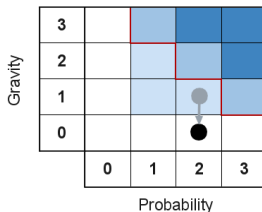
Gravity	0	1	2	3
Delay	●			
Costs	●			
Receipts	●			
Performance	●			
Other				
Global	●			



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%	

- New clients

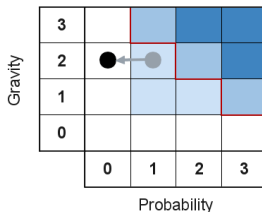
Gravity	0	1	2	3
Delay	● ← ●			
Costs	●			
Receipts	●			
Performance	● ← ●			
Other				
Global	● ← ●			



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%	

- Slow rendering

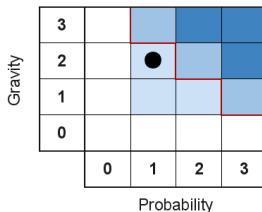
Gravity	0	1	2	3
Delay			●	
Costs	●			
Receipts	●			
Performance			●	
Other				
Global			●	



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%	

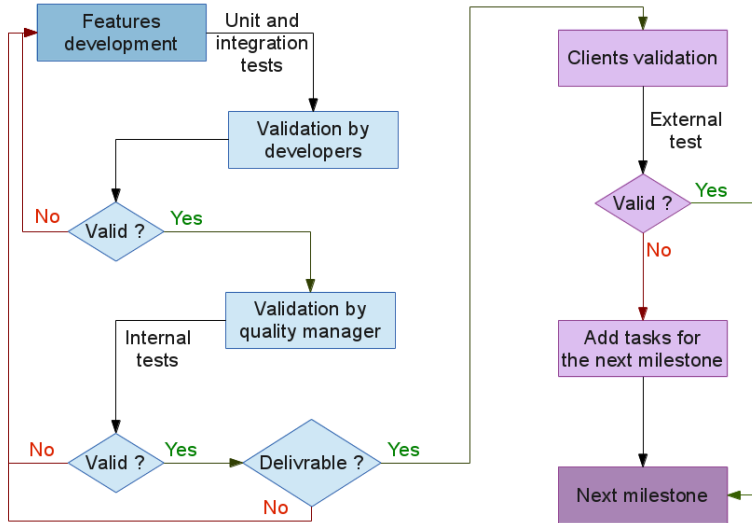
- Evolution of the generation algorithm

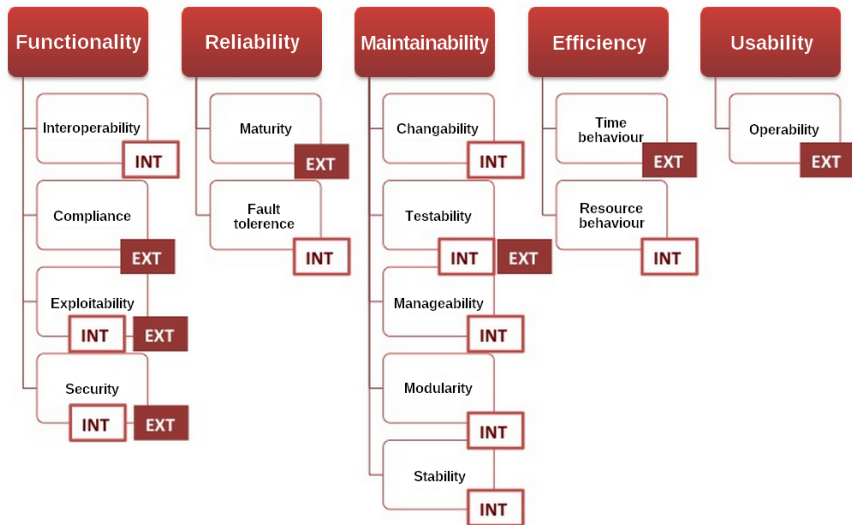
Gravity	0	1	2	3
Delay	●			
Costs	●			
Receipts	●			
Performance			●	
Other				
Global			●	



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



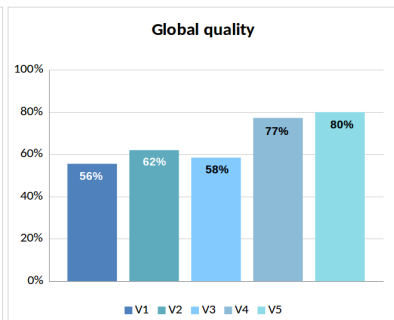
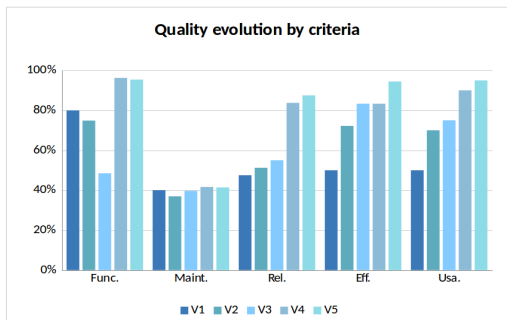


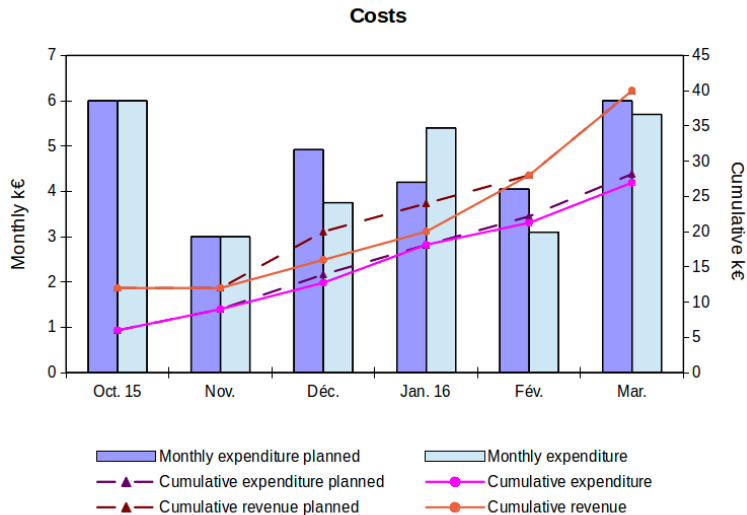
Software quality measurement

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

1	Functionality	Version 5	
		Int	Ext
1	Interoperability		
Goal	Ability to interact with one or more systems		
Question	Is the application uses norms and technical standards?		
Evaluation		95.83%	
1	Adequacy		
Goal	Checking the adequacy of spots against the needs		
Question	Does each function is adequate to the customer need?		
Evaluation			90%
0.5	Operability		
Goal	The ability to properly use the software system		
Question	At what level the software is usable?		
Evaluation			100%
	Note I/E	95.83%	95%
	Functionality	95.41%	

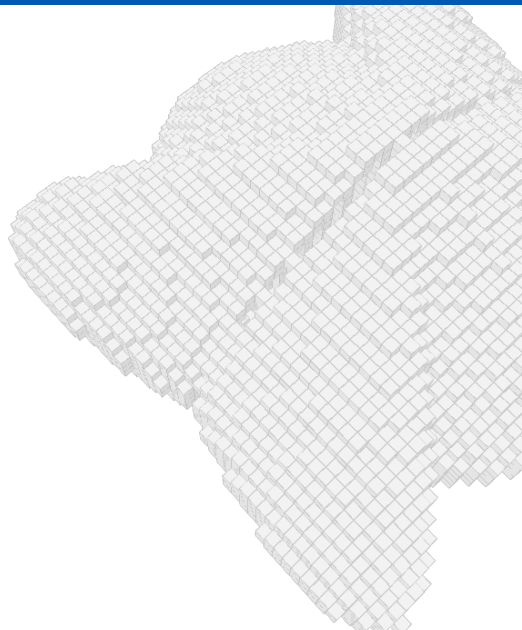
Software quality evaluation





Outline

- 1 Introduction
- 2 Work achieved
- 3 Project management
- 4 Conclusion
 - Technical aspect
 - Project management aspect
 - Personnel review



- Clients satisfied by the application
- Application available on XLIM website
- All main functionalities developed
- Final deliverable in two stage
- Possible improvements
 - X3D export
 - More information for users
 - More types of curve
- Perspectives
 - New algorithms
 - Lydie's internship subject

- Weekly meetings with the pedagogic supervisor
- Interactions with clients
- Example of quality insurance plan
- Planning management
- Risks encountered

- Improvement in Javascript
 - classes, inheritance, worker, etc.
 - jQuery, MathJS, FileSaver
 - WebGL
- Spiral development : new experience
- Solving mathematical problem
 - base matrices
 - drawing implicit curves
 - etc.

Discrete 3D surfaces of revolution

Final presentation

Thanks for your attention.

Are there any questions ?