U N I V E R S I T Y O F L I È G E

Faculty of Applied Sciences
Department of Aerospace and Mechanics

PhD Thesis

In fulfillment of the requirements
for the Degree of

Doctor in Engineering Sciences

presented by

Pierre Vueghs
Aspirant FNRS

Innovative Ray Tracing Algorithms for Space Thermal Analysis

Academic year 2008 – 2009

Préambule

Cette thèse de doctorat a été défendue, avec succès, le 9 mars 2009, en vue de l'obtention du grade de Docteur en Sciences de l'Ingénieur, en Faculté des Sciences Appliquées de l'Université de Liège.

Le jury de thèse était composé de

- Monsieur M. Hogge, Professeur à l'Université de Liège, Doyen de la Faculté de Sciences Appliquées, Président,
- Monsieur P. Beckers, Professeur à l'Université de Liège, promoteur,
- Monsieur P. Rochus, Chargé de cours à l'Université de Liège, co-promoteur,
- Monsieur G. Kerschen, Chargé de cours à l'Université de Liège,
- Monsieur J.-J. Embrechts, Professeur à l'Université de Liège,
- Monsieur D. Lemonnier, Professeur à l'École Nationale Supérieure de Mécanique et d'Aérotechnique de Poitiers,
- Madame V. Feldheim, Professeur au service de Thermique et Combustion de la Faculté Polytechnique de Mons,
- Monsieur E. van den Bulck, Professeur à la division Applied Mechanics and Energy Conversion à l'Université Catholique de Louvain,
- Monsieur T. van Eekelen, de la société Samtech, à Liège,
- Monsieur H.-P. de Koning, de l'Agence Spatiale Européenne, à Noordwijk, Pays-Bas.

Cette recherche a été financée par le Fonds National de la Recherche Scientifique de Belgique ainsi que l'Agence Spatiale Européenne, dans le cadre du programme NPI (Networking / Partnering Initiative), contrat 20180/06/NL/PA.

A Elisabeth,
qui m'a rappelé
dans un moment de doute
la raison de mon
engagement dans ce travail.

Acknowledgements

I would like to acknowledge my supervisor, Prof. Pierre BECKERS, from the Department of Aerospace and Mechanics, for the freedom he gave me during my PhD and the corrections he made to the manuscript. He gave me the opportunity to perform my thesis in a field specific to space, space that fascinates me since childhood.

I would like to express my gratitude to Prof. Pierre ROCHUS, from the Department of Aerospace and Mechanics, who gave me the opportunity to present my work during a seminar in CSL (Liège Space Center), accepted to become co-promotor of my PhD, and helped me put together the jury.

I would like to thank Prof. Pierre DUYSINX, from the Department of Aerospace and Mechanics, who helped me put together the jury.

My thanks also go to Prof. Gaëtan KERSCHEN, from the Department of Aerospace and Mechanics, for his careful reading of the manuscript.

I express my acknowledgements to Prof. Michel HOGGE, head of the Faculty of Applied Sciences at the University of Liège, who agreed to be the president of the jury.

I am very grateful to Prof. Jean-Jacques EMBRECHTS, from the Department of Electrical Engineering and Computer Science at University of Liège, Prof Véronique FELDHEIM, from the service of Heat and Combustion at the Faculté Polytechnique de Mons, Prof. Eric VAN DEN BULCK, from the division of Applied Mechanics and Energy Conversion (TME) at the Katholieke Universiteit of Leuven, and Prof. Denis LEMONNIER, head of the Laboratory of Thermal Studies of ENSMA, at Poitiers, who accepted to be member of the jury.

I would like to acknowledge Olivier PIN and Hans-Peter DE KONING for the support they gave me during my PhD. They helped me identify the challenges in terms of computer software for thermal analysis in space and gave a new interest to my thesis. I am very grateful to Olivier, who included me as a member of his team at ESTEC. I would like to thank Hans-Peter for all the interesting discussions we had together that have clarified difficult points and introduced new ideas in this work.

I would like to thank all the people I met during my stays at ESTEC, for their help and friendship. I would like to thank Harrie Rooijackers for helping me optimize my source code; James Etchells for his explanations about finite elements; Duncan Gibson for his help when I was looking for archives; Philippe Poinas for his discussions and friendship; Elena Checa for giving me access to the XEUS model which was used to illustrate the algorithms developed during my PhD; Simon Appel for helping me integrate the model into my algorithm. And I am grateful to all the other people I met and I forgot to include in these acknowledgements.

I would like to express my gratitude to Tom van Eekelen, Didier Granville and Guy Robert, for giving me access to the Samcef source code. I am particularly grateful to Tom van Eekelen for his technical support.

I would like to thank the Alstom company that markets the Esarad-Esatan software for providing a license.

This work has been supported by the Belgian National Fund for Scientific Research (FNRS) and under ESA Contract 20180/06/NL/PA.

My sincere acknowledgements go to my office mates and their friendship during these four years at the Department of Aerospace and Mechanics and at the European Space Research and Technology Centre: Loïc Stefanski, Fabien Poncelet, Pablo Ibáñez Tarragó and Jihong Zhu, Karim Haddad and Jérôme Montigaud. I also thank my friends Andreas Arnold-Bos and Daniel Reese, for their encouragement.

Finally, I want to thank my wife and my family for support and encouragement they have given me throughout this work. Without their help during moments of discouragement, this work would not have succeed.

Contents

Nomenclature

Capital letter

A, B, C	Components of the local normal	
A_i	Area of patch P_i	$[m^2]$
B_{i-j}	Gebhart, absorption, radiative exchange factor	$[-]$
$C_\odot$	Solar constant	$1\,371 W/m^2$
$\mathcal{C}_i$	Contour of patch P_i	
C_i	Capacitance	$[J/K]$
$\mathcal{C}_u$, $\mathcal{C}_v$	Surface iso parametric curves	
$\mathcal{D}$	Surface parametric domain of definition	
D	Signed distance between plane and 3D origin	$[m]$
E	Emissive heat flux	$[W/m^2]$
	Expected value	$[-]$
E_b	Black body emissive heat flux	$[W/m^2]$
F_{di-j}	Point wise view factor	$[-]$
$\mathcal{F}_{i-j}$	Exchange factor	$[-]$
F_{i-j}	View factor	$[-]$
F^s_{i-j}	Specular, extended view factor	$[-]$
G_i	Global view factor	$[-]$
GL	Conductive coupling	$[W/K]$
GR	Radiative coupling	$[m^2]$
H	Incoming radiation	$[W/m^2]$
	Height of cylinder	$[m]$
$\mathcal{H}$	Objective function	$[-]$
H_0	External radiation	$[W/m^2]$
I_i	Surface diffuse importance	$[m^2]$
J	Thermal radiosity	$[W/m^2]$
K	Accuracy parameter	$[-]$
	Thermal conductivity matrix element	$[W/K]$
L	Radiance	$[W/m^2/std]$
$\mathcal{L}$	Likelihood function	$[-]$
L^d	Diffuse component of radiance	$[W/m^2/std]$
M	Resolution of the hemisphere	$[-]$
N	Number of rays, of nodes, of elements	$[-]$

$N(r)$	Shape function	$[-]$
$\mathcal{P}$	Probability	$[-]$
P_i	Patch i	$[-]$
$\overrightarrow{P_R}$	Origin of the ray	
Q	Volumic heat source term	$[W/m^3]$
QA	Albedo absorbed power	$[W]$
QE	Planetary infrared absorbed power	$[W]$
QI	Internal dissipated power	$[W]$
QS	Solar absorbed power	$[W]$
Q_{i-j}	Net radiative heat power	$[W]$
R	Radius (of cylinder, sphere or disc)	$[m]$
$R_{\oplus}$	Radius of the Earth	$6\,378km$
$R_{\odot}$	Solar radius	$696\,000km$
$\mathcal{R}$	Reflection operator, transport operator	
$\overrightarrow{R}$	Point on a ray	
$\mathcal{R}^d$	Operator of diffuse reflection	
R_i	Importance receptor term	$[m^2]$
$\mathcal{R}^s$	Operator of specular reflection	
$\mathcal{R}^{s*}$	Operator of multiple specular reflections	
$\mathcal{S}$	Boundary of control volume	
S_i	Radiosity source term	$[W/m^2]$
$\mathcal{S}_q$	Boundary subject to heat flux conditions	
$\mathcal{S}_T$	Boundary subject to constant temperature	
T	Absolute temperature	$[K]$
$T_{\oplus}$	Temperature of the Earth	
$T_{\odot}$	Solar temperature	$5\,780K$
T_0	Reference temperature	$[K]$
$\mathcal{V}$	Control volume	
$V_{\odot}$	Local solar visibility	$[-]$
V_{i-j}	Visibility function	$[-]$
W_k	Weight of integration point k	$[-]$

Lower case letter

a	Planetary albedo reflection coefficient	$[-]$
c_0	Velocity of light in vacuum	$2.998\,10^8 m/s$
c	Velocity of light	$[m/s]$
	Specific heat	$[J/kg/K]$
	Geometrical linear parameter	
d	Error linear parameter	
	Distance Sun-Earth	$149\,600\,000km$
dA_i	Elementary area, point	$[m^2]$
$d\omega_{i-j}$	Solid angle of dA_j from dA_i	$[std]$

g_i	Sum line constraint	$[m^2]$
h	Height of intersection point (cylinder, sphere)	$[m]$
	Height of a satellite	$[km]$
k	Thermal conductivity	$[W/m/K]$
n	Refractive index	$[-]$
$\overrightarrow{n}$	Surface normal	$[-]$
$\overrightarrow{n_\Pi}$	Normal vector of plane Π	$[-]$
n_{lat}	Hemisphere's resolution in latitude	$[-]$
n_{lon}	Hemisphere's resolution in longitude	$[-]$
p_i	Shadow ratio	$[-]$
	Probability distribution function	$[-]$
q	Radiative heat flux	$[W/m^2]$
r	Distance	$[m]$
	Radius of intersection point (disc)	$[m]$
$\overrightarrow{r_d}$	Direction of a ray	
$\overrightarrow{r}$	Position	$[-]$
$\overrightarrow{r_i}$	Incident direction	$[-]$
$\overrightarrow{r_r}$	Reflected direction	$[-]$
t	Distance to the intersection point	$[m]$
u, v	Coordinates of impact point	$[-]$
$v(J)$	Radiative energy	$[W]$
x, y, z	3D axes	

Greek capital letter

$\Delta\mathcal{R}$	Error of the transport operator	$[-]$
ΔVF	Delta view factor	$[-]$
$\Delta\phi$	Angular aperture in longitude	$[rad]$
$\Delta\theta$	Angular constant in latitude	$[rad]$
Ω_j	Solid angle subtended by A_j	$[std]$

Greek lower case letter

α	Absorptance	$[-]$
	Confidence interval	$[-]$
β	Angle Sun-Planet-Spacecraft	$[rad]$
γ	Latitude of the sphere	$[rad]$
$\gamma_\odot$	Solar angular aperture	$[rad]$
γ_P	Planetary angular aperture	$[rad]$
ϵ	Emittance	$[-]$
	Maximum relative error	$[-]$
η_{i-j}	Estimator of exchange factor $\Omega_i\mathcal{F}_{i-j}$, from ray tracing	$[m^2]$
$\widetilde{\eta}_{i-j}$	Exact value of $\Omega_i\mathcal{F}_{i-j}$	$[m^2]$
$\dot{\eta}_{i-j}$	Estimator of $\Omega_i\mathcal{F}_{i-j}$, after reciprocity enforcement	$[m^2]$
$\ddot{\eta}_{i-j}$	Estimator of $\Omega_i\mathcal{F}_{i-j}$, after closure enforcement	$[m^2]$

θ	Latitude of the hemisphere	$[rad]$
κ	Reciprocity weight	$[-]$
λ	Wavelength	$[m]$
λ_i	Lagrange multiplier	
$\lambda_{\max}$	Dominant wavelength	$[m]$
ν	Frequency	$[Hz]$
ξ_1, ξ_2	Uniform random numbers	$[-]$
ρ	Reflectance	$[-]$
	Density	$[kg/m^3]$
ρ^d	Diffuse reflectance	$[-]$
ρ^s	Specular reflectance	$[-]$
σ	Stefan-Boltzmann constant	$5.67\,10^{-8} W/m^2/K^4$
	Variance	$[-]$
τ	Transmittance	$[-]$
ϕ	Longitude of the hemisphere	$[rad]$
χ	Longitude of the sphere	$[-]$
ψ	Parametrization of a surface	
ω	Solid angle	$[std]$
ω_{i-j}	Least square smoothing's weight$[-]$	

Acronyms

AHF	Absorbed Heat Flux	$[W/m^2]$
ASHF	Absorbed Solar Heat Flux	$[W/m^2]$
AU	Astronomical Unit ($149.598\ 10^6 km$)	
BC	Boundary Conditions	
BRDF	Bidirectional Reflectance Distribution Function	
CAD	Computer Aided Design	
CLRT	Cosine Law Ray Tracing	
CNES	Centre National d'Études Spatiales	
CPU	Central Processing Unit	
CSG	Constructive Solid Geometry	
DHF	Direct Heat Flux	$[W/m^2]$
DSC	Detector SpaceCraft	
DSHF	Direct Solar Heat Flux	$[W/m^2]$
ESA	European Space Agency	
ESTEC	European Space Research and Technology Centre	
EVF	Extended View Factor	$[-]$
FCI	Fractional Confidence Interval	
FEM	Finite Element Method	
FV	Fractional Variance	
GL	Linear conductor (Esatan formulation)	$[W/K]$
GM	Gebhart (matrix) Method	
GR	Radiosity, radiative conductor (Esatan formulation)	$[m^2]$
HF	Heat Flux	$[W/m^2]$
IXO	International X-ray Observatory	
LSS	Least-Square Smoothing	
MCRT	Monte Carlo Ray Tracing	
MLE	Maximum Likelihood Estimation	
MSC	Mirror SpaceCraft	
PDF	Probability Distribution Function	
PV	Personal Variation (by PV)	
REF	Radiative Exchange Factor	$[-]$
RMS	Root Mean Square	
SAC	Statistical Accuracy Control	

TCS	Thermal Control (Sub)System	
TLP	Thermal Lumped Parameter method	
TMM	Thermal Mathematical Model	
TOP	Thermo-Optical Properties	[–]
USD	Uniform Spatial (Sub-)Division	
VF	View Factors	[–]
XEUS	X-ray Evolving Universe Spectroscopy	
XRO	X-Ray Observatory	

Chapter 1

Objectives, requirements and structure of the algorithm

Contents

The aim of this Chapter is to present the background of the thesis and to detail the objectives that the proposed algorithms must fulfil. These objectives are then expressed as a list of requirements.

An overview of the different aspects that have to be addressed by the thesis is given. Different components of standard spacecraft thermal analysis are detailed, highlighting the elements which received a personal contribution in this thesis; these components have been developed as basic building blocks of the full thermal analysis.

Then the structure of the proposed algorithm is presented; the different parts are briefly described and the innovative aspects developed in the context of this thesis are referenced.

1.1 Background

In order to design the thermal control system of a space mission, the thermal engineer often uses dedicated software, like Esarad and Esatan from ALSTOM or Thermica and Thermisol from EADS ASTRIUM. As the radiative component can be predominant, software is very

often based on Monte Carlo ray tracing to compute the energetic exchanges between the surfaces which compose the geometrical model, as well as the external heat fluxes on the spacecraft, and thermal radiation to the cold deep space.

Currently, with the increasing performance of computers, the geometrical models developed with computer aided design tools contain more details and are constituted of many thousands radiative faces. With the emergence and development of finite element methods in most engineering disciplines, the geometrical models can be defined as a set of tens of thousands of elements (generally first- or second-order triangles and/or quadrangles). Ray tracing algorithms must run faster than today's codes to handle these numbers of faces.

Today, the radiative heat transfer is computed in two spectral bands: one for the solar, visible light and one for the infrared radiation. However, the development of cryogenics applications, such as the scientific observatories Planck and Herschel, requires the decomposition of the spectral domain in more wavelength bands.

The Monte Carlo processes used in thermal software introduce random noise in the obtained results. The convergence is a function of the number of samples (in our case, the number of traced rays). Another problem is the fact that the accuracy level is not well controlled by the thermal engineer.

Currently, we observe a strong trend towards increased use of the finite element formulation in addition to the classical use of the lumped parameter (finite difference) formulation for space thermal analysis. This could allow us to better represent complex geometries with fewer elements, as well as to use finite element shape functions for the temperature field (versus isothermal patches, used in the thermal lumped parameter formulation). Another interest is the automatic generation of the conductive couplings, which is not yet accessible and reliable. To deal with finite elements, most of the radiative software for space analysis requires special attention.

On the basis of these observations, we have defined the objectives that the thesis should fulfill.

1.2 Objectives

To handle the increasing size of the geometrical models based on finite element meshes, our first objective is to develop an algorithm that achieves an order of magnitude improvement in performance (speed) over existing Monte Carlo ray tracing techniques used for space thermal analysis. The algorithm shall present a good scalability, from small to large models.

To optimize multi-wavelength bands computations, significant changes of the used thermal formulation must be implemented, as well as an optimized algorithm to save CPU time

and storage. While the thermal equations are well known, an adapted, smart strategy in the ray tracing process can save a large amount of resources.

In order to deal with the stochastic nature of the ray tracing process, one contribution of this work is to propose a new way to generate the rays, characterized by a convergence superior to existing Monte Carlo ray tracing algorithms. This method is fitted with a statistical accuracy control. The thermal engineer can require a desired accuracy (defined by a maximum relative error ϵ and a convergence interval α) and the algorithm automatically adapts the number of rays to the considered geometrical configuration.

The development of finite element methods and their introduction in almost all engineering disciplines lead us to find a way to extend the thermal software to a finite element formulation. Moreover, the ray tracing process must also be modified to fit the finite element formulation.

This thesis contains the theoretical developments necessary to achieve these objectives. We present also the development of the prototype algorithm and the testing process.

1.3 Requirements

On the basis of the objectives presented in the previous section, requirements that an optimal algorithm should fulfil have been defined. The final algorithm:

1. shall be a ray tracing algorithm, where each ray is associated with a given energy and can be considered as a bundle of photons.

2. shall compute the geometrical view factors and the radiative exchange factors. In case of specular reflection and/or transmission, the algorithm shall also compute the extended view factors. The algorithm shall also compute the in-orbit external heat fluxes.

3. shall be mathematically well-behaved, *i.e.* the final results shall converge with increasing number of casted rays.

4. shall include diffuse reflection, specular reflection, glossy reflection.

5. shall be capable to support two spectral bands (one for the visible radiation and one for the infrared component).

6. shall support either a finite element formulation or a lumped parameter formulation, also called finite difference formulation (in order to compute the overall thermal behaviour) and shall be adapted both for finite element models (containing volumes and surfaces) imported from CAD and geometrical primitives (included primitives are triangles, rectangles, quadrangles, discs, cylinders, cones, spheres and paraboloids).

7. shall provide a statistical accuracy report accompanying the results in order to give a confidence interval to the user and shall provide a statistical accuracy control to the user. As a function of a user-specified accuracy, the algorithm shall be capable of modifying locally the number of casted rays where the accuracy is not achieved.

8. shall be able to correct the computed exchange factors in function of their associated variance.

9. should achieve at least an order of magnitude improvement in performance over existing Monte Carlo ray tracing techniques used for space thermal analysis. The comparative software shall be Esarad. Comparisons shall also be performed with the finite element software Samcef.

1.4 Spacecraft thermal analysis

Figure 1.1 presents the different steps of a satellite thermal analysis. The innovative aspects of the thesis are highlighted in red and/or in orange (in bold and/or italic); red (italic) corresponds to theoretical developments and new mathematical formulations, while orange (bold) represents the algorithmic developments, implying prototyping, and illustrated by computer results.

The first module corresponds to the definition of the satellite. A geometrical model is defined using a CAD tool. The geometry can also be represented by a finite element triangulation. Thermo-optical properties (TOP) and boundary conditions (BC) shall also be defined.

Once a geometrical model is available, a ray tracing process, corresponding to the second module of Figure 1.1, is necessary to compute the exchange factors (view factors, noted VF, or radiative exchange factors, noted REF) and the external heat fluxes (direct incident, noted DHF, or absorbed ones, noted AHF). A ray tracing algorithm consists in generating a large number of rays, usually from random origins which are uniformly distributed across the surfaces of the model. A ray tracing is also based on the computation of ray-surfaces intersections.

The ray tracing process can be very time consuming if no precaution is taken. This is the objective of the ray tracing acceleration (orange block, in italic, on the right) which is one of the most important parts of the ray tracing algorithm. A classification of different ray tracing acceleration techniques is given at the beginning of Section 4.3.

Once the ray tracing process is finished, we have to correct the exchange factors (third green module). The exchange factors obtained by ray tracing do not respect the reciprocity relation. A reciprocity enforcement has to be coded. Then the exchange factors do not respect the unity relation anymore. This problem has to be addressed before solving the

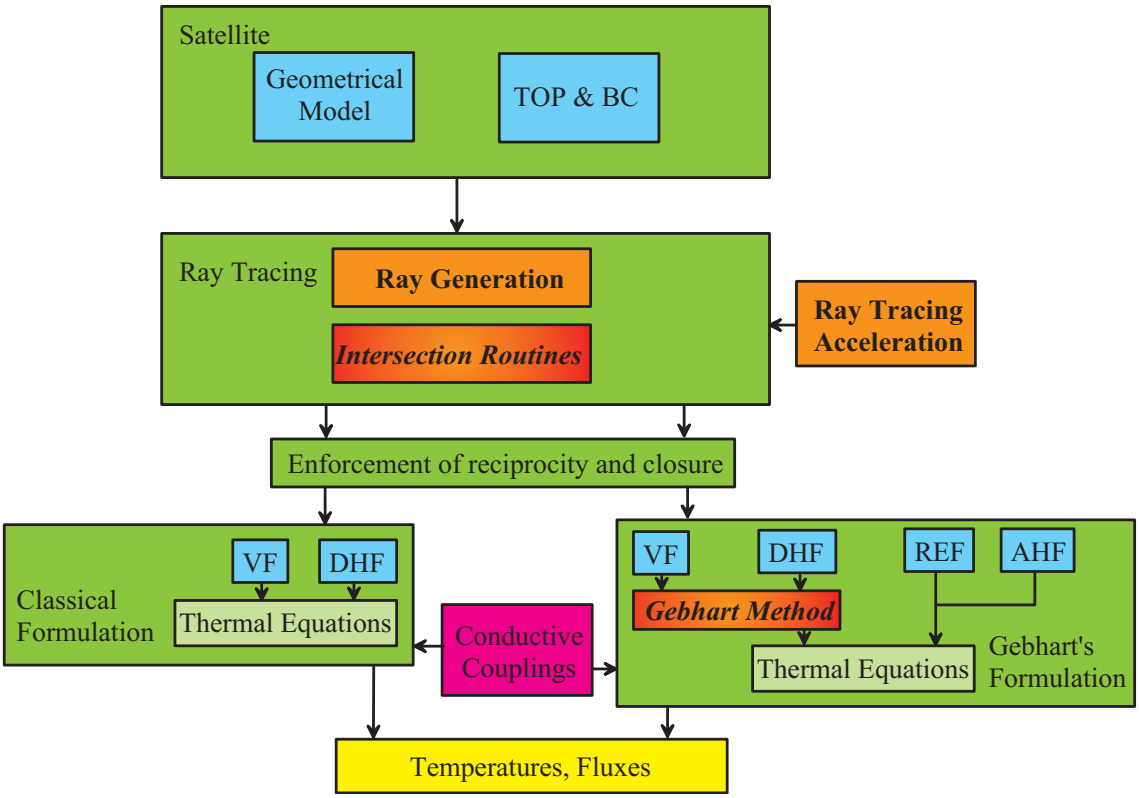


Figure 1.1: Satellite thermal analysis.

thermal problem.

In order to solve the thermal problem, the conductive links have to be computed. This operation can be performed by an external module, represented by the magenta block. In current thermal software, this is usually carried out by the thermal engineer himself; however, procedures for automatic computation of conductive links appear in existing software.

Once the exchange factors and the heat loads have been computed, the thermal problem can be solved. In general, it can be performed thanks to two different formulations. A first one is based on view factors and direct heat fluxes. The second one is the most commonly used in thermal software and is based on radiative exchange factors and absorbed heat fluxes. If the exchange factors produced by the ray tracing process are the view factors, Gebhart's matrix method can be used under some hypotheses to obtain the radiative exchange factors.

Once all these steps have been fulfilled, the satellite thermal analysis is completed and the engineer has access to the distributions of temperature and heat flux throughout the geometrical model, represented in yellow in Figure 1.1.

1.5 Structure of the algorithm

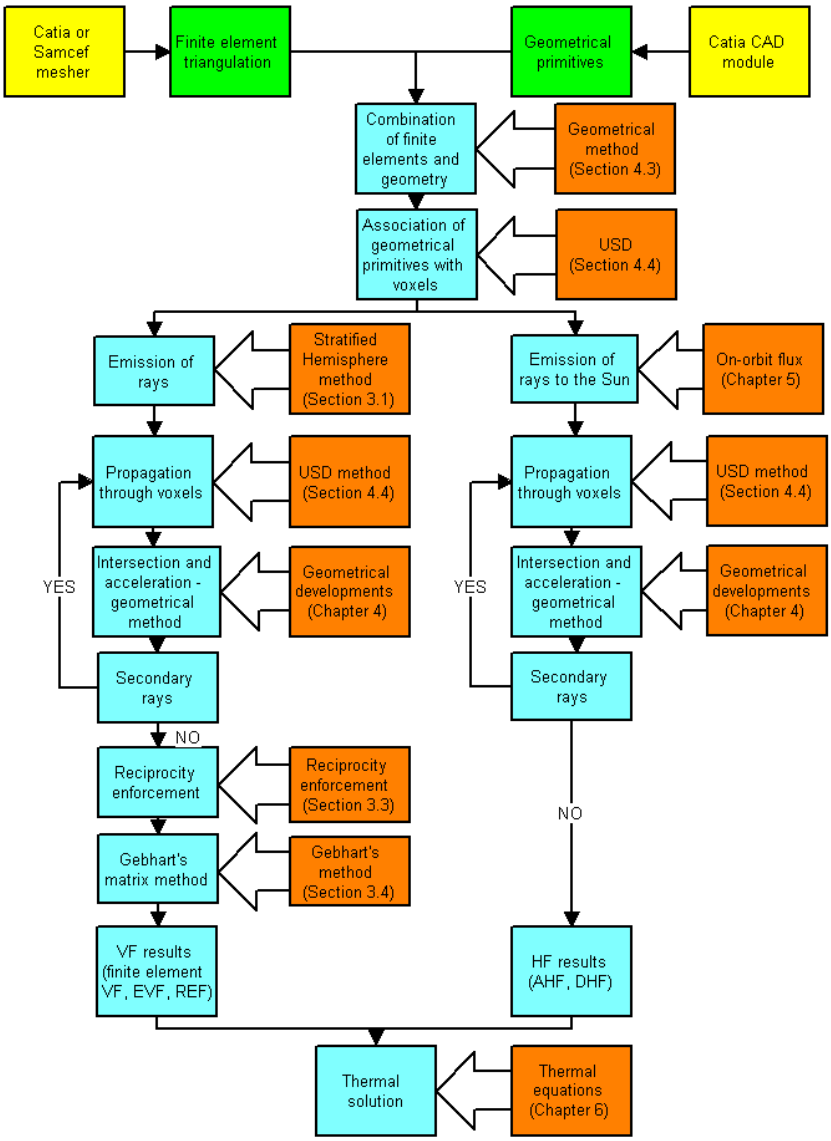


Figure 1.2: Structure of the algorithm.

Figure 1.2 shows the structure of our algorithm. Some modifications can occur from one version to the other. To fulfill the requirements, a set of modules has been developed. These modules can be combined in different ways, depending on the objective. The green blocks correspond to the main entries of the algorithm (geometrical primitives, associated finite element meshes, thermo-optical properties...); additional entries should be defined, such as the solar direction and angular aperture, the position and attitude of the spacecraft, but they have not been displayed in Figure 1.2 for the clarity. The yellow blocks correspond to external software used to generate the entries of our algorithm; the geometry can be

obtained from CAD software Catia while the finite element meshes can be given by the mesh-modules of Samcef or Catia. The blue blocks represent the different steps of the algorithm and are associated to the corresponding personal developments, corresponding to Chapters or Sections of this thesis and schematized by the orange boxes.

1.5.1 Pre-ray tracing operations

Our algorithm is based on the association of the geometrical model with a corresponding finite element mesh. This association between two definitions of the model has been used to implement a new ray tracing acceleration method, which is called the *geometrical method* and is detailed in Section 4.3.

This is combined with another acceleration method, called *uniform spatial subdivision* (USD), based on the decomposition of the bounding box of the model into sub-volumes, called *voxels*. This two-level acceleration method is used for both exchange factors (left part of the diagram) and heat fluxes (right part of the diagram).

1.5.2 Ray tracing process

Our ray tracing for exchange factors (left column of the diagram) is based on a particular ray generation, called the *hemisphere method*, presented in Section 3.1; this method is characterized by optimal statistical properties and a convergence rate superior compared to existing methods.

The ray tracing acceleration is the combination of two methods (the geometrical method and the USD method); the intersections are based on optimized routines (see Chapter 4).

The second part of the ray tracing, *i.e.* the computation of heat fluxes, is based on the generation of shadow rays (see Chapter 5). The ray tracing acceleration is the same as the one used for exchange factors, described in Chapter 4.

1.5.3 Post-ray tracing operations

Once the ray tracing process is finished, the reciprocity of the exchange factors has to be enforced. In Section 3.3, different techniques are compared in order to identify the one which will be used in the proposed algorithm.

Before the resolution of the thermal problem, the other heat transfer modes, especially conduction, have to be taken into account. A very easy way to do it is based on the geometrical method, by using the underlying finite element mesh to automatically compute the conductivity matrix (see Section 4.3). The mathematical formulation is given in Section 6.1 and a large-scale application can be found in Chapter 7.

If the ray tracing has been used to compute view factors and if the Gebhart's formulation is chosen to solve the thermal problem, Gebhart's matrix method can be used to derive the radiative exchange factors from the view factors. In Section 3.4, the hypotheses of the classical method are extended to consider non-diffuse reflection and non-isothermal surfaces.

As a post-processing step, our algorithm can deliver an accuracy report, characterizing all individual links between the surfaces. This energy accuracy measure is an innovative part of the complete algorithm. It is based on a geometrical accuracy measure, presented in Section 3.1.6 and is detailed in Section 6.2, as a mathematical development.

1.6 Structure of the thesis

The thesis comprises eight chapters.

The second chapter provides the necessary background related to the topic addressed in the thesis so that the reader can pursue the reading with all basic notions in mind.

Chapter 3 describes the computation of the first radiative quantity: the exchange factors. An optimized emission process, named hemisphere, is studied in detail in Section 3.1, in the case of diffuse surfaces. The case of more complex thermo-optical properties is considered in 3.2. The exchange factors must be enforced for reciprocity and closure. It is carried out in Section 3.3. On the basis of the view factors computed by ray tracing, the Gebhart's matrix method is used to derive the radiative exchange factors. In Section 3.4, this method is extended to take into account complex thermo-optical properties and non-isothermal elements.

Chapter 4 is dedicated to geometrical developments used to accelerate the ray tracing process, needed to compute the exchange factors and the radiative heat fluxes. Ray-surface intersections are optimized in Section 4.2. A new ray tracing acceleration method, adapted to the constraints of thermal analysis for space applications, is presented in Section 4.3.

Chapter 5 is dedicated to the computation of the external solar heat flux. This computation is adapted to the acceleration method presented in Chapter 4.

Chapter 6 contains all the innovative mathematical developments performed in the framework of the thesis.

Chapter 7 presents the results of the validation process.

Finally, Chapter 8 presents the conclusions and perspectives, as well as recommendations for further research.

Chapter 2

Heat transfer for space applications

Contents

This Chapter describes in general terms the space thermal analysis discipline, why it is needed for the development of space systems and how it interacts with the other disciplines needed for the successful design of space systems. It contains notions of heat transfer required for the thesis, as well as a description of the kind of discrete numerical models that are used including the relevant physical quantities, and the typical sequence of analysis steps. The difficulties that must be overcome to perform a complete space thermal analysis task are also explained. The Chapter ends with the establishment of the two main mathematical formulations that can be applied to solve the thermal problem, based on the theoretical notions introduced in this Chapter.

2.1 Introduction - Objective of the thermal control subsystem

The purpose of the thermal control subsystem is to maintain suitable temperatures for all components of a space system (satellite, space probe, manned vehicle) under operating conditions in all of the thermal environments encountered during the mission [CNE05, Gil02, Ale02, Roc03]. To maintain appropriate temperatures is necessary to insure the optimal status of the whole system and is performed by the so-called *thermal control subsystem*. The thermal control engineer should create a design that manages and distributes the heat

in such a way that the system can perform according to its intended purposes. This involves taking both passive and active measures. Examples of passive design measures are the selection of materials and coatings with appropriate properties (thermal conductance, heat capacity, expansion coefficient, emittance, absorptance, reflectivity...) and behaviour over time, as well as adequate shapes and 3D configuration, including radiator surfaces to reject heat to cold deep space. Examples of active design measures are electric heaters with a temperature sensor and a controller, Peltier elements, fluid loops, etc. Sometimes also the trajectory or orientation of a spacecraft are adapted in order to obtain favorable thermal conditions.

The tasks of the engineer are to design and verify a thermal control subsystem, which insures that each component remains in a given (operating or survival) temperature range, in function of the mission scenario. These tasks are usually performed through dedicated thermal analysis software. The use of thermal software is necessary during the early phases of the development in order to design an adapted, efficient thermal control subsystem. As the development progresses and a design is selected, the software is used to simulate tests that will be carried out with the actual hardware (e.g. in vacuum chamber) and produce test predictions. The software is also used to compute the thermal behaviour of the space system during its mission (to produce so-called flight predictions).

The physical tests are expensive, and it is complex to create all the conditions experienced by the system during its life. In this context, the interest of a reliable prediction of the thermal behaviour of the system, based on computer simulations, is clear. The experimental tests are performed to verify the functionality and performance of the built system as well as to verify the computer simulations and to correct the virtual thermal model, if necessary. Once the virtual model is reliable, it can be used to simulate conditions that are not tested or cannot be created in a test facility.

This Chapter provides a detailed discussion of how a thermal analysis for space applications is performed and what software is used. In order to establish a reliable model of a thermal system, we need to introduce some basic notions of heat transfer. The radiative heat transfer will be carefully addressed, and the definition of the thermo-optical properties will be given. Based on these preliminary notions, we will review the complete design process of the mathematical model, from the geometrical definition and material selection to the resolution of heat transfer, using the *thermal lumped parameter* method, which is the discretisation/modelling method most commonly used in space thermal engineering.

2.2 Notions of heat transfer and radiation

Heat can be exchanged in three different heat transfer modes, depending on the participating medium:

1. *conduction*, through a solid medium;

2. *convection*, where the transfer is performed by a fluid in motion;

3. *radiation*, which does not require any medium to insure the heat transfer.

In this work, we consider unmanned vehicles, like satellites and exploration probes. In the vacuum of space, no convection appears. The heat transfer is then governed by conduction and radiation. Generally, the main heat sources are the Sun and other celestial bodies like planets and moons, as well as internal dissipation. The main heat sink is the cold deep space, at approximately $5K$. It implies that radiative heat transfer plays a major role for space applications. This is why the present thesis is dedicated to radiation. No conduction will be addressed in this theoretical Chapter.

Radiation relates to the exchange of energy due to the emission and absorption of electromagnetic radiation. Thermal radiation can either be considered as electromagnetic waves or bundles of photons [CT97]. The second definition is used by ray tracing, where we model radiation as rays of light. The propagation of the rays is only based on the laws of geometrical optics. Thermal radiation is characterized by a velocity c, equal to the *velocity of light*. This velocity varies as a function of the encountered medium. The velocity of light in vacuum is equal to:

$$c_0 = 2.998\,10^8 m/s \tag{2.1}$$

The velocity of light in a medium characterized by a *refractive index* n is given by the relation:

$$c = \frac{c_0}{n} \tag{2.2}$$

The frequency ν of an electromagnetic wave is invariant with the refractive index while c and wavelength λ vary with n. The frequency is linked to the wavelength λ through the formula:

$$c = \lambda\nu \tag{2.3}$$

Thermal radiation is a part of the electromagnetic wave spectrum, as it can be seen in Figure 2.1. The thermal spectrum (in red in Figure 2.1) is defined from $0.1\mu m$, in the ultraviolet domain, to $100\mu m$ in the infrared domain, including the visible spectrum (from 0.4 to $0.7\mu m$).

2.2.1 Black body emissive power

A body emits electromagnetic radiation as a function of the local temperature and the material properties. In order to have a reference thermal source, scientists defined an ideal body, called *black body*, which completely absorbs any incident radiation.

The emissive power E $[W/m^2]$ of a black body at temperature T, expressed in Kelvin, is given by *Stefan-Boltzmann's formula* [Mod03]:

$$E = \sigma T^4 \tag{2.4}$$

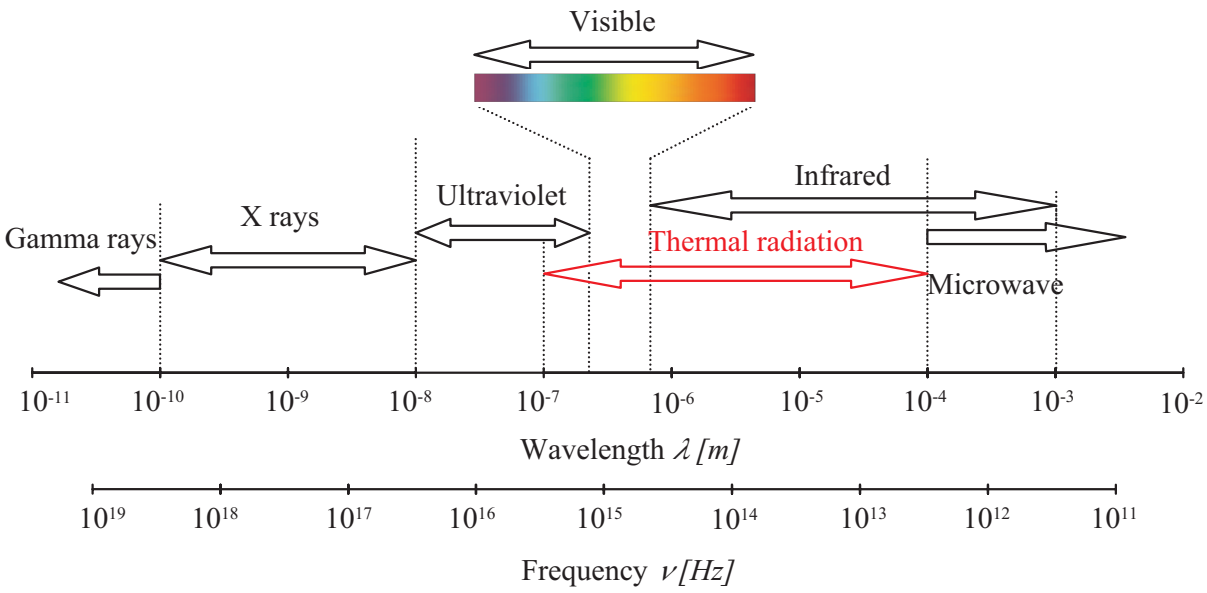


Figure 2.1: Electromagnetic wave spectrum.

where σ is the Stefan-Boltzmann constant, equal to:

$$\sigma = 5.67\,10^{-8} W/m^2/K^4 \tag{2.5}$$

The emissive power is characterized by a continuous spectral distribution. The dominant wavelength of the emissive power is a function of the temperature, known as *Wien's displacement law*:

$$\lambda_{\max} = \frac{2\,898}{T} \tag{2.6}$$

where $\lambda_{\max}$ is expressed in μm and T in K.

Figure 2.2 represents the evolution of the black body emissive power as a function of the wavelength λ, for five different temperatures, together with the evolution of Wien's displacement law (black dashed line). An increase in temperature yields an increase in the black body emissive power as well as a decrease in the main wavelength $\lambda_{\max}$.

2.2.2 Emittance

The emissive power of a surface, at a given temperature, is compared to the black body at the same temperature. The emittance ϵ is defined as the ratio of the real emissive power and the emissive power of the black body. So the emissive power of the real surface is given by:

$$E = \epsilon \sigma T^4 \tag{2.7}$$

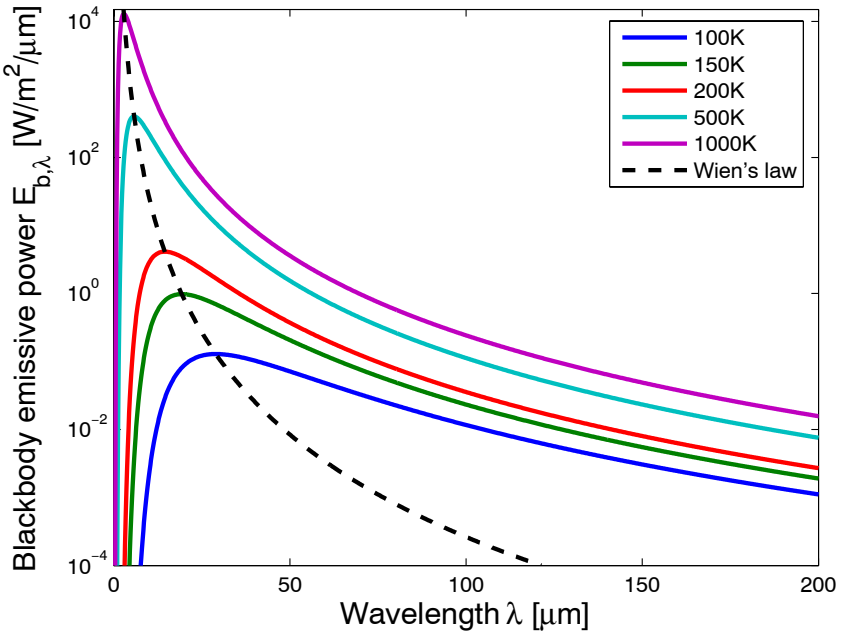


Figure 2.2: Black body emissive power spectrum.

As the maximum possible emissive power is emitted by a black surface, at a given temperature, the emittance is defined between 0 and 1. For a black surface, we consider that the emittance is equal to 1. The emittance of a surface is a function of the wavelength, the temperature, and the surface roughness.

2.2.3 Absorptance - reflectance - transmittance

Real surfaces cannot be considered as black ones. A real surface cannot absorb all the incident radiations (see Figure 2.3). A fraction ρ of the incident radiation is reflected. If we consider a non-opaque medium of finite thickness, the rest of the incident radiation is transmitted through the slab. The following thermo-optical properties are defined [Mod03]:

- the absorptance α is defined as the fraction of the incoming radiation which is absorbed;
- the reflectance ρ is the fraction of the irradiation which is reflected in the incoming hemisphere;
- the transmittance τ is the fraction of the radiation which is transmitted through the slab.

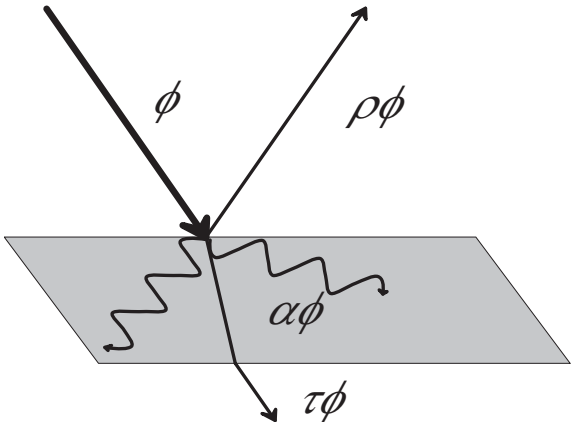


Figure 2.3: Absorption, reflection and transmission.

The conservation of energy yields the following relation, where ϕ is the energy of the incoming radiation:

$$\alpha\phi + \rho\phi + \tau\phi = \phi \tag{2.8}$$

$$\alpha + \rho + \tau = 1 \tag{2.9}$$

If the medium is opaque, the corresponding transmittance τ is equal to zero, and the previous relation becomes:

$$\alpha + \rho = 1 \tag{2.10}$$

The reflection of radiation on a surface is governed by its bidirectional reflection function ρ. It is a complex function of the incoming and outgoing directions, and of the wavelength of the incoming radiation. Usually, two different reflections are considered in heat transfer [Mod03] as well as in rendering [WC87]: diffuse reflection and specular reflection. A specular surface is a surface which reflects an incoming radiation following the optic laws (see Figure 2.4–a). The specular reflectance ρ^s is defined as the fraction of the incident radiation which is specularly reflected. A diffuse surface reflects an incoming radiation in all directions (see Figure 2.4–b for the case of a purely diffuse surface following the *Lambert's law* or *cosine law*). The diffuse reflectance ρ^d is the fraction of the incident radiation which is diffusely reflected.

Equation (2.9) can be rewritten as follows:

$$\alpha + \rho^d + \rho^s + \tau = 1 \tag{2.11}$$

These four radiative properties are non-dimensional; they are defined between 0 and 1. A black surface is associated with a unitary absorptance, zero reflectance, and zero transmittance. They constitute the thermo-optical properties of the surface.

(a) Specular surface (b) Diffuse surface

Figure 2.4: Reflection processes.

2.2.4 Kirchhoff's law

Kirchhoff's law is based on thermodynamic equilibrium. Following Kirchhoff's law, for a given wavelength λ, a given polarization, noted $\pm$, and a given direction, identified by the angles θ and ϕ, the absorptance and emittance are equal [Mod03]:

$$\alpha_{\lambda_\pm,\theta,\phi} = \epsilon_{\lambda_\pm,\theta,\phi} \tag{2.12}$$

This relation can be integrated into a given wavelength band and solid angle.

2.2.5 Gray surface

While a black surface is an ideal, unreal surface defined for reference purposes, a more common hypothesis for real surfaces is the gray approximation: we consider that the spectral emittance ϵ_λ is constant on the whole spectral range. No real surface is truly gray. However, it often happens that ϵ_λ is relatively constant over that part of the spectrum where $E_{b,\lambda}$ is substantial, making the simplifying assumption of gray surface acceptable [Mod03].

2.2.6 Exchange factors

The thermal problem for space applications is often solved with the thermal lumped parameter method. This implies a discretization of the geometrical model. The surfaces of the components are also meshed. In order to model the radiative heat transfer, additional links between the surfaces must be added. These links are built on the basis of diffuse view factors, extended view factors and radiative exchange factors. These three quantities are called *exchange factors*, in a generalized way.

2.2.6.1 View factors

When we consider radiative heat transfers between diffuse surfaces, the transfers of energy are conducted by the geometrical configuration of these surfaces. This geometrical dependency is expressed by a function called *view factor* [SP94]. Other names found in

the literature are angle factor[1], configuration factor and shape factor[2]. The view factor between two surfaces depends only on their sizes, their orientations and the distance between them.

By definition, *the view factor F_{i-j} from a surface i to a surface j is the fraction of the diffuse energy leaving A_i which is directly intercepted by A_j.*

Let us consider two patches P_i and P_j, with areas A_i and A_j. Let us note dA_i and dA_j two elementary surfaces on these two patches. We will first consider the radiative exchanges between elementary surfaces before extending them to finite surfaces. Let us note $d\omega_{ij}$ the solid angle with which dA_j is seen from from dA_i. This angle is defined by the following equation:

$$d\omega_{ij} = \frac{\cos(\theta_j)}{r_{i-j}^2} dA_j \tag{2.14}$$

where $r_{i-j} = ||\overrightarrow{r_{i-j}}||$ is the distance between dA_i and dA_j; θ_j is the angle between the local normal $\overrightarrow{n_j}$ and the vector $\overrightarrow{r_{i-j}}$. The geometrical parameters appearing in these equations are defined in Figure 2.5.

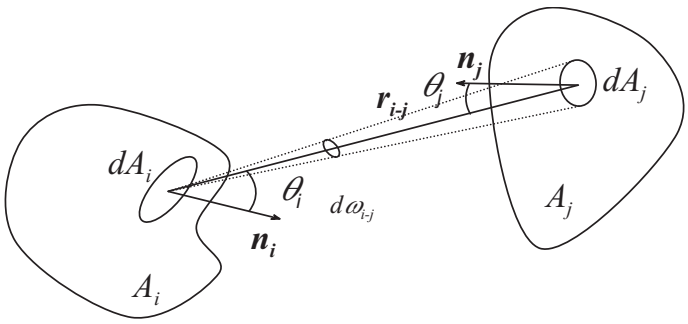


Figure 2.5: View factor - definition of the geometrical parameters.

The total radiative flux emitted by dA_i is given by the following relationship, in the case of a diffuse surface:

$$dQ_i = \pi L_i dA_i \tag{2.15}$$

[1]The term of angle factor is linked to the formula of Lambert [Stü95, NN93, NN86b, NN85], where the view factor is completely defined in terms of the angles of the different surfaces.

$$F_{di-j} = \frac{1}{2\pi} \sum_k \beta_k \cos(\delta_k) \tag{2.13}$$

where β_k corresponds to the angular aperture of the edge k, projected onto the local hemisphere, δ_k is the dihedral angle between the local horizontal plane and the plane defined by edge k and the origin dA_i.

[2]The term shape factor has not been retained in this document because it could induce confusion with the shape functions that will be defined in the framework of the finite element formulation.

L_i is the power emitted by dA_i, in a specified direction, per unit area perpendicular to the direction of travel, per unit solid angle. If the emitter is supposed to be diffuse, L_i is equal in each direction. The flux emitted by dA_i which reaches dA_j is equal to:

$$dQ_{i-j} = L_i dA_i \cos(\theta_i) d\omega_{ij} V_{i-j} \tag{2.16}$$

$$= L_i dA_i \cos(\theta_i) \frac{\cos(\theta_j)}{r_{i-j}^2} V_{i-j} dA_j \tag{2.17}$$

V_{i-j} is the visibility function, equal to 1 if the two points are mutually visible, equal to 0 otherwise. This function allows us to take obstacles into account.

The geometrical view factor from dA_i to dA_j is defined as the ratio of the radiative flux received by dA_j and the total radiative flux from dA_i:

$$F_{di-dj} = \frac{dQ_{i-j}}{dQ_i} \tag{2.18}$$

$$= \frac{\cos(\theta_i)\cos(\theta_j)}{\pi r_{i-j}^2} V_{i-j} dA_j \tag{2.19}$$

The geometrical view factor from A_i to A_j is obtained by computing the double integral onto the two areas. A first integration gives the view factor from dA_i to A_j and the second one, the global view factor.

$$F_{di-j} = \int_{A_j} \frac{\cos(\theta_i)\cos(\theta_j)}{\pi r_{i-j}^2} V_{i-j} dA_j \tag{2.20}$$

$$F_{i-j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos(\theta_i)\cos(\theta_j)}{\pi r_{i-j}^2} V_{i-j} dA_j dA_i \tag{2.21}$$

Equations (2.19)–(2.21) point out that the view factor between two surfaces is purely geometrical. The thermo-optical properties (emittance, reflectance, absorptance) do not appear in these formulas.

The point wise view factor (equation (2.20)) can be seen as the limit of the view factor when the area of the emitter A_i tends to zero. Numerous methods have been designed to compute the point wise view factor, making it very attractive[3]. Nevertheless, available software directly computes the view factors from surfaces to surfaces, usually based on ray tracing approaches. A large number of rays are sent from the emitter's surface (from fixed points on A_i or from points randomly distributed across A_i). Among the N rays, N_j rays will directly intersect receptor A_j. No reflection is taken into account in this process. We

[3]Among these methods, we can mention the Unit Sphere Method [RV04, Jar03], the Lambert's formula (2.13), the hemi cube [CG85]... These methods and many others are detailed in the internal report [Vue05].

only consider the first intersection of each ray. If the rays are distributed following the cosine distribution, the desired view factor is given by the relation:

$$F_{i-j} = \frac{N_j}{N} \tag{2.22}$$

2.2.6.2 Extended view factors

When specular reflectors are present in the model, the previous definition of the view factor must be extended to take into account all the specular paths that the light can follow from the emitter to the receptor [Mod03, SP89].

The specular view factor F^s_{i-j} is defined as the fraction of the diffuse energy leaving i which is intercepted by a receptor j, either by direct travel or after any number of specular reflections.

A specular reflector can be seen as a mirror, a window opening on a virtual world. The surfaces of the virtual world can be seen from the real world through the reflector, attenuated by the specular reflectivity ρ^s of the mirror. This is the basis of the *image method* [RT90].

In the image method, a specular view factor is obtained by considering the surfaces of the real world as well as the surfaces of the virtual world(s) that are visible from the emitter. The images must be attenuated in function of the different specular reflections that the light has experienced. The corresponding specular view factor is obtained by summing the diffuse view factors of all the surfaces, the real ones and the virtual ones. This is illustrated in Figure 2.6. We can observe that a surface j which is not directly visible from i (the corresponding diffuse view factor is equal to zero) can be seen through the surface k, giving a non-zero specular view factor.

The specular view factors can be computed by ray tracing. When a ray is generated, it is followed through the model. When it encounters a first surface, this first intersection represents the diffuse contribution to the view factor. If the intersected surface is (at least partially) specular, a secondary ray is shot in the direction given by the following equation:

$$\overrightarrow{r_r} = \overrightarrow{r_i} - 2\,(\overrightarrow{n} \cdot \overrightarrow{r_i})\,\overrightarrow{n} \tag{2.23}$$

where $\overrightarrow{r_i}$ represents the direction of the incident ray, $\overrightarrow{r_r}$ is the direction of the reflected ray and $\overrightarrow{n}$ is the local normal at the intersection point. The energy associated with the ray is attenuated by the specular reflectivity of the intersected surface. The secondary ray is followed throughout the model, to its next intersection; the corresponding contribution is added to the view factor. If the newly intersected surface is specular, another secondary ray is generated. This process is performed until the energy of the ray goes below a given threshold.

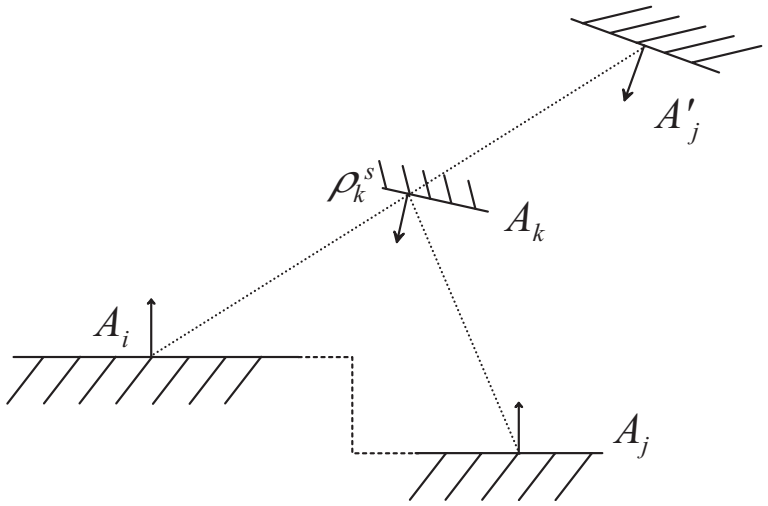


Figure 2.6: Specular view factor - image method.

2.2.6.3 Radiative exchange factors

The radiative exchange factors, also called Gebhart's factors, were introduced in 1957 in [Geb57] cited in [Geb61b]. In Gebhart's formulation, the interactions between the surfaces link directly the self emitted power of these surfaces. This formulation is simpler than the formulation based on view factors. In fact, the difficulty is transferred to the radiative exchange factors, which are far more complex than the geometrical view factors.

By definition, *Gebhart's factor B_{i-j} from a surface i to a surface j is the fraction of the energy leaving A_i which is finally absorbed by A_j, after any number of diffuse and/or specular reflections and/or transmission.*

The emission process is not specified in the previous definition. In this work however, only diffuse emission will be considered.

In the initial article, Gebhart derives the radiative exchange factors from the view factors by the so-called *matrix method* in the case of isothermal, diffuse surfaces. In a dedicated Chapter, it will be extended to more complex situations.

The radiative exchange factors can be computed by ray tracing. A large number of rays are generated from the emitter's surface. When an intersection is detected, a random process determines if the ray is absorbed, diffusely or specularly reflected. In case of reflection, a secondary ray is traced throughout the model.

A drawback of the radiative exchange factors is that they are function of the thermo-optical properties, which can be temperature-dependent. A solution can be to consider gray surfaces (see Appendix A.2), with thermo-optical properties independent of temperature.

If this hypothesis is not valid, a band approximation can be used (see Appendix A.3).

2.3 Thermal model

In this Section, the definition of the spacecraft model is presented, as it is performed with reference program Esarad. It includes the creation of the model on the basis of geometrical primitives, the attributions of the thermo-optical properties and the decomposition of the model. The resulting thermal equations are also given.

2.3.1 Geometrical model

The thermal model is first defined by the geometrical representation of the spacecraft. The boundaries of the model have to be fixed. This model is constituted of macroscopic objects, called components.

The geometrical model has to be adapted to the scale of the studied spacecraft. It is common to simplify the model by removing details as filet, small hole and element.

The geometry is based on geometrical primitives, such as rectangles, planar quadrangles, triangles, discs, spheres, cones, cylinders and paraboloids.

If the boolean operators *union*, *intersection* and *difference* are defined, the geometrical primitives can be combined in order to generate more complex surfaces. This method is known as *Constructive Solid Geometry* [Ath83, Han89]. The three operators are defined as follows:

- the union operator computes the merger of two objects into one;
- the intersection operator yields the portion common to the two objects;
- the difference operator results in the substraction of one object from the other.

An example is shown in Figure 2.7, from [Wik]. It is composed of three cylinders, a sphere and a cube. The union operator is used to combine the three cylinders. The intersection between the cube and the sphere is computed. Finally, the two obtained components are combined with the difference operator. The objects obtained with Constructive Solid Geometry can be represented by binary trees, where leaves represent primitives and nodes correspond to boolean operations.

In Esarad, the geometrical primitives are first decomposed into patches following isoparametric curves. The thermo-optical properties are attributed to the geometrical primitives. Then, the meshed primitives are combined following the constructive solid geometry.

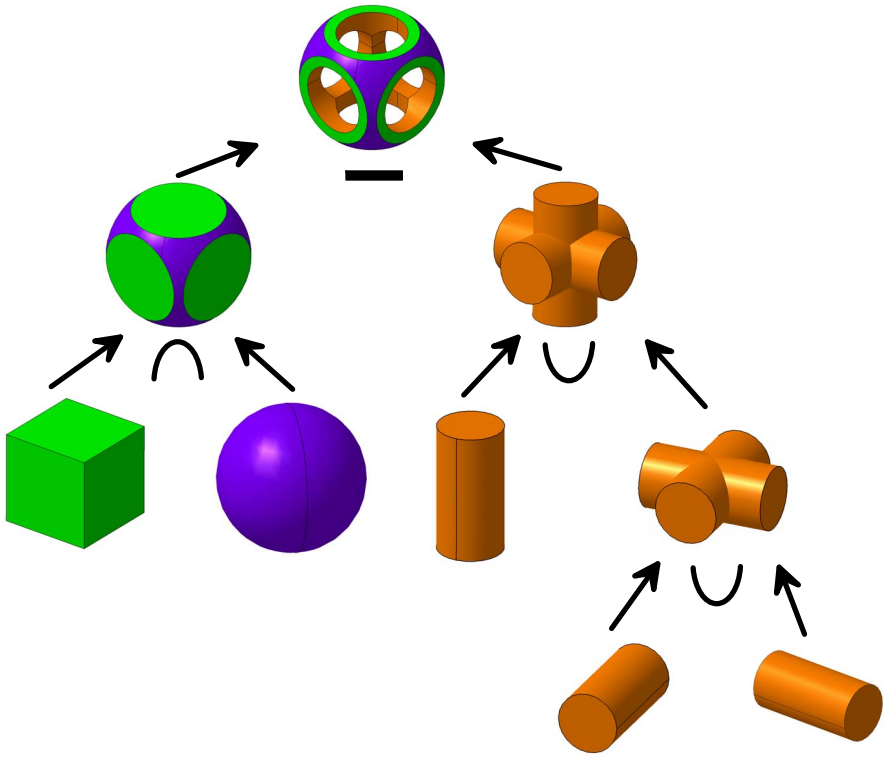


Figure 2.7: Constructive Solid Geometry.

This implies that the intersection of two objects is exactly computed, although this intersection can be complex[4]. The ray tracing is performed on this definition of the model.

2.3.2 Mathematical representation of the thermal control system

As mentioned previously, the common method used in heat transfer for space applications is the thermal lumped parameter method. Esarad-Esatan [Cen03] and Thermica-Thermisol are based on this approach. This method is a means of deriving a first-order finite-difference approximation to the differential heat transfer equation (2.24):

$$\text{div}\left[k\overrightarrow{\text{grad}(T)}\right] + Q = \rho c \frac{\partial T}{\partial t} \tag{2.24}$$

where ρ is the density of the medium, c is the specific heat and k is the thermal conductivity. Equation (2.24) results from the principle of conservation of heat energy over volume $\mathcal{V}$. The first term represents the rate of heat which is conducted into volume $\mathcal{V}$ across its boundary $\mathcal{S}$; it contains the radiative heat fluxes. The source term Q corresponds to the rate of heat generation within the volume. The second member is the rate of increase of

[4]In the method developed in this thesis, it will be shown that, despite the ray tracing is performed on the geometrical primitives, the intersection of two objects is approximated by triangular elements, as it is done in Samcef.

heat in $\mathcal{V}$ [LMTS96].

The thermal lumped parameter method implies a decomposition of the model in isothermal elements called *thermal nodes*. For this reason, the emissive power is constant across each node. The thermal properties of each node are concentrated in one point which can be the geometrical centre of the node. The nodal decomposition also implies that the thermo-optical properties are constant across each thermal node.

The resulting thermal equations express the energetic balance of the model. The power emitted by a node i is equal to the absorbed power plus the internal production. The corresponding equation is given by the following formula, for a stationary problem [Cen03]:

$$QI_i + QS_i + QA_i + QE_i + \sum_{j=1}^{N} GL_{i-j} \left(T_j - T_i\right) + \sigma \sum_{j=1}^{N} GR_{i-j} \left(T_j^4 - T_i^4\right) = 0 \qquad (2.25)$$

where QI is the internal dissipation, QS corresponds to the absorbed solar heat flux, QA represents the absorbed albedo heat flux, QE is the absorbed planetary infrared heat flux. The thermal model is then made of N thermal nodes characterized by:

- a temperature T_i, in Kelvins;
- a capacitance $C_i = \rho_i c_i$, in Joules per Kelvin;
- an internal dissipated power QI_i, in Watts.

The model contains the thermal couplings between the nodes, stored in three $N \times N$ matrices:

- the conductive, linear couplings GL, in Watts per Kelvin;
- the radiative couplings, also called radiosities[5] GR, in square meters;
- the convective couplings[6].

[5]The radiosity GR_{i-j} is defined as the radiative exchange factor B_{i-j} from node i to j, multiplied by the product of the emittance ϵ_i by the area A_i:

$$GR_{i-j} = \epsilon_i A_i B_{i-j} \qquad (2.26)$$

This radiosity is expressed in m^2. This radiosity should not be confused with the thermal radiosity J_i, defined as the total power diffusely emitted by a surface:

$$J_i = \epsilon_i E_{b,i} + \rho_i H_i \qquad (2.27)$$

This latter radiosity is expressed in W/m^2.

[6]These couplings are mentioned to have a complete description of the thermal model. However, they will not be further considered in this document.

An equation similar to (2.25) can be written for each of the N nodes. We obtain a set of N non-linear equations with N unknowns (the nodal temperatures). The resolution of this set of equations can be performed with thermal solvers, such as Esatan and Thermisol.

If transient effects are considered, an additional term must be added in order to take into account the thermal inertia of each node:

$$QI_i + QS_i + QA_i + QE_i + \sum_{j=1}^{N} GL_{i-j}\left(T_j - T_i\right) + \sigma \sum_{j=1}^{N} GR_{i-j}\left(T_j^4 - T_i^4\right) = C_i \frac{\partial T_i}{\partial t} \quad (2.28)$$

This yields a set of N differential non-linear equations. Esatan can be used in order to compute the unknown temperatures.

Remark: The isothermal hypothesis must carefully be verified. If a thermal gradient appears in a region of the model, it must be meshed in order to locally satisfy the isothermal condition. On the other hand, a homogeneous region can be meshed with few elements.

The coupling matrices can contain a lot of zero-elements and a compression process is necessary. It allows to reduce the storage and the CPU time needed to solve the thermal equations.

The external heat loads are computed by ray tracing thanks to a dedicated software, such as Esarad and Thermica. These programs contain an orbitographic module, computing the orbital position and attitude of the spacecraft. A large panel of orbits and attitudes are available in these codes.

The resolution of equations (2.25) and (2.28) is performed by a thermal solver, such as Esatan and Thermisol. This solver should contain a lot of routines allowing steady-state thermal resolutions, the computation of transient, eventually cycling, thermal distributions. Temperature- and time-dependent properties can be defined in the solver.

2.4 Radiosity formulation

In the previous section, the equations used in the thermal lumped parameter method have been presented. This method is based on Gebhart's formulation, presented in [Geb61b, Geb59]. It is also presented in the monographs [SH01, Geb61a]. In our algorithm, we used either this formulation or the radiosity formulation, based on view factors. In this Section, the radiosity equation is briefly presented, first in the case of diffuse surfaces and then in the case of specular reflectors. A more detailed description is available in [Mod03, SH01].

The radiosity formulation can be modeled with electrical resistances, using the electrical analogy [Opp56, ZS66]. The Kennelly's delta-star transformation [Ken99] can be used to switch from one formulation to the other one.

2.4.1 Diffuse reflectors and radiosity equation

In this Section, all the surfaces of the model are assumed to be ideal diffuse emitters and reflectors [Mod03]. The diffuse reflectivity is assumed to be constant for all the wavelengths that govern the problem.

A fraction ϵ of an incoming radiation is then absorbed while the complementary fraction $1-\epsilon$ is diffusely reflected. The radiosity $J(r)$ is defined as the total diffuse energy leaving a position r. The radiosity is the sum of two components: the self-emitted power and the reflection of the incoming radiation:

$$J(r) = \epsilon(r)E_b(r) + \rho(r)H(r) \tag{2.29}$$

The net radiative heat flux is equal to the difference of the total emitted heat flux (*i.e.* the radiosity) and the total incoming heat flux (the irradiance $H(r)$). It is also equal to the difference of the self-emitted power and the absorbed component of irradiation (see Figure 2.8).

$$\begin{aligned} q(r) &= J(r) - H(r) && (2.30)\\ &= \epsilon(r)\left(E_b(r) - H(r)\right) && (2.31) \end{aligned}$$

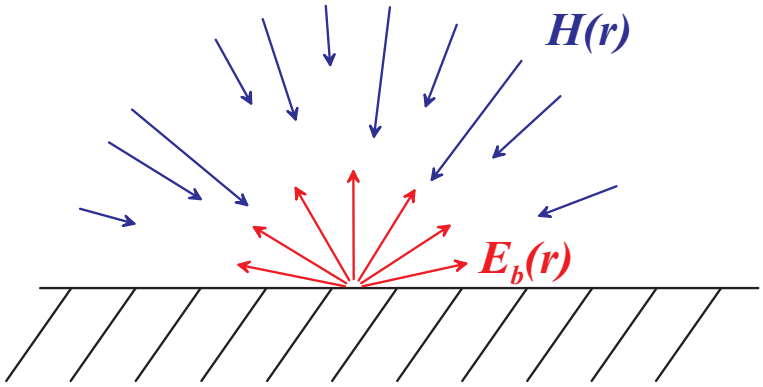


Figure 2.8: Heat fluxes on point r.

By combining equations (2.30) and (2.31), the following equation is obtained, which links the heat flux, the black body emission and the radiosity:

$$J(r) = E_b(r) - \left(\frac{1}{\epsilon(r)} - 1\right) q(r) \tag{2.32}$$

The incoming energy in r, given by $H(r)dA$, is equal to the external contribution, $H_0(r)dA$, plus the contribution of all the surfaces of the 3D-model. This last term is obtained by summing the radiosity of all the surfaces, weighted by their view factor with dA.

$$H(r)dA = H_0(r)dA + \int_{A'} J(r')dF_{dA'-dA}dA' \tag{2.33}$$

We use the reciprocity of the view factors and introduce the irradiance in equation (2.31).

$$q(r) = \epsilon(r) \left[E_b(r) - \int_{A'} J(r') dF_{dA-dA'} - H_0(r) \right] \tag{2.34}$$

Remark: if the model is made of black surfaces, the emissivity is equal to unity and the radiosity reduces to the black body emission. The following relation is then applicable:

$$q(r) = E_b(r) - \int_{A'} E_b(r') dF_{dA-dA'} - H_0(r) \tag{2.35}$$

If the expression (2.32) of the radiosity $J(r)$ is introduced in the equation of the radiative heat flux (2.34), the following relation is obtained, function of $q(r)$ and $E_b(r)$:

$$q(r) = \epsilon(r) \left\{ E_b(r) - \int_{A'} \left[E_b(r') - \left(\frac{1}{\epsilon(r')} - 1 \right) q(r') \right] dF_{dA-dA'} - H_0(r) \right\} \tag{2.36}$$

If the model is decomposed into N patches assumed to be isothermal and associated with a uniform emissivity, the following relation is obtained:

$$\frac{q_i(r_i)}{\epsilon_i} = E_{b,i} - \sum_{j=1}^{N} \left[E_{b,j} - \left(\frac{1}{\epsilon_j} - 1 \right) q_j(r_j) \right] F_{dA_i - A_j} - H_{0,i}(r_i) \tag{2.37}$$

The radiative heat flux is not necessarily constant because the irradiation is not uniform and $F_{dA_i - A_j}$ can vary on A_i. The previous relation has to be averaged on A_i.

$$\frac{q_i}{\epsilon_i} = E_{b,i} - \sum_{j=1}^{N} \left[E_{b,j} - \left(\frac{1}{\epsilon_j} - 1 \right) q_j \right] F_{i-j} - H_{0,i} \tag{2.38}$$

This relation can be recast into:

$$\sum_{j=1}^{N} \left[\frac{\delta_{ij}}{\epsilon_j} - \left(\frac{1}{\epsilon_j} - 1 \right) F_{i-j} \right] q_j = \sum_{j=1}^{N} \left(\delta_{ij} - F_{i-j} \right) E_{b,j} - H_{0,i} \tag{2.39}$$

This expression implies that the distribution of the fluxes q_j across surface j is governed by two sources:

- the self-emitted energy from the surfaces which compose the geometrical model;
- the distribution of the external irradiation $H_{0,i}$.

The heat fluxes q_j can be seen as the sum of two distinct components:

- the component q_j^E is necessary to balance the thermal system when only the self-emitted power is taken into account;

- the component q_j^I is used to balance the system when only external irradiation is considered.

$$\sum_{j=1}^{N} \left[\frac{\delta_{ij}}{\epsilon_j} - \left(\frac{1}{\epsilon_j} - 1 \right) F_{i-j} \right] q_j^E = \sum_{j=1}^{N} \left(\delta_{ij} - F_{i-j} \right) E_{b,j} \tag{2.40}$$

$$\sum_{j=1}^{N} \left[\frac{\delta_{ij}}{\epsilon_j} - \left(\frac{1}{\epsilon_j} - 1 \right) F_{i-j} \right] q_j^I = -H_{0,i} \tag{2.41}$$

Equation (2.40) is equivalent to the equation (A.7) of Gebhart while (2.41) is equivalent to (A.8). The equivalence between these two formulations is detailed in Appendix B.1.

2.4.2 Specular reflection

In the previous Section, only diffuse surfaces have been considered. However, common surfaces exhibit more complex BRDF[7] [Mod03]. Here, the BRDF is assumed to be decomposed into two components: an ideal diffuse one, noted ρ^d, and an ideal specular one, noted ρ^s, such as[8]:

$$\rho = \rho^d + \rho^s \tag{2.42}$$

The reflection operator, which is used to compute the distribution of light after one reflection, is given by:

$$\mathcal{R} = \mathcal{R}^d + \mathcal{R}^s \tag{2.43}$$

Each operator, when applied to a radiance distribution, yields a term of the final radiance distribution. L^d is defined as the diffuse component of the final radiance. If the emission is supposed to be diffuse, L^d is given by:

$$L^d = \mathcal{R}^d + L_e \tag{2.44}$$

The diffuse radiance L^d is obtained by removing the specular component from the total distribution.

$$L = L^d + \mathcal{R}^s L \tag{2.45}$$

In this equation, only a specular radiative balance is considered, with a source term L^d which takes into account the distribution of diffuse luminance. It yields:

$$L = \left(\mathcal{I} - \mathcal{R}^s \right)^{-1} L^d \tag{2.46}$$

$$L = \sum_{n=0}^{+\infty} \left(\mathcal{R}^s \right)^n L^d \tag{2.47}$$

[7] Bidirectional Reflectance Distribution Function.

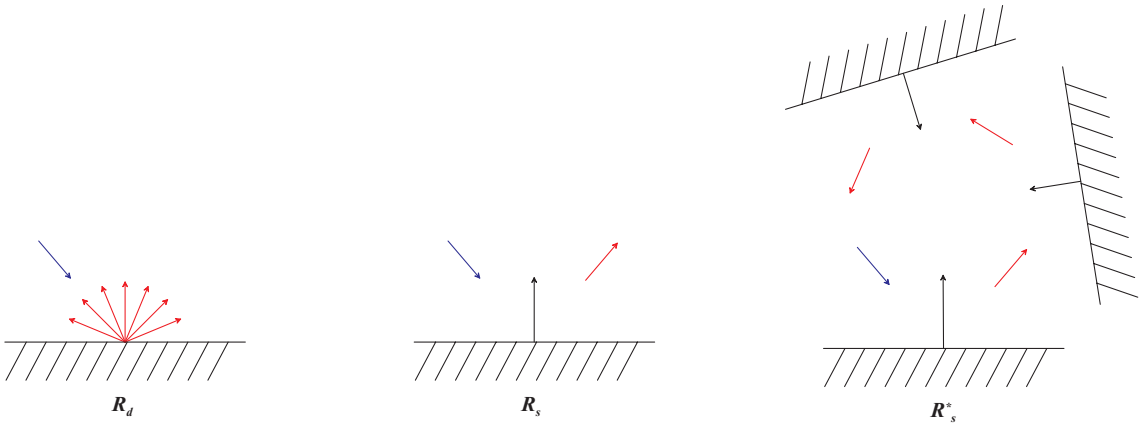
[8] For the sake of readability, the notation of the diffuse reflectivity in the previous section has been simplified. Only the diffuse component, noted here ρ^d, was taken into account in the previous Section.

The last equation has been obtained thanks to a decomposition following the *Neumann series* [Kaj86]. The global specular reflection operator is defined as:

$$\mathcal{R}^{s*} = \sum_{n=0}^{+\infty} (\mathcal{R}^s)^n \tag{2.48}$$

$$L = \mathcal{R}^{s*} L^d \tag{2.49}$$

This operator takes into account the effects of any number of specular reflections. The effects of the three operators are described in Figure 2.9. Figure 2.9–a represents the application of the $\mathcal{R}^d$ operator, where an incoming radiation is diffusely reflected. Figure 2.9–b is the effect of one specular reflection, modeled by the operator $\mathcal{R}^s$. Finally, Figure 2.9–c shows the effect of the global specular reflection operator on the incoming radiance.



(a) $\mathcal{R}^d$ (b) $\mathcal{R}^s$ (c) $\mathcal{R}^{s*}$

Figure 2.9: Reflection operators.

Specular reflection is a reflection where the light ray keeps its history in memory. Once the ray experiences a diffuse reflection, its memory is lost and the ray is reflected in a arbitrary direction, independently of the incoming direction.

In radiative heat transfer, only the diffusely absorbed luminance is interesting. In rendering, an additional operation is necessary, related to the specular component and the position of the observer. In radiative heat transfer, the computation is ended when a diffuse reflection is encountered, after any number of specular reflections. It yields:

$$L^d = L_e + \mathcal{R}^d L \tag{2.50}$$

$$L^d = L_e + \mathcal{R}^d \mathcal{R}^{s*} L^d \tag{2.51}$$

The operator $\mathcal{R}^d\mathcal{R}^{s*}$ represents any succession of specular reflections which ends with a diffuse one:

$$L^d = \left(\mathcal{I} - \mathcal{R}^d\mathcal{R}^{s*}\right)^{-1} L_e \tag{2.52}$$

This equation is equivalent to the previous, diffuse system but the matrix is based on the specular, extended view factors. The heat flux is now given by the general equation (2.31):

$$\begin{aligned} q(r) &= \epsilon(r)\left[E_b(r) - H(r)\right] && (2.53)\\ &= \epsilon(r)E_b(r) + \rho^d(r)H(r) + \rho^s(r)H(r) - H(r) && (2.54)\end{aligned}$$

The radiosity is defined as the diffusely emitted energy, sum of the self-emitted radiation and the diffuse reflection of irradiation:

$$J(r) = \epsilon(r)E_b(r) + \rho^d(r)H(r) \tag{2.55}$$

Remark: If a surface is purely specular ($\rho^d = 0$), the radiosity is equal to $\epsilon(r)E_b(r)$ and $H(r)$ is undefined. In fact, the corresponding irradiance H is specularly reflected to diffuse surfaces and then attributed to these surfaces.

The irradiance $H(r)$ is extracted from the equation (2.55):

$$H(r) = \frac{J(r) - \epsilon(r)E_b(r)}{\rho^d(r)} \tag{2.56}$$

If the expression of the irradiance (2.56) is introduced into (2.53), it yields:

$$\begin{aligned} q(r) &= \epsilon(r)\left[E_b(r) - \frac{J(r)}{\rho^d(r)} + \frac{\epsilon(r)}{\rho^d(r)}E_b(r)\right] && (2.57)\\ &= \frac{\epsilon(r)}{\rho^d(r)}\left[\left(1 - \rho^s(r)\right)E_b(r) - J(r)\right] && (2.58)\end{aligned}$$

From the previous equation, the expression of the radiosity related to the specular reflectors can be obtained:

$$J(r) = -\frac{\rho^d(r)}{\epsilon(r)}q(r) + \left(1 - \rho^s(r)\right)E_b(r) \tag{2.59}$$

The irradiance is obtained by summing the contributions of all the surfaces of the model. Using the specular view factors F^s_{i-j}, the following relation is obtained:

$$H(r) = \int_{A'} J(r')dF^s_{dA-dA'} + H^s_0(r) \tag{2.60}$$

The external irradiation $H^s_0(r)$ represents the distribution of external heat fluxes received either directly or after any number of specular reflections. By convention, a specular surface cannot receive any external irradiation; the reflected heat flux is assigned to the

next encountered diffuse surface.

The irradiance from (2.56) and the radiosity from (2.59) are used:

$$E_b(r) - \int_{A'} (1 - \rho^s(r))\, E_b(r') dF_{dA-dA'} = \frac{q(r)}{\epsilon(r)} - \int_{A'} \frac{\rho^d}{\epsilon} q(r') dF_{dA-dA'} + H_0^s(r) \qquad (2.61)$$

If all the surfaces are ideally diffuse, the specular reflectivity is equal to zero; the diffuse reflectivity is equal to $1-\epsilon$. The extended view factors are identical to the diffuse ones. The external irradiation does not experience any specular reflection: $H_0^s \equiv H_0$. Equation (2.36) is obtained for diffuse surfaces.

The model is decomposed into N isothermal patches with uniform thermo-optical properties. The system is averaged on the surface A_i:

$$E_{b,i} - \sum_{j=1}^{N} \left(1 - \rho_j^s\right) F_{i-j}^s E_{b,j} = \frac{q_i}{\epsilon_i} - \sum_{j=1}^{N} \rho_j^d F_{i-j}^s \frac{q_j}{\epsilon_j} + H_{0,i}^s \qquad (2.62)$$

If Kronecker's symbol is used, the following relation is obtained:

$$\sum_{j=1}^{N} \left(\delta_{ij} - (1 - \rho_j^s) F_{i-j}^s\right) E_{b,j} = \sum_{j=1}^{N} \left(\delta_{ij} - \rho_j^d F_{i-j}^s\right) \frac{q_j}{\epsilon_j} + H_{0,i}^s \qquad (2.63)$$

2.5 Conclusions

In this Chapter, the definition of the mathematical model which is used by the thermal engineer to design the thermal control subsystem of a spacecraft has been described. The different elements of the mathematical model have been detailed, as well as the commonly used methods.

The radiosity formulation which has been used in the developed algorithm to solve the thermal problem for purely radiative cases has also been presented. Cases including conduction have been solved with a finite element formulation, as it will be presented in Section 6.1.

Chapter 3

Computation of the exchange factors

Contents

This Chapter describes a new method designed to compute diffuse view factors by ray tracing. The extension to more complex thermo-optical properties is addressed in the second Section. This Chapter also considers the enforcement of reciprocity and closure of the exchange factors. It presents also recent developments concerning Gebhart's method, which can be used in order to derive the radiative exchange factors directly from the view factors by a matrix operation.

3.1 Computation of the view factors

3.1.1 Introduction

In the field of image synthesis or thermal radiation, the challenge is to calculate the radiative exchanges between the surfaces of a 3D model. To quantify the interactions between the surfaces, an adimensional number, called *view factor* and noted F_{i-j}, is defined. The view factor represents the fraction of the diffuse energy leaving a surface A_i which directly reaches a surface A_j. The calculation of the view factors is a very difficult problem. Different techniques have been designed in image synthesis. In this thesis, an innovative method, named *stratified hemisphere*, is presented. It is based on the *Nusselt's Analogy*. This new method is simpler and more natural than most of the other ones.

Section 3.1.2 presents a short state-of-the-art and details some techniques designed in the field of image synthesis, like the hemi cube and the single plane, because they present some similarities with the stratified hemisphere method. Section 3.1.3 is dedicated to the Nusselt's Analogy, which is the basis of stratified hemisphere. Section 3.1.4 describes the establishment and description of the hemisphere's mesh. In Section 3.1.5, two versions of the stratified hemisphere's method are presented; the first one is deterministic while the second one is stochastic. The stochastic version is described in detail in this Section. Section 3.1.6 is dedicated to a statistical accuracy measure adapted to the stratified hemisphere and the implementation of a statistical accuracy control. Section 3.1.7 presents the extension of the point wise stratified hemisphere in order to compute patch-to-patch view factors. An adaptation to finite element view factors is also given. Section 3.1.8 presents an alternative version of the stratified hemisphere, more efficient in terms of implementation: the sphere. Section 3.1.9 compares the classical Monte Carlo ray-tracing used in thermal software to stratified hemisphere.

3.1.2 State of the art

In the field of image synthesis, several methods have been designed in order to compute the view factors in a geometrical model. Figure 3.1 represents a classification of these different methods [CW93]. Among the analytical methods, some particular geometrical configurations are associated with analytical formulas [How82, Won76, ESA89]. However, these configurations are limited and the obtained expressions are quite complex. In case of polygonal surfaces, the Nusselt's Analogy can be used to compute point wise view factors [RV04, Bou00]. A complex formula can be established for the view factor between two polygons [SH93a, SH93b]. An additional difficulty resides in the computation of visibility, if obstacles are present in the model.

Among the numerical methods, a first distinction can be made between point wise view factor and view factor between surfaces of finite areas. In the first class, two sub-classes are defined. The first one is based on a sampling of the hemisphere subtended by the local normal; the second one is based on a sampling of the target area. The second class of methods contains methods such as Monte Carlo sampling and hierarchical subdivision [HSA91]. A more detailed description of different methods used to compute the view factors can be found in the internal report [Vue05].

In this Section, some methods of the first sub-class, called *hemisphere sampling* (in yellow in Figure 3.1) will be presented because they are linked to the stratified hemisphere method. Some of these methods, like the hemi cube and the single plane, have directly inspired the deterministic version of the stratified hemisphere. On the other hand, the techniques based on ray tracing and *Monte Carlo sampling*, like the *Malley's method*, can be seen as simplified versions of the stochastic stratified hemisphere. The method developed in this thesis is based on an improved stochastic sampling of the hemisphere (in orange in Figure 3.1).

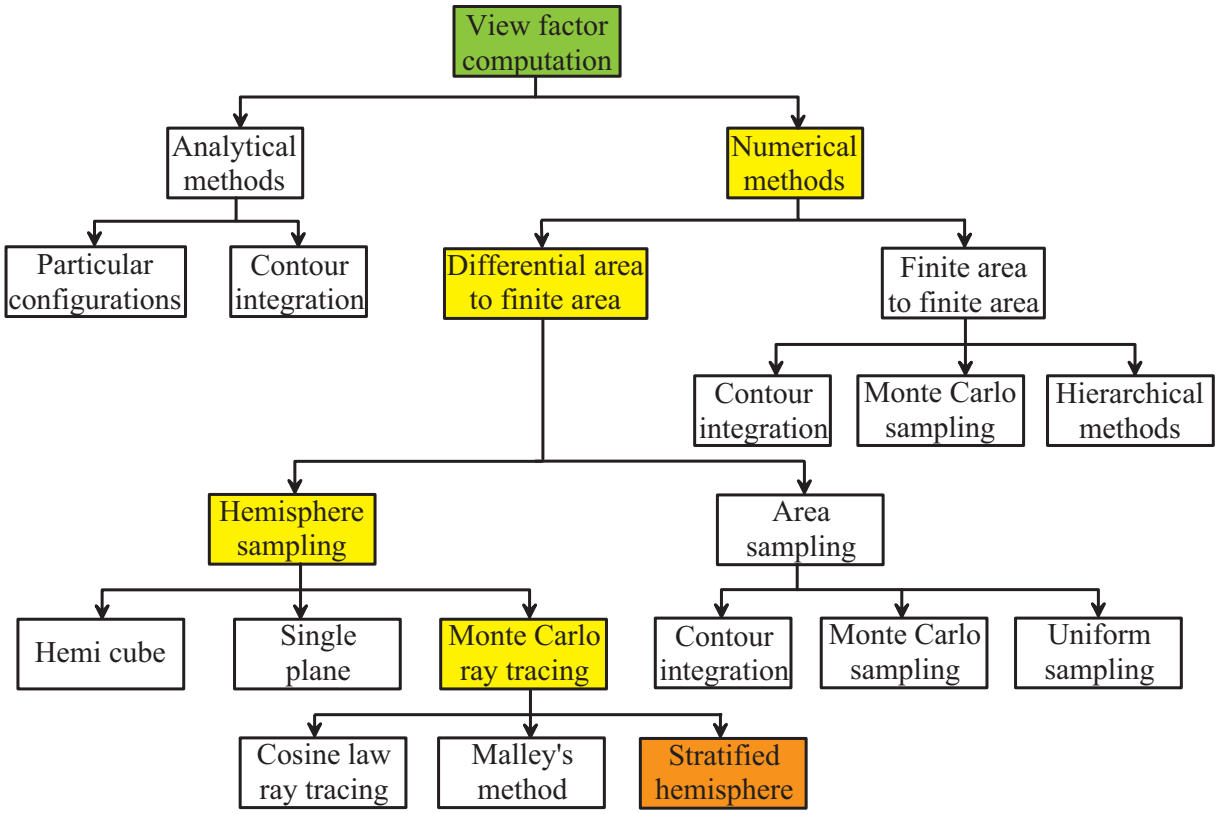


Figure 3.1: State of the art - Classification.

3.1.2.1 Hemi cube

The hemi cube has been designed and presented in [CG85]. It has first been used in image synthesis and after to solve thermal analysis [JCG93, CJMD95, JCG92].

This method is used to compute the diffuse point wise view factor. It consists in drawing a cube centered around the origin. The z-axis coincides with the normal vector $\overrightarrow{n_i}$. The construction is represented in Figure 3.2–a according to references [Jar03, CG85].

The cube is decomposed into cells (see Figure 3.2–a). Each cell is associated with an element of view factor, called *delta view factor* and noted ΔVF. These ΔVF are elementary contributions to the point wise view factor. The distribution of the ΔVF is represented in Figure 3.2–b. The cells near the z-axis are associated with the highest values and the cells located near the edges are characterized by lower values. This over sampling can be reduced if we apply a non-uniform mesh to the hemi cube, as it is performed with the single plane (see Figure 3.3–a from reference [SP89]). In Figure 3.2–a, a surface is projected onto the hemi cube; the projection covers two different faces of the cube. The corresponding contributions are summed to obtain the point wise view factor. This method presents drawbacks :

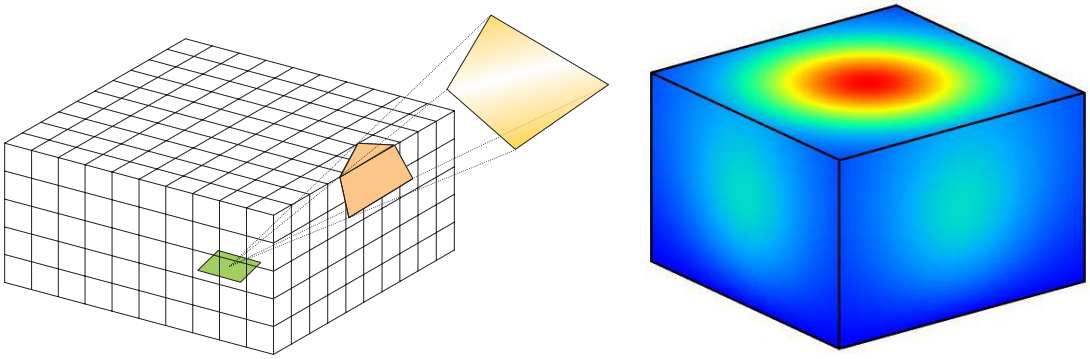


Figure 3.2: Hemi cube – a) geometrical configuration – b) distribution of ΔVF.

- The hemi cube is based on an irregular sampling of the space, implying over sampling to guarantee a minimal accuracy.
- Five projections (one for each face of the hemi cube) are needed in order to compute the view factors with the complete environment.
- The hemi cube, like the other methods based on a deterministic scheme to integrate the view factors, is subject to aliasing. It is not obvious to introduce a random process which respects the distribution of the view factor (*i.e.* the cosine law) in the formulation of the hemi cube.

Nevertheless, the main advantage of this technique is that occlusions can easily be taken into account by performing a test based on the distance of the surfaces projected onto the hemi cube. If many surfaces are projected onto the same cell of the hemi cube, only the nearest one is kept; the others will not be visible from the origin dA_i.

An alternative to the hemi cube, based on a tetrahedron, has been designed to reduce the number of projections [BKP91, BKP92].

The stratified hemisphere method presented in this thesis initially derives from the hemi cube method, developed to correct the drawbacks of the hemi cube.

3.1.2.2 Single plane

The method of the single plane has been presented in [SP89]. This method can be derived from the hemi cube. The lateral faces of the hemi cube are removed; only the top face is kept. It is extended following the $x-$ and $y-$axis in order to take into account the surfaces previously projected onto the lateral faces of the hemi cube. The plane is meshed following the x- and y-directions to obtain cells with approximately equal elementary view factors. There is no exact solution to this problem. The mesh presented in Figure 3.3–a has been

proposed in [SP89].

This method requires only one projection onto the plane. Moreover, one array is sufficient to store all the elementary view factors. If the mesh of Figure 3.3–a is chosen, one real number is sufficient to store the elementary view factor. The drawback of this method is that the view factors between the origin and the surfaces near the horizontal plane are neglected.

An alternative method has been proposed in [RV04, VVL+01, VLV99, VLV04]. The single plane is expressed in polar coordinates, more adapted to the axial symmetry of the view factor's formula. The single plane is broken down to cells characterized by an equal elementary view factor. In Figure 3.3–b, the evolution of the radius is plotted in function of the cumulated view factor. With a radius of 3, 90% of the whole view factor is represented while a radius of 10 corresponds to 99%.

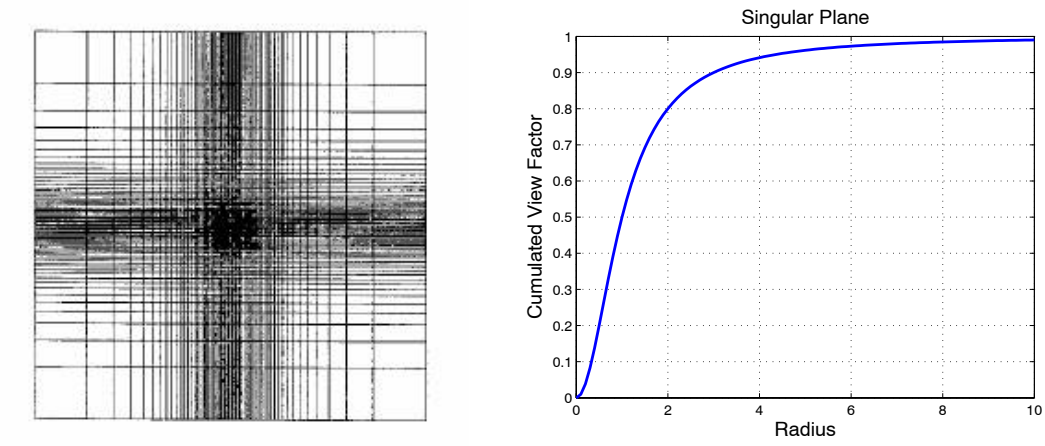


Figure 3.3: Single plane – a) mesh [SP89] – b) cumulated view factor.

The stratified hemisphere method that will be presented in the next sections allows us to combine the advantages of the previous mentioned methods, without their drawbacks. It is similar to the single plane because the deterministic mesh is composed of cells of equal elementary view factors.

3.1.2.3 Malley's method

The two previous methods are based on a projection of the environment onto a surface (a cube or a plane). The method presented in this section is based on a ray tracing algorithm. In this case, a large number of rays are generated. The intersections with each surface of the model are counted in order to compute the view factor. The distribution of the rays

direction must be chosen in such a way it obeys the cosine law.

Following the Nusselt's Analogy, if the unit disc is uniformly sampled, the obtained distribution satisfies the cosine law. This method has been designed in [Mal88] cited in [SP94]. The Malley's method consists in generating a set of points uniformly distributed across the unit disc (see Figure 3.4). The direction of each ray is obtained by projecting each point vertically from the unit disc to the hemisphere. Then, the ray is traced perpendicularly to the hemisphere's surface.

For each surface in the environment, the number of intercepted rays N_{i-j} is counted. The view factor is obtained by dividing the number N_{i-j} by the total number of casted rays.

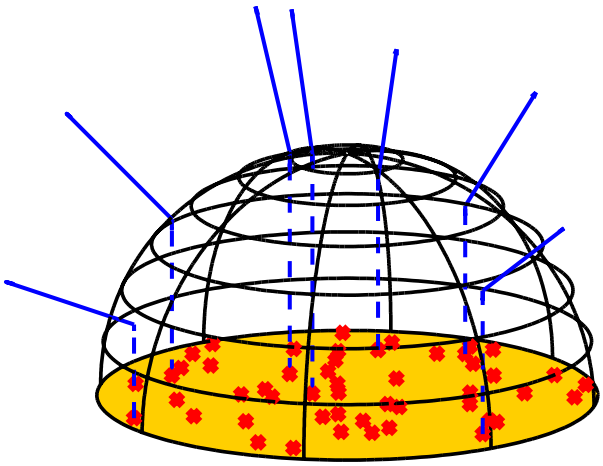


Figure 3.4: Malley's method - illustration of the method [CW93].

The Malley's method can be seen as a simplified version of the innovative method presented in this thesis. The two methods are designed to compute point wise view factors, they are based on ray tracing with a distribution of the rays which follows the cosine law.

The drawback of a stochastic process is the appearance of a random noise that affects the results. Some techniques can be used to decrease the variance of the stochastic process. One of them is *importance sampling*. By using a probability distribution function that corresponds to the cosine law, the Malley's method takes advantage of this technique.

3.1.2.4 Monte Carlo ray tracing

While Malley's method is used in order to compute the point wise view factor, the Monte Carlo ray tracing is designed to evaluate the view factors between patches of finite size. This is done by a two-step process:

- a sampling of the emitter's surface;

- a ray direction sampling.

If the sampling of the emitter's surface is uniform, this step does not introduce any weight (associated with the current ray). If the ray direction is defined by two uniformly angles θ (latitude) and ϕ (longitude), the weight of the ray must be multiplied by $\cos(\theta)$, in order to follow the cosine law (3.2). As the rays are characterized by unequal weights, the variance associated with MCRT is quite large. This variance can be reduced by choosing an adapted ray direction sampling.

3.1.2.5 Cosine law ray tracing

While the θ-sampling in the MCRT is uniform, the cosine law ray tracing is based on a non-uniform sampling of the latitude. This new distribution follows the cosine law and is characterized by a lower variance, since each ray is associated with an equal weight. This is the method implemented in common software, like Esarad [Doc04], Thermica[Doc03] and Samcef [Doc07].

3.1.3 Nusselt's Analogy

Following [SP94], the point wise view factor from a point dA_i on a patch P_i of area A_i to a patch P_j of area A_j is denoted F_{di-j} and is given by the relationship (3.1):

$$F_{di-j} = \int_{A_j} \frac{\cos(\theta_i)\cos(\theta_j)}{\pi r_{i-j}^2} V_{i-j} dA_j \tag{3.1}$$

where the parameters are defined as follows (see Figure 3.5):

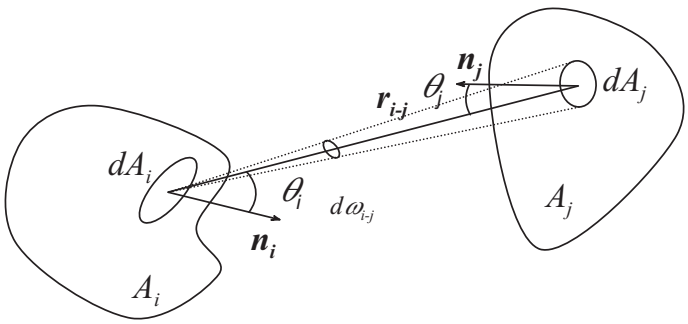


Figure 3.5: View factor - definition of the geometrical parameters.

- The point dA_i on the patch P_i is located by the vector $\overrightarrow{r_i}$;
- $\overrightarrow{r_{i-j}} = \overrightarrow{r_j} - \overrightarrow{r_i}$ is the vector joining the point dA_i and a point dA_j on the patch P_j;
- θ_i is the angle between the vector $\overrightarrow{r_{i-j}}$ and the local normal $\overrightarrow{n_i}$ of the surface A_i at the point dA_i; θ_j is the angle between the vector $\overrightarrow{r_{i-j}}$ and the local normal $\overrightarrow{n_j}$ of the surface A_j at the point dA_j;

- V_{i-j} is the visibility function. It is equal to 1 when the two points $\overrightarrow{r_i}$ and $\overrightarrow{r_j}$ can see each other, it is equal to 0 otherwise. This factor is responsible for discontinuities of the kernel when it is integrated on the surface A_j. This increases the difficulty of the computation.

Following the Nusselt's Analogy (cfr [Nus28] cited in [SP94]), the point wise view factor F_{di-j} can be considered as the result of two successive projections:

- a first projection onto the unit sphere centered on $\overrightarrow{r_i}$. This step corresponds to the factor $\frac{\cos(\theta_j)}{r_{i-j}^2}$ in the relation (3.1). The solid angle subtended by the surface dA_j is given by $d\omega_j = \frac{\cos(\theta_j)dA_j}{r_{i-j}^2}$.
- a second orthogonal projection down onto the plane of the surface A_i, which corresponds to the factor $\cos(\theta_i)$.

Therefore, the relation (3.1) can be expressed in terms of solid angles. This yields the following expression:

$$F_{di-j} = \int_{\Omega_j} \frac{\cos(\theta_i)}{\pi} d\omega_j \tag{3.2}$$

where Ω_j is the solid angle subtended by A_j. The factor π which appears at the denominator is a normalization constant. If Ω_j corresponds to the whole hemisphere defined by the local normal $\overrightarrow{n_i}$, the view factor F_{di-j} is equal to unity. This equation will be referred as the *cosine law*.

To conclude this Section, the Nusselt's Analogy and the double projection are illustrated. Figure 3.6–a shows a 3D geometrical configuration. In this case, the view factor is computed from the centre of the green square to the quadrangular blue patch.

The first step consists in projecting the blue patch onto the unit hemisphere sustained by the normal vector at the origin (at the centre of the green square). This is done at Figure 3.6–b. In this Figure, a spherical mesh has been superimposed on the spherical projection.

Figure 3.6–c represents the result of the orthogonal projection. The hemisphere is projected down to the horizontal plane. The final surface of projection surface is called the *unit disc*, limited by the blue circle. In this disc, the view factor is proportional to the area of the projection. The normalization constant is the area of the unit disc, equal to π.

3.1.4 Stratified hemisphere

The new method presented in this thesis and called *stratified hemisphere*, is based on a mesh of the unit disc. Two classes of meshes can be considered. The first one is based on a uniform decomposition of the unit disc and includes the stratified hemisphere. Each cell

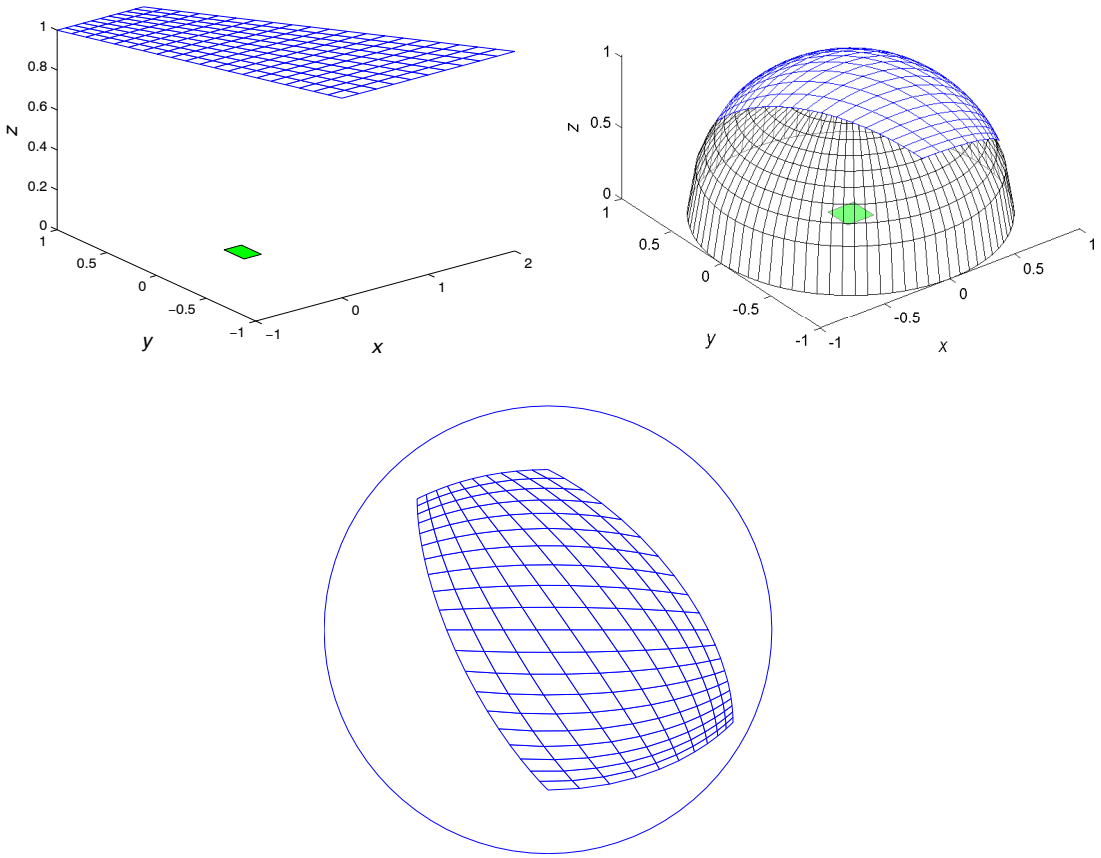


Figure 3.6: Nusselt's Analogy – a) 3D configuration – b) spherical projection – c) orthogonal projection.

is associated with an equal area, that is with an equal elementary view factor (following the Nusselt's Analogy). The polar decomposition presented in this document is exact: the contour of the disc is correctly represented. On the other hand, an orthogonal division of the unit disc yields a uniform mesh but the contour is approximated. This induces errors in the computation of the view factors since the summation rule is no longer verified (the surface of the approximated disc is not exactly equal to π). The same problem is encountered with a Delaunay discretization[1].

The second class of meshes is based on a weighted decomposition. Typically, it is the case of the hemi cube method. The single plane also belongs to this class. The main drawback of this class of meshes is that it can induce an over sampling of the domain. Some regions are over sampled while others are not sufficiently meshed. These meshes are not uniform.

[1]The unit disc is decomposed into triangles of the same area. The computation is performed on a first set of points; then the triangulation is iteratively modified in order to obtain triangles of the same size.

3.1.4.1 Decomposition of the unit disc

In this Section, with the help of the Nusselt's Analogy, the hemisphere is decomposed into solid angles. Each solid angle is characterized by the same elementary view factor.

The elementary point wise view factor is given by equation (3.2):

$$F_{di-j} = \int_{\Omega_j} \frac{\cos(\theta_i)}{\pi} d\omega_j$$

This equation, expressed in spherical coordinates on the superior hemisphere, gives us the following expression, where the variables θ and ϕ have been separated:

$$F_{di-j} = \int_{\theta_i \in \Omega_j} \int_{\phi_i \in \Omega_j} \frac{\cos(\theta_i)\sin(\theta_i)}{\pi} d\theta_i d\phi_i \tag{3.3}$$

$$= \int_{\theta_i \in \Omega_j} \cos(\theta_i)\sin(\theta_i) d\theta_i \int_{\phi_i \in \Omega_j} \frac{d\phi_i}{\pi} \tag{3.4}$$

So the decomposition will be performed following the parallels and the meridians, *i.e.* in terms of latitude and longitude. This technique allows us to prevent an useless discretization of the regions near the horizon. The total number of cells N will be obtained by multiplying the numbers of subdivisions following the longitude and latitude directions, n_{lon} and n_{lat}.

First, the hemisphere will be decomposed in terms of latitude, obtaining a succession of rings at the surface of the unit sphere. This preserves the axial symmetry of the definition of the view factor. Afterwards the rings are decomposed in sectors of equal angular aperture, in longitude.

In terms of latitude On the basis of the Nusselt's Analogy, a succession of n_{lat} concentric rings in the unit disc is considered. Figure 3.7 represents a succession of 10 such rings. If the rings have the same area, they will correspond to spherical rings characterized by an equal elementary view factor. The ring k is characterized by inner and outer radii equal to the following values, where n_{lat} is the resolution of the hemisphere in latitude:

$$\begin{cases} \rho_{\text{in}} = \sqrt{\dfrac{k-1}{n_{\text{lat}}}} \\ \rho_{\text{out}} = \sqrt{\dfrac{k}{n_{\text{lat}}}} \end{cases} \tag{3.5}$$

To obtain the corresponding solid angles, the radii have to be converted in terms of latitudes. Each radius can be projected onto the hemisphere, so the corresponding set of latitudes is obtained:

$$\theta_k = \arcsin\left(\sqrt{\frac{k}{n_{\text{lat}}}}\right) \tag{3.6}$$

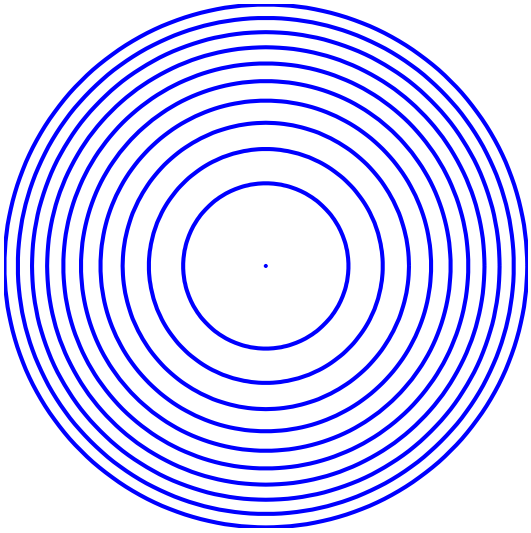


Figure 3.7: 10 concentric rings of equal elementary view factor in the unit disc.

For each cell, a view factor equal to $1/n_{\text{lat}}$ is attributed. In the framework of the deterministic hemisphere, this must be associated to the centre of the cell, in terms of latitude, following the θ-profile[2].

In terms of longitude After that, the discretization of the hemisphere in longitude can be chosen simply by dividing the rings in a determined number n_{lon} of sectors. All the sectors must have the same angular aperture $\Delta\phi = 2\pi/n_{\text{lon}}$.

3.1.4.2 Resulting mesh

So the hemisphere has been subdivided along the two spherical coordinates θ (latitude) and ϕ (longitude). The view factor associated to each cell is equal to $1/n_{\text{lat}}n_{\text{lon}}$. Figure 3.8 presents the hemisphere subdivided in longitude and latitude. The cells are quadrangular, except the cells which have a vertex vertically above the centre of the hemisphere and which are triangular.

3.1.4.3 Analysis of the mesh

We will now briefly detail some characteristics of the stratified hemisphere's mesh.

[2]The centres of the cells are defined by the sequence:

$$F_{di-j} = \left\{ k \in [1, n_{\text{lat}}] : \theta_k = \arcsin\left(\sqrt{\frac{2k-1}{2n_{\text{lat}}}}\right) \right\} \tag{3.7}$$

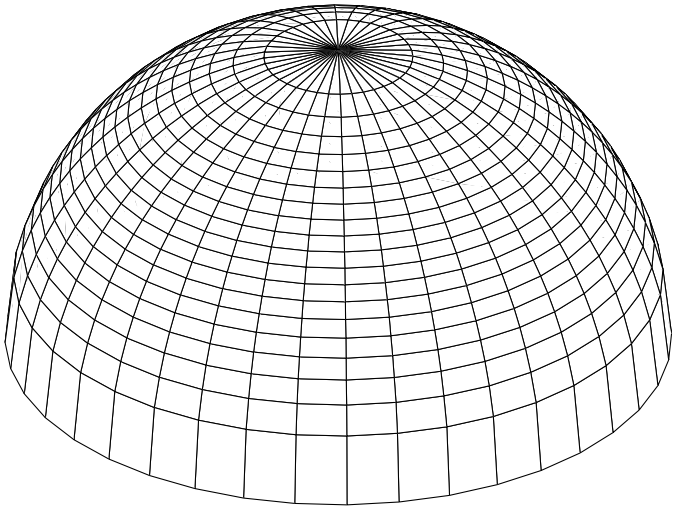


Figure 3.8: Wire frame view of the hemisphere.

Orthogonal projection Figure 3.9–a presents the result of the orthogonal projection of the mesh onto the horizontal plane. The area of each cell is a constant, equal to $1/n_{\text{lon}}n_{\text{lat}}$. This is in agreement with the Nusselt's Analogy: equal view factors are associated with equal areas in the unit disc.

Equivalent projection The cells of the mesh are not equal in terms of solid angle, except for the cells between two successive parallels. All cells are characterized by the same angular aperture in longitude. In Figure 3.9–b, the result of the equivalent projection of the stratified hemisphere's mesh is presented. This projection preserves area: spherical caps are represented by discs of the same area. For a sphere of unit radius, this projection preserves solid angles. A more detailed description of the equivalent projection can be found in [Bec03]. When compared to the orthogonal projection (Figure 3.9–a), the central cells of Figure 3.9–b have a smaller area, because of the reduced solid angle. The area of the cells increases when going from the centre to the external circle.

Remark : in order to obtain a strict equality between the spherical area and the projected area, a scale factor has to be introduced during the projection. Indeed, the area of the whole hemisphere is equal to $2\pi\rho_s^2$ (where ρ_s is the radius of the sphere); the radius of the equivalent disc is given by $\rho_d = \rho_s\sqrt{2}$. There is a scale factor equal to $\sqrt{2}$ between the orthogonal and the equivalent projections.

3.1.5 Computation of the view factor

The first version of the stratified hemisphere is a deterministic one. It is subject to aliasing, leading to unacceptable error. A stochastic version is presented in order to obtain a more robust method.

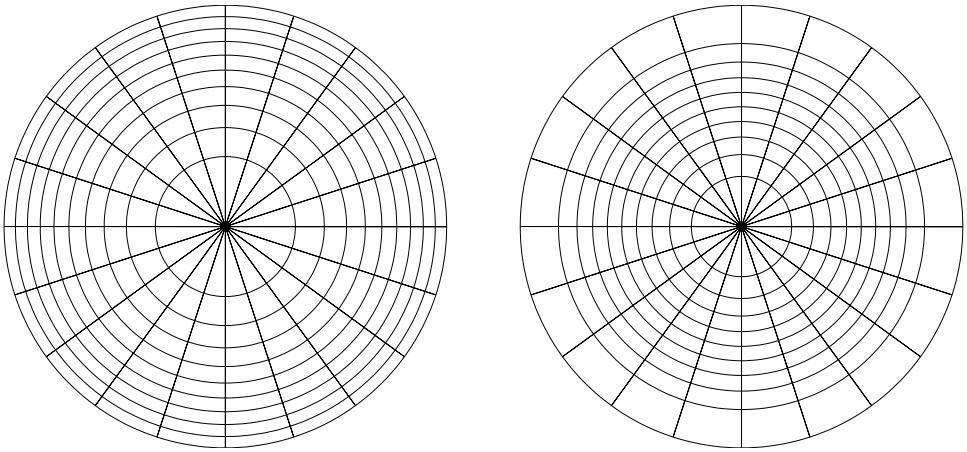


Figure 3.9: Analysis of the mesh – a) orthogonal projection – b) equivalent projection.

3.1.5.1 Deterministic stratified hemisphere

One ray is traced through the centre of each cell (in terms of view factors). For each ray, the possible intersection of the ray with each surface of the 3D model is computed. If no intersection is detected, the geometrical model presents an aperture and the ray is lost. This is the case of a satellite in orbit around the Earth and which radiates its energy to the deep space. The elementary view factor is associated with an additional node which represents the deep space. If one intersection is identified, the elementary view factor is attributed to the corresponding surface. In case of multiple intersections, the distance of each intersection is stored. At the end, only the nearest intersected surface is kept. This simple technique allows us to naturally take into account the visibility factor V_{i-j} appearing in the definition of the view factor (3.1).

This deterministic method has been used to compute the diffuse view factors between surfaces, for several simple geometrical configurations (for which an analytical formula can be derived). For some "optimal configurations", an accuracy which evolves proportionally to $1/N = 1/n_{\text{lat}}n_{\text{lon}}$ has been observed. For optimal configurations, each ray is used optimally. The convergence is then proportional to $1/N$, where N is the total number of casted rays.

Nevertheless, in some configurations, an error linked to the deterministic aspect of the method can be observed. This is called *aliasing* and it results in unacceptable levels of the computed error. In Figure 3.10–b, the relative error in the case of two perpendicular rectangles sharing a common edge is presented (see Figure 3.10–a). The error is function of the resolution in longitude (x-axis) and latitude (y-axis). The color scale is logarithmic; a value of -4 corresponds to a relative error equal to 10^{-4}. This result has been obtained with 5×5 Gauss points. The Gauss's quadrature will be presented with more details in

the next Sections.

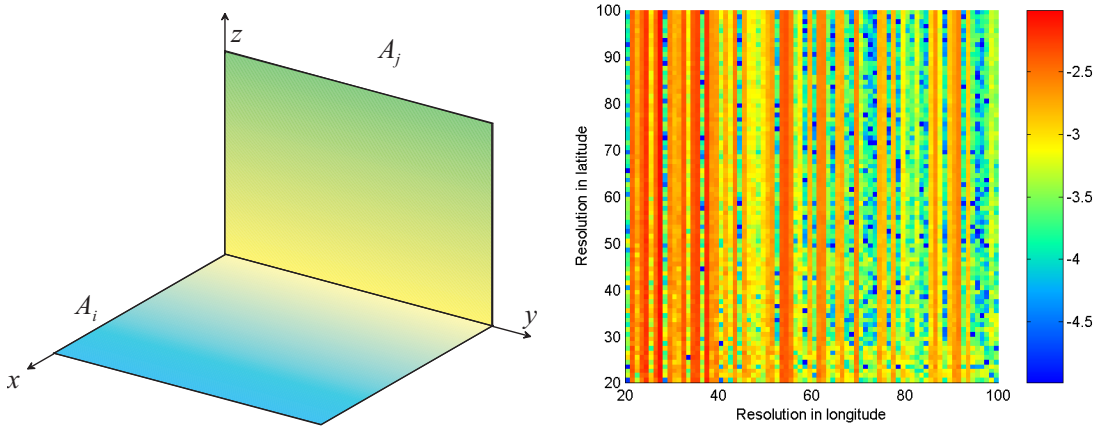


Figure 3.10: Perpendicular rectangles – a) 3D configuration – b) Relative error.

Vertical periodical bands can be observed. This implies that the error presents oscillations in longitude, independent of the latitude resolution. These oscillations are due to the particular geometrical configuration. For each of the 25 stratified hemispheres placed across the horizontal surface, the vertical edges of the second rectangle are projected along a meridian. The stratified hemisphere, which is based on a deterministic scheme, will detect or neglect a whole column of rays casted at a fixed longitude, along a same meridian. From a resolution to the other, n_{lat} rays will be either counted or lost. The number of rays responsible for the error is proportional to n_{lat}; the error is about $n_{\text{lat}}/N = 1/n_{\text{lon}}$.

If the resolution in latitude is assumed to be equal to the resolution in longitude, the error is about $1/\sqrt{N}$. This error can be compared to the error which characterizes stochastic processes. This illustrates the idea developed in the reference [LCN95]: *for a totally a priori unknown result, a uniform deterministic sampling of the basic probability function is always superior, or at least as good as, uniformly distributed random sampling for any abinitio fixed number of trials.* In this case, a deterministic scheme yields results whose quality is equivalent to a purely stochastic process. This reference will be reminded during the study of the stochastic version of the hemisphere.

Remark : A solution to the oscillations could be to rotate the stratified hemisphere along the local normal in such a way that the initial longitude is shifted with respect to a vertical edge of the second surface. It must be noted that this solution is only efficient for this simple configuration. If the 3D model is more complex, this method is no longer valid. The result is displayed in Figure 3.11. The vertical bands have disappeared but a second oscillation in latitude appears when the resolution increases.

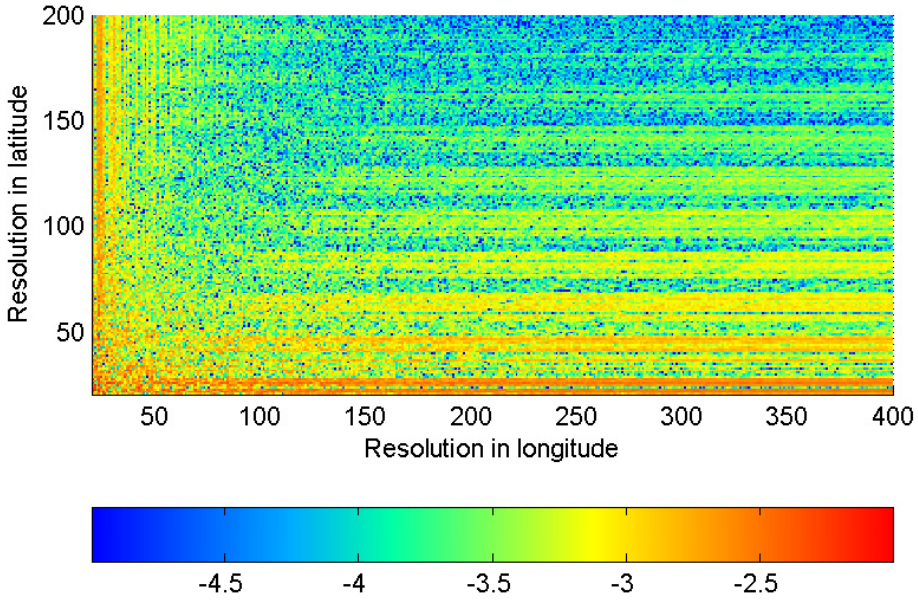


Figure 3.11: Perpendicular rectangles with rotated hemispheres - relative error.

In this simple example, it appears that a deterministic scheme is not suitable. This error is linked to a phenomenon which affects every deterministic process: the *aliasing*. Aliasing is illustrated in Figure 3.12, in the case of a rectangular grid and five projections of a triangle. A cell is assumed to be covered by a projection if the centre is covered. In function of the position of a triangle, the projection covers zero, one, two or three cells (colored in green). So the same surface can be associated to different numbers of cells, depending on its position on the grid. Even worse, it can be missed if its projection falls between the centres of the cells.

One way to prevent aliasing is *jittering*, that is adding some random noise on the position of the centres [Coo86]. This leads us to the stochastic version of the hemisphere.

3.1.5.2 Stochastic stratified hemisphere

The idea developed in this section is to use the mesh of the stratified hemisphere to stochastically cast rays through each cell of the mesh.

This method corresponds to *stratified sampling* [DBB03, SWZ96]. In general, the methods for which a large number of samples are generated can suffer from an accumulation of samples in a region of the domain of definition, the other regions remaining depleted. To be sure that the samples are uniformly distributed, a large number of them have to be generated. Stratified sampling consists in a decomposition of the domain of definition in disjoint sub-domains called *strata*. A given number of samples are then generated in each stratum to guarantee that each region is equally sampled. Stratified sampling is a

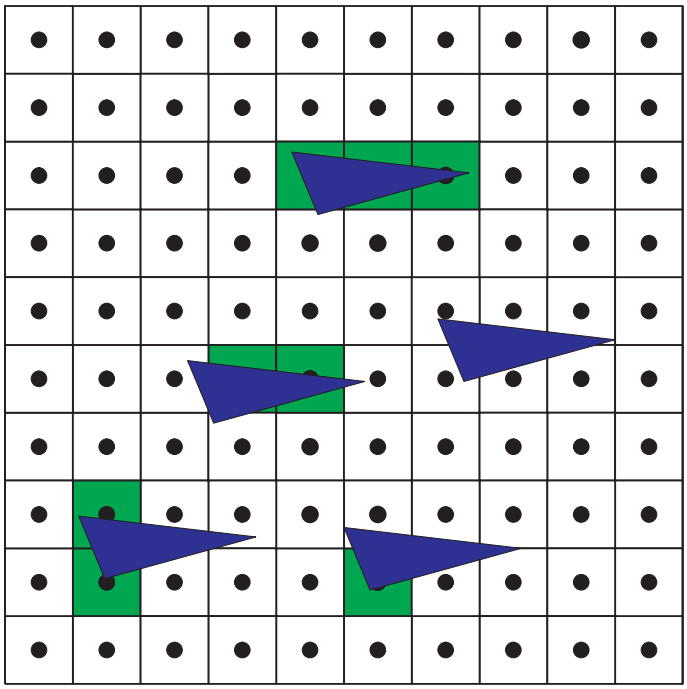


Figure 3.12: Illustration of aliasing on a rectangular grid.

method to reduce variance. Stratified sampling is a sampling of the domain guided by a stratification of the domain.

Distribution in latitude The cells of the unit disc are comprised between radii equal to $\sqrt{\frac{k-1}{n_{\text{lat}}}}$ and $\sqrt{\frac{k}{n_{\text{lat}}}}$. A random radius can be generated into each ring. Then, the random radius is projected onto the hemisphere's mesh, yielding the corresponding set of latitudes.

$$\theta_k \quad = \quad \arcsin\left(\sqrt{\frac{k-\xi_1}{n_{\text{lat}}}}\right) \tag{3.8}$$

where ξ_1 is a first random number uniformly distributed between 0 and 1.

Distribution in longitude The distribution in longitude is easy to find. The angular aperture in longitude is constant for each cell of the mesh and equal to $\Delta\phi = 2\pi/n_{\text{lon}}$. If the inferior limit of a cell ϕ_{inf} is given, random longitudes can be generated in this cell by using the formula, where ξ_2 is a second random number uniformly distributed between 0 and 1:

$$\phi = \phi_{\text{inf}} + \xi_2 \Delta\phi \tag{3.9}$$

Illustration In Figure 3.13, the evolution of the relative error can be observed in the case of two perpendicular rectangles sharing a common edge (see the Figure on the right).

The resolutions in longitude and latitude vary between 20 and 400. This result must be compared with Figure 3.10–b, obtained with the deterministic version of the stratified hemisphere. The periodical vertical bands have disappeared. The stochastic version of the stratified hemisphere is more robust than the deterministic one.

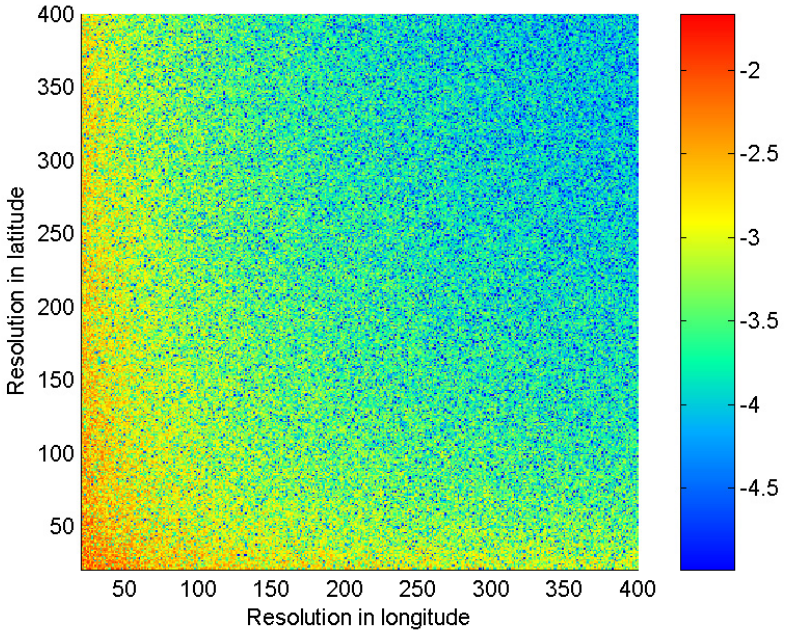


Figure 3.13: Perpendicular rectangles - relative error.

3.1.6 Statistical accuracy control

The objective of this Section is to develop a method capable to compute the view factors with a Monte Carlo ray tracing algorithm within a given accuracy. The user defines a maximum relative error ϵ and a confidence interval α; the statistical accuracy control determines the necessary number of rays N in function of the geometrical configuration. Here, a statistical accuracy measure applied to the stochastic stratified hemisphere is presented. This is a generalization of the formulation developed for classical Monte Carlo ray tracing [Pla93]. This statistical accuracy measure is then applied in order to establish a statistical accuracy control.

3.1.6.1 Study of the stratified hemisphere's error

In the stratified hemisphere method, if we consider an equal resolution in latitude and in longitude, we can note M the resolution in each direction, $n_{\text{lon}} = n_{\text{lat}} = M$. The total number of traced rays is equal to $N = M^2$. Let us assume that the number of rays intercepted by a surface A_j, without any obstacle, is equal to n^2, giving the computed view

factor $F^*_{di-j} = n^2/N$. n is linked to the total number of rays by the relation $n = \sqrt{F^*_{di-j} N}$. Approximately cn rays (or cn cells of the hemisphere) correspond to the contour $\mathcal{C}_j$ of the surface. c is a constant related to the shape of the contour. The cn cells are subject to error and must be studied carefully if we want to know the global error of the method.

Each cell has a different behaviour, each shot ray can be supposed independent and must be considered individually. For each cell k of the hemisphere, the *shadow ratio* p_k is defined as the fraction of the cell k which is covered by the spherical projection of the surface A_j. This shadow ratio is proportional to the elementary view factor ΔA_j associated with the considered cell. The shadow ratio is noted p_k and is comprised between 0 and 1. It is equal to 0 when the surface A_j does not cover the cell; it is equal to 1 when the coverage is total. The shadow ratio is not known *a priori* but we have the following relation:

$$\frac{1}{N} \sum_{k=1}^{N} p_k = F_{di-j} \tag{3.10}$$

where F_{di-j} is the expected view factor.

Remark : the shadow ratio defined here is not the elementary view factor $\Delta F = 1/N$ which is constant for each cell in the hemisphere method, equal to the inverse of the number of casted rays, and which is independent of the surface A_j. Here, the shadow ratio is proportional to the true elementary view factor through the cell.

In each cell k, the stochastic problem follows a binomial distribution. The ray randomly traced through the cell is tested for intersection with the surface A_j. A random variable T_k is introduced. It can only take two values: 1 if the intersection test is true and 0 if the ray misses the surface. In the case of a binomial distribution, the probability of intersecting the surface must be known. This probability is different for each cell k and is equal to the shadow ratio p_k of cell k. The expected value and the variance are then equal to:

$$\left\{ \begin{array}{rcl} E(T_k) & = & p_k \\ \sigma^2(T_k) & = & p_k(1-p_k) \end{array} \right. \tag{3.11}$$

The view factor is computed by the relation $F^*_{di-j} = \frac{1}{N} \sum_{k=1}^{N} t_k$, where t_k is the score of the binomial variable T_k. The expected value and variance of the distribution are:

$$\left\{ \begin{array}{rcl} E(F^*_{di-j}) & = & F_{di-j} \\ \sigma^2(F^*_{di-j}) & = & \frac{1}{N^2} \sum_{k=1}^{N} p_k(1-p_k) \end{array} \right. \tag{3.12}$$

If the hypotheses of the central limit theorem [Gra78] are verified, it is established that if the number of samples N is sufficiently large, F^*_{di-j} is normally distributed. The three hypotheses of the central limit theorem are verified:

1. the random variables t_k are independent;
2. their expectation and variance exist (see equations (3.11));
3. the ratio of the variance of a given variable to the sum of the variances $\frac{\sigma^2(T_k)}{\sum_{k=1}^{N}\sigma^2(T_k)} = \frac{p_k(1-p_k)}{\sum_{k=1}^{N}p_k(1-p_k)} \approx \frac{1}{N}$ tends to zero when the number of samples increases indefinitely.

The relative error is defined as $e = \frac{F^*_{di-j} - F_{di-j}}{F_{di-j}}$. The expectation and variance of this process are:

$$\begin{cases} E(E) &= 0 \\ \sigma^2(E) &= \frac{\sigma^2(F^*_{di-j})}{F^2_{di-j}} = \frac{1}{N^2F^2_{di-j}}\sum_{k=1}^{N}p_k(1-p_k) \end{cases} \tag{3.13}$$

Since F^*_{di-j} is normally distributed, so will be E. The probability α of having a relative error smaller than ϵ, also called the confidence interval, is given by the following relation:

$$\alpha = \mathcal{P}(|e| < \epsilon) = \operatorname{erf}\left(\frac{\epsilon}{\sigma(E)\sqrt{2}}\right) \tag{3.14}$$

By introducing the expression of the variance (equation (3.13)) in the last equation, the following relations are obtained:

$$\left(\frac{\epsilon}{\operatorname{erf}^{-1}(\alpha)}\right)^2 = \frac{2}{F^2_{di-j}N^2}\sum_{k=1}^{N}p_k(1-p_k) \tag{3.15}$$

$$N^2 = \frac{2}{F^2_{di-j}}\left(\frac{\operatorname{erf}^{-1}(\alpha)}{\epsilon}\right)^2\sum_{k=1}^{N}p_k(1-p_k) \tag{3.16}$$

Among the N terms of the sum, only the terms relative to the contour $\mathcal{C}_j$ contribute to the global error. The other terms correspond to shadow ratios equal to 0 or 1, which cancel the corresponding contributions. We obtain an upper bound to the number of rays by setting the shadow ratios of the cells of $\mathcal{C}_j$ to 50%. The sum in equation (3.16) is over-estimated by $c\sqrt{F^*_{di-j}N}/4$. To obtain the final expression, we approximate the exact view factor F_{di-j} by the computed value F^*_{di-j}.

$$N^2 = \frac{c\sqrt{F^*_{di-j}N}}{2F^2_{di-j}}\left(\frac{\operatorname{erf}^{-1}(\alpha)}{\epsilon}\right)^2 \tag{3.17}$$

$$N = \frac{\sqrt[3]{c^2}}{\sqrt[3]{4}F_{di-j}}\left(\frac{\operatorname{erf}^{-1}(\alpha)}{\epsilon}\right)^{\frac{4}{3}} \tag{3.18}$$

Remark : Formula (3.16) can lead to the formula obtained in [Pla93], in the case of a cosine law ray tracing. This is equivalent to N stratified hemispheres with only one cell in longitude and latitude ($n_{\text{lon}} = n_{\text{lat}} = 1$). The N hemispheres are supposed to be independent. In this case, all the probabilities p_k are equal to the expected view factor F_{di-j}, all the rays are subject to error. The sum becomes:

$$\sum_{k=1}^{N} p_k(1-p_k) = NF_{di-j}(1-F_{di-j}) \tag{3.19}$$

This result is introduced in the equation (3.16):

$$N = 2\left(\frac{\text{erf}^{-1}(\alpha)}{\epsilon}\right)^2 \frac{1-F_{di-j}}{F_{di-j}} \tag{3.20}$$

This is the formula presented in [Pla93] for diffuse view factors. In the next Figure, the curve corresponding to the intersection of the results of [Pla93] and ours has been plotted (discontinuous red curves).

Figure 3.14 represents the number of rays needed to obtain a relative error comprised between $[-\epsilon, \epsilon]$ with a confidence interval α of 99% in function of the view factor. The maximum relative error is defined between 100 and 1%. To obtain these curves, the erroneous cells have been assumed to be characterized by a shadow ratio of 50%. This hypothesis guarantees that the computed number N is an upper bound, because a ratio of 50% maximizes the corresponding error term $p_k(1-p_k)$. In Figure 3.14, the curves obtained with formula (3.18) are compared with the S-shape curves obtained in the case of a cosine law ray tracing [Pla93], *i.e.* without any directional optimization. In Figure 3.14, the red dashed line represents the intersection of the two families of curves, in function of the relative error. For most values of view factors, the stratified hemisphere is more efficient than cosine law ray tracing.

The hypothesis of a shadow ratio equal to 50% for all the erroneous cells is not realistic, especially when the view factor tends to 1. When the view factor tends to 1, the shadow ratio of most of the contour cells also tends to 1. This produces a decrease of the real error which is not reproduced in the curves of the stratified hemisphere.

Let us define the parameter K by the following expression:

$$K = 2^{\frac{5}{3}}\left(\frac{\text{erf}^{-1}(\alpha)}{\epsilon}\right)^{\frac{2}{3}} \tag{3.21}$$

The number of rays of the stratified hemisphere (relation (3.18)) becomes:

$$N = \frac{c^{\frac{2}{3}}K^2}{16F_{di-j}} \tag{3.22}$$

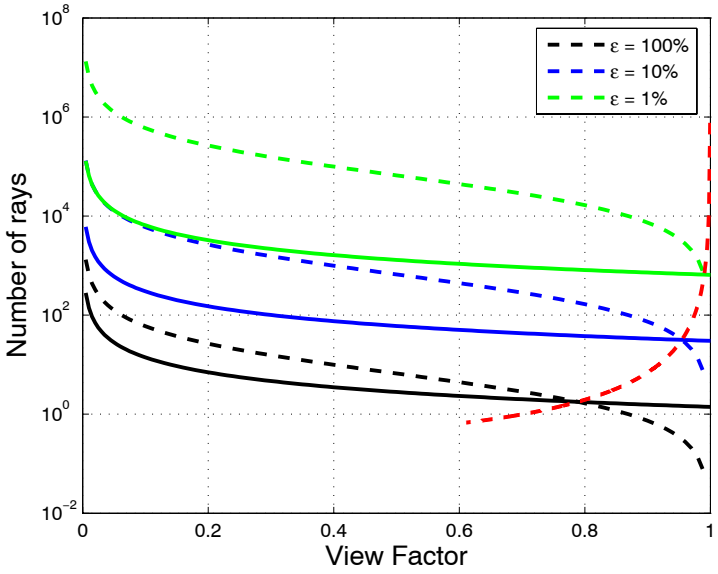


Figure 3.14: Application of formula (3.18). Comparison with the curves of [Pla93], for a cosine law ray tracing.

The equation (3.20) of [Pla93], in the case of a cosine law ray tracing, can be rewritten as:

$$N_{\text{CLRT}} = \frac{K^3\,(1 - F_{di-j})}{16 F_{di-j}} \tag{3.23}$$

In a dedicated section, these two last equations will be used to simplify the comparison between the stratified hemisphere method and the cosine law ray tracing.

3.1.6.2 Convergence of the hemisphere method

Previously, the convergence of a deterministic scheme (between $\frac{1}{N}$ and $\frac{1}{\sqrt{N}}$) has been compared with the convergence of a random one ($\frac{1}{\sqrt{N}}$). It appears that, for a given number of rays, the deterministic scheme is superior, or at least as good as random sampling. The convergence of the stratified hemisphere can be found from equation (3.18). For a given confidence interval α, the relative error ϵ is function of the number of traced rays N:

$$\epsilon = \frac{\sqrt{c}}{\sqrt{2} F_{di-j}^{\frac{3}{4}}} \frac{\text{erf}^{-1}(\alpha)}{N^{\frac{3}{4}}} \tag{3.24}$$

For a given configuration, the view factor and the constant c are fixed. The relative error is proportional to $N^{\frac{4}{3}}$. This convergence is superior to the classical convergence of random processes, such as the cosine law ray tracing. This is the reason for the superiority of the stratified hemisphere. A deterministic process can achieve a better convergence but it depends on the geometrical configuration. In some case, the convergence can be

identical to the convergence of random processes. The proposed method ensures a constant convergence, with an error proportional to $N^{\frac{4}{3}}$.

3.1.6.3 Estimation of the error

In the formula (3.16), the sum concerns in fact the m cells subject to error:

$$N^2 = \frac{2}{F_{di-j}^2} \left(\frac{\text{erf}^{-1}(\alpha)}{\epsilon} \right)^2 \sum_{k=1}^{m} p_k(1-p_k) \tag{3.25}$$

Indeed, the cells inside the projection of A_j onto the hemisphere are characterized by a shadow ratio equal to 0, which cancels the corresponding term $p_k(1-p_k)$. In the same way, the cells located outside the projection's contour are associated with an unity shadow ratio, which cancels the term of error. It appears that only the cells located on the contour present a partial shadow ratio and contribute to the error.

A Matlab routine has been used to compute the projection of a target surface A_j onto the hemisphere and derive the distribution of the shadow ratios. Varying resolutions M have been considered to study the evolution of different quantities (number of cells inside the contour, number of cells along the contour, error term...) in function of M. Two interesting results are the linear behaviours of the number of cells inside the contour and the error term in function of the resolution, yielding the definition of a new constant d such as:

$$m \approx cM \tag{3.26}$$

$$\sum_{k=1}^{m} p_k(1-p_k) \approx dM \tag{3.27}$$

The description of the Matlab test and results, as well as the establishment of equation (3.27), are given in Appendix C.

3.1.6.4 Test

To conclude this Section, the formula (3.16) has been used in the reference configuration used in Section 3.1.6.3, detailed in Appendix C. For a fixed number of rays, for a given view factor, the evolution of the confidence interval α in function of the maximum observed relative error can be plotted (see Figure 3.15). The green curve corresponds to the solution of equation (3.16). The incremental blue curve represents the experimental result. The studied configuration is characterized by a view factor equal to 0.55413, the resolution of the hemisphere is 100, the total number of shot rays is 10 000.

The steps observed in Figure 3.15 are due to the limited resolution of the hemisphere, given by $\Delta\epsilon = 10^{-4}/F_{di-j} \approx 1.8\,10^{-4}$ in this particular case.

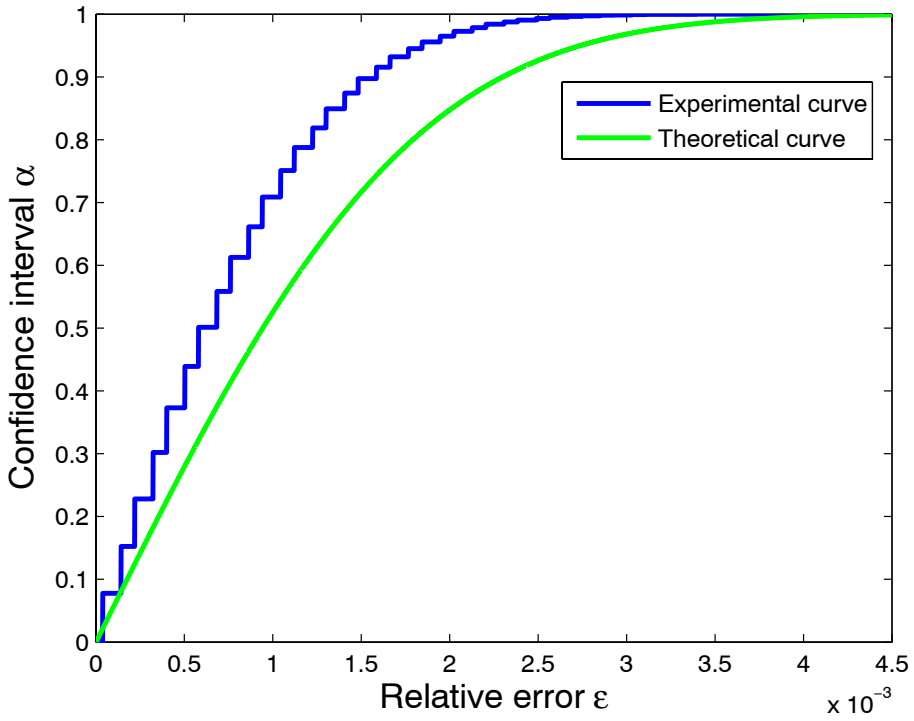


Figure 3.15: Evolution of the confidence interval in function of the relative error.

Theoretical green curve systematically underestimates the confidence interval. The theoretical formula underestimates the reliability of the results computed with the stratified hemisphere method. This is due to the restrictive hypothesis used in the demonstration of this equation. Namely the shadow ratio of all contour's cells has been considered equal to 50%. This increases the contribution of each cell to the global error.

We have now an expression which overestimates the necessary effort to obtain a fixed relative error, with a fixed confidence interval and for a given configuration, that is for a given view factor.

3.1.6.5 Statistical accuracy control

In this Section, the statistical measure is used in order to obtain a statistical accuracy control. This control requires two passes. During the first pass, a first ray tracing is used to determine the geometrical configuration (estimated view factor, number of cells along the contour). Then the optimal number of rays is computed by the statistical accuracy measure and a second ray tracing is performed.

First pass The aim of the first pass is to yield a first estimation of the view factor. It is also used to estimate the constant c which links the number of rays across the contour with the resolution M. This is done with a low-resolution hemisphere (typically from 25×25 to

50×50). The rays are traced through the corners of each cell, they are not randomly shot through each cell. Because this ray tracing is deterministic, aliasing can be observed.

The first pass allows us to estimate the shadow ratio of the different cells of the hemisphere. When the rays emitted through the four corners of a cell impact a same patch P_j, the coverage of this cell by P_j is assumed to be complete. This is not always true but it is acceptable in this first pass, which only yields approximated results. When no ray impacts P_j, the shadow ratio is set to zero. For the intermediate cases, the shadow ratio is discontinuous and varies by steps of 25%.

With the shadow ratios, the global view factor is estimated with the formula (3.10):

$$F_{di-j} = \frac{1}{N}\sum_{k=1}^{N} p_k$$

Moreover, the geometrical constant c and the error constant d can be derived by counting the number of cells across the contour, characterized by a partial shadow ratio. Relation (3.26) is used to derive c from the resolution chosen for the first pass; equation (3.27) yields d in function of M.

Computation of the number of rays Giving a maximum relative error ϵ and a confidence interval α, the statistical accuracy control computes the number of rays that must be shot to achieve the desired accuracy. The resolution of the stratified hemisphere is given by the relation (3.16). Relation (3.26) yields an estimation of the number m of non-zero terms. For these terms, the corresponding shadow ratios are arbitrarily fixed at 50%.

$$\begin{aligned} N^2 &= \frac{2}{F_{di-j}^2}\left(\frac{\mathrm{erf}^{-1}(\alpha)}{\epsilon}\right)^2 \sum_{k=1}^{m} p_k(1-p_k) \\ M &= \sqrt[3]{\frac{c}{2F_{di-j}^2}\left(\frac{\mathrm{erf}^{-1}(\alpha)}{\epsilon}\right)^2} \end{aligned} \tag{3.28}$$

Fully refined hemisphere The resolution of the hemisphere used in the second pass is defined by the relation (3.28). A complete refinement of the first hemisphere is performed, without any consideration for the local shadow ratios computed during the first pass. Each hemisphere implies generation of M^2 rays. In this approach, the results of the first pass are not completely reused but it is the only way to insure the accuracy of the ray tracing.

Another method has been designed on the basis of the first pass. It can induce a bias in the results and can be seen as unreliable. But it can be used if the level of accuracy obtained with the first pass is similar to the required level (*i.e.* if we want to improve the accuracy of one digit).

Partly refined hemisphere From the first pass, the distribution of the shadow ratios across the cells of the hemisphere is obtained (see Figure C.2 in Appendix C). If the four corners of a cell are covered by the projection of the patch P_j, the cell is supposed totally covered by the projection. On the other hand, if the four corners are not covered by the projection, the corresponding shadow ratio is zero. So the error is supposed to be located on the cells characterized by a partial coverage.

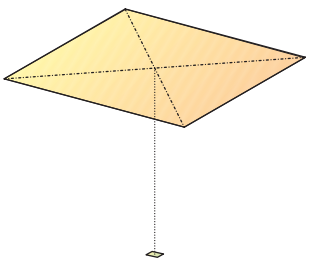
At the end of the first pass, a new resolution of the hemisphere is computed. The partly refined hemisphere method consists in decomposing the contour's cells to locally achieve the resolution imposed by the equation (3.28) and keeping the other cells unchanged. Let M_1 be the resolution of the hemisphere used in the first pass. Let n_B be the number of totally covered cells. Let N^2_{cell} be the number of new cells obtained by the decomposition of each old erroneous cell. Finally, let us suppose that $N2$ rays have been intercepted by P_j during the second pass. The final view factor is given by the following relation:

$$F_{di-j} = \frac{n_B + \dfrac{N2}{N^2_{\text{cell}}}}{M_1^2} \tag{3.29}$$

The advantage of this method is to limit the number of rays during the second pass. It is sufficient to reduce the relative error with a small additional cost but it introduces a bias that can not be removed. This implies that the convergence of this approach is limited. In practice, this method can be used if the initial guess is closed to the desired accuracy. The reached accuracy can be checked by considering the number of rays traced during the first pass, the estimation of the view factor obtained from the shadow ratios and the estimation of the error term. If the reached accuracy is too far from the desired one, a complete second pass is necessary.

3.1.6.6 Comparison of the strategies

Here we consider the case of the point wise view factor to a square, centered vertically above the origin, in a parallel plane, at a distance equal to the length of the square. The analytical value is 0.23946. The first pass is performed with 2 500 rays. 1 000 simulations are considered in order to establish statistics. The maximum accepted relative error is equal to 0.1% with a confidence interval α equal to 99.9%. This level of accuracy is superior to the levels commonly used in ray tracing (ϵ from 1 to 5% with α between 90 to 95% [Coo94]).



The obtained results are given in Table 3.1. The value obtained by computing the average of the shadow ratios is a good approximation of the reference value. If the first

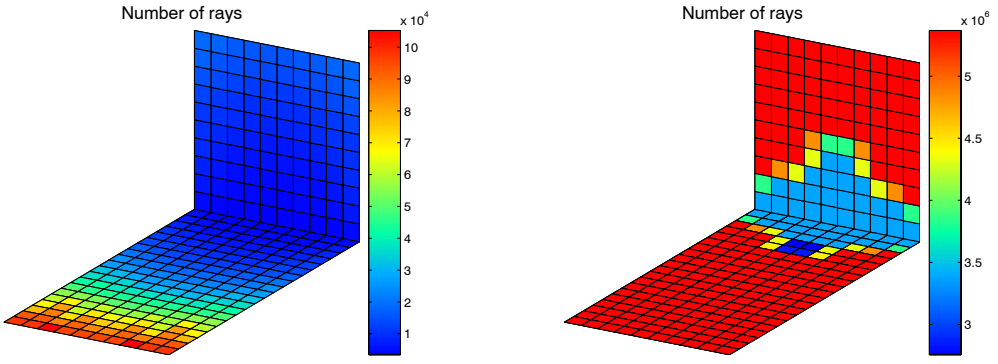
pass is used to refine the cells of partial shadow ratios, the relative error is reduced by a factor 2. The refinement improves the result of one digit. But this result presents a bias, as mentioned before. With a complete second pass, a result of higher quality (more than two order magnitude improvement) can be obtained.

Method	Value	Relative error
Reference value	0.23946	
Shadow ratio	0.23880	$2.7395\,10^{-3}$
Partially refined hemisphere	0.23920	$1.0897\,10^{-3}$
Fully refined hemisphere	0.23946	$1.6041\,10^{-5}$

Table 3.1: Average of the computed view factors.

3.1.6.7 Application

The case of two perpendicular rectangles sharing a common edge is considered. Each surface is meshed into square elements. The statistical accuracy control is used in order to predict the number of rays necessary to reach the desired accuracy. Two cases are considered. In the first one, the statistical accuracy control is based on the view factor from an element to the whole surface in front of it. In the second case, this is the view factor between each couple of elements which is used. The number of rays is displayed on the next two figures, for a same level of accuracy. Figure 3.16–a represents the result of the first case while Figure 3.16–b corresponds to the second case. The number of rays needed by the first case is reduced by two orders of magnitude. We can also remark that the distributions are quite different.



(a) Global view factors

(b) Individual view factors

Figure 3.16: Number of rays – a) Global view factors – b) Individual view factors.

From a thermal point of view, we can assess that the conduction will introduce heat fluxes between neighboring elements. This will attenuate the thermal gradients between elements induced by the radiative transfer. The quantity that must be accurately computed is the surface view factor, from an element to the whole surface facing it.

In terms of efficiency, the first case requires less computation performance. The second case requires a large number of rays, which is uselessly time consuming.

3.1.7 Application to a finite element mesh

The stratified hemisphere method, presented in the previous sections, has been designed to compute point wise view factors, that is view factors from points to their environments. The hemisphere method optimizes the discretization of the directions. In finite element software like Samcef as in space thermal software like Esarad and Thermica, view factors between finite areas are required to solve the thermal problem. The point wise view factors F_{di-j} need to be integrated on the surface of the emitter A_i. In this Section,three different ways to perform this integration are presented. The first one is the way used in a pure Monte Carlo ray tracing and cannot be combined with the stratified hemisphere method. The two other ways are based on uniform and Gauss quadrature rules. At the end of this Section, we present also how the stratified hemisphere can be used to compute finite element view factors in such a manner that no surface integration is necessary.

3.1.7.1 Monte Carlo ray tracing

In a classical Monte Carlo ray tracing, there is no optimization of the rays directions. Moreover, the origins of the rays are randomly generated across the emitter's surface. If N rays are generated, the hemisphere method can be used to compute N directions. The N origins of the rays must be uniformly distributed across the emitter in order to correctly integrate the view factor. The drawback of this approach is that the structure of the stratified hemisphere is lost, the previous demonstration based on the shadow ratios is no longer valid. The performance of this method is comparable to a cosine law ray tracing, the optimization of the hemisphere is lost.

3.1.7.2 Uniform integration

Another way to integrate the view factors is to use sample points distributed across the emitter's surface. Each sample point is used as the origin of a complete hemisphere. The integration is led by the following equation:

$$F_{i-j} = \sum_{k=1}^{n} W_k F_{di-j}^k \tag{3.30}$$

where

- n is the number of integration points on the surface i;
- F^k_{di-j} denotes the view factors computed from the integration point k;
- W_k is the weight of the point k.

The weight W_k is equal to the fraction of the area A_i associated with point k. In the case of uniformly distributed points, all the weights are equal to $\frac{1}{n}$. This method is expected to yield more accurate results than the previous one.

3.1.7.3 Gauss integration

The last method consists in using Gauss quadrature to compute the integration. This method is similar to the previous one, where uniformly distributed points were used. Now, the position of the points is optimized. The positions and weights of the Gauss points are given in Table 3.2 in function of the degree of integration.

n	x_i	W_i
2	$\pm\frac{1}{3}\sqrt{3}$	1
3	0	$\frac{8}{9}$
	$\pm\frac{\sqrt{15}}{5}$	$\frac{5}{9}$
4	$\pm\frac{\sqrt{525-70\sqrt{30}}}{35}$	$\frac{18+\sqrt{30}}{36}$
	$\pm\frac{\sqrt{525+70\sqrt{30}}}{35}$	$\frac{18-\sqrt{30}}{36}$
5	0	$\frac{128}{225}$
	$\pm\frac{\sqrt{245-14\sqrt{70}}}{21}$	$\frac{322+13\sqrt{70}}{900}$
	$\pm\frac{\sqrt{245+14\sqrt{70}}}{21}$	$\frac{322-13\sqrt{70}}{900}$

Table 3.2: Positions and weights of the Gauss points.

These points are symmetrically distributed on the interval $[-1, 1]$. For a given degree, the sum of the weights is equal to 2. In 2D, the points are distributed on the isoparametric square, represented in Figure 3.17. The weight of a point is obtained by multiplying the weights corresponding to the position of the point along the two directions. Symmetrical points are characterized by a same weight. For a given degree, the sum of all the weights is equal to 4, the area of the isoparametric square.

The position of the Gauss points is chosen in order to reduce the integration error. It has been proven that this method is the most accurate. It is also shown in the next results.

Finally, in the case of surfaces sharing common edges, to place integration points along the edges can lead to significant errors, because of the intersection tests. It is then necessary to choose integration points inside the emitters surfaces. Gauss points verify this condition.

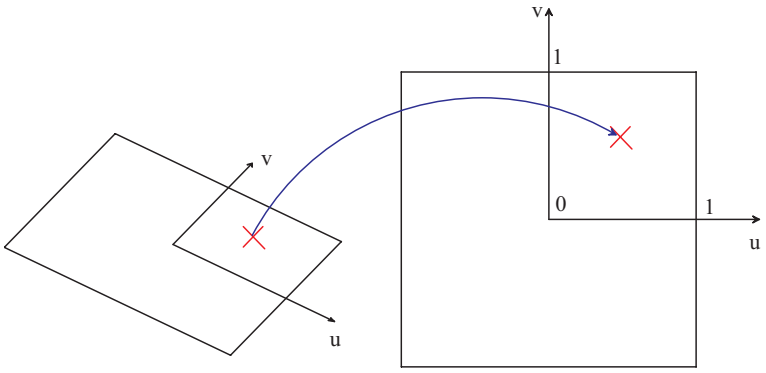
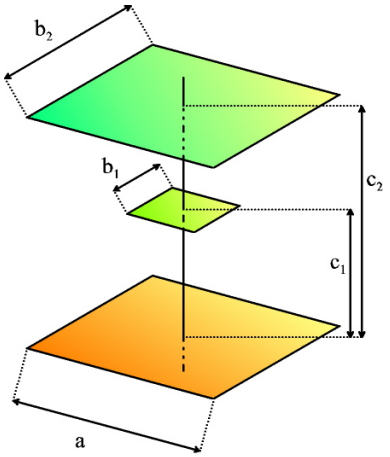


Figure 3.17: Quadrangle - isoparametric representation.

3.1.7.4 Results

In this section, we compute the view factors between two squares, in parallel planes, separated by a distance c_2. The emitter is characterized by a length a while the receptor has a side equal to b_2. An obstacle is placed between the emitter and the receptor. This obstacle is defined by the parameters b_1 and c_1. The centres of the emitter, the receptor and the obstacle are aligned along an axis perpendicular to the surfaces.



This configuration is displayed on the right, where the emitter is colored in orange, while the receptor and the obstacle are in green.

In this document, for the sake of brevity, we only show the results obtained with the Gauss quadrature. In the case of the Gauss quadrature, the evolution of the error is presented in Figure 3.18. The blue curve represents the theoretical evolution of the stratified hemisphere's error while the magenta curve is the error introduced by the numerical integration, linked to the number of integration points. If the number of integration points is increased, the magenta curve goes to lower values. By choosing the correct number of points, the integration error is made negligible when compared to the hemisphere's error, which becomes the dominant error. The formula obtained in the framework of the point wise hemisphere can be extended to patch-to-patch view factors if the number of integration points is sufficient. For applications, 2×2 or 3×3 integration points are sufficient. The green curve correspond to the evolution of the computed error when the resolution M increases. The error decreases when the number of rays increases, which is an illustration of the convergence of the method. We can also note that the green curve (hemisphere's error) is inferior to the blue curve (theoretical value).

Finally, we can notice that the cases with and without obstacles yield similar results. The Gauss integration can be used in the case of occlusions without any loss of accuracy.

Remark : let us note that this remark can be thought to be in contradiction with the Samcef method (.R3D command), which switches to uniform integration when obstacles are announced by the user. The fact is that the Samcef method and the hemisphere do not compute the same quantity. The stratified hemisphere method computes a vector of view factors from the origin to the complete environment by tracing a large number of rays. The Samcef method computes a view factor between a couple of surfaces by shooting a limited set of rays linking the integration points. In the second case, the presence of obstacles is more detrimental.

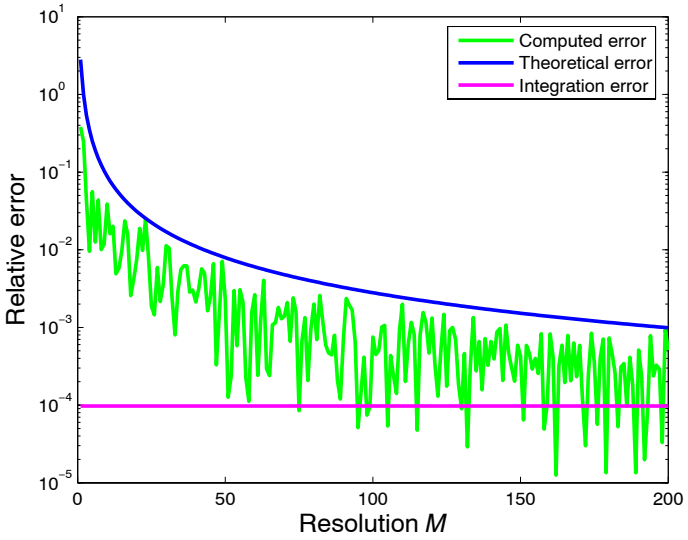


Figure 3.18: Gauss error with obstacle.

3.1.7.5 Illustration

To illustrate the use of the hemisphere method combined with Gauss integration, we have considered the case of two rectangles, perpendicular to each other and sharing a common edge. Figure 3.19–a represents the difference between the analytical solution and the result obtained with only one Gauss point for each patch. This result corresponds to the point wise view factor from the centre of each patch to the other surface. The distribution of the error is located along the common edge. In this case, the error induced by the degree of integration is larger than the error of the hemisphere method. It is necessary to increase the number of integration points.

If we increase the number of integration points (2×2 in Figure 3.19–b), we reduce the integration error and we can observe a random error across the two surfaces. This error is smaller than the previous one, of one order of magnitude. In this case, the error due to

the hemisphere is predominant. The distribution of this error is random and its average is zero.

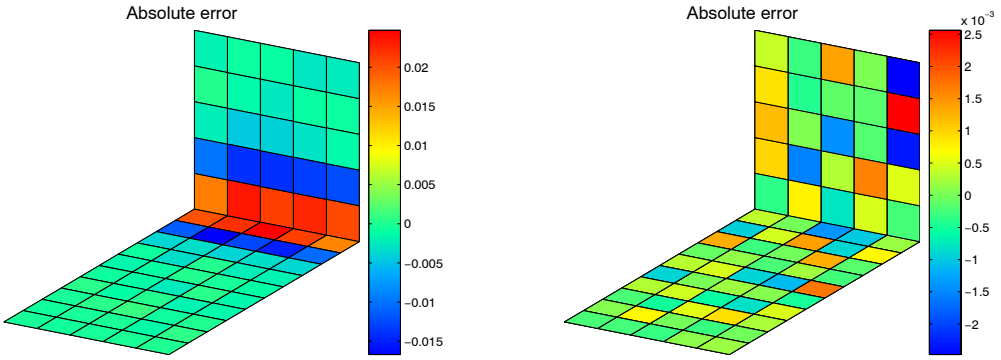


Figure 3.19: Difference of the view factors – a) 1 Gauss point – b) 4 Gauss points.

3.1.7.6 Finite element view factor

The finite element view factor is presented in a dedicated section and is given by equation (6.23):

$$F_{dA_l}^{i-k} = \int_{A_k} N_l(r_k) dF_{dA_i - dA_k}^{s}$$

where the top indices i and k refer to the surfaces while the down ones correspond to the point of origin dA and node l.

The computation of finite element view factors gives access to a linear (or quadratic) field of view factors. It is really interesting for situations governed by radiation, in order to obtain a more accurate representation with few additional amount of computation. In Section 6.1.5.1, we explain how to extend the ray tracing to distribute the incoming energy among the nodes of an impacted element. Here, we show how the hemisphere method can be used for the generation of the rays by simply placing one hemisphere above each vertex of the finite element mesh. Then a linear interpolation can be used to display the behaviour of the view factors' field. If we consider elements of higher degree, we place one hemisphere above each node and use the shape functions of the new element to represent the field of the view factors.

3.1.8 An alternative implementation: the sphere

In this Section, an alternative implementation to the stratified hemisphere is introduced. This new method has the same statistical properties as stratified hemisphere. Its advantage

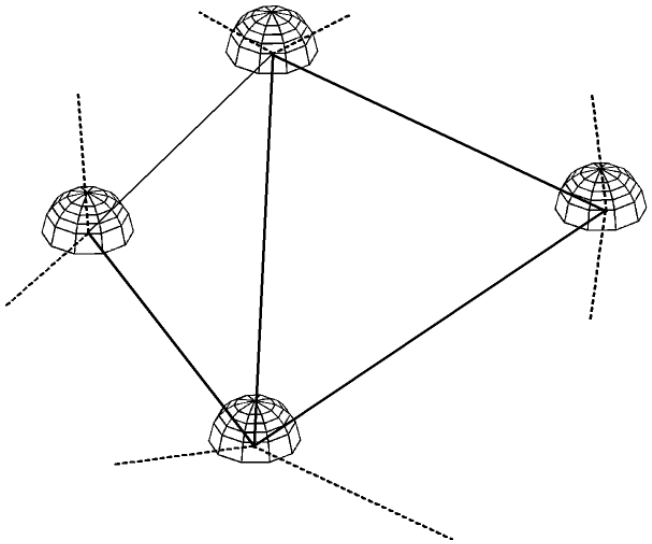


Figure 3.20: Finite element view factor - emission of the rays.

is that its application is simpler than the hemisphere's one. With stratified hemisphere, the rays are generated in function of the local normal. A change of coordinates is necessary to transpose these rays in the 3D model. The new method allows us to prevent the change of coordinates and to accelerate the generation of the rays.

3.1.8.1 Presentation of the method

This method is inspired from [Kow99]. For each origin, a sphere is built, the centre of which is placed along the local normal. Random points are uniformly generated across the surface of the sphere, yielding the directions of the rays. The points must be uniformly distributed on the surface of the sphere. This uniform distribution is linked with the *method of the inner sphere*, as it is presented in the book of Modest [Mod03].

The geometrical configuration of the sphere is presented in Figure 3.21, where the origin is located on the blue patch. The local normal is represented by the red vertical line, of unit length. The centre of the unit sphere is placed at the end of the local normal. In this Figure, the sphere is meshed in cells of equal area.

The direction of the news rays is given by the following relation:

$$\overrightarrow{r_d} = \overrightarrow{P_R} + \overrightarrow{n} + \begin{pmatrix} \sin(\gamma)\cos(\chi) \\ \sin(\gamma)\sin(\chi) \\ \cos(\gamma) \end{pmatrix} \tag{3.31}$$

where $\overrightarrow{P_R}$ designates the origin of the ray, $\overrightarrow{n}$ corresponds to the local normal. γ and χ respectively represent the latitude and the longitude. The advantage of this method is to

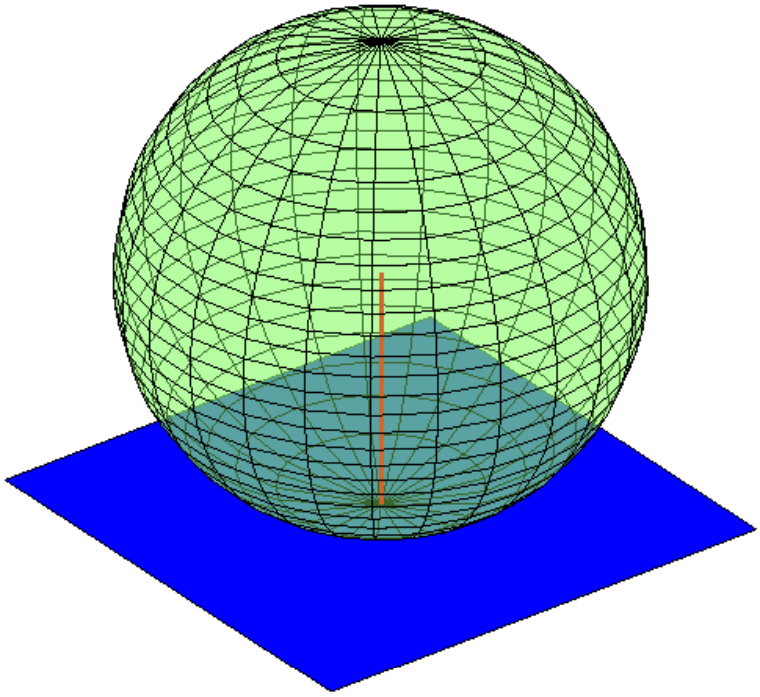


Figure 3.21: Construction of the sphere.

not require any change of coordinates.

3.1.8.2 Application to the stratified hemisphere method

In the framework of the stratified hemisphere, the link with the hemisphere method can be established. The sphere is decomposed into cells of equal area. Let γ note the latitudes and χ the longitudes of the sphere. The latitude γ is given by the following relation:

$$\gamma = \arccos(2\xi_1 - 1) \tag{3.32}$$

A sequence of uniform random variables $\xi = [0, \frac{1}{M}, ...\frac{M-1}{M}, 1]$ is defined, where M is the resolution. The following sequence of latitudes must be compared to the sequence of latitudes of the hemisphere.

$$\left\{ \begin{array}{rcl} \gamma_i & = & \arccos\left(\frac{2i}{M} - 1\right) \\ \theta_i & = & \arcsin\left(\sqrt{\frac{M-i}{M}}\right) \end{array} \right. \tag{3.33}$$

The following relation between the two families of latitudes can be easily established:

$$\gamma = 2\theta \tag{3.34}$$

In order to illustrate the equivalence between the two methods, the distribution of the two latitudes are plotted in Figure 3.22–a. The red crosses correspond to the points

uniformly distributed on the sphere, following the distribution (3.33). The blue lines correspond to the directions generated with the stratified hemisphere. The length of these lines has been computed in such a way the extremity is located on the unit sphere.

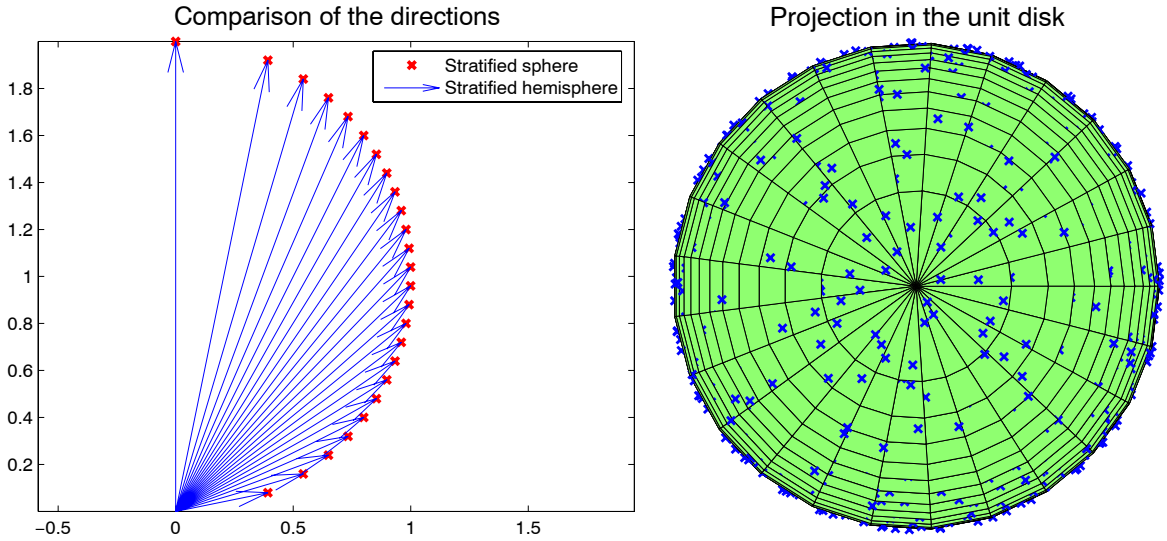


Figure 3.22: Sphere – a) comparison of the directions – b) projection of the samples.

The two distributions coincide exactly. However, this coincidence is only rigorous when the local normal corresponds to the z-axis. To obtain a random method with the same statistic properties as the hemisphere method, one random point must be generated into each cell. The distribution must be uniform in terms of area. Two uniform random numbers ξ_1 and ξ_2 are generated and introduced into the following relations:

$$\left\{ \begin{array}{rcl} \gamma_i & = & \arccos\left(1 - \frac{2(i-\xi_1)}{M}\right) \\ \chi_j & = & \frac{2\pi}{M}(j - \xi_2) \end{array} \right. \tag{3.35}$$

The optimization obtained with the stratified sampling in the case of the hemisphere is still working for the sphere because the two formulations are based on the same hypothesis. The obtained results are identical but we observed a reduction of the CPU time of 30% in the case of diffuse view factors. The advantage is less important if we model the specular of reflection, because the optimization only concerns the emission of the rays.

To illustrate the equivalence of the two distributions, the points of the sphere are projected onto the hemisphere and then onto the unit disc. The result is presented in Figure 3.22–b. The distribution is correct; each cell in the unit disc contains one and only one sample.

3.1.9 Comparison between CLRT and stratified hemisphere method

In this section, we will justify the superiority of the hemisphere method on cosine law ray tracing.

3.1.9.1 Discussion

The introduction of two random variables in the initial hemisphere method, *i.e.* in a regular grid, will lead us to an optimized random method.

In the case of a "blind" Monte Carlo method, the rays are casted in arbitrary directions. The samples can be badly distributed over the domain. To avoid samples being clumped together, *stratified sampling* has been used. This is a way usually used to decrease the variance of random methods. The domain is decomposed into sub-domains called *strata*. One sample is generated into each stratum. This method prevents the clumping of the samples. The statistical stratified hemisphere naturally takes advantage of this method by decomposing the hemisphere into cells.

Another way to reduce the variance of statistical processes is called *importance sampling*. This has already been presented for Malley's method and cosine law ray tracing. Instead of using an uniform probability distribution function (PDF) to generate samples, importance sampling is based on a non uniform distribution function. If we want to generate samples according to a given PDF, a good sampling function matches the shape of the PDF as closely as possible [DBB03]. This is done by the hemisphere method when decomposing the hemisphere into cells of equal view factors. In this case, the PDF is given by:

$$p = \frac{\cos(\theta)}{\pi} \tag{3.36}$$

and the decomposition is performed according to this function.

The stochastic hemisphere is the natural combination of these two ways designed to reduce the variance of statistic processes. This allows us to assume that the stochastic hemisphere is a very efficient random method to compute diffuse view factors.

3.1.9.2 Gain of the hemisphere

We have established the ratio between the numbers of rays required by both methods to achieve the desired accuracy. These numbers are given by the equations (3.18) and (3.20). The parameter K (equation (3.21)) is defined to simplify the computed ratio. The following

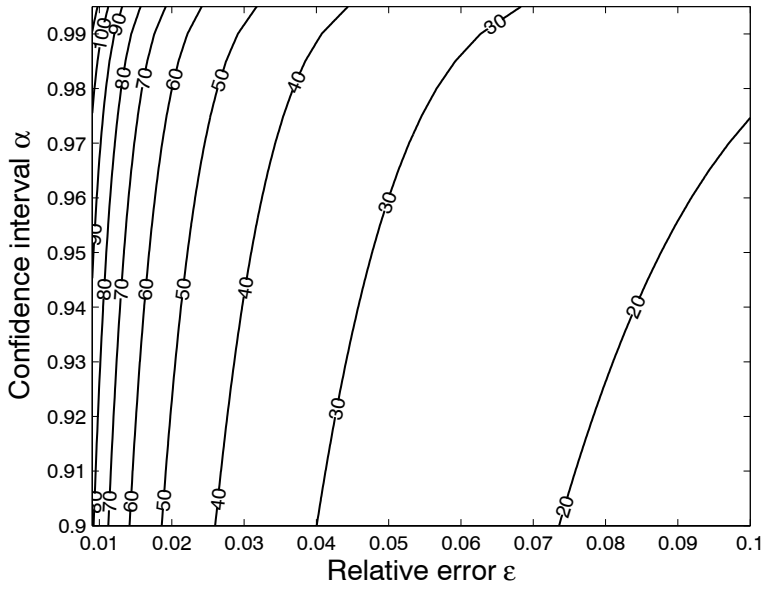


Figure 3.23: Variation of K in function of α and ϵ.

relationship is obtained:

$$\frac{N_{\text{CLRT}}}{N} = \frac{2^{\frac{5}{3}}}{\sqrt[3]{c^2}} \left(\frac{\text{erf}^{-1}(\alpha)}{\epsilon} \right)^{\frac{2}{3}} (1 - F_{di-j}) \tag{3.37}$$

$$= \frac{K}{\sqrt[3]{c^2}} (1 - F_{di-j}) \tag{3.38}$$

So the gain is directly proportional to the parameter K. The evolution of this parameter K in function of α and ϵ is given in Figure 3.23. α is defined between 90 and 99.9% while ϵ varies from 1 to 10%. K increases with the accuracy, that is when the relative error decreases and the confidence interval increases. The parameter K can achieve values larger than 100 for relative errors of 1% and confidence intervals larger than 99%.

The evolution of the gain is plotted in Figure 3.24–a, in function of the view factor and the maximum relative error, for α equal to 99%. For a realistic value of ϵ, between 1% and 10%, the superiority of the hemisphere is clear. For a fixed accuracy, the gain is directly proportional to the view factor.

The evolution of the maximum relative error achieved with a fixed number of rays is given in Figure 3.24–b, for N equal to 1 000, 10 000 and 100 000 rays. The S-shape discontinuous curves correspond to a cosine law ray tracing. The continuous curves correspond to the stratified hemisphere. The red curve represents the intersections of the two families of curves. For a realistic number of rays, the stratified hemisphere yields smaller levels of relative error than the cosine law method, for nearly all configurations.

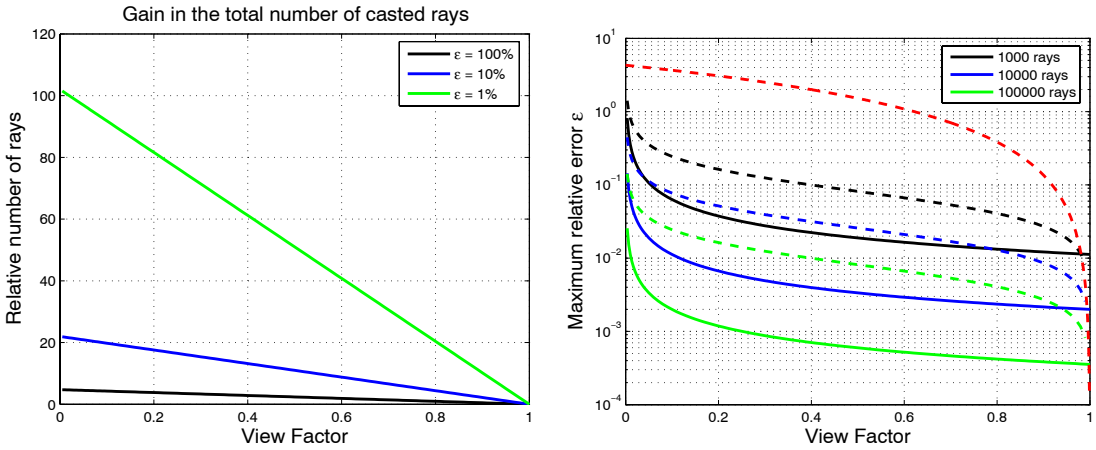


Figure 3.24: Gain of the hemisphere – a) Gain of the hemisphere with respect to CLRT – $\alpha = 99\%$ – b) Variation of the maximum relative error in function of the number of rays.

To complete this analysis, let us consider the example of Section 3.1.6.4, detailed in Appendix C, tested with a cosine law ray tracing. By combining the equations (3.22) and (3.23), the parameter K is removed. It yields the following relation which is used to predict the number of rays that must be shot by the cosine law ray tracing to achieve the same accuracy as the stratified hemisphere with N rays, for the same geometrical configuration.

$$N_{\text{CLRT}} = \frac{4\left(1 - F^*_{di-j}\right)\sqrt{F^*_{di-j}}\sqrt{N^3}}{c} \tag{3.39}$$

In this case, the geometrical constant c has been found to be equal to 2.4. A cosine law ray tracing will need more than 550 000 rays to achieve the same level of accuracy. This is shown in Figure 3.25. 100 000 simulations have been performed in order to establish statistics. The blue curve corresponds to the evolution $\{\alpha, \epsilon\}$ obtained with the cosine law ray tracing and 553 536 rays while the green curve represents the evolution obtained with only 10 000 rays and the stratified hemisphere. The stratified hemisphere method provides better results than the cosine law ray tracing, despite a smaller number of rays.

3.1.9.3 Homogeneity of stratified hemisphere

In this Section, we want to establish that the distribution of directions obtained with stratified hemisphere is more homogeneous than with a classical cosine law ray tracing. In order to obtain a measure of the uniformity of the two distributions, the same number of points are generated in the unit disc. A Delaunay mesh is then used to build an unique triangular mesh. On the basis of the numerous triangles of the two meshes, statistics can be established. The mean area of the triangles is computed for each distribution, as well

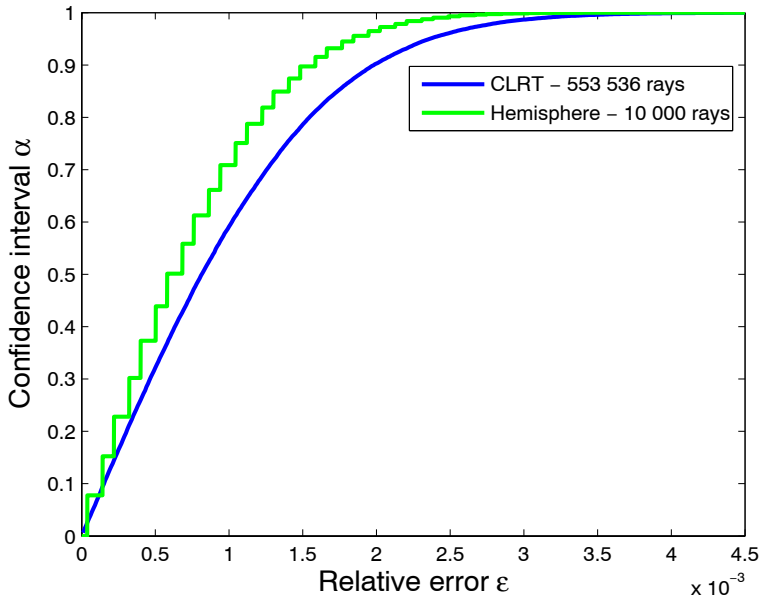


Figure 3.25: Evolution of the confidence interval in function of the relative error.

as the associated variance. The more uniform distribution, the smaller variance.

The generated random points are projected onto the unit disc. Figure 3.26–a corresponds to stratified hemisphere. The same number of points are generated with a cosine law formulation[3]. These points are represented in Figure 3.26–b. The points generated with the stratified hemisphere method are more uniformly distributed than the points generated with the CLRT formulation. Less clumps of samples, less regions empty of samples are observed with the stratified hemisphere method.

To quantify the uniformity of a distribution in the unit disc, the corresponding Delaunay mesh is built. The triangulation is unique for a given distribution of points. The area of each triangle is computed, as well as the average area and the standard deviation of these triangles. A uniform distribution is characterized by triangles of the same area, the associated variance is small. Figure 3.27–a corresponds to the mesh built with the stratified hemisphere's points while Figure 3.27–b has been obtained from the CLRT distribution. At the first glance, the hemisphere's triangles seem more uniform than the CLRT's ones.

The evolution of the standard deviation of the triangles' areas generated by the hemisphere method and the cosine law formulation are plotted in Figure 3.28–a, in function of the resolution (square root of the number of samples). The stratified hemisphere method

[3]In fact, two cosine law formulations have been implemented. The first one is based on the angular parameters, as the method used in classical ray-tracing algorithm. The second one is based on a sampling of the domain $[-1, 1]^2$ with rejection of the points outside the unit disc, for the same number of samples (see *rejection sampling* [Gra78]).

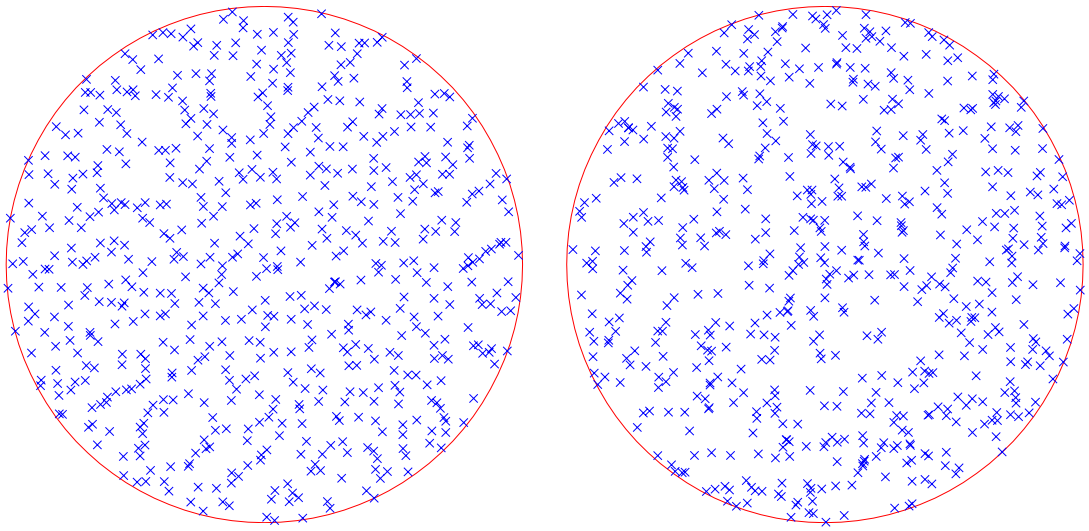


Figure 3.26: Distribution of the points in the unit disc – a) stratified hemisphere – b) CLRT.

(magenta curve) is characterized by a standard deviation smaller than the two cosine law formulations (blue and green curves). This allows us to state that the hemisphere method can yield a more uniform distribution than the classical CLRT. Figure 3.28–b represents the relative error on the total area. The area of the unit disc is equal to π. No triangular mesh can exactly represent a curved surface. However, the error associated to the hemisphere method is smaller because this method allows to better sample the contour of the disc.

3.1.10 Conclusion

In this Section, a new method for the generation of rays has been presented. The stratified hemisphere must be considered as an optimized method which must be included in a more complex ray tracing algorithm. The establishment of the stratified hemisphere mesh has been detailed. A random ray generator based on this deterministic grid has been developed. A statistical accuracy control has been established and the superiority of the new method with respect to cosine law ray tracing has been demonstrated. The extension of the method to the computation of view factors between surfaces of finite size has been studied. Another faster implementation of the method has also been given.

The stratified hemisphere method is adapted to diffuse emitters. This is due to the use of the cosine law when generating the rays. If the emission is not diffuse but directionally dependent, the stratified hemisphere can still be used. The θ-distribution must be changed in order to uniformly sample the new emission function. The stratified sampling has to be

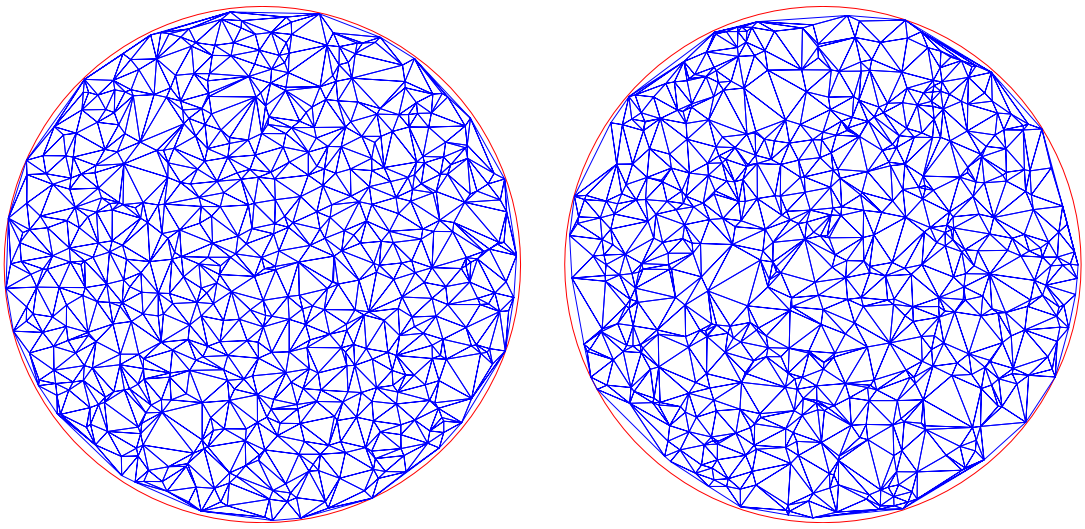


Figure 3.27: Delaunay mesh in the unit disc – a) stratified hemisphere – b) CLRT.

applied to the new emission function. This development has not been performed in this thesis, because diffuse emission was an hypothesis of this work. Nevertheless, it could be an interesting extension to the hemisphere method.

In the next Section, we present a possible extension to more complex thermo-optical properties, including specular and glossy reflections.

3.2 Extension to specular and glossy reflections, specular transmission

3.2.1 Introduction

The previous Section was dedicated to the computation of diffuse view factors, between diffuse reflectors. In terms of ray tracing, it implies that only the first intersection of the rays emitted with the stratified hemisphere method needs to be computed. No secondary ray is re-emitted. The diffuse view factors only depend on the geometrical configuration; they do not vary with the thermo-optical properties. A unique set of view factors needs to be stored, even if multi-wavelength problems are considered. In this Section, an extension to more complex thermo-optical properties, including specular reflections and transmissions, has been studied.

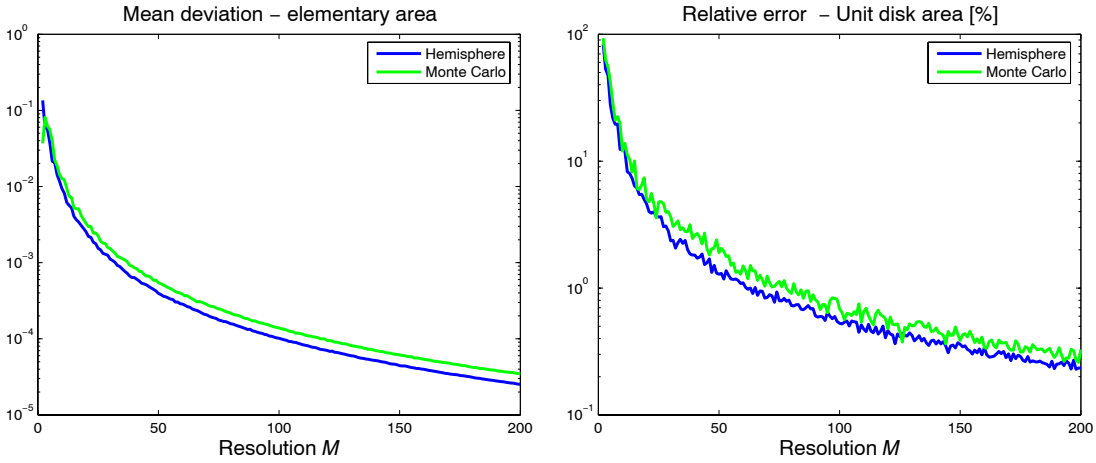


Figure 3.28: Deviation and error – a) Standard deviation – b) Sampling error.

3.2.2 Specular reflection

In case of surfaces which are partly or purely specular reflectors, the view factors must be extended in order to add additional, specular paths between the surfaces. These new view factors are called extended or specular view factors. During the ray tracing process, when a specular surface is impacted by a ray, a secondary ray is emitted in the mirror-direction.

Reflected direction and energy Given an incoming ray $\overrightarrow{r_i}$ and the local normal vector $\overrightarrow{n}$ at the intersection point, the outgoing direction $\overrightarrow{r_r}$ is given by relation (2.23):

$$\overrightarrow{r_r} = \overrightarrow{r_i} - 2\,(\overrightarrow{r_i} \cdot \overrightarrow{n})\,\overrightarrow{n}$$

The energy E_r of the outgoing ray is attenuated by the specular reflectivity ρ^s of the intersected surface:

$$E_r = \rho^s E_i \tag{3.40}$$

Termination criterions During a ray tracing, each ray can experience a large number of specular reflections. A termination criterion is needed in order to insure that the algorithm will stop. In this work, four kinds of termination criterion can be considered.

1. The ray impacts a non-specular surface and no secondary ray is re-emitted.
2. The ray leaves the geometrical model through an aperture and is supposed to reach the deep space.
3. The number of intersections experienced by the ray goes above a fixed threshold.
4. The energy associated with the ray goes below a fixed threshold; this energy is then neglected.

Each of these criteria implies the extinction of the current ray. The third criterion will introduce a bias in the computation of view factors because a fraction of the light transport is neglected. This can be made negligible if the corresponding threshold is not too strict.

The fourth criterion is purely deterministic. Its drawback is that it introduces a bias in the computed values by truncating the paths of each ray. This criterion can be replaced by a stochastic process named *Russian roulette*, designed in neutron transport simulation [LM84, Cla71, SG69] cited in [AK90] and [OY66]. This technique is based on the weight, that is the energy associated with the ray. Once the ray's energy ω is beyond a given threshold Ω, the ray is stochastically stopped with probability P. A uniform random number ξ is generated; it is tested with respect to P. If ξ is higher than P, the ray survives and the associated weight ω is multiplied by $\frac{1}{1-P}$. The energy of the outgoing ray is no longer given by the equation (3.40). The expected value of the weight of the secondary ray is given by the following relation:

$$E(W) = \mathcal{P}(\text{termination})0 + \mathcal{P}(\text{survival})\frac{\omega}{1-P} \tag{3.41}$$

As the probability of survival is equal to $1 - P$, the expected value is given by:

$$\begin{aligned} E(W) &= (1-P)\frac{\omega}{1-P} && (3.42)\\ &= \omega && (3.43) \end{aligned}$$

The energy of the outgoing ray is equal to the energy of the incident ray, on average. This method increases the variance of the whole process but it removes the bias associated with the fourth criterion. Moreover, this variance can be reduced if the weights of the rays are sufficiently low (defined by the chosen threshold Ω).

Multi wavelength problems If the thermo-optical properties of a model are assumed to be constant for all wavelengths that govern the problem, the surfaces are said to be *gray*; only one set of view factors needs to be computed and stored.

Generally, for space applications, at least two spectral bands are necessary. The first one corresponds to the solar and albedo heat fluxes while the second one is linked to the infrared fluxes from the planet and from the surfaces of the model. In each spectral band, the thermo-optical properties are supposed to be constant.

In some applications, more spectral bands are needed [Pan05, Rue08]. It is the case for low temperature applications, such as cryogenics satellites, like Planck [ESAb] and Herschel [ESAa]. In this case, the *semigray approximation* is necessary. The spectral domain is broken into separate wavelength bands. In each band, the Kirchhoff's law[4] is supposed to be satisfied; each node radiates a fraction of energy related to that specific band, instead

[4]Following the Kirchhoff's law [SH01], the spectral emissivity of a surface k, in a given direction, is

of σT^4. This procedure is described in the references [Mod03, SH01]. For each spectral band, a complete set of view factors needs to be computed and stored.

If the spectral distribution of the thermo-optical properties is not a function of temperature, no temperature iteration is needed. But if the spectral distribution of some thermo-optical properties is temperature-dependent, the problem must be iteratively solved and new view factor matrices need to be re-computed for the new temperatures.

In order to reduce the CPU time needed for the computation of the view factors, the following innovative strategy is proposed. Only one ray tracing process can be used for all the n wavelength bands. Each ray is associated with a vector of n weights. For each intersection, each weight is attenuated with the corresponding specular reflectivity, in its wavelength band. As long as at least one weight is superior to a given threshold, the ray tracing is continued, and the view factor matrices are simultaneously computed for all the spectral bands. Once all the weights are below the threshold, the Russian roulette is applied.

In order to save storage, the view factor matrices must be compressed, by only keeping the non-zero elements. This compression has been developed in the framework of this thesis, as well as adapted routines for the enforcement of reciprocity, Gebhart's matrix method and thermal resolution, based on the compressed form of the view factor matrix. Although it was not a requirement of the thesis, this step was necessary to obtain a competitive algorithm.

3.2.3 Glossy reflection

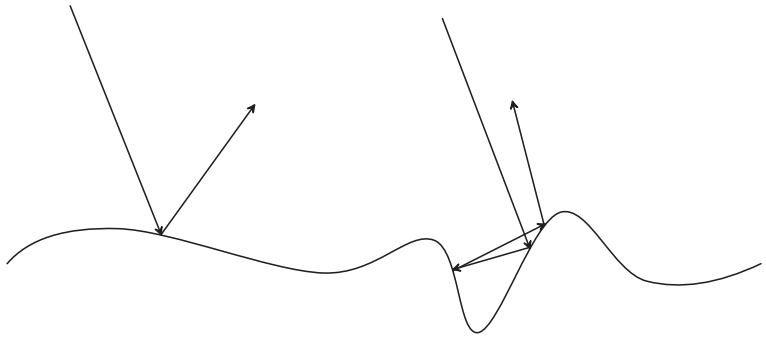


Figure 3.29: First reflection and multi-reflection.

equal to the absorptivity of this surface, in the same direction:

$$\epsilon_\lambda(\lambda, \theta, \phi, T_k) = \alpha_\lambda(\lambda, \theta, \phi, T_k) \tag{3.44}$$

The main advantage of the ray tracing is its ability to model a large panel of physical phenomenons. In the previous Sections, reflectors were assumed to be ideally diffuse and specular. However, more complex reflections can easily be modeled. In [HTSG91, HHP+92, NN95], the reflectivity is decomposed into three components: the uniform diffuse one, the directional diffuse and specular ones. The latter two components are strongly directional, due to the light which is directly reflected by the surface. The first one experiences multi-reflection on the surface and subsurface scattering. In Figure 3.29, an example of first reflection by the surface (on the left) is shown while the ray on the right experiences multi-reflection.

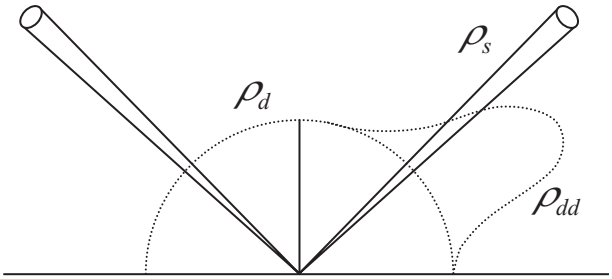


Figure 3.30: Complete BRDF.

The resulting BRDF is shown in Figure 3.30, including the three components of the reflectivity function. The two cones correspond to the incoming direction and the outgoing specular component. The dashed circle is the uniform diffuse component while the bulb corresponds to the directional diffuse term.

If the diffuse component is removed, it results in a function of the outgoing direction. This function can be randomly sampled. Several models have been designed in image synthesis, like the Phong model [Pho75], the Blinn model [Bli77], the model of Cook and Torrance [CT82] or the one of He and Sillion [HTSG91]. These models have been designed for rendering. Some are not physical. For example, the Phong model does not satisfy the conservation of energy [DBB03]. It is then not really adapted for thermal radiation.

The best solution should be to introduce data from experiments. A reflection profile could be established from these data. An adapted Monte Carlo sampling could then be used to generate the direction of the reflection [LRR04]. In the framework of the thesis, the available data are too weak to implement such methods but it should be considered for future developments.

3.2.4 Transmission

The transmission can also be modeled during the ray tracing process. The transmission occurs when a ray encounters a semi-transparent surface. Part of the incoming ray is

refracted in the direction given by the Snell-Descartes's law:

$$n_2 \sin(\theta_r) = n_1 \sin(\theta_i) \tag{3.45}$$

where n_1 and n_2 are the indices of refraction of the first and second media; θ_i is the angle of incidence and θ_r is the angle of refraction.

The computation of the extended view factors is performed as follows. In the case of a ray impinging surface k which is a partly specular and transparent, a uniform random number ξ is generated. The reference value is given by the following formula:

$$X = \frac{\rho_k^s}{\rho_k^s + \tau_k} \tag{3.46}$$

If the random number ξ is inferior to X, the ray is specularly reflected. If ξ is superior to X, the ray is transmitted.

This process introduces an additional random noise in the computation of extended view factors. This noise cannot be easily taken into account in the statistical accuracy measure of the stratified hemisphere so that the hemisphere's performances are no longer insured. The statistical accuracy measure of [Pla93] is then the only applicable one.

In the case of multiple wavelength bands, the developments presented in the previous Section can be extended. Only one ray tracing process is performed to compute the extended view factors. Once all the weights are below a given threshold, the Russian roulette is used to statistically interrupt the ray tracing without introducing any bias.

3.2.5 Radiative exchange factors

The radiative exchange factors are even more complex to compute by ray tracing[5]. Once a ray is emitted, it can experience specular reflection and transmission, as well as diffuse reflection. At each intersection point, a uniform random number ξ_1 is generated. It is compared to two thresholds X_1 and X_2, given by the following relations:

$$\begin{cases} X_1 = \dfrac{\rho^d}{\rho^d + \rho^s + \tau} \\ X_2 = \dfrac{\rho^d + \rho^s}{\rho^d + \rho^s + \tau} \end{cases} \tag{3.47}$$

If ξ_1 is inferior to X_1, the ray is diffusely reflected. Two additional uniform random numbers ξ_2 and ξ_3 are generated in order to compute a new direction from the cosine law. These two angles are obtained with respect to the local normal:

$$\begin{cases} \theta = \arcsin\left(\sqrt{\xi_2}\right) \\ \phi = 2\pi\xi_3 \end{cases} \tag{3.48}$$

[5]Another method, based on matrix operations, is detailed in Section 3.4. This method is called Gebhart's matrix method.

If ξ_1 is comprised between X_1 and X_2, the ray is specularly reflected. If ξ_1 is higher than X_2, the ray is specularly transmitted.

In this case, the statistical accuracy control developed for the stratified hemisphere method is no longer valid. The formulation of [HTSG91] needs to be used. Another solution consists in computing the view factors thanks to the stratified hemisphere and using the Gebhart's method to obtain the radiative exchange factors. In this case, the statistical accuracy control of the stratified hemisphere is still valid.

3.2.6 Non-diffuse emission

The hemisphere method is based onto the combination of importance- and stratified-sampling. In the case of diffuse emission, the Nusselt's analogy can be used to yield an easy-to-sample domain of definition: the unit disc. If the emission is diffuse, the importance of each elementary surface on the unit disc is constant.

If the emission is no longer diffuse, stratified sampling can still be used but importance sampling is no longer valid. If the new emissivity profile is known, an alternative projection to the Nusselt's analogy can be determined. The grid in the unit disc can be modified in function of this new projection so that the developments performed in the framework of the stratified hemisphere method remain valid.

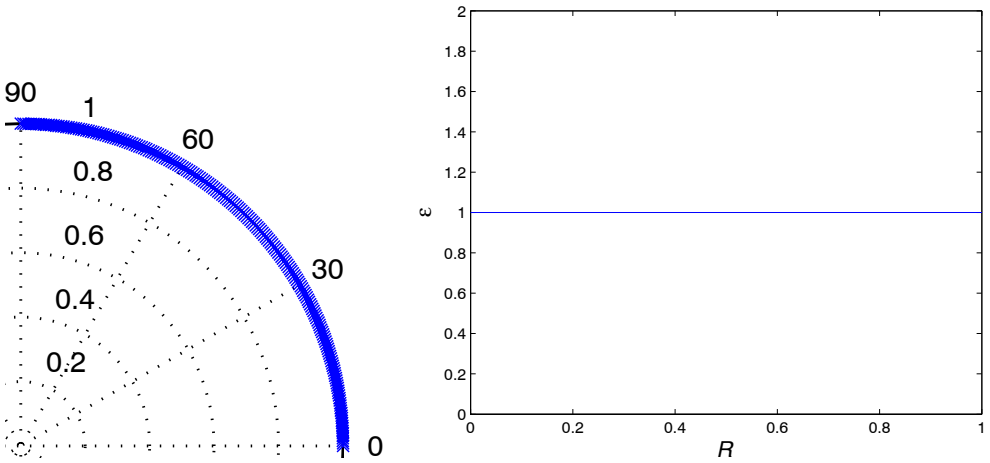
Figure 3.31 illustrates this extension of the importance sampling for non-diffuse emissivity. Figure 3.31–a represents the diffuse emissivity, which is constant on the whole hemisphere, for $\theta \in [0, \frac{\pi}{2}]$. Following the Nusselt's Analogy, the corresponding importance in the unit disc is also constant (see Figure 3.31–b). If a non-diffuse emissivity profile is considered (see Figure 3.32–a, based on a prediction of the directional emissivity of nonconductors [Mod03]), the corresponding importance in the unit disc is given in Figure 3.32–b. This function can be used in order to modify the stratified hemisphere's grid, following the non-uniform importance of the new emissivity profile.

Concerning Gebhart's method, which is based on the supposition that the emissivity is diffuse, it is our belief that the non-diffuse emissivity profile could be associated with a non-diffuse reflectivity characterized by an identical profile (eventually scaled). This could lead to new developments.

Let us recall that diffuse emissivity is an hypothesis at the basis of this work and then non-diffuse emissivity has not been carefully studied.

3.2.7 Conclusions

In this Section, possible extensions have been presented to model complex thermo-optical properties and to extend view factors, taking specular and glossy reflections into account.



(a) Diffuse angular emissivity (b) Diffuse importance in the unit disc

Figure 3.31: Extension of the importance sampling for non-diffuse emissivity.

In the next Section, the enforcement of reciprocity and closure applied to exchange factors is addressed. It allows us to reduce the error associated with the ray tracing process.

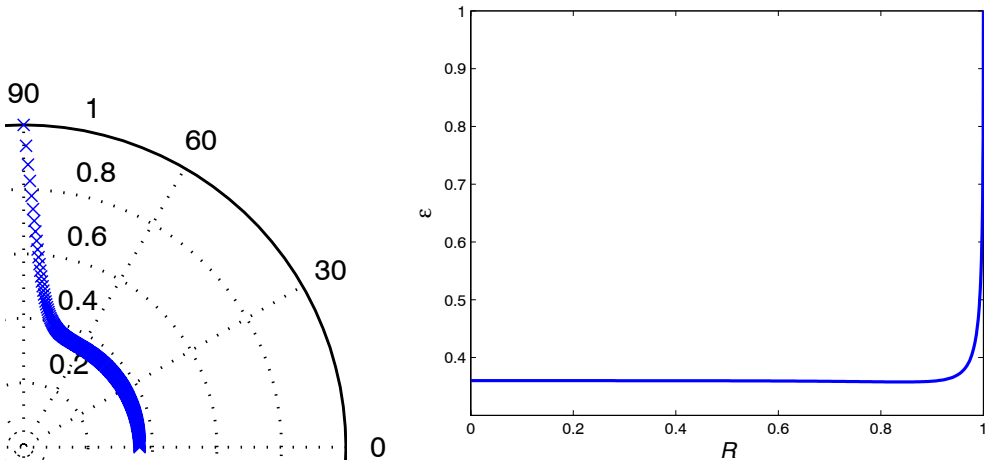
3.3 Enforcement of reciprocity and closure

3.3.1 Introduction

In heat transfer, the exchange of energy are based on adimensional numbers, which can be the geometrical view factors, the extended view factors or the radiative exchange factors. In this work, these numbers are called *exchange factors* and are noted $\mathcal{F}_{i-j}$ in order to generalize the theory presented in this Section. The exchange factors must obey the constraints of reciprocity and closure. In this thesis, the exchange factors are computed by ray tracing. By definition, the exchange factors obtained by ray tracing obey the closure relation. Because of the statistical nature of the ray tracing, the reciprocity relation is not verified, unless the number of rays approaches infinity. This results in a violation of the second law of thermodynamics and induces errors in the thermal results.

Once the reciprocity has been enforced, the obtained exchange factors do not respect the closure relation anymore. A second step of enforcement is necessary. The resulting exchange factors are more accurate than the one computed by the ray tracing process.

Moreover, the enforcement of reciprocity yields a triangular matrix. The number of exchange factors that must be stored reduces to $\frac{N(N-1)}{2}$, where N is the number of lines in the initial exchange factor matrix, that is the number of elementary surface patches in the model.



(a) Non-diffuse angular emissivity

(b) Non-diffuse importance in the unit disc

Figure 3.32: Extension of the importance sampling for non-diffuse emissivity.

In heat transfer, it has been shown that enforcing closure and reciprocity can greatly reduce the error in the thermal analysis [CS94]. Different methods have been designed to enforce closure and reciprocity [Tsu92, TL95, vL89]. In this Section, different ways are presented to enforce the closure and the reciprocity relations.

3.3.2 Generalities

First some generalities concerning the closure rule and the reciprocity rule are described.

3.3.2.1 Closure rule

The exchange factors obtained by ray tracing naturally obey the closure rule because the total energy emitted by each surface is finally associated with other surfaces of the model (or with an additional node modeling the background). This rule is equivalent to the first principle of thermodynamics, which is the conservation of the energy. If view factors are considered, all the traced rays impact a surface of the model. In the case of radiative exchange factors, each ray is finally absorbed by a surface.

This relation is valid if the N surfaces form a closed cavity. If the geometrical model presents an aperture, an additional surface is used to close the model and the closure relation concerns the $N + 1$ surfaces.

When the reciprocity is enforced, the exchange factors are modified and the closure is no longer valid. A closure enforcement is necessary.

3.3.2.2 Reciprocity rule

If the closure rule corresponds to the first principle of thermodynamics, the reciprocity can be associated with the second principle. The second principle governs the exchanges of energy between the different surfaces which compose the geometrical model. Following the second principle of thermodynamics, the net heat flux between two surfaces at the same temperature must be zero. This is used to demonstrate the reciprocity relation for each type of exchange factors.

If the closure is naturally satisfied by the ray tracing, the reciprocity is generally not verified and a reciprocity enforcement is mandatory.

3.3.3 Exchange factors

In this Section, the definitions of the different exchange factors that are used to compute the inter-reflection in a geometrical model are briefly recalled. The expression of each rule is also recalled.

3.3.3.1 View factor

The view factor F_{i-j} between two patches P_i and P_j is the fraction of the uniform diffuse radiation leaving P_i that directly reaches P_j. The reciprocity relation which governs the view factors is given by the following equation:

$$A_i F_{i-j} = A_j F_{j-i} \tag{3.49}$$

This relation is valid when the view factors are supposed to be constant across the elementary surfaces. The closure relation is expressed by the next equation:

$$\sum_{j=1}^{N} F_{i-j} = 1 \tag{3.50}$$

In this Section, $\eta_{i-j} = A_i F_{i-j} = \Omega_i \mathcal{F}_{i-j}$ denotes the estimator corresponding to the view factors. This estimator will be enforced for reciprocity and closure.

3.3.3.2 Extended view factor

The extended view factor F^s_{i-j} between two patches P_i and P_j is the fraction of the uniform diffuse radiation leaving P_i that reaches P_j, directly or after an arbitrary number of specular reflections. The reciprocity relation applied to the extended view factors is similar to equation (3.49):

$$A_i F^s_{i-j} = A_j F^s_{j-i} \tag{3.51}$$

The closure relation applied to the extended view factors is given by the following equation:

$$\sum_{j=1}^{N}(1-\rho_j^s)F_{i-j}^s = 1 \tag{3.52}$$

In this Section, $\eta_{i-j} = A_i F_{i-j}^s = \Omega_i \mathcal{F}_{i-j}$ denotes the estimator corresponding to the extended view factors.

3.3.3.3 Radiative exchange factors

The radiative exchange factor B_{i-j} from a patch P_i to a patch P_j is the fraction of the radiation leaving P_i which is finally absorbed by P_j, whatever the followed path may be. In this definition, the emission process is not specified. In this work however, only diffuse emission is considered. The reciprocity relation applied to the radiative exchange factors is given by the next equation:

$$\epsilon_i A_i B_{i-j} = \epsilon_j A_j B_{j-i} \tag{3.53}$$
$$GR_{i-j} = GR_{j-i} \tag{3.54}$$

where GR_{i-j} is the radiosity. This definition of the radiosity is different from the radiosity J which is associated with purely diffuse surfaces. Indeed, the radiosity GR represents the energy exchanged by two surfaces, while the radiosity J represents the energy emitted by a purely diffuse surface, *i.e.* the self emitted power plus the reflected component of the irradiance. The unit of J is W/m^2 while GR is an area (m^2).

The closure relation in terms of radiative exchange factors is given by the following equation:

$$\sum_{j=1}^{N} B_{i-j} = 1 \tag{3.55}$$

In this Section, $\eta_{i-j} = \epsilon_i A_i B_{i-j} = \Omega_i \mathcal{F}_{i-j}$ denotes the estimator corresponding to the radiative exchange factors.

3.3.4 Impact of the reciprocity on the variances

This Section is inspired from the thesis of C. N. Zeeb [Zee02]. The final result is of great interest for the enforcement of reciprocity. The exchange factors are assumed to be exactly known and noted $\widetilde{\mathcal{F}}_{i-j}$ and $\widetilde{\mathcal{F}}_{j-i}$. The reciprocity relation is verified and the following equations are obtained:

$$\widetilde{\eta}_{i-j} = \Omega_i \tilde{\mathcal{F}}_{i-j} \tag{3.56}$$
$$= \Omega_j \tilde{\mathcal{F}}_{j-i} \tag{3.57}$$
$$= \widetilde{\eta}_{j-i} \tag{3.58}$$

where Ω_i is a general coefficient, equal to A_i for the view factors and to $\epsilon_i A_i$ for the radiative exchange factors.

In a classical ray tracing algorithm, the exchange factor is computed by the formula $\mathcal{F}_{i-j} = \frac{N_{i-j}}{N_i}$, where N_i is the number of rays emitted from P_i and N_{i-j} is the number of these rays received by P_j. The reciprocal exchange factor is given by the similar expression $\mathcal{F}_{j-i} = \frac{N_{j-i}}{N_j}$. The uncertainty is different for the two exchange factors $\mathcal{F}_{i-j}$ and $\mathcal{F}_{j-i}$; it is measured by the standard deviation. For a classical ray tracing algorithm, the exchange factors follow a binomial law (each ray can be associated with a random variable which can take only two possible values: 1 if the ray impacts the receptor and 0 otherwise). The probability of the binomial distribution is equal to the corresponding exchange factor. The standard deviation of $\widetilde{\eta}_{i-j}$ and $\widetilde{\eta}_{j-i}$ are given by the following relationships:

$$\sigma_{\widetilde{\eta}_{i-j}} = \Omega_i \sqrt{\frac{\widetilde{\mathcal{F}}_{i-j}\left(1-\widetilde{\mathcal{F}}_{i-j}\right)}{N_i}} = \widetilde{\eta}_{i-j}\sqrt{\frac{1-\widetilde{\mathcal{F}}_{i-j}}{N_i \widetilde{\mathcal{F}}_{i-j}}} \tag{3.59}$$

$$\sigma_{\widetilde{\eta}_{j-i}} = \Omega_j \sqrt{\frac{\widetilde{\mathcal{F}}_{j-i}\left(1-\widetilde{\mathcal{F}}_{j-i}\right)}{N_j}} = \widetilde{\eta}_{j-i}\sqrt{\frac{1-\widetilde{\mathcal{F}}_{j-i}}{N_j \widetilde{\mathcal{F}}_{j-i}}} \tag{3.60}$$

On the basis of equations (3.59) and (3.60), the ratio of the two standard deviations can be computed:

$$\frac{\sigma_{\widetilde{\eta}_{i-j}}}{\sigma_{\widetilde{\eta}_{j-i}}} = \sqrt{\frac{N_i}{N_j}\frac{\left(1-\frac{\Omega_i}{\Omega_j}\widetilde{\mathcal{F}}_{i-j}\right)}{\left(\frac{\Omega_i}{\Omega_j}\right)\left(1-\widetilde{\mathcal{F}}_{i-j}\right)}} \tag{3.61}$$

This last expression is a function of $\widetilde{\mathcal{F}}_{i-j}$ which increases monotonically with $\widetilde{\mathcal{F}}_{i-j}$. Its lowest value is given by the following expression:

$$\lim_{\widetilde{F}_{i-j}\to 0} \frac{\sigma_{\widetilde{\eta}_{j-i}}}{\sigma_{\widetilde{\eta}_{i-j}}} = \sqrt{\frac{N_i}{N_j}\frac{\Omega_j}{\Omega_i}} \tag{3.62}$$

The ratio of the standard deviation is roughly proportional to the ratio of the Ω's and inversely proportional to the ratio of the numbers of traced rays. In a pair of exchange factors, the smallest error is observed for the surface characterized by the smallest ratio $\frac{\Omega}{N}$, that is for the surface with small Ω and large N.

3.3.5 Enforcement of the reciprocity - State of the art

In this Section, different techniques used to enforce the reciprocity of the exchange factors computed by ray tracing are described.

The ray tracing process requires the computation of all the exchange factors throughout the geometrical model, yielding a complete set of estimators η_{i-j}. This implies redundant computation. Nevertheless, this redundant data can be used in order to reduce the error which affects the factors of a pair of patches, by enforcing the reciprocity. This results in a set of corrected estimators $\dot{\eta}_{i-j}$. Then, a method of least-squares smoothing will be used to enforce the closure (see Section 3.3.8.1), providing a final set of estimators $\ddot{\eta}_{i-j}$.

3.3.5.1 Van Leersum enforcement

A naive enforcement of reciprocity consists in retaining the view factors above the main diagonal and discarding the others [TL95]. The other view factors excluding the diagonal are obtained by using the reciprocity relation. The diagonal terms are computed using the closure relation. The drawback of this simple method is that it does not insure non-negative exchange factors. An iterative scheme has been designed to insure non-negative exchange factors [vL89].

3.3.5.2 Esarad triangulation

The method implemented in Esarad is called matrix triangulation [Doc04] and is based on an empirical formula demonstrated in reference [DP93]. This method is also used in Thermica [Doc03]. On the basis of the two values η_{i-j} and η_{j-i}, an estimator $\dot{\eta}_{i-j}$ is built by linear averaging:

$$\dot{\eta}_{i-j} = \kappa\eta_{i-j} + (1-\kappa)\eta_{j-i} \tag{3.63}$$

The weight κ must be positive and lower or equal to unity. The weight is computed by the following relationship. The demonstration of this expression can be found in [DP93].

$$\kappa = \frac{1}{2}\left(1 + \operatorname{sign}(Y)\,|Y|\,Y^n\right) \tag{3.64}$$

where

$$Y = \frac{\frac{\Omega_j}{N_j} - \frac{\Omega_i}{N_i}}{\frac{\Omega_j}{N_j} + \frac{\Omega_i}{N_i}} \tag{3.65}$$

$$n = 0.4 \tag{3.66}$$

The value 0.4 of the exponent n has empirically been found to be the best one.

3.3.5.3 Fractional confidence interval

This method is based on the standard deviations which characterize the two exchange factors to be enforced. As the error of an exchange factor is linked to its standard deviation σ, a new weight κ based on σ can be defined. The following equation is used:

$$\kappa = \frac{\sigma_{\eta_{j-i}}}{\sigma_{\eta_{i-j}} + \sigma_{\eta_{j-i}}} \tag{3.67}$$

If the exchange factor η_{j-i} is characterized by a large uncertainty, the associated standard deviation $\sigma_{\eta_{j-i}}$ is important. The weight κ is close to unity and the estimator $\dot{\eta}_{i-j} = \dot{\eta}_{j-i}$ is close to the factor η_{i-j}, which is the most reliable value. The weight κ takes into account the relative values of the Ω's, the exchange factors $\mathcal{F}$ and the numbers of rays N. If the weight and the Ω's are constant, the variance of the new estimator is given by the following relation:

$$\sigma^2_{\dot{\eta}_{i-j}} = \kappa^2\Omega_i^2\frac{\mathcal{F}_{i-j}(1-\mathcal{F}_{i-j})}{N_i} + (1-\kappa)^2\Omega_j^2\frac{\mathcal{F}_{j-i}(1-\mathcal{F}_{j-i})}{N_j} \tag{3.68}$$

3.3.5.4 Fractional variance

The method of the fractional variance, introduced in [LDB95], is similar to the fractional confidence interval. The weight κ is built on the basis of the variances of the two exchange factors:

$$\kappa = \frac{\sigma^2_{\eta_{j-i}}}{\sigma^2_{\eta_{i-j}} + \sigma^2_{\eta_{j-i}}} \tag{3.69}$$

The variance associated with the new estimator is given by the relation (3.68), if the weight κ and the Ω's are constant.

3.3.5.5 Maximum likelihood estimation

In the maximum likelihood estimation method presented in [DMH05], several probability distributions parameterized by an unknown parameter θ are considered; the probability distributions are associated with a known probability mass function, noted $p(x|\theta)$. The probability of all independent observations occurring together is given by the likelihood function $\mathcal{L}(\theta)$:

$$\mathcal{L}(\theta) = \prod_{k=1}^{n} p(x_k|\theta) \tag{3.70}$$

The maximum likelihood estimation consists in finding the value of θ that maximizes $\mathcal{L}(\theta)$. This value will be the most probable value of θ. This value of θ fulfills the following statements:

$$\begin{cases} \dfrac{\partial\mathcal{L}(\theta)}{\partial\theta} &= 0 \\ \dfrac{\partial^2\mathcal{L}(\theta)}{\partial\theta^2} &< 0 \end{cases} \tag{3.71}$$

Since the likelihood function is strictly positive, the logarithm of equation (3.70) can be computed to turn the product into a sum. Finding the value of θ that maximizes $\mathcal{L}(\theta)$ is equivalent to finding the value which maximizes $\ln(\mathcal{L}(\theta))$:

$$\frac{1}{\mathcal{L}}\frac{\partial\mathcal{L}(\theta)}{\partial\theta} = \frac{\partial\ln(\mathcal{L}(\theta))}{\partial\theta} \tag{3.72}$$

In ray tracing, each ray can be treated as a Bernoulli variable with a binary outcome; the ray is received by the patch P_j, or it is not. The exact exchange factor can be considered as the probability that a ray emitted from the patch P_i will be received by P_j. A set of random binary variables X_{i-j}^k are considered. Once the k^{th} experiment is performed, a numerical value x_{i-j}^k is assigned to X_{i-j}^k, where $x_{i-j}^k = 1$ if the k^{th} ray is received by P_j, $x_{i-j}^k = 0$ otherwise. The probability of getting exactly N_{i-j} rays received by P_j is given by the probability density function:

$$p(N_{i-j}, N_i, \mathcal{F}_{i-j}) = C_{N_i}^{N_{i-j}} \mathcal{F}_{i-j}^{N_{i-j}} \left(1 - \mathcal{F}_{i-j}\right)^{N_i - N_{i-j}} \tag{3.73}$$

If two exchange factors $\mathcal{F}_{i-j}$ and $\mathcal{F}_{j-i}$ are considered, the common parameter θ can be chosen as the estimator $\dot{\eta}_{i-j}$ in order to obtain a binomial maximum likelihood estimation. If the reciprocity relation is used, the exchange factors can be replaced by the following expressions:

$$\begin{cases} \mathcal{F}_{i-j} &= \dfrac{\dot{\eta}_{i-j}}{\Omega_i} \\ \mathcal{F}_{j-i} &= \dfrac{\dot{\eta}_{i-j}}{\Omega_j} \end{cases} \tag{3.74}$$

The expression of the corresponding likelihood function is obtained using equation (3.73):

$$\begin{aligned} \mathcal{L}(\theta) &= p(N_{i-j}, N_i, \mathcal{F}_{i-j}) p(N_{j-i}, N_j, \mathcal{F}_{j-i}) \\ &= \frac{N_i! N_j!}{N_{i-j}!(N_i - N_{i-j})! N_{j-i}!(N_j - N_{j-i})!} \left(\frac{\dot{\eta}_{i-j}}{\Omega_i}\right) \left(1 - \frac{\dot{\eta}_{i-j}}{\Omega_i}\right) \left(\frac{\dot{\eta}_{i-j}}{\Omega_j}\right) \left(1 - \frac{\dot{\eta}_{i-j}}{\Omega_j}\right) \end{aligned} \tag{3.75}$$

This expression can then be derived with respect to $\dot{\eta}_{i-j}$ to compute the most probable value of $\dot{\eta}_{i-j}$.

This method can only be applied to binomial distributions. The exchange factors must be the ratio of two natural numbers. This method cannot be directly used with the stratified hemisphere method, where the Gauss quadrature is applied in order to integrate the exchange factors onto the emitter's surface. Nevertheless, the maximum likelihood can be used in order to demonstrate the adequate way to enforce the reciprocity of the exchange factors obtained with the stratified hemisphere.

3.3.6 Enforcement of the reciprocity - Application to the hemisphere method

In the literature, the exchange factors are computed by generating a large number of rays from each surface. These rays are propagated throughout the geometrical model until being received by a surface. In order to stop the ray, a common method is based on the Russian roulette. When it encounters a surface, a ray is completely absorbed or completely reflected. The computed exchange factors are then based on binomial distributions, which simplify the enforcement of the reciprocity. The obtained exchange factors are the ratios

of two integer numbers.

In the Section dedicated to the stratified hemisphere method, we pointed out the fact that it can only be used to compute point wise view factors. The integration onto the emitter's surface must be performed with the help of the Gauss quadrature. The resulting view factors are reals. This increases the difficulty of the statistical theory that must be developed to enforce the reciprocity. The exchange factor is given by the following relation:

$$\mathcal{F}_{i-j} = \sum_{k=1}^{NG} W_k \mathcal{F}_{i-j}^{(k)} \tag{3.76}$$

where the sum is performed on the NG Gauss points of the patch P_i. W_k is the weight of the Gauss point k. $\mathcal{F}_{i-j}^{(k)}$ is the exchange factor computed at the point k.

The main difference with the classical Monte Carlo ray tracing is that the exchange factor is no longer described by a binomial distribution. It is now a combination of binomial distributions. The variance associated with $\mathcal{F}_{i-j}$ is equal to:

$$\sigma^2_{\mathcal{F}_{i-j}} = \sigma^2 \left(\sum_{k=1}^{NG} W_k F_{i-j}^{(k)} \right) \tag{3.77}$$

$$= \sum_{k=1}^{NG} W_k^2 \sigma^2_{\mathcal{F}_{i-j}^{(k)}} \tag{3.78}$$

Because the statistics involved with the stratified hemisphere method are more complex, the maximum likelihood estimation method cannot be directly applied to the stratified hemisphere. Nevertheless, it can be used to demonstrate the validity of the fractional variance applied to the stratified hemisphere.

3.3.7 New formulations

In Section 3.3.5, the state of the art in the domain of reciprocity enforcement has been presented. The presented formulations are based on a linear combination of the two exchange factors involved in each pair. These formulations differ in the definition of the factor κ. In this Section, new formulations based on different definitions of κ are introduced.

3.3.7.1 Maximum likelihood estimation

The exchange factors obtained with the stratified hemisphere method do not follow a binomial distribution. If the number of traced rays is large enough, the Central-Limit theorem [Gra78] can be used to establish that the exchange factors follow a normal distribution. This is illustrated in Appendix D.1. The mean and variance of the normal

distribution followed by a exchange factor $\mathcal{F}_{i-j}$ are given by the following relations:

$$\begin{cases} E\left(\mathcal{F}_{i-j}\right) &= \tilde{\mathcal{F}}_{i-j} \\ \sigma^2\left(\mathcal{F}_{i-j}\right) &= \sum_{k=1}^{NG} W_k^2 \sigma^2_{\mathcal{F}^{(k)}_{i-j}} \end{cases} \tag{3.79}$$

Now two distributions of reciprocal exchange factors, noted $\mathcal{F}_{i-j}$ and $\mathcal{F}_{j-i}$, are considered. For each distribution, the mean and variance can be defined by means of equations (3.79). The probability density function of η_{i-j} is given by the following relation:

$$f\left(x|\eta_{i-j},\sigma_{\eta_{i-j}}\right) = \frac{1}{\sqrt{2\pi}\sigma_{\eta_{i-j}}} \exp\left[-\frac{1}{2}\left(\frac{x-\eta_{i-j}}{\sigma_{\eta_{i-j}}}\right)^2\right] \tag{3.80}$$

The mean and variance of this distribution can be obtained from the mean and variance of $\mathcal{F}_{i-j}$:

$$\begin{cases} E(\eta_{i-j}) &= \Omega_i \mathcal{F}_{i-j} \\ \sigma^2_{\eta_{i-j}} &= \Omega_i^2 \sigma^2_{\mathcal{F}_{i-j}} \end{cases} \tag{3.81}$$

In the same way, the probability density of the reciprocal exchange factor η_{j-i} can be established:

$$f\left(x|\eta_{j-i},\sigma_{\eta_{j-i}}\right) = \frac{1}{\sqrt{2\pi}\sigma_{\eta_{j-i}}} \exp\left[-\frac{1}{2}\left(\frac{x-\eta_{j-i}}{\sigma_{\eta_{j-i}}}\right)^2\right] \tag{3.82}$$

Then, the probability function $\mathcal{L}(x)$ is built as the product of the two previous probability density functions:

$$\mathcal{L}(x) = f\left(x|\eta_{i-j},\sigma_{\eta_{i-j}}\right) f\left(x|\eta_{j-i},\sigma_{\eta_{j-i}}\right) \tag{3.83}$$

$$= \frac{1}{2\pi\sigma_{\eta_{i-j}}\sigma_{\eta_{j-i}}} \exp\left[-\frac{1}{2}\left(\frac{x-\eta_{i-j}}{\sigma_{\eta_{i-j}}}\right)^2 - \frac{1}{2}\left(\frac{x-\eta_{j-i}}{\sigma_{\eta_{j-i}}}\right)^2\right] \tag{3.84}$$

The first derivative of the logarithm of this function with respect to x, equated to zero, yields the estimator x of η_{i-j} which satisfies the reciprocity.

$$\frac{\partial \ln(\mathcal{L})}{\partial x} = \frac{\partial}{\partial x}\left[\ln\left(\frac{1}{2\pi\sigma_{\eta_{i-j}}\sigma_{\eta_{j-i}}}\right) - \frac{(x-\eta_{i-j})^2}{2\sigma^2_{\eta_{i-j}}} - \frac{(x-\eta_{j-i})^2}{2\sigma^2_{\eta_{j-i}}}\right] \tag{3.85}$$

$$= \frac{\eta_{i-j}-x}{\sigma^2_{\eta_{i-j}}} + \frac{\eta_{j-i}-x}{\sigma^2_{\eta_{j-i}}} = 0 \tag{3.86}$$

$$\Leftrightarrow x = \frac{\eta_{i-j}\sigma^2_{\eta_{j-i}} + \eta_{j-i}\sigma^2_{\eta_{i-j}}}{\sigma^2_{\eta_{i-j}} + \sigma^2_{\eta_{j-i}}} \tag{3.87}$$

$$= \eta_{i-j}\frac{\sigma^2_{\eta_{j-i}}}{\sigma^2_{\eta_{i-j}} + \sigma^2_{\eta_{j-i}}} + \eta_{j-i}\frac{\sigma^2_{\eta_{i-j}}}{\sigma^2_{\eta_{i-j}} + \sigma^2_{\eta_{j-i}}} \tag{3.88}$$

This expression is similar to the fractional variance, except that the variances in equation (3.88) correspond to the variances of the stratified hemisphere method.

3.3.7.2 Fractional confidence interval - fractional variance

In the previous Section, we presented the fractional confidence interval and fractional variance methods, in the case of a binomial distribution. In Section 3.3.6, we established that the variance and standard deviation obtained with the stratified hemisphere method are more complex than the expressions derived in the case of a binomial distribution. However, we decide to approximate the hemisphere distribution by a binomial law, in order to use the binomial expression of the variance.

$$\sigma^2_{\mathcal{F}_{i-j}} = \frac{\mathcal{F}_{i-j}\left(1-\mathcal{F}_{i-j}\right)}{N_i}$$

This introduces an error in the variance. Indeed, firing N_i rays from each gaussian point of a surface i can be seen as distributing each of the N_i rays among the gaussian points. Then a fraction W_k of the energy of each ray is attributed to the k^{th} gaussian node.

The ray-tracing process is no longer ruled by a binomial distribution. However, this approximation yields a simple expression to evaluate the variance. This approximation can be acceptable in practice. This variance is then used in the expression of the fractional confidence interval and fractional variance methods.

3.3.7.3 Personal variation (PV)

During the tests, a variation of the fractional variance method has been tried, where the κ factor is based on the square of the variance. Although this formulation is not supported by any theoretical background, contrary to the maximum likelihood estimation, it yields the best results in terms of reduction of the RMS error.

3.3.8 Enforcement of the closure

In this Section, we consider a set of exchange factors $\dot{\eta}_{i-j}$ which obey to the reciprocity rule but not to the closure rule. The most efficient method is based on least-squares smoothing. The resulting set of exchange factors is noted $\ddot{\eta}_{i-j}$.

3.3.8.1 Least-squares smoothing

Once the reciprocity has been enforced, the closure is generally no longer valid. The closure has to be enforced in order to insure the conservation of energy. Here, a formulation based on the least-squares smoothing method [LH86, LCL01] is presented. This method has been designed to correct a system which verifies the reciprocity but not the closure. The authors supposed that the reciprocity has been previously enforced to reduce the storage of the exchange factors.

The least-squares smoothing method requires the definition of an objective function $\mathcal{H}$, which will be minimized. This function must contain the variations of the $\dot{\eta}$-values. The

aim is to obtain exchange factors which verify the closure, while limiting the variations of the initial exchange factors.

$$\mathcal{H} = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\left(\dot{\eta}_{i-j} - \ddot{\eta}_{i-j}\right)^2}{2\omega_{i-j}} \tag{3.89}$$

where $\dot{\eta}$ corresponds to the value obtained after enforcement of the reciprocity, $\ddot{\eta}$ represents the value obtained by applying the closure law. Due to the symmetry of the $\dot{\eta}$, the weight matrix ω must also be symmetrical. The weights allow us to assign penalties to certain factors.

The reciprocity constraints must be included in the lagrangien $\mathcal{L}$, which must not be confused with the likelihood function. The lagrangien is defined by the following expression:

$$\mathcal{L} = \mathcal{H} + \sum_{i=1}^{N} \lambda_i \left(g_i + g_i^*\right) \tag{3.90}$$

where λ_i are the Lagrange multipliers and g_i and g_i^* are, respectively, the row- and column-sum constraints:

$$\left\{ \begin{array}{rcl} g_i & = & \Omega_i - \sum_{j=1}^{N} \ddot{\eta}_{i-j} \\ g_i^* & = & \Omega_i - \sum_{j=1}^{N} \ddot{\eta}_{j-i} \end{array} \right. \tag{3.91}$$

The previous relations consist in $2N$ constraints. In fact, only N constraints are independent. The column-sum constraints can be obtained from the line-sum constraints by using the reciprocity relations. The introduction of these two sets of constraints is necessary to insure the conservation of the reciprocity during the enforcement of closure. Differentiating $\mathcal{L}$ with respect to $\ddot{\eta}$ gives:

$$\frac{\partial \mathcal{L}}{\partial \ddot{\eta}_{i-j}} = -\frac{\dot{\eta}_{i-j} - \ddot{\eta}_{i-j}}{\omega_{i-j}} - \lambda_i - \lambda_j \tag{3.92}$$

The new estimator is given by equating the previous relation to zero:

$$\ddot{\eta}_{i-j} = \dot{\eta}_{i-j} + \omega_{i-j} \left(\lambda_i + \lambda_j\right) \tag{3.93}$$

Expression (3.93) is introduced in the closure constraint (3.91) to obtain the values of the Lagrange multipliers:

$$\Omega_i - \sum_{j=1}^{N} \dot{\eta}_{i-j} = \lambda_i \sum_{j=1}^{N} \omega_{i-j} + \sum_{j=1}^{N} \lambda_j \omega_{i-j} \tag{3.94}$$

This is a linear system $Ax = b$ which must be solved to determine the Lagrange multipliers. The components can be written as:

$$A_{i-j} = \omega_{i-j} \quad \text{if} \quad i \neq j \tag{3.95}$$

$$A_{i-i} = \omega_{i-i} + \sum_{j=1}^{N} w_{i-j} \tag{3.96}$$

$$b_j = \Omega_j - \sum_{i=1}^{N} \dot{\eta}_{j-i} \tag{3.97}$$

Once the Lagrange multipliers are computed, they are introduced in the equation (3.93) and we finally obtain the estimators which obey the closure rule minimize the function $\mathcal{H}$.

This method is really efficient when the geometrical model is totally closed. If the model presents an aperture, this method must be adapted. It is performed in Section 3.3.10.

Moreover, the weights ω_{i-j} have not be defined. The only constraint is that they must be symmetrical to conserve the reciprocity of the exchange factors. In Section 3.3.9, the results obtained for different choices of weights are presented. We consider three weights based on the variance (σ_{i-j}, σ^2_{i-j} and σ^4_{i-j}) and three weights based on the exchange factor itself ($\sqrt{\eta_{i-j}}$, η_{i-j} and η^2_{i-j}). These different closure enforcements are combined with different reciprocity enforcements in the case of a closed model.

3.3.9 Test results

To identify the best method, different reciprocity enforcement have been implemented. These methods have been combined with different formulations of the least-squares smoothing. Only closed models have been considered here. The extension to open models will be performed in a following Section.

3.3.9.1 Applied methods

For the reciprocity enforcement, fractional confidence interval (FCI), fractional variance (FV), maximum likelihood estimation (MLE), a personal variation (PV) and the Esarad method have been implemented.

For the closure enforcement, several formulations of the least-squares smoothing are tested, with different definitions of the weights ω_{i-j}. Weights based on different powers of the variance or of the estimator itself are considered.

3.3.9.2 Test cases

We considered a set of simple geometrical configurations for which an analytical solution can be found in the literature. References [How82, Won76] have been used to compare the

computed values with an analytical result. The present work considers only view factors. Figure 3.33 represents the geometrical configuration of the test case 4.

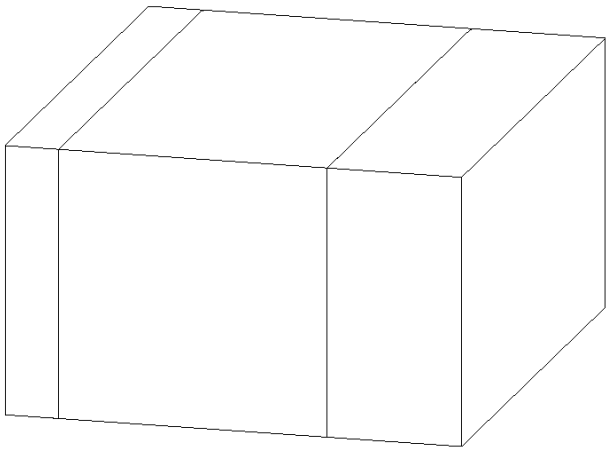


Figure 3.33: Test case 4 - Geometrical configuration.

3.3.9.3 Definition of the error

In order to compare the quality of the different tested methods, the RMS error associated with each matrix of view factors has been chosen. The RMS error is defined by the following relation:

$$\epsilon = \sqrt{\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(F_{i-j,\text{exact}} - F_{i-j,\text{computed}}\right)^2} \tag{3.98}$$

For each test case, 1 000 successive ray tracings have been considered in order to establish statistics. The computed view factor which appears in the formula (3.98) is in fact the average view factor computed from the set of 1 000 results.

3.3.9.4 Results

Table 3.3 lists the RMS error obtained when the different methods of reciprocity enforcement are combined with the different definitions of the weight ω_{i-j} used in the least-squares smoothing. The initial RMS error (concerning the exchange factors obtained by ray-tracing, without any post-treatment) is equal to $2.0350\,10^{-3}$.

The first line lists the RMS error observed when only reciprocity is enforced. This is a good indicator of the efficiency of the different methods. For the FCI, the RMS error increases, when compared to the initial error. For this reason, this method will not be used in the proposed algorithms.

	FCI	FV	MLE	PV	Esarad
none	$2.3479\,10^{-3}$	$1.6756\,10^{-3}$	$1.7265\,10^{-3}$	$1.5254\,10^{-3}$	$1.9493\,10^{-3}$
$\sqrt{\eta}$	$2.0907\,10^{-3}$	$1.5121\,10^{-3}$	$1.5599\,10^{-3}$	$1.3771\,10^{-3}$	$1.8388\,10^{-3}$
η	$1.7947\,10^{-3}$	$1.4554\,10^{-3}$	$1.4769\,10^{-3}$	$1.3802\,10^{-3}$	$1.6569\,10^{-3}$
η^2	$1.7012\,10^{-3}$	$1.6234\,10^{-3}$	$1.6352\,10^{-3}$	$1.4773\,10^{-3}$	$1.3652\,10^{-3}$
σ	$2.0784\,10^{-3}$	$1.5189\,10^{-3}$	$1.5692\,10^{-3}$	$1.4076\,10^{-3}$	$1.8314\,10^{-3}$
σ^2	$1.8984\,10^{-3}$	$1.5236\,10^{-3}$	$1.5636\,10^{-3}$	$1.4287\,10^{-3}$	$1.7019\,10^{-3}$
σ^4	$1.8432\,10^{-3}$	$1.6560\,10^{-3}$	$1.6900\,10^{-3}$	$1.4954\,10^{-3}$	$1.4864\,10^{-3}$

Table 3.3: Comparison of the error norms for different methods.

Another interesting result is that no large difference is observed when the exact variance of the stratified hemisphere method (MLE) is replaced by the variance of the binomial distribution (FV).

The best method seems to be the PV method, which leads to the lowest levels of the RMS error.

The method implemented in Esarad and Thermica yields improvements of the RMS error. However, these improvements are lower than the other methods, except the FCI. Figure 3.34 plots the evolution of the RMS error obtained with the Esarad triangulation, in function of the exponent n. The associated standard deviation is represented by the height of the corresponding lines. The green curve corresponds to the initial RMS error, when no enforcement is applied. It is constant, equal to $2.0350\,10^{-3}$. If the Esarad triangulation is used to enforce reciprocity, the obtained error is given by the blue curve. The minimum occurs for n=0.05. If the σ^4 are chosen as the weights of the least-squares smoothing, the red curve is obtained. It yields a minimum of the RMS error for a non-physical value of the exponent n (which can only be defined between 0 and 1). If the value 0.4 is chosen for he exponent, it appears that the Esarad triangulation reduces the RMS error; the enforcement of closure induces a slight reduction of the remaining error.

3.3.10 Open models

In the previous Sections, only closed models have been studied, that is models which do not present any aperture. In space applications, this situation can be encountered when the interior of a spacecraft is considered as an independent cavity, where the classical least-squares smoothing can be applied without any restriction. If the exterior of the spacecraft is considered, the situation is quite more complex because of the presence of an additional surface, which is necessary to model the exchanges with the deep space.

When an open model is studied, the main difficulty lies in the fact that the ray tracing only computes the exchange factors from the N surfaces of the model to the whole model

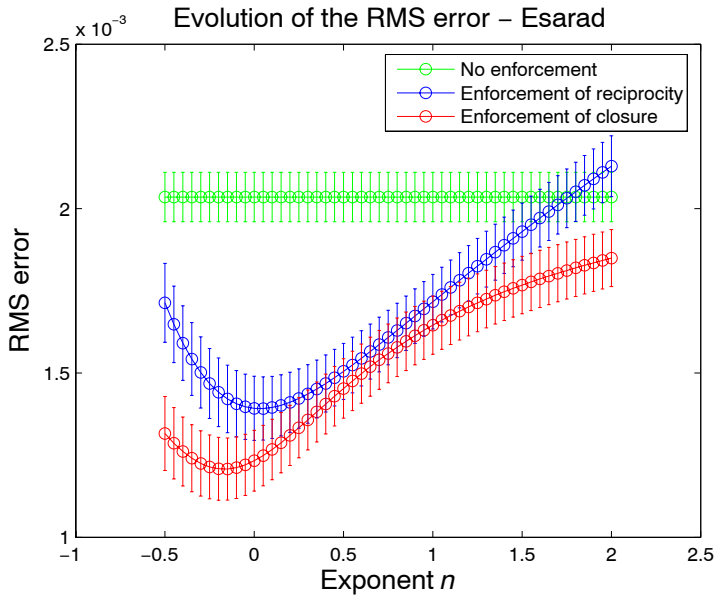


Figure 3.34: Esarad method - Evolution of the RMS error in function of the exponent n.

and to an additional node representing the deep space. So the obtained matrix contains N lines and $N+1$ columns. When the reciprocity is enforced, the N exchange factors to the deep space cannot be corrected if the exchange factors from the deep space to the model have not been computed. If the exchange factors matrix is not square, all the terms cannot be corrected, while conserving the reciprocity.

3.3.10.1 Exact exchange factors to the deep space

A first solution consists in considering that the exchange factors from the model to the deep space are exact and do not require any modification. The line- and column-constraints g_i and g_i^* are completed by adding to each term the corresponding exchange factor to the deep space. The corresponding extension of the least-squares method is presented in Appendix D.2.

Because the exchange factors to the deep space can really be affected by a large error, the enforcement of closure based on such an hypothesis can lead to an increase of the associated RMS error. This has been observed during the tests, as it is mentioned in [Vue08]. For this reason, this method is not recommended when the geometrical model presents an aperture.

3.3.10.2 Correction of the exchange factors to the deep space

Another method consists in relaxing the column-constraints g_i^*. The final exchange factors will no longer obey the reciprocity relation. The advantage of this method is to allow corrections of the "model-to-deep-space" exchange factors. The corresponding extension

of the least-squares smoothing is presented in Appendix D.3.

Because the column-constraints are not imposed anymore, the reciprocity of the exchange factors is lost. Compared to the first method, this one does not reduce the accuracy of the final exchange factors. But the efficiency of this method is limited as the reciprocity relation applied to the exchange factors to the deep space cannot be insured.

During the tests, the performances of this method have been observed to be limited. From the reciprocity enforcement to the closure enforcement, the RMS error stays more or less constant. Moreover, the reciprocity is lost. A second enforcement of reciprocity can be necessary but this does not reduce the RMS error. For these reasons, the interest of enforcing closure when an aperture is detected in the geometrical model is not obvious.

3.3.10.3 Deep-space-to-model exchange factors

The third coded method is more complex than the previous ones because it requires huge modifications of the ray tracing algorithm. To obtain a square matrix of exchange factors, an additional surface must be added to model the deep space.

A sphere centered on the centroid of the model is built; it is the smallest sphere containing the whole model. The sphere is decomposed into cells of equal area following the longitude and latitude directions. This is done in order to obtain the statistical optimization due to stratified sampling [DBB03]. For each couple of cells, a random point is uniformly generated into each cell. A ray is traced between the two random points. If the ray is intercepted by a surface, the exchange factor from the deep space to that surface receives a contribution equal to $1/n_{\text{lon}}^2 n_{\text{lat}}^2$, where n_{lon} and n_{lat} designate respectively the longitude- and latitude-resolutions.

Each cell is associated twice with another cell; first as the emitter, then as the receiver.

This yields a square exchange factors matrix. The least-squares smoothing can be directly used. The drawback of this method is that it requires a lot of additional rays to be traced in order to obtain a sufficient accuracy for the "deep-space-to-model" exchange factors, to observe a reasonable improvement of the accuracy of the exchange factors.

As a conclusion of this Section, in the case of an open model, it is our strong belief that an enforcement of the closure will not improve the quality of the exchange factors without an unacceptable increase of the computation resource. For this reason, no enforcement of closure has been implemented in case of an open model. The enforcement of reciprocity only allows us to reduce significantly the RMS error associated with the exchange factors.

3.3.11 Conclusions

On the basis of the results, we have established the enforcement strategy that will be implemented in our algorithms.

Concerning the enforcement of reciprocity, it is used for each geometrical configuration, *i.e.* for closed and open models. The selected method is the fractional variance (FV). The personal variation (PV) remains a good competitor but the lack of theoretical material is a serious drawback. The Esarad method, the fractional confidence interval and the maximum likelihood method are not retained for further implementation.

Concerning the enforcement of the closure, the estimator itself is selected as the weight ω_{i-j} used in the least-squares smoothing when the model is closed. When the model presents an aperture, closure is not enforced.

In order to conclude the Chapter dedicated to exchange factors, we present in the next Section the extension of Gebhart's matrix method, used to derive the radiative exchange factors from the view factors.

3.4 Gebhart's method

3.4.1 Introduction

To apply Gebhart's formulation of radiative heat transfer, we need to use the radiative exchange factors (REF) between all the surfaces which compose the 3D geometrical model, as well as the absorbed heat fluxes (AHF), *i*.e. the heat fluxes absorbed by each surface of the model, either directly or via multi reflections.

REF and AHF can either be obtained by Monte Carlo ray tracing (MCRT) or by Gebhart's method, also called the matrix method [Doc06, Doc03].

The so-called matrix method has been designed by Benjamin Gebhart in the sixties in order to derive the radiative exchange factors from the geometrical, diffuse view factors [Geb61b, Geb59]. A second formulation of this matrix method can be used in order to derive the absorbed heat fluxes from the direct heat fluxes (DHF) [Doc04].

The main advantage of Gebhart's method is that it requires only one computation run in order to extract the geometrical information from the 3D model, in the form of a set of view factors. These view factors are then used to compute the REF and the AHF.

The computation of the view factors and direct heat fluxes can be performed by a Monte Carlo ray tracing. Radiative exchange factors and absorbed heat fluxes can either be obtained by MCRT or by Gebhart's method. It will be shown that using MCRT to

compute the view factors converges faster than using it to compute the REF.

Several hypotheses limit the use of Gebhart's method. Each hypothesis will be presented and discussed. Some of these limitations will be canceled but the last one remains necessary. The distinction between the two formulations of Gebhart's method and the last hypothesis will have a determinant impact on the way we propose to use Gebhart's method.

3.4.2 Gebhart's formulation of heat transfer

The formulation developed by Gebhart [Geb61a] can be used to solve a complete thermal problem. The energy balance is based on the self emitted power of the surfaces which compose the model. The temperatures can be obtained by solving the following set of equations:

$$Q_i = \epsilon_i A_i E_{b,i} - \sum_{j=1}^{N} \epsilon_j A_j B_{j-i} E_{b,j} \tag{3.99}$$

Compared to the classical formulation which is based on view factors, Gebhart's formulation presents the advantage that the size of the system to solve is reduced. Let us suppose that the n_1 first surfaces are characterized by a fixed heat flux and that the temperature of the remaining n_2 surfaces is known (with $n_1 + n_2 = N$). Only the n_1 surfaces of unknown temperatures have to be considered. Once the temperatures have been computed by solving equation (3.99) for i between 1 and n_1, the unknown heat fluxes are computed using equation (3.99) for i between $n_1 + 1$ and N.

The drawback of Gebhart's formulation is that the matrix of the radiative links is full. Each surface of a given enclosure interacts with all the other surfaces of this enclosure. A full matrix is not suitable for the resolution of the thermal system.

3.4.3 Equations

In this Section, the equations relative to the two formulations of Gebhart's method are presented. The first formulation is relative to the radiative exchange factors while the second one concerns the radiative heat fluxes.

3.4.3.1 Radiative exchange factors

In [Geb61a], the author presented a way to compute the REF, that he called *absorption factors*, by inverting a matrix based on the view factors. In the original version, the author only considered diffuse reflections. No specular effect was taken into account.

The radiative exchange factor B_{i-j} can be considered as the sum of two terms. The first one corresponds to the fraction of energy which is emitted by i and is directly absorbed by j. The second term is a sum on all the possible intermediary surfaces. Each term k of

this sum is the fraction of energy emitted by i which is diffusely reflected by k and which is finally absorbed by j. It yields the following expression:

$$B_{i-j} = \epsilon_j F_{i-j} + \sum_{k=1}^{N} F_{i-k}\rho_k B_{k-j} \tag{3.100}$$

Radiative exchange factors are a function of the diffuse view factors, the emissivity ϵ_j of the receptor and the diffuse reflectivity ρ_k of all the intermediary surfaces k.

Remark: The author solved this expression by the determinant method. In the algorithms presented in this thesis, an iterative scheme has been preferred. The emission term $\epsilon_j F_{i-j}$ is chosen as the initial value of each factor B_{i-j}. Then the operator $\mathcal{R} = \rho F$ is applied. Each iteration can then be considered as the result of one reflection step.

3.4.3.2 Absorbed heat fluxes

In Gebhart's formulation, the irradiation term corresponds to the external heat fluxes which are absorbed by each surface of the model. This implies that the multi-reflection process must be taken into account to compute this term. Just as the REF are the combination of two distinct contributions, so will be the absorbed heat fluxes. First, the incident heat fluxes must be computed. If $H_{0,i}$ denotes the incident heat flux received by a patch P_i, the absorbed power is equal to $A_i\epsilon_i H_{0,i}$. The second term is the power which is diffusely reflected by each surface j and finally absorbed by i. The following relation [Doc04] is obtained:

$$Q_i = A_i\epsilon_i H_{0,i} + \sum_{j=1}^{N} B_{j-i}A_j\rho_j H_{0,j} \tag{3.101}$$

3.4.4 Sequence of the computations

In order to obtain the quantities required by Gebhart's formulation, several combinations of computations are possible.

3.4.4.1 Radiative exchange factors

The different possibilities to obtain the radiative exchange factors are shown in Figure 3.35. The radiative exchange factors can be directly computed by a Monte Carlo ray tracing including the diffuse component of reflection. They can also be computed with Gebhart's method, from the view factors (which are computed by a ray tracing process) and from the thermo-optical properties of the radiative model.

3.4.4.2 Absorbed heat fluxes

The situation is more complex for the absorbed heat fluxes, because their computation can involve a two-step process, if Gebhart's method is applied. In Figure 3.36, the different

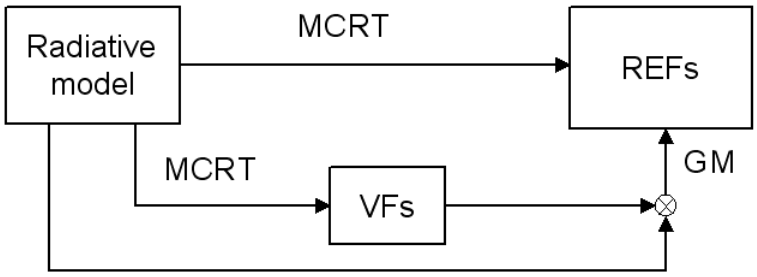


Figure 3.35: Gebhart's Method - Radiative exchange factors.

possibilities are given for the absorbed solar heat fluxes (ASHF) case. The first step concerns the solar radiative exchange factors. They can either be computed by a complete Monte Carlo ray tracing process or by Gebhart's method. The direct solar heat fluxes (DSHF) are computed by ray tracing. They can be used with the radiative exchange factors in the framework of the second formulation of Gebhart's method to obtain the absorbed solar heat fluxes. The absorbed solar heat fluxes can also be directly computed by a complete Monte Carlo ray tracing.

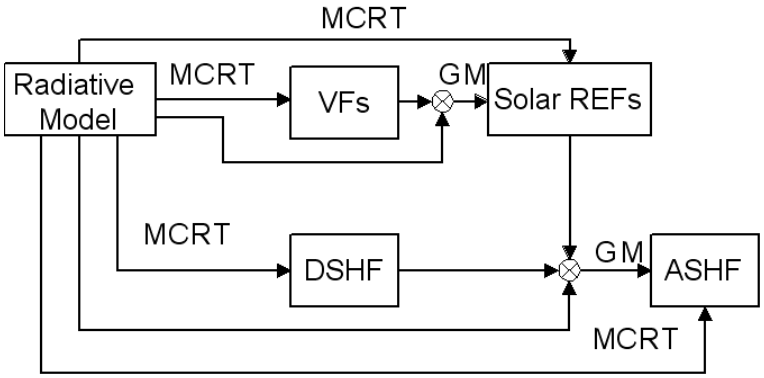


Figure 3.36: Gebhart's Method - Absorbed solar heat fluxes.

3.4.5 Advantages of Gebhart's method

In this Section, two ways of computing the radiative exchange factors are compared: the classical Monte Carlo ray tracing and Gebhart's method. Gebhart's method for heat fluxes is not considered in this Section.

3.4.5.1 Convergence of the ray tracing processes

If the REF are directly computed by ray tracing, a ray intercepted by a surface k has a probability equal to ϵ_k of being absorbed. A secondary ray will be re-emitted with a probability equal to $1-\epsilon_k = \rho_k$. If the model is characterized by a low average emissivity, there

will be a large number of secondary rays to be traced through the model. This implies a slow convergence of the ray tracing.

If the view factors are computed by ray tracing and if Gebhart's method is used in order to derive the radiative exchange factors, no secondary ray will be re-emitted during the ray tracing (if we consider only diffuse reflectors). This is why Gebhart's method is known to be a faster way to obtain the radiative exchange factors.

Remark: applying equation (3.100) can be very fast, if it is properly implemented. So the CPU time relative to this step can be considered negligible in comparison with the ray tracing process. The convergence of this process is presented in Section 3.4.5.3.

3.4.5.2 Reuse of the intermediary results

As mentioned before, the view factors depend only on the geometry of the model. They do not depend on the thermo-optical properties of the surfaces.

If different sets of thermo-optical properties are associated with a same geometry, the interest of Gebhart's method appears clearly because the geometrical configuration has not to be processed several times, once for each set. The view factors and the direct heat fluxes do not need to be recalculated.

3.4.5.3 Convergence of the iterative process

Once the view factors have been computed, Gebhart's method is used to compute the radiative exchange factors. In the proposed algorithms, an iterative scheme has been implemented. Each iteration represents the result of one reflection step. An error criterion is evaluated, from one step to the next. If the criterion is below a user-defined threshold, convergence is achieved and the iterative process is stopped.

The convergence of this method depends on the average diffuse reflectivity of the model. This average reflectivity can be obtained by weighting each patch's diffuse reflectivity according to its surface area [SP94]:

$$\rho^d = \frac{\sum_{i=1}^{N} A_i \rho_i^d}{\sum_{i=1}^{N} A_i} \tag{3.102}$$

This has been illustrated in the case of two concentric spheres (this case is described in Section 3.4.7.1). Four sets of thermo optical properties (corresponding to an average diffuse reflectivity ρ^d varying from 20% to 80%) are considered and the error criterion is plotted in function of the index of the iteration. An increase in the average diffuse reflectivity will slow down the rate of convergence of the iterative process.

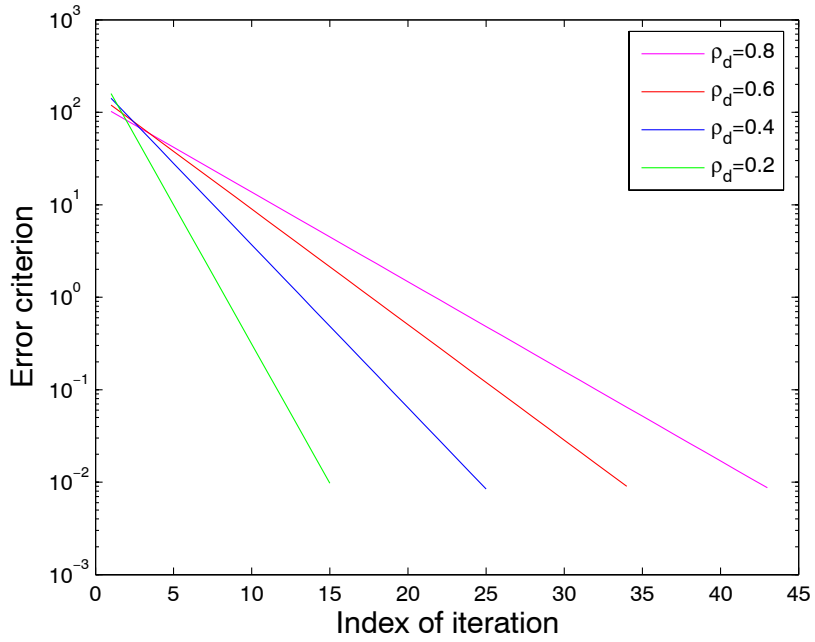


Figure 3.37: Convergence of the iterative scheme in function of the diffuse reflectivity.

3.4.6 Limitative hypotheses

In [Pla97], three restrictions which apply to Gebhart's method have been established:

1. (a) the thermal emission must be diffuse;
   (b) the reflection can only be diffuse;
2. (a) the surfaces which compose the geometrical model must be isothermal;
   (b) the emissivity of these surfaces must be uniform;
3. the surfaces which compose the geometrical model must be uniformly irradiated.

Several comments must be made concerning these hypotheses.

3.4.6.1 Comments on the first hypothesis

The first hypothesis is linked to the main advantage of Gebhart's method. This method allows us to combine an emission process and a reflection process, both of which are characterized by the same angular distribution, *i.e.* the diffuse one. It is possible to imagine combining an emission process and a reflection process which are non-diffuse but have the same angular distribution in an improved, extended Gebhart's method.

The view factors allow us to compute the energetic balance in the 3D model, where the balance is based on the radiosity of the surfaces (*i.e.* the sum of the diffuse self emitted

power and the diffuse reflection). Since the self emission and the diffuse reflection are combined, the diffuse reflection is assumed to be computed with the same level of accuracy as the emission process. For example, if 10 000 rays are traced from an emitter, Gebhart's method allows us to consider the first intersection of these rays both for the emission process and the reflection.

In a classical MCRT used to compute the REF, this combination is not used. A ray impacting a diffuse surface will be reflected in an unique direction with the whole energy. To accurately model the diffuse reflection, the whole hemisphere should be decomposed in a large number of samples, each one carrying a fraction of the whole energy. This sampling is not possible because it would imply an exponential generation of secondary rays. The modeling of the diffuse reflection with a unique ray is not correct. However, regarding the large number of rays traced during a complete MCRT, this error decreases rapidly.

3.4.6.2 Comments on the second hypothesis

Concerning the second hypothesis, a uniform temperature field can be necessary to insure a uniform emissivity, if temperature-dependent emissivity is considered.

Another comment is the fact that the necessary data to model an emissivity varying across the surfaces were not available for this thesis.

3.4.6.3 Comments on the third hypothesis

The last hypothesis is the most restrictive one. "Uniformly irradiated" concerns:

1. the irradiation H received from the other surfaces;
2. the external irradiation H_0, received from the Sun and other bodies, like planets and moons.

The first component is necessary to compute the balance of the thermal system when no external irradiation is considered, when only the self emitted power of the surfaces is considered. This component only concerns the geometry of the model. This distinction appears naturally during the development of the thermal equations. It is crucial for the demonstration of the equivalence of the radiosity equations and Gebhart's formulation.

The third hypothesis points out the weakness of Gebhart's method. This method is the result of two steps: the computation of the view factors by ray tracing and the matrix derivation of the Gebhart's factors. Between the two steps, the considered surfaces are not the same. During the ray tracing process, the discontinuities are taken into account. If a surface has a partial visibility to another, the shadow can be correctly computed so that the view factor of this surface is accurate. During the second step, it is the whole surface which is characterized by the previously computed view factor. The computation of the

shadow is lost and the whole surface is characterized by a constant, uniform view factor. This can induce errors in the two formulations of Gebhart's method.

3.4.7 Extension of Gebhart's method

In the previous Sections, the advantages and restrictions of Gebhart's method have been presented. Now, we will explain how we can extend the field of applications of this method.

3.4.7.1 First hypothesis

The first extension concerns the diffuse thermal emission. As previously mentioned, diffuse emissivity is a hypothesis of this work. Then the first extension consists in including specular reflection into Gebhart's method. The formulation of the extended Gebhart's method is given by the following formula, which is proved in Appendix E.1.

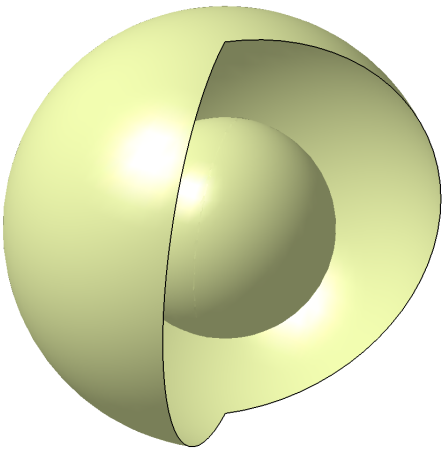
$$B_{i-j} = \epsilon_j F^s_{i-j} + \sum_{k=1}^{N} F^s_{i-k} \rho_k B_{k-j} \tag{3.103}$$

where F^s_{i-j} is the extended, specular view factor from patch P_i to P_j. The energy diffusely emitted by a surface P_i can be reflected either diffusely or specularly. The specular reflection is taken into account by the specular view factor (during the ray tracing process) while the diffuse reflection is computed with Gebhart's method, when equation (3.103) is solved.

If there are specular reflectors in the 3D model, secondary rays are observed during the ray tracing. In case of collision based ray tracing [DBB03], the probability that an incoming ray on P_k is re-emitted is equal to ρ^s_k. This probability must be compared with the probability of re-emission in the case of a complete ray tracing, which is equal to $1 - \epsilon_k = \rho^s_k + \rho^d_k$. Clearly, the interest of Gebhart's method decreases when highly specular reflectors are present in the model.

Remark: we suspect that the first hypothesis could have been motivated by the fact that the European software for space analysis (Esarad [Doc04] and Thermica [Doc03]) may have an error when computing the extended view factors. If the extended view factors are wrong, it is obvious that the results of Gebhart's method will be incorrect. But this error is not due to Gebhart's method and should not be an argument against the method.

In order to illustrate the extension of Gebhart's method to specular reflectors, the model presented in the right Figure has been implemented. It is composed of two concentric spheres (in the Figure on the right, the outer sphere has been opened to show the inner one). Each sphere only has one active face, the one which is orientated to the other sphere. The two spheres are characterized by a specular reflectance equal to 80%. Equation (3.103) is solved for this particular two-node model. Extended Gebhart's method yields the value 0.555556. This value is considered as the reference. Then Esarad is used to compute the corresponding link GR between the two spheres. It yields a radiative exchange factor equal to 0.556024, from the inner sphere to the outer one. This value is close to the reference.



This tends to validate the first extension of Gebhart's method. This result has also be compared with the geometrical method, presented in Section 4.3. The returned value is close to the reference. This tends to validate the geometrical method, combined with Gebhart's method, in order to obtain the radiative exchange factors, in the presence of specular reflectors.

3.4.7.2 Second hypothesis

The second hypothesis, concerning isothermal patches, has also been removed. Gebhart's method has been extended to the case of finite elements.

With a finite element formulation, the temperatures are computed at particular positions, on the nodes of the finite elements. The temperature field is then obtained by a polynomial interpolation on the element.

This allows us to consider non-isothermal elements. In this case, nodal, finite element view factors can be computed. Gebhart's formula (3.100) can be extended in order to be expressed on the nodes.

The example shown below is constituted of square elements of the first degree. The view factors are interpolated linearly across each element (see Figure 3.38–a). These nodal view factors can be used in Gebhart's formula in order to obtain the nodal radiative exchange factors, as represented in Figure 3.38–b.

These Figures must be compared with the results obtained with isothermal elements and uniform surface-to-surface view factors. Figure 3.39–a represents the distribution of the uniform surface-to-surface view factors while Figure 3.39–b corresponds to Gebhart's

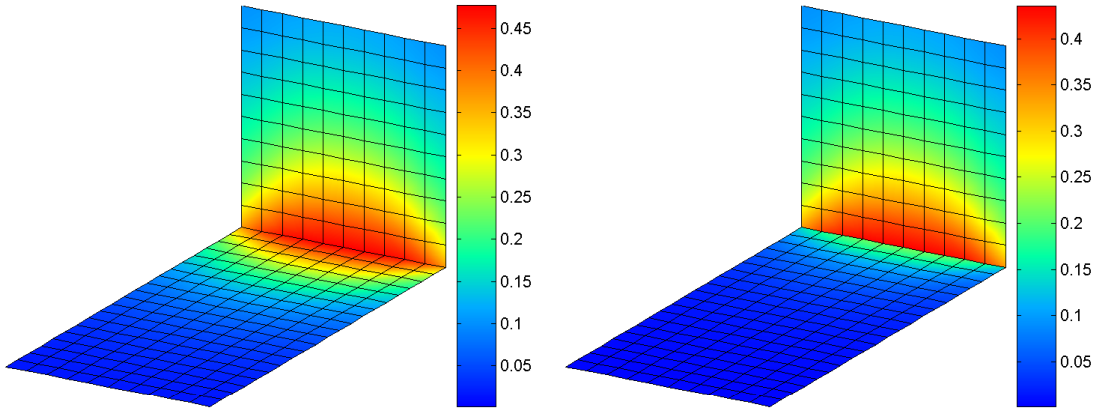


Figure 3.38: Second hypothesis – a) nodal view factors – b) nodal Gebhart's factors.

factors. The color scales of the nodal exchange factors are extended. The nodal exchange factors can reach higher values than the uniform surface-to-surface ones, especially along the common edge. This allows us to obtain a more accurate representation of the exchange factors. For example, the view factor near the common edge is expected to be close to 50%, because almost half of the local hemisphere is covered by the second surface. While the distribution of the uniform view factor can barely exceed 45%, the nodal view factor is closer to the expected value. The use of nodal exchange factors yields results of a higher accuracy.

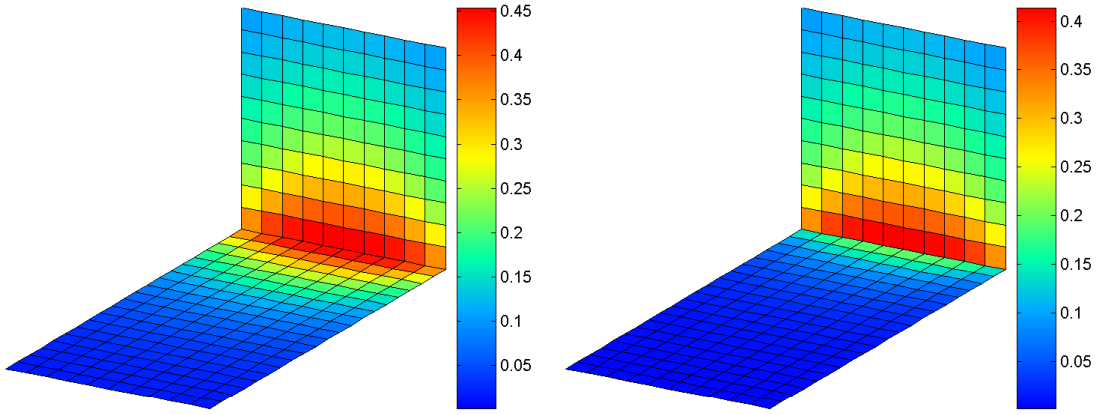


Figure 3.39: Second hypothesis – a) uniform view factors – b) uniform Gebhart's factors.

3.4.7.3 Third hypothesis

The third hypothesis must be studied in two steps. When requiring a uniform irradiation on a given surface, the third hypothesis concerns the irradiation H from the other surfaces

and the external irradiation H_0.

Irradiation from the geometrical model In [Pla97], the author identified three configurations which violated Gebhart's method. The first two only concern the geometrical model. The third one is linked to the external irradiation.

This Section shows how the first two issues can be solved by an adapted mesh of the geometry.

The first case (see Figure 3.40–a) corresponds to two surfaces, numbered 1 and 3, separated by a surface 2. Surfaces 1 and 3 cannot see each other. The corresponding view factors are equal to zero. The true Gebhart's factors are also equal to zero. However, if the two sides of the second surface are characterized by the same identifier, Gebhart's method will introduce an artificial link from surface 1 to surface 2, then from surface 2 to surface 3, so that Gebhart's factor B_{1-3} is larger than zero. The solution consists in decoupling the two faces of element 2, as it is done in Figure 3.40–b. Element 2 is decomposed into a surface 2' (in red) and a surface 2" (in blue). No artificial link is possible. This artifact can be solved with a proper mesh.

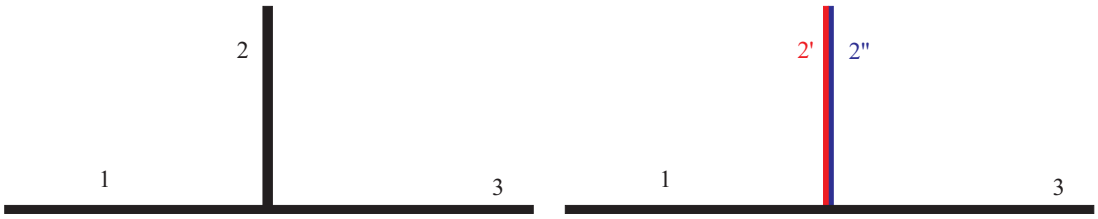


Figure 3.40: First situation – a) configuration – b) correction.

The second case is represented in Figure 3.41–a. It represents two rectangles of different sizes, sharing a common edge. During the computation of the view factors by ray tracing, the rays emitted by surface 1 which impact surface 2 are not uniformly distributed across 2. When computing Gebhart's factor from 1 to outer space, the energy from 1 to 2 is artificially uniformly distributed across 2. As the view factor from 2 to outer space is large, the energy emitted by 1, reflected by 2 has a large visibility to outer space. Gebhart's method will over-estimate the link from 1 to outer space. The solution is represented in Figure 3.41–b. The large surface must be decomposed into elements of equivalent areas.

In order to illustrate this case, the test case of Figure 3.42–a is considered. The geometrical method is used with an underlying finite element mesh made of 36 triangles. The computed view factors are accurate. For example, the computed view factor from the small surface to the large one is equal to 0.4509, which is close to the analytical value of 0.4508.

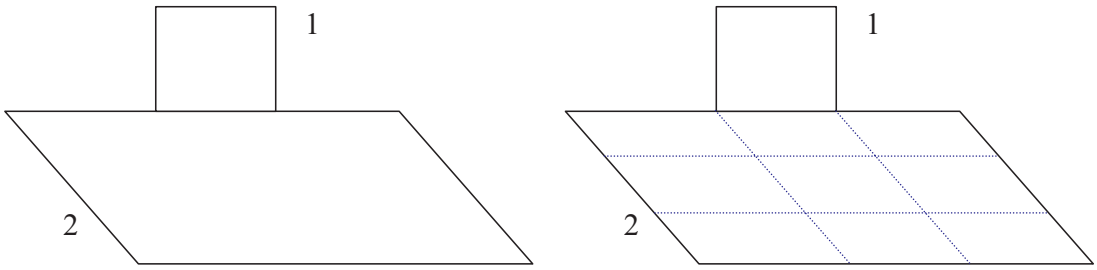


Figure 3.41: Second situation – a) configuration – b) correction.

On the other hand, the radiative exchange factors are not correct. The coupling between the small surface and itself is under-estimated (0.0105 for the computed value, 0.0203 for the analytical value; the relative error is larger than 48%). This is because Gebhart's factor from the small surface to outer space is over estimated. This error can be decreased with the geometrical method by choosing a finer underlying finite element mesh.

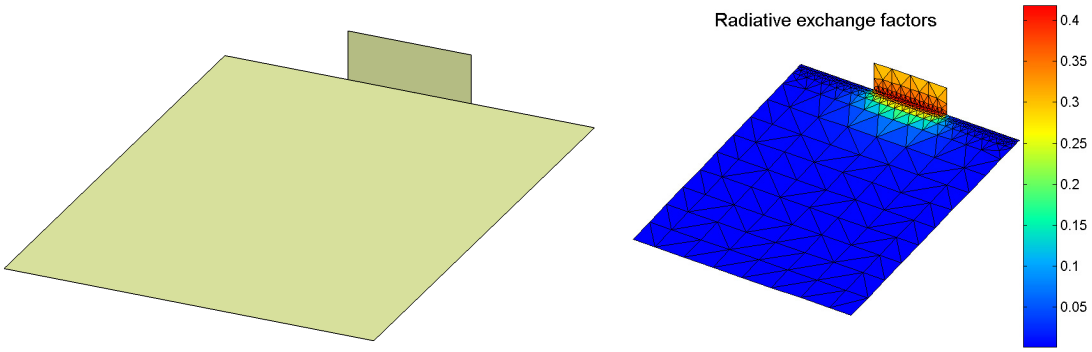


Figure 3.42: Test case – a) configuration – b) radiative exchange factors.

To obtain more accurate results, an adaptive mesh has been used, based on a characteristic length increasing with the distance to the common edge. The results are plotted in Figure 3.42–b.

One way to perform the refinement of the mesh has been implemented in the hierarchical method [VB07, Vue06b]. The circular approximation [WEH89, Vue05] can be used in order to quickly over-estimate the view factors between two surfaces. This approximation is based on the point wise view factor from a point dA_i to a disc A_j, of radius r, centered above dA_i, at a distance h (see Figure 3.43–a). This view factor is given by the following relation [How82, Won76]:

$$F_{di-j} = \frac{r^2}{r^2 + h^2} \tag{3.104}$$

In the circular approximation, a patch P_j of area A_j (see the dashed contour in Figure 3.43–b) is replaced by a circle of the same area. The relative orientations of the surfaces are taken into account to yield the following approximation:

$$F_{di-j} = \frac{A_j}{A_j + \pi h^2} \cos(\theta_i) \cos(\theta_j) \tag{3.105}$$

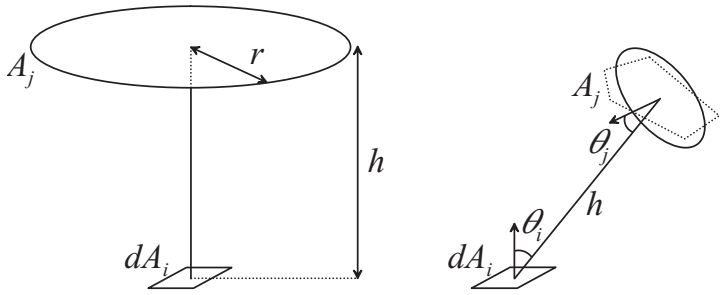
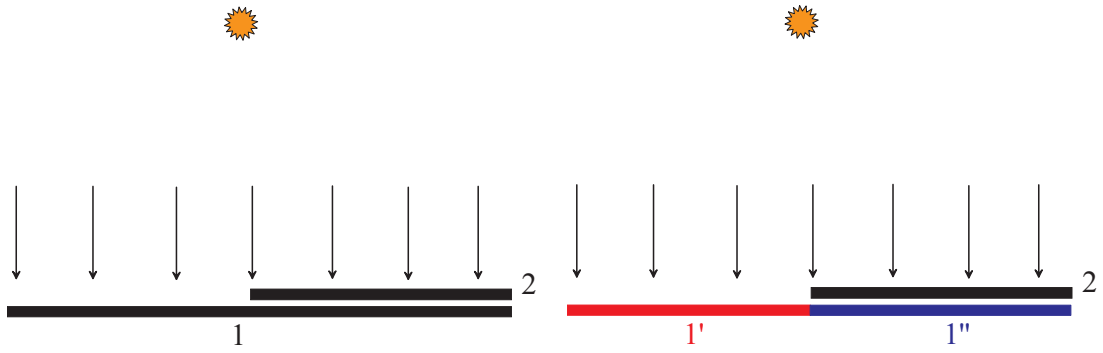


Figure 3.43: Circular approximation – a) configuration – b) approximation.

When the approximation is greater than a given threshold, the mesh must be locally refined. This method can be used in order to identify situations similar to the second case and to yield an adapted, corrected mesh. This adaptive mesh can cancel the geometrical component of the third hypothesis of Gebhart's method, *i.e.* the component relative to the computation of the radiative exchange factors (equation (3.100)).

External irradiation The third test case is represented in Figure 3.44–a. A first surface is partially exposed to the Sun, pointing to the Sun; a second surface partially covers the first one, pointing in the opposite direction. If Gebhart's method for heat fluxes (equation (3.101)) is used, an artificial link appears from the Sun to surface 2. Indeed, surface 1 is characterized by non-zero view factors to the Sun and to surface 2. At the end of the view factors computation, the solar energy is considered as uniformly distributed on surface 1. Then Gebhart's method creates a link from the Sun to surface 2, although surface 2 is totally masked.

A solution would be to subdivide surface 1 following the projection of the contour of 2, following the direction of the Sun (see Figure 3.44–b). The drawback of this solution is that it would require the computation of the different projections for the different positions of the satellite around its orbit. The different projections should be included in one mesh. This solution is not easily applicable. For this reason, the third hypothesis concerning Gebhart's method for heat fluxes cannot be removed. More over, sharp shadows are commonly observed in space applications. For this reason, we do not recommend Gebhart's method for heat fluxes (equation (3.101)).



(a) Third case (b) Third case corrected

Figure 3.44: Third situation – a) configuration – b) correction.

3.4.8 Proposed application of Gebhart's method and conclusions

We have shown that Gebhart's method is not unique. The name "Gebhart's method" corresponds to two distinct formulations. The first one, relative to radiative exchange factors, can be used while the second one, relative to absorbed heat fluxes, is definitely not adapted for radiative heat transfer with solar irradiation.

During the extension of the method, we have concluded that the third limitation could not easily be removed. In fact, this hypothesis can be fulfilled for the geometrical component, that is for the computation of the radiative exchange factors. On the other hand, the component relative to the external irradiation, that is for the computation of the absorbed heat fluxes, could not be easily adapted for Gebhart's method.

For this reason, we recommend using Gebhart's method only for the computation of the radiative exchange factors. The use of an adaptive mesh, based on a geometrical criterion such as the circular approximation, is also recommended. A complete ray tracing process should be necessary to compute the absorbed heat fluxes.

In this Section, the restrictions imposed on Gebhart's method for computing radiative exchange factors have been removed. Some of the extensions are purely theoretical. Others require a special algorithm in order to obtain an adapted mesh.

The limitations concerning the absorbed heat fluxes cannot easily be removed, especially for the solar irradiation, where sharp shadows must be considered.

This concludes the Chapter dedicated to the exchange factors. As these exchange factors are obtained by ray tracing, the next Chapter is dedicated to a new acceleration technique, combining optimized intersection routines and ray tracing acceleration methods.

Chapter 4

Acceleration of the ray tracing

Contents

This Chapter brings together the developments performed in order to accelerate the ray tracing, either for the computation of exchange factors and external heat fluxes. After a state of the art, a description of optimized ray-surface intersection procedures is given. The detection of the nearest candidates for intersection is also addressed. The geometrical method, which consists in the combination of the geometry with a finite element mesh, is detailed; this method results in an acceleration of the ray tracing, as well as a way to smoothly integrate conduction with radiation, while keeping the geometrical definition of the model. The geometrical method is combined with uniform spatial subdivision (USD), yielding a two-level acceleration method.

4.1 State of the art

Before detailing the acceleration methods developed in this thesis, as well as the reasons which motivated this choice, a state of the art of some acceleration techniques developed in image synthesis will be presented; some of these methods have already been applied to radiative heat transfer.

The efficiency of a ray tracing algorithm is based on the ray-surface (or the ray-element) intersection procedure. The complexity of this operation depends on the primitive objects considered for intersection. The required resources are also function of the number of times that the operation is performed.

4.1.1 Estimation of the number of intersection computations

Let N_{ray} be the number of traced rays, for each element; N_{elem} the number of elements; n denotes the number of secondary rays, *i.e.* the number of reflections, for each ray. If no acceleration technique is implemented, each of the N_{elem} surfaces needs to be tested for intersection, for each traced ray. The number of intersection tests N_{int} is then proportional to the square of the number of elements:

$$N_{\text{int}} = \left(N_{\text{ray}} \times N_{\text{elem}}\right) N_{\text{elem}} \times (n+1) \tag{4.1}$$

$$= (n+1) N_{\text{ray}} N_{\text{elem}}^2 \tag{4.2}$$

For example, if 10 000 rays are traced from 5 000 elements and if an average number of secondary rays of 9 (corresponding to an exponential attenuation ΔE of 99%, with an average attenuation factor α of 50%) is considered, $2.5\,10^{12}$ intersection operations have to be computed, which is prohibitive.

A ray tracing acceleration technique is obviously necessary. The choice of the technique is based on several factors, such as the scope (nature of the primitive objects, consideration of the temporal dimension), the performances (speed, behaviour in complex environment), the required resources (trade-off between CPU-time and storage), the simplicity of the algorithm and its implementation [Ver08].

4.1.2 Classification of acceleration techniques

In Figure 4.1, from [AK02], a broad classification of the acceleration techniques is given. Three different strategies can be considered:

1. the average cost of intersecting a ray can be reduced;
2. the number of traced rays can be reduced;
3. rays can be replaced with more general entities, considered as bundles of rays.

The first category includes the acceleration of the ray-surface intersections and the reduction of the number of ray-object intersections. In the first subcategory, the use of efficient ray-primitive intersections can be found, as it will be presented in Section 4.2, and the definition of object bounding volumes[1] [RW80]. The second subcategory includes bounding volume hierarchies [KK86, EWM08], space subdivision techniques as uniform spatial subdivision (USD) [FTI86], where the volume is subdivided in a regular 3D grid, the octree [Gla84, RUL00, Lev90], which is able to take into account the spatial density of the surfaces, or the kd-tree [FS88], more adapted to the real density of the model. The

[1] This method is based on the use of bounding volumes [SHH99], such as spheres [Whi80], which are the simplest surfaces for intersection, and convex hulls made of several planes [KK86]. In reference [WHG84], the author studied the trade-off between the tightness of the bounding volume and the cost of the intersection process.

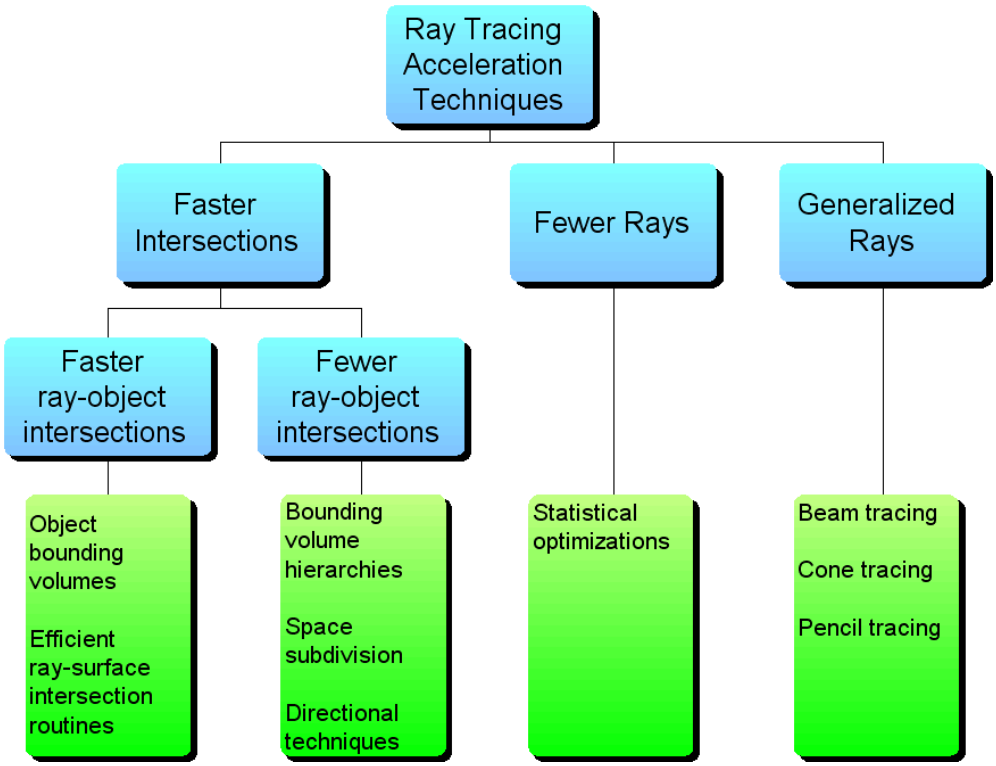


Figure 4.1: A broad classification of acceleration techniques.

main interest of these methods is that *they consider objects along a given ray roughly in the order that they occur along the ray* [KK86].

The second category, named "fewer rays", includes statistical optimization, such as the stratified hemisphere method, and other combinations of sampling techniques [VG95].

The last category includes the methods which consist in replacing the concept of ray with a more general entity, such as cones [Ama84], pencils and beams [DKW85]. The problem of these generalized rays is that their cross section becomes more and more complex after partial intersections. Moreover, it only considers planar surfaces for intersection (the reflection of a beam by a curved surface is no longer a beam). These generalized rays cannot be used for space applications, where the geometry is made of spheres, cylinders and cones. For this reason, they have not been studied in this thesis.

The geometrical method developed in this thesis and presented in Section 4.3 can be classified in the second subcategory because it reduces the number of ray-object intersections by using the spatial coherence of the model. In Section 4.3, the motivations which lead us to this choice are presented.

4.2 Computing intersections between rays and surfaces

4.2.1 Introduction

The purpose of this Section is to present the equations of the ray-surface intersections, for the primitives used in thermal software for space applications[2]. Optimized techniques are presented. The ray-surface intersection is an operation which is performed several times and needs to be as quick as possible. The equations needed to determine the $\{u, v\}$ coordinates of the intersection points on the surfaces are also given. Computing the intersection of a ray with a surface is a mathematically well-defined problem: this can be expressed in the form of a system of three equations with three unknowns.

4.2.2 Equation of the ray

The notations are similar to the ones used in [Hai89]. A point is referenced by a capital letter while a vector is represented by a lowercase letter. A ray is defined by an origin, noted $\overrightarrow{P_R}$ and a direction identified by the vector $\overrightarrow{r_d}$. The equation of the ray is given by the following relation:

$$\overrightarrow{R} \quad = \quad \overrightarrow{P_R} + t\overrightarrow{r_d} \quad \text{with} \quad t > 0 \tag{4.3}$$

The vector $\overrightarrow{r_d}$ is normalized. This condition is not mandatory but it simplifies the next computation step. If the vector is normalized, the parameter t represents the distance between the origin and the intersection point. If the vector is not normalized, the distance is expressed in terms of the length of $\overrightarrow{r_d}$.

The parameter t must be positive in order to identify an intersection. A negative value corresponds to a point behind the origin, which is not part of the ray. The value $t = 0$ will not be associated with an intersection, due to numerical problems.

4.2.3 Ray-plane intersection

Before considering the case of triangles and quadrangles, the intersection of the ray with a plane will be optimized. This is a common operation which must be carefully implemented in order to have a competitive algorithm.

[2]The reference software is Esarad; the implemented primitives are triangles, quadrangles, cones, cylinders, discs, etc. Additional primitives can be considered for ray tracing, such as fractal surfaces, prisms and surfaces of revolution [Kaj83]. In [vW84], ray tracing with objects defined by sweeping planar cubic splines is addressed. Generalized cylinders, *i.e.* surfaces defined by sweeping a two-dimensional contour along a three-dimensional trajectory, are considered in [BK85]. These surfaces will not be presented in this document.

4.2.3.1 Equations of the plane

Given a point P_0 belonging to a plane Π and the normal unit vector $\overrightarrow{n_\Pi}$, the plane Π can be defined as the set of points $\overrightarrow{P}$ which satisfy the following equation [Eti86]:

$$\overrightarrow{P_0} \cdot \overrightarrow{n_\Pi} = 0 \tag{4.4}$$

This relation expresses that Π is the set of the points $\overrightarrow{P}$ which form, with $\overrightarrow{P_0}$, a vector lying in the plane Π and then orthogonal to the normal vector. From equation (4.4), the algebraic formula of the plane can be obtained:

$$\begin{aligned} \overrightarrow{P_0} \cdot \overrightarrow{n_\Pi} &= 0 \\ \Leftrightarrow \overrightarrow{OP} \cdot \overrightarrow{n_\Pi} &= \overrightarrow{OP_0} \cdot \overrightarrow{n_\Pi} && (4.5) \\ &= -D && (4.6) \\ \Leftrightarrow Ax + By + Cz + D &= 0 && (4.7) \end{aligned}$$

where D is a constant, equal to the dot product of the vector $\overrightarrow{OP_0}$ and the normal vector; A, B and C are the three components of the normal vector (the three components cannot be simultaneously equal to zero); x, y and z are the three components of point $\overrightarrow{P}$. Equation (4.7) is the Cartesian equation of the plane. $|D|$ represents the distance from the origin $\{0, 0, 0\}$ to the plane if the normal vector is normalized.

In the internal report [Vue07], we demonstrate also the vectorial equation of the plane, based on three non-aligned points $\overrightarrow{P_A}$, $\overrightarrow{P_B}$ and $\overrightarrow{P_C}$ belonging to Π:

$$\begin{aligned} \overrightarrow{P_A P} &= u\overrightarrow{P_A P_B} + v\overrightarrow{P_A P_C} && (4.8) \\ \Leftrightarrow \overrightarrow{P} &= \alpha\overrightarrow{P_A} + \beta\overrightarrow{P_B} + \gamma\overrightarrow{P_C} && (4.9) \end{aligned}$$

The parameters u and v which appear in the equation (4.8) are real, such as α, β and γ. Two parameters are sufficient to describe a 2-D surface. The following relation exists between the three parameters of the last equation: $\alpha + \beta + \gamma = 1$. The last equation expresses that any point in plane Π can be obtained by a convex linear combination of three points (if they are not aligned).

4.2.3.2 Intersection with the plane

The expressions of the plane and the rays are equalized, in order to establish the equation of the intersection. The following relation is obtained:

$$\overrightarrow{P_R} + t\overrightarrow{r_d} = \overrightarrow{P_A} + u\overrightarrow{P_A P_B} + v\overrightarrow{P_A P_C} \tag{4.10}$$

This is a set of three equations (one for each spatial direction) with three unknowns, t being the distance between the origin and the intersection, u and v localizing the intersection point in Π. Following [Eti86], *a straight line d, defined by an point* $\overrightarrow{P_R}$ *and a vector* $\overrightarrow{r_d}$,

and a plane Π*, defined by a point* $\overrightarrow{P_A}$ *and a normal vector* $\overrightarrow{n_\Pi}$*, can intersect if and only if*

$$\overrightarrow{r_d} \cdot \overrightarrow{n_\Pi} \neq 0 \tag{4.11}$$

The demonstration of this relation can be found in reference [Eti86]. If the definition of the ray (4.3) is introduced in the expression of the plane (4.7), the following relation of the unknown t is obtained:

$$A(x_R + tx_d) + B(y_R + ty_d) + C(z_R + tz_d) + D = 0 \tag{4.12}$$

The resolution of this equation yields the following expression of t:

$$t = -\frac{Ax_R + By_R + Cz_R + D}{Ax_d + By_d + Cz_d} \tag{4.13}$$

$$= -\frac{\overrightarrow{n_\Pi} \cdot \overrightarrow{P_R} + D}{\overrightarrow{n_\Pi} \cdot \overrightarrow{r_d}} \tag{4.14}$$

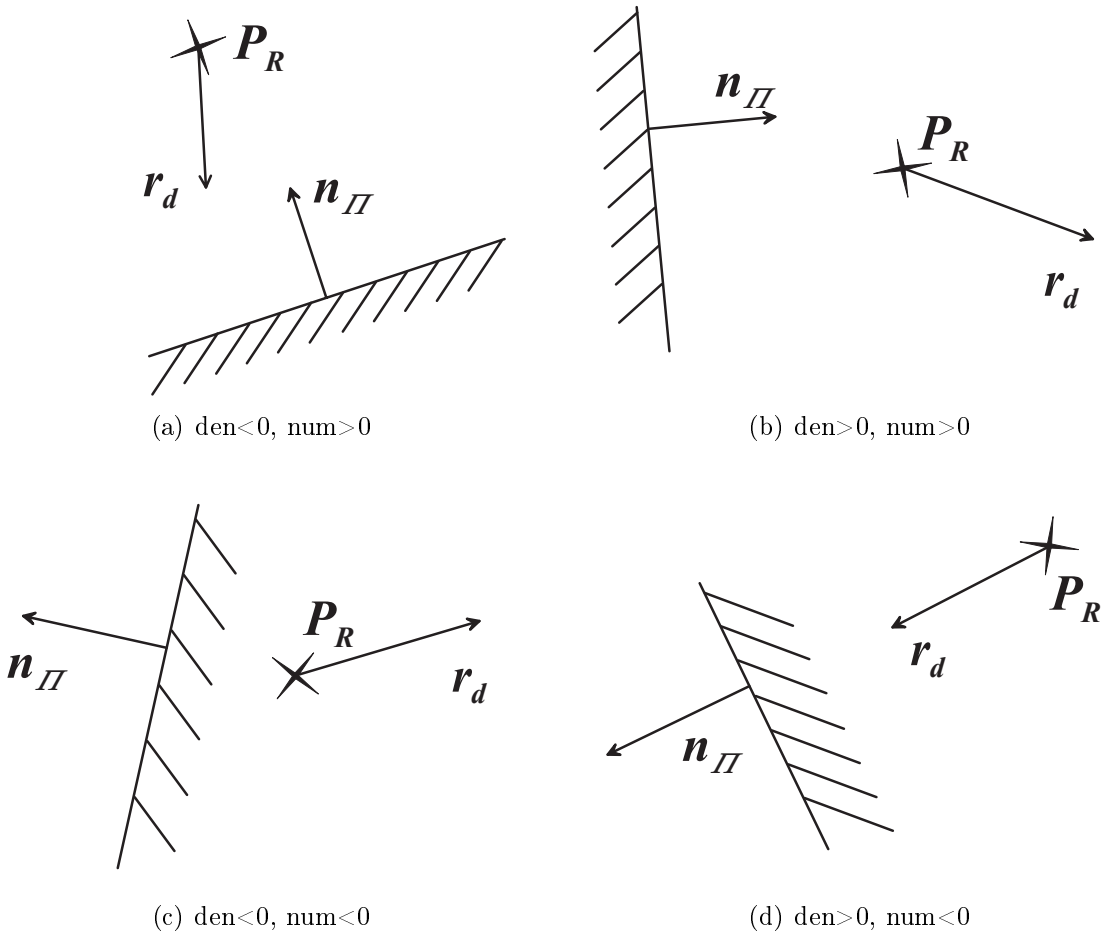
$$= -\frac{\text{num}}{\text{den}} \tag{4.15}$$

First of all, the denominator $\text{den} = \overrightarrow{n_\Pi} \cdot \overrightarrow{r_d}$ has to be computed. If it is equal to zero, the intersection is not determined and the ray is parallel to the plane. If the origin of the ray $\overrightarrow{P_R}$ belongs to the plane, the numerator is also equal to zero and the whole ray is included in the plane. If the origin does not belong to the plane, no intersection can be observed.

The numerator and denominator can be interpreted geometrically. The numerator is linked to the visibility of the plane, *i.e.* to the position of the ray's origin with respect to the plane. If it is positive, it implies that the origin is placed beyond the plane, the origin can be seen from the plane (see Figures 4.2–a and –b). The denominator is linked to the relative orientation of the normal $\overrightarrow{n_\Pi}$ and the direction of the ray $\overrightarrow{r_d}$. If it is positive, this implies that the vector normal to the plane is not oriented towards the incoming ray (see Figures 4.2–b and –d). No intersection is possible.

The computation process is decomposed into five successive steps:

1. Computation of the denominator;
2. if $\text{den} \geq 0$, the situation corresponds to the configurations 4.2–b or –d, no intersection can be possible and the process is stopped;
3. if $\text{den} < 0$, the numerator is computed;
4. if $\text{num} \leq 0$, the case in figure 4.2–c applies, the procedure is interrupted;
5. the distance t is computed using the formula (4.15).



(a) den<0, num>0 (b) den>0, num>0

(c) den<0, num<0 (d) den>0, num<0

Figure 4.2: Geometrical interpretation of the denominator and numerator of equation (4.15).

Steps 1 and 2 allow us to reject about half of the surfaces in the model. This step is called *backface culling* [ZH97, HG03]; it requires three multiplications and two additions corresponding to the dot product, and a comparative test. This is more efficient than the test performed in steps 3 and 4. Moreover, the number of rejected surfaces with steps 3 and 4 is low (much lower than 50%) if a ray tracing algorithm such as the uniform spatial subdivision or the octrees is implemented.

In order to accelerate the intersection computation, the parameter D and the normal $\overrightarrow{n_\Pi}$ are computed and stored during a pre-processing step, before the ray tracing process.

Remark: other formulations of the computation of t have been studied in the internal report [Vue07]. Here, only the most efficient one is considered.

4.2.4 Ray-triangle intersection

Once the parameter t has been computed and found to be positive, the position of the intersection point can be computed, thanks to the equation (4.3). If it is introduced into the relation (4.10), a set of three equations with only two unknowns is obtained. The size of this set (and the relative complexity of its resolution) can be reduced by only considering two of these three equations. The choice of the retained equations is based on the normal $\overrightarrow{n_\Pi}$. The dimension corresponding to the main component of $\overrightarrow{n_\Pi}$ (in absolute value) is removed. The intersection point and the triangle are projected in one of the main planes[3]. The direction's selection based on the normal guarantees the accuracy of the intersection computation and prevents the degeneration of the triangle during the projection. Let us note r and s the two remaining components (selected among x, y and z). The equation (4.10) can be rewritten as follows:

$$\begin{pmatrix} r_R \\ s_R \end{pmatrix} + t \begin{pmatrix} r_d \\ s_d \end{pmatrix} = \begin{pmatrix} r_A \\ s_A \end{pmatrix} + u \begin{pmatrix} r_{AB} \\ s_{AB} \end{pmatrix} + v \begin{pmatrix} r_{AC} \\ s_{AC} \end{pmatrix} \tag{4.16}$$

The two values u and v are given by the following relations:

$$\begin{cases} u = \dfrac{s_{AC}(r_R + tr_d - r_A) - r_{AC}(s_R + ts_d - s_A)}{s_{AC}r_{AB} - r_{AC}s_{AB}} \\ v = \dfrac{r_{AB}(s_R + ts_d - s_A) - s_{AB}(r_R + tr_d - r_A)}{s_{AC}r_{AB} - r_{AC}s_{AB}} \end{cases} \tag{4.17}$$

The denominator corresponds to the highest component of the normal vector, due to the particular choice of the components r and s. This insures the good numerical resolution of this set of equations.

In order to accelerate the ray-triangle intersection process, the values of r_{AB}, r_{AC}, s_{AB}, s_{AC}, r_R and s_R can be computed and stored during a pre-processing step, as well as the index of the main component of the normal $\overrightarrow{n_\Pi}$.

The ray-triangle intersection routine identifies an intersection with the triangle defined by the three corners $\overrightarrow{P_A}$, $\overrightarrow{P_B}$ and $\overrightarrow{P_C}$ if the values u and v satisfy the following relations:

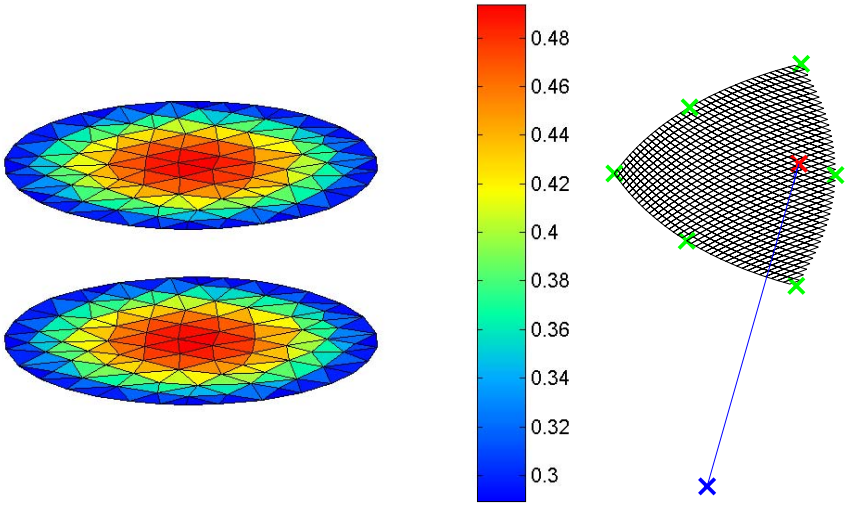
$$\begin{cases} u \geq 0 \\ v \geq 0 \\ u + v \leq 1 \end{cases} \tag{4.18}$$

4.2.5 Triangle of the second degree

In the geometrical method, the geometrical primitives will be associated with a triangular mesh. This implies an approximation of the curved contours. In the case of a disc meshed with triangles, the use of triangles of the first degree will introduce a geometrical error

[3] "Main planes" refer to the three planes defined by the relations $x = 0$, $y = 0$ and $z = 0$.

in the view factors between the discs (the view factors are displayed in Figure 4.3–a). A way to reduce this error is to use triangles of the second degree (see Figure 4.3–b). These triangles are defined by six nodes: three nodes on the extremities and three nodes on the middle of each edge. An edge can then be limited by a parabolic contour.



(a) Test case

(b) Ray-triangle intersection

Figure 4.3: Triangle of the second degree.

If numbers 1 to 3 refer to the nodes on the extremities and numbers 4 to 6 correspond to the nodes on the middle of the edges, the following shape functions are defined in function of the $\{u, v\}$ parameters:

$$\left\{\begin{array}{rcl} N_1(u,v) & = & [2(1-u-v)-1]\,[1-u-v] \\ N_2(u,v) & = & (2u-1)\,u \\ N_3(u,v) & = & (2v-1)\,v \\ N_4(u,v) & = & 4u\,(1-u-v) \\ N_5(u,v) & = & 4uv \\ N_6(u,v) & = & 4v\,(1-u-v) \end{array}\right. \tag{4.19}$$

The components of the intersection are linear combinations of these shape functions. These equations can be written as quadratic functions of u and v. The following relations are obtained:

$$\left\{\begin{array}{rcl} \alpha_1+\alpha_2 u+\alpha_3 v+\alpha_4 uv+\alpha_5 u^2+\alpha_6 v^2+\alpha_7 t & = & 0 \\ \beta_1+\beta_2 u+\beta_3 v+\beta_4 uv+\beta_5 u^2+\beta_6 v^2+\beta_7 t & = & 0 \\ \gamma_1+\gamma_2 u+\gamma_3 v+\gamma_4 uv+\gamma_5 u^2+\gamma_6 v^2+\gamma_7 t & = & 0 \end{array}\right. \tag{4.20}$$

where the coefficients α_i are function of the x-component of the six nodes, the origin and the direction of the ray; the coefficients β_i are related to the y-component of these entities

and γ_i correspond to the z-components. These equations cannot be solved analytically. A Newton scheme has been implemented, given by the following formula [Lit99]:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{4.21}$$

where x_n is the vector of unknown at the iteration number n. If the system of equation (4.20) is written in the form of a matrix F of the three unknowns t, u and v, the equation (4.21) is rewritten as follows:

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix}_{n+1} = \begin{bmatrix} t \\ u \\ v \end{bmatrix}_{n} - \text{inv}\,(F')\,F \tag{4.22}$$

The convergence of this algorithm depends on the initialization. If the initial point (in the space of unknowns) is close to the solution, a few iterations are necessary to achieve the final result. The second-order triangle has been approximated with the first-order one to obtain an initialization which is close to the solution. Then the convergence is very fast. The intersection is illustrated in Figure 4.3–b. The ray is represented in blue while the triangle is defined by the six green nodes. The intersection is identified by the red cross.

Remark: a second-order triangle can be non-planar if the six nodes are not included in the same plane. Our intersection routine is adapted to this possibility. Nevertheless, the use of second-order triangles must be limited to the case of curved contours because it implies additional operations, when compared to the first-order triangle.

4.2.6 Ray-rectangle intersection

The ray-rectangle intersection is similar to the ray-triangle intersection, except that the u and v parameters must satisfy different relations.

$$\begin{cases} 0 \leq u \leq 1 \\ 0 \leq v \leq 1 \end{cases} \tag{4.23}$$

4.2.7 Ray-quadrangle intersection

If an irregular quadrangle is considered, the equations of u and v are more complex. The four corners of the quadrangle are assumed to be coplanar. The quadrangle is defined by the following equation:

$$\overrightarrow{P_M} = \overrightarrow{P_A} + u\left(\overrightarrow{P_B} - \overrightarrow{P_A}\right) + v\left(\overrightarrow{P_D} - \overrightarrow{P_A}\right) + uv\left(\overrightarrow{P_A} - \overrightarrow{P_B} + \overrightarrow{P_C} - \overrightarrow{P_D}\right) \tag{4.24}$$

This equation is no longer linear because of the term uv. This is a bilinear interpolation following u and v. r and s denote the smallest components of the normal vector. The

intersection set of equations becomes:

$$\left\{ \begin{array}{rcl} r_M & = & ur_N + vr_P + uvr_Q \\ s_M & = & us_N + vs_P + uvs_Q \end{array} \right. \tag{4.25}$$

where

- the index M represents the vector $\overrightarrow{P_M} - \overrightarrow{P_A}$;
- the index N corresponds to the vector $\overrightarrow{P_B} - \overrightarrow{P_A}$;
- the index P identifies the vector $\overrightarrow{P_D} - \overrightarrow{P_A}$;
- the index Q designates the vector $\overrightarrow{P_A} - \overrightarrow{P_B} + \overrightarrow{P_C} + \overrightarrow{P_D}$.

The resolution of the equation (4.25) yields the following system:

$$\left\{ \begin{array}{rcl} v & = & \frac{s_M - us_B}{s_D + us_F} \\ u^2(s_B t_F - t_B s_F) + u(t_M s_F - s_M t_F + s_B t_D - t_B s_D) + (t_M s_D - s_M t_D) & = & 0 \end{array} \right. \tag{4.26}$$

The second equation yields two values for u. These two values correspond to the same point but only one can be in the interval $[0, 1]$. If one of these two u-values is in the interval $[0, 1]$, this value is used to compute the corresponding value of v. The intersection routine identifies an intersection if the relation (4.23) is verified. This formulation is exact and faster than the iterative scheme implemented in Esarad [Doc04].

4.2.8 Ray-disc intersection

The disc is a planar surface. The parameter t can be computed by the equation (4.14) and the intersection point is given by the following relation:

$$\overrightarrow{P_M} = \overrightarrow{P_R} + t\overrightarrow{r_d} \tag{4.27}$$

4.2.8.1 Equation of the disc

The disc is defined by a centre, noted $\overrightarrow{P_1}$, a normal vector $\overrightarrow{n_\Pi}$ and a radius R. A local coordinate system is defined with $\overrightarrow{z} \equiv \overrightarrow{n_\Pi}$; $\overrightarrow{x}$ and $\overrightarrow{y}$ are contained in the disc's plane. The disc is then defined as the set of points $\overrightarrow{P}$ which satisfy the following relation:

$$\overrightarrow{P} = \overrightarrow{P_1} + r\cos(\alpha)\overrightarrow{x} + r\sin(\alpha)\overrightarrow{y} \tag{4.28}$$

where $r \in [0, R]$ and $\alpha \in [0, 2\pi[$.

4.2.8.2 Intersection with the disc

To compute the intersection in polar coordinates (this is necessary to apply the geometrical method, presented in the next Section), the coordinates of the intersection point (4.27) have to be introduced into the equation of the disc (4.28). The following expressions are obtained:

$$\left\{ \begin{array}{rcl} r & = & \left|\left|\overrightarrow{P_1P_M}\right|\right| \\ \alpha & = & \arctan\left(\frac{\overrightarrow{P_1P_M}\cdot\overrightarrow{y}}{\overrightarrow{P_1P_M}\cdot\overrightarrow{x}}\right) \end{array} \right. \tag{4.29}$$

Remark: The computation of α must be performed carefully. The function `arctan` returns a value on the interval $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ while the angle α is defined on $[0, 2\pi]$. The Fortran function `atan2` solves the problem, returning a value in $[-\pi, \pi]$. Then, a simple translation of 2π allows us to obtain the expected value.

4.2.9 Ray-cylinder intersection

The case of a right circular cylinder is considered. This surface is not a planar one. An optimized ray-cylinder intersection technique developed in the references [CK94, CW94] is presented.

4.2.9.1 Equation of the cylinder

A cylinder is defined by a point $\overrightarrow{P_1}$, centre of the circular basis, an axis $\overrightarrow{n} \equiv \overrightarrow{z}$, a height H which is the length of the cylinder following the $\overrightarrow{z}$ direction and a radius R. An additional point $\overrightarrow{P_3}$ is defined in the basis of the cylinder, which will be used as a reference for the angles α.

A point $\overrightarrow{P}$ of the cylinder is defined by two parameters: h is the height of the point along the axis and α is the angle between $\overrightarrow{P_3}$ and the projection of $\overrightarrow{P}$ in the basis of the cylinder. It yields the following relation:

$$\overrightarrow{P} = \overrightarrow{P_1} + R\cos(\alpha)\overrightarrow{x} + R\sin(\alpha)\overrightarrow{y} + h\overrightarrow{Z} \tag{4.30}$$

where $\overrightarrow{x}$ is the unit vector corresponding to $\overrightarrow{P_1P_3}$, in the basis of the cylinder, and $\overrightarrow{y}$ completes the local coordinates system.

4.2.9.2 Intersection with the cylinder

The solution presented in this section is based on analytical geometry [Eti86] and has been detailed in reference [CW94]. An alternative method can be found in [CK94].

$\overrightarrow{d}$ corresponds to the common perpendicular to the axis $\overrightarrow{z}$ of the cylinder and to the direction $\overrightarrow{r_d}$ of the ray. This vector is given by the relation (4.31). Let $\overrightarrow{P_C}$ be the point

of the ray where the ray is the closest to the axis of the cylinder. The shortest distance δ between $\overrightarrow{z}$ and $\overrightarrow{r_d}$ is identified thanks to the relations (4.32) and (4.33):

$$\begin{aligned} \overrightarrow{d} &= \frac{\overrightarrow{r_d} \wedge \overrightarrow{z}}{||\overrightarrow{r_d} \wedge \overrightarrow{z}||} && (4.31) \\ \overrightarrow{P_R} + t_C \overrightarrow{r_d} + \delta \overrightarrow{d} &= \overrightarrow{P_1} + h\overrightarrow{z} && (4.32) \\ \delta &= \left| \overrightarrow{P_R P_1} \cdot \overrightarrow{d} \right| && (4.33) \end{aligned}$$

If the distance δ is lower than the radius R, the ray presents two intersections with the cylinder. These two points are noted $\overrightarrow{P_A}$ and $\overrightarrow{P_B}$; they are symmetrical on both sides of the point $\overrightarrow{P_C}$. Distance t_C can be deduced from equation (4.32). The following relation is obtained:

$$t_C = \frac{\overrightarrow{P_R P_1} \cdot \left(\overrightarrow{d} \wedge \overrightarrow{z} \right)}{\overrightarrow{r_d} \cdot \left(\overrightarrow{d} \wedge \overrightarrow{z} \right)} \tag{4.34}$$

Let us consider Figure 4.4. The ray direction is represented by the red line. A straight line $\overrightarrow{z'}$ is built, parallel to the axis $\overrightarrow{z}$ and containing the point $\overrightarrow{P_C}$. The distance between $\overrightarrow{P_C}$ and the axis is equal to δ. The two intersection points are symmetrical about $\overrightarrow{P_C}$. A vector $\overrightarrow{o}$ is built, perpendicular to the axis of the cylinder and to $\overrightarrow{d}$ (the blue construction on figure 4.4). Two rectangular triangles are built, characterized by the relation $k^2 + \delta^2 = R^2$.

s represents the distance between the point $\overrightarrow{P_C}$ and an intersection point ($\overrightarrow{P_A}$ or $\overrightarrow{P_B}$, in red on figure 4.4). This distance is given by the following relations:

$$\begin{aligned} \overrightarrow{o} &= \frac{\overrightarrow{d} \wedge \overrightarrow{z}}{\left|\left| \overrightarrow{d} \wedge \overrightarrow{z} \right|\right|} && (4.35) \\ s\overrightarrow{r_d} \cdot \overrightarrow{o} &= \pm k && (4.36) \\ s &= \left| \frac{\sqrt{R^2 - \delta^2}}{\overrightarrow{r_d} \cdot \overrightarrow{o}} \right| && (4.37) \end{aligned}$$

Then, the two intersection points are identified by the two values t_A and t_B, given by the following formula:

$$\begin{cases} t_A = t_C - s \\ t_B = t_C + s \end{cases} \tag{4.38}$$

Once the intersection point is identified, the corresponding h- and α-values are obtained thanks to the following relations:

$$\begin{cases} h = \overrightarrow{P_1 P} \cdot \overrightarrow{z} \\ \alpha = \arctan\left(\frac{\overrightarrow{P_1 P} \cdot \overrightarrow{y}}{\overrightarrow{P_1 P} \cdot \overrightarrow{x}} \right) \end{cases} \tag{4.39}$$

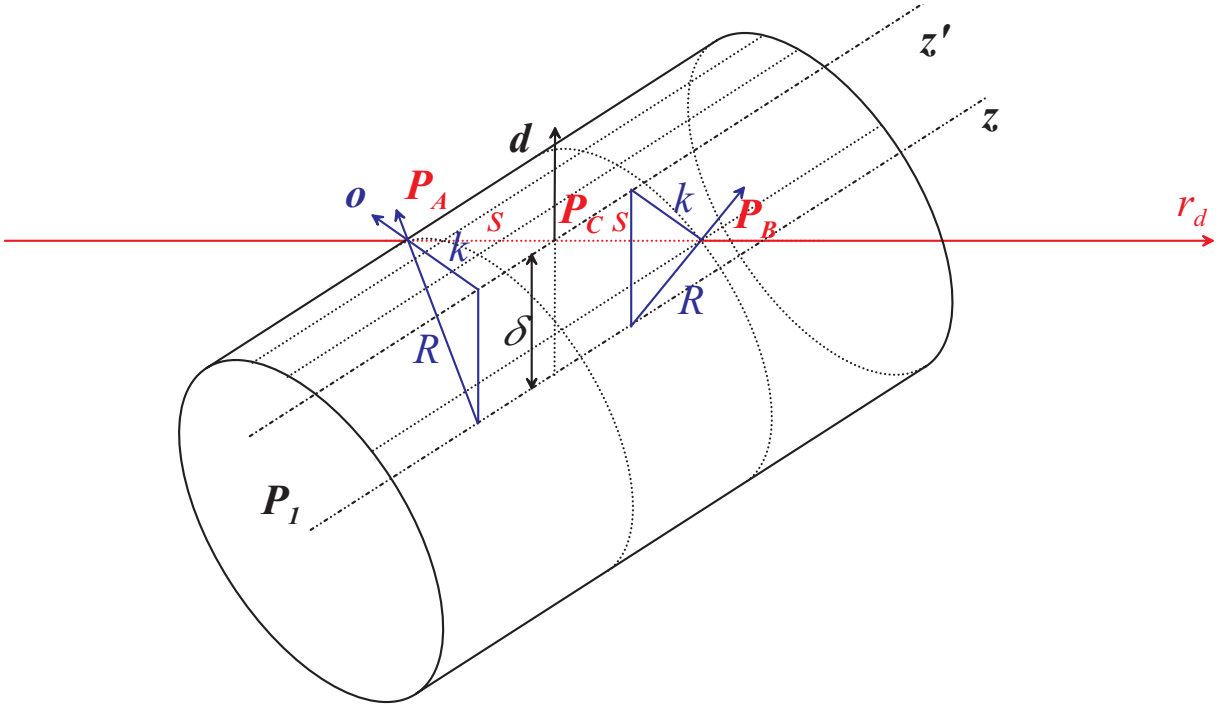


Figure 4.4: Ray-cylinder intersection.

The advantage of this method is that δ can be quickly computed and the configuration (zero-, one- or two-intersection point(s)) can easily be identified. If δ is higher than the radius R, the intersection test is early stopped. The complete intersection computation requires ten dot products and two cross products.

4.2.9.3 Surface normal vector

Once the intersection point has been computed, the local normal defined by the angle α can be obtained thanks to the following relation:

$$\overrightarrow{n} = \cos(\alpha)\overrightarrow{x} + \sin(\alpha)\overrightarrow{y} \tag{4.40}$$

This normal vector will be necessary for the re-emission of the incoming ray. It will also be used for the relations between the finite element mesh and the geometrical primitive, if the geometrical method is used (see the next Section).

4.2.10 Ray-sphere intersection

Now, the case of the sphere is considered. Although this problem can be addressed in Cartesian coordinates, by introducing the expression of the ray (4.3) into the equation of the sphere, a more efficient solution is presented here [Hai89]. In [Hai89], the author compares its solution to the Cartesian formulation and demonstrates that the number of mathematical operations is reduced, allowing to save computational effort.

4.2.10.1 Equation of the sphere

The sphere is defined by its center, noted $\overrightarrow{P_1}$, and the radius R. A vector $\overrightarrow{z}$ can also be defined in the direction joining the poles. A point $\overrightarrow{P}$ on the sphere can be identified by a longitude $\alpha \in [0, 2\pi[$ and a height $h \in [-R, R]$. An additional point $\overrightarrow{P_3}$ is defined as a reference for the longitude angles. The unit vector $\overrightarrow{x}$ corresponds to the vector $\overrightarrow{P_1P_3}$ and $\overrightarrow{y}$ completes the local coordinates system. The point $\overrightarrow{P}$ is defined by the following relation:

$$\overrightarrow{P} = \overrightarrow{P_1} + \sqrt{R^2 - h^2}\cos(\alpha)\overrightarrow{x} + \sqrt{R^2 - h^2}\sin(\alpha)\overrightarrow{y} + h\overrightarrow{z} \tag{4.41}$$

Another expression can be obtained by defining the point $\overrightarrow{P}$ through its longitude and latitude θ:

$$\overrightarrow{P} = \overrightarrow{P_1} + R\cos(\alpha)\cos(\theta)\overrightarrow{x} + R\sin(\alpha)\cos(\theta)\overrightarrow{y} + R\sin(\theta)\overrightarrow{z} \tag{4.42}$$

4.2.10.2 Intersection with the sphere

The first step consists in computing the distance between the center of the sphere and the origin of the ray, in order to determine if the origin is inside or outside of the sphere. Let $\overrightarrow{P_1P_R}$ be the vector from the center to the origin and d_{1R}^2 the square of the distance between them.

The second step consists in computing the point $\overrightarrow{P_C}$ of the ray where the ray is the closest to the center. The notations are given in Figure 4.5. The corresponding distance t is given by the relation (4.43):

$$t_C = -\overrightarrow{P_1P_R} \cdot \overrightarrow{r_d} \tag{4.43}$$

If the origin of the ray is outside of the sphere and if t_C is negative, the ray does not point towards the sphere and no intersection will be computed. The computation can be stopped.

The third step consists in computing the distance between $\overrightarrow{P_C}$ and the center of the sphere. The following relations are used. The parameters are defined in Figure 4.6.

$$t_{HC}^2 = R^2 - D^2 \tag{4.44}$$
$$D^2 = d_{1R}^2 - t_C^2 \tag{4.45}$$
$$t_{HC}^2 = R^2 - d_{1R}^2 + t_C^2 \tag{4.46}$$

If t_{HC}^2 is negative, no intersection occurs. If it is equal to zero, the ray is tangent to the sphere and only one intersection is possible. If t_{HC}^2 is positive, two intersections points are obtained, given by the following relations:

$$\begin{cases} t_A = t_C - t_{HC} \\ t_B = t_C + t_{HC} \end{cases} \tag{4.47}$$

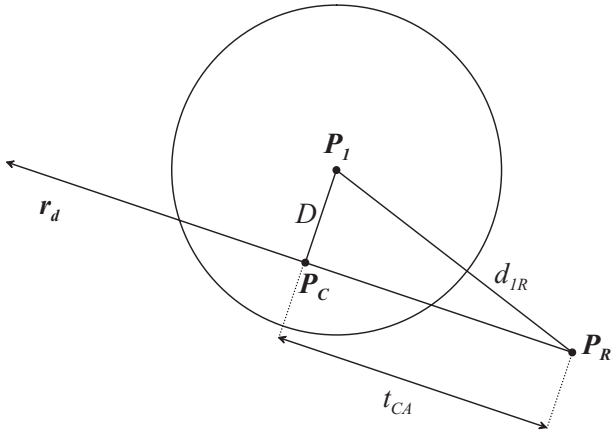


Figure 4.5: Ray-sphere intersection.

This method can be compared with the classical technique, where the components of the ray are introduced into the equation of the sphere [Hai89]. The method presented in this Section requires less computation. Moreover, we can earlier determine when a ray misses the sphere and save computation time by stopping the intersection process.

Once the intersection point has been computed, the corresponding angle α and height h can be determined, thanks to the following relations:

$$\begin{cases} \alpha & = & \arctan\left(\frac{\overrightarrow{P_1P}\cdot\overrightarrow{y}}{\overrightarrow{P_1P}\cdot\overrightarrow{x}}\right) \\ h & = & \overrightarrow{P_1P}\cdot\overrightarrow{z} \end{cases} \tag{4.48}$$

If the second expression of the sphere (4.42) is used, the height h is replaced by the angle θ, given by the following relation:

$$\theta = \arcsin\left(\frac{\overrightarrow{P_1P}\cdot\overrightarrow{z}}{R}\right) \tag{4.49}$$

4.2.10.3 Surface normal vector

In the case of a sphere, the local normal is the vector linking the center of the sphere to the intersection point. A division by the radius yields a normalized vector.

$$\overrightarrow{n} = \frac{\overrightarrow{P_1P}}{R} \tag{4.50}$$

4.2.11 Ray-cone intersection

The last implemented primitive will now be presented: the right circular cone.

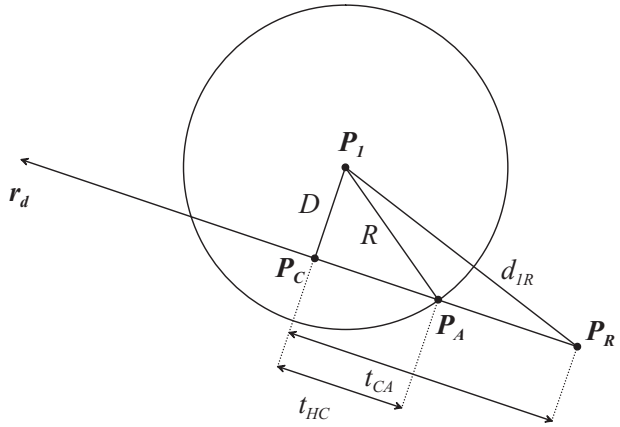


Figure 4.6: Ray-sphere intersection.

4.2.11.1 Equation of the cone

A cone is defined by a point $\overrightarrow{P_1}$ at the center of its circular basis, the radius R of its basis and its height H. The axis of revolution is identified by the vector $\overrightarrow{z}$; the origin of the angles α is given by an additional vector $\overrightarrow{x}$ and the vector $\overrightarrow{y}$ completes the local coordinate system. Here only non-truncated cones are considered. A point $\overrightarrow{P}$ on the cone is given by the following relation:

$$\overrightarrow{P} = \overrightarrow{P_1} + (1-u)R\left(\cos(\alpha)\overrightarrow{x} + \sin(\alpha)\overrightarrow{y}\right) + hu\overrightarrow{z} \tag{4.51}$$

where the parameter u is defined between 0 and 1, the angle α between $-\pi$ and π.

4.2.11.2 Intersection with the cone

This method has been presented in reference [Doc04]. An optimized version can be found in [CK95]. It consists in solving the intersection as a set of three equations with three unknowns. The parameters u and α of the cone are eliminated in order to obtain the distance from the origin of the ray to the cone. It yields:

$$\overrightarrow{P} - \overrightarrow{P_1} = \overrightarrow{P_R} + t\overrightarrow{r_d} - \overrightarrow{P_1} \tag{4.52}$$

$$= \overrightarrow{P_1P_R} + t\overrightarrow{r_d} \tag{4.53}$$

The elimination of u and α yields the following relations:

$$\left(\overrightarrow{P_1P}\cdot\overrightarrow{x}\right)^2+\left(\overrightarrow{P_1P}\cdot\overrightarrow{y}\right)^2 = R^2\left(1-\frac{\overrightarrow{P_1P}\cdot\overrightarrow{z}}{h}\right)^2 \tag{4.54}$$

$$\left(\underbrace{\overrightarrow{P_1P_R}\cdot\overrightarrow{x}}_{a}+t\underbrace{\overrightarrow{r_d}\cdot\overrightarrow{x}}_{m}\right)^2 + \left(\underbrace{\overrightarrow{P_1P_R}\cdot\overrightarrow{y}}_{b}+t\underbrace{\overrightarrow{r_d}\cdot\overrightarrow{y}}_{n}\right)^2$$

$$= R^2\left(\underbrace{1-\frac{\overrightarrow{P_1P_R}\cdot\overrightarrow{z}}{h}}_{c}-t\underbrace{\overrightarrow{r_d}\cdot\overrightarrow{z}}_{o}\right)^2 \tag{4.55}$$

The resolution of this last equation yields two solutions for the distance t. Only the nearest intersection point is kept. Then the height h of the point is computed thanks to the dot product $\overrightarrow{P_1P}\cdot\overrightarrow{z}$. It is necessary to test this value and to check it is comprised between 0 and H. The angle α is obtained from the x- and y-components of the vector $\overrightarrow{P_1P}$.

4.2.11.3 Surface normal vector

Once the intersection point has been computed, the local normal vector can be computed by the following expression, which needs to be normalized:

$$\overrightarrow{n}\equiv\cos(\alpha)\,\overrightarrow{x}+\sin(\alpha)\,\overrightarrow{y}+\frac{R}{h}\overrightarrow{z} \tag{4.56}$$

4.2.12 Conclusions

In this Section, we have presented the ray-surfaces intersection equations for the geometrical primitives used in thermal software for space applications. We are now able to compute the intersection points, as well as the $\{u, v\}$ coordinates of these points onto the geometrical primitive. These coordinates will be used in the geometrical method, presented in the following Section.

4.3 Combining geometry and finite elements

4.3.1 Introduction

In this Section, an innovative method designed in order to accelerate the ray tracing process is presented. For reasons detailed in Section 4.3.2, a finite element mesh is combined with the geometrical definition of the model. The so-called geometrical method is combined with uniform spatial subdivision (USD), resulting in a two-level acceleration technique. The first acceleration level, presented in Section 4.3.4, is based on the association of finite elements with the geometrical primitives. In Section 4.3.5, the implemented primitives are

listed and some difficulties are detailed. In Section 4.3.6, the acceleration achieved by the geometrical method is characterized. The second acceleration level is the object of the next Section.

4.3.2 Motivation

In this Section, some aspects of thermal computation for space applications are rapidly recalled, as they strongly impact the choice and development of our acceleration method.

4.3.2.1 Use of finite elements

In the field of heat transfer for space applications, a strong will to use a finite element formulation, based on a finite element mesh, is observed. The reasons are multiple [CHSP95].

1. This could allow us to consider non-isothermal patches. Isothermal patches is a common hypothesis in thermal software used for space applications. With a finite element formulation, the temperature field can be described by polynomial functions of arbitrary degree, called *shape functions*. This yields more accurate results for a few amount of additional degrees of freedom.

2. The generation of conductive links can be efficiently performed with a finite element mesh. This is usually not the case with the nodal formulation used in space thermal software.

3. The conductive component of heat transfer can smoothly be integrated with the radiative one, allowing the resolution of the complete thermal problem [CPF89].

4. With the finite element method, more complex geometries can be considered with a reduced amount of additional degrees of freedom than with the thermal lumped parameter method.

Different techniques have been designed in order to obtain a triangulation from different surfaces. In [EPO91], a method is presented for the triangulation of objects limited by numerous planes. In this thesis, we assume that a finite element can be associated with any geometrical primitive.

4.3.2.2 Stochastic ray tracing

To model the radiative component of heat transfer, including diffuse, specular and glossy reflections, the ray tracing method seems to be a good candidate to compute the interactions between the surfaces of the tri-dimensional model. It is a simple, reliable method; it can be easily implemented and can handle a large number of physical phenomena.

The drawbacks of the stochastic ray tracing are its slow convergence and the difficulty to identify the good candidates for intersection. The convergence can be improved by using

adapted ray tracing methods, such as the hemisphere method. The second difficulty has been discussed in Section 4.1 and can be solved if an acceleration technique is coded.

4.3.2.3 Use of the geometry

If finite elements are adapted for the computation of the conductive component of heat transfer, the geometrical model is more interesting for ray tracing processes. Indeed, a finite element mesh is based on a polynomial parametric approximation of the real geometry. It can induce errors in the ray tracing process, if curved surfaces are present in the model; this error is more significant in case of specular reflectors. For example, in the case of a cylinder, if the real curvature of the surface is considered (see Figure 4.7–a), the local normal can be exactly obtained, by means of analytical formulas, and varies continuously through the surface. If the surface is facetted (for example with linear elements as in Figure 4.7–b), the local normal is approximated and supposed to be constant piecewise. The corresponding error is a function of the geometrical error; it decreases when the resolution of the mesh increases.

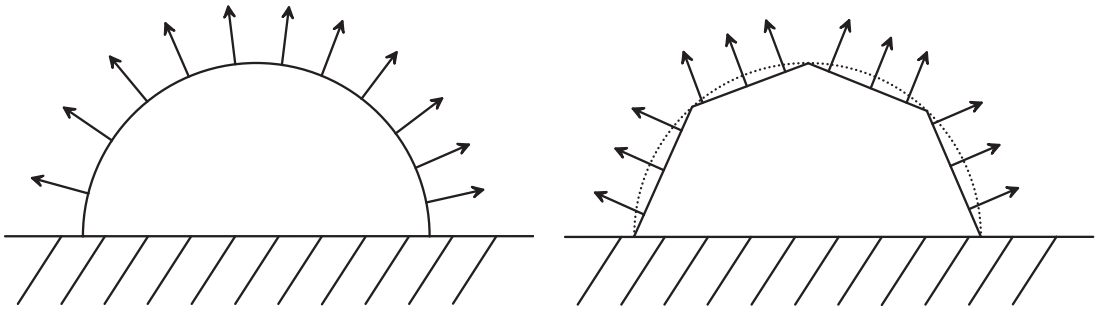


Figure 4.7: Use of the geometry – a) curved surface – b) linear approximation.

4.3.2.4 Discussion

Given these reasons, the interest to combine the CAD model with a finite element mesh appears clearly. On one hand, a method adapted to ray tracing and which can automatically generate the conductive links is obtained. On the other hand, a new ray tracing acceleration technique is designed, yielding a reduction of the number of elements candidate for intersection. This combination yields the two-level geometrical definition used in our technique.

4.3.3 Model of the spacecraft

In our acceleration method, the geometrical model of the spacecraft (satellite, space probe...) is the combination of geometrical primitives, such as spheres, cones, cylinders... In Fi-

gure 4.8, the considered satellite is CoRot[4]. The primitives are combined to obtain the model. This is the first level of the geometrical definition. Then, each geometrical primitive is meshed using finite elements, taking into account possible cutting operations (see next remark). For each geometrical level, a dedicated acceleration technique has been designed.

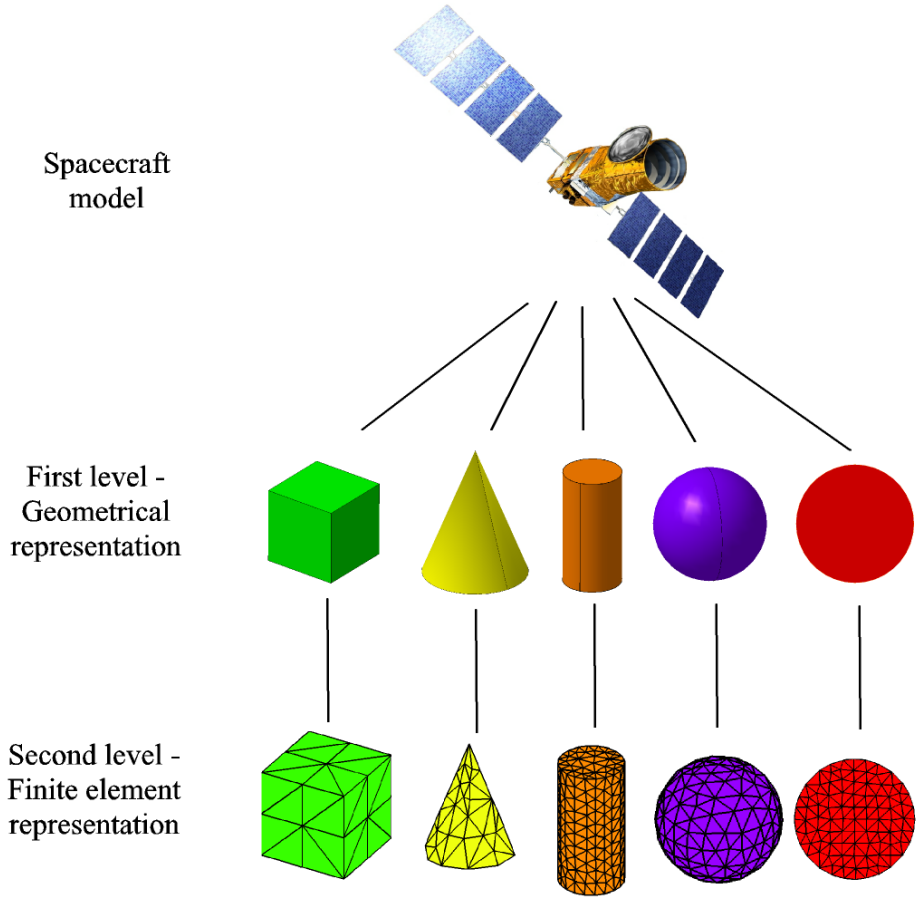


Figure 4.8: Two-level geometrical definition of a model.

Remark: in our implementation, no cutting operation is directly computed; the intersection of two primitives is not analytically computed. Instead, the finite element mesh is used to limit the portion of the geometrical primitive which is eventually cut by another shell. This results in an approximation of complex intersections with straight lines, if elements of the first degree are used. To reduce this error, triangles of the second degree have been introduced in our implementation (see Section 4.2.5). Figure 4.9–a represents a square cut by a cylinder. In our implementation, the geometrical primitive is not cut

[4]CoRot is a scientific mission leaded by CNES. Its mission is to study the rotation of stars and to identify exoplanets, thanks to photometry. I chose this example because CoRot has been the first space mission on which I worked when I was undergraduate. More information on the mission can be found on website http://smsc.cnes.fr/COROT/Fr/

(the square is still visible in the hole, in Figure 4.9–b); the cut surface is approximated by triangles. This results in an approximation of the intersection contour.

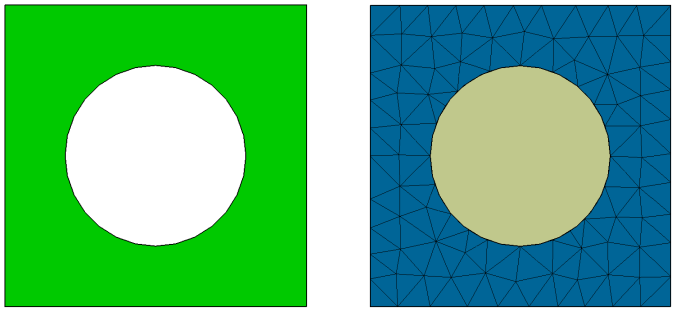


Figure 4.9: Approximation of the intersection – a) cut surface – b) linear approximation.

4.3.4 First acceleration level - geometrical method

The geometrical method is the combination of four successive steps presented in the next subsections. The final objective is to simplify the intersection of the rays with the finite elements. This method can be used to compute view factors, extended view factors, radiative exchange factors, direct and absorbed heat fluxes.

4.3.4.1 Structured mesh

The surfaces that are commonly implemented in thermal software for space applications are planar surfaces, like triangles, quadrangles and discs, and quadrics, like spheres, cones and cylinders. These surfaces can be defined by two parameters, noted u and v, varying on the domain of definition $\mathcal{D} \equiv [0,1]^2$. A point $\overrightarrow{P}$ on a surface $\mathcal{S}$ can then be obtained by using a parametrization ψ of $\mathcal{S}$:

$$\overrightarrow{P} \equiv \psi(u,v) \qquad \{u,v\} \in [0,1]^2 \tag{4.57}$$

A structured mesh of the surface $\mathcal{S}$ can be obtained by fixing one parameter and varying the other. Two families of iso-parametric curves can be generated, given by the relation (4.58). The curves $\mathcal{C}_u$ are obtained by fixing the parameter $u = \xi$ while the curves $\mathcal{C}_v$ correspond to the parameter $v = \eta$:

$$\begin{cases} \mathcal{C}_u &= \psi(\xi, v) \\ \mathcal{C}_v &= \psi(u, \eta) \end{cases} \tag{4.58}$$

The resulting mesh is composed of cells. For example, in the case of a sphere, the longitude ϕ and the latitude θ can be chosen in order to generate the surface. The parametrization is given by the relation (4.59), where R is the radius of the sphere. The main axis in

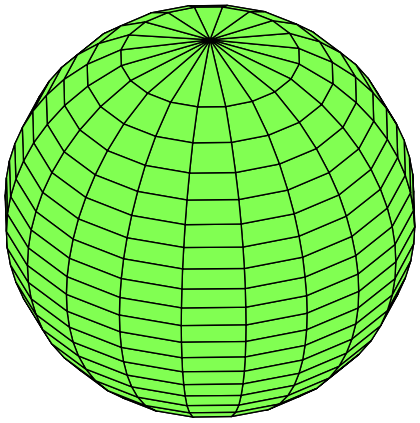


Figure 4.10: Structured mesh of the sphere.

this parametrization is the z-axis (see Figure 4.10):

$$\overrightarrow{P} = R \begin{bmatrix} \cos\left(\pi\frac{2u-1}{2}\right)\cos\left(2\pi v\right) \\ \cos\left(\pi\frac{2u-1}{2}\right)\sin\left(2\pi v\right) \\ \sin\left(\pi\frac{2u-1}{2}\right) \end{bmatrix} \tag{4.59}$$

Figure 4.10 represents the structured mesh of a sphere where all the cells are characterized by the same area. The curves $\mathcal{C}_\theta$ and $\mathcal{C}_\phi$ are defined by the following values of θ and ϕ:

$$\begin{cases} \theta = \arccos(2u-1) & \text{where} \quad u = 0, \frac{1}{n}, \cdots \frac{n-1}{n}, 1 \\ \phi = \pi(2v-1) & \text{where} \quad v = 0, \frac{1}{n}, \cdots \frac{n-1}{n}, 1 \end{cases} \tag{4.60}$$

4.3.4.2 Finite element mesh

In the second step, the surface is meshed into planar triangles and/or quadrangles, resulting in a surface finite element mesh. In this work, the mesh modules of Samcef and Catia have been used.

4.3.4.3 Combination of the two meshes

In the third step, the surface finite elements are associated with the geometrical cells obtained during the first step. For a given cell, all the finite elements that cover it, partially or completely, have to be identified. This step is performed by considering the projection of the finite elements onto the geometrical primitive. This projection is bounded by four lines $\mathcal{C}_{u,v}$. This contour is called the *envelope* of the projection. The geometrical cells covered by this envelope are identified. Finally, these relations are inverted. We are now able to identify the finite elements covered by each geometrical cell. This relatively complex operation is only performed once, during the preprocessing phase. This is illustrated here in the case of a sphere.

Projection onto the geometrical primitive In the case of a sphere, the straight lines of the polygonal elements (if linear) are projected following great circle arcs. Figure 4.11 represents the projection of a triangle in the parametric space $\{\theta, \phi\}$. Generally, the projection of finite elements yields complex curves in the parametric space $\{u, v\}$.

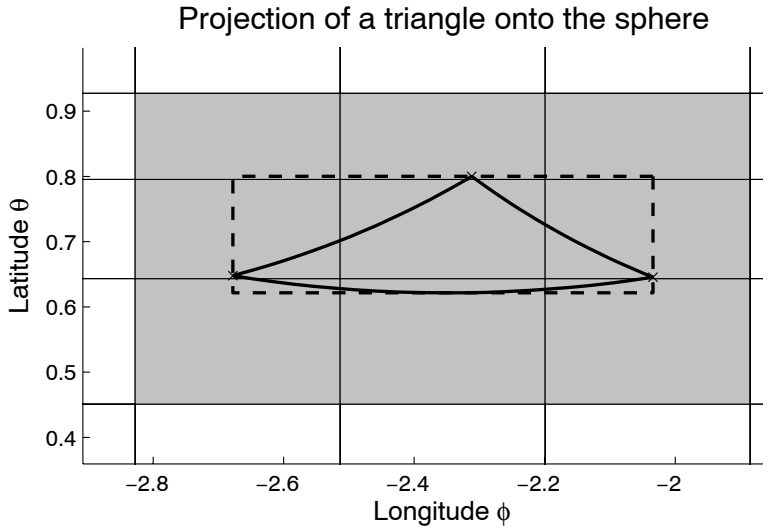


Figure 4.11: Projection of a triangle in the parametric space of the sphere $\{\theta, \phi\}$.

Computation of the envelope The next step consists in computing the smallest envelope bounding the projection of the finite element. In the parametric space, the envelope is a rectangle limited by the values $\{u_{\min}, u_{\max}\} \times \{v_{\min}, v_{\max}\}$ of the current finite element. In Figure 4.11, the envelope is represented by four dashed lines. A spherical arc is not necessarily bounded by the latitudes of its extremities. In the presented case, the extremum value of the latitude, noted θ_m, must be identified. It can be shown that this latitude is given by the relation:

$$\theta_m = \arccos(n_\Pi(3)) \tag{4.61}$$

where $n_\Pi(3)$ is the z-component of the unit normal to the plane Π defined by the current edge and the center of the sphere.

Coverage of the geometrical cells The envelope of the current finite element has been computed. The geometrical cells that can cover it can now be determined by a simple test based on the limits $\{u, v\}$ of each geometrical cell. In our implementation, a cell is assumed to be covered by a finite element if it is covered by the envelope of this element, for the sake of simplicity. In Figure 4.11, the covering cells are represented in gray. Due to the last remark, the upper-left and upper-right cells, which do not cover the element, are nevertheless associated with it. This operation is performed once for each finite element.

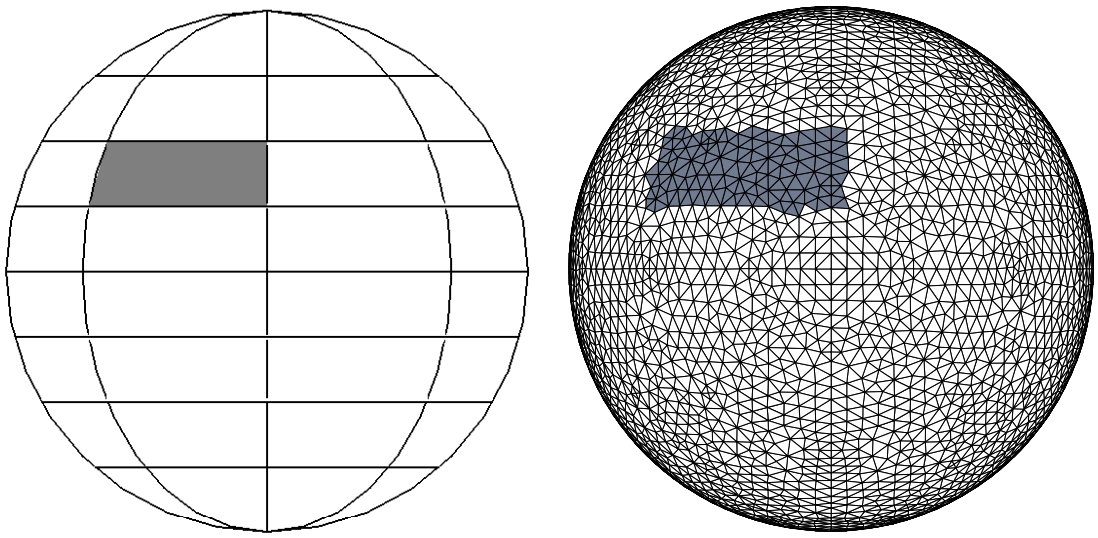


Figure 4.12: Identification of a geometrical cell through the covering finite elements – a) geometrical cells – b) finite element mesh.

The previous relations are then inverted in order to associate each geometrical cell with the finite elements that cover it. In Figure 4.12, a decomposition of a sphere into geometrical cells and a fine finite element mesh of the sphere can be observed. A geometrical cell has been selected(see Figure 4.12–a); the corresponding elements have been displayed in gray (see Figure 4.12–b). The shape of the geometrical cell can easily be distinguished from the rest of the finite element mesh.

4.3.4.4 Ray tracing process

The last step is an acceleration of the ray tracing. Each ray is tested for intersection with the geometrical surface. If the intersection test is positive, the parametric coordinates $\{u, v\}$ of the impact are computed and the corresponding geometrical cell is identified. Then the impact point is projected onto the finite element mesh. Only the finite elements that cover the identified cell have to be tested. The number of tested elements is then considerably reduced.

4.3.5 Mathematical primitives

In the geometrical method, the primitives used in Esarad [Doc04] and Thermica [Doc03] have been implemented, except for the paraboloid[5]. In this Section, the different primitives will be presented, the difficulties will be listed, and the adopted process will be explained.

[5]The implementation of the paraboloid does not implies any serious difficulty. The paraboloid has not be implemented because no test case involving a paraboloid was available.

4.3.5.1 Sphere

The sphere has been presented in Section 4.3.4.3. The main difficulty consists in computing the intermediary latitudes θ_m and detecting the configurations where this computation is necessary. This is the purpose of this Section.

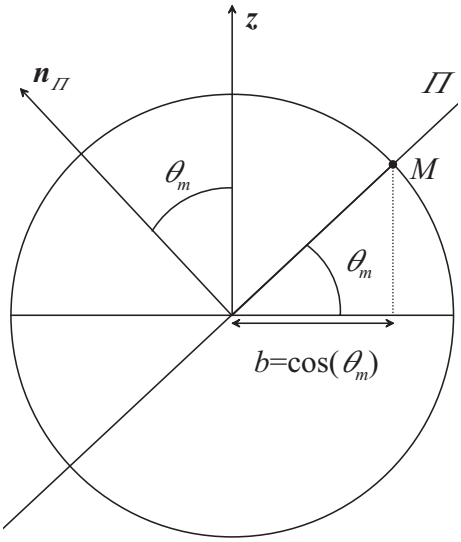


Figure 4.13: Representation of the intermediary latitude θ_m.

A spherical arc is defined by its two extremities and the center of the sphere. If they are not aligned, these three points define a plan Π. Let $\overrightarrow{n_\Pi}$ be the unit vector normal to the plane. The angle θ_m is the angle between z-axis and $\overrightarrow{n}_\Pi$. Figure 4.13 represents the sphere cut by the plane Θ, containing z and $\overrightarrow{n_\Pi}$[6]. The extremum latitude will be encountered in the plane Θ [Bec03], at point M.

The next problem is to identify if this configuration is encountered. First the unit normal $\overrightarrow{n_\Pi}$ is computed. If the z-component is equal to 0, the arc is in fact a meridian; the latitude interval is defined by the extremities, no additional computation is required. If the z-component is non-zero, the arc is projected in the equatorial x-y plane. The projection is an elliptical arc, following internal report [Vue05], where a similar development has been performed for the unit sphere method[7]. The elliptical arc is characterized by a semi-major axis $a = R = 1$ and a semi-minor axis $b = Rn_\Pi(3) = \cos(\theta_m)$ (see figure 4.13). The direction of the semi-minor axis is given by the projection of the unit normal $\overrightarrow{n_\Pi}$ in the

[6]Because it contains the unit vector $\overrightarrow{n_\Pi}$, the plane Θ is perpendicular to Π. The trace of Π in Θ (the intersection between Π and Θ) is given by the straight line containing point M in Figure 4.13.

[7]The unit sphere method refers to an analytical method implemented in Samcef (command .R3D) to compute diffuse view factors when no obstacle is present. This method is based on the Nusselt's Analogy (see Section 3.1.3).

equatorial plane[8]:

$$\overrightarrow{n_\Pi}' = \begin{pmatrix} n_\Pi(1) \\ n_\Pi(2) \\ 0 \end{pmatrix} \tag{4.62}$$

An intermediate latitude θ_m is encountered if and only if the elliptical arc passes through the semi-minor axis, identified by the longitude ϕ_Π and given by the relation:

$$\phi_\Pi = \arctan\left(\frac{n_\Pi(2)}{n_\Pi(1)}\right) \tag{4.63}$$

ϕ_Π is defined between $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. An intersection test must also be computed for the angle $\phi'_\Pi = \phi_\Pi + \pi$ (which is the extension of the semi-minor axis).

As it can be seen in Figure 4.13, the cosinus of the latitude θ_m is equal to the dot product of the z-axis with the normal vector $\overrightarrow{n_\Pi}$:

$$\cos(\theta_m) = \overrightarrow{z} \cdot \overrightarrow{n_\Pi} \tag{4.64}$$

$$\theta_m = \arccos\left(n_\Pi(3)\right) \tag{4.65}$$

4.3.5.2 Rectangle - quadrangle

In the case of a rectangle (or a quadrangle), the relations from the geometrical cells to the covered finite elements are straightforward. Here, the case of a rectangle is used to illustrate the acceleration of the ray-surface intersection process with the geometrical method.

Let us consider the case of the rectangle shown in Figure 4.14–a. The primitive is decomposed into geometrical cells, following the iso-parametric curves. A geometrical mesh with parameters $n_u = 10$ and $n_v = 4$ has been chosen. The surface is considered for intersection with a ray. The intersection point is identified by the red dot. The same primitive is triangulated (see Figure 4.14–b). The mesh contains 74 elements. If the basic ray tracing process is considered, each triangle must be tested for intersection with the ray. With the geometrical method, the number of elements that need to be tested can be reduced. Consider Figure 4.14–c. If the parameters $\{u, v\}$ of the intersection point are known (identified by the red point in Figure 4.14), the intersected geometrical cell can easily be identified (in green in Figure 4.14–c). Finally, only eight triangles can be candidate for intersection (in green in Figure 4.14–d).

[8]The direction of the semi-major axis can be found by projecting the unit vector tangent to the sphere at point M: $\overrightarrow{t}' = \begin{pmatrix} n_\Pi(2) \\ -n_\Pi(1) \\ 0 \end{pmatrix}$. The dot product of these two vectors is equal to zero; the two axes of an ellipse are orthogonal.

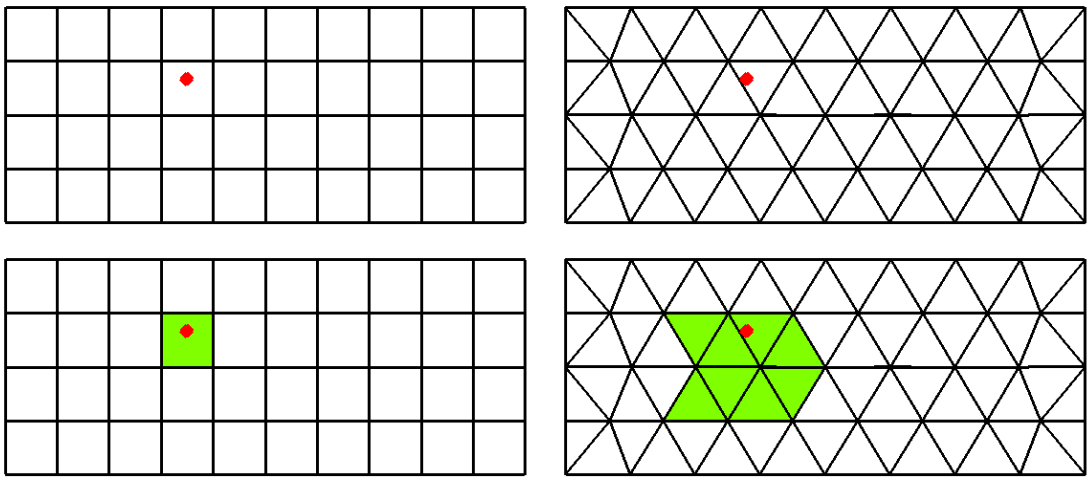


Figure 4.14: Acceleration of the ray-rectangle intersection – a) geometrical cells – b) finite element mesh – c) identification of the impacted geometrical cell – d) identification of the elements candidate for intersection.

Remark: to identify the impacted geometrical cell, the boundaries of the geometrical cell have to be tested with respect to the $\{u, v\}$ coordinates of the intersection point. Each cell can be tested, from the first one to the intersected one. The number of tests is proportional to $\frac{(n_1+1)(n_2+1)}{4}$. Another method is based on a dichotomic approach. For one dimension, the geometrical boundaries $\{i_{\min}, i_{\max}\}$ are initiated at $\{1, n_1 + 1\}$. While the boundaries are separated by more that one, the medium value $\frac{i_{\min} + i_{\max}}{2}$ of the interval is computed. This value is tested to the boundaries of the interval and one of the boundaries is replaced by the medium value, to decrease the width of the interval. Finally, when the boundaries are separated by one, the process is stopped. The number of tests with the dichotomic approach is proportional to $\sqrt{n_1 + 1}\sqrt{n_2 + 1}$.

Once the impacted geometrical cell has been identified, only the finite elements covered by this cell have to be tested (see the green triangles in Figure 4.14–d). The number of elements is equal to 8. The number of candidate elements for intersection is then considerably reduced.

Remark: This acceleration technique is a two-step process: the ray-triangle intersection tests are only performed if the ray intersects the primitive. This first step yields a reduction of the rays number that must be tested for ray-finite element intersection. The second step is based on the examination of the impacted geometrical cell, as previously explained.

4.3.5.3 Cylinder

The case of the cylinder presents no particular difficulty. The envelope of a finite element is defined by the $\{u, v\}$ coordinates of its extremities. No extremum has to be considered.

4.3.5.4 Disc

The fourth implemented primitive is the disc. While the mesh of the rectangle is based on Cartesian coordinates, the mesh of the disc is polar. The $\{u, v\}$ parameters of the disc are functions of the radius ρ and the polar angle θ. The difficulty of the disc is that the envelope of a finite element is not always defined by the $\{u, v\}$ coordinates of its vertices. The situation is similar to the case of the sphere. Here, the sensitive parameter is the radius ρ. The configurations which require the computation of an intermediary radius ρ_m can be identified thanks to the spherical analogy (see Figure 4.15).

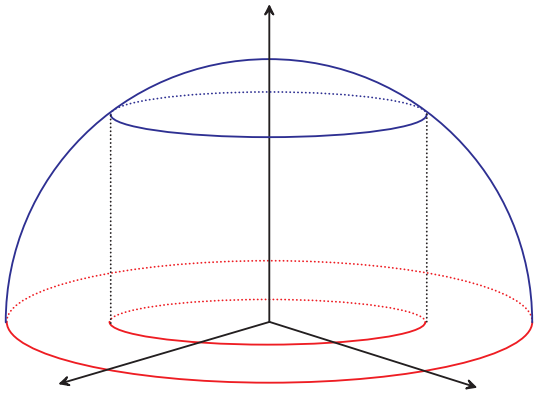


Figure 4.15: Spherical analogy - case of the disc.

A finite element mesh $\mathcal{T}$ on the disc $\mathcal{D}$ can be considered as the orthogonal projection of a triangulation $\mathcal{T}'$ on the hemisphere $\mathcal{H}$ subtended by $\mathcal{D}$. The relations developed for the sphere can be used in order to identify the critical configurations and to compute the values θ_m. Finally, the values θ_m are converted into polar radii ρ_m, using the following relation:

$$\begin{aligned} \rho_m &= R\cos(\theta_m) \\ &= Rn_{\Pi}(3) \end{aligned} \tag{4.66}$$

Remark: the equatorial plane used to establish the relations of the sphere should not be confused with the disc. The triangulation in the disc is made of straight lines while the mesh in the equatorial plane results of the two-step projection of planar triangles and is made of elliptical arcs.

4.3.5.5 Cone

The last primitive implemented in the geometrical method is the cone. The two $\{u, v\}$ parameters of the cone are functions of the polar angle θ and the height h. The cone presents the same difficulty as the sphere and the disc (*i.e.* the surfaces of revolution). The developments presented in this Section can be used for any surface of revolution, supposing that an adapted ray-surface intersection procedure is available for this surface [Kaj83].

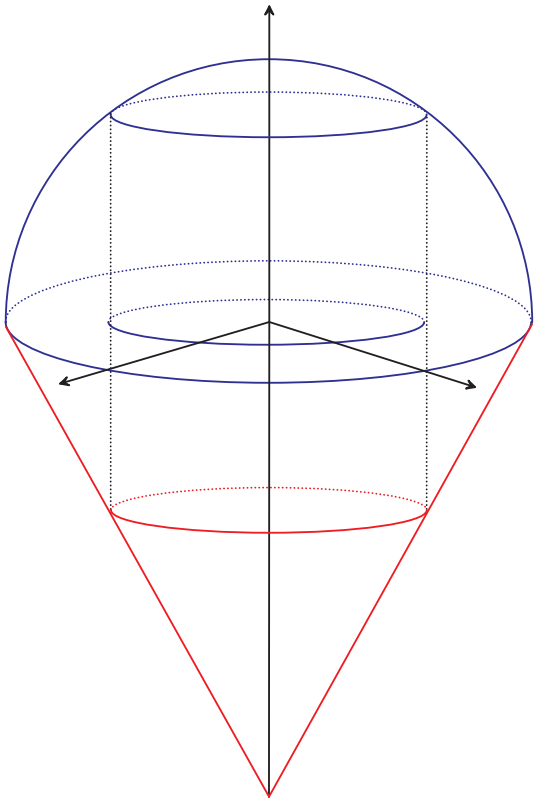


Figure 4.16: Spherical analogy - case of the cone.

In the case of a straight line projected onto the cone, the covered height interval does not necessarily correspond to the height of the two extremities. The critical configurations can be detected by using the spherical analogy (see Figure 4.16). A finite element mesh $\mathcal{T}$ on a cone can be projected down onto the disc (basis of the cone), then onto the hemisphere subtended by the basis of the cone. The relations developed in the framework of the sphere can then be used in order to identify the critical configurations and to compute the intermediary values h_m.

4.3.6 Acceleration results

As mentioned previously, the geometrical method is the combination of two processes that can be used to accelerate a ray tracing algorithm. The first acceleration consists in establishing the intersection of the ray with the geometrical primitive. If the test is negative, the computation is finished, the finite elements do not need to be tested for intersection. This method is similar to the acceleration by *bounding boxes*[9], presented in [Gla02]. If the test is positive, the identification of the impacted cell is used to reduce the number of elements that must be tested for intersection. This second acceleration step is similar to *uniform spatial subdivision* (USD) method, also presented in [Gla02]. The main difference with respect to USD is that the decomposed domain is the 2D-surface and not the 3D-space occupied by the geometrical model.

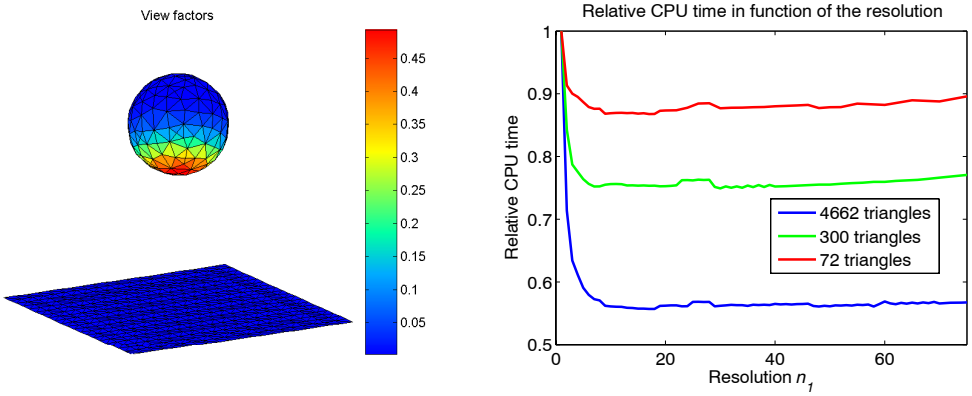


Figure 4.17: Acceleration of the view factor computation – a) view factors – b) evolution of the CPU time.

In order to characterize the acceleration of the geometrical method, the case of a sphere centered vertically above a square has been considered. The view factors from the sphere to the square are displayed in Figure 4.17–a[10]. This result is in good agreement with the analytical solution that can be found in the literature [How82, Won76].

For the square, three finite element meshes of increasing quality (coarse, normal and fine) are considered. Then the square is geometrically meshed following the u- and v-

[9]Each object of a 3-D geometrical model is bounded by a parallelepiped parallel to the main axes x, y and z. The parallelepiped is limited by $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}]$, defined by the extremal values experienced by the object. The acceleration consists in testing a ray for intersection with the bounding box. If the test is negative, the computation is stopped, saving CPU (computational effort).

[10]The displayed quantity is the cumulated view factor, *i.e.* for each line, the sum on the columns $\sum_{j=1}^{N} F_{i-j}$. This quantity is the percentage of the unit disc covered by the surfaces visible from i, in case of diffuse reflection; the complementary quantity $1 - \sum_{j=1}^{N} F_{i-j}$ is the percentage of the unit disc which can see the deep space.

directions. n_1 denotes the resolution in the first direction and the resolution in the other direction is assumed to be identical. When the resolution is equal to 1, the primitive is made of one cell which contains all the elements. For this resolution, the second acceleration process is disabled. The corresponding CPU time is chosen as the reference value. Then, for each of the three finite element meshes, the measured CPU time is divided by the reference time in order to obtain comparable curves for the three meshes. Figure 4.17–b represents the evolution of the three CPU times (one curve for each mesh) in function of the resolution n_1. It can be observed that the gain in terms of CPU time increases with a fine mesh. For low resolutions, the CPU time decreases rapidly when the resolution increases. The obtained gain remains constant for resolutions between 20 and 60. The resolution must be considerably increased to observe an increase of the CPU time. This loss of efficiency is due to the fact that the gain obtained by refining the cells can not compensate the computation charge.

4.3.7 Discussion

The method presented in this Section consists in associating a detailed finite element mesh with the geometrical primitives which compose the 3D model. It results in a detailed description of the thermal situation and it can be used for the detailed analysis of a subsystem aboard a satellite. Another approach has been developed by Thales Alenia Space, resulting in a tool called `THERM3D` [Dud06, BDB08]. This approach can be considered dual of the geometrical method developed in this thesis. The method starts from a detailed finite element mesh to compute the conductive links; a model reduction is used to transfer the resulting links to the system level where a ray tracing process can be used to compute the radiative links. This approach is more adapted for the thermal computation, at system-level.

A combination of these two approaches in a dedicated software can be considered for future developments.

4.4 Uniform spatial subdivision

If the geometrical method is used without any other acceleration technique, the method is not optimal. Remember that a ray tracing is the combination of three processes: emission of the ray, evolution throughout the model (selection of the candidate surfaces for intersection) and intersection of the ray. The geometrical method is dedicated to the third process, while it partially optimizes the second one by reducing the number of candidate elements on a particular surface.

If few primitives are considered, this implementation is sufficient. If the geometrical model is more complex, the selection of the primitives candidate for intersection needs to be accelerated. This can be done by bounding the primitives with rectangular boxes

and performing uniform spatial subdivision, as it is already implemented in space thermal software [Koe88]. This is the purpose of the current Section.

4.4.1 Bounding boxes

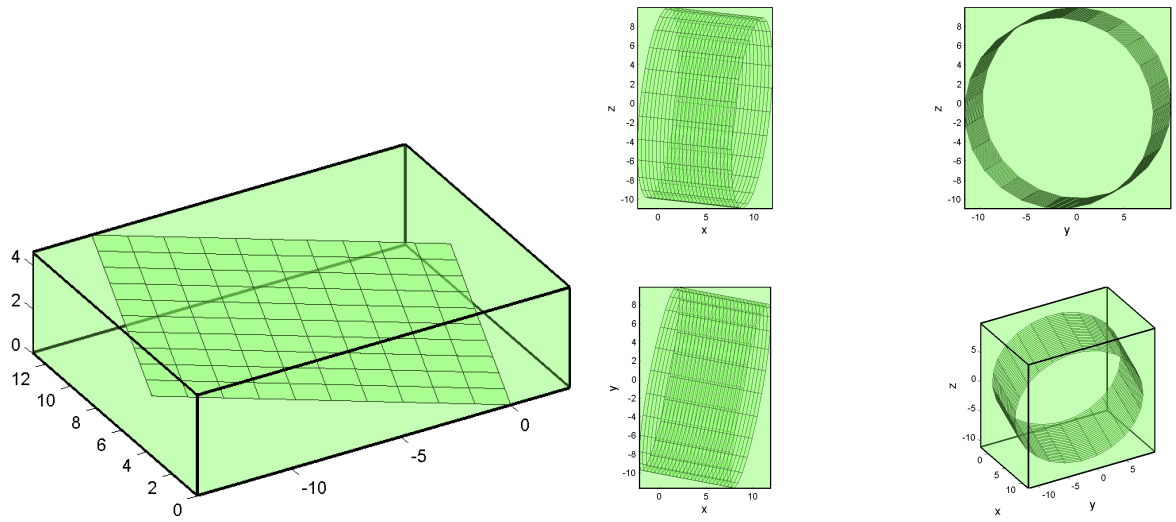


Figure 4.18: Bounding boxes – a) rectangle – b) cylinder.

The first step consists in computing the bounding boxes associated with the primitives. The bounding boxes are aligned along the 3D axes. This step is straightforward for the sphere and the rectangle (see Figure 4.18–a). Care must be taken in the case of the cylinder (Figure 4.18–b), the disc (Figure 4.19–a) and the cone (Figure 4.19–b). For the three last primitives, the three main projections (following y, x and z) are reported, as well as a perspective.

4.4.2 Bounding box of the model and voxels

The next step consists in computing the bounding box of the complete model itself[11]. This bounding box is partitioned into smaller volumes, called *voxels*. A voxel is a cubic, axis-aligned rectangular prism [Gla02]. The term voxel is an extension of the 2D *pixel* or *picture element* to the 3D *volume element*. During the preprocessing step, the primitives are associated with the voxels through their bounding boxes. Each voxel is associated with a list of all the surfaces that it contains.

[11]This step is common to Esarad, Thermica and Samcef.

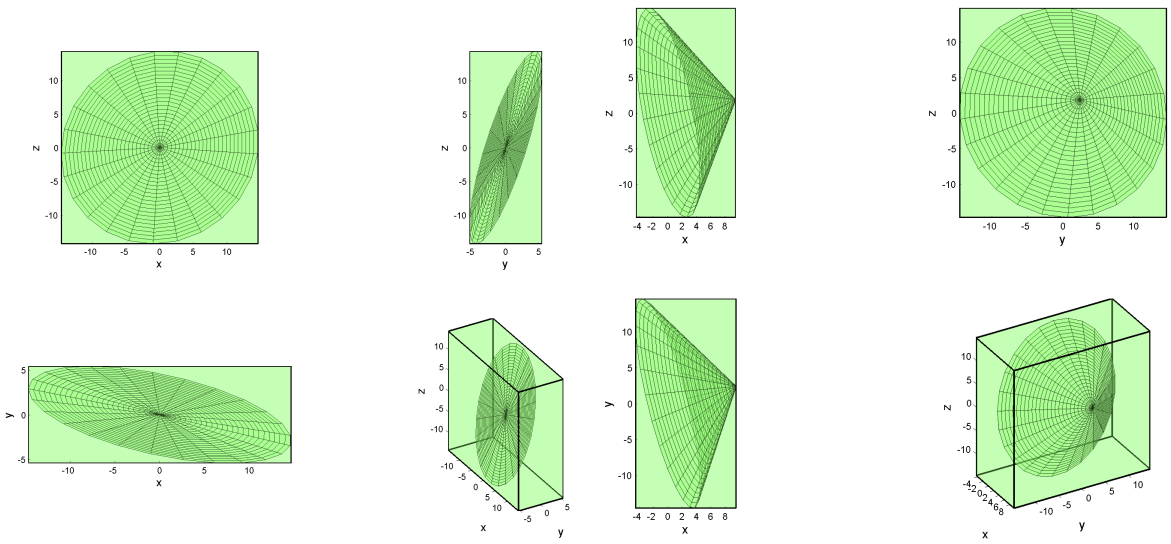


Figure 4.19: Bounding boxes – a) disc – b) cone.

4.4.3 Voxel traversal algorithm

The USD method has been implemented in our ray tracing, in several ways. The first one is directly applied on the finite elements, as it is done in Samcef; it is very efficient, with regard to the CPU time, and could be preferred to the geometrical method for parts of the model where the geometry is not critical and a finite element approximation is acceptable. The second one is applied on the primitives and is the second acceleration level of our geometrical method. The implementation of reference [AW87] has been used because it is more efficient than the procedure described in [Doc04]. In our implementation, no main direction and no axis inversion are needed; the number of mathematical operations is reduced. The traversal of the voxels is as fast as possible.

The strategy is based on two steps: initialization and incremental traversal [AW87]. The initialization phase begins by identifying the voxel in which the ray origin is found. Next, the value of t at which the ray crosses the first voxel boundary following x is determined; it is stored in a variable `tmaxX`. The same computation is performed for the $y-$ and z-directions. The next voxel will be characterized by the lowest value `tmax`. Then `tDeltaX`, `tDeltaY` and `tDeltaZ` are computed, where `tDeltaX` corresponds to the distance along the ray to achieve the next voxel in the $x-$, $y-$ and $z-$ directions. Now all the elements to propagate the ray in the voxels' network are available. The detailed algorithm can be found in [AW87]. To go to the next voxel, the algorithm requires two floating point comparisons, one floating point addition, two integer comparisons and one integer addition. This is below the requirements of the algorithm described in [Doc04].

In Figure 4.20, the emission of a ray in a voxels' network is considered. The initial voxel

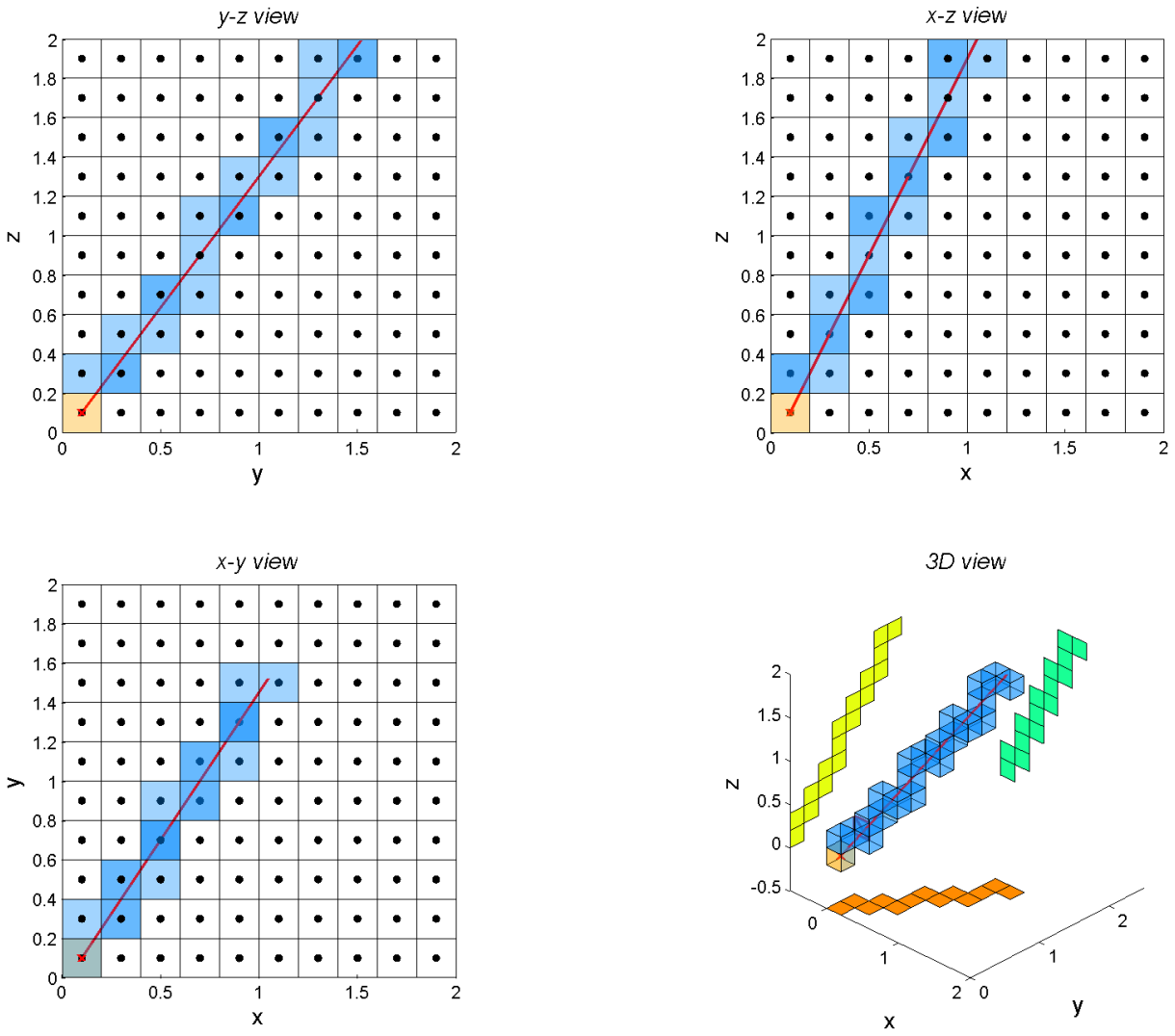


Figure 4.20: Illustration of the USD method.

is identified and drawn in orange. The ray is identified by the red line. The succession of traversed voxels is shown on the four pictures in the form of blue cells. The first three figures correspond to the projection of the 3D figure on the three main planes $x = 0$, $y = 0$ and $z = 0$. The last one is the 3D view. The traversed voxels are correctly identified. In this routine, the solution of Amanatides and Woo [AW87] has been implemented.

4.4.4 Acceleration of the ray tracing

Once a ray is emitted, the coordinates of the emission point are tested to identify the original voxel. Then all the primitives associated with this original voxel are tested. If no element of the current voxel is intersected, the ray is propagated through the model to the next voxel; then all the elements in this new voxel are considered for intersection with

the ray. This method uses the notion of proximity to accelerate the ray tracing: the first tested primitives are the nearest ones.

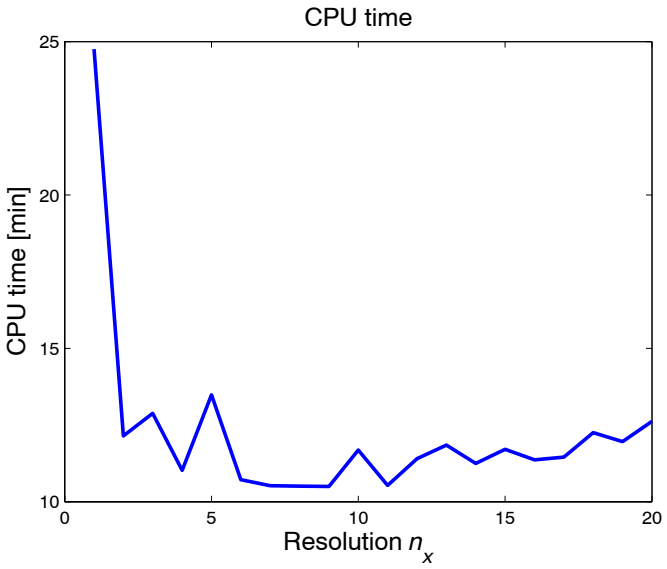


Figure 4.21: CPU time in function of the spatial resolution.

Figure 4.21 plots the reduction of the CPU time in function of the resolution n_x. The resolution in the other directions n_y and n_z are assumed to be identical to n_x. The case of a satellite has been considered (the gain of the second level of the acceleration method cannot be pointed out on a too basic model). A behaviour similar to the one of the geometrical method, first level of our acceleration method, can be observed (see Figure 4.17-b). After a strong decrease of the CPU time, the curve presents a constant plateau.

4.4.5 Finite element object

In the geometrical method, each surface is defined by a geometrical primitive. In order to generalize the method, an additional object has be defined in the form of *finite element object*, that is as a set of finite elements (quadrangles and/or triangles) which are not supported by a geometrical primitive. The ray tracing corresponding to this new object cannot benefit from the first level of acceleration. However, the second level is still active.

This extended geometrical method allows the simultaneous definition of geometrical objects and facettized ones. In Section 7.3.6, the proposed algorithm has been compared with Samcef for finite element objects (set of triangles). The geometrical definition of the model has also be considered. It appears that the ray tracing for finite element objects is faster than the ray tracing for geometrical objects. This is due to the fact that the ray-triangle intersection is an operation which is simpler than other ray-surface intersections.

This can be used in order to accelerate the ray tracing process. Surfaces which require the definition of the geometry (specular surfaces, optical subsystems) can be mixed with surfaces approximated with facets. The computation effort can then be affected to the most sensitive parts of the model.

The acceleration method presented in this Section can then be seen as a combination of the acceleration techniques used in software like Esarad for geometrical surfaces and the techniques used in finite element software for triangulations. In function of the encountered surface, the acceleration method can adapt itself to obtain an optimal acceleration.

4.5 Conclusions

This concludes the Chapter dedicated to geometrical developments used to accelerate the ray tracing process, necessary to the computation of exchange factors. Another radiative quantity requires a ray tracing process: the external heat fluxes. In the next Chapter, the computation of the solar heat flux is addressed.

Chapter 5

Computation of on-orbit heat loads

Contents

In this Chapter, the computation of the on-orbit heat fluxes is presented. A description of the three radiative heat sources (solar and albedo radiation in the visible part of the spectrum and the infra-red radiation coming from the planet) is first given.

The solar heat flux is the most complex one because it requires the computation of sharp shadows. In this Chapter, we focus on the solar heat flux and present a computation combined with the geometrical method in order to accelerate the ray tracing for heat fluxes.

5.1 External heat loads and deep space

In order to complete the thermal model, the boundary conditions have to be carefully defined; these boundary conditions are the external heat loads from the Sun and the planet, as well as the cold deep space, at a temperature of approximately $5K$.

Once the orbit and attitude of the satellite are defined, Esarad and Thermica can compute the radiative heat fluxes received by each external surface thanks to a ray tracing process.

5.1.1 Solar heat flux

The main heat source in our solar system is our star, the Sun. It is a G2-star in the spectral classification. Its diameter is equal to $1\,391\,000km$. The surface temperature is equal to $5\,780K$. The mean distance between the Sun and the Earth is equal to $149\,600\,000km$. The Sun is a stable heat source. On Earth, the main variation is due to the eccentricity of the planetary orbit, as it can be seen in Figure 5.1. The distance between the Sun and the Earth is not constant. On the summer solstice, the Earth is farther from the Sun and the received heat flux is equal to $1\,322W/m^2$. On the winter solstice, the Earth is closer and the heat flux is equal to $1\,415W/m^2$. For this reason, the summers are warmer and the winters colder in the southern hemisphere than in the northern one.

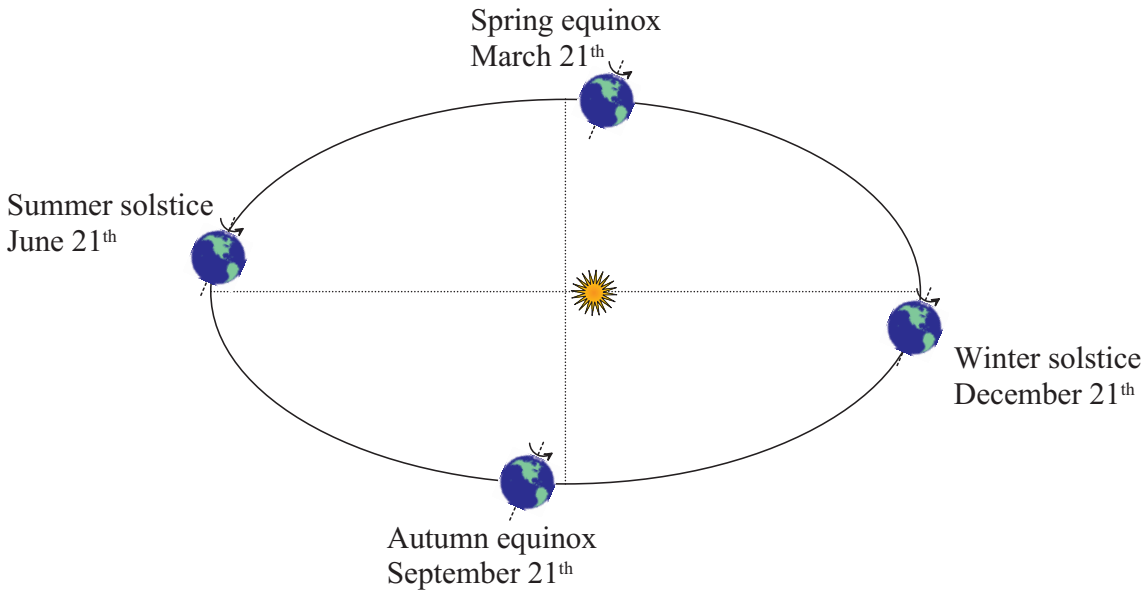


Figure 5.1: Orbit of the Earth around the Sun.

The two previous values can be obtained by computing the energetic balance between the power emitted by the Sun, considered as a black sphere at a temperature $T_\odot$=5 780K with a radius $R_\odot$ of 696 000km, and the power received by a fictitious sphere, centred on the Sun, with a radius equal to the distance Sun-Earth d. The following relations are obtained:

$$4\pi R_\odot^2 \sigma T_\odot^4 = 4\pi d^2 C_\odot \tag{5.1}$$

$$C_\odot = \frac{R_\odot^2 \sigma T_\odot^4}{d^2} \tag{5.2}$$

where $C_\odot$ is the *solar constant*. It is equal to the heat flux received by a unitary surface pointing to the Sun, at the distance d.

5.1.2 Albedo heat flux

The albedo heat flux is the flux emitted by the Sun which is reflected by a planet and its atmosphere to the surrounding space. The intensity of this flux strongly depends on the ground overflew by the spacecraft. For example, snow and clouds can reflect up to 80% of the incoming solar flux. On the other hand, forest and oceans only reflect 5 to 10% of the solar power. In order to simplify the thermal computations, the albedo coefficient a is defined as the fraction of the solar flux which is reflected by the whole planet. This albedo coefficient depends on the nature of the planetary surface, on the possible presence of an atmosphere. A common hypothesis consists in choosing a diffuse albedo reflection. In these conditions, the albedo heat flux received by a surface orbiting a planet depends on:

- the nature of the overflew ground, taken into account by the coefficient a;
- the angle β between the direction Earth-Sun and the direction Earth-spacecraft;
- the height h of the spacecraft (shortest distance from the planetary surface).

The albedo heat flux is then given by the following relation:

$$q = C_{\odot} a F_a \tag{5.3}$$

where F_a is the albedo view factor. This factor allows to model the visibility from the spacecraft to the illuminated face of the planet, taking into account the varying sunshine level across the planet. Figure 5.2 represents the evolution of the albedo view factor in function of the β-angle and the height h. Here we consider a spherical planet and an elementary surface pointing to the centre of the planet. The iso-view-factor curves never intersect. The red curve corresponds to the point-wise view factor to a sphere (of radius equal to $R_{\oplus} = 6\,378 km$). This curve covers the curve corresponding to $\beta = 0°$. This is due to the fact that the albedo view factor is attenuated by the sunshine of the visible points.

The curves corresponding to $\beta < 90°$, for low heights, an increase in height has a small impact on the view factor. When β is superior to 90°, a local maximum can be observed. For low altitudes, the satellite is in the shadow of the planet; the albedo flux is close to zero. As the height increases, the field of view is larger and can contain regions illuminated by the Sun; the albedo flux increases. In general, the albedo view factor decreases with the altitude, as the spacecraft goes away from the planet. This tendency can be observed with the red curve, which presents this decay linked to the distance.

5.1.3 Infrared planetary heat flux

The Earth (or any other planet in the case of an interplanetary mission) can be considered as a spacecraft orbiting the Sun. Its mean temperature is the result of the thermal balance

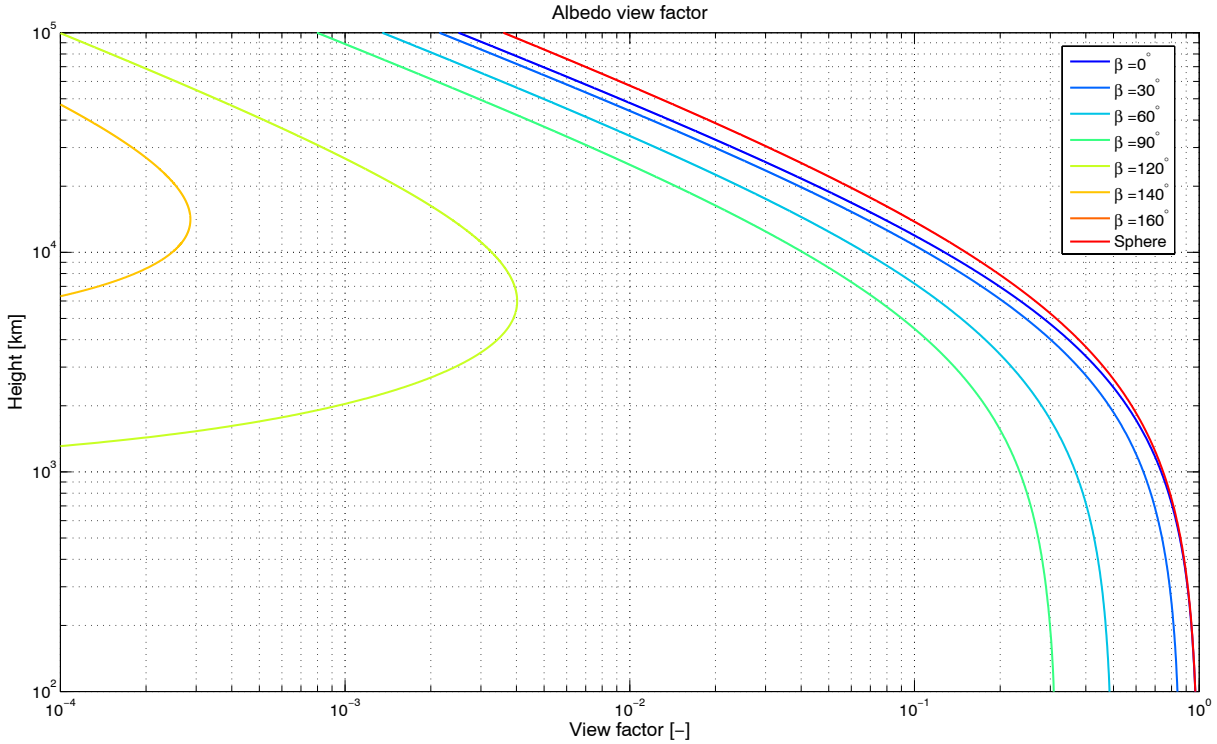


Figure 5.2: Albedo view factor in function of the β-angle and height h.

between the absorbed solar heat flux and the infrared flux emitted by the planet to the deep space.

$$C_{\odot}\,(1-a)\,\pi R_{\oplus}^2 = 4\pi R_{\oplus}^2 \sigma T_{\oplus}^2 \tag{5.4}$$

$$T_{\oplus} = \sqrt[4]{\frac{C_{\odot}\,(1-a)}{4\sigma}} \tag{5.5}$$

The parameter a corresponds to the albedo coefficient. If a mean solar constant equal to $1\,371 W/m^2$ and a mean albedo coefficient of 30% are considered, a terrestrial temperature equal to $255.05K$ is obtained. In fact, the local temperature is a complex function of the local time, the local latitude, the geographical situation, the atmospheric conditions.

5.1.4 Cold deep space

In addition to the thermal sources, the space environment contains a thermal sink which is the fossil radiation or Cosmic Background Radiation at $2.7K$. It is really homogeneous, with variations around $200\mu K$. The cold deep space is modeled by a black body at approximately $5K$, taking into account fossil radiation and stars.

This fossil radiation is the residual trace of the high temperatures experienced during the first moments of the Universe. Discovered in 1965 by the astrophysicians Arno Penzias

and Robert Wilson (Nobel Prize in Physics), this residual radiation has been the first proof confirming the Big Bang theory.

Now the external heat sources have been detailed, we explain the method based on ray tracing that we have developed in the framework of this thesis.

5.2 Shadow and illumination rays

The computation of solar heat fluxes requires the determination of the solar visibility. A surface can be partially illuminated by the Sun; occlusions can cause sharp shadows across the model.

5.2.1 Point wise Sun

If the Sun is considered as a point wise source, that is the solar radius is equal to zero and the distance to the Sun is infinite, the rays coming from the Sun are parallel to each other. Let us consider the configuration presented in Figure 5.3. Two surfaces are receiving a solar flux. Surface 1 is partially occluded by surface 2. As the source is point wise, the transition between light and shadow is discontinuous on surface 1. In order to determine the local illumination on surface 1, uniform random points must be generated across the surface (see Section 5.3). From each random point, a ray is sent towards the Sun. This ray is called a *shadow ray*. A shadow ray is like any other ray, except that we are interested in reaching the solar source, *i.e.* leaving the geometrical model without hitting any object along the way. If the shadow ray cannot reach the Sun, it is useless to compute exactly the intersection point; once an occlusion is identified, the process is interrupted. If the shadow ray reaches the source without hitting an obstacle, another ray, called an *illumination ray*, is sent from the Sun to the model. This ray carries solar light and impacts the model on the initial random point. This illumination ray is then propagated throughout the geometrical model in function of the thermo-optical properties.

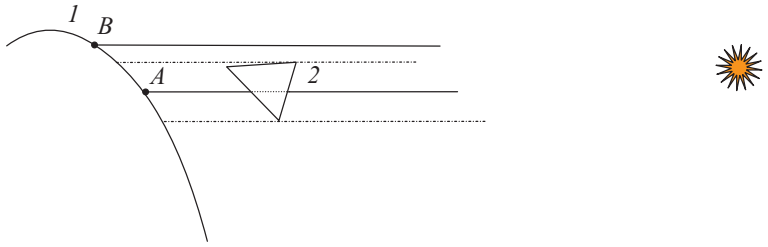


Figure 5.3: Computation of the visibility - shadow rays.

5.2.2 Sun of finite angular aperture

If the Sun is considered as a spherical source, characterized by a finite radius $R_{\odot}$ and a finite distance d, the solar rays are no longer parallel. The extended light source causes a penumbra region on surface 1 [NN86a] (see Figure 5.4).

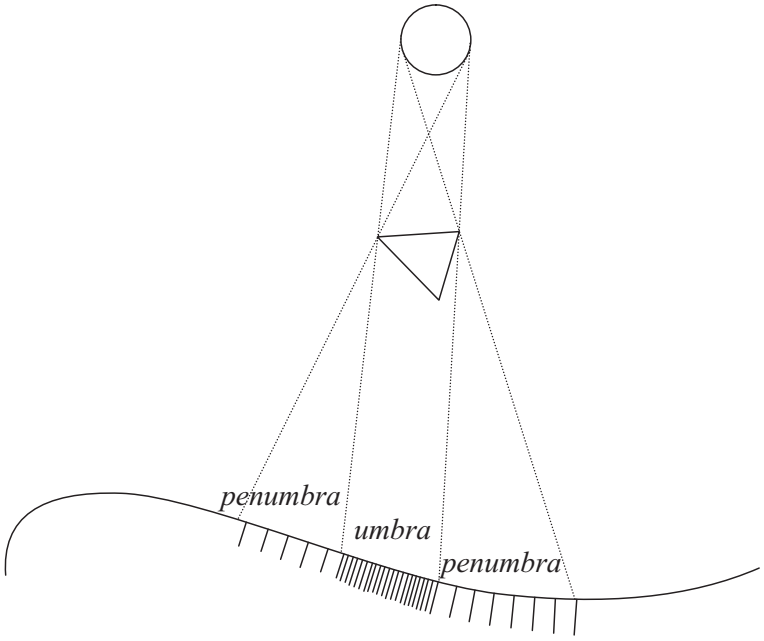


Figure 5.4: Computation of the visibility - solar angular aperture.

In this case, the direction of the shadow rays must be randomly generated, in the cone subtended by the solar disc (see Section 5.6). If the shadow ray reaches the Sun without hitting any object, an illumination ray is sent backward, to the initial random point.

5.3 Generation of surface uniform random points

In order to define the shadow rays, uniform random points must be generated across the surfaces of the model. Stratified sampling can be used to decrease the variance of the process. As it has been done for the stratified hemisphere method, a two-pass ray tracing can reduce the necessary number of rays.

In this section, we describe the mathematical reasoning which yields the uniform sampling of the surface. It can also be used for stratified sampling.

The presented method has been designed by J. Arvo [Arv01, Arv95]. This method is based on the computation of an area-preserving parametrization, noted ψ, deriver from a first arbitrary parametrization ϕ of the surface. Samples can be generated on the unit

square and then map them onto the surface with a warping function w, which can be obtained with the following procedure. Let $\mathcal{S}$ be the surface that we want to uniformly sample.

The first step implies the definition of a parametrization ϕ from the unit square $[0,1]^2$ to the surface $\mathcal{S}$.

Step 2 consists in defining a function σ that links the parametrization ϕ to the surface area of $\mathcal{S}$. This function is the determinant of the Jacobian matrix linked to the parametrization ϕ. The partial derivatives of ϕ and their cross product must be computed.

$$\sigma(s,t) = \left|\left|\frac{\partial\phi}{\partial s}(s,t) \wedge \frac{\partial\phi}{\partial t}(s,t)\right|\right| \tag{5.6}$$

Step 3 requires the definition of two cumulative functions $F(s)$ and $G_s(t)$ such as:

$$F(s) = \frac{\int_0^1\int_0^s \sigma(u,v)dudv}{\int_0^1\int_0^1 \sigma(u,v)dudv} = \frac{\text{area}(\mathcal{S}_s)}{\text{area}(\mathcal{S})} \tag{5.7}$$

$$G_s(t) = \frac{\int_0^t \sigma(s,v)dv}{\int_0^1 \sigma(s,v)dv} \tag{5.8}$$

Following the definition of $F(s)$, it is equivalent to the fraction of the area which is defined for u between 0 and s, in grey in Figure 5.5.

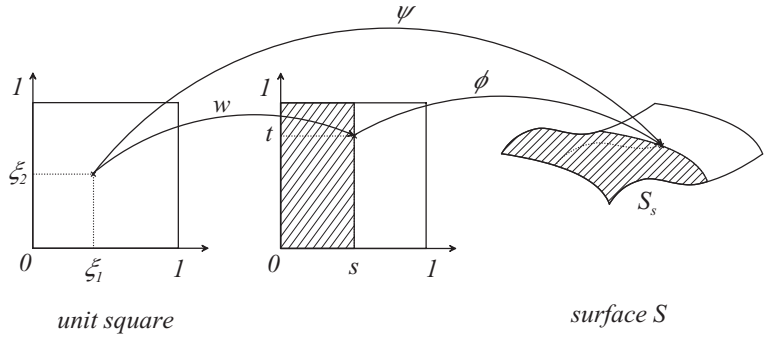


Figure 5.5: Definition of the parametrizations ϕ and ψ and the warping function w.

Step 4 implies the inversion of the two previous cumulative functions:

$$f(\xi) = F^{-1}(\xi) \tag{5.9}$$

$$g(\xi_1,\xi_2) = G^{-1}_{f(\xi_1)}(\xi_2) \tag{5.10}$$

The mapping $(\xi_1,\xi_2) \mapsto (s,t) = (f(\xi_1), g(\xi_1,\xi_2))$ defines the warping function that converts the original parametrization ϕ into the area-preserving parametrization ψ.

$$\psi(\xi_1,\xi_2) = \phi(f(\xi_1), g(\xi_1, xi_2)) \tag{5.11}$$

This parametrization has been used to generate uniform samples on a triangle (see Figure 5.6-a). If a uniform decomposition of the unit square is defined, stratified samples can be generated (see Figure 5.6-b). At the first sight, we can guess that the stratified samples are more uniformly distributed across the triangle. The interest of combining stratified sampling with the computation of solar visibility (and solar cross section) is established in Appendix F.

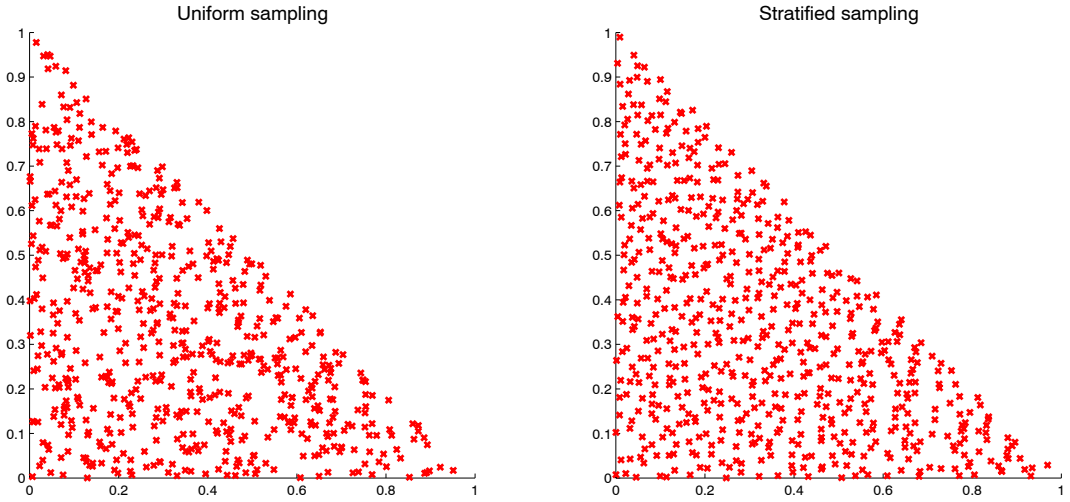


Figure 5.6: Random distribution - a) Uniform sampling - b) Stratified sampling.

This method can be used to generate uniform random points on different surfaces for which a parametrization is available.

5.4 Geometrical method

The geometrical method has been designed in order to accelerate the ray tracing process for elements (triangles and/or quadrangles) on the basis of the geometrical, mathematical definition of the model. The computed radiative quantities have to be associated with the elements. The random origins are generated across the elements and are projected onto the corresponding geometrical primitives. For each location, the local normal to the real primitive is analytically computed.

The direction of the Sun, from the current origin, is defined by two angles: the elevation and the azimuth. A first test allows us to determine if the local normal is oriented towards the Sun. If the test is positive, the shadow ray is propagated through the geometrical model, to the Sun.

5.5 Thermal formulations

5.5.1 Radiosity formulation

If the radiosity formulation is used to solve the thermal problem, *i.e.* the formulation based on (diffuse of extended) view factors, it can be shown that the external irradiation term H_0 is computed on the basis of the solar visibility and the shadow rays. It corresponds to the direct heat fluxes computed by Esarad [Doc06].

The solar visibility of a surface is defined as the fraction of this surface which receives direct illumination from the Sun.

The local visibility $V_{\odot}$ can be linked with the local direct heat flux if the cosine of the angle α between the solar direction and the local unit normal is introduced.

$$q_{\odot} = C_{\odot} \cos(\alpha) V_{\odot} \tag{5.12}$$

where $C_{\odot}$ is the solar constant.

The previous assessment is exact only if the reflectors are purely diffuse. If the model contains specular surfaces, the irradiation term must be extended in order to take specular reflections into account. When a shadow ray goes to the Sun, an illumination ray has to be traced throughout the model. The reflection processes must only be specular. No diffuse reflection must be added here. The propagation of the illumination rays can then be considered as a purely deterministic process. Only the origins' generation is random.

5.5.2 Gebhart's formulation

Gebhart's formulation is based on the notion of Gebhart's absorption factors, which directly link the surface emissive power, taking into account all types of reflections.

As the diffuse component must also be modeled during the ray tracing phase, the computation of radiative exchange factors requires more CPU time than view factors. Moreover, the radiative exchange factor matrix is no longer sparse, the compression procedure is less efficient, if not useless.

In order to use Gebhart's formulation, absorbed heat fluxes must be calculated. These absorbed heat fluxes take into account multi-reflection (diffuse and specular).

While the radiative exchange factors can be derived from the view factors, thanks to Gebhart's matrix method, the absorbed heat fluxes cannot be derived from the direct heat fluxes, as it has been discussed in Section 3.4.8. A full ray tracing must be performed to yield the absorbed heat fluxes. Let us point out that this ray tracing is not the most time-consuming; the computation of radiative exchange factors is the most demanding one.

5.6 Solar angular aperture

For now, the Sun has been considered as a point wise source, at an infinite distance. If the distance d from the Sun is known (see Table 5.1), the solar angular aperture $\gamma_\odot$ is given by the following formula, where $R_\odot$ is the solar radius:

$$\tan(\gamma_\odot) = \frac{R_\odot}{d} \tag{5.13}$$

Planet	Distance $[AU]$	Distance $[10^6 km]$	Angular aperture $[rd]$
Mercury	0.39	57.9	0.0120
Venus	0.723	108.2	0.0064
Earth	1	149.6	0.0047
Mars	1.524	227.9	0.0031
Jupiter	5.203	778.3	$8.9426\,10^{-4}$
Saturn	9.539	1 427	$4.8774\,10^{-4}$
Uranus	19.18	2 869.6	$2.4254\,10^{-4}$
Neptune	30.06	4 496.6	$1.5478\,10^{-4}$
Pluto	39.53	5 900	$1.1797\,10^{-4}$

Table 5.1: Data of the planets.

The direction of the shadow ray must be uniformly distributed across the solar disc. The direction of the Sun is defined by two angles $\{\theta_\odot, \phi_\odot\}$. If the Sun is characterized by a non-zero angular aperture, the shadow rays must be sent in the cone of direction $\overrightarrow{r_\odot} \equiv \{\theta_\odot, \phi_\odot\}$ characterized by the angle $\gamma_\odot$. The trihedral $[\overrightarrow{x_1}, \overrightarrow{x_2}, \overrightarrow{r_\odot}]$ is defined and the plane $[\overrightarrow{x_1}, \overrightarrow{x_2}]$ including the centre of the Sun[1] is considered. In this plane, the Sun appears as a disc of radius $R_\odot$. The directions of the shadow rays must be randomly uniformly distributed across the solar disc. This is achieved with two uniform random numbers ξ_1 and ξ_2 thanks to the following relations, where r is the polar radius and η the polar angle.

$$r = \sqrt{\xi_1} R_\odot \tag{5.16}$$
$$\eta = 2\pi\xi_2 \tag{5.17}$$

Figure 5.7 represents the point distribution obtained with 1 000 samples. The red circle corresponds to the limit of the solar disc. As only one shadow ray is traced from each origin, stratified sampling has not been coded here.

[1] The two unit vectors $\overrightarrow{x_1}$ and $\overrightarrow{x_2}$ can be computed through the following relations:

$$\overrightarrow{x_2} = \frac{\overrightarrow{r_\odot} \wedge \overrightarrow{n_i}}{|\overrightarrow{r_\odot} \wedge \overrightarrow{n_i}|} \tag{5.14}$$
$$\overrightarrow{x_1} = \overrightarrow{x_2} \wedge \overrightarrow{r_\odot} \tag{5.15}$$

where $\overrightarrow{n_i}$ is the local vector normal to the current element i.

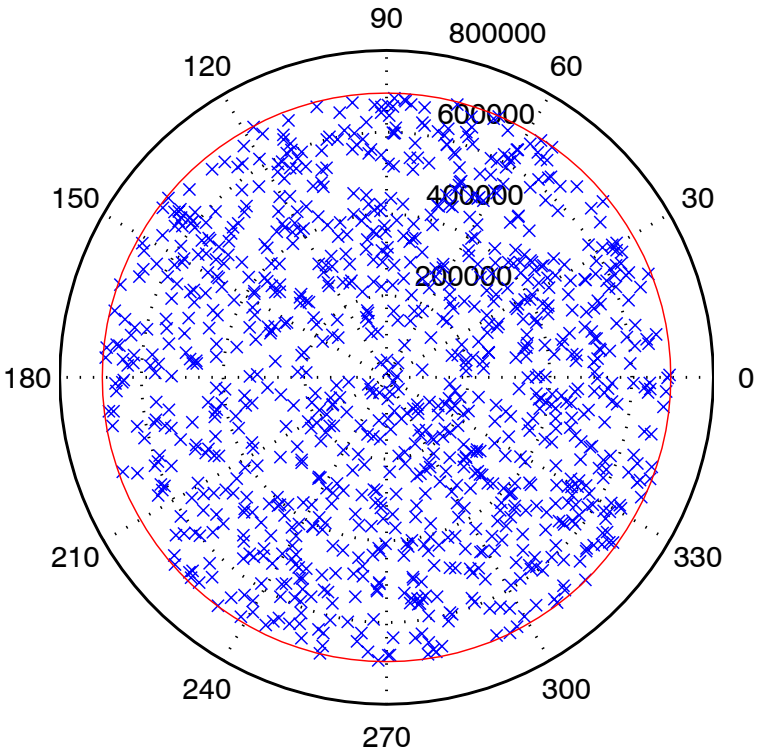


Figure 5.7: Uniform distribution across the solar disc.

The direction of the rays is expressed with two angles, noted χ and η. The conical angle χ, defined on the interval $[0, \gamma_{\odot}]$, is the angle defined by the shadow ray and the direction to the centre of the Sun. χ is linked to the polar radius.

$$\chi = \arctan\left(\frac{\sqrt{\xi_1} R_{\odot}}{d}\right) \tag{5.18}$$

On the basis of the definition of the vectors $\overrightarrow{x_1}$ and $\overrightarrow{x_2}$, the generated direction is given by the following expression (see Figure 5.8).

$$\overrightarrow{r'_{\odot}} = \cos(\chi)\overrightarrow{r_{\odot}} + \sin(\chi)\left[\cos(\eta)\overrightarrow{x_1} + \sin(\eta)\overrightarrow{x_2}\right] \tag{5.19}$$

The obtained vector is unitary and contained in the cone subtended by the Sun. The random distribution is uniform on the solar disc.

Once the direction of the shadow ray has been computed, the cosine of the local normal and the direction of the shadow ray is computed. If the obtained value is not negative, the ray is propagated through the model. If an intersection is detected, it means that the Sun is not directly visible from the origin, in the direction $\overrightarrow{r'_{\odot}}$, the shadow ray is lost. If the ray leaves the model, the solar visibility is ensured, a ray is traced from the origin and propagated throughout the model. The result of the shadow ray tracing is the computation of the solar visibility across the model.

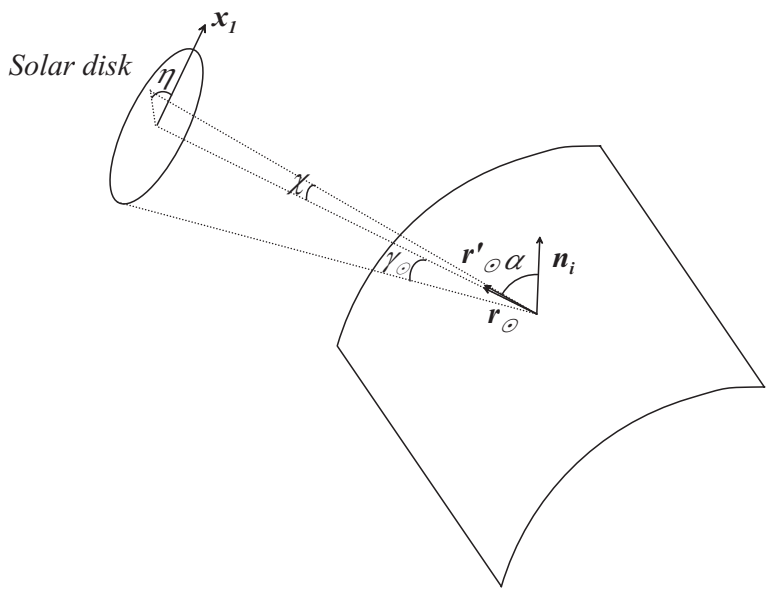


Figure 5.8: Generation of the shadow rays.

5.7 Planetary heat fluxes

In this Section, we explain how the hemisphere method could be extended to compute the planetary heat fluxes (albedo and infrared), as a perspective of the thesis. For the computation of view factors, the deterministic grid is defined for θ between 0 and 90°. If the ray casting is limited within solid angle from spacecraft to planet, the grid can be reduced for θ between 0° and γ_P, where γ_P is the angular aperture of the planet. A same grid could be used to generate one single ray tracing process, instead of two in the current algorithms.

The advantage of this method is to extend the statistical accuracy control of the hemisphere to the computation of planetary heat fluxes.

Another advantage is to easily combine this ray tracing with planet albedo and temperature maps, expressed in terms of planet latitude and longitude. For each impact, the values of the different nodes of the impacted cell can be interpolated, as it has been done with finite element view factors.

5.8 Conclusions

In this Chapter, we have detailed the computation of the external heat loads. We carefully addressed the computation of solar heat flux, taking shadows and penumbras into account. As a perspective of the thesis, we mentioned the extension of the hemisphere method to the computation of planetary heat fluxes.

Chapter 6

Mathematical formulation

Contents

This Chapter is dedicated to the mathematical developments of the innovative concepts presented in this thesis. A rigorous expression of the finite element view factor is presented. The adjoint equations are developed and a measure of the radiative error which characterizes each link between two surfaces is established. This set of equations could be a good starting point to extend some methods developed in image synthesis to radiative heat transfer.

6.1 Finite element formulation

In the previous Sections, only isothermal patches were considered. Now a formulation is presented where the temperature field is projected on a set of polynomial functions. The resulting temperatures are linear combinations of these functions, the coefficients of these functions being the new unknowns. From the general heat transfer equation, the finite element formulation of the radiative component is derived.

6.1.1 Heat transfer equation

In case of a steady-state situation, the temperature distribution through the volume $\mathcal{V}$, limited by the surface $\mathcal{S}$, is determined by the tri-dimensional heat transfer equation [SH01, LMTS96]:

$$\mathrm{div}\left[k\overrightarrow{\mathrm{grad}(T)}\right] + Q = 0 \qquad \text{on } \mathcal{V} \tag{6.1}$$

The problem must be completed with boundary conditions:

$$\left\{\begin{array}{rcccl} T(r) & = & f(r) & \text{on} & \mathcal{S}_T \\ q(r) & = & -k\overrightarrow{\text{grad}(T)}\cdot\overrightarrow{n} & & \\ & = & \aleph(r) & \text{on} & \mathcal{S}_q \end{array}\right. \tag{6.2}$$

where $f(r)$ and $\aleph(r)$ are prescribed functions of the position r. The two domains $\mathcal{S}_T$ and $\mathcal{S}_q$ must be complementary and cannot overlap:

$$\left\{\begin{array}{rcl} \mathcal{S}_T \cap \mathcal{S}_q & = & \emptyset \\ \mathcal{S}_T \cup \mathcal{S}_q & = & \mathcal{S} \end{array}\right. \tag{6.3}$$

6.1.2 Variational formulation

A set of trial functions $\mathcal{T}$ is defined. These functions must satisfy the boundary conditions on $\mathcal{S}_T$. The functions $\mathcal{T}$ are assumed to be sufficiently derivable. These functions must satisfy the following relation:

$$\int_{\mathcal{V}} \left\{\text{div}\left[k\overrightarrow{\text{grad}(T)}\right] + Q\right\}\mathcal{T}d\mathcal{V} = 0 \tag{6.4}$$

Then the following result is used:

$$\text{div}\left[k\overrightarrow{\text{grad}(T)}\right]\mathcal{T} = \text{div}\left[\mathcal{T}k\overrightarrow{\text{grad}(T)}\right] - k\overrightarrow{\text{grad}(T)}\cdot\overrightarrow{\text{grad}(\mathcal{T})} \tag{6.5}$$

The divergence theorem can be used in order to obtain the following expression:

$$\int_{\mathcal{V}}\left[k\overrightarrow{\text{grad}(T)}\right]\mathcal{T}d\mathcal{V} = \int_{\mathcal{S}_T+\mathcal{S}_q} k\overrightarrow{\text{grad}(T)}\mathcal{T}\cdot\overrightarrow{n}d\mathcal{S} - \int_{\mathcal{V}} k\overrightarrow{\text{grad}(T)}\cdot\overrightarrow{\text{grad}(\mathcal{T})}d\mathcal{V} \tag{6.6}$$

The previous equation can be introduced into the relation (6.4) to obtain the variational formulation:

$$\int_{\mathcal{S}_T+\mathcal{S}_q} k\overrightarrow{\text{grad}(T)}\mathcal{T}\cdot\overrightarrow{n}d\mathcal{S} - \int_{\mathcal{V}} k\overrightarrow{\text{grad}(T)}\cdot\overrightarrow{\text{grad}(\mathcal{T})}d\mathcal{V} + \int_{\mathcal{V}} Q\mathcal{T}d\mathcal{V} = 0 \tag{6.7}$$

If the functions $\mathcal{T}$ are chosen to satisfy the homogeneous form of the boundary condition on $\mathcal{S}_T$ (called therefore *essential* boundary condition), the previous relation can be simplified:

$$\int_{\mathcal{V}} k\overrightarrow{\text{grad}(T)}\cdot\overrightarrow{\text{grad}(\mathcal{T})}d\mathcal{V} = \int_{\mathcal{V}} Q\mathcal{T}d\mathcal{V} - \int_{\mathcal{S}_q}\aleph\mathcal{T}d\mathcal{S} \tag{6.8}$$

This is called the weak variational formulation. The boundary conditions on $\mathcal{S}_q$ are called *natural* boundary conditions as they are not directly enforced by the choice of the functions $\mathcal{T}$ but are satisfied by the solution $T(x)$ of the formulation.

6.1.3 Galerkin method

The previous variational formulation is used to develop the so-called *Galerkin method* for solving the thermal problem. First, a function $T_0(x)$ is defined which satisfies:

$$T_0(x) = f(x) \qquad \text{on } \mathcal{S}_T \tag{6.9}$$

Then a set of functions N_j which satisfy the homogeneous form of the boundary conditions on $\mathcal{S}_T$ is built:

$$N_j(x) = 0 \qquad \text{on } \mathcal{S}_T \tag{6.10}$$

The same functions are selected to project the temperature field and to build the trial functions $\mathcal{T}$:

$$T(x) = T_0(x) + \sum_{j=1}^{+\infty} a_j N_j(x) \tag{6.11}$$

$$\mathcal{T}(x) = \sum_{j=1}^{+\infty} b_j N_j(x) \tag{6.12}$$

where a_j and b_j are constant. In equations (6.11) and (6.12), the sums contain an infinite number of terms. These terms have to be truncated in order to consider a finite number of functions N_j.

$$\widetilde{T}(x) = T_0(x) + \sum_{j=1}^{M} a_j N_j(x) \tag{6.13}$$

$$\widetilde{\mathcal{T}}(x) = \sum_{j=1}^{M} b_j N_j(x) \tag{6.14}$$

The Galerkin method consists in computing the function $\widetilde{T}$ which satisfies the following relation:

$$\int_{\mathcal{V}} k\overrightarrow{\text{grad}(\widetilde{T})} \cdot \overrightarrow{\text{grad}(\mathcal{T})} d\mathcal{V} = \int_{\mathcal{V}} Q\mathcal{T} d\mathcal{V} - \int_{\mathcal{S}_q} \aleph \mathcal{T} d\mathcal{S} \tag{6.15}$$

If the approximation (6.14) is used for the trial function, the relation (6.15) is equivalent to the set of equations:

$$\int_{\mathcal{V}} k\overrightarrow{\text{grad}(\widetilde{T})} \cdot \overrightarrow{\text{grad}(N_i)} d\mathcal{V} = \int_{\mathcal{V}} Q N_i d\mathcal{V} - \int_{\mathcal{S}_q} \aleph N_i d\mathcal{S} \qquad \text{with } i = 1, ...M \tag{6.16}$$

If the approximate temperature field (6.13) is introduced into (6.16), the following set of equations is finally obtained:

$$\begin{aligned} \int_{\mathcal{V}} k\overrightarrow{\text{grad}(T_0)} \cdot \overrightarrow{\text{grad}(N_i)} d\mathcal{V} + \sum_{j=1}^{M} \int_{\mathcal{V}} \left[k\overrightarrow{\text{grad}(N_j)} \cdot \overrightarrow{\text{grad}(N_i)} d\mathcal{V} \right] a_j = \\ \int_{\mathcal{V}} Q N_i d\mathcal{V} - \int_{\mathcal{S}_q} \aleph N_i d\mathcal{S} \qquad \text{with } i = 1, ...M \end{aligned} \tag{6.17}$$

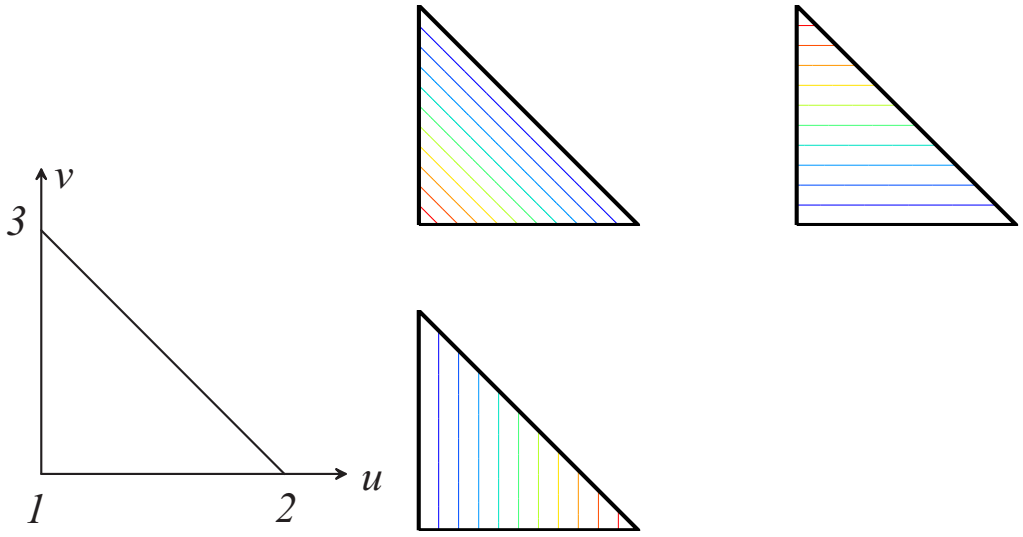
The last term of equation (6.17) corresponds to the external fluxes applied on the surface $\mathcal{S}_q$. This term contains the radiative component.

6.1.4 Finite element method

On the basis of equation (6.17), the finite element method is based on a set of polynomial functions N_j. The domain is decomposed into a number of elements. For a given node j of an element, the function N_j is defined such as:

$$N_j = \begin{cases} 1 & \text{on node } j \\ 0 & \text{on node } i \neq j \end{cases} \tag{6.18}$$

If this choice is adopted, the function $\mathcal{T}(x)$ is such that $\mathcal{T} = a_j$ when it is evaluated on node j. The unknowns a_j can be rewritten as T_j. The functions N_j are called *shape functions* and are described in the next Section.



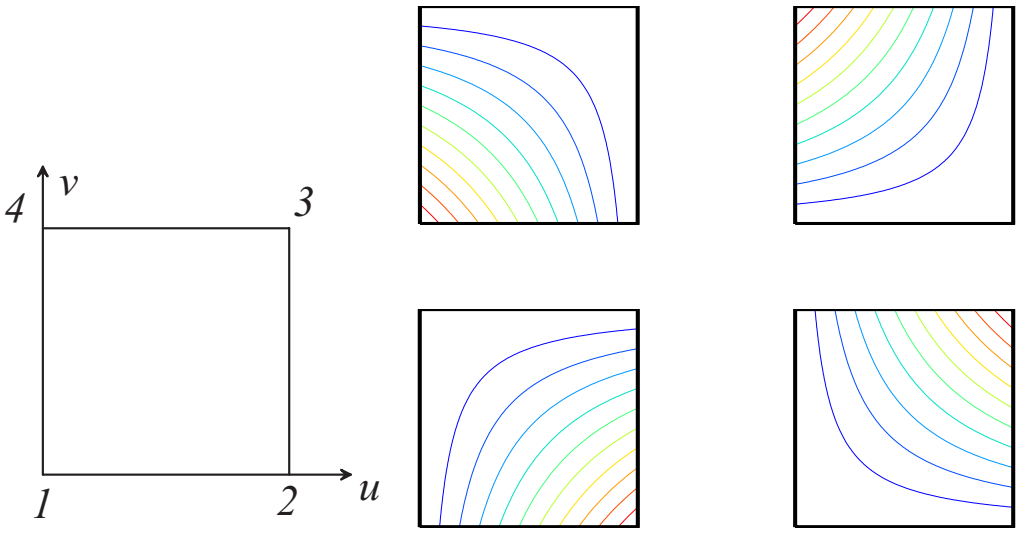
(a) Triangular element (b) Shape functions

Figure 6.1: First-order triangle.

In the proposed algorithms, the model is decomposed into a number N of patches, which are first-order triangles and quadrangles. Linear shape functions are chosen for the triangle. Figure 6.1–a represents the standard three-node triangle defined in $\{u, v\}$ coordinates. The shape functions are numbered following the nodes. Figure 6.1–b shows the evolution of the three shape functions across the triangle. These shape functions are equal to unity on the corresponding node and to zero on the other nodes. The sum of the

three shape functions is equal to unity for each point of the triangle.

$$\left\{\begin{array}{rcl} N_1(u,v) &=& 1-u-v \\ N_2(u,v) &=& u \\ N_3(u,v) &=& v \end{array}\right. \tag{6.19}$$



(a) Quadrangular element (b) Shape functions

Figure 6.2: First-order quadrangle.

The first-order quadrangle is shown in Figure 6.2–a. Figure 6.2–b corresponds to the shape functions related to the four nodes. These functions are bilinear due to the term uv. The shape functions are given by the following relations:

$$\left\{\begin{array}{rcl} N_1(u,v) &=& 1-u-v+uv \\ N_2(u,v) &=& u-uv \\ N_3(u,v) &=& uv \\ N_4(u,v) &=& v-uv \end{array}\right. \tag{6.20}$$

6.1.5 Radiative heat flux

Let us recall that the radiative heat flux normal to a surface at some location r is given by equation (2.61):

$$\frac{q(r)}{\epsilon(r)} = E_b(r) - \int_{A'} \left(1-\rho^s(r)\right) E_b(r') dF_{dA-dA'} + \int_{A'} \frac{\rho^d}{\epsilon} q(r') dF_{dA-dA'} - H_0^s(r) \tag{6.21}$$

Each term must be multiplied by the shape function N_j^i, where the index i refers to the element and j corresponds to the node. Each term must then be integrated onto the element A_i. Before giving the final result, let us consider a particular term appearing in this expression, linked to the multi-reflection.

6.1.5.1 Finite element view factor

During the integration of equation (6.21) on A_i, the term $\int_{A_k} N_l(r_k) dF^s_{dA_i - dA_k}$ must be computed. The index i is related to the emission surface while k corresponds to the target. The index l refers to one of the nodes of k and $N_l(r_k)$ is the value of the shape function evaluated at a point on A_k. The integral on A_k can be performed by a Monte Carlo integration [CHSP95, CPF89]. A large number of rays are generated and surface k is tested for intersection. If the intersection test is positive, the parameters $\{u, v\}$ of the intersection point are computed. The shape functions can be evaluated. The energy E_i of each ray is split between the nodes according to the shape functions. The overall energy is conserved.

$$E_l = E_i N_l(u, v) \tag{6.22}$$

The *finite element view factor* is defined by the following relation:

$$F^{i-k}_{dA_i - l} = \int_{A_k} N_l(r_k) dF^s_{dA_i - dA_k} \tag{6.23}$$

where the top indices i and k refer to the surfaces while the down ones correspond to the point dA_i on the origin patch and node l.

In particular, the finite element view factor F^{i-k}_{j-l} represents the fraction of energy emitted by a node j through the surface i which reaches node l on surface k.

6.1.6 Heat flux equation

After the integration of (6.21) on A_i and some long manipulations (not reproduced in this document for the sake of brevity), the following relation is obtained:

$$\begin{aligned} q_j &= \epsilon_i E_{b,j} - \epsilon_i \sum_{k=1}^{N} (1 - \rho^s_k) \sum_{l=1}^{NG_k} E_{b,l} \frac{\sum_i F^{i-k}_{j-l} \det(J_i)}{\sum_i \det(J_i)} \\ &+ \epsilon_i \sum_{k=1}^{N} \frac{\rho^d_k}{\epsilon_k} \sum_{l=1}^{NG_k} q_l \frac{\sum_i F^{i-k}_{j-l} \det(J_i)}{\sum_i \det(J_i)} - \epsilon_i H^s_0(u_j, v_j) \end{aligned} \tag{6.24}$$

The term $\det(J_i)$ represents the determinant of the jacobian matrix of the transformation from the Cartesian coordinates to the $\{u, v\}$ standard coordinates. In the case of a first-order triangle, this determinant is twice the surface area: $\det(J_i) = 2A_i$.

The finite element view factors cannot simply be added on a same node. They must be weighted by the determinant of the jacobian matrix of each element i.

6.1.7 Conductive component

Although the conductive component can also be added in the previous formulations, it is briefly introduced in this Section because the generation of the conductive links is simple

and fast with finite elements.

From equation (6.17), a $N \times N$ matrix can be defined, corresponding to:

$$K_{ij} = \int_{\mathcal{V}} k\overrightarrow{\text{grad}(N_j)} \cdot \overrightarrow{\text{grad}(N_i)} d\mathcal{V} \tag{6.25}$$

In case of an isotropic material, the coefficients K_{ij} are only function of the geometry of the elements, *i.e.* the discretization of the geometrical model.

6.1.8 Ray tracing implementation

To conclude this Section, we present now the implementation of the ray tracing, allowing the computation of finite element view factors, for a model defined by a geometry and a finite element mesh. The emission process is first detailed. Then, the reflection is addressed.

6.1.8.1 Emission process

In this work, two different emission processes have been defined. The first one is based on the Gaussian points and can be used for isothermal elements, associated with uniform patch-to-patch view factors. The second one is based on the nodes of the elements and can be used with polynomial approximations of the temperature field and view factors.

Isothermal patches The rays are generated with the stratified hemisphere method from the Gaussian points of each triangular element. For curved primitives (cones, cylinders and spheres), these points are not exactly placed on the surfaces. They have to be projected onto the corresponding surface, following the local normal. The local normal will also be needed for the emission of the rays. Given the $\{u, v\}$ parameters of a Gaussian point, analytical formulas are available for each primitive in order to compute the projected origin and the local normal (see Section 4.2).

Polynomial elements If finite elements are considered, the nodes can be chosen as the origins of the rays. These points are already on the surface so no projection is necessary. Each node belongs to a single surface; no discontinuity of the local normal occurs. Along a common edge, for example between two squares, the nodes need to be duplicated. One set of nodes belongs to the first square while the other one corresponds to the second square. In this case, the critical nodes must be slightly shifted in the opposite direction to the common edge, to prevent ray tracing artifacts due to unadapted geometrical tolerance. If the rays were sent from the edge, they could be intercepted by the other surface sharing the edge. So the associated view factor could be highly over-estimated (up to 100%).

6.1.8.2 Intersection- and reflection- computations

The ray-surface intersection can be analytically computed, thanks to adequate equations. Once an intersection has been detected, the $\{u, v\}$ parameters of the intersection point on the surface are computed to retrieve the local normal. This normal vector is necessary to calculate the direction of the reflected ray (if any), either for diffuse or specular reflection. The intersection point is projected onto the triangular element, following the local normal. The incident energy[1] is attributed to the corresponding element. In the case of a polynomial element, the $\{u, v\}$ coordinates of the projection onto the element are computed. The incident (absorbed) energy is then split between the nodes of the element thanks to the shape functions:

$$E_i = E_0 N_i(u, v) \tag{6.26}$$

where E_0 is the incident (absorbed) energy; E_i is the energy attributed to the node i; $N_i(u, v)$ is the shape function of the node i evaluated at the intersection point $\{u, v\}$.

This method satisfies the conservation of energy since the sum of the shape functions is equal to unity, for each point $\{u, v\}$ on the element.

6.2 Adjoint equations and accuracy measure

In this Section, the notion of *importance* is presented, as it is defined in image synthesis. The *adjoint equations* are developed, the transport operator is defined, and the notion of importance is extended to be applicable in radiative heat transfer. We explain how this can be used in a post-treatment in order to yield an accuracy measure on the radiative heat transfer computation. In this developments, only diffuse surfaces have been considered with constant thermo-optical properties. The extension to more complex thermo-optical properties and non-isothermal elements can be considered for future developments.

6.2.1 Background

The accuracy of Monte Carlo ray tracing is function of the number of traced rays; the correct number of rays is left to the discretion of the engineer. A bad estimate may lead to an unacceptable error or an unnecessary computation load. A statistical accuracy control, which could automatically compute the number of rays to achieve the required level of accuracy, is a clear improvement for software dedicated to space thermal engineering. This Section proposes a solution to bridge this gap.

[1] If the radiative exchange factors are computed, in the framework of Gebhart's formulation, the absorbed energy needs to be calculated.

6.2.2 Definition of importance in rendering

In rendering, a first category of algorithms are based on the propagation of light from the sources, throughout the geometrical model, represented by radiosity algorithms; a second class of algorithms consider only the light reaching the camera, typified by ray tracing. These two methods can be considered to be dual [SAS92]: the first one represents the propagation of the light from the sources to the receptors while the last one traces rays from the receptors. If the first one is based on the resolution of the transport equation (*i.e.* the radiosity equation), the second one is based on the adjoint of the initial transport equation. These two strategies can be combined, leading to a solution of higher efficiency.

The second transport equation (the adjoint equation) is based on the notion of importance. This notion has been intensively used in rendering in order to accelerate the generation of realistic images [Chr03]. Let us consider an environment where all the lights are turned off [PM95]. We shine light from the important surfaces. This light is propagated throughout the model and experiences multiple reflections before being absorbed. The contribution of each surface to the final image is proportional to the received light. The interest of importance in rendering is that the computation can be focused on the surfaces that contribute most to the image. A similar remark can be formulated in heat transfer for space applications, when a particular instrument is modeled aboard a whole spacecraft.

Importance is propagated throughout the model like light but in the opposite way. It can be shown that the importance distribution is governed by the following relation [SAS92]:

$$I_i = R_i + \sum_{j=1}^{N} \rho_j F_{j-i} I_j \tag{6.27}$$

where $I_i = A_i W_i$ is the surface importance of patch P_i and R_i is the reception term, dual to the source term E_i (light emitted by P_i as a lamp). R_i represents the initial emission of importance. In heat transfer, in the case of diffuse, isothermal patches, the reception term of a patch P_i is equal to its area A_i. The demonstration of this equation can be found in Appendix B.2.

6.2.3 Radiosity equation

Following [Mod03], the radiative balance of the thermal model can be expressed by the following set of equations, based on the radiosities J_j and the view factors:

$$\frac{q_i}{\epsilon_i} = E_{b,i} - \sum_{j=1}^{N} F_{i-j} J_j - H_{0,i} \tag{6.28}$$

The radiative heat flux q_i, the self-emitted power $E_{b,i}$ and the radiosity J_i of a patch

P_i are linked by the following relation:

$$\frac{q_i}{\epsilon_i} = \frac{1}{1-\epsilon_i}\left(E_{b,i} - J_i\right) \tag{6.29}$$

For each patch P_i, two expressions of the radiosity equation can be obtained, in function of the nature of the boundary condition. Let us assume that the temperatures of the n first patches are known while the corresponding heat fluxes are unknown. On the other hand, the radiative heat fluxes of the $N-n$ other patches are fixed and their temperatures have been computed. For each patch, the radiosity equation is given by the following relation, in function of the boundary condition:

$$\begin{array}{rcll} J_i & = & \epsilon_i E_{b,i} + \rho_i H_{0,i} + \rho_i \sum_{j=1}^{N} F_{i-j} J_j & \forall \quad i \in [1,n] \\ J_i & = & q_i + H_{0,i} + \sum_{j=1}^{N} F_{i-j} J_j & \forall \quad i \in [n+1,N] \end{array} \tag{6.30}$$

The set of radiosity equations can be rewritten as follows:

$$\underbrace{\begin{pmatrix} 1-\rho_1 F_{1-1} & \dots & -\rho_1 F_{1-n} & \dots & -\rho_1 F_{1-N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ -\rho_n F_{n-1} & \dots & 1-\rho_n F_{n-n} & \dots & -\rho_n F_{n-N} \\ -F_{n+1-1} & \dots & -F_{n+1-n} & \dots & -F_{n+1-N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ -F_{N-1} & \dots & -F_{N-n} & \dots & 1-F_{N-N} \end{pmatrix}}_{\text{transport operator } \mathcal{R}} \begin{pmatrix} J_1 \\ \vdots \\ J_n \\ J_{n+1} \\ \vdots \\ J_N \end{pmatrix} = \underbrace{\begin{pmatrix} \epsilon_1 E_{b,1} + \rho_1 H_{0,1} \\ \vdots \\ \epsilon_n E_{b,n} + \rho_n H_{0,n} \\ q_{n+1} + H_{0,n+1} \\ \vdots \\ q_N + H_{0,N} \end{pmatrix}}_{\text{thermal source } S} \tag{6.31}$$

where the matrix $\mathcal{R}$ is called the *transport operator*. The thermal source S is based on the boundary conditions and the external irradiation; this vector is assumed to be exact. Relation (6.31) will be referred as the *thermal radiosity equation*.

6.2.4 Transport operator and definition of importance in heat transfer

The operator $\mathcal{R}$ is defined as the discrete approximation of the transport operator. It is a matrix of real numbers, which implies that $\mathcal{R}^* = \mathcal{R}^T$. This operator is described hereafter:

$$\mathcal{R} = \begin{bmatrix} 1-\rho_1 F_{1-1} & \dots & -\rho_1 F_{1-n} & \dots & -\rho_1 F_{1-N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ -\rho_n F_{n-1} & \dots & 1-\rho_n F_{n-n} & \dots & -\rho_n F_{n-N} \\ -F_{n+1-1} & \dots & -F_{n+1-n} & \dots & -F_{n+1-N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ -F_{N-1} & \dots & -F_{N-n} & \dots & 1-F_{N-N} \end{bmatrix} \tag{6.32}$$

The definition of importance (6.27), introduced in rendering, must be modified for thermal purpose in function of the boundary condition of the patches. Importance becomes

a mathematical quantity which is governed by the adjoint set of equations, based on the transport operator $\mathcal{R}$. To keep the definition of rendering, we need to define an alternative set of thermo-optical properties ρ'; if we consider a fictive model, where the patches with fixed heat fluxes are associated with diffuse reflectance ρ'_i equal to 100%, the definition of rendering remains valid.

The radiosity and importance equations can be rewritten as follows:

$$\mathcal{R}J = S \tag{6.33}$$
$$\mathcal{R}^* I = R \tag{6.34}$$

Equations (6.33) and (6.34) are called *adjoint equations*.

6.2.5 Radiative energy

$v(J)$ is defined as the scalar product of the radiosity vector J by the reception vector R. It is also equal to the scalar product of the importance vector by the source vector, illustrating the duality of radiosity and importance:

$$v(J) = R^T J = I^T S \tag{6.35}$$

In heat transfer, the reception vector is equivalent to the area vector A_i, yielding:

$$v(J) = \sum_{i=1}^{N} A_i J_i \tag{6.36}$$

Each term $A_i J_i$ is the diffuse energy emitted by the surface i; it is the sum of the self emission and the reflection of all incident radiations. $v(J)$ is called *radiative energy* and corresponds to the energy which is diffusely emitted by all the surfaces of the tridimensional model.

6.2.6 Measure of the error

The geometrical model has been discretized into patches *i.e.* into surfaces of finite area. In the proposed algorithms, the view factors are computed by stochastic ray tracing. As the number of rays is limited, the view factors are approximated. As the ray tracing is based on a random process, an error measure can be derived for each single view factor (see Section 3.1.6 applied to stratified hemisphere). This error measure is only geometrical. It is not representative of the error energy. Here, we present a way to derive such an error measure.

Let $\widetilde{\mathcal{R}}$ be the approximated transport operator, obtained by ray tracing:

$$\widetilde{\mathcal{R}} = \mathcal{R} + \Delta\mathcal{R} \tag{6.37}$$

The diffuse reflectivities can be assumed to be exact. The error is only due to the view factors. Each term $\widetilde{\mathcal{R}}_{i-j}$ of the transport operator is affected by an error $-\rho'_i \Delta F_{i-j}$.

The thermal radiosity, computed by inverting equation (6.33), is also approximated:

$$\widetilde{J} = J + \Delta J \tag{6.38}$$

It is an acceptable hypothesis to suppose that the source term S is exactly known. Equation (6.30) then yields the following relation:

$$\widetilde{\mathcal{R}}\widetilde{J} = S \tag{6.39}$$

By combining equations (6.37) and (6.39), we obtain the following expression, where the operator is the exact transport operator and where the source term is perturbed by a quantity $\Delta\mathcal{R}\widetilde{J}$:

$$\mathcal{R}\widetilde{J} = S - \Delta\mathcal{R}\widetilde{J} \tag{6.40}$$

The error energy is defined thanks to the radiative energy $v(J)$:

$$v(J - \widetilde{J}) = R^T(J - \widetilde{J}) = I^T \Delta\mathcal{R}\widetilde{J} \tag{6.41}$$

The quantity $I^T\Delta\mathcal{R}\widetilde{J}$ is the error introduced in the radiative energy $v(J)$ by the approximations of the transport operator $\mathcal{R}$ and the radiosities J_i. The importance cannot be exactly computed; it is associated with an error due to the approximation of the transport operator. $\widetilde{I}^T\mathcal{R}\widetilde{J}$ is used as the approximation of $I^T\mathcal{R}\widetilde{J}$. Equation (6.41) is a double sum on all the surfaces which compose the geometrical model. A particular term $\widetilde{I}_i\Delta\mathcal{R}_{i-j}\widetilde{J}_j = \widetilde{I}_i\rho'_i\Delta F_{i-j}\widetilde{J}_j$ corresponds to the error energy from surface j to surface i. This expression allows us to establish a measure of the error induced by the approximation of the transport term $\mathcal{R}_{i-j}$. From the expression of the transport operator (6.32), supposing that the error is linked to the computation of the view factors, each term $\mathcal{R}_{i-j}$ of the transport operator is affected by an error $-\rho'_i\Delta F_{i-j}$.

6.2.7 Perspectives

After the resolution of the radiative thermal problem, once the radiosities J_i are computed, the statistical accuracy measure developed for the ray tracing can be used to obtain a measure of the energy error. The radiative error associated with the geometrical model is determined by equation (6.36). Each link contributes to the global error of the radiative energy $v(J)$. The knowledge of ΔF_{i-j} is equivalent to know the radiative error associated with this link.

Currently, we can design a statistical accuracy measure associated with the radiative energy. The distribution of the energy error through the geometrical model can be computed. A next step could be to design a statistical accuracy control in order to adapt the ray tracing; the number of rays can be locally modified in order to achieve an energy accuracy defined by the user in the form of a maximum relative error ϵ and a confidence interval α. This would lead to a fully statistically controlled ray tracing.

6.3 Conclusions

In this Chapter, the innovative mathematical developments performed in the framework of the thesis have been presented. The expression of the finite element view factor has been rigorously established and the adjoint equations have been developed as a starting point for further mathematical developments, based on an active accuracy control.

Chapter 7

Validation

Contents

This Chapter presents a selection of the test cases implemented to validate the mathematical behaviour and the performances of the proposed algorithms. It contains one large application, model containing a large number of elements. The results are then compared to industrial tools (Samcef and/or Esarad-Esatan).

In this Chapter, the stratified hemisphere method has been used to generate the rays; the geometrical method, combined with uniform spatial subdivision, has been implemented to accelerate the ray tracing process.

The results displayed in this Chapter are diffuse and extended view factors, absorbed heat fluxes. Numerical comparisons are also given. Distributions of temperatures are also provided, although it was not an initial requirement of the thesis. Finite element view factors and the integration of conduction are illustrated. The Chapter ends with a study of the mathematical behaviour of the algorithms (evolution of the computation time in function of the different parameters).

7.1 Analytical configurations

The first class of test cases are based on simple geometrical configurations for which an analytical solution can be established. The results can then directly be compared with the theoretical value, in order to obtain relative errors and verify the statistic properties of the solutions.

7.1.1 Samcef cylinders - View factors

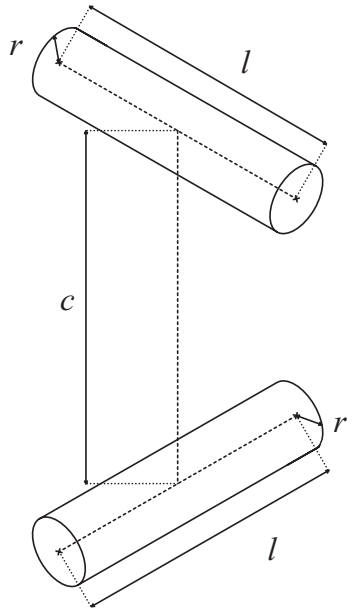


Figure 7.1: Samcef cylinders - Geometrical configuration.

The test case presented in this Section is inspired by the Samcef manual [Doc07]. The geometry is composed of two cylinders of equal radius and equal finite length, with closest distance at centers (see Figure 7.1).

This example allows to compare the proposed algorithm with the reference software Esarad and the finite element software Samcef. The compared quantity is the view factor from one cylinder to the other. An approximated formula is used to validate the results from the three codes.

The view factors between the two cylinders are displayed in Figure 7.2. The distributions are similar for the three compared software. For the sake of brevity, only a modified display of the Esarad results is displayed.

7.1.1.1 Approximated solution

Following [AD82] cited by [How82], the view factor between the two cylinders is given by the following relation:

$$\begin{aligned} F_{1-2} &= 0.178\left(\tfrac{X}{2.59}\right)^{-0.95}\left(\tfrac{L}{X}\right)^{-0.16}\exp\left(-\tfrac{1}{1.86}\left|\ln\left(\tfrac{L}{X}\right)\right|^{1.61}\right) \quad \text{if} \quad \tfrac{L}{X}<1 \\ &= 0.178\left(\tfrac{X}{2.59}\right)^{-0.95}\left(\tfrac{L}{X}\right)^{-0.16}\exp\left(\tfrac{1}{0.494}\left|\ln\left(\tfrac{L}{X}\right)\right|^{0.889}\right) \quad \text{if} \quad \tfrac{L}{X}>1 \end{aligned} \tag{7.1}$$

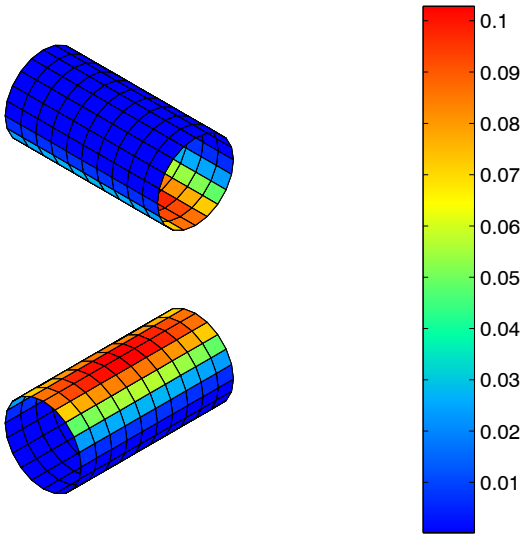


Figure 7.2: Samcef cylinders.

where the parameters C, L and X are defined as follows:

$$\begin{cases} C &= \frac{c}{r} \\ L &= \frac{l}{r} \\ X &= 2.42C - 2.24 \end{cases} \tag{7.2}$$

It yields a view factor equal to $2.543\,10^{-2}$, with an accuracy within 6%. The view factor is then comprised between $2.3905\,10^{-2}$ and $2.6956\,10^{-2}$.

7.1.1.2 Pvtan solution

The two cylinders are meshed with triangular elements. The geometrical method is used to compute the view factors between elements. If these elementary view factors are combined, it yields a surface view factor which is close to the analytical value. The view factor from cylinder to cylinder is equal to $2.620\,10^{-2}$, which is in good agreement with the approximated solution of Section 7.1.1.1.

7.1.1.3 Esarad and Samcef solutions

Esarad is used to compute the diffuse view factors between the elements. The results are exported in a **.d* file, which is the standard file from Esarad to Esatan, containing all the radiative quantities computed by Esarad, and Matlab is used to display the view factors in a way common to Pvtan and Samcef (see Figure 7.2). The view factor from cylinder to cylinder is equal to $2.622\,10^{-2}$, which is in good agreement with the approximated solution

of Section 7.1.1.1.

Finally, the previous solutions are compared with a Samcef result. The view factor from cylinder to cylinder is equal to $2.502\,10^{-2}$, which is in good agreement with the approximated solution of Section 7.1.1.1.

The different results are summarized in Table 7.1. The small differences in the results are linked to the different used methods. Samcef approximates the cylinders with planar quadrangles and computes the view factors thanks to contour integration; it applies an exact method to an approximate geometry. Esarad computes the view factors between the cylinders by ray tracing; it combines a numerical method with the real geometry. Pvtan is based on an intermediary approach, where the geometry is used for the ray tracing but where the external contours (two circles for each cylinder) are approximated by straight lines. However, this difference with Esarad is not predominant and the two results are similar.

Software	View factor
Pvtan	$2.620\,10^{-2}$
Esarad	$2.622\,10^{-2}$
Samcef	$2.502\,10^{-2}$

Table 7.1: Values of the view factor from cylinder to cylinder.

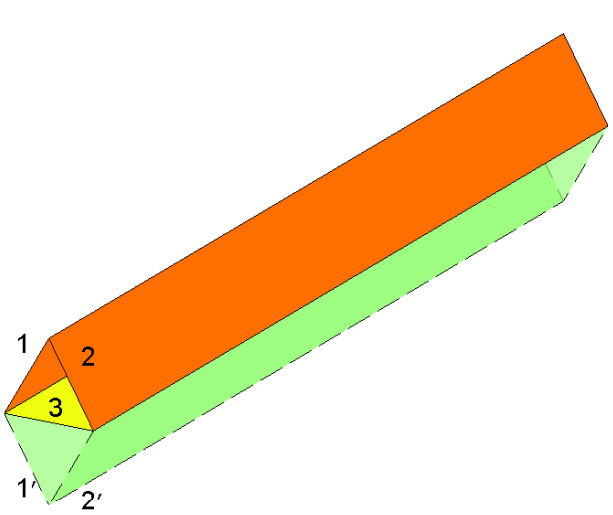
7.1.2 Triangular prism with specular reflection - Extended view factors

In order to validate the proposed algorithms in case of specular reflectors, the extended view factors related to the tenth unit test case of Esarad have been computed. The geometry is composed of three rectangular surfaces assembled into a triangular prism (see Figure 7.3–a). Surfaces 1 and 2 are purely diffuse while surface 3 (in orange) is a specular reflector. If the image method is used (see Section 2.2.6.2), the image of the surfaces 1 and 2, noted 1' and 2', are obtained (in green, transparent, limited by dashed lines in Figure 7.3–a).

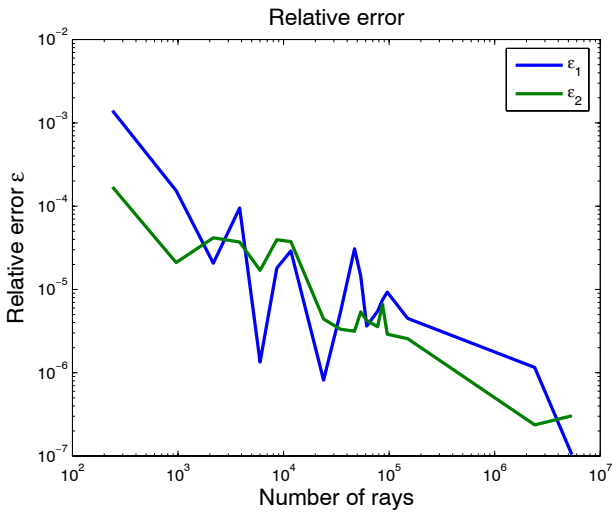
7.1.2.1 Analytical solution

For this configuration, an analytical solution can be established:

$$F = \begin{bmatrix} \rho_3^s F_{1-1'} & F_{1-2} + \rho_3^s F_{1-2'} & F_{1-3} \\ F_{1-2} + \rho_3^s F_{2-1'} & \rho_3^s F_{2-2'} & F_{2-3} \\ F_{3-1} & F_{3-2} & 0 \end{bmatrix} \tag{7.3}$$



(a) Geometrical configuration



(b) Relative errors

Figure 7.3: Test case 10 – a) Geometrical configuration – b) Relative errors.

7.1.2.2 Pvtan solution

The Pvtan method is used to compute the extended view factors. Different number of rays are considered. For each number of rays, 1 000 simulations are performed in order to compute average values and compare them to analytical values.

To illustrate the correct mathematical behaviour of the proposed algorithm with the number of rays, surfaces 1 and 2 have been considered. The global view factor[1] of the two surfaces are compared to the analytical values and relative errors are computed, for different numbers of rays. The evolution of these two relative errors is plotted in Figure 7.3–b; the convergence of the method when the number of rays increases can be observed. The logarithm of the relative errors is roughly a linear function of the logarithm of the number of rays.

7.1.3 Concentric spheres with specular reflection

This example illustrates the impact of the discretization error on the extended view factor. The case of two concentric spheres is considered. The inner one is diffuse while the outer

[1]For each line i of the view factor matrix, the global view factor G_i is the sum on the column elements:

$$G_i = \sum_{j=1}^{N} F_{i-j} \tag{7.4}$$

In case of extended view factors, following definition in Section 2.2.6, the global view factor can exceed unity [Mod03].

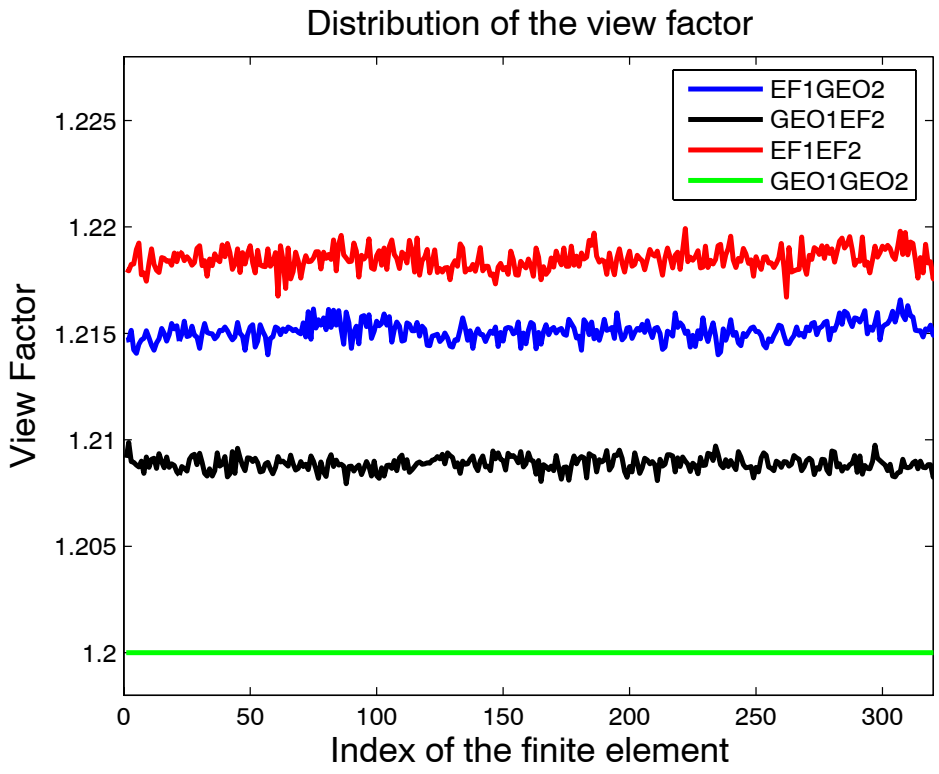


Figure 7.4: Resolution of the geometrical cells.

one is characterized by a specular reflectance $\rho^s = 0.2$. During the ray tracing, all the rays emitted by the inner sphere will impact the outer sphere, they will be reflected specularly and then will necessarily impact the inner surface. In this configuration, an analytical solution is available [How82]. The following values are obtained:

$$\left\{ \begin{array}{rclrcl} F^s_{1-1} & = & 0.2 & F^s_{1-2} & = & 1 \\ F^s_{2-1} & = & 0.25 & F^s_{2-2} & = & 0.9375 \end{array} \right. \tag{7.5}$$

The sum of the view factors gives the following values (corresponding to the global view factors):

$$\left\{ \begin{array}{rcl} F^s_{1-1} + F^s_{1-2} & = & 1.2 \\ F^s_{2-1} + F^s_{2-2} & = & 1.1875 \end{array} \right. \tag{7.6}$$

A mesh composed of 320 triangles has been used. In order to establish the impact of the discretization of the curved surfaces by a finite element mesh, four cases are considered. In the first one, the surfaces are associated with their geometrical primitives. In the second case, only the outer sphere is combined with its primitive. In the third case, only the inner primitive is identified as a sphere. In the last configuration, the two surfaces are considered as finite element objects.

In Figure 7.4, the results obtained for the different cases are plotted in function of the index of the finite elements (from 1 to 320). The green curve corresponds to the geometrical method. It is exactly equal to the analytical value. If one of the surfaces is replaced by the finite element mesh (see the blue and black curves), it yields an over-estimation of the real value. When the outer sphere is meshed, the reflection of the rays is not based on the real curvature of the surface. It is responsible for the error observed on the black curve. When the inner sphere is meshed, the emission of the rays does not follow the curvature of the surface. Moreover, incident rays can miss the contour of the surface. They can undergo multiple reflections before being absorbed by the inner sphere. These two sources of errors cause the larger error observed on the blue curve. Finally, if the two surfaces are considered as finite element objects *i.e.*, we do not know anymore that the two surfaces are spheres, the previous sources of errors cause the red curve.

This simple example illustrates the fact that a geometrical representation of the model is necessary to model specular reflections by ray tracing. This justifies the interest of the geometrical method concerning the radiative component of heat transfer.

7.1.4 Mirrors and collector - Heat fluxes

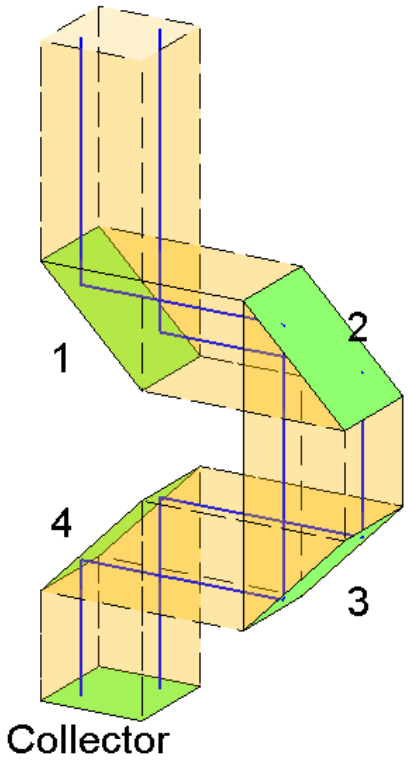
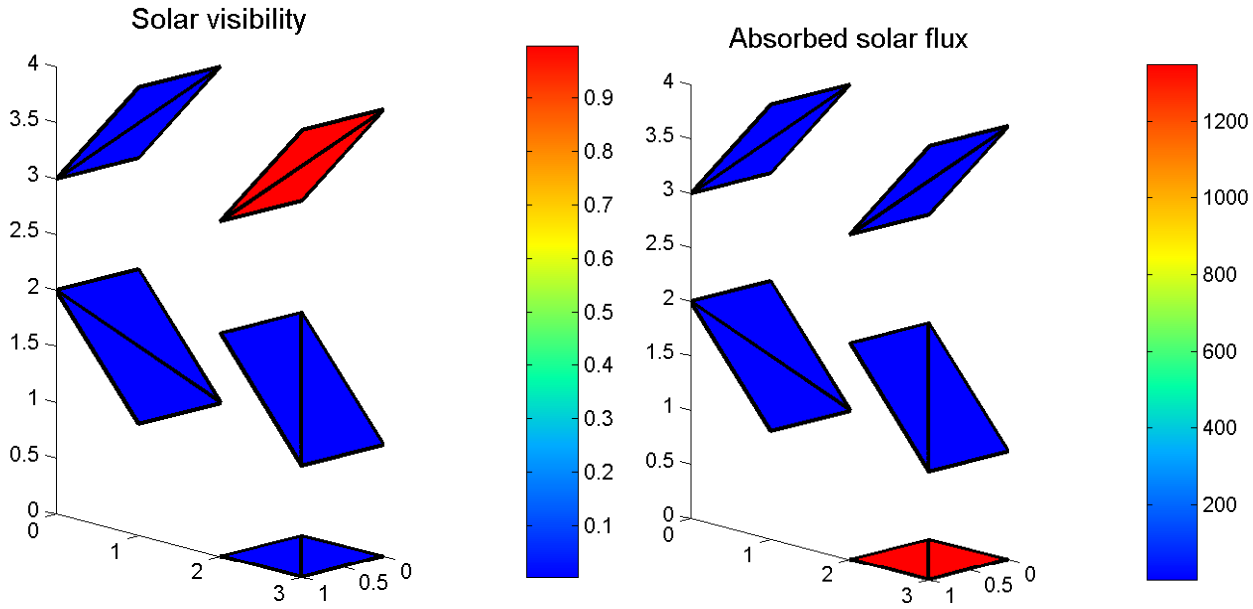


Figure 7.5: Mirrors and collector - Geometrical configuration.

The next test case has been selected to illustrate the computation of solar heat fluxes. It is made of a succession of four purely specular mirrors (specular reflectivity ρ^s equal to 1) and one ideal solar collector (solar absorptivity α equal to 1). Mirror 1 is illuminated by the Sun and is characterized by a solar visibility of 100%; the other surfaces cannot directly be seen from the Sun (the solar visibility is shown in Figure 7.6–a). The dimensions of the last three mirrors and solar collector are fixed in order to exactly fit the solar beam reflected by mirror 1 (see the orange volume, limited by dashed lines in Figure 7.5, which materializes the solar beam).

This simple example can be used to check the accuracy of the heat flux computation. In this case, the direct heat flux and absorbed heat flux coincide, as the collector's absorptance is equal to 1. Figure 7.6–b represents the resulting distribution of the heat flux. The four mirrors receive no heat flux, *i.e.* no direct and no absorbed heat flux, as they are purely specular, as a direct heat flux can only be associated with a (at least partially) diffuse reflector; specularly reflected heat flux is not assigned to the specular reflector but to the next encountered diffuse surface. The collector exactly receives the power incident on mirror 1; the incoming heat flux is conserved.



(a) Solar visibility

(b) Heat flux

Figure 7.6: Mirrors and collector.

7.2 Large test cases

The second class of test cases is made of "large scale" models, for which no analytical solution is available. The results of our algorithm can be compared with the results generated by other software, as Esarad-Esatan and Samcef.

7.2.1 QuartSatellite - Adjoint equations

In order to illustrate the use of the adjoint equations, the model `QuartSatellite`, presented in references [VdKPB08a] and [VdKPB08b], has been considered. The model represents a quarter of the inside of a spacecraft. It is composed of 12 geometrical primitives (cylinders, rectangles and spheres) modeling a tank, an electronic box, a cylinder and two panels. The finite element mesh contains 1 100 nodes and 1 800 triangles. This model has been solved with the proposed algorithm Pvtan, as well as with the industrial tool Samcef [Doc07] and Esarad-Esatan [Doc06], allowing us to guarantee that the obtained results are correct. The height of the model is equal to $700mm$ while the two other dimensions are equal to $300mm$.

7.2.1.1 Boundary conditions

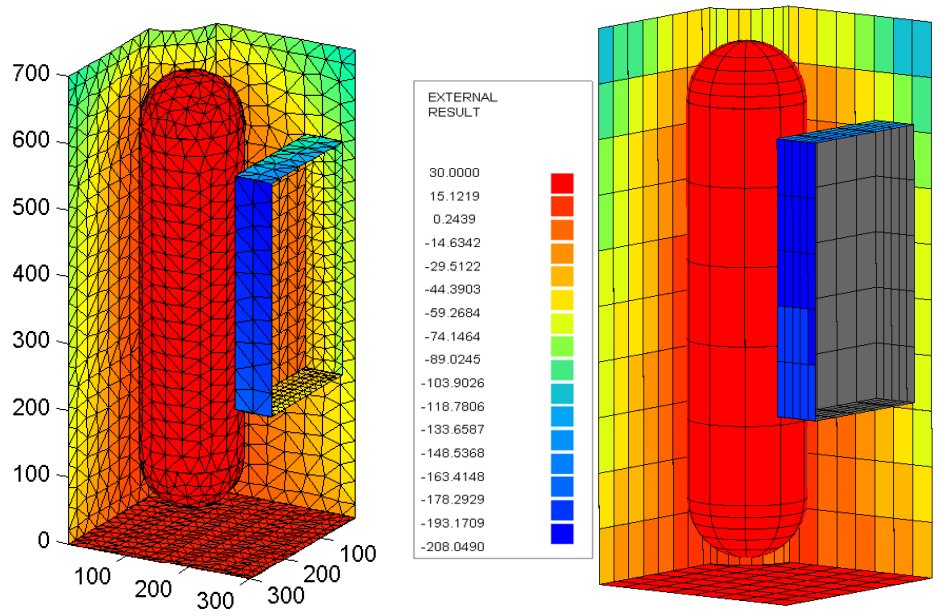
The geometry is defined and meshed with Samcef. The view factors are computed separately with the geometrical method. The temperature of the basis is fixed at $293K$ while the tank is maintained at $303K$. The other components are assumed to be adiabatic. An additional node represents the cold space and is set at $0K$. The basis of the satellite, the cylinder, the reinforcements and the electronic box are characterized by a coating `coating1` while the tank presents a coating `coating2`.

Coating	Emittance	Diffuse reflectance	Specular reflectance
`coating1`	0.95	0.05	0.
`coating2`	0.5	0.5	0.

Table 7.2: Thermo-optical properties.

7.2.1.2 Temperature distribution

The use of adjoint equations has been presented as a post-processing step. It is based on a previous computation of the temperatures and heat fluxes. The geometrical method has been used to compute the temperature distribution through the model. The result is displayed in Figure 7.7–a. Esarad and Esatan are also used to compute the temperature distribution; the result is presented in Figure 7.7–b. A discussion about the accuracy of these two results has been performed in reference [VdKPB08b].



(a) Pvtan solution (b) Esarad solution

Figure 7.7: Temperature distribution.

7.2.1.3 Thermal radiosity, radiative energy and importance

The use of adjoint equations is based on the computation of two quantities: thermal radiosity and importance.

Once the thermal problem is solved, the temperatures and radiative heat fluxes of the surfaces are accessible. Using relation (2.32), the distribution of thermal radiosity throughout the model can be computed (see Figure 7.8–a). The basis presents the highest radiosity, as it is the hottest surface and is characterized by a high emittance. The radiosity of the tank is not constant, although it is isothermal, because it is characterized by a reflectance of 50% and reflects an important fraction of the received radiation. The contours of the electronic box are associated with nearly-zero radiosity, given that they do not interact with the rest of the satellite and emit their energy to the cold space.

On the basis of the thermal radiosities, equation (6.36) can be used to compute the radiative energy associated with the model. We obtain:

$$v(J) = 288.872W \tag{7.7}$$

In heat transfer for diffuse surfaces, the reception vector is equivalent to the area vector A_i (see Section 6.2). The importance associated with each element is function of the initial area of the element, and can highly vary, in function of the quality of the discretization. If the evolution of surface importance is displayed (see Figure 7.8–b), it appears more

continuous. Given the definition of the reception vector, the surface importance is always higher or equal to unity.

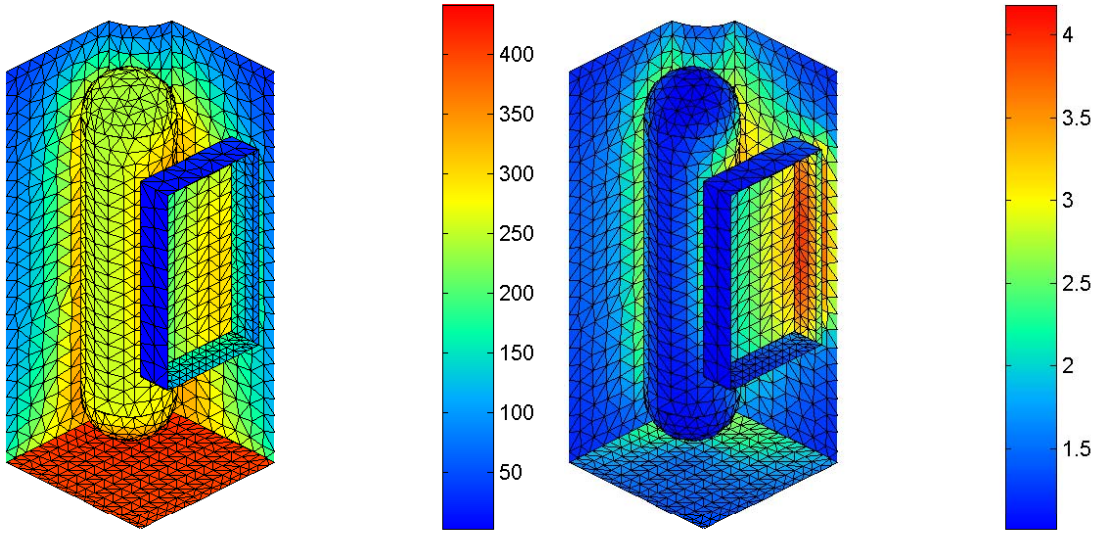


Figure 7.8: Importance distribution – a) thermal radiosity – b) surface importance.

In order to compute the importance, an iterative scheme has been implemented, on the basis on the examination of an error criterion. The convergence of this routine is illustrated in Figure 7.9. The exponential diminution of the residual, linked to the attenuation due to the transport operator can be observed.

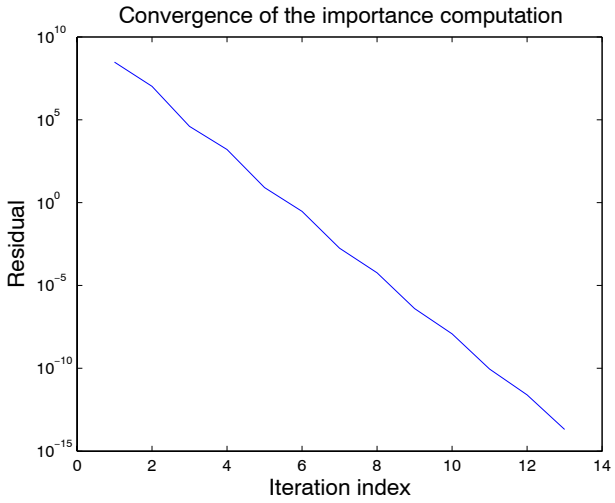


Figure 7.9: Convergence of the importance computation.

7.2.1.4 View factors

The stratified hemisphere method has been used to compute the view factors between the 12 geometrical primitives. 10 000 rays have been traced from each triangle. The confidence interval is fixed to 99%. For each pair of view factors, equation (3.24) has been used to compute the associated relative error.

It is our belief that this level of accuracy could be not necessary for thermal applications, where conductivity will induce heat fluxes between adjacent elements and reduce the surface temperature gradients caused by different irradiations. Then, elementary view factors have been integrated onto the geometrical surfaces. Here, to simplify the analysis of the results, the five sub-systems have been grouped: the basis, the support cylinder, the planar supports, the tank and the electronic box. Finally, a matrix of 5×5 radiative links is obtained.

The view factor matrix has been compacted in order to give the view factors from one subsystem to the others. The absolute errors of the view factors are given in Table 7.3.

[−]	Base	Cylinder	Supports	Tank	Box
Base	0.0000	$1.6979\,10^{-4}$	$1.3977\,10^{-3}$	$6.4148\,10^{-4}$	$8.7071\,10^{-4}$
Cylinder	$5.4881\,10^{-4}$	0.0000	$1.5002\,10^{-3}$	$1.7719\,10^{-3}$	$3.8328\,10^{-4}$
Supports	$2.0300\,10^{-4}$	$1.8044\,10^{-4}$	$1.8264\,10^{-5}$	$1.8654\,10^{-4}$	$3.6622\,10^{-4}$
Tank	$1.9220\,10^{-3}$	$1.2843\,10^{-3}$	$3.4213\,10^{-3}$	0.0000	$1.3541\,10^{-3}$
Box	$1.9441\,10^{-4}$	$3.9061\,10^{-5}$	$6.0642\,10^{-4}$	$3.6891\,10^{-4}$	0.0000

Table 7.3: Absolute view factor error, from one subsystem to another.

A non-zero view factor exists between the supports, as they can see each other. This link is associated with a low but non-zero value. It appears that the planar surfaces are associated with small errors while the curved ones could require additional rays to reduce the associated error.

7.2.1.5 Energy error

As the relative energy error is a matrix, one map can be displayed for each particular surface *i.e.* for each line of the matrix. To simplify the display of the results, the relative error on the radiative heat flux is cumulated, from one sub-system to another. This error is reported in Table 7.4.

[–]	Base	Cylinder	Supports	Tank	Box
Base	0.0000	$1.2955\,10^{-6}$	$7.6923\,10^{-6}$	$8.0509\,10^{-6}$	$3.4152\,10^{-6}$
Cylinder	$1.5905\,10^{-4}$	0.0000	$2.1802\,10^{-4}$	$5.8724\,10^{-4}$	$3.9698\,10^{-5}$
Supports	$1.9980\,10^{-4}$	$1.2347\,10^{-4}$	$9.0150\,10^{-6}$	$2.0997\,10^{-4}$	$1.2883\,10^{-4}$
Tank	$1.0587\,10^{-3}$	$4.9182\,10^{-4}$	$9.4503\,10^{-4}$	0.0000	$2.6656\,10^{-4}$
Box	$8.1720\,10^{-5}$	$1.1415\,10^{-5}$	$1.2783\,10^{-4}$	$1.7733\,10^{-4}$	0.0000

Table 7.4: Relative energy error – cumulated error, from one subsystem to another.

The energy exchanges from the tank to the other surfaces are the most inaccurate. This is due to the curvature of the initial surface [Pla97] and its higher radiosity (see Figure 7.8–a). A non-zero link exists between the supports, because this combined surface is not convex. The tank is associated with the larger errors and could require additional rays to be traced to achieve a better accuracy.

7.2.2 XEUS model

XEUS, **X**-ray **E**volving **U**niverse **S**pectroscopy, was a mission initiated by ESA. In July 2008, the **I**nternational **X**-ray **O**bservatory (IXO) mission supersedes the XEUS mission concept. IXO is a joint X-ray observatory with participation from ESA, NASA and JAXA. XEUS was planned to be the successor to XMM-Newton mission, which is the ESA's current X-ray observatory. The objective of XEUS was to investigate how supermassive black holes form and grow, how feedback from these black holes influences galaxy growth, how large-scale structure evolves and the role of dark matter, how the baryonic component of this structure becomes chemically enriched, and how gravity behaves in the strong field limit.

The specifications of the mission require a focal length of $35m$; this could be obtained by two spacecrafts formation flying (see Figure 7.10–a). The first spacecraft (DSC) contains the detector while the second one, noted MSC, hosting the mirror, focus the X-rays on the detector (see Figure 7.10–b).

In this study, the MSC has been used, as it has been modeled in [Che06], with Esarad. The purpose of this Section is to list the different comparisons made between the reference program Esarad and our algorithm, in order to establish its reliability. Comparisons with Samcef are also provided.

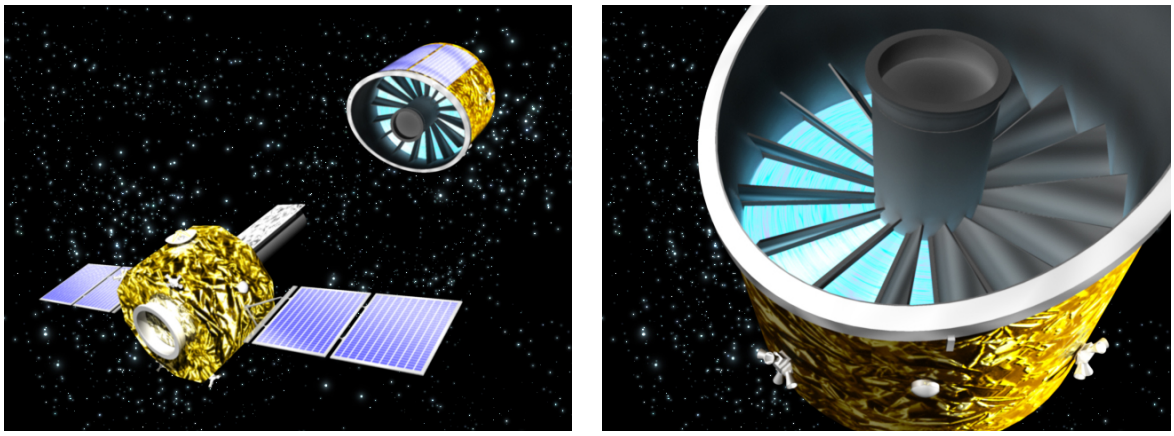


Figure 7.10: XEUS spacecraft – a) spacecraft – b) mirror.

7.2.2.1 Geometrical model

Some modifications have been introduced in the geometrical model in order to obtain a model compatible with our algorithm. The external side of the spacecraft appears in red, which means that the surface is radiatively inactive. The cutting operations have been redefined because they were lost during the conversion of the model. The contact conductances have been removed because they are not modeled with the proposed algorithm (the computation of surface conductance can easily be performed with finite element software, like Samcef; this functionality was not considered for the definition of the thesis's requirements). The holes appearing in the external cylinder represent the extremities of the spokes; as these surfaces are not radiatively active, they have been removed from the model.

The model is composed of 1 156 geometrical primitives (discs, portions of cylinders and rectangles). The internal cylinder (in red in Figure 7.11–a) represents the service module while the external one corresponds to the solar shield (in orange). If the bottom view is considered (see Figure 7.11–b), 16 portions of discs can be seen, called *petals* (in blue), fixed between spokes (in yellow in Figure 7.11) (the bottom and upper views are symmetrical for this model). The geometrical model is common to Esarad and the proposed algorithm. During the pre-process of the model, the mesh-module of Catia has been used to generate a corresponding finite element mesh. This mesh is common to our algorithm and Samcef. To set the dimensions of the model, note that the height of the solar shield is between $2.1m$ and $3.7m$, that the diameter of the shield is about $4.3m$.

7.2.2.2 Characteristics of the model

Among the characteristics of this model, the presence of different meshes can be noted. For example, petals are finely meshed (in order to correctly model the masking effects) while the inner and outer cylinders are coarsely meshed (where a quite uniform temperature

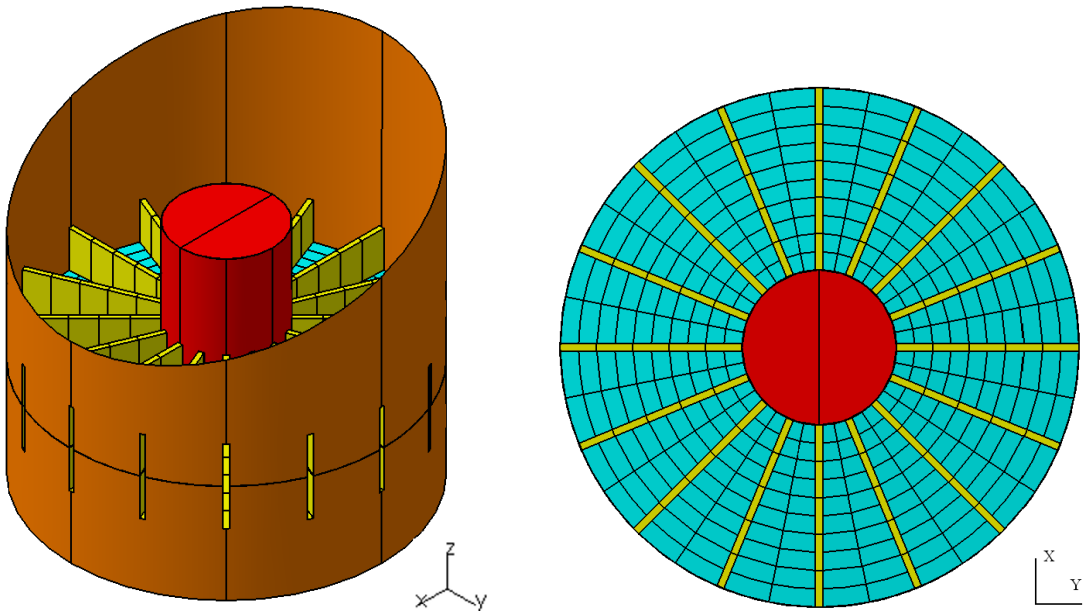


Figure 7.11: Geometry – a) global view – b) bottom view.

field is observed, due to the solar irradiation). The geometry presents also different cutting contours: straight and curved ones. As mentioned in Section 4.3.3, the curved intersection of two surfaces is not exactly computed but it is approximated by straight lines. In this application, we will assess whether our assumption does not introduce appreciable error. Another interest is due to the visibility problem: while some surfaces have a clear visibility with others (for example, from one cylinder to the next one), some surfaces are surrounded by a lot of obstacles (see the elements of the spokes and the petals). The differences of visibility of the space is also interesting, from a thermal point of view, because it has a strong impact on the final temperature. The final interest of this model is linked to the presence of obstacles for the computation of heat fluxes.

7.2.2.3 Space view factor

The first compared quantity is the space view factor, *i.e.* the view factor from a surface to the deep space. This quantity is complementary to the global view factor defined by relation (7.4).

The numerical values of the space view factors from Pvtan have been compared with the numerical values exported from Esarad to Esatan. As the model contains a large number of surfaces (1 156 primitives), only chosen surfaces have been checked (the top and bottom of the central cylinder, the inner and outer cylinders, the first spoke and the first petal, with the +Z and -Z parts). The differences between the individual view factors (maximum and minimum values) are given in Table 7.5. The surfaces associated with large view factors are

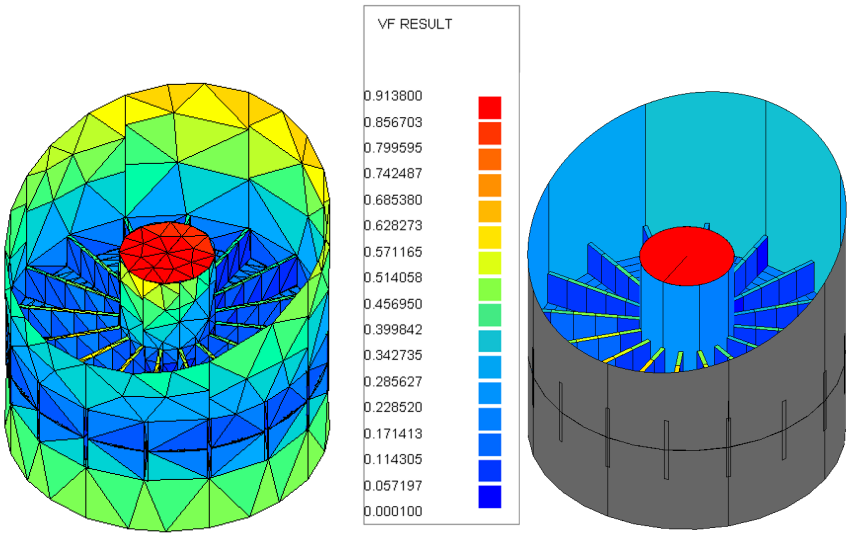


Figure 7.12: Space view factor – a) Pvtan solution – b) Esarad solution.

more accurate than the surfaces with small value of the space view factor. This is in good agreement with the theory developed in the hemisphere method. The largest differences between Esarad and Pvtan are observed on the cut surfaces, which are characterized by non-zero view factors to inactive node in Esarad.

	Maximum error	Minimum error
Top service module	$7.6\,10^{-4}$	$7.5\,10^{-4}$
Bottom service module	$1\,10^{-4}$	$9\,10^{-5}$
Service module	$1.41\,10^{-2}$	$7.3\,10^{-4}$
Solar shield	$4.91\,10^{-3}$	$3.33\,10^{-3}$
Spoke 1	$1.71\,10^{-2}$	$1.1\,10^{-4}$
Petal 1 +Z	$2.67\,10^{-3}$	$1\,10^{-4}$
Petal 1 –Z	$2.67\,10^{-3}$	$7\,10^{-5}$

Table 7.5: Space view factor error.

If the limits of the view factor are considered, it appears that the maximum view factor is located at the top of the service module, at the centre of the spacecraft. It is equal to 0.9138 in Esarad. The corresponding space view factor in Pvtan is equal to 0.9146.

Compared to Esarad, Pvtan gives access to detailed field of view factors across the large surfaces, as the solar shield (see Figure 7.12–a). If a fixed number of rays are traced per unit area, it does not require additional computational effort w.r.t. Esarad (Figure 7.12–b).

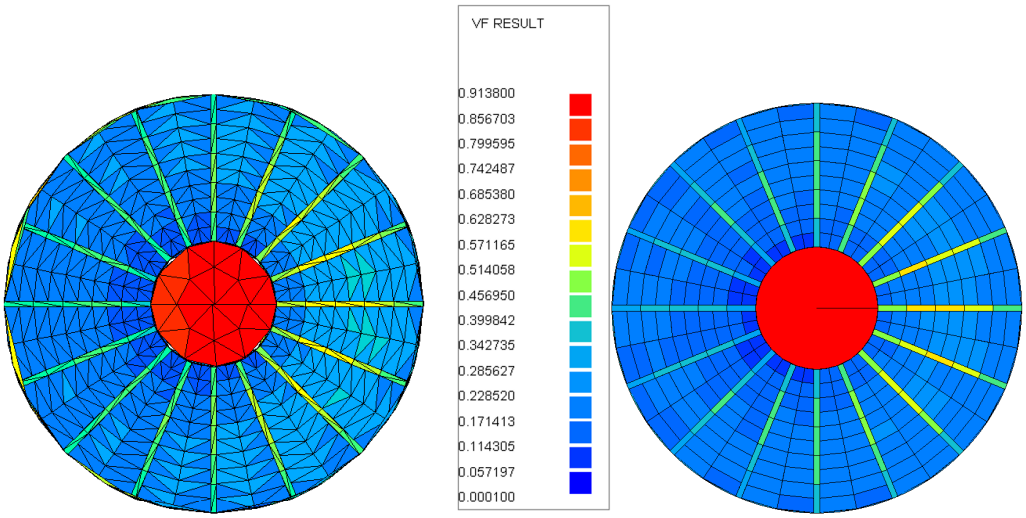


Figure 7.13: Space view factor – a) Pvtan +Z view – b) Esarad +Z view.

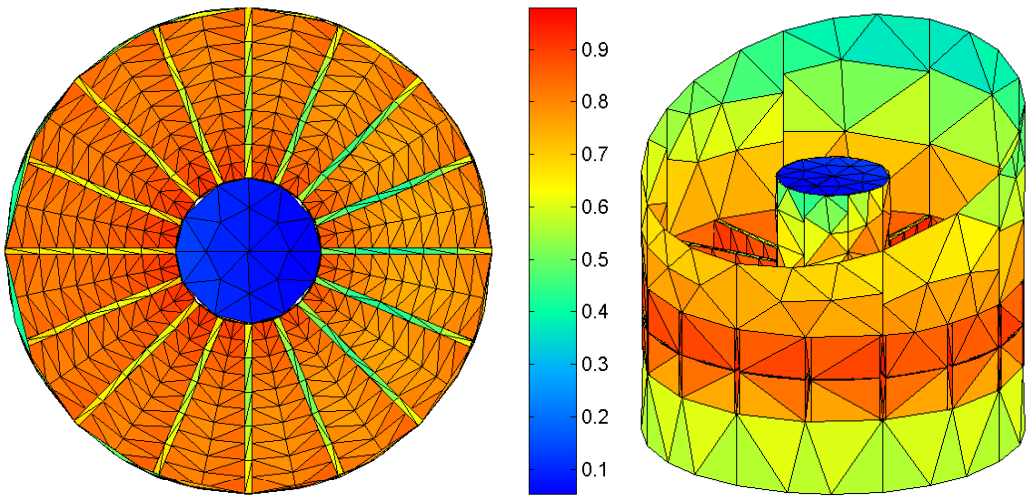
From the results, it appears that the space view factor is larger at the bottom of the solar shield, as there are fewer obstacles in this part of the model. The more we go to the petals, the more obstacles are encountered, the less space view factor are observed. A similar observation can be done when considering the internal cylinder (the service module). The top of the service module presents a higher space view factor than the rest, due to a larger visibility to deep space.

In Figures 7.13–a and 7.13–b, the non-symmetrical distributions of the space view factor across the spokes and the petals can be observed. These distributions (where gradients are limited, contrary to Figures 7.12) are identical for the two programs.

7.2.2.4 Global view factor

Samcef displays the global view factors (see equation 7.4). Here, the results from our algorithm are compared with the view factors computed by Samcef.

The source code of Samcef has been modified in order to extract the view factors. Matlab is used to display the global view factors. Figure 7.14–a represents the distribution of the global view factors throughout the model, computed by our algorithm. Figure 7.14–b corresponding to the solution of Samcef. As the two distributions are similar, two different views have been chosen. The limits of the scale are the same for the two distributions. The values obtained with Samcef are compared with the values of Pvtan. The differences between the two distributions do not exceed 1%, which is in good agreement with the desired accuracy.



(a) Pvtan solution (b) Samcef solution

Figure 7.14: Global view factor.

	Maximum error	Minimum error
Top service module	$8.54\,10^{-4}$	$7.54\,10^{-4}$
Bottom service module	$1.40\,10^{-3}$	$1.16\,10^{-3}$
Service module	$2.89\,10^{-3}$	$1.73\,10^{-3}$
Solar shield	$2.73\,10^{-3}$	$1.09\,10^{-3}$
Spoke 1	$8.53\,10^{-3}$	$7.07\,10^{-5}$
Petal 1 +Z	$3.65\,10^{-3}$	$1.75\,10^{-4}$
Petal 1 –Z	$4.07\,10^{-3}$	$2.51\,10^{-5}$

Table 7.6: Global view factor error.

7.2.2.5 Radiative exchange factors

Once again, the numerical values computed by Pvtan have been compared with the values exported from Esarad to Esatan. The correspondence is very good; the difference is below 1%, which is in good agreement with the desired accuracy. In the same way, the results extracted from Samcef are close to the results from Pvtan, with differences inferior to 1%. For the sake of simplicity, the results of the comparisons with Esarad and Samcef are not displayed in this document.

7.2.2.6 Solar absorbed heat flux

The next presented result is the solar absorbed heat flux. In Gebhart's formulation, secondary rays in the ray tracing experience diffuse reflection; the attenuation of the secondary rays is slower than for direct heat flux.

The proposed algorithm gives access to a gradient of values across the large surfaces (inner and outer cylinders), yielding larger bounds of the color scale if the values are displayed for each triangular finite element (see Figures 7.15–a and –b). To compare these results with Esarad, the values have to be integrated onto the surface primitives, *i.e.* the thermal nodes of the Esarad model (see Figures 7.15–c and –d). If the numerical values of the primitives are compared, a good agreement can be found for absorbed heat fluxes. The maximum heat flux is equal to $811W$ for Esarad; it is equal to $812W$ for Pvtan.

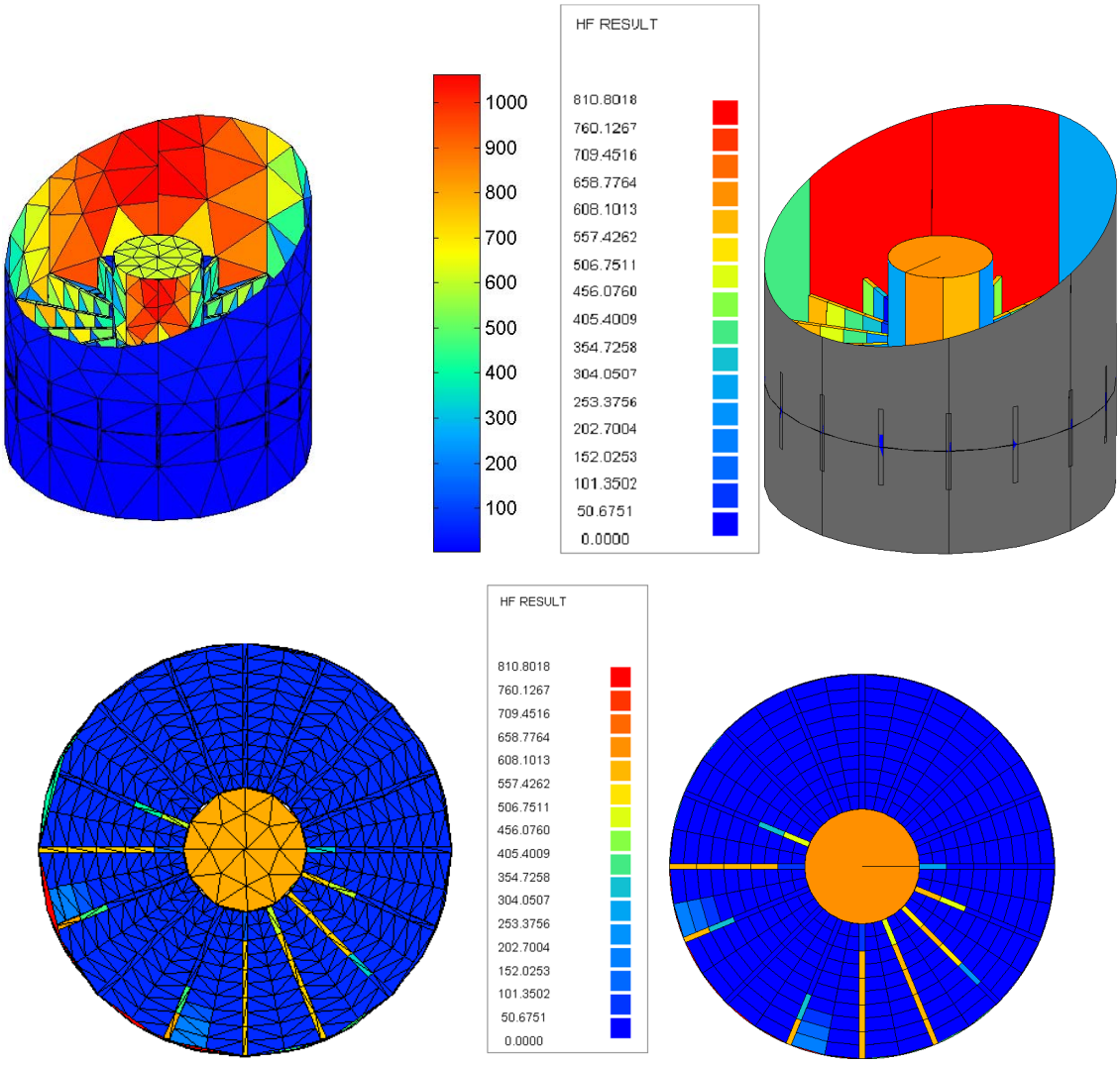


Figure 7.15: Absorbed heat flux – a) detailed Pvtan result – b) Esarad result – c) Pvtan result integrated onto Esarad thermal nodes – d) Esarad result.

7.2.2.7 Temperatures

The comparison of the temperatures computed by Pvtan and Esarad is given in Figures 7.16–a and –b (in degrees Celsius). While Figure 7.16–b represents the temperature of the different thermal nodes defined in Esarad, Figure 7.16–a associates a temperature with each triangle. The color scale is more extended for the Pvtan result (from $-125.4°C$ to $148.7°C$) than for the Esarad result (from $-119.6°C$ to $137.2°C$). Based on the triangulation of the large geometrical primitives (outer and inner cylinders), a detailed variation of the temperature is accessible. A gradient can be observed across the large thermal nodes of Esarad, which were assumed to be isothermal. For example, if the larger thermal nodes of the outer cylinder are considered, a gradient of around $40°C$ is observed. The observed gradients are coherent with what was expected.

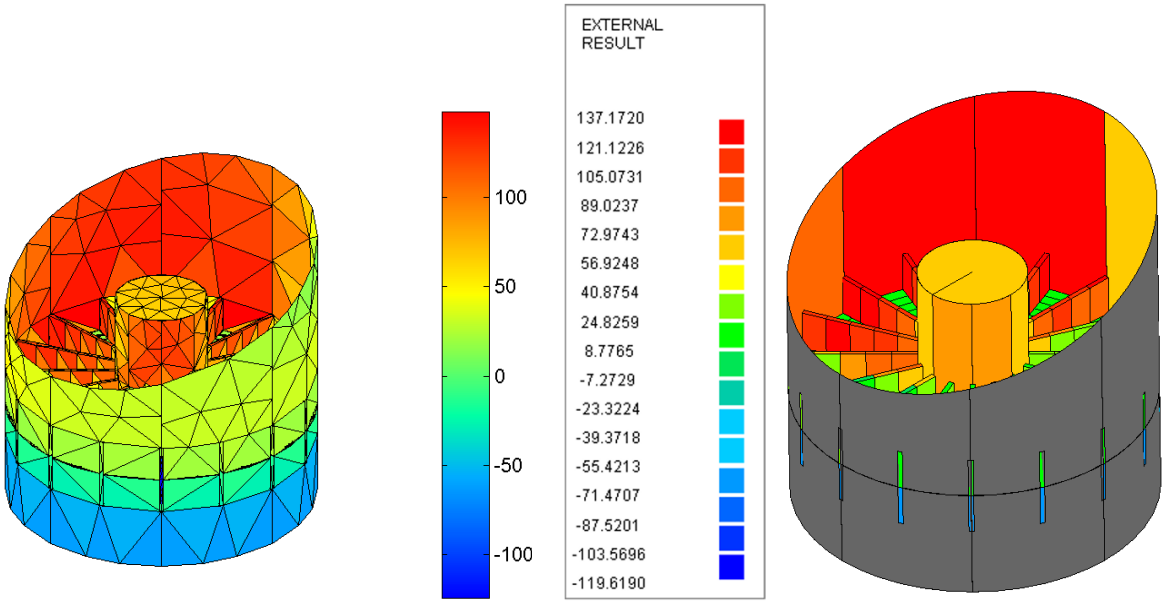


Figure 7.16: Temperatures [degrees Celsius] – a) Pvtan solution – b) Esarad solution.

7.2.3 Finite element view factors

In this work, triangles of the first degree have been considered but elements of higher degree can also be used. Uniform view factors and finite element view factors of the first degree are presented here. Figures 7.17–a and –b represent the global view factor (the complementary value of the space view factor) with uniform and first-order elements. At the first sight, the two distributions seem to be identical but the second result (Figure 7.17–b) is more detailed. In Figure 7.17–a, it seems that the view factor field roughly follows an annular distribution. This distribution is more straightforward with the finite element view factor

(see Figure 7.17–b).

The uniform view factors have been obtained with the stratified hemisphere method and three Gauss quadrature points (see Section 3.1.7) while the first order finite element view factors have been computed by placing one stratified hemisphere on each node (see Section 3.1.7.6) and then interpolating the nodal values across the elements. This model contains 8 620 triangles and 6 345 nodes. The computation of the uniform view factors required 258 600 000 rays (10 000 rays from each Gauss point, with 3 Gauss points per triangle) while the computation of the first order view factors only required 63 450 000 rays.

From a performance point of view, the computation of the finite element view factors does not require additional CPU in Pvtan. This point is checked in Section 7.3.6, where the computation time of our algorithm is compared with Samcef, which computes uniform view factors. In terms of storage, as the number of points is a bit larger than the number of surfaces, more space is needed. But the compression process (storage of non zero-terms only) can reduce the impact on memory.

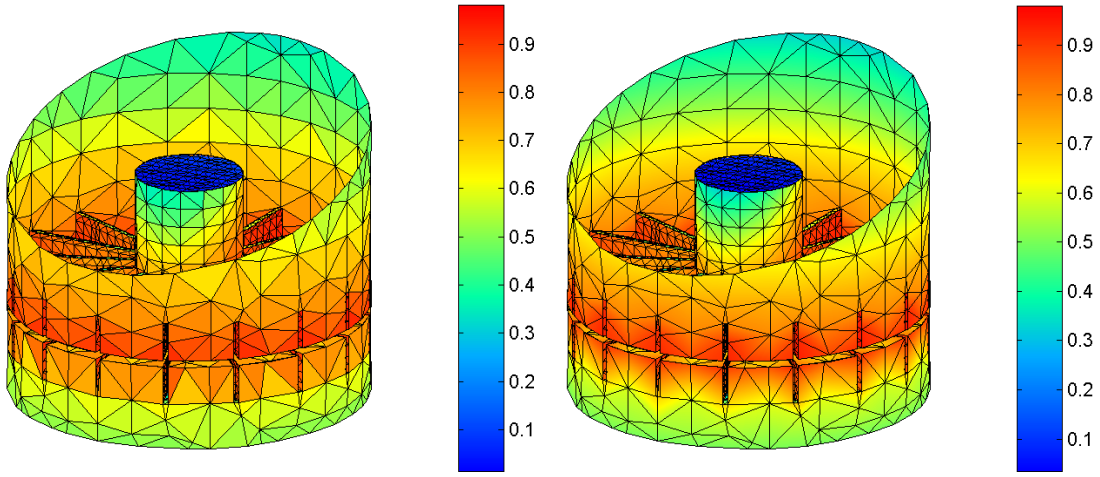


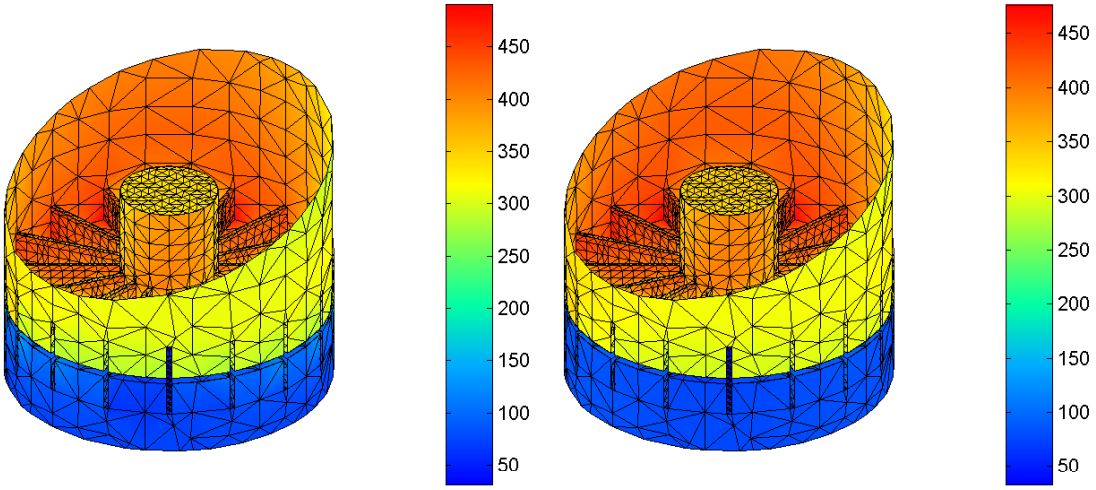
Figure 7.17: Finite element view factors – a) uniform view factors – b) first order view factors.

7.2.4 Combined radiation and conduction

Here, the interest of the geometrical method is illustrated by combining radiation with conduction. Due to a limitation of our formulation, the only encoded boundary condition is fixed net flux; all the surfaces can vary in temperature. This limitation is not linked to the geometrical method but only to our implementation of the thermal formulation. An incoming solar heat flux of $1\,350 W/m^2$ has been simulated. An iterative scheme has been encoded. First, the temperatures without conduction are computed (see Figure 7.18–a).

Then this distribution of temperature is used to initiate the iterative process which solves the temperature distribution including conduction. With a thickness of $1cm$ and a homogeneous conductivity of $500W/m/K$, the following temperature distribution is obtained (Figure 7.18–b). The limits of the temperature's scale are reduced (from $29.6K$ to $491.0K$ without conduction to $31.5K$ to $477.4K$ with conduction). This is due to the smoothing effect of conduction. The main difference between the two distributions of temperatures are mainly located at the top of the spacecraft, along the shadow caused by the inner cylinder on the outer cylinder. Differences are also observed along the shadows of the spokes on the petals. This is due to the fact that, if only radiation is considered, the shadowed nodes are cold while the adjacent, illuminated nodes, are warm. The conduction smoothes this distribution of temperatures; local temperature gradients are decreasing.

From these results, it appears that in this case, radiation is the predominant mode of heat transfer. The distribution of temperatures is mainly due to the solar irradiance, which causes sharp shadows, especially on the outer cylinder.



(a) Radiation only (b) Combination with conduction

Figure 7.18: Temperature distributions.

7.3 Mathematical behaviour

In this Section, the mathematical behaviour of the proposed algorithm is studied. Different parameters are considered, as the number of surfaces, the number of traced rays and the resolution of the USD voxels n_x; the evolution of the CPU time required to execute the ray tracing is examined. The convergence of the algorithm when the number of rays increases is also checked.

7.3.1 Number of rays

In this Section, different numbers of rays are considered, that is different resolutions of the stratified hemisphere. Figure 7.19–a represents the evolution of the CPU time when the number of rays increases. It points out the linear behaviour which is linked to the fact that each new ray is considered independently from the others. Figure 7.19–b shows the convergence of the algorithm. The error criterion is built as follows:

- A reference solution is computed by tracing a large number of rays, with a stratified hemisphere's resolution of 400×400.
- For each hemisphere's resolution,
  - For each element, *i.e.* for each line of the view factor matrix, the global view factor G is computed;
  - For each surface k, the elementary global view factors G are integrated in order to obtain the surface view factors, noted F_k;
  - The error is computed thanks to the following relation:

$$\epsilon = \sum_{k=1}^{N_{\text{surf}}} \left(F_k|_{\text{ref}} - F_k|_{\text{hemis}} \right)^2 \tag{7.8}$$

It can be seen in Figure 7.19–b that the convergence of Pvtan is roughly a linear function of the logarithm of the number of traced rays.

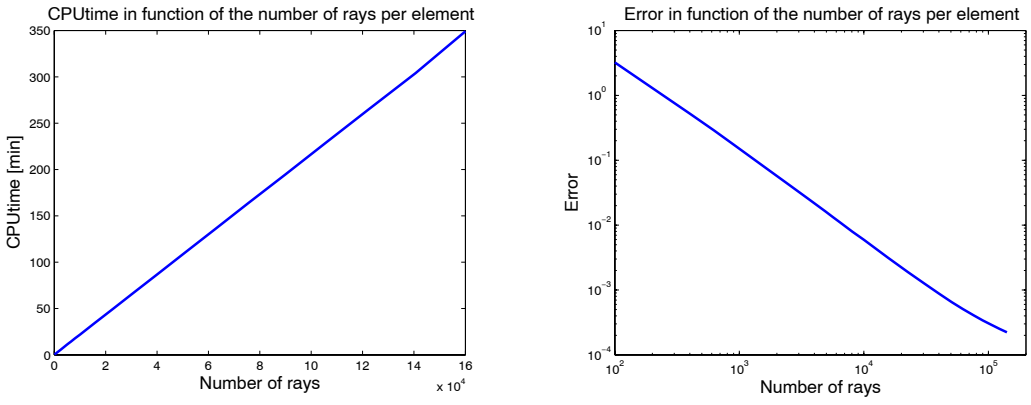


Figure 7.19: Influence of the number of rays – a) CPU time – b) convergence.

7.3.2 Numbers of triangles and surfaces

The two next considered parameters are the numbers of triangles and surfaces defining the geometrical models. Different versions of the same geometrical model are considered,

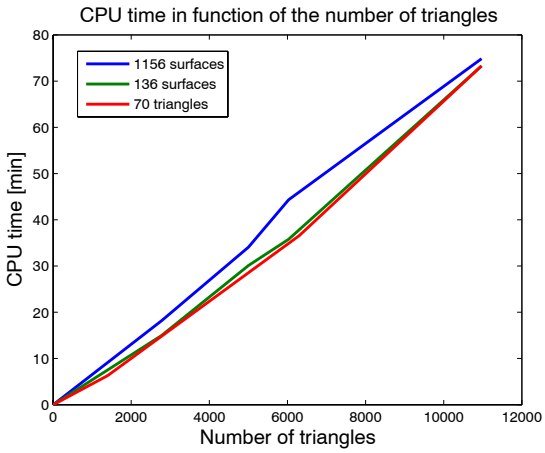


Figure 7.20: Numbers of triangles and surfaces.

with different meshes. Some surfaces are combined in order to reduce the total number of surfaces[2]. Figure 7.20 shows the evolution of the CPU time in function of the number of triangles, for three geometrical models differing by the number of surfaces (the characteristics of the different models are given in Table 7.7. The evolution of the CPU time is a roughly linear function of the number of triangles (if the resolution of the voxels is adapted to each model). The CPU time is roughly insensitive to the number of surfaces (if the resolution of the geometrical cells is adapted to each mesh).

Model's name	Surfaces' number	Elements' number	Mean density of the cells
Model 1–a	70	1 398	19.971
Model 1–b	70	6 301	90.014
Model 1–c	70	10 970	156.714
Model 2–a	136	2 776	20.412
Model 2–b	136	5 001	36.772
Model 2–c	136	6 031	44.346
Model 2–d	136	10 970	80.662
Model 3–a	1 156	2 776	2.401
Model 3–b	1 156	5 001	4.326
Model 3–c	1 156	6 031	5.217
Model 3–d	1 156	10 970	9.490

Table 7.7: Characteristics of the three models of Figure 7.20.

[2]We used a Nastran conversion of the Esarad model. In this version, each thermal node (Esarad elementary patch) of a same surface is considered as a new, independent surface. Different nodes can be merged into one surface, reducing the total number of surfaces.

7.3.3 Resolution of the USD voxels

Figure 7.21 represents the evolutions of the CPU time (upper curve) and the density (lower curve) that is, the average number of surfaces per voxel. An increase of the voxel's resolution n_x yields a strong decrease of the CPU time, followed by a constant level when the resolution n_x is higher than 15. If the second curve is considered, a decrease of the density is observed when the resolution increases. The decrease of the density is a roughly linear function of the logarithm of n_x, for small values of n_x.

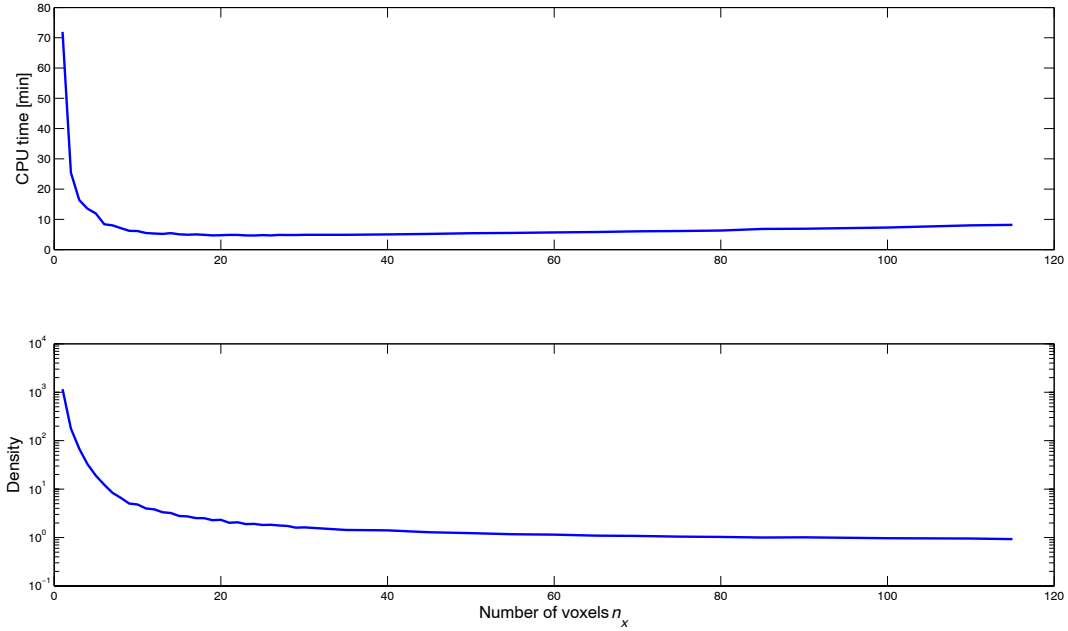


Figure 7.21: Resolution of the USD voxels.

7.3.4 Resolution of the geometrical cells

Figure 7.22 represents the evolutions of the CPU time and the density in function of the resolution of the geometrical cells. The considered case is characterized by a low initial density (see the lower curve); it means that the number of elements is similar to the number of surfaces. The acceleration due to the geometrical method is of limited impact in this case; the main acceleration is due to the spatial decomposition in voxels. A slight decrease of the CPU time is observed, followed by a constant level. For large values of the resolution, the CPU time can exceed the initial CPU time.

It appears that a choice of 20 for the resolution seems to yield acceptable CPU time, for every considered configurations.

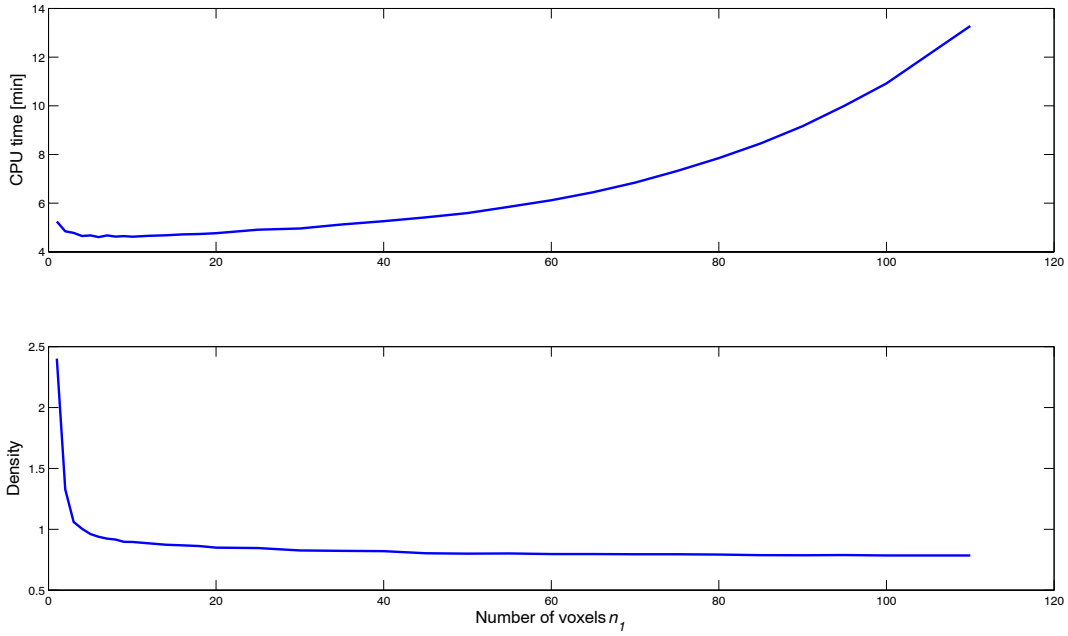


Figure 7.22: Resolution of the geometrical cells.

Remark: In this case, the impact of the first level of acceleration is limited because of the low initial density. In the case of a model with a large initial density, the first level will be more efficient. This two-level approach yields a global acceleration which is insensitive to the two initial densities (the finite element one, linked to the first level, and the geometrical primitives, associated with the second level). It adapts itself to the considered geometrical configuration and associated finite element mesh.

7.3.5 Comparison with Esarad

Our algorithm has been compared with Esarad, for a comparable number of surfaces, based on the geometrical definition of the model (see Figure 7.23–a). Different number of rays have been tried and the evolution of the computational time has been reported. It is represented in the next Figure. It appears that our algorithm is faster than Esarad for the considered test case.

7.3.6 Comparison with Samcef

To conclude the study of the performances of our algorithm, it has been compared with Samcef for a geometry made of triangles (see Figure 7.23–b). Our algorithm is still a little slower than Samcef; the proposed algorithm is 1.25 times longer than Samcef. Maybe it is due to the intersection routine, which returns more information ($\{u, v\}$ parameters of the impact to compute the shape functions). Not computing $\{u, v\}$ can simplify the

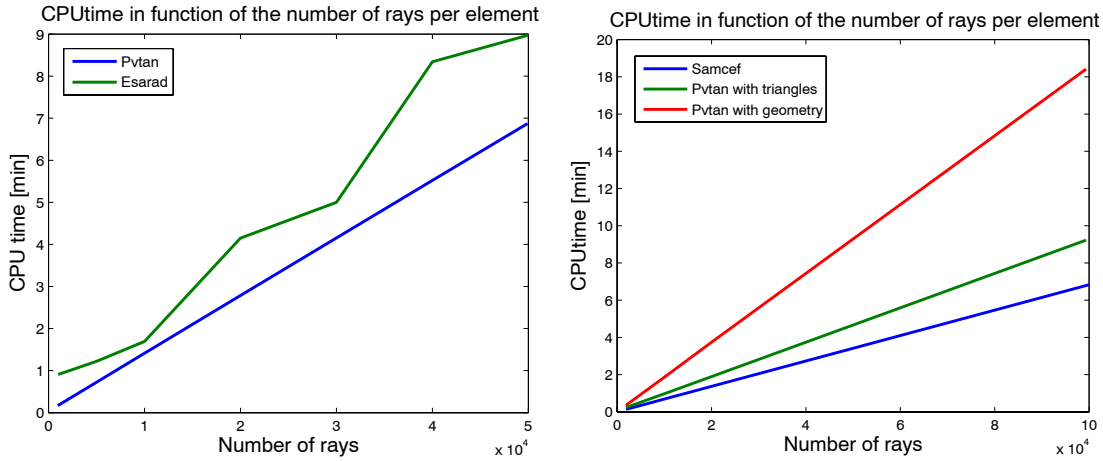


Figure 7.23: Comparison with reference software – a) Esarad – b) Samcef.

acceleration routine. Another point is that Samcef is an industrial, optimized tool, while the algorithms presented in this thesis are used for prototyping purposes. An optimisation step is still necessary.

If geometrical primitives are considered, the CPU time increases (see the red curve in Figure 7.23–b). The introduction of the geometry, which is not currently feasible in Samcef, doubles the computation time of Pvtan. It is straightforward that the ray-surfaces intersections are more complex for geometrical primitives like cylinders, discs and spheres than for triangles. In order to save computation time, the geometry could be used only where it is necessary (for optical system, specular surface and caustics) and approximates the other surfaces with triangles and/or quadrangles, as it is done in Samcef.

These results have been obtained with the XEUS test case. Another test case could yield different ratios between the different methods.

Chapter 8

Conclusions and perspectives

The main objective of this PhD research project was to develop a new ray tracing algorithm for radiative heat transfer, roughly having the same functionality as the reference software Esarad. The proposed algorithm computes all relevant radiative quantities, such as exchange factors and heat fluxes.

After briefly reviewing the requirements presented in Section 1.3, we can conclude the following.

Requirements 1 and 2 are fulfilled: it is a ray tracing algorithm, and the new algorithm computes the relevant exchange factors and external heat loads.

Requirement 3 is fulfilled: the algorithm is mathematically well-behaved, which has been checked with the Esarad unit test cases and illustrated with the XEUS case (see Section 7.3).

Requirement 4 is fulfilled: the different surface phenomena - diffuse, specular and glossy reflection - have been addressed in Section 3.2; several test cases have been implemented for diffuse and specular reflections.

Requirement 5 is partially fulfilled: the support for two spectral bands and the extension to multiple wavelength bands (e.g. for cryogenics applications) have been considered in the mathematical part (appendix A); applications have been implemented in internal report [Vue06a] but not in the present document, for the sake of brevity.

Requirement 6 is fulfilled and even more functionality has been developed: both finite element and lumped parameter formulations for heat transfer model are supported. Although our initial objectives concerned radiative heat transfer only, we developed a ray tracing acceleration technique which allows an easy integration of radiation with conduction, based on a finite element representation of the initial geometrical model. This technique allows us to bridge the gap between the common methods used in space thermal

engineering and the finite element formalism. The definition of finite element view factors, based on the demonstration of Section 6.1.5.1, and its use with the hemisphere method yield a very efficient ray tracing algorithm giving access to a detailed field of view factors with a very limited amount of additional computation time.

The ray tracing performance obtained with the new algorithm has been compared with Esarad and Samcef, and presents comparable results, without dedicated performance tuning. This implies that an efficient industrial implementation is feasible.

Requirement 7 is fulfilled: statistical accuracy control is included in the new algorithm. We have carefully addressed the computation of exchange factors by ray tracing. The stratified hemisphere method (presented in Section 3.1) has been designed in order to compute diffuse view factors and, in a more general way, to obtain the visibility from a diffuse surface. Characterized by a convergence superior to other random processes, the hemisphere method is associated with a statistical accuracy control (SAC) feature linking the desired accuracy level (maximum relative error ϵ and confidence interval α) to the geometrical configuration (view factor) and the number of traced rays.

On the basis of the statistical accuracy control developed for the hemisphere, we have established a SAC for the solar heat flux. Concerning the planetary heat fluxes, we propose to use a variant of the hemisphere method, where the deterministic grid is defined between $0°$ and γ_P, where γ_P is the angular aperture of the planet. The SAC of the hemisphere method can then be used to govern the computation of planetary heat fluxes (albedo and infrared) in one single ray tracing process.

Requirement 8 is fulfilled: the algorithm enables to correct the exchange factors as a function of the computed variance. We have studied the extension of view factors to handle more complex surface phenomena (specular reflection, transmission, non-diffuse emission). We compared different ways to enforce reciprocity and closure, yielding a correction method for the exchange factors depending on their associated variance (see Section 3.3).

Requirement 9 is partially achieved: with the new algorithm a substantial performance improvement for the execution of a thermal-radiative analysis case can be realized compared to using classical Monte Carlo ray tracing methods (as e.g. implemented in Esarad). The performance improvement is achieved thanks to the stratified hemisphere method, which requires less rays to obtain a given accuracy. Computing view factors rather than radiative exchange factors requires less extensive ray tracing steps because no diffusely reflected rays need to be traced and the other reflected rays are more rapidly attenuated. The full radiative heat transfer characterization is still obtained thanks to Gebhart's matrix method that we extended in Section 3.4 or by using the radiosity formulation, based on view factors, presented in Section 2.4. The last key feature is to optimally combine the different ray tracing process steps needed to execute a thermal-radiative analysis case for a complete space trajectory, which is usually one whole orbit around a planet. With

a classical space thermal analysis tool for every orbit position at least three ray tracing steps need to be performed: for solar, albedo and planetary infra-red heat fluxes. We can reduce this number by combining the two planetary heat flux computations in a single ray tracing step. If multiple spectral bands need to be considered, a classical tool needs one ray-tracing step per band. With the new algorithm it is possible to reduce the computation to one single ray tracing (see Section 3.2) with only some additional ray tracing when specular reflection is present in the model. Overall both the computer processing and memory requirements are significantly reduced with the new algorithm, however it remains to be established whether "an order of magnitude" better performance can be achieved.

A last perspective is based on the establishment of adjoint equations for radiosity and importance (see Appendix B.2). This set of equations, jointly used with the measure of the geometrical error induced by the stratified hemisphere method, enables the derivation of a measure for the error in the radiative heat flow between any two surfaces. These equations are expected to form the basis for effective usage or adaptation of more methods developed in digital image synthesis to the field of numerical radiative heat transfer. This would be a good subject for a follow-up research activity.

Appendix A

Mathematical formulation

In this appendix, we group the different mathematical formulations of radiative heat transfer used in the framework of this thesis which have been inspired from literature and do not require particular developments.

A.1 Gebhart's formulation

A different formulation has been developed by Benjamin Gebhart in the sixties. While the previous equations are based on the notion of radiosity and the use of view factors, the Gebhart's equations directly link the temperatures of the different surfaces, through the self-emitted power. This formulation has first been presented in the articles [Geb61b, Geb59]. It is also presented in the books [SH01, Geb61a].

The Gebhart's factor[1] B_{i-j} is defined to model the different paths that the light emitted by a surface i can experience before being absorbed by j.

Then, the net radiative power of a surface i is the difference between the self-emitted power (equal to $A_i\epsilon_i E_{b,i}$) and the power received from the other surfaces. This last power is obtained by summing the contributions[2] on all the surfaces which have a non-zero Gebhart factor with the surface i.

$$Q_i = A_i\epsilon_i E_{b,i} - \sum_{j=1}^{N} B_{j-i} A_j \epsilon_j E_{b,j} \tag{A.1}$$

If we consider the radiative heat flux at a point r, it is given by the general formula (2.31):

$$q(r) = \epsilon(r)E_b(r) - \epsilon(r)H(r) \tag{A.2}$$

[1] The Gebhart's factor is named *absorption factor* by Gebhart himself [Geb61a], *transfer factor* in [SH01] when multiplied by the initial emissivity $\mathcal{F}_{i-j} = \epsilon_i B_{i-j}$.

[2] $A_j\epsilon_j E_{b,j}$ represents the power diffusely emitted by a surface j. The fraction B_{j-i} is finally absorbed by the surface i.

The absorbed energy at the point r is the sum of two components, given by the following relation:

$$\epsilon(r)H(r)dA = \epsilon(r)H_0(r)dA + \int_{A'} \epsilon(r')E_b(r')B_{dA'-dA}dA' \tag{A.3}$$

The first term represents the absorbed component of the external irradiation. The second term corresponds to the radiation emitted by the surfaces of the 3D-model, taking multi-reflection into account.

If we use the reciprocity relation of the Gebhart factors, we obtain the following expression of the irradiance:

$$H(r) = H_0(r) + \int_{A'} E_b(r')dB_{dA-dA'} \tag{A.4}$$

If we decompose the model into N isothermal patches of uniform emissivity, we obtain:

$$q_i(r_i) \quad = \quad \epsilon_i E_{b,i} - \epsilon_i H_{0,i}(r_i) - \epsilon_i \sum_{j=1}^{N} E_{b,j} B_{dA_i - A_j} \tag{A.5}$$

The radiative heat flux $q_i(r_i)$ is not constant on A_i because the irradiance can vary on A_i, as well as the point wise Gebhart factor $B_{dA_i-A_j}$. We have to average the previous equation on A_i.

$$q_i = \epsilon_i E_{b,i} - \epsilon_i H_{0,i} - \epsilon_i \sum_{j=1}^{N} E_{b,j} B_{i-j} \tag{A.6}$$

which is identical to the initial equation developed by Gebhart, with an additional term related to the external irradiation.

If we refer to a previous remark, the heat flux is balanced by two different contributions: the external irradiation and the black body radiation emitted by the surfaces.

$$q_i^E \quad = \quad \epsilon_i E_{b,i} - \epsilon_i \sum_{j=1}^{N} E_{b,j} B_{i-j} \tag{A.7}$$

$$q_i^I \quad = \quad -\epsilon_i H_{0,i} \tag{A.8}$$

The equivalence of the Gebhart equations (A.7) and (A.8) with the diffuse equations (2.40) and (2.41) is proved in appendix B.1.

A.2 Semi gray approximation

In heat transfer for space applications, the radiations cover a large spectral band, from thermal infrared to ultraviolet. The thermo optical properties can widely vary on this

spectral domain. A common approximation consists in separating the spectral domain into two separated bands: the visible one which corresponds to the irradiation from the Sun and the infrared one which is linked to the emission of the surfaces around $-200°C$ and $200°C$. The visible domain is characterized by wavelengths inferior to $3\mu m$ while the infrared domain corresponds to wavelengths larger than $3\mu m$. This approximation is justified by the fact that the spectral range of these two radiations do not overlap.

Due to the separation of the spectral domain, we have to develop two sets of equations. If we suppose that the surfaces of the model do not emit visible light, we can decouple the two sets of equations.

A first analysis of the semi gray approximation has been performed by Bobco and coworkers [BAO67] for diffuse reflection only. In this section, we use an approach similar to [Mod03] in order to include specular reflection.

For the sake of readability, we will note the quantities related to the visible band with an asterisk *. The infrared quantities are noted as in the previous sections.

A.2.1 Visible component

If we assume that the self-emission of the model's surfaces is equal to zero in the visible range, the visible radiosity is only due to diffuse reflection:

$$J^*(r) = \rho^{d*}(r) H^*(r) \tag{A.9}$$

The visible component of the heat flux can be obtained from the visible irradiance as follows:

$$\begin{aligned} q^*(r) &= J^*(r) + \rho^{s*}(r) H^*(r) - H^*(r) && \text{(A.10)} \\ &= -\alpha(r) H^*(r) && \text{(A.11)} \end{aligned}$$

For each surface, the visible heat flux reduces to the absorption of the irradiance. This irradiance is the sum of the external irradiance $H_0^{s*}(r)$, received either directly or after any number of specular reflections, and the radiosity received from the geometrical model:

$$H^*(r) = H_0^{s*}(r) + \int_{A'} J^*(r') dF^s_{dA-dA'} \tag{A.12}$$

The radiosity is linked to the radiative heat flux by the simple relation:

$$J^*(r) = -\frac{\rho^{d*}(r)}{\alpha(r)} q^*(r) \tag{A.13}$$

The radiative heat flux can then be rewritten as:

$$\frac{q^*(r)}{\alpha(r)} = -H_0^{s*}(r) + \int_{A'} \frac{\rho^{d*}(r')}{\alpha(r')} q^*(r') dF^s_{dA-dA'} \tag{A.14}$$

We decompose the model into isothermal patches characterized with constant thermo-optical properties. We average each equation i on the surface area A_i.

$$\frac{q_i^*}{\alpha_i} = -H_{0,i}^{s*} + \sum_{j=1}^{N} \frac{\rho_j^{d*}}{\alpha_j} q_j^* F_{i-j}^s \tag{A.15}$$

Remark: the hypothesis of isothermal patches is not used for the visible set of equations. However, it is necessary for the infrared set of equations.

Remark: the thermal radiosity J_i^* corresponds to the radiosity B_i computed in rendering [GTGB84]. Indeed, we can consider the visible sources (the Sun and the planet reflecting the albedo flux) as "lamps", light sources. The power of these sources is a data, given by the position and orientation of the satellite on its orbit.

A.2.2 Infrared component

Once the distribution of absorbed visible heat fluxes q_i^* has been computed, we must introduce it in the thermal equations depending on the local boundary condition.

When the temperature T_i of a surface i is known, the total heat flux is the sum of the infrared and visible components:

$$q_i|_{\text{total}} = q_i^* + q_i \tag{A.16}$$

If the heat flux is fixed at a given value $q_i|_{\text{total}}$, the infrared component is then fixed at:

$$q_i = q_i|_{\text{total}} - q_i^* \tag{A.17}$$

The infrared equations are identical to the radiosity equations (2.62).

A.3 Multiple spectral bands

For some applications such as cryogenics equipments, the previous semi-grey approximation is not sufficient and more spectral bands must be defined. In this case, we use the *band approximation* [Mod03, Pan05]. We decompose the spectral domain into M bands. We assume that the thermo-optical properties are constant on each spectral band. A development similar to the one used previously will yield the following set of equations:

$$E_{b,i}^{(m)} - \sum_{j=1}^{N} \left(1 - \rho_j^{s(m)}\right) F_{i-j}^{s(m)} E_{b,j}^{(m)} = \frac{q_i^m}{\epsilon_j^m} - \sum_{j=1}^{N} \frac{\rho_j^{d(m)}}{\epsilon_j^{(m)}} F_{i-j}^{(m)} q_j^{(m)} + H_{0,i}^{s(m)} \tag{A.18}$$

The total power emitted by a surface i is obtained by summing the components of the different spectral bands:

$$E_{b,i} = \sum_{m=1}^{M} E_{b,i}^{(m)} \tag{A.19}$$

In the same way, the total radiative heat flux is the sum of the different spectral bands:

$$q_i = \sum_{m=1}^{M} q_i^{(m)} \tag{A.20}$$

The total irradiance corresponds to the sum of the irradiances used in the M spectral bands:

$$H_{0,i} = \sum_{m=1}^{M} H_{0,i}^{s(m)} \tag{A.21}$$

The width of the spectral bands must be chosen in such a way that the thermo-optical properties can be assumed constant, on each band. On the other hand, if we consider specular surfaces, we need to store a complete set of view factors for each spectral band.

Appendix B

Radiosity equation, Gebhart's formulation and importance

B.1 Equivalence between the radiosity equations and Gebhart's formulation

In this appendix, we prove the equivalence between the radiosity equations (2.40) and (2.41) and the Gebhart's equations (A.7) and (A.8). The first demonstration is related to the self-emitted power of the surfaces and the heat flux q_i^E. The second demonstration concerns the external irradiation and the component q_i^I of the radiative heat flux.

B.1.1 Self-emitted power

The demonstration is based on a three-step development:

1. we consider the equation (2.40), we isolate the radiative heat flux and compute the total net power;
2. we use the Gebhart's equation (A.7) and use the iterative formula of the Gebhart factors based on the view factors;
3. we compare the two resulting equations and find again the equation (A.7).

The first step yields the following development:

$$\begin{aligned} \sum_{j=1}^{N} \left[\frac{\delta_{i,j}}{\epsilon_j} - \left(\frac{1}{\epsilon_j} - 1 \right) F_{i-j} \right] q_j^E &= \sum_{j=1}^{N} \left(\delta_{i,j} - F_{i-j} \right) E_{b,j} \\ \Leftrightarrow Q_i = A_i q_i^E &= A_i \epsilon_i E_{b,i} - \sum_{j=1}^{N} A_i \epsilon_i F_{i-j} E_{b,j} + \sum_{j=1}^{N} A_i \epsilon_i \rho_j F_{i-j} \frac{q_j^E}{\epsilon_j} \end{aligned} \quad \text{(B.1)}$$

If we consider the second step, we introduce the Gebhart's formula for the radiative exchange factors into the equation (A.7):

$$\begin{aligned} Q_i = A_i q_i^E &= A_i \epsilon_i E_{b,i} - \sum_{j=1}^{N} A_j \epsilon_j E_{b,j} B_{j-i} \\ &= A_i \epsilon_i E_{b,i} - \sum_{j=1}^{N} A_i \epsilon_i F_{i-j} \epsilon_j E_{b,j} - \sum_{j=1}^{N} A_i \epsilon_i \sum_{k=1}^{N} F_{i-k} \rho_k B_{k-j} E_{b,j} \end{aligned} \quad \text{(B.2)}$$

Finally, we compare the equations (B.1) and (B.2) and obtain the following equation:

$$A_i \epsilon_i \left[\sum_{j=1}^{N} F_{i-j}(1-\epsilon_j) E_{b,j} - \sum_{j=1}^{N} \sum_{k=1}^{N} F_{i-k} \rho_k B_{k-j} E_{b,j} \right] = A_i \epsilon_i \sum_{j=1}^{N} \rho_j F_{i-j} \frac{q_j^E}{\epsilon_j} \quad \text{(B.3)}$$

We permute the indices j and k of the second term of the first member and we simplify the factor $A_i \epsilon_i$:

$$\sum_{j=1}^{N} F_{i-j}(1-\epsilon_j) E_{b,j} - \sum_{k=1}^{N} \sum_{j=1}^{N} F_{i-j} \rho_j B_{j-k} E_{b,k} = \sum_{j=1}^{N} \rho_j F_{i-j} \frac{q_j^E}{\epsilon_j} \quad \text{(B.4)}$$

$$\Leftrightarrow \sum_{j=1}^{N} \rho_j F_{i-j} \left(E_{b,j} - \sum_{k=1}^{N} B_{j-k} E_{b,k} \right) = \sum_{j=1}^{N} \rho_j F_{i-j} \frac{q_j^E}{\epsilon_j} \quad \text{(B.5)}$$

$$\Rightarrow \frac{q_j^E}{\epsilon_j} = \left(E_{b,j} - \sum_{k=1}^{N} B_{j-k} E_{b,k} \right) \quad \text{(B.6)}$$

$$\Leftrightarrow Q_j = A_j q_j^E = A_j \epsilon_j E_{b,j} - \sum_{k=1}^{N} A_j \epsilon_j B_{j-k} E_{b,k} \quad \text{(B.7)}$$

$$= A_j \epsilon_j E_{b,j} - \sum_{k=1}^{N} A_k \epsilon_k B_{k-j} E_{b,k} \quad \text{(B.8)}$$

We finally obtain the initial equation of Gebhart. The two formulations are equivalent if this initial equation is correct. This is one of the hypothesis of this demonstration. The two formulas (B.1) and (B.2) are equivalent, so are equations (2.40) and (A.7).

B.1.2 External irradiation

The component q_i^I of the radiative heat fluxes can be computed with the equation (2.41) from the radiosity formulation. Here we will demonstrate that this formulation is equivalent to equation (A.8).

First, we obtain the radiative heat flux on i in function of the irradiance on i and the heat fluxes of the other surfaces j.

$$\begin{aligned} \sum_{j=1}^{N} \left[\frac{\delta_{i-j}}{\epsilon_j} - \left(\frac{1}{\epsilon_j} - 1 \right) F_{i-j} \right] q_j^I &= -H_{0,i} \\ \Leftrightarrow \frac{q_i^I}{\epsilon_i} &= -H_{0,i} + \sum_{j=1}^{N} \rho_j F_{i-j} \frac{q_j^I}{\epsilon_j} \end{aligned} \tag{B.9}$$

Following the Gebhart formulation, the net power on i is given by the following relation:

$$Q_i = A_i \epsilon_i H_{o,i} + \sum_{j=1}^{N} B_{j-i} A_j \rho_j H_{0,j} \tag{B.10}$$

This relation implies that the power absorbed by a surface i due to the external irradiation is the sum of the incoming irradiance and the reflected irradiance from all the surfaces of the model. We start from the equation (B.10) and use the reciprocity relation of the Gebhart factors.

$$Q_i = A_i \epsilon_i H_{0,i} + \sum_{j=1}^{N} B_{i-j} A_i \epsilon_i \frac{\rho_j}{\epsilon_j} H_{0,j} \tag{B.11}$$

We introduce the Gebhart equation for the radiative exchange factors in function of the view factors.

$$Q_i = A_i \epsilon_i H_{0,i} + \sum_{j=1}^{N} F_{i-j} A_i \epsilon_i \rho_j H_{0,j} + \sum_{j=1}^{N} \sum_{k=1}^{N} F_{i-k} \rho_k B_{k-j} A_i \epsilon_i \frac{\rho_j}{\epsilon_j} H_{0,j} \tag{B.12}$$

We divide the previous expression by $A_i \epsilon_i$ and permute the indices j and k appearing the the double sum term:

$$\frac{q_i}{\epsilon_i} = -H_{0,i} - \sum_{j=1}^{N} F_{i-j} \rho_j H_{0,j} + \sum_{k=1}^{N} \sum_{j=1}^{N} F_{i-j} \rho_j B_{j-k} \frac{\rho_k}{\epsilon_k} H_{0,k} \tag{B.13}$$

Finally, we compare this last equation with the relation (B.9):

$$\frac{q_j}{\epsilon_j} = -H_{0,j} - \sum_{k=1}^{N} B_{j-k} H_{0,k} \frac{\rho_k}{\epsilon_k} \tag{B.14}$$

$$Q_j = A_j \epsilon_j H_{0,j} + \sum_{k=1}^{N} A_j \epsilon_j B_{j-k} H_{0,k} \frac{\rho_k}{\epsilon_k} \tag{B.15}$$

$$= A_j \epsilon_j H_{0,j} + \sum_{k=1}^{N} A_k B_{k-j} \rho_k H_{0,k} \tag{B.16}$$

We find the initial equation of Gebhart applied to the external irradiance. This concludes the demonstration.

This demonstration has been performed for only one spectral band and for diffuse reflectors but it can easily be extended to the case of multiple spectral bands and specular reflectors.

B.2 Importance equations

In this Appendix, we will demonstrate the formulation of the importance (6.27) equation.

B.2.1 Importance equation

Let us consider an environment without any light source, *i.e.* without any thermal radiation [PM95]. If a source is introduced, each surface in the model can receive a flux, either directly or after any number of reflections. The received flux is function of the proximity to the source. This can be described by the notion of importance. The closer to the source, the higher flux, the higher importance.

Let us introduce a detector, noted k, in the model. This detector is assumed to be ideal *i.e.* it does not affect the distribution of radiation. The detector receives a fraction of the radiation emitted by the source, numbered 1. Let us note $W_k(x_1)$ the power received by k. W_k is called the importance.

In order to establish the equation of importance, we must first consider the direct component of importance. We define a function g_k which is equal to 1 in the region of the detector k and 0 elsewhere. We note χ_k the area of the detector k. g_k represents the initial importance, *i.e.* the direct contribution.

The contribution due to multi-reflection corresponds to the flux received after any number of reflections. Let us note y the nearest point from x in the direction Θ_x (see figure B.1).

We have to consider all incident radiations on y which are reflected in the direction of x. An elementary contribution on y is given by $\rho_y \cos(\theta_y) d\omega_y W_k(y)$. We finally obtain the importance equation:

$$W_k(x) = g_k(x) + \int_{\Omega_y} \rho_y \cos(\theta_y) W_k(y) d\omega_y \tag{B.17}$$

We decompose the model into N patches assumed to be isothermal and associated with uniform thermo-optical properties. We integrate the previous equation on the surface A_i

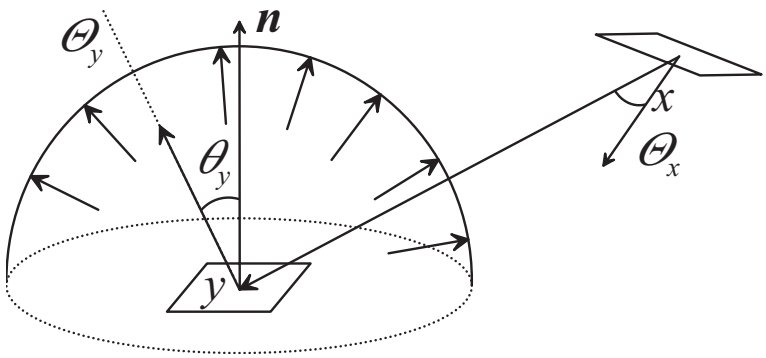


Figure B.1: Direction Θ_y on the hemisphere Ω_y

of patch P_i. We obtain:

$$I_i = R_i + \sum_{j=1}^{N} \rho_j F_{j-i} I_j \tag{B.18}$$

where

$$\begin{array}{rcccl} I_i & = & A_i W_i & = & \int_{A_i} W(x) dA_i \\ R_i & = & \int_{A_i} g(x) dA_i & = & \left\{ \begin{array}{cc} A_i & \text{if} \quad A_i \in \chi \\ 0 & \text{if} \quad A_i \notin \chi \end{array} \right. \end{array} \tag{B.19}$$

χ is the set of the considered surfaces. In heat transfer, all surfaces must be considered. We cannot remove some faces for visibility reasons, as it is done in rendering.

Appendix C

Estimation of the error term

We used a Matlab routine in order to compute the projection of a target surface A_j onto the hemisphere. The function INPOLYGON has been used to evaluate the shadow ratio of the cells of the stratified hemisphere. Given a contour and a set of points, the function INPOLYGON determines if the points are inside or outside the contour. For each cell, the shadow ratio is approximated by the number of corners situated in the contour. So a discrete approximation of the shadow ratio is obtained; the approximation increases by step of 25%.

Here the results obtained with the Matlab routine are presented in the case of two squares of unit length, aligned in parallel planes, separated by a unit distance (see Figure C.1). In this case, the distance between the patches P_i and P_j is equal to the length of an edge. The results obtained with this routine are the contour of projection, the shadow ratios, the evolution of the error in function of the resolution M of the hemisphere, the evolution of the number of internal and external cells in function of M... This will help us to establish laws of evolution in function of M and to estimate the error term in the equation (3.16).

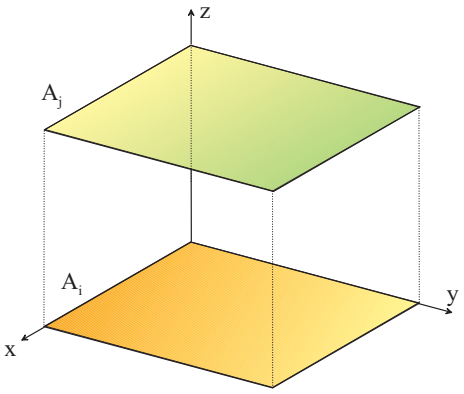


Figure C.1: Parallel squares - geometrical configuration.

C.1 Projection of the target area

Figure C.2–a represents the result of the projection of the target patch onto the hemisphere. The resolution has been limited for sake of visibility. Figure C.2–b represents the shadow ratio of each cell, varying from 0 to 1.

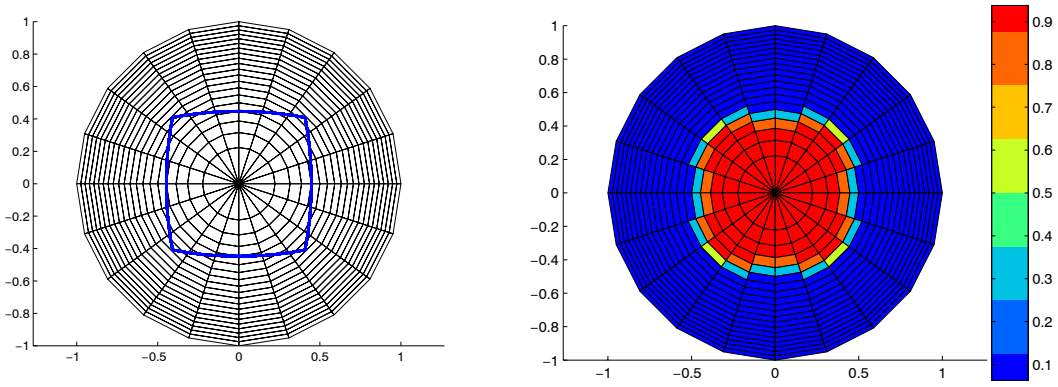


Figure C.2: Projection of the target area – a) projection of the contour onto the unit disc – b) shadow ratios across the unit disc.

C.2 In terms of erroneous cells

Figure C.3–a represents the number of rays impacting the interior of the contour of the target patch. These rays are not subject to error (the corresponding shadow ratio is equal to 1). The evolution of this number in function of the hemisphere's resolution is quadratic. This confirms equation $F_{di-j} = \frac{n^2}{M^2}$. Figure C.3–b corresponds to the number of rays impacting the contour of the target patch. These rays can be subject to error since their shadow ratios vary between 0 and 1. We can observe that the evolution of this number of rays is effectively linear, as we supposed previously. We can also recognize periodical perturbations linked to the shape of the target contour. These perturbations are due to aliasing. This aliasing is linked to the projection of $\mathcal{C}_j$ on the hemisphere mesh in function of M. The number of cells partially covered by the projection is driven by a deterministic process.

Figure C.3–b justifies the definition of the geometrical parameter c that we introduced in equation (3.18) and which allows us to approximate the number of cells subject to error by the following expression. Let us point out that the value of c is a function of the geometrical configuration and must be evaluated for each model.

$$m \approx cM$$

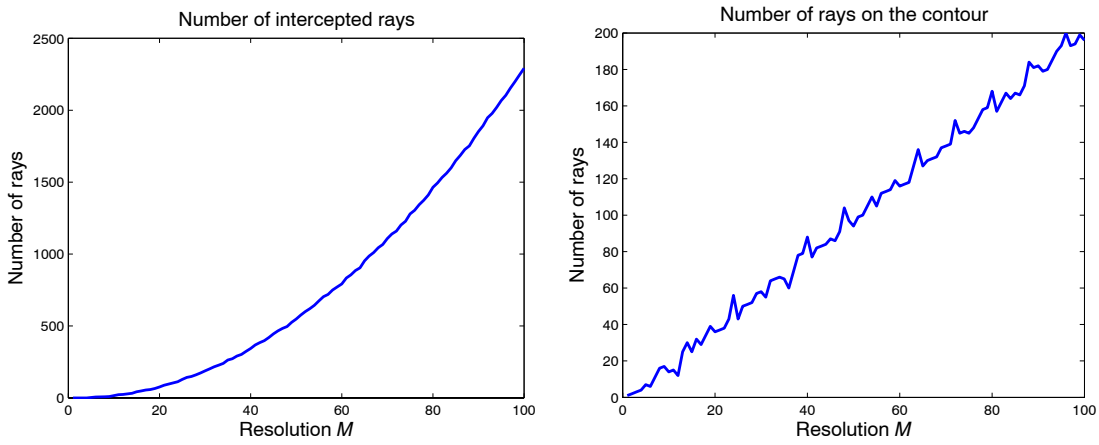


Figure C.3: Impact numbers – a) interior of the target – b) contour of the target.

C.3 In terms of shadow ratios

We will now consider the evolution of the shadow ratios in function of the hemisphere resolution. Figure C.4 represents the evolution of the cumulated shadow ratio in function of the resolution. We can observe that it is affected by the same perturbations as in Figure C.3–b. This is due to the deterministic projection of the contour onto the hemisphere mesh. We can consider that the evolution of the cumulated shadow ratio is roughly linear. We can define a constant s and approximate the sum by a linear law:

$$\sum_{k=1}^{m} p_k \approx sM \tag{C.1}$$

where the sum is performed on the m cells of the contour that is, the cells which are characterized by a partial shadow ratio. The totally covered cells are not counted here.

C.4 In terms of the cumulated error

We will now evaluate the sum which appears in the equation (3.16). In Figure C.5–a, we trace the evolution of the cumulated error. We can observe that this error follows a linear law in function of the hemisphere resolution. In Figure C.5–b, we divided the cumulated error by the current resolution, for each resolution. For small resolutions, we can observe a chaotic behaviour. Let us note that the mean error does not exceed 0.25, which is in agreement with the previous theory (maximum error for a shadow ratio of 50% and given by$p_k(1 - p_k)$. When the resolution increases, the mean error converges to a value which only depends on the geometrical configuration. Thanks to the linear behaviour of the cumulated error, we can introduce a new constant d which allows us to approximate the

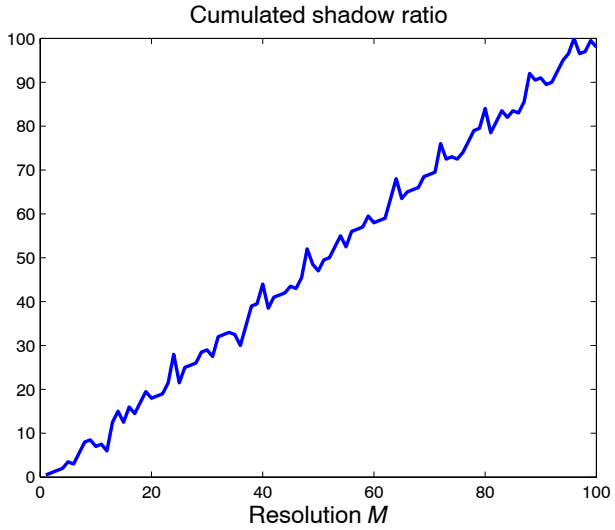


Figure C.4: Cumulated shadow ratio.

sum term by the following linear law (see equation (3.27)):

$$\sum_{k=1}^{m} p_k(1-p_k) \quad \approx \quad dM$$

Let us note here that the sum can be performed on all the cells of the hemisphere, only the partially covered ones will yield non-zero terms.

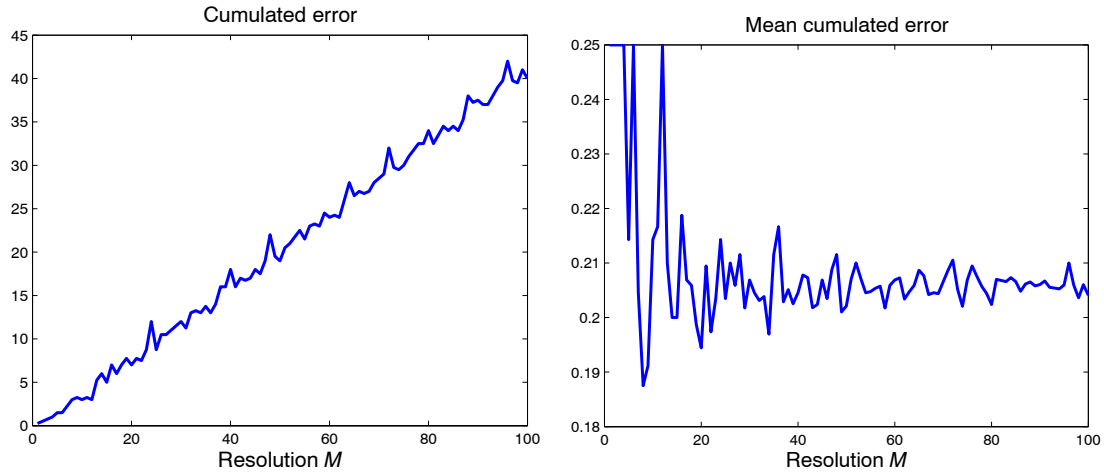


Figure C.5: Cumulated error – a) cumulated error – b) mean cumulated error.

Many configurations have been studied, the same linear evolutions have been observed for the previous studied quantities.

Appendix D

Normal behaviour of the stratified hemisphere method and exchange factor to space

D.1 Illustration of the normal behaviour of the view factors computed with the hemisphere method

In this Appendix, we will illustrate the normal behaviour of the distribution of view factors computed with the stratified hemisphere method and Gauss quadrature.

We consider a particular view factor from a simple test case, composed of two perpendicular rectangles sharing a common edge. The values obtained for 1000 ray-tracing processes are stored. They are then sorted by increasing values; the result is plotted in Figure D.1. This curve corresponds to the cumulative distribution function of the considered view factor. This function is discontinuous. The function is fitted with a Fourier function (order 6). The fitting curve is plotted in red in Figure D.1.

The next step consists in deriving the fitting function with respect to the view factor in order to obtain the probability density function. The result is shown in Figure D.2. This function is fitted with a gaussian curve. The two curves are in good agreement, in spite of the successive operations performed on the initial data, limited to 1000 samples.

We can conclude that the view factors obtained with the stratified hemisphere method follow a normal law. This property can be demonstrate by using the Central-Limit theorem.

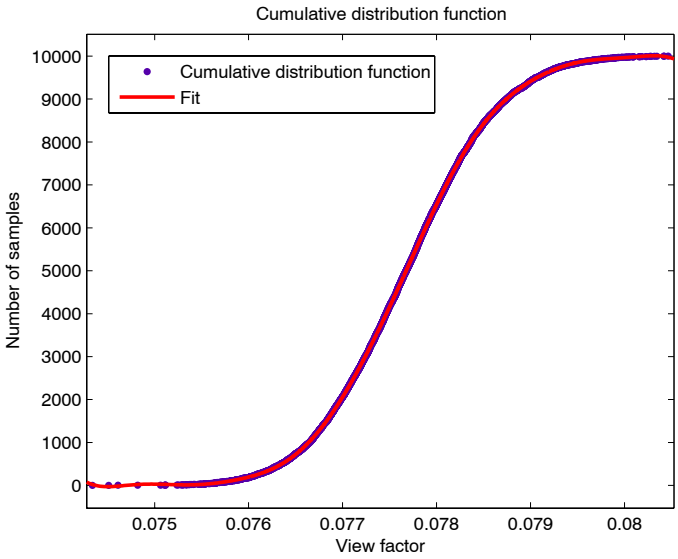


Figure D.1: Cumulative distribution function.

D.2 Exact exchange factors to the deep space - demonstration

In this Appendix, we present the mathematical developments related to the first method designed to enforce closure in case of an open model. The fundamental hypothesis is that the "model-to-deep-space" exchange factors, noted $\eta_{k,N+1}$, are exact and do not need any correction. In this case, the line-constraints must be completed as follows:

$$g_k = \Omega_k - \sum_{l=1}^{N+1} \ddot{\eta}_{k-l} \tag{D.1}$$

$$= \Omega_k - \eta_{k,N+1} - \sum_{l=1}^{N} \ddot{\eta}_{k-l} \tag{D.2}$$

In the same way, the column-constraints must be rewritten. The following expressions are obtained:

$$g_k^* = \Omega_k - \sum_{l=1}^{N+1} \ddot{\eta}_{l-k} \tag{D.3}$$

$$= \Omega_k - \sum_{l=1}^{N} \ddot{\eta}_{l-k} - \eta_{N+1,k} \tag{D.4}$$

$$= \Omega_k - \sum_{l=1}^{N} \ddot{\eta}_{l-k} - \eta_{k,N+1} \tag{D.5}$$

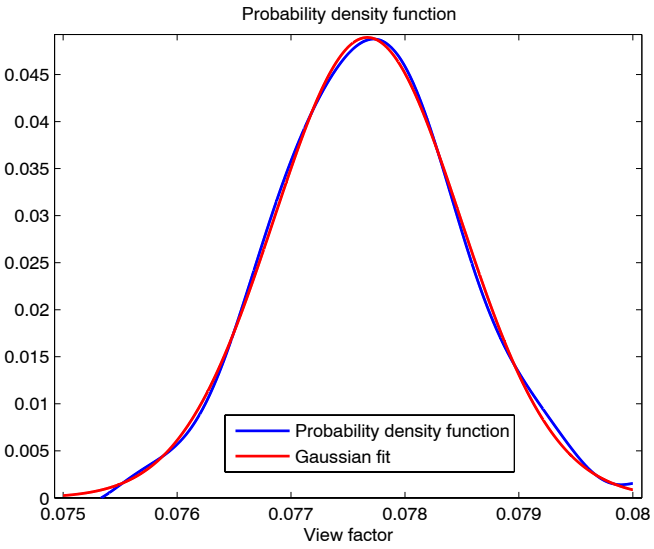


Figure D.2: Probability density function.

The last equation has been obtained by using the exactness of the exchange factors to the deep space. This implies that the reciprocity relation is also exact. The Lagrangien consists also of the objective function $\mathcal{H}$. As the exchange factors to the deep space are supposed to be exact, it is not necessary to modify the initial formulation of the least-squares smoothing. The first derivative of the Lagrangien is computed. We find again the relation (3.93). The introduction of the closure relation yields the following set of equations:

$$\Omega_k = \sum_{l=1}^{N} \dot{\eta}_{k-l} + \lambda_k \sum_{l=1}^{N} w_{k-l} + \sum_{l=1}^{N} \lambda_l w_{k-l} + \eta_{k,N+1} \tag{D.6}$$

The matrix stays unchanged. But the vector b must also be completed. The following relation is obtained:

$$b_j = \Omega_j - \sum_{i=1}^{N} \dot{\eta}_{j-i} - \eta_{j,N+1} \tag{D.7}$$

D.3 Correction of the exchange factors to the deep space - demonstration

Here the least-square smoothing is extended to open models by removing the column-constraints g_i^*. This suppression is necessary because we have no information concerning the line of the exchange factor matrix relative to the deep space.

The main difference with the first method is that the exchange factors to the deep space are no longer supposed to be exact. They are characterized by an error and must be

corrected. Without any information concerning the reciprocal exchange factor, the "model-to-deep-space" exchange factors are not concerned by the reciprocity enforcement. They can be modified by the closure enforcement and must appear in the objective function $\mathcal{H}$. The following expression of the Lagrangien is obtained:

$$\mathcal{L} = \sum_{i=1}^{N}\sum_{j=1}^{N+1}\left(\frac{\dot{\eta}_{i-j}-\ddot{\eta}_{i-j}}{2\omega_{i-j}}\right)^2+\sum_{i=1}^{N}\lambda_i g_i \tag{D.8}$$

$$\mathcal{L} = \sum_{i=1}^{N}\sum_{j=1}^{N+1}\left(\frac{\dot{\eta}_{i-j}-\ddot{\eta}_{i-j}}{2\omega_{i-j}}\right)^2+\sum_{i=1}^{N}\lambda_i\left(\Omega_i-\sum_{j=1}^{N+1}\ddot{\eta}_{i-j}\right) \tag{D.9}$$

The first derivative of the Lagrangien with respect to $\ddot{\eta}_{i-j}$ yields the desired value of the estimator. It yields the following results:

$$\frac{\partial\mathcal{L}}{\partial\ddot{\eta}_{i-j}} = -\frac{\dot{\eta}_{i-j}-\ddot{\eta}_{i-j}}{\omega_{i-j}}-\lambda_i=0 \tag{D.10}$$

$$\Leftrightarrow \ddot{\eta}_{i-j} = \dot{\eta}_{i-j}+\omega_{i-j}\lambda_i \tag{D.11}$$

The introduction of this last expression in the closure law yields the following expression:

$$\sum_{j=1}^{N+1}\omega_{i-j}\lambda_i = \Omega_i-\sum_{j=1}^{N+1}\dot{\eta}_{i-j} \tag{D.12}$$

$$\lambda_i = \frac{\Omega_i-\sum_{j=1}^{N+1}\dot{\eta}_{i-j}}{\sum_{j=1}^{N+1}\omega_{i-j}} \tag{D.13}$$

Appendix E

Extension of Gebhart's method for specular view factors and non-isothermal surfaces

In this Appendix, two extensions of Gebhart's method are analytically proved. The first one corresponds to the introduction of non-diffuse reflectors. The second formula has been established in the framework of the finite element method and illustrates the extension to non-isothermal surfaces.

E.1 Specular reflectors

In this Section, the case of combined diffuse and specular reflections is considered. The emission process remains purely diffuse. Gebhart's method is developed in function of the extended, specular view factors. The patches are assumed to be isothermal.

Let us suppose that the specular view factors have been previously computed. The power diffusely emitted by the patch P_i is equal to:

$$Q_i = A_i \epsilon_i E_{b,i} \tag{E.1}$$

The power emitted by P_i which reaches P_j (directly or after any number of specular reflections) and which is finally absorbed by P_j, is given by the following relation:

$$Q_{i-j}|_{\text{direct-specular}} = A_i \epsilon_i E_{b,i} F^s_{i-j} \epsilon_j \tag{E.2}$$

The power emitted by P_i, which reaches a patch P_k (directly or after any number of specular reflections) and which is diffusely reflected by P_k, is equal to:

$$Q_{i-j}|_{\text{diffuse}} = A_i \epsilon_i E_{b,i} F^s_{i-k} \rho^d_k \tag{E.3}$$

The fraction of this power which is then absorbed by P_j is equal to B_{k-j}. Finally, the power emitted by P_i which is absorbed by P_j is given by:

$$Q_{i-j}|_{\text{total}} = A_i \epsilon_i F^s_{i-j} \epsilon_j + \sum_{k=1}^{N} A_i \epsilon_i E_{b,i} F^s_{i-k} \rho^d_k B_{k-j} \tag{E.4}$$

The corresponding Gebhart's factor is obtained by dividing the previous result by the total power emitted by P_i:

$$B_{i-j} = \epsilon_j F^s_{i-j} + \sum_{k=1}^{N} F^s_{i-k} \rho^d_k B_{k-j} \tag{E.5}$$

This relation has been compared to [DR97]. The two expressions are equivalent.

E.2 Finite element formulation

In the case of finite elements, non-isothermal patches are obtained. In this Section, we present the corresponding formulation of Gebhart's method.

We suppose that the finite element view factors have been previously computed. The power emitted by node j on patch i is given by the following relation:

$$\begin{aligned} Q^i_j &= \int_{A_i} N^i_j(r) \epsilon(r) E_b(r) dA_i & \text{(E.6)} \\ &= \epsilon_j E_{b,j} \det\left(J_i\right)|_j W_j & \text{(E.7)} \end{aligned}$$

where $N^i_j(r)$ is the shape function of node j, evaluated on point r on patch P_i, $\det\left(J_i\right)|_j$ is the determinant of the Jacobian evaluated on node j and W_j is the weight of node j.

The power which leaves node j through patch P_j, which directly reaches node l through patch P_k and is finally absorbed by it is equal to:

$$Q^{i-k}_{j-l} = \epsilon_j E_{b,j} \det\left(J_i\right)|_j W_j F^{i-k}_{j-l} \epsilon_l \tag{E.8}$$

where F^{i-k}_{j-l} is the view factor from node j on patch i to node l, through its influence area on patch P_k.

The power which leaves node j through patch P_j, which directly reaches node n through P_m and is diffusely reflected, is given by the following relation:

$$Q^{i-m}_{j-n} = \epsilon_j E_{b,j} \det\left(J_i\right)|_j W_j F^{i-m}_{j-n} \rho_n \tag{E.9}$$

The fraction of this power which is finally absorbed by node l is equal to B_{n-l}^{m-k}. The power emitted by j through P_i which is absorbed by node l on P_k is equal to:

$$Q_{j-l}^{i-k}\big|_{\text{total}} = \epsilon_j E_{b,j} \det\left(J_i\right)\big|_j W_j F_{j-l}^{i-k} \epsilon_l + \epsilon_j E_{b,j} \det\left(J_i\right)\big|_j W_j \sum_{m,n} F_{j-n}^{i-m} \rho_n B_{n-l}^{m-k} \tag{E.10}$$

where the sum is performed on all the intermediary nodes, through their influence area.

The corresponding Gebhart's factor is the fraction of the emitted power which is finally absorbed by node k:

$$B_{j-l}^{i-k} = \epsilon_l F_{j-l}^{i-k} + \sum_{m,n} F_{j-n}^{i-m} \rho_n B_{n-l}^{m-k} \tag{E.11}$$

Appendix F

Solar visibility

In this Appendix, we demonstrate the interest of stratified sampling applied to the computation of solar visibility, with a reasoning similar to the one developed for the hemisphere method. The extension from visibility to visible cross section, which is the really useful quantity for heat flux computation, is not straightforward.

Then, the basis of a statistical accuracy control are given, in a way similar to the hemisphere method.

F.1 Stratified sampling

If we decompose a surface A into N sub-surfaces A_k, we define the *visibility ratio* v_k as the fraction of A_k which is visible from the Sun.

In sub-surface k, we generate a random point, uniformly distributed on A_k. Each sub-surface is governed by a binomial distribution. We define a binomial variable T_k which can only take two values: 1 if the shadow ray reaches the Sun and 0 otherwise. The probability of this binomial distribution is equal to the visibility ratio v_k. The expected value and variance of this distribution are:

$$\left\{ \begin{array}{rcl} E(T_k) & = & v_k \\ V(T_k) & = & v_k(1-v_k) \end{array} \right. \tag{F.1}$$

The visibility of the whole surface A is defined as $v^* = \frac{1}{N}\sum_{k=1}^{N} t_k$, where t_k is the score of the binomial variable T_k. The expected value and variance of this new distribution are:

$$\left\{ \begin{array}{rcl} E(V^*) & = & v \\ V(V^*) & = & \frac{1}{N^2}\sum_{k=1}^{N} v_k(1-v_k) \end{array} \right. \tag{F.2}$$

We can verify that the distribution V^* satisfies the hypothesis of the Central limit theorem. The distribution V^* is normal. We define the relative error $e = \frac{v^*-v}{v}$, whose

expected value and variance are:

$$\left\{ \begin{array}{rcl} E(E) & = & 0 \\ V(E) & = & \frac{1}{N^2 v^2} \sum_{k=1}^{N} v_k(1-v_k) \end{array} \right. \tag{F.3}$$

Since V^* is normally distributed, so will be E. The probability α of having a relative error smaller than ϵ is given by:

$$\alpha = \mathrm{erf}\left(\frac{\epsilon}{\sqrt{2V(E)}} \right) \tag{F.4}$$

We introduce this relation in the expression (F.3) of the variance and obtain:

$$\left(\frac{\epsilon}{\mathrm{erf}^{-1}(\alpha)} \right)^2 = \frac{2}{N^2 v^2} \sum_{k=1}^{N} v_k(1-v_k) \tag{F.5}$$

Among the N terms of the sum, only the terms relative to the shadow limits $\mathcal{S}$ contribute to the global error. The other terms are associated with visibility ratios equal to 0 or 1 which cancel the corresponding contributions. These conclusions are similar to the ones of the hemisphere method.

If we want to compute the visible cross section, we must multiply the local visibility v by the cosinus of the angle defined by the local normal and the solar direction. The cross section $a_\perp$ is approximated by the following relation:

$$a_\perp^* = \frac{A}{N} \sum_{k=1}^{N} t_k \cos(\theta_k) \tag{F.6}$$

The presence of $\cos(\theta_k)$ makes the new distribution to be non-binomial. The computation of the variance becomes more complex; it depends on the curvature of the surfaces. If we note $V(\cos(\Theta)T)$ the variance of this new distribution [Pla93], the number of random points is found to be equal to:

$$N^2 = 2 \left(\frac{\mathrm{erf}^{-1}(\alpha)}{\epsilon} \right)^2 \frac{A^2}{a_\perp^2} V(\cos(\Theta)T) \tag{F.7}$$

The difficulty of this expression lies in the establishment of the variance term, which is function of the solar direction and the local curvature of the surface A.

F.2 Statistical accuracy control

With respect to view factor computation and the hemisphere method, a given surface is generally crossed by only one solar shadow. The number of sub-surfaces A_k crossed by the shadow $\mathcal{S}$ is generally limited. This number is evaluated during a first pass. Then the optimal number of rays is given by relation (F.5) and a second ray tracing is performed.

F.2.1 First pass

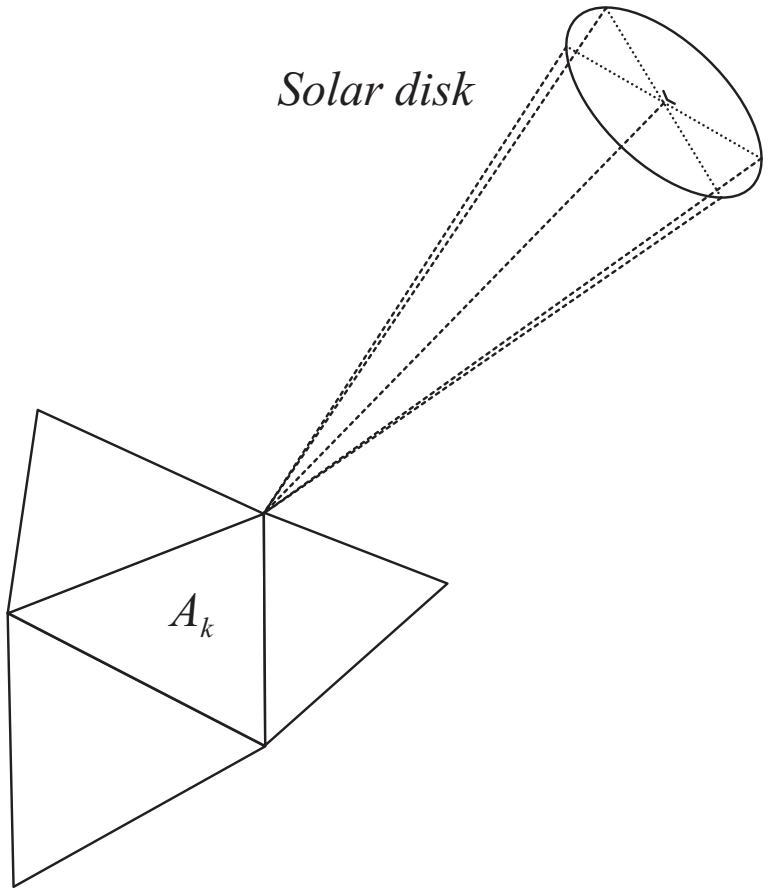


Figure F.1: First pass - solar angular aperture.

The objective of the first pass is to yield a first approximation of the shadow $\mathcal{S}$. This is done by casting a ray from each node of the sub-surfaces A_k, if the Sun is considered as a point wise source, at an infinite distance. If we consider a non-zero solar angular aperture, it means that the shadow is not sharp but is composer of umbra and penumbra (see figure 5.4). It can be estimated by tracing five rays, one to the centre of the solar disc and four on the contour of the disc, on two orthogonal axis (see figure F.1). If the five rays reach the Sun, the node is fully illuminated. If no ray reaches the Sun, the node is in the umbra and its solar visibility is equal to zero. For intermediate cases, the node is in penumbra and the nodal solar visibility varies by steps of 25%.

The visibility ratio v_k of sub-surface A_k is obtained by averaging the solar visibility of its nodes.

At the end of the first pass, we know the number of sub-surfaces A_k which are characterized by a partial visibility ratio v_k. We have also access to an estimation of the v_k's.

F.2.2 Computation of the number of rays

Giving a maximum relative error ϵ and a confidence interval α, the statistical accuracy control computes the number of rays that must be shot to achieve the desired accuracy, by relation (F.5).

F.2.3 Partly refined ray tracing

From the first pass, we obtain the distribution of the visibility ratios across the sub-surfaces A_k. We can decide to only refine the ray tracing on the sub-surfaces which are characterized by a partial visibility ratio, in order to save computational time.

Bibliography

[AD82] Ameri A. and Felske J. D. Radiation configuration factors for obliquely oriented finite length circular cylinders. *International Journal of Heat and Mass Transfer*, 25:728–736, May 1982.

[AK90] J. Arvo and D. Kirk. Particle transport and image synthesis. *Computer Graphics*, 24(4):63–66, 1990.

[AK02] J. Arvo and D. Kirk. A survey of ray tracing acceleration techniques. In A.S. Glassner, editor, *An Introduction to Ray Tracing*, pages 201–262. Morgan Kaufmann Publishers, Inc, 2002.

[Ale02] I. Alet. *Le Contrôle Thermique des Engins Spatiaux*. SupAéro, 2002. Notes de cours.

[Ama84] J. Amanatides. Ray tracing with cones. *Siggraph Computer Graphics*, 18(3):129–135, 1984.

[Arv95] J. Arvo. Stratified sampling of spherical triangles. In *In Computer Graphics (SIGGRAPH '95 Proceedings*, pages 437–438, 1995.

[Arv01] J. Arvo. Stratified sampling of 2-manifolds. In *State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis*, 2001.

[Ath83] P. R. Atherton. A scan-line hidden surface removal procedure for constructive solid geometry. *Computer Graphics*, 17(3):73–82, 1983.

[AW87] J. Amanatides and A. Woo. A fast voxel traversal algorithm for ray tracing. *Proceedings of EuroGraphics'87*, pages 3–9, 1987.

[BAO67] R. P. Bobco, G. E. Allen, and P. W. Othmer. Local radiation equilibrium temperatures in semigray enclosures. *Journal of Spacecraft and Rockets*, 4(8):1076–1082, 1967.

[BDB08] T. Basset, J. P. Dudon, and F. Brunetti. Thermal analysis tools in cannes: Software and policy. In ESA, editor, *22th European Workshop on Thermal and ECLS Software*, Noordwijk, The Netherlands, 2008. ESA.

[Bec03] P. Beckers. *Communication Graphique*. Editions de l'Université de Liège, 2003.

[BK85] W. F. Bronsvoort and F. Klok. Ray tracing generalized cylinders. *ACM Transactions on Graphics*, 4(4):291–303, October 1985.

[BKP91] J. C. Beran-Koehn and M. J. Pavicic. A cubic tetrahedral adaptation of the hemi-cube algorithm. In *Graphics GEMS II - Academic Press Inc*, pages 299–302, 1991.

[BKP92] J. C. Beran-Koehn and M. J. Pavicic. Delta form-factor calculation for the cubic tetrahedral algorithm. In *Graphics GEMS III*, pages 324–328, 1992.

[Bli77] J. F. Blinn. Models of light reflection for computer synthetized pictures. *Computer Graphics (SIGGRAPH '77 Proceedings)*, 11(2):192–198, 1977.

[Bou00] J. F. Bouraima. Calcul du facteur de vue pour les echanges radiatifs tridimensionnels sous bacon. Master's thesis, Université Léonard de Vinci, avril-juillet 2000.

[Cen03] Alstom Power Technology Centre. *Esatan Engineering Manual*. ALSTOM, Cambridge road, Whetstone, em–esatan–056 edition, April 2003.

[CG85] M. F. Cohen and D. P. Greenberg. The hemi cube - a radiosity solution for complex environments. *SIGGRAPH*, 19(3):31–40, July 1985.

[Che06] E. Checa. Xro mirror spacecraft. thermal mathematical model description and analysis. Technical note, ESA/ESTEC, 2006.

[Chr03] P. H. Christensen. Adjoint and importance in rendering: An overview. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):329–340, 2003.

[CHSP95] G. M. Cook, C. Hodgetts, N. J. Stock, and P. Planas. Using finite element discretisation for spacecraft thermal analysis. *SAE Technical Paper Series*, July 1995.

[CJMD95] A. R. Curran, K. R. Johnson, E. A. Marttila, and S. P. Dudley. Automated radiation modeling for vehicle thermal management. *SAE International Congress and Exposition*, 1995.

[CK94] S. Ching-Kuang. *Graphics GEMS IV*, chapter Computing the Intersection of a Line and a Cylinder, pages 353–355. Academic Press Inc, 1994.

[CK95] S. Ching-Kuang. *Graphics GEMS V*, chapter Computing the Intersection of a Line and a Cone, pages 227–231. Academic Press Inc, 1995.

[Cla71] F. H. Clark. Methods and data for reactors shield calculations. *Advances in Nuclear Science and Technology*, 5:95–183, 1971.

[CNE05] Centre National d'Etudes Spatiales CNES. *Techniques et Technologies des Véhicules Spatiaux*. Cépaduès Editions, 2005.

[Coo86] R. L. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, 5(1):51–72, January 1986.

[Coo94] G. M. Cook. Statistical accuracy control – esarad-110. Technical report, Mechanical Engineering Centre of European Gas Turbines Ltd, 1994.

[CPF89] J. H. Chin, T. D. Panczak, and L. Fried. Finite element and raytracing in coupled thermal problems. In Pineridge Press, editor, *Sixth International Conference on Numerical Methods in Thermal Problems*, pages 683–701, 1989.

[CS94] R. Clarksean and C. Solbrig. Minimization of the effect of errors in approximate radiation view factors. *Nuclear engineering and design ISSN 0029-5493 CODEN NEDEAU*, 149(1-3):431–440, 1994.

[CT82] R. L. Cook and K. E. Torrance. A reflection model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24, 1982.

[CT97] C. Cohen-Tanoudji. *Mécanique Quantique*. Hermann, 1997.

[CW93] M. F. Cohen and J. R. Wallace. *Radiosity and Realistic Image Synthesis*. Morgan Kaufmann Publishers, Inc., 1993.

[CW94] J. M. Cychosz and W. N. Jr. Waggenspack. *Graphics GEMS IV*, chapter Intersecting a Ray with a Cylinder, pages 356–365. Academic Press Inc, 1994.

[DBB03] P. Dutré, P. Bekaert, and K. Bala. *Advanced Global Illumination*. A K Peters, 2003.

[DKW85] N. Dadoun, D. G. Kirkpatrick, and J. P. Walsh. The geometry of beam tracing. *SCG '85: Proceedings of the first annual symposium on computational geometry*, pages 55–61, 1985.

[DMH05] K. J. Daun, D. P. Morton, and J. R. Howell. Smoothing monte carlo exchange factors through constrained maximum likelihood estimation. *ASME Journal of Heat Transfer*, 127:1124–1128, 2005.

[Doc03] Astrium Documentation. *Thermica User's Manual*. ASTRIUM, version 3.2 edition, 2003.

[Doc04] Alstom Documentation. Esarad kernel algorithms document. Technical report, ALSTOM Power Technology Center, May 2004.

[Doc06] Esarad Documentation. *Esarad User Manual*. Alstom Power Technology Centre, Cambridge road, Whetstone, um–esarad–024, version 6.8 edition, January 2006.

[Doc07] Samcef Documentation. *Samcef V12-1 - Documentation*. SamTech, 2007.

[DP93] B. Desaunettes and C. Puillet. Monte-carlo radiative software statistical accuracy control. In D. Gibson, editor, *Seventh European Thermal and ECLS Software Workshop*. Analysis and Verification Section of the Thermal Control and Life Support Division of ESA/ESTEC, 1993.

[DR97] J. M. Defise and P. Rochus. Lessons learned from the thermal design of an instrument (eit, the extreme-uv imaging telescope) on board soho. *SAE transactions*, 106(1):1079–1094, 1997.

[Dud06] J. P. Dudon. A new tool for 3d thermal modeling. In ESA, editor, *20th European Workshop on Thermal and ECLS Software*, Noordwijk, The Netherlands, 2006. ESA.

[EPO91] A. B. Ekoule, F. C. Peyrin, and C. L. Odet. A triangulation algorithm from arbitrary shaped multiple planar contours. *ACM Trans. Graph.*, 10(2):182–199, 1991.

[ESAa] ESA. Herschel. http://sci.esa.int/science-e/www/area/index.cfm?fareaid=16.

[ESAb] ESA. Planck science team home. http://www.rssd.esa.int/index.php?project=PLANCK&page=index.

[ESA89] ESA. *Spacecraft Thermal Control Design Data*. European Space Research and Technology Centre, 1989.

[Eti86] J. Etienne. *Géométrie Analytique - Notions de Géométrie Différentielle*. Université de Liège, 1986.

[EWM08] M. Eisemann, C. Woizischke, and M. Magnor. Ray tracing with the single-slab hierarchy. In *Proc. Vision, Modeling and Visualization (VMV'08)*, 2008.

[FS88] D. Fussell and K. R. Subramanian. Fast ray tracing using k-d trees. *Department of Computer Sciences, University of Texas at Austin*, pages 1–22, 1988.

[FTI86] A. Fujimoto, T. Tanaka, and K. Iwata. Arts: Accelerated ray tracing system. *Computer Graphics and Applications, IEEE*, 6(4):16–26, 1986.

[Geb57] B. Gebhart. Unified treatment for thermal radiation transfer. *ASME Paper 57-A-34*, 1957.

[Geb59] B. Gebhart. A new method for calculating radiant exchanges. *ASHRAE Transactions*, 65:321–332, 1959.

[Geb61a] B. Gebhart. *Heat Transfer*. McGraw-Hill Book Company, 1961.

[Geb61b] B. Gebhart. Surface temperature calculations in radiant surroundings of arbitrary complexity - for gray diffuse radiation. *International Journal of Heat and Mass Transfer*, 3:341–346, 1961.

[Gil02] D. G. Gilmore. *Spacecraft Thermal Control Handbook - Volume I : Fundamental Technologies*. The Aerospace Press, 2002.

[Gla84] A. S. Glassner. Space subdivision for fast ray tracing. *Computer Graphics and Applications, IEEE*, 4(10):15–22, 1984.

[Gla02] A. S. Glassner. *An Introduction to Ray Tracing*. Morgan Kaufmann Publishers Inc., 2002.

[Gra78] B. Grais. *Méthodes Statistiques*. Dunod, 1978.

[GTGB84] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. Modeling the interaction of light between diffuse surfaces. *Computer Graphics*, 18(3):213–222, July 1984.

[Hai89] E. Haines. Essential ray tracing algorithms. In A. Glassner, editor, *An Introduction to Ray Tracing*, pages 33–77, San Diego, CA, 1989. Academic Press.

[Han89] P. Hanrahan. A survey of ray-surface intersection algorithms. In A. Glassner, editor, *An Introduction to Ray Tracing*, pages 79–119, San Diego, CA, 1989. Academic Press.

[HG03] M. Hesse and M. Gavrilova. A quantitative analysis of culling techniques for real-time rendering of digital elevation models. In *WSCG short papers proceedings (WSCG'2003)*, 2003.

[HHP+92] X. D. He, P. O. Heynen, R. L. Phillips, K. E. Torrance, D. H. Salesin, and D. P. Greenberg. A fast and accurate light reflection model. *Computer Graphics*, 26(2):253–254, 1992.

[How82] J. R. Howell. *A Catalog of Radiation Heat Transfer Configuration Factors*. McGray Hill, 1982. http://www.me.utexas.edu/~howell/.

[HSA91] P. Hanrahan, D. Salzman, and L. Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics*, 25(4):197–206, August 1991.

[HTSG91] X. D. He, K. E. Torrance, F. X. Sillion, and D. P. Greenberg. A comprehensive physical model for light reflection. *Computer Graphics*, 25(4):175–186, July 1991.

[Jar03] S. Jardon. Première approche sur l'étude des facteurs de vue. Technical report, Université de Liège, 2003.

[JCG92] K. R. Johnson, A. R. Curran, and T. G. Gonda. Present state and future of infrared signature models. *Third Annual Ground Target Modeling and Validation Conference*, 1992.

[JCG93] K. R. Johnson, A. R. Curran, and T. G. Gonda. Development of a signature supercode. *Technical Proceedings 1938 : Advances in Sensors, Radiometric Calibration, and Processing of Remotely Sensed Data*, April 1993.

[Kaj83] J. T. Kajiya. New techniques for ray tracing procedurally defined objects. *ACM Transactions on Graphics*, 2(3):161–181, July 1983.

[Kaj86] J. T. Kajiya. The rendering equation. *SIGGRAPH*, 20(4):143–150, August 1986.

[Ken99] A. E. Kenelly. Equivalence of triangles and stars in conducting networks. *Electrical World and Engineer*, 34:413–414, 1899.

[KK86] T. L. Kay and J. T. Kajiya. Ray tracing complex scenes. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 269–278, New York, NY, USA, 1986. ACM Press.

[Koe88] C. Koeck. Improved ray tracing techniques for radiation heat transfer modeling. In ESA, editor, *3rd European Symposium on Space Thermal Control and Life Support Systems*, pages 255–260, Noordwijk, The Netherlands, 1988. ESA.

[Kow99] F. Kowsary. A computationally efficient method for monte carlo simulation of diffuse radiant emission or reflection. *International Journal of Heat and Mass Transfer*, 42(1):193–195, 1999.

[LCL01] M. S. Liu, C. K. Choi, and C. W. Leung. Startup analysis of oil-fired furnace - the smoothing monte carlo model approach. *Heat and Mass Transfer*, 37:449–457, 2001.

[LCN95] B.-W. Li, H.-G. Chen, and B.-L. Ning. Uniform deterministic discrete method for ratiave viewfactors in complex geometric systems. *Journal of Thermal Sciences*, 4(2):100–103, 1995.

[LDB95] R. I. Loehrke, J. S. Dolaghan, and P. J. Burns. Smoothing monte carlo exchange factors. *ASME Journal of Heat Transfer*, 117:524–526, 1995.

[Lev90] M. Levoy. Efficient ray tracing of volume data. *ACM Trans. Graph.*, 9(3):245–261, 1990.

[LH86] M. E. Larsen and J. R. Howell. Least-squares smoothing of direct-exchange areas in zonal analysis. *ASME Journal of Heat Transfer*, 108:239–242, 1986.

[Lit99] F. X. Litt. *Analyse Numérique*. Université de Liège, 1999.

[LM84] E. E. Lewiss and W. F. Miller. *Computational Methods of Neutron Transport*. J. Wiley, New York, 1984.

[LMTS96] R. W. Lewis, K. Morgan, H. R. Thomas, and K. N. Seetharamu. *The Finite Element Method in Heat Transfer Analysis*. John Wiley & Sons Ltd, October 1996.

[LRR04] J. Lawrence, S. Rusinkiewicz, and R. Ramamoorthi. Efficient brdf importance sampling using a factored representation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 496–505, New York, NY, USA, 2004. ACM.

[Mal88] T. Malley. A shading method for computer generated images. Master's thesis, University of Utah, June 1988.

[Mod03] M. F. Modest. *Radiative Heat Transfer, second edition*. Academic Press, 2003.

[NN85] T. Nishita and E. Nakamae. Continuous tone representation of three-dimensional objects taking account of shadows and interreflection. *Computer Graphics*, 19(3):23–30, July 1985.

[NN86a] T. Nishita and E. Nakamae. Continuous tone representation of three-dimensional objects illuminated by sky light. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 125–132, 1986.

[NN86b] T. Nishita and E. Nakamae. Half-tone representation of 3-d objects illuminated by area sources or polyhedron sources. *IEEE COMPSAC Proceedings*, 12:239–241, 1986.

[NN93] T. Nishita and E. Nakamae. A new radiosity approach using area sampling for parametric patches. *Eurographics*, 12(3):385–398, 1993.

[NN95] L. Neumann and A. Neumann. Radiosity and hybrid methods. *ACM Trans. Graph.*, 14(3):233–265, 1995.

[Nus28] W. Nusselt. Graphische bestimmung des winkelverhaltnisses bei der wärmestrahlung. *Zeitschrift des Vereines Deutscher Ingenieure*, 72(20):673, 1928.

[Opp56] A. K. Oppenheim. Radiation analysis by the network method. *Transactions of ASME, Journal of Heat Transfer*, 78:725–735, 1956.

[OY66] R. K. Osborn and S. Yip. *The Foundation of Neutron Transport Theory.* Gordon and Breach, Science Publishers, Inc., 1966.

[Pan05] T. Panczak. Non-grey and temperature dependent radiation analysis methods. In *TFAWS Short Courses*, 2005.

[Pho75] B. T. Phong. Illumination for computer-generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[Pla93] P. Planas. Statistical error control for radiative software based on monte carlo methods. *Proceedings of the 23rd ICES Conference, SAE Technical Paper 932258*, 1993.

[Pla97] P. Planas. Accuracy of monte carlo ray-tracing thermal radiation calculations: a practical discussion. *Proceedings of the 6th European Symposium on Space Environmental Control Systems*, pages 579–591, August 1997.

[PM95] S. N. Pattanaik and S. P. Mudur. Adjoint equations and random walks for illumination computation. *ACM Transactions on Graphics*, 14(1):77–102, January 1995.

[Roc03] P. Rochus. *Contrôle Thermique - Echanges Radiatifs.* Université de Liège, 2003. Notes de cours.

[RT90] H. E. Rushmeier and K. E. Torrance. Extending the radiosity method to include specularly reflecting and translucent materials. *ACM Trans. Graph.*, 9(1):1–27, 1990.

[Rue08] C. Ruel. New technology for modeling and solving heat transfer using TMG. In 22^{nd} *European Workshop on Thermal and ECLS Software*, 2008.

[RUL00] J. Revelles, C. Ureña, and M. Lastra. An efficient parametric algorithm for octree traversal. In *Proc. Winter School on Computer Graphics*, pages 212–219, 2000.

[RV04] V. Rosell and R. Vivo. Improving form factor accuracy by projecting onto the hemisphere base. *Universidad Politécnica de Valencia (Spain)*, 2004.

[RW80] S. M. Rubin and T. Whitted. A three-dimensional representation for fast rendering of complex scenes. *Computer Graphics*, 14(3):110–116, July 1980.

[SAS92] B. E. Smits, J. R. Arvo, and D. H. Salesin. An importance-driven radiosity algorithm. *Computer Graphics (ACM SIGGRAPH 92 Proceedings)*, 26(4):273–282, July 1992.

[SG69] J. Spanier and E. Gelbard. *Monte Carlo Principles and Neutron Transport Problems.* Addison-Wesley Publishing Company, 1969.

[SH93a] P. Schröder and P. Hanrahan. A closed form expression for the form factor between two polygons. *Research Report*, January 1993.

[SH93b] P. Schröder and P. Hanrahan. On the form factor between two polygons. *Research Report*, January 1993.

[SH01] R. Siegel and J. Howell. *Thermal Radiation Heat Transfer, fourth edition.* Taylor and Francis, 2001.

[SHH99] S. Suri, P. M. Hubbard, and J. F. Hughes. Analyzing bounding boxes for object intersection. *ACM Trans. Graph.*, 18(3):257–277, 1999.

[SP89] F. Sillion and C. Puech. A general two-pass method integrating specular and diffuse reflection. *Computer Graphics*, 23(3):335–344, July 1989.

[SP94] F. X. Sillion and C. Puech. *Radiosity and Global Illumination.* Morgan Kaufmann Publishers, Inc., 1994.

[Stü95] W. Stürzlinger. Exact projections onto the hemisphere. *Computer Graphics*, May 1995.

[SWZ96] P. Shirley, C. Wang, and K. Zimmerman. Monte carlo techniques for direct lighting calculations. *ACM Trans. Graph.*, 15(1):1–36, 1996.

[TL95] R. P. Taylor and R. Luck. A case study of view-factor rectification procedures for diffuse-gray radiation enclosure computations. In *NASA. Lewis Research Center, The Sixth Annual Thermal and Fluids Analysis Workshop p 115-131 (SEE N95-27345 09-34)*, pages 115–131, January 1995.

[Tsu92] G. T. Tsuyuki. TRASYS form factor matrix normalization. In *NASA. Lewis Research Center, The Fourth Annual Thermal and Fluids Analysis Workshop p 71-82 (SEE N93-13385 03-34)*, pages 71–82, 1992.

[VB07] P. Vueghs and P. Beckers. Comparison of samcef and hierarchical technique for radiative heat transfer. In *10th Samtech Users Conference*, 2007.

[VdKPB08a] P. Vueghs, H. P. de Koning, O. Pin, and P. Beckers. Use of geometry in finite elements thermal radiation combined with ray tracing. In M. Hogge, R. Van Keer, L. Noels, L. Stainier, J.-P. Ponthot, J.-F. Remacle, and E. Dick, editors, *Fourth International Conference on Advanced COmputational Methods in ENgineering (ACOMEN 2008).* University of Liege, Belgium, May 2008.

[VdKPB08b] P. Vueghs, H. P. de Koning, O. Pin, and P. Beckers. Utilisation de la géométrie dans le calcul éléments finis thermique radiatif utilisant le lancer de rayons. In P. Millan, editor, *Congrès Français de Thermique – Thermique Aéronautique et Spatiale*, pages 833–838. ONERA Centre Midi-Pyrénées, Juin 2008.

[Ver08] P. Versali. Simulation de l'effet des réflexions multiples sur la signature thermique de scènes 3d par le calcul de cartes azimut-elévation. Master's thesis, Université de Liège - Faculté des Sciences Appliquées, 2007-2008.

[VG95] E. Veach and L. J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, 1995.

[vL89] J. van Leersum. A method for determining a consistent set of radiation view factors from a set generated by a nonexact method. *International Journal of Heat and Flow*, 10(1):83–85, 1989.

[VLV99] M. J. Vicent, J. Lluch, and R. Vivo. Un método para el cálculo del factor de forma alternativo al hemi-cubo. *Congreso Español de Informática Gráfica CEIG'99*, pages 303–316, 1999.

[VLV04] M. J. Vicent, J. Lluch, and R. Vivo. Natural lighting of vegetation based on polar planes. *WSCG poster proceedings (WSCG'2004)*, 2004.

[Vue05] P. E. Vueghs. Calcul des facteurs de vue – synthèse des méthodes disponibles. Technical report sf-251, Université de Liège, 2005.

[Vue06a] P. E. Vueghs. Equations de thermique – démonstrations et illustrations. Technical report sf-256, Université de Liège, 2006.

[Vue06b] P. E. Vueghs. Présentation de la méthode hiérarchique. Technical report sf-255, Université de Liège, 2006.

[Vue07] P. E. Vueghs. Intersections rayons–surfaces. Technical report sf-259, Université de Liège, 2007.

[Vue08] P. E. Vueghs. Application de la réciprocité et de la fermeture dans le calcul des facteurs d'echange. Technical report sf-262, Université de Liège, 2008.

[VVL+01] R. Vivo, M. J. Vicent, J. Lluch, R. Molla, and P. Jorquera. Study of the form factor calculation by single polar plane. *IASTED International Conference on Visualisation imaging and image processing*, 2001.

[vW84] J. J. van Wijk. Ray tracing objects defined by sweeping planar cubic splines. *ACM Transactions on Graphics*, 3(3):223–237, July 1984.

[WC87] J. R. Wallace and M. F. Cohen. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. *Computer Graphics*, 21(4):311–320, 1987.

[WEH89] J. R. Wallace, K. A. Elmquist, and E. A. Haines. A ray tracing algorithm for progressive radiosity. *Computer Graphics*, 23(3):315–324, July 1989.

[WHG84] H. Weghorst, G. Hooper, and D. P. Greenberg. Improved computational methods for ray tracing. *ACM Transactions on Graphics*, 3(1):52–69, 1984.

[Whi80] T. Whitted. An improved illumination model for shaded display. *Communication of the ACM*, 23(6):343–349, 1980.

[Wik] Wikipedia. Constructive solid geometry. http://en.wikipedia.org/wiki/Constructive_solid_geometry.

[Won76] H. Y. Wong. *Handbook of Essential Formulae and Data on Heat Transfer for Engineers*. Longman, 1976.

[Zee02] C. N. Zeeb. *Performances and Accuracy Enhancements of Radiative Heat Transfer Modeling via Monte Carlo*. PhD thesis, Department of Mechanical Engineering - Colorado State University, 2002.

[ZH97] Hansong Zhang and Kenneth E. Hoff. Fast backface culling using normal masks. In *Symposium on Interactive 3D Graphics*, pages 103–106, 189, 1997.

[ZS66] M. B. Ziering and A. F. Sarofim. The electrical network analog to radiative transfer : Allowance for specular reflection. *ASME Journal of Heat Transfer*, 88:341–342, 1966.