Appendix D

New Technology for Modeling and Solving Radiative Heat Transfer using TMG

Christian Ruel
(MAYA, Canada)

Abstract

As engineers increasingly rely on numerical models within the framework of a collaborative development process, demands on solution performance are becoming much more severe. In order to effectively address these demands, we believe that a massive, quantum improvement in the solution speed of spacecraft thermal analysis systems is required. To achieve such a breakthrough, MAYA has undertaken the parallelization of the TMG software system, enabling full exploitation of multiprocessing computer environments (consisting of multiprocessor servers or networked workstations or clusters).

Maya is also developing an innovative numerical method for the simulation of radiative heat transfer in cryogenic systems, based on the radiosity method, in which the radiating spectrum is discretized into spectral bands. A surface at a given temperature will radiate and absorb in all the bands, but the coefficients of emissivity and absorptivity - while equal to each other in a given band - will vary from one band to the next.

22st European Workshop on
Thermal and ECLS Software

New Technology for Modeling and Solving Radiative Heat Transfer using TMG

October 28, 2008

Introduction

MAYA has undertaken development of two major new technologies for radiative heat transfer simulation:

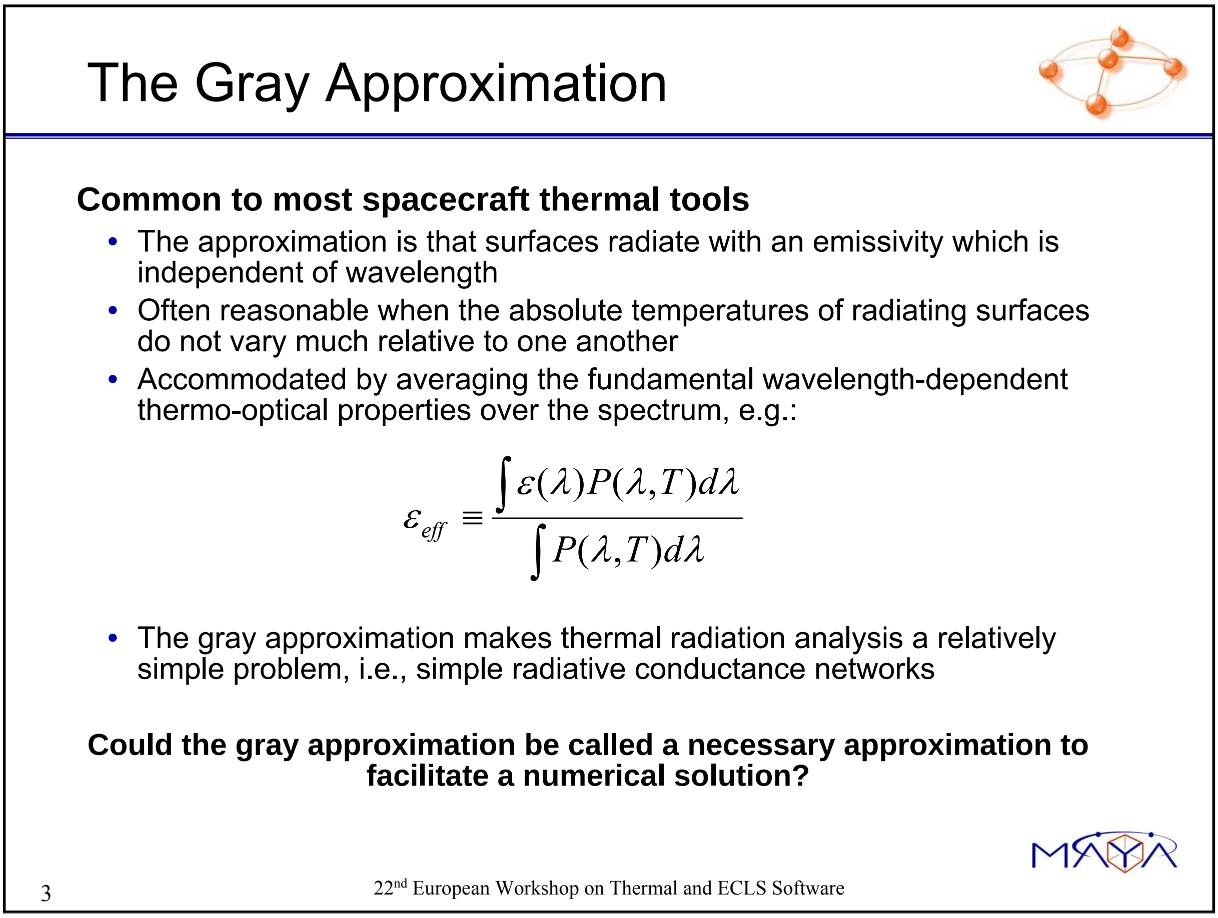
- Enable treatment of wavelength dependence in radiative exchange
- New solver technology to enable faster processing of high definition models
- Projects co-sponsored by the Canadian Space Agency

Nongray radiative exchange

- Gray approximation is widely used in spacecraft thermal analysis
- Treatment of nongray effects become important at cryogenic temperatures

Parallelization

- Target software modules which use the most CPU
- Provide a parallel solution which is deployable to most client sites today
- View factor computations are “inherently parallel,” so have been targeted as the first candidates for parallelization



The Gray Approximation

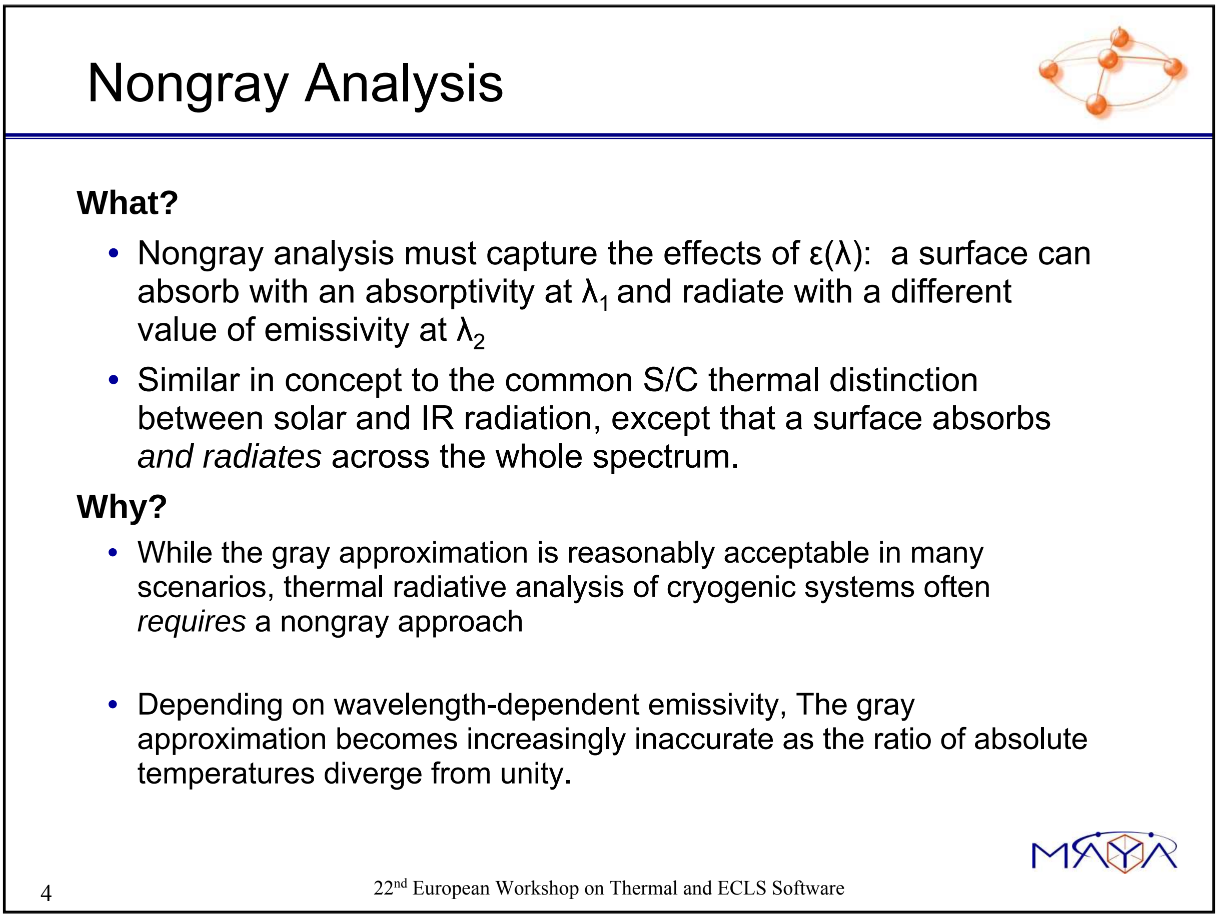
Common to most spacecraft thermal tools

- The approximation is that surfaces radiate with an emissivity which is independent of wavelength
- Often reasonable when the absolute temperatures of radiating surfaces do not vary much relative to one another
- Accommodated by averaging the fundamental wavelength-dependent thermo-optical properties over the spectrum, e.g.:

$$\varepsilon_{eff} \equiv \frac{\int \varepsilon(\lambda) P(\lambda, T)\, d\lambda}{\int P(\lambda, T)\, d\lambda}$$

- The gray approximation makes thermal radiation analysis a relatively simple problem, i.e., simple radiative conductance networks

Could the gray approximation be called a necessary approximation to facilitate a numerical solution?



Nongray Analysis

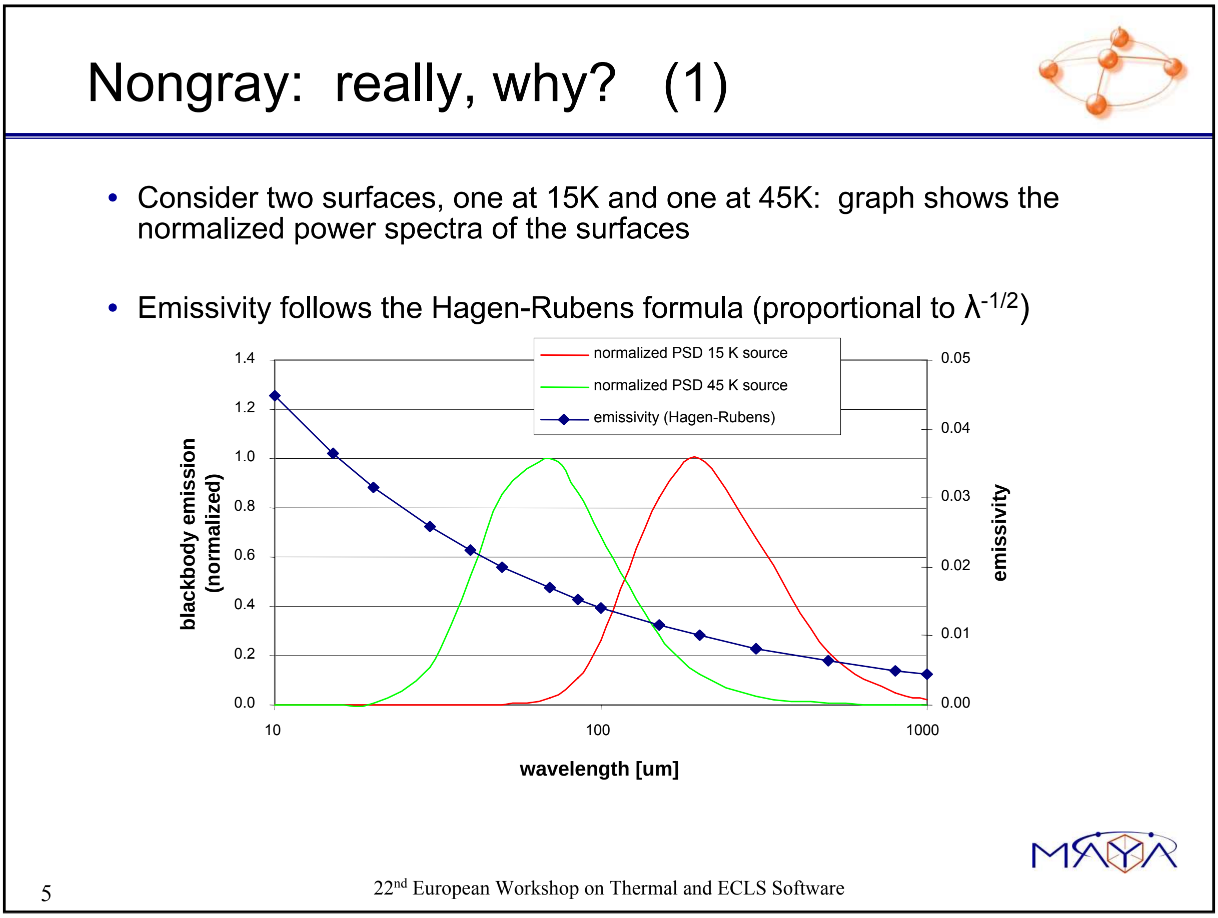
What?

- Nongray analysis must capture the effects of $\varepsilon(\lambda)$: a surface can absorb with an absorptivity at λ_1 and radiate with a different value of emissivity at λ_2
- Similar in concept to the common S/C thermal distinction between solar and IR radiation, except that a surface absorbs *and radiates* across the whole spectrum.

Why?

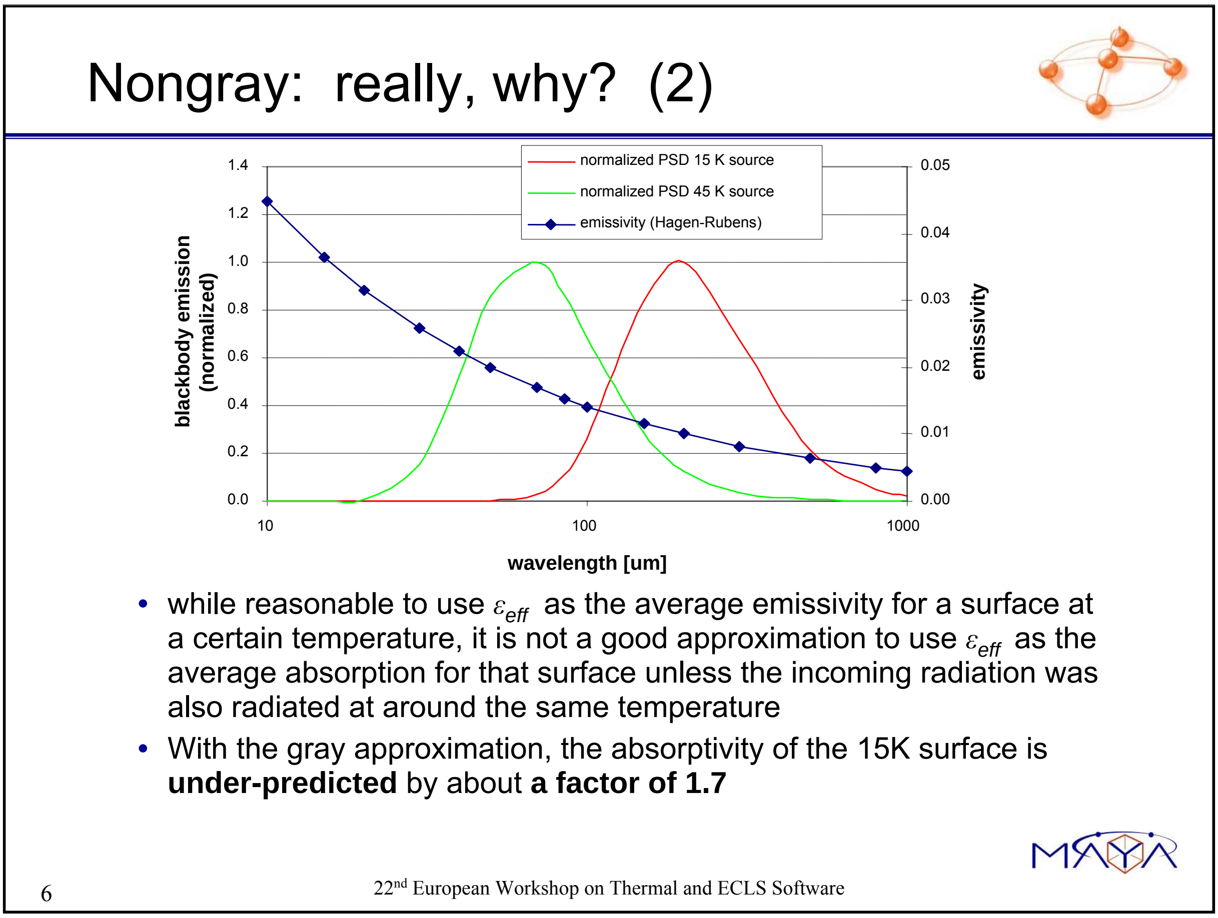
- While the gray approximation is reasonably acceptable in many scenarios, thermal radiative analysis of cryogenic systems often *requires* a nongray approach

- Depending on wavelength-dependent emissivity, The gray approximation becomes increasingly inaccurate as the ratio of absolute temperatures diverge from unity.



Nongray: really, why? (1)

- Consider two surfaces, one at 15K and one at 45K: graph shows the normalized power spectra of the surfaces
- Emissivity follows the Hagen-Rubens formula (proportional to $\lambda^{-1/2}$)



Nongray: really, why? (2)

- while reasonable to use ε_{eff} as the average emissivity for a surface at a certain temperature, it is not a good approximation to use ε_{eff} as the average absorption for that surface unless the incoming radiation was also radiated at around the same temperature
- With the gray approximation, the absorptivity of the 15K surface is **under-predicted** by about **a factor of 1.7**

Nongray Analysis in TMG

Discretization

- The fundamental equations for radiative exchange between surfaces are discretized in terms of wavelength
- The discretization takes the form of *N wavelength bands*
- Thermo-optical properties are now defined band-wise:

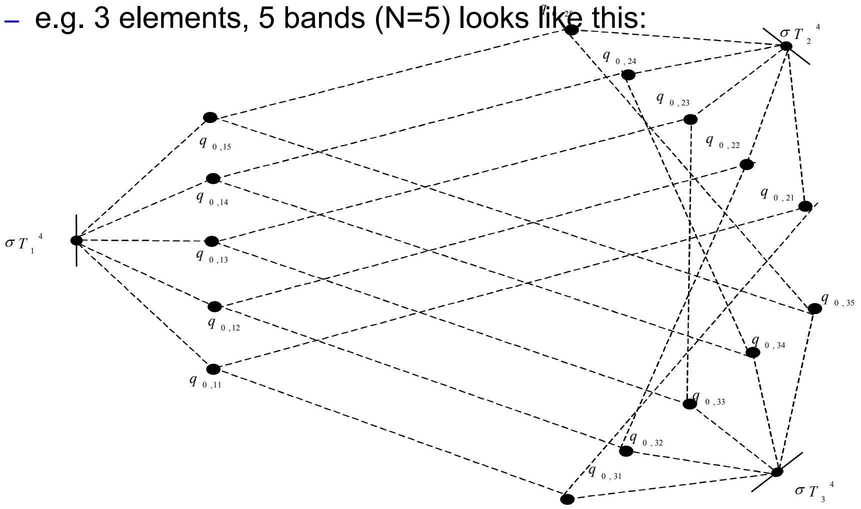
$$\varepsilon_{kg} \equiv \frac{\int_{\lambda_{g-1}}^{\lambda_g} \varepsilon(\lambda)P(\lambda,T)d\lambda}{\int_{\lambda_{g-1}}^{\lambda_g} P(\lambda,T)d\lambda} \approx \frac{\int_{\lambda_{g-1}}^{\lambda_g} \varepsilon(\lambda)d\lambda}{\int_{\lambda_{g-1}}^{\lambda_g} d\lambda} = \frac{1}{\Delta\lambda}\int_{\lambda_{g-1}}^{\lambda_g} \varepsilon(\lambda)d\lambda$$

- *g* is the band number
- number of bands and band spacing is user-input

Nongray Analysis in TMG

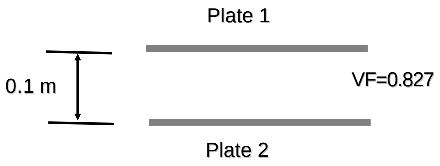
Multiband Radiosity Method

- The radiosity method has been rederived using the band structure
- Each radiating element takes N radiosity (‘Oppenheim’) elements
- A distinct radiative conductance network is created in each band
  - e.g. 3 elements, 5 bands (N=5) looks like this:



Nongray Validation

Two Plates in space



- Plate 1:
  area = 1 m^2, Sink @ T_1, ε_{1g}, g=1..N
- Plate 2:
  area = 1 m^2, ε_{2g}, g=1..N

- Total heat emitted and absorbed by plate 2 can be derived analytically:

$$Q_{2,emit} = \sum_{g=1}^{N} \varepsilon_{2,g} p_g(T_2) A \sigma T_2^4$$

$$Q_{2,abs} = \sigma \sum_{g=1}^{N} \left\{ \left[\varepsilon_{1g} A_1 p_g(T_1) T_1^4 VF_{12} \varepsilon_{2g} + \left(\varepsilon_{2g}\right)^2 A_2 p_g(T_2) T_2^4 VF_{21} (1-\varepsilon_{1g}) VF_{12} \right] \times \left[\frac{1}{1 - VF_{12} VF_{21} (1-\varepsilon_{1g})(1-\varepsilon_{2g})} \right] \right\}$$

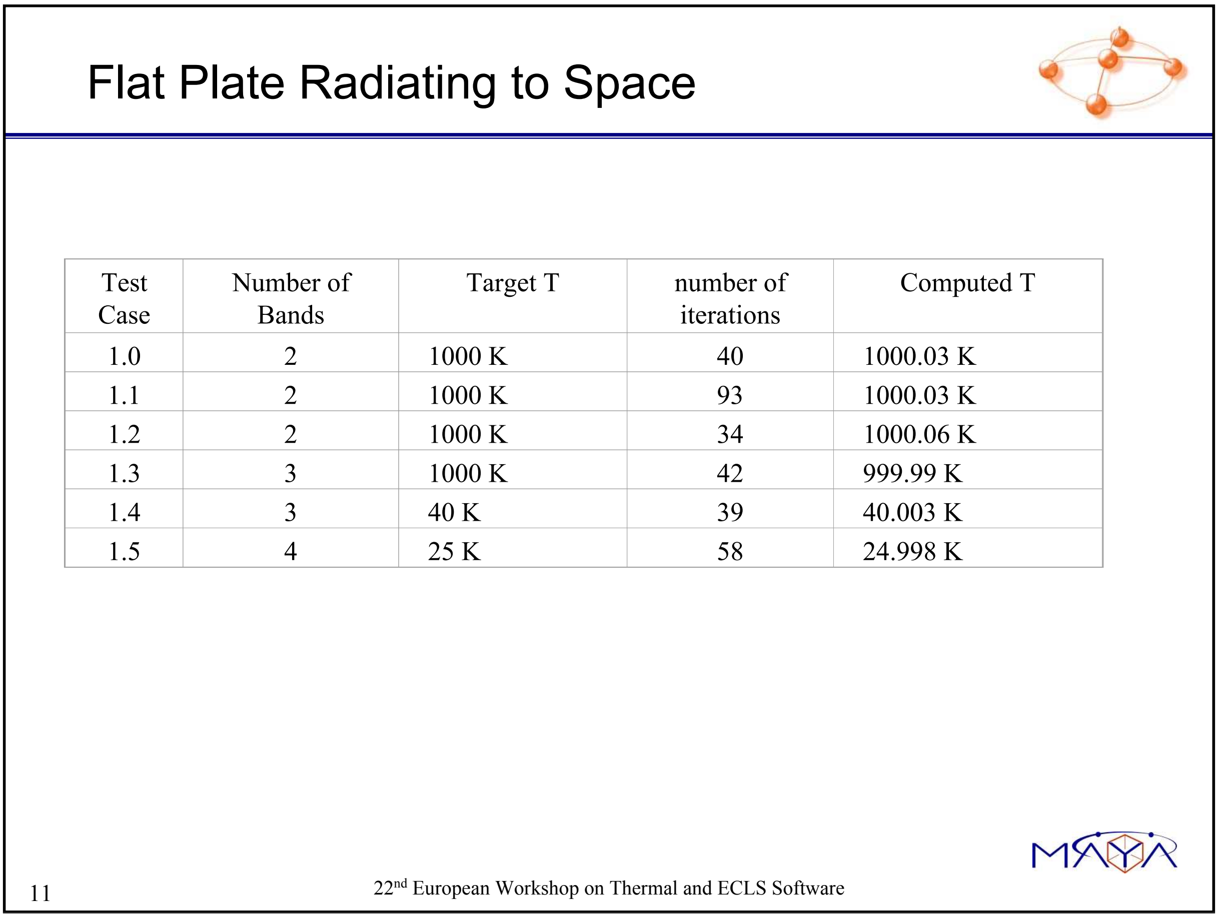
- 4 test cases varying T_1, N, ε_{1g} and ε_{2g}

Nongray Validation

Two Plates in Space: Test Matrix

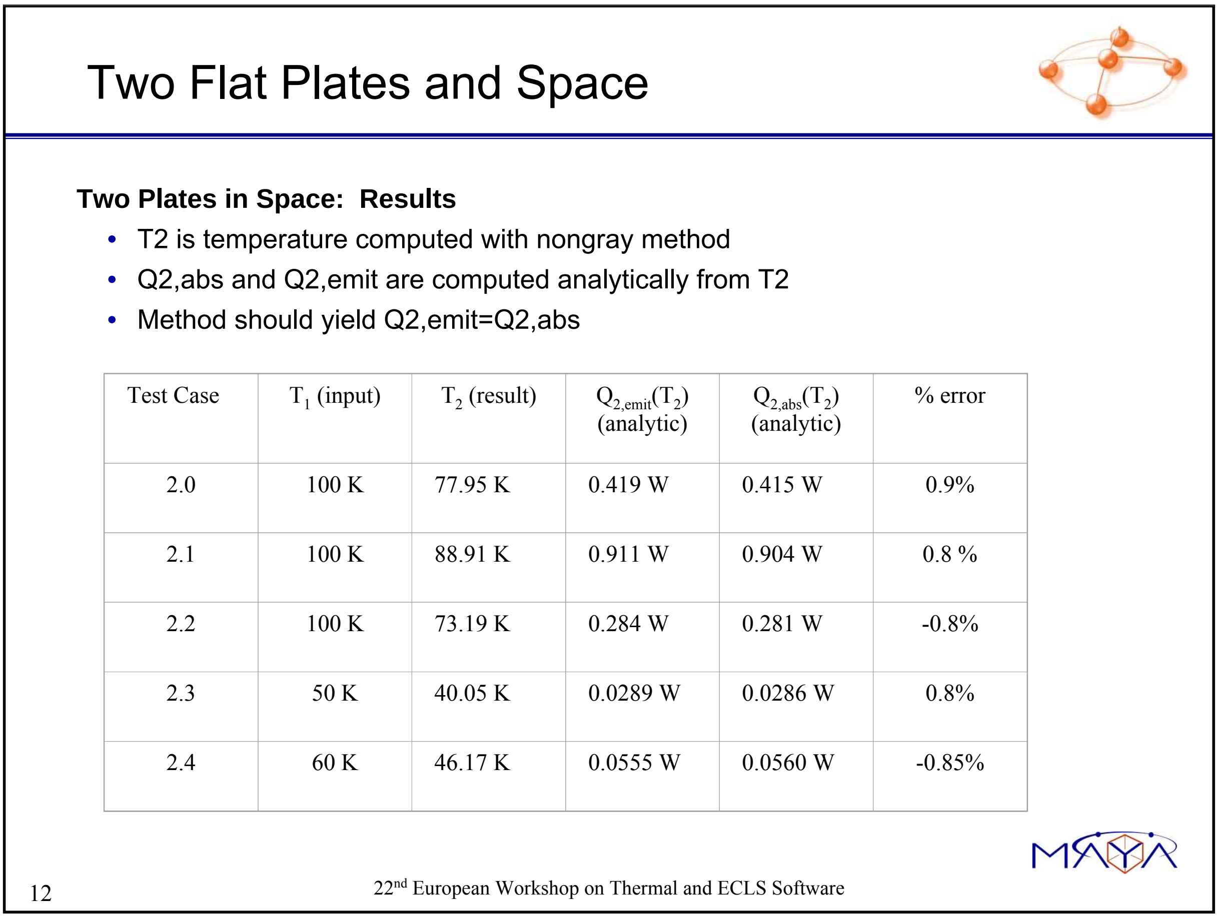
Test Case	Number of Bands	Band limits (micrometers)					T_1 = Element 1 Temperature (sink)
		λ_0	λ_1	λ_2	λ_3	λ_4	
2.0	1	-	-	-	-	-	100 K
2.1	2	0	40.0	4.E3	-	-	100 K
2.2	2	0	40.0	4.E3	-		100 K
2.3	2	0	40	4.E3	6.E3		50 K
2.4	4	0	40.0	80.0	120.0	1.2E5	60 K

Test Case	Number of Bands	Band Emissivities (element 1)				Band Emissivities (element 2)			
		ε_1	ε_2	ε_3	ε_4	ε_1	ε_2	ε_3	ε_4
2.0	1	0.5	-	-	-	0.5	-	-	-
2.1	2	0.1	0.25	-	-	0.1	0.2	-	-
2.2	2	0.5	0.05	-	-	0.1	0.2	-	-
2.3	2	0.1	0.25	-	-	0.1	0.2	-	-
2.4	4	0.1	0.25	0.15	.05	0.3	0.25	0.2	0.18



Flat Plate Radiating to Space

Test Case	Number of Bands	Target T	number of iterations	Computed T
1.0	2	1000 K	40	1000.03 K
1.1	2	1000 K	93	1000.03 K
1.2	2	1000 K	34	1000.06 K
1.3	3	1000 K	42	999.99 K
1.4	3	40 K	39	40.003 K
1.5	4	25 K	58	24.998 K



Two Flat Plates and Space

Two Plates in Space: Results

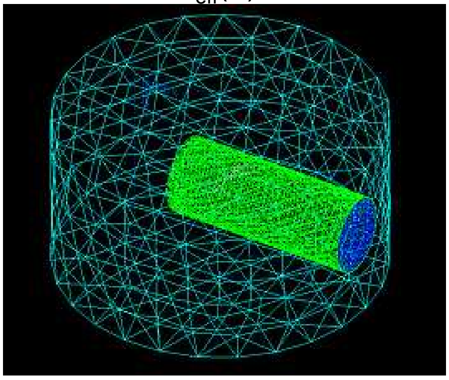
- T2 is temperature computed with nongray method
- Q2,abs and Q2,emit are computed analytically from T2
- Method should yield Q2,emit=Q2,abs

Test Case	T_1 (input)	T_2 (result)	$Q_{2,emit}(T_2)$ (analytic)	$Q_{2,abs}(T_2)$ (analytic)	% error
2.0	100 K	77.95 K	0.419 W	0.415 W	0.9%
2.1	100 K	88.91 K	0.911 W	0.904 W	0.8 %
2.2	100 K	73.19 K	0.284 W	0.281 W	-0.8%
2.3	50 K	40.05 K	0.0289 W	0.0286 W	0.8%
2.4	60 K	46.17 K	0.0555 W	0.0560 W	-0.85%

Nongray: Sample Application

Simplified model of telescope instrument with cryogenic optics

- ε(λ) for the three materials in the model were used to determine emissivities for three separate analyses :
  - classical gray analysis with constant ε_{eff}
  - gray model with temperature dependent emissivities $\varepsilon_{eff}(T)$
  - two-band nongray model



- Cryocooler modeled as a 31 K nongeometric sink coupled to the end of the telescope
- Critical design issue is how much heat load goes into the cryocooler

---

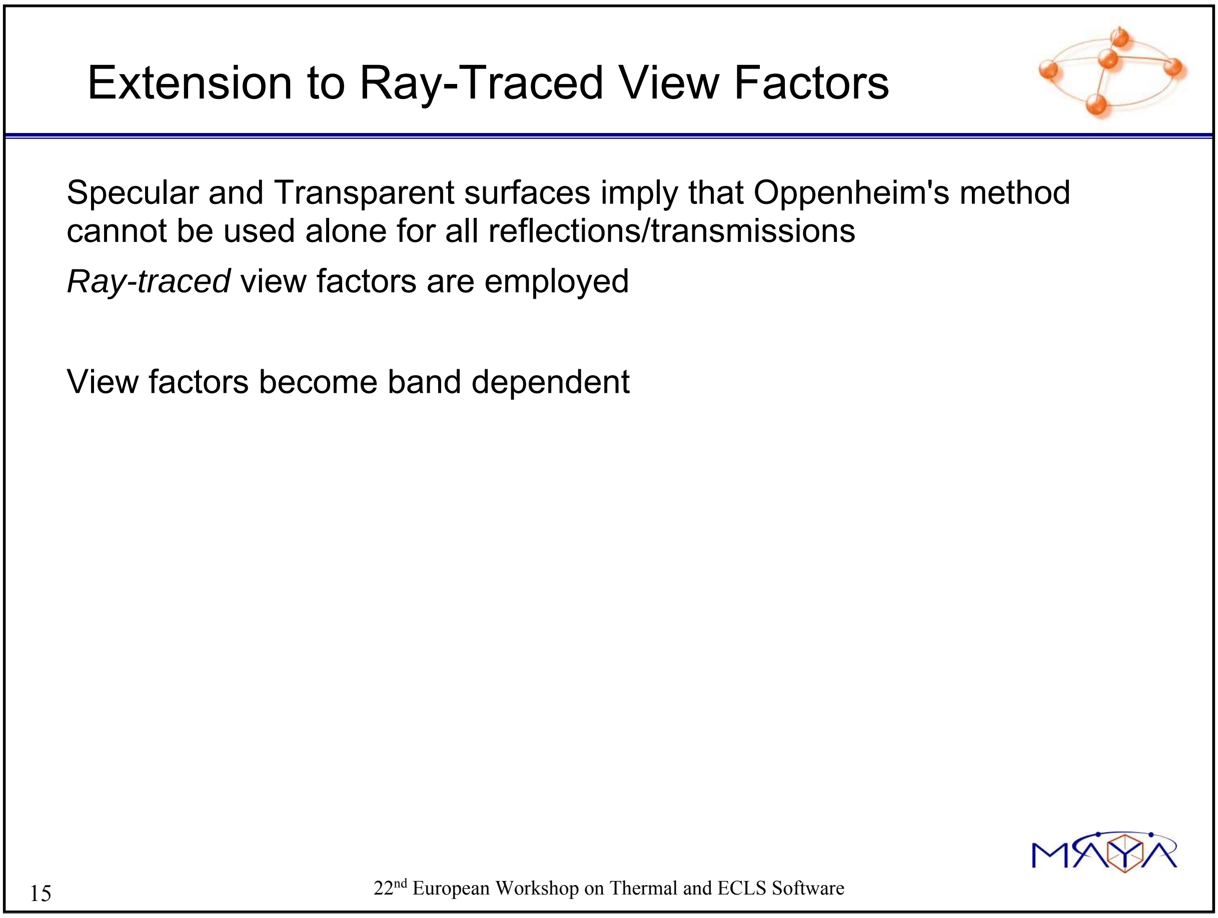
Nongray: Sample Application

Comparison of Heat Loads into Cryocooler

Case	Heat Load into 31K Cryocooler
Classical Gray Analysis	0.168 W
Gray with ε(T)	0.159 W
Nongray 2 bands	**0.209 W**

Remarks:

- The 2 band nongray calculation shows the cryocooler needs to draw about 24% more heat than that shown by the gray analysis.
- Temperature dependent emissivity gives worse results!

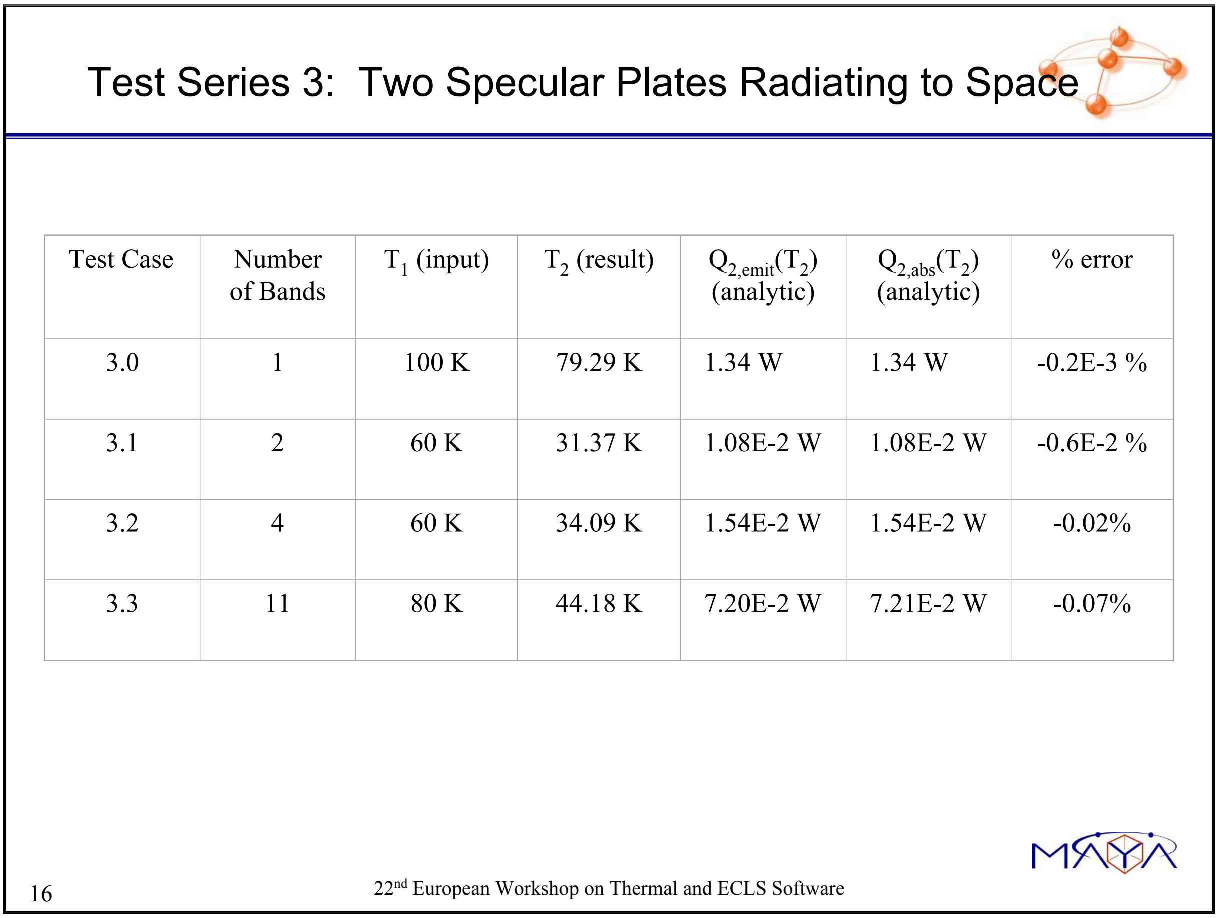


Extension to Ray-Traced View Factors

Specular and Transparent surfaces imply that Oppenheim's method cannot be used alone for all reflections/transmissions

Ray-traced view factors are employed

View factors become band dependent

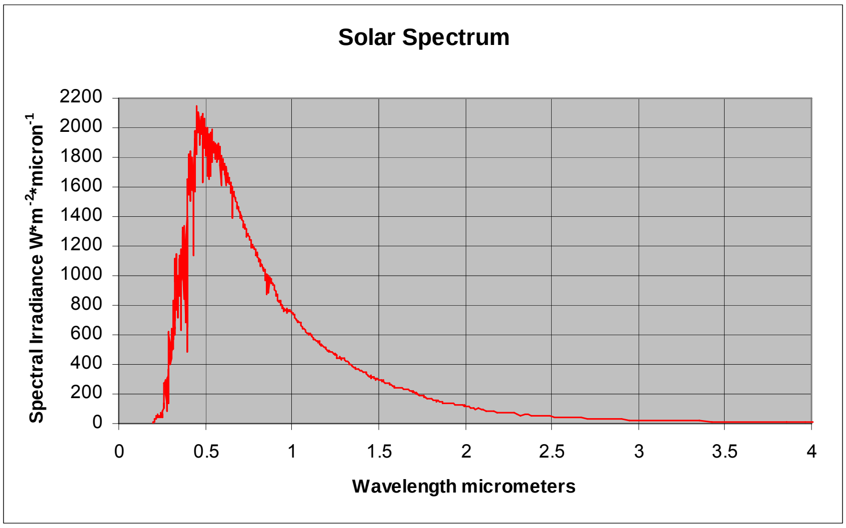


Test Series 3: Two Specular Plates Radiating to Space

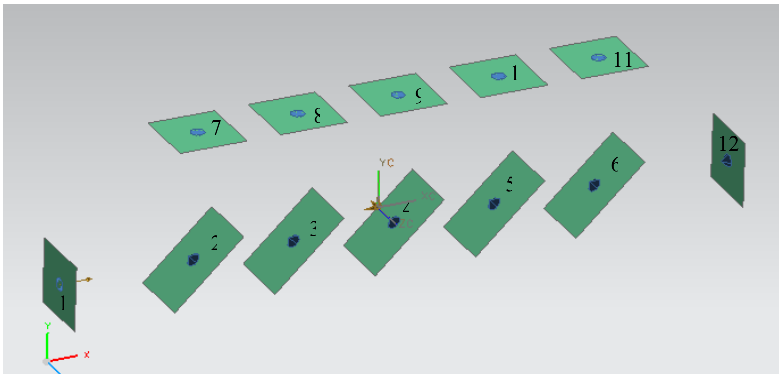
Test Case	Number of Bands	T_1 (input)	T_2 (result)	$Q_{2,emit}(T_2)$ (analytic)	$Q_{2,abs}(T_2)$ (analytic)	% error
3.0	1	100 K	79.29 K	1.34 W	1.34 W	-0.2E-3 %
3.1	2	60 K	31.37 K	1.08E-2 W	1.08E-2 W	-0.6E-2 %
3.2	4	60 K	34.09 K	1.54E-2 W	1.54E-2 W	-0.02%
3.3	11	80 K	44.18 K	7.20E-2 W	7.21E-2 W	-0.07%

Solar Spectrum

For multi-band analysis, the solar spectrum is integrated over bands defined by the user. Can also be input.

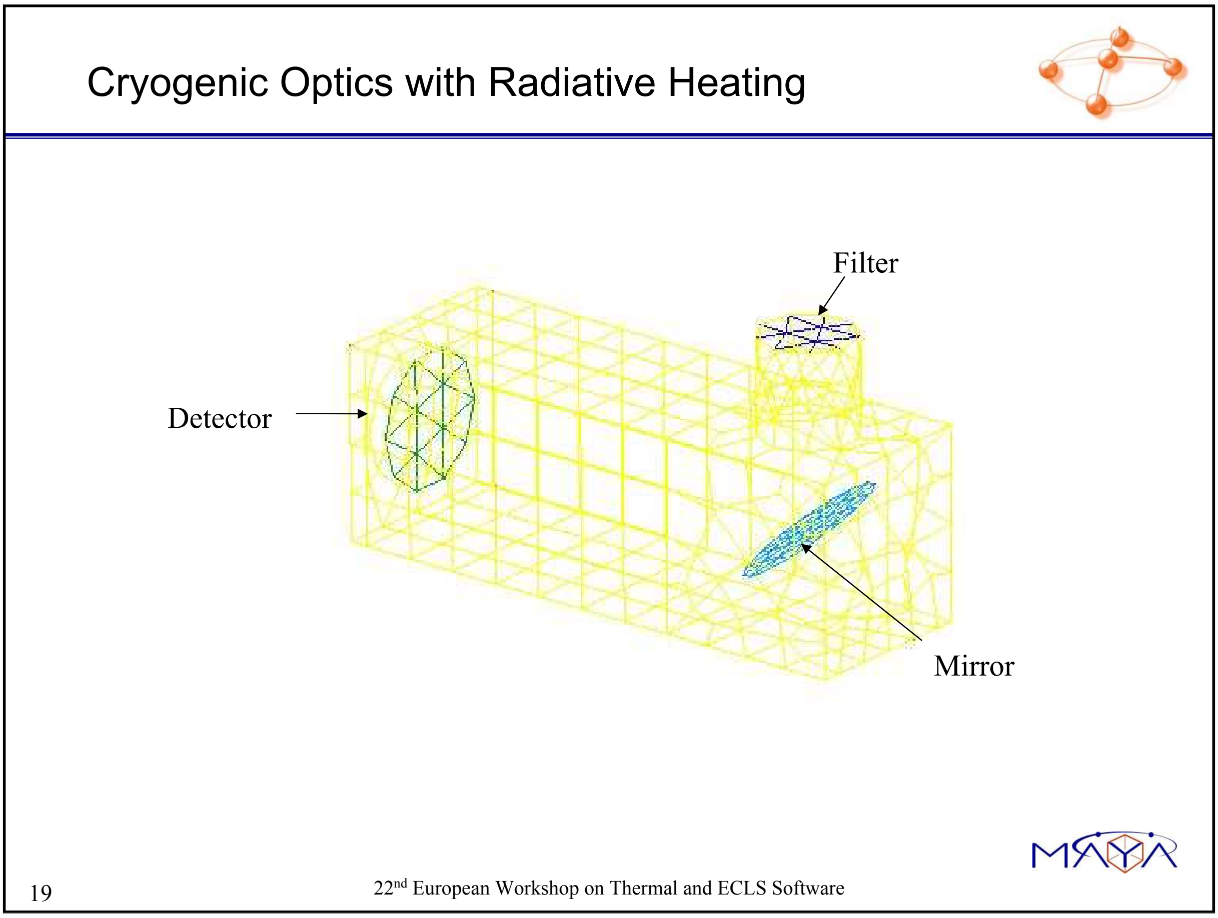


Wavelength dependent heat sources

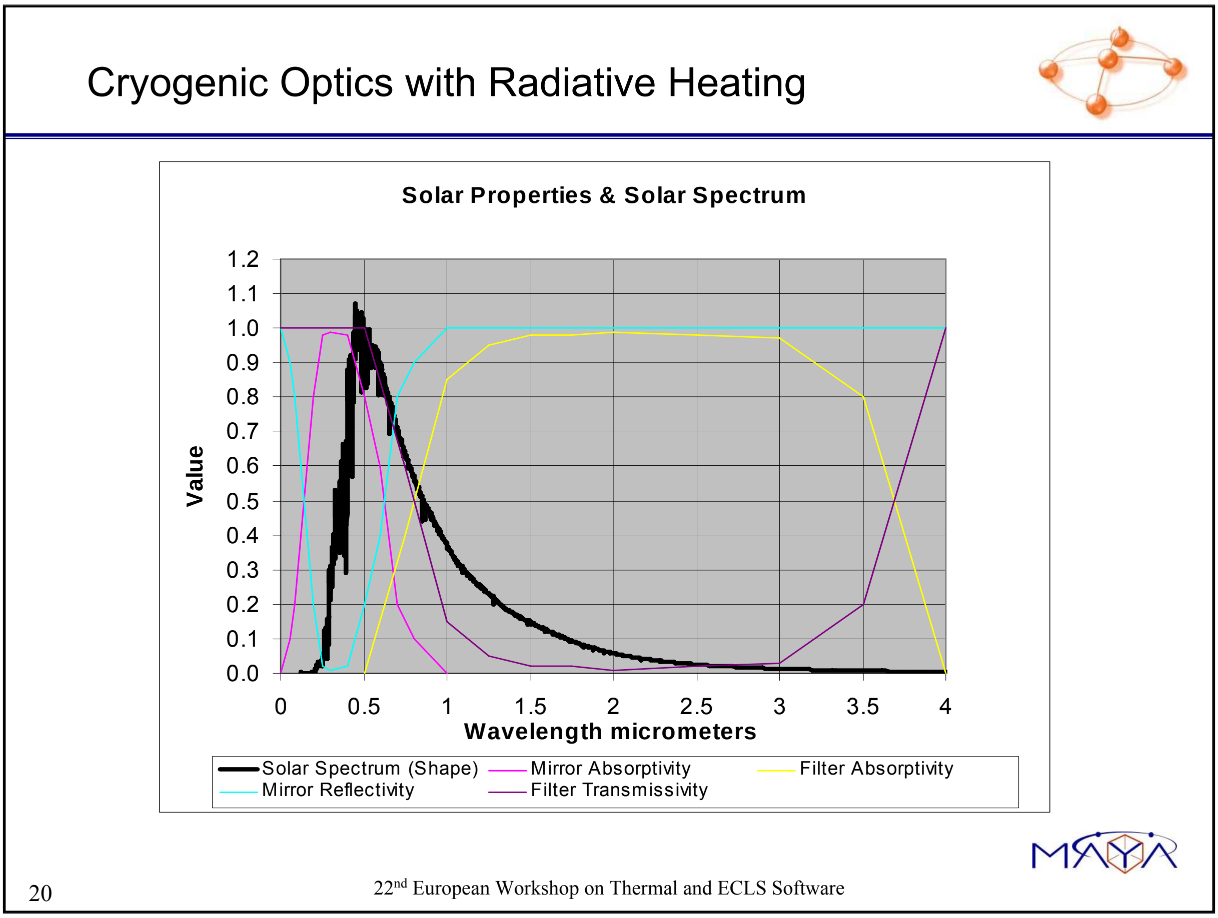


11 bands used, each intermediate plate able to reflect & transmit energy in various bands

One simple and one more complicated set of material properties



Cryogenic Optics with Radiative Heating

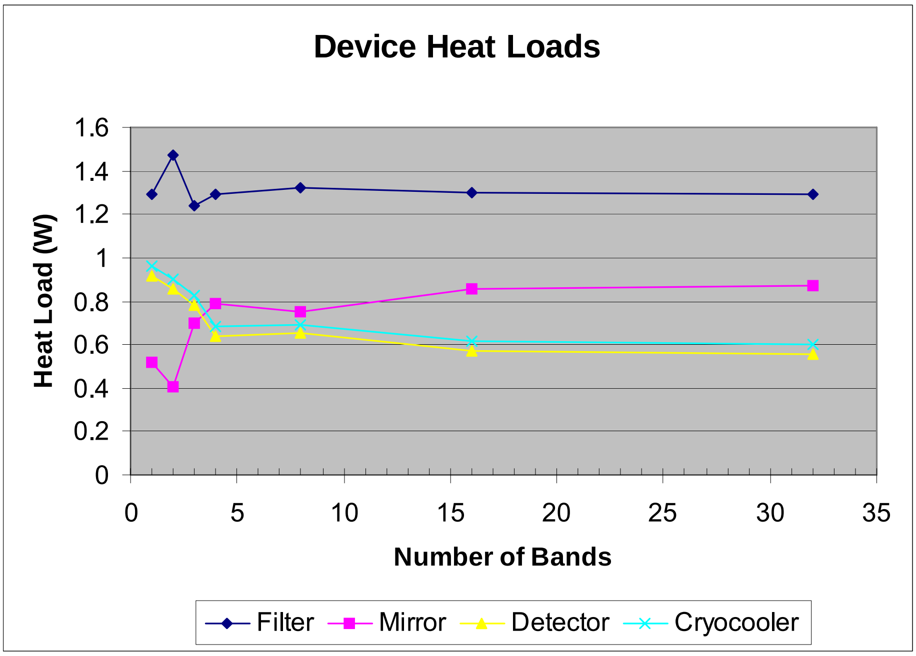


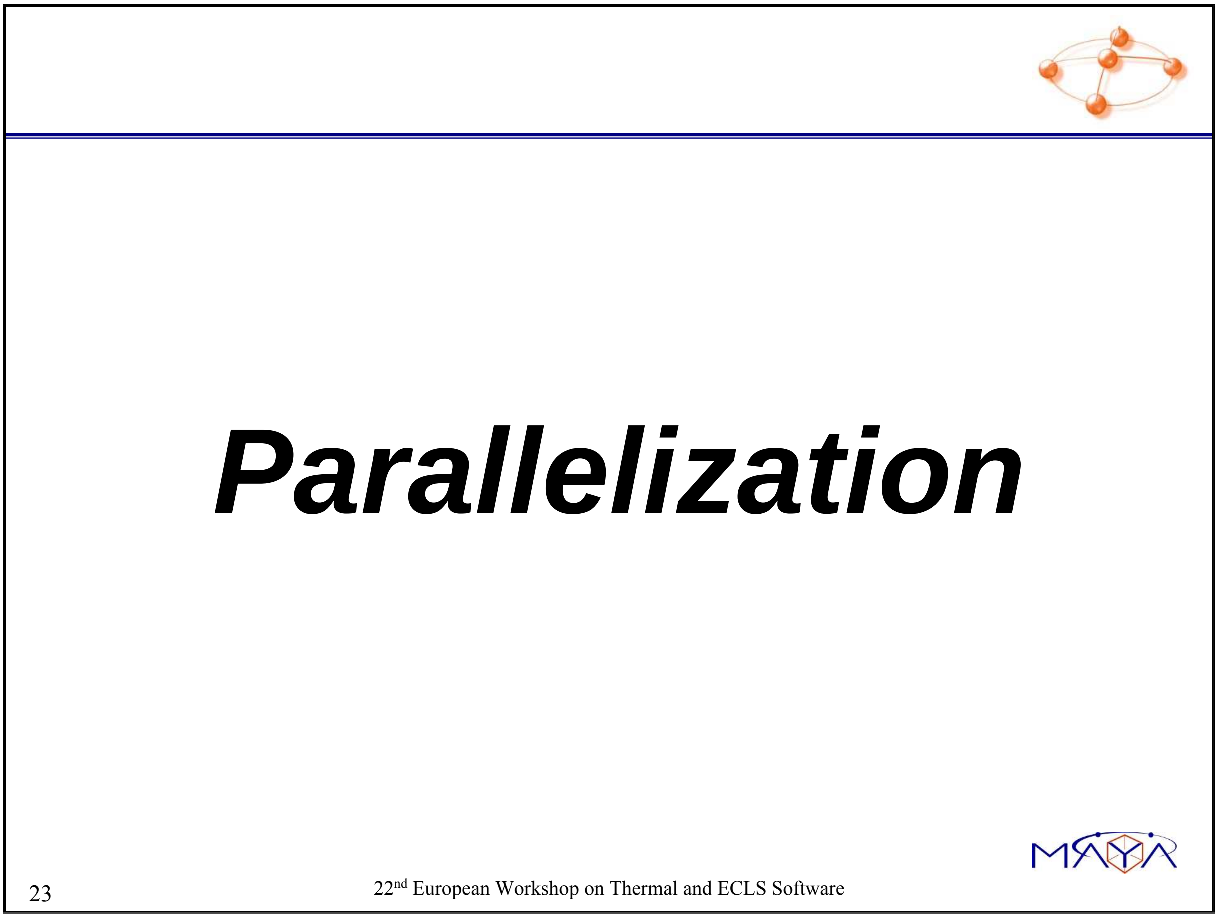
Cryogenic Optics with Radiative Heating

Cryogenic Optics with Radiative Heating

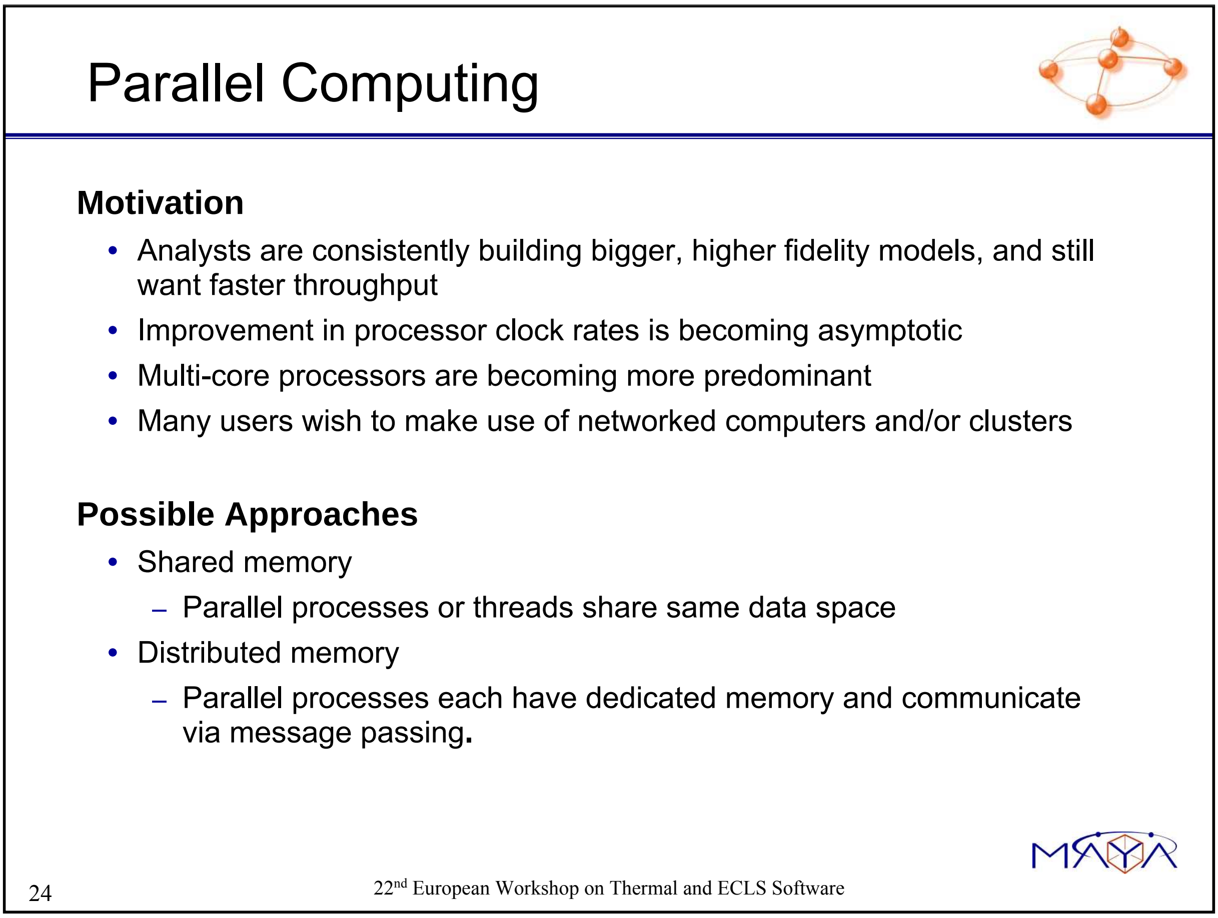
Case	Solar Bands	Solar Load on Lens (W)	Solar Load on Mirror (W)	Solar Load on Sample (W)	Heat Flow into Cryocooler (W)
Gray	1	1.29	0.515	0.915	0.958
3 bands	2	1.47	0.404	0.860	0.903
4 bands	3	1.24	0.698	0.780	0.823
5 bands	4	1.29	0.788	0.639	0.683
9 bands	8	1.32	0.748	0.650	0.693
17 bands	16	1.30	0.854	0.570	0.613
33 bands	32	1.29	0.872	0.559	0.603

Cryogenic Optics with Radiative Heating





Parallelization



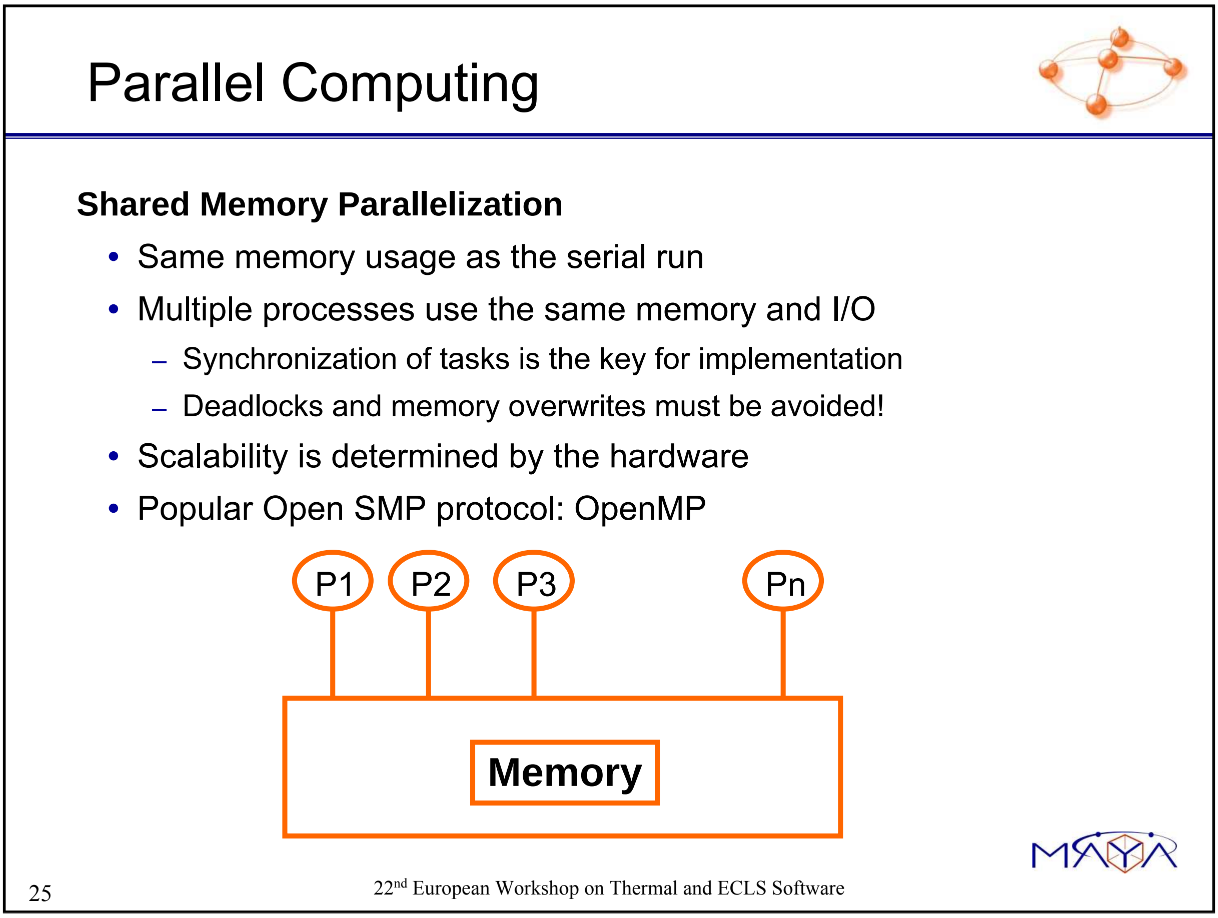
Parallel Computing

Motivation

- Analysts are consistently building bigger, higher fidelity models, and still want faster throughput
- Improvement in processor clock rates is becoming asymptotic
- Multi-core processors are becoming more predominant
- Many users wish to make use of networked computers and/or clusters

Possible Approaches

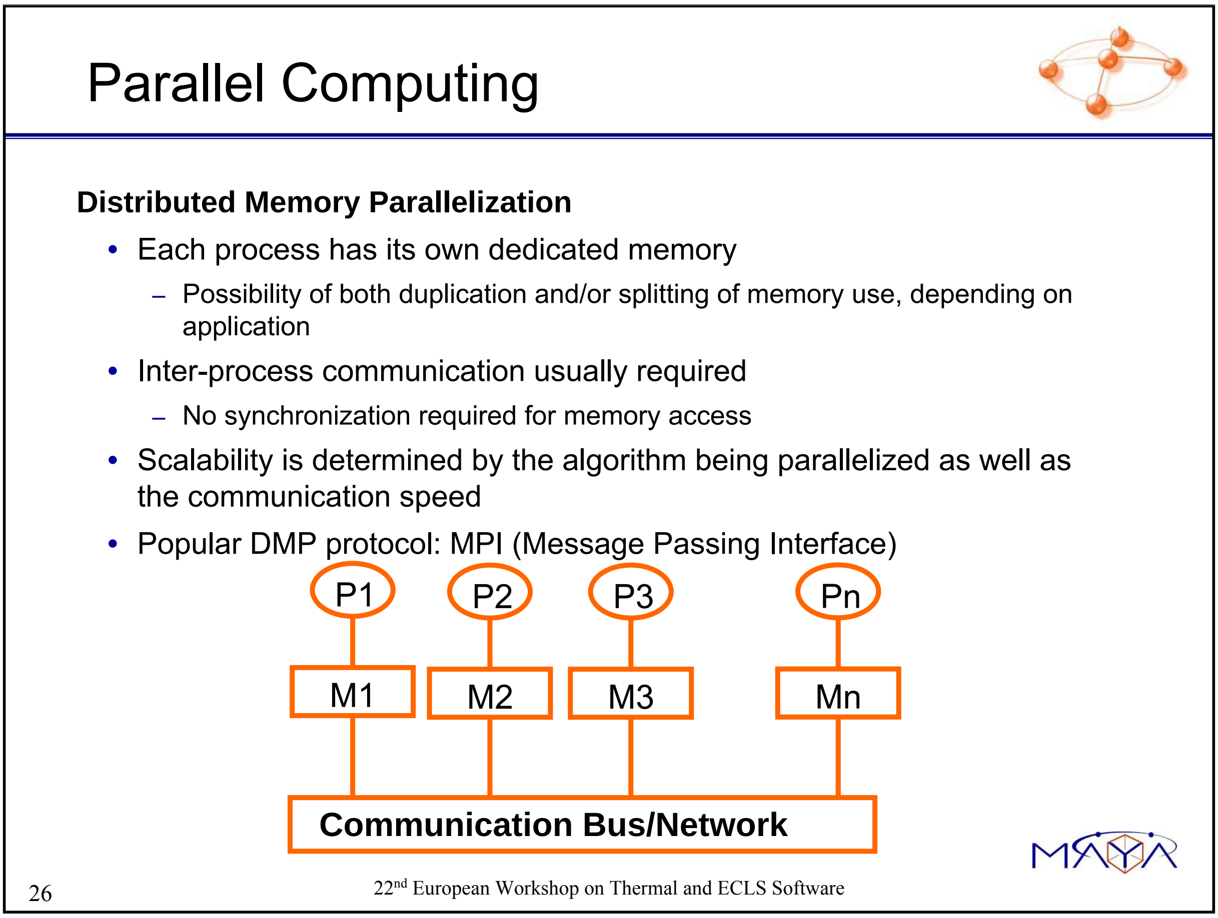
- Shared memory
  - Parallel processes or threads share same data space
- Distributed memory
  - Parallel processes each have dedicated memory and communicate via message passing.



Parallel Computing

Shared Memory Parallelization

- Same memory usage as the serial run
- Multiple processes use the same memory and I/O
  - Synchronization of tasks is the key for implementation
  - Deadlocks and memory overwrites must be avoided!
- Scalability is determined by the hardware
- Popular Open SMP protocol: OpenMP



Parallel Computing

Distributed Memory Parallelization

- Each process has its own dedicated memory
  - Possibility of both duplication and/or splitting of memory use, depending on application
- Inter-process communication usually required
  - No synchronization required for memory access
- Scalability is determined by the algorithm being parallelized as well as the communication speed
- Popular DMP protocol: MPI (Message Passing Interface)

Parallel Computing

MAYA has begun parallelizing its solvers using the *Distributed Memory* paradigm

- The DMP approach accommodates user's existing hardware
  - With DMP, parallelization is achievable with multicore, multi-processor, network, and ***cluster*** architectures; SMP requires multicore or multi-CPU boxes (excludes networks and clusters)
  - All users with a network could in principle use DMP today; not so with SMP
- DMP scalability not as limited by available hardware
  - With SMP, if the best machine available is a quadcore processor, no more than 4 processors can be used
  - Given a scalable algorithm and a good network or hub, more than 4 processors can easily be brought to bear on a solve
- DMP is more cost effective to implement in existing code
  - SMP often requires paradigm shift & re-architecture, DMP not as much

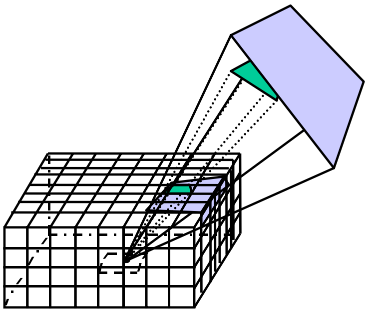
DMP Parallelization of the Hemicube Method

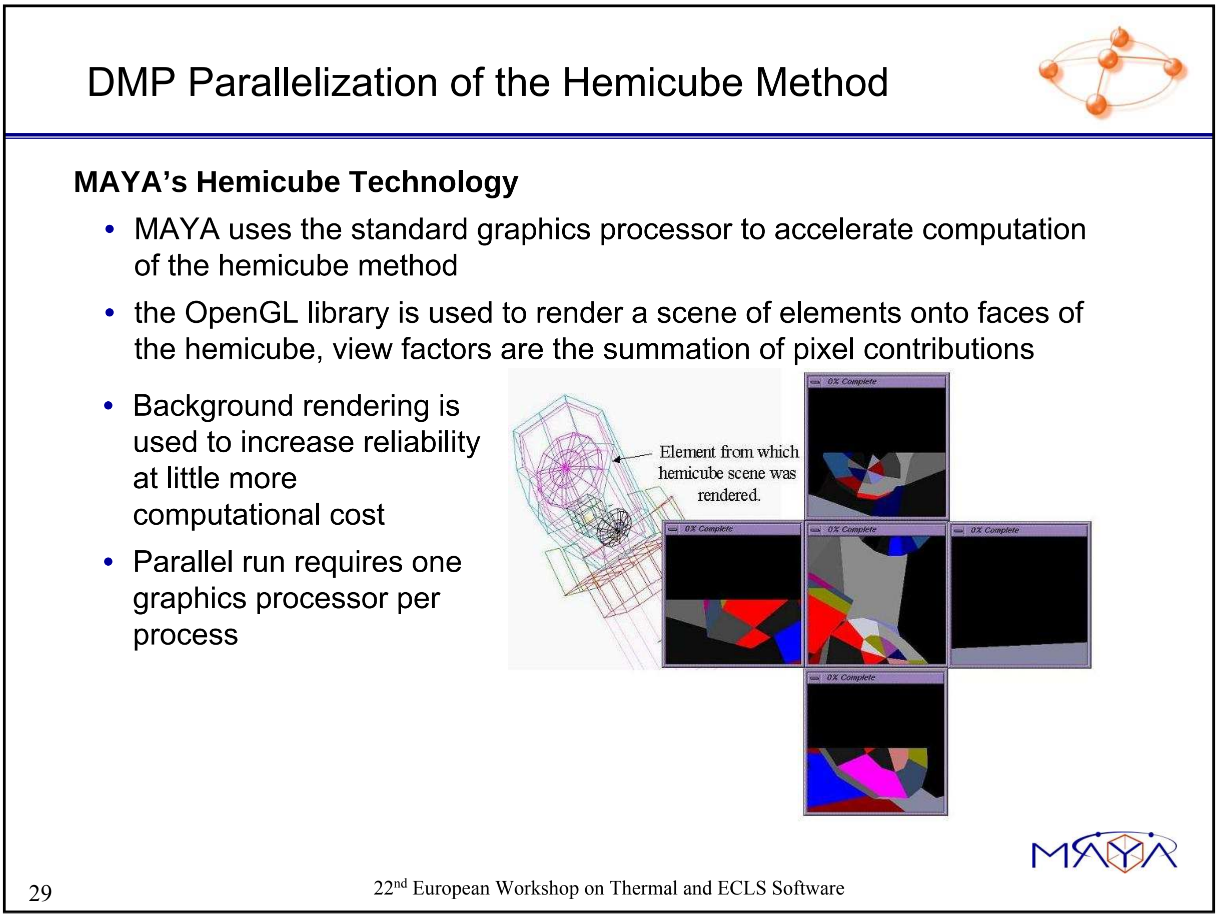
Parallelization of View Factor Computation

- View factor algorithms are *inherently parallel*, because view factors do not depend on one another
- Each process holds the model of the entire radiation environment, which independently computes a subset of the view factors

Hemicube Method: TMG *Hemiview* module

- variant of the Nusselt sphere method
- each face of the cube is divided into pixels: each pixel has a known view factor contribution
- hemicube is centered on a receiver element and the image of surrounding "emitter" elements are projected onto the hemicube
- view factors are tallied through pixel contributions



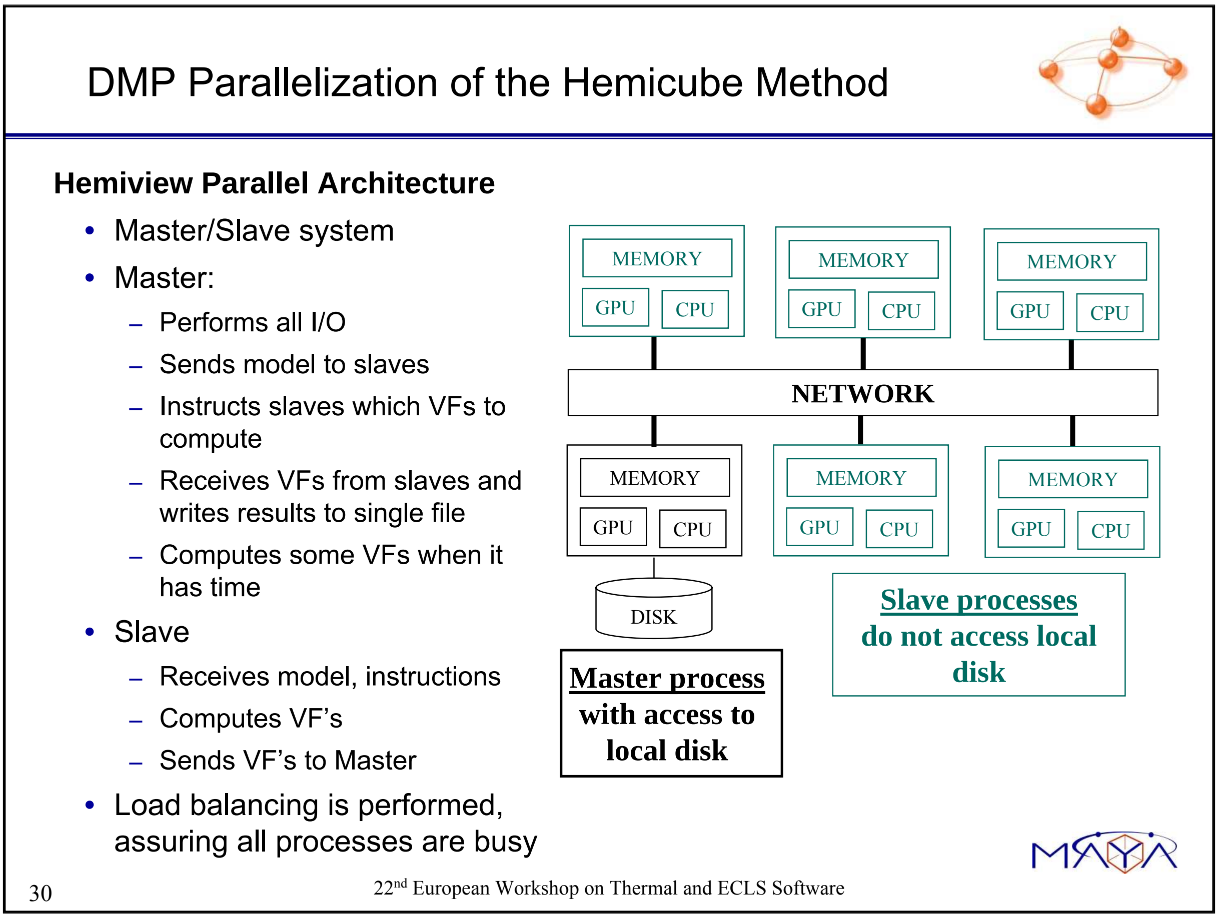


DMP Parallelization of the Hemicube Method

MAYA’s Hemicube Technology

- MAYA uses the standard graphics processor to accelerate computation of the hemicube method
- the OpenGL library is used to render a scene of elements onto faces of the hemicube, view factors are the summation of pixel contributions
- Background rendering is used to increase reliability at little more computational cost
- Parallel run requires one graphics processor per process

Element from which hemicube scene was rendered.



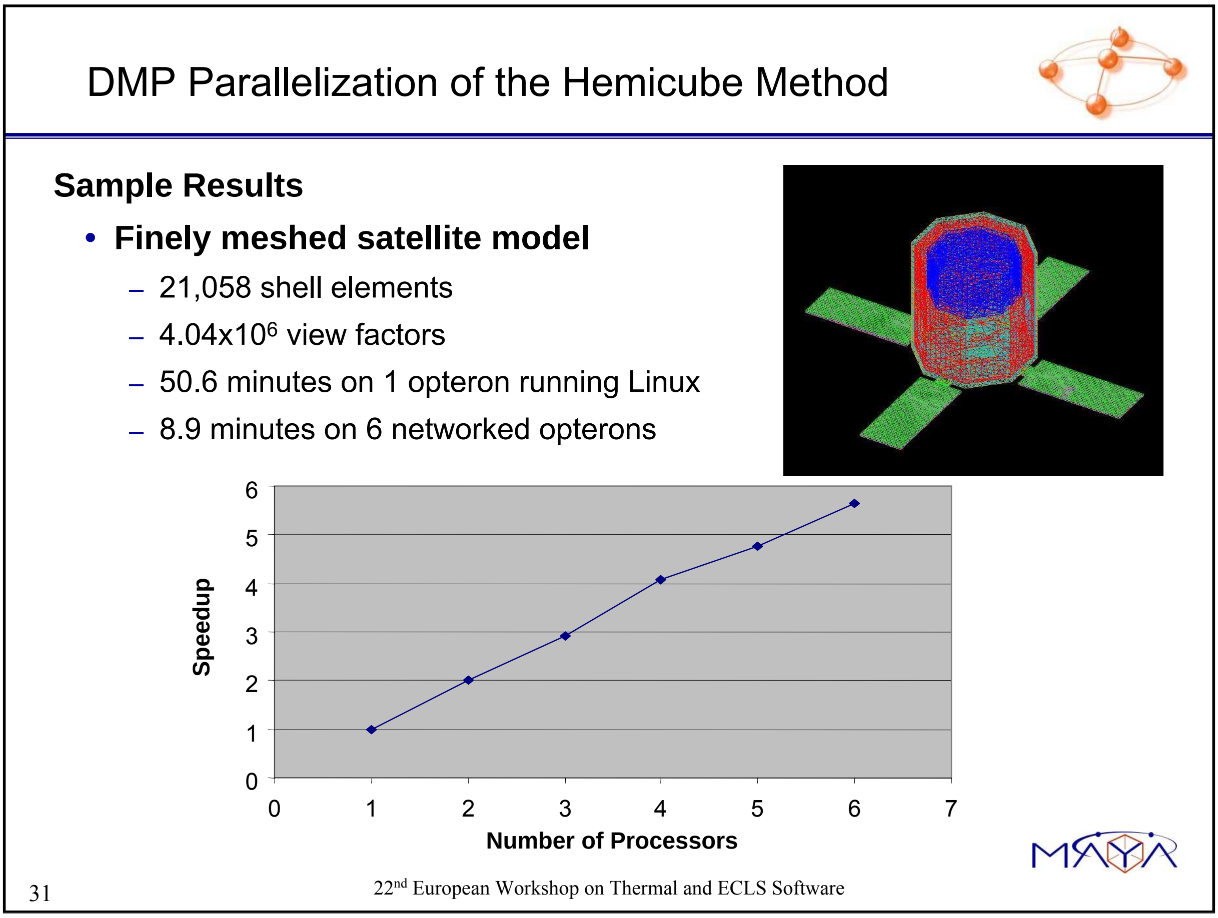
DMP Parallelization of the Hemicube Method

Hemiview Parallel Architecture

- Master/Slave system
- Master:
  - Performs all I/O
  - Sends model to slaves
  - Instructs slaves which VFs to compute
  - Receives VFs from slaves and writes results to single file
  - Computes some VFs when it has time
- Slave
  - Receives model, instructions
  - Computes VF’s
  - Sends VF’s to Master
- Load balancing is performed, assuring all processes are busy

Slave processes do not access local disk

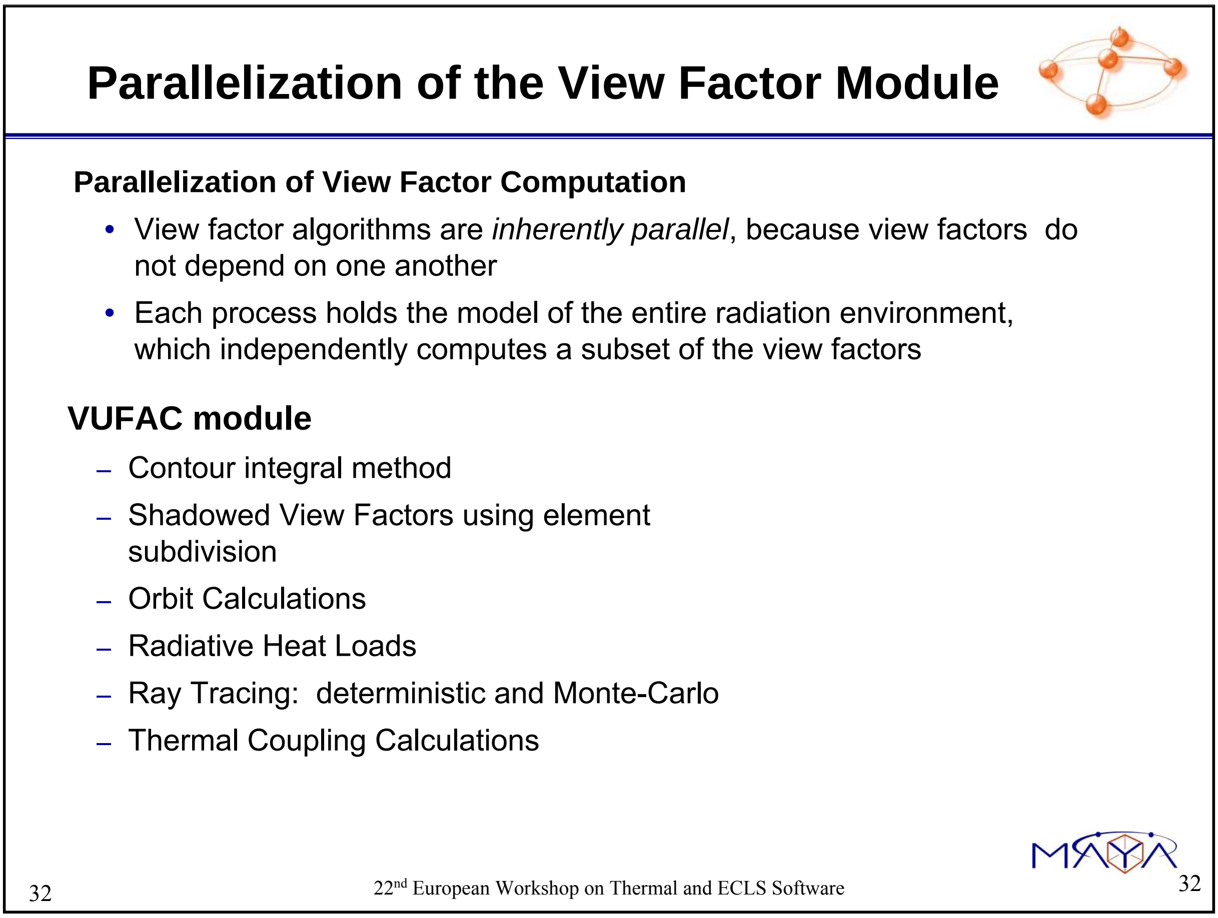
Master process with access to local disk



DMP Parallelization of the Hemicube Method

Sample Results

- **Finely meshed satellite model**
  - 21,058 shell elements
  - 4.04x10^6 view factors
  - 50.6 minutes on 1 opteron running Linux
  - 8.9 minutes on 6 networked opterons



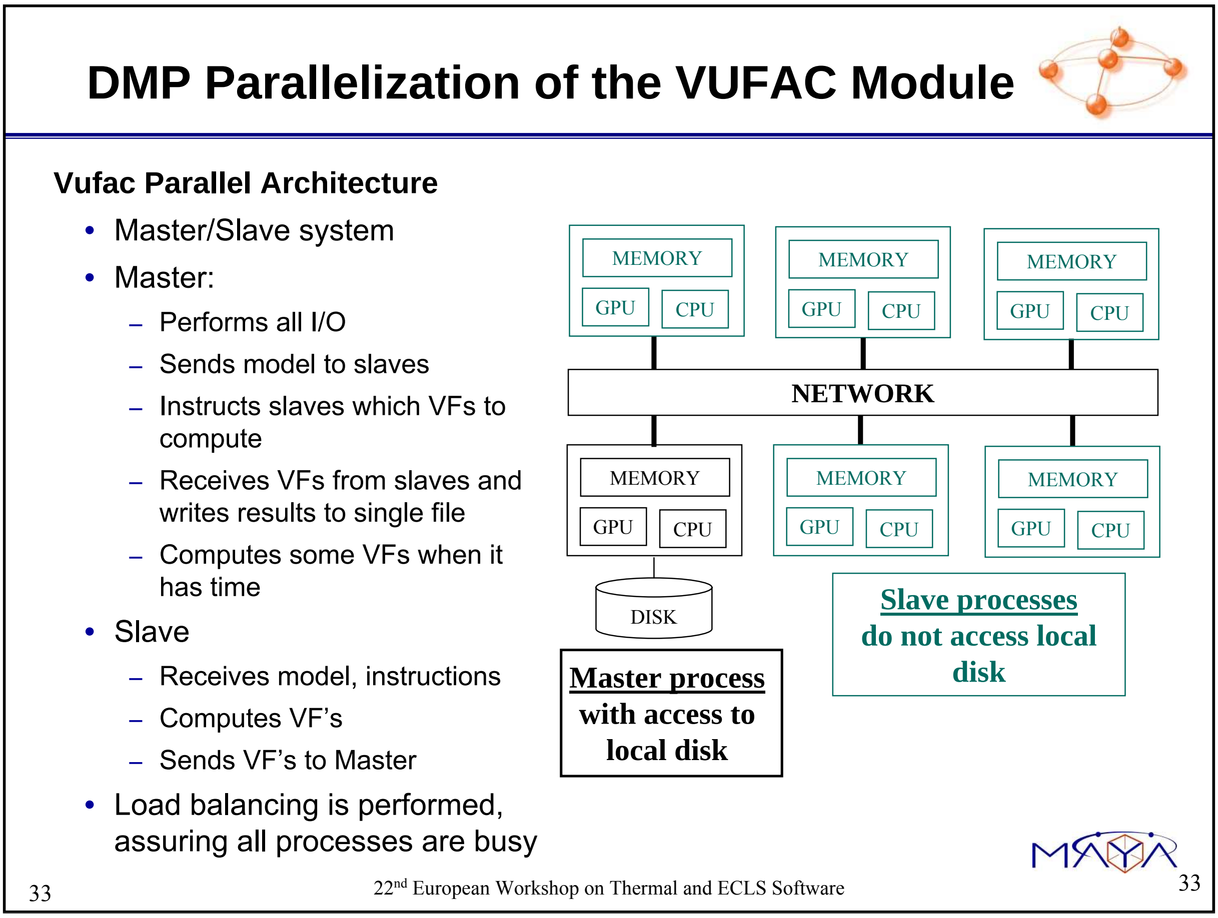
Parallelization of the View Factor Module

Parallelization of View Factor Computation

- View factor algorithms are *inherently parallel*, because view factors do not depend on one another
- Each process holds the model of the entire radiation environment, which independently computes a subset of the view factors

VUFAC module

- Contour integral method
- Shadowed View Factors using element subdivision
- Orbit Calculations
- Radiative Heat Loads
- Ray Tracing: deterministic and Monte-Carlo
- Thermal Coupling Calculations



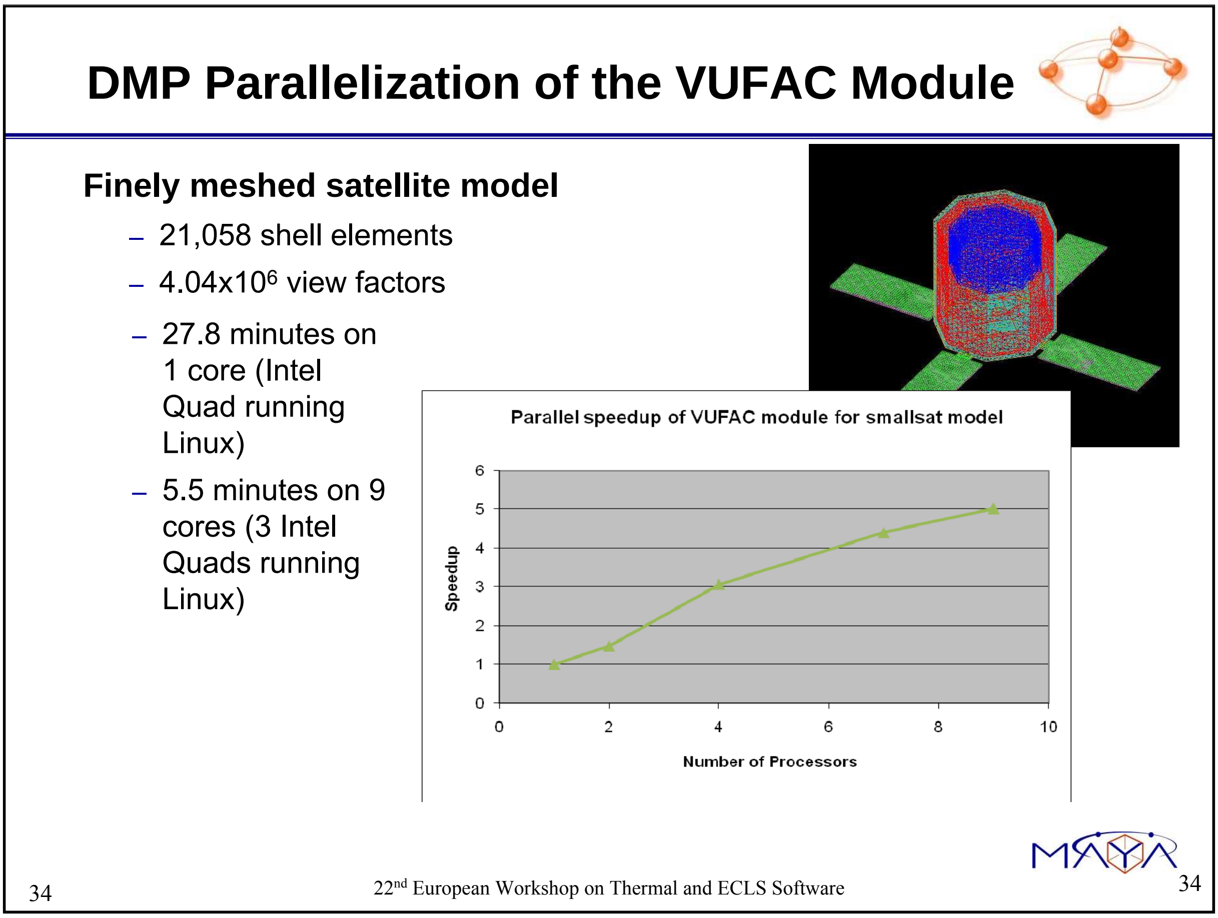
DMP Parallelization of the VUFAC Module

Vufac Parallel Architecture

- Master/Slave system
- Master:
  - Performs all I/O
  - Sends model to slaves
  - Instructs slaves which VFs to compute
  - Receives VFs from slaves and writes results to single file
  - Computes some VFs when it has time
- Slave
  - Receives model, instructions
  - Computes VF's
  - Sends VF's to Master
- Load balancing is performed, assuring all processes are busy

Master process with access to local disk

Slave processes do not access local disk



DMP Parallelization of the VUFAC Module

Finely meshed satellite model

- 21,058 shell elements
- 4.04x10^6 view factors
- 27.8 minutes on 1 core (Intel Quad running Linux)
- 5.5 minutes on 9 cores (3 Intel Quads running Linux)

Parallel speedup of VUFAC module for smallsat model

DMP Parallelization of View Factor Calculations

- Technology already commercialized !
  - **NX Advanced Thermal**
  - **NX Space Systems Thermal**
- Requires installation of MPI on all machines
  - MPICH2 is open source library
- Only a single installation of NX Thermal is necessary
- Parallelization of solver is in progress

"What took about 7 days of CPU time on the single CPU system only took 2 days when running 4 processors (on two machines)...I almost cried."

User from NASA GSFC

Thank you