A fast and efficient adaptive parallel ray tracing based model for thermally coupled surface radiation in casting and heat treatment processes.

J Fainberg and W Schaefer
MAGMA Gießereitechnologie GmbH, Aachen, Germany
E-Mail: J.Fainberg@magmasoft.de

Abstract. A new algorithm for heat exchange between thermally coupled diffusely radiating interfaces is presented, which can be applied for closed and half open transparent radiating cavities. Interfaces between opaque and transparent materials are automatically detected and subdivided into elementary radiation surfaces named tiles. Contrary to the classical view factor method, the fixed unit sphere area subdivision oriented along the normal tile direction is projected onto the surrounding radiation mesh and not vice versa. Then, the total incident radiating flux of the receiver is approximated as a direct sum of radiation intensities of representative “senders” with the same weight factor. A hierarchical scheme for the space angle subdivision is selected in order to minimize the total memory and the computational demands during thermal calculations. Direct visibility is tested by means of a voxel-based ray tracing method accelerated by means of the anisotropic Chebyshev distance method, which reuses the computational grid as a Chebyshev one. The ray tracing algorithm is fully parallelized using MPI and takes advantage of the balanced distribution of all available tiles among all CPU's. This approach allows tracing of each particular ray without any communication. The algorithm has been implemented in a commercial casting process simulation software. The accuracy and computational performance of the new radiation model for heat treatment, investment and ingot casting applications is illustrated using industrial examples.

1. Introduction
Surface-to-surface radiation is an important heat transfer mechanism in many metal casting and heat treatment processes. It needs to be considered in physical process models in order to accurately describe the heat exchange between different surfaces of the mold/casting or between the casting setup and surrounding equipment. The numerical modelling of the radiative heat transfer in a very complex environment is a computationally expensive task due to the long range effects and the multidimensional nature of radiation. Various numerical methods can be applied for calculating thermally coupled radiation.

The type of computational model depends on the volume and radiating surface properties to be described. The most comprehensive commercially established approaches are based on different variants of the discrete ordinates model (DOM) [1, 2], where combinations of the diffusive and specular surface properties, simultaneously with volumetric emission, absorption and scattering, can be considered.

The case of grey, opaque and diffusely radiating surfaces is the most typical one for radiation
processes occurring in daily foundry praxis. For applications without absorption, emission and scattering in a volume, simpler models can be applied. Usually, the radiosity method, with the calculation of view factors as a prerequisite, is applied \cite{3, 4, 5}. Here, a net radiative flux has to be computed for the thermal balance at the radiating surface. The incident absorbed radiation flux is summed via view factors computed for pairs of radiating surfaces with direct visibility. The straightforward method for computation of view factors and thermal coupling can lead to an enormous computational effort for the complex geometries typical for industrial applications. Therefore, various accelerations based on GPU hardware and sophisticated software were recently developed. The model presented here is solely a software-based solution using the message passing protocol (MPI) parallelization technique.

Different measures for acceleration of the computational radiation model are described in detail in the paper. First, an improved space angle subdivision is introduced. The organization of the ray tracing procedure is important as it has a strong impact on the performance of the radiation model preprocessing and is described next. Finally, the method of parallelization of ray shooting is reported. The achieved computational performance is then demonstrated by means of typical foundry applications.

2. Radiating surfaces
The computational work involved in radiation modelling is linearly proportional to the number of radiation tiles used. Computational time can be saved if the fineness of the used grid surpasses the required resolution in the distribution of the integrated heat flux (=irradiance) along the radiating surface. In this case, it would be sufficient to have fewer, coarser tiles than contained in the numerical grid. The reduction can be achieved by clustering of neighboring tiles.

Here, the numerical grid currently used in the commercial casting process simulation software MAGMASOFT\textsuperscript{®} was utilized directly for discretization of the radiant surfaces. The three-dimensional grid of MAGMASOFT\textsuperscript{®} is made up of 3 systems of grid lines in the cartesian spatial directions, which run through the entire model and embed it in a cuboid. The individual radiant surfaces in the model are initialized by assigning as a tile a right-angled face of a grid cell with the normal vector in one of the 6 coordinate directions, if the cell face divides an opaque and a transparent material. The clustering of neighboring faces is enabled in the same plane for grid cells of the same material. Up to 3 clustered faces are permitted in each of the two lateral directions; hence a maximum of 9 faces can be joined into a tile. The list of grid cells belonging to a clustered radiation tile is assembled and stored for each radiation tile. The ray tracing starts for each composed tile from its geometric center. Radiation sources are stored for the entire tile rather than for each component. The emitted heat flux of the composed tile is calculated accounting for cell temperatures within the cluster. As a result, acceleration by a factor of 2 or slightly more is achieved on average through tile clustering in the case of complex geometries.

3. Space angle by subdivision
The spatial angle has to be discretized in order to evaluate the incident radiation flux as a sum over discrete portions of the total space angle. This sum approximates the true integral over the space angle by a numerical quadrature over a finite number of angle subdivisions. The easiest manner of subdivision is to draw the unit hemisphere around the center of the radiating surface and to project the radiating grid onto it. The individual view factor for each pair of interacting elementary radiating surfaces can be interpreted directly from the projected area on the unit hemisphere. The number of view factors to be computed and stored varies approximately quadratically with the available radiating surfaces, leading to very high computational demand in industrial applications.

Inverting the projection direction can also be used for the space angle discretization. This means that the unit sphere is subdivided in a fixed manner in advance. This subdivision is then projected onto the radiating mesh in order to map the space angle subdivision to the corresponding radiation sources, and not vice versa as in the direct view factors method. The computational and memory requirements then scale linearly with the number of surfaces.
A hemi-cube projection is the best known technique for view factor calculations of this type [6, 7]. The space angle subdivision is achieved by subdivision of the facets of the unit cube surface into a finite number of rectangles. Then, the corresponding radiation source is looked for in the direction prescribed by the middle point of the rectangle placed at the central tile position. Finally, the full subdivided hemi-cube surface is projected onto the radiating mesh with radiating sources. This method is widely used in current commercial surface radiation models.

New in this procedure is that each subdivision of the solid angle corresponds to the same view factor \( VF \), namely \( VF_1 = 1/N = \text{const} \), where \( N \) is the total number of subdivisions. A homogeneous subdivision of the unit sphere of this kind in relation to view factors is not a homogeneous subdivision of its surface on account of the term \( \int \frac{\Omega \phi}{N} \) in the corresponding integral:

\[
\int \left( \frac{\Omega \phi}{N} \right) \partial \Omega = \int \int \int \int \sin(\theta) \cos(\theta) \partial \phi \partial \theta = \frac{\pi}{N} = \text{const}
\]

The advantage of a subdivision of this kind is seen when calculating the incident heat flux. A view factor which remains constant can be excluded, since determination of the incident flux changes into an averaging of the outgoing flux.

\[
q_{in} = \int \left( \frac{\Omega \phi}{N} \right) q_{out} \partial \Omega, \quad q_{in} = \frac{1}{N} \sum_{j=1}^{N} q_{out}
\]

A uniform distribution of the unit sphere according to the view factors is not unambiguous and can take place in innumerable ways. The method chosen here additionally requires certain symmetry in subdivision in relation to the normal vector of the radiation tile and is easy to handle.

**Figure 1.** An illustration of the chosen principle for fixed subdivision of the unit sphere, the view “from above” along the normal direction of the radiation tile, where the subdivision resembles a target and comprises the azimuthally arranged segments, which in turn form radial rings.

The subdivision of the unit sphere starts from a circle at the north pole and is followed by a series of radially concentric rings. Each ring is then subdivided into a different number of ring segments in the azimuthal direction, as shown in Figure 1. The number of subdivisions in individual rings forms an arithmetical progression, which limits the aspect ratio of the segments.

The subdivision is parameterized by the number of rings \( n \) in the meridian direction and the number of azimuthal segments of the first ring around the north pole \( r \). For each possible parameterization \((n, r)\) there is a single solution for the meridian ring coordinates, so that each spherical segment defines the same view factor:

\[
\theta = \frac{1}{2} \arccos \left( 1 - \frac{(n-i+1)(2r+n+i-2)}{N} \right), \quad \text{with} \quad N = \left( \frac{2r+n-1}{2} \right) n+1
\]

\( N \) in equation (3) is the total number of subdivisions according to the arithmetical progression. The number of subdivisions \( N \) and therefore also the number of rays per one tile grows quadratically according to equation (3) with the number of defined meridian rings \( n \).

The segments in a ring as shown in Figure 1 may be turned in the azimuth direction about an arbitrary free angle, so that a greater angle distance is created between segments of the adjacent azimuthal rings. Finally the system of unit vectors from the template shown in Figure 2 is rotated to
match the unit vector of the north pole with the normal vector of the tile.

Figure 2. An example of the system of direction vectors for parameterization \((r=4, n=10)\) resulting from the equation (3), where the total number of directions according to (3) is equal to \(N=86\).

3.1. Hierarchical system
An excessively rough subdivision of the unit sphere leads to numerical errors in the integration of the heat flux. In order to achieve a higher angle resolution in relation to the geometry of the radiating grid, this method uses a hierarchical system of discretization levels and is similar to the various refinement stages of the numerical grid in an explicit multi-grid method.

The first level is represented by the system of spatial directions produced according to equation (3), see Figure 2. The next level is produced by subdividing each segment of the first level by halving in the azimuth and meridian direction, i.e. quartering. One exception is the first refinement of the circular region at the north pole into 4 spherical triangles. Further recursive subdivisions repeat the same procedure, and the 4 spherical triangles around the north pole are treated as abnormal spherical rectangles from the third refinement level.

This produces a geometric progression of the number of rays at each successive level. For \(k\) levels of refinement

\[
N_{\text{tot}} = N \frac{4^k - 1}{3} = \left(\frac{2r + n - 1}{n + 1}\right) \left(\frac{4^k - 1}{3}\right)
\]

rays result overall on all refinement levels. The directional vectors according to the refinement are shown in Figure 3 A, B and C.

Figure 3. A: direction vectors from second refinement level created from the first level with subdivision \((r, n) = (4, 10)\); B: third level created from the second level; C: direction vectors from all three levels, wherein vectors from different levels are characterized using different symbols at the origin of the vectors. The rays for all directions shown are sent out according to the ray tracing method.

Three refinement levels are used in the current implementation. In the hierarchical method of ray tracing, rays are initially sent out in succession from all refinement levels as shown in Figure 3C, and corresponding radiation sources are stored.

3.2. Adaptation to the geometry and temperature distribution
It would be possible to only send rays for the vectors of a single or the finest level. However, if the rays are sent by all defined refinement levels, the possibility exists for the angle resolution to be adapted locally.
The result of ray tracing for the thermal calculation without adjustment would be a complete list of radiation sources, which all result e.g. from one level of subdivision. The local adaptation of the angle discretization replaces a list of this kind with sub-lists from all discretization levels. The approach in this case is as follows:

Starting point is a discretization that contains only vectors of the most refined level and proceeds upward to the next coarser level. Each time, the radiation sources of the preceding finer level are checked. If any of the 4 quarters belongs to a different material, if the initial temperatures of the found materials differ from each other by more than a predefined threshold, or there exist both external sources and sources originated from other tiles, the radiation source of the finer level is retained. Otherwise, the 4 quarters are replaced by a single radiation source from the coarser spherical rectangle.

The procedure described above supplies $k=3$ lists of radiation sources for 3 levels of uniform space angle discretization. The adapted angle discretization in this case is chosen in such a manner that a concentration of representative radiation sources occurs in the angle regions where jumps of the radiation intensity with respect to the spatial direction can potentially be expected. The adapted discretization of the solid angle for the radiation tile takes place in this way through the selection of radiation sources.

One example of the adapted discretization of the solid angle is shown in Figure 4.

![Figure 4](image)

**Figure 4.** A view factor of an “H”-shaped face is approximated by means of an adaptive space angle subdivision. For this purpose, 3 discretization levels according to equation (3) were used, with the first level parameterized with $(n=15,r=4)$. The number of rays is equal to 166, 664 and 2656 at the 3 discretization levels. A total of 3486 rays are sent out to scan the hemisphere. After adaptation has been carried out, 218 representative rays have been selected, which is smaller by a factor of 16 than at the finest level. The selected 218 spherical rectangles from different levels are marked in the figure.

The outgoing heat fluxes, connected to the radiation sources from different levels, must be weighted differently in the calculation of the incident heat flux. Equation (5) for the incident heat flux is supplemented by pre-exponential factors of different levels. It describes the calculation of the incident radiation flux for an arbitrary number of hierarchical levels $k$.

$$q_{in}^{j} = \frac{1}{N} \sum_{level}^{k} \left( \frac{1}{4^{k-1}} \sum_{j=1}^{N} q_{out}^{i} \right) + q_{ext}$$

The last term in equation (5) is the incident radiation flux from the space angle spanned on the external ambient environment in the case of half open cavities.

According to equation (5), just the outgoing total radiation fluxes of tiles have to be summed in order to obtain the total incident flux for the considered tile. The weight factors in front of the inner sum mimic the subsequent recursive subdivision of the space angle. The advantage of the developed method is that values of individual view factors are not required explicitly according to equation (5).

**4. Ray tracing**

A voxel-based ray tracing method with reverse ray tracing is used. All rays are initially sent out from the tile that receives the radiation and the radiation sources are traced by rays according to geometrical optics.

**4.1. Voxels**

The rays are sent out from all refinement levels of the space angle discretization. Ray tracing involves
the search for intersections between the continuation of a ray and faces of a grid cell (voxel) in which the ray is currently located, see Figure 5.

A voxel is a cube-shaped volume where information on geometric objects is contained. In a cartesian grid, a tensor product grid is available where each grid cell becomes a voxel. Up to three tile ID’s referring to three faces on the “back”, “east” and “north” side of the grid cell are stored per grid cell to support ray tracing.

The three possible faces of the grid cell are checked for the next intersection with the traced ray. The faces are given by the sign of the three components of the ray direction \( \Omega \). The face of the cell with the minimum length up to the intersection is searched for, and the ray is continued as far as the located intersection by the located length \( \Delta X_{\text{ray}} \).

\[
X_{\text{MIN}} = \text{MIN} \left\{ x_i - x_i^{\text{ray}}, \Omega_i \right\}, \quad \Delta X_{\text{ray}} = X_{\text{MIN}} \Omega_i
\]

Once the next intersection has been found, the global tile ID is requested on the face of the cell, the ray tracing is stopped and the found radiation source is returned. In the case of a symmetry plane, the directional component normal to the symmetry plane is inverted and the reflected ray is traced further. Otherwise, the ray tracing procedure is repeated in the next grid cell until the ray hits a tile or leaves the bounding box of the numerical grid. In the latter case, a fixed ID of the outer space source is returned.

4.2. Chebyshev distance
The conventional ray tracing described above requires each transparent grid cell lying on the ray path to be visited, see Figure 5. The processing of empty cells is expensive for geometries with large, empty cavities and/or a fine grid there. Significant ray tracing acceleration can be achieved by ray “tunneling” over larger blocks of empty cells, but requires information about the size of empty cell blocks.

The so-called Chebyshev or chess board distance is utilized for such tunneling [8]. A two-dimensional example of Chebyshev distance usage is shown in Figure 6. The distance from the dotted grid cells is measured. Figure 6A shows the isotropic Chebyshev distance. Cells at the same distance from the dotted zero cells are arranged in a layer around the original square. The minimum distance for all Cartesian directions is used in the isotropic case. The anisotropic Chebyshev distance for the upper left quadrant is shown in Figure 6B.

Use of the Chebyshev distance allows adaptive steps in ray tracing. When the ray approaches a surface, the Chebyshev distance diminishes and with it the jump size, too. The closer a ray gets to the surface, the slower the ray tracing progress.

The anisotropic distance is measured from the zero distance cells, but only for Cartesian directions inside of the particular quadrant. It is evident from Figure 6 A and B that the anisotropic distance achieves greater values than the isotropic one. The anisotropic case allows jumps through larger empty grid blocks, as shown in Figure 6 C. In this example, 2 intersections were searched for; without using the Chebyshev distance it would be 9. In the three-dimensional case, the anisotropic Chebyshev distance is computed and stored in advance for 8 octants. For tracing of a ray in a particular direction \( \tilde{\Omega} \), the Chebyshev measure from the corresponding opposite octant is retrieved and used.
Despite the additional work and memory involved to support the grid jumps, the Chebyshev distance provides a substantial acceleration of the method, particularly for models with large empty / transparent spaces.

Figure 6. Calculation of the 2D Chebyshev distance. The directions observed are marked with the arrows in the zero cells in the respective cases. A: isotropic Chebyshev distance; B: anisotropic Chebyshev distance. C: an example of 2D ray tracing supported by anisotropic Chebyshev distance distribution from (B). The ray in the octant (+X,-Y) is processed assisted with distance in the octant (-X, +Y). The initial position of the ray in the intersection with a cell face at top left lies in the cell at distance 3. The next intersection with the 3x3 cube is therefore searched. Following displacement of the position to the intersection, the Chebyshev distance is taken from the next cell in the direction of the ray path. It is once again equal to 3, and the next jump leads the ray to the target at bottom right.

4.3. Parallel ray tracing
Prior to ray tracing, calculation of the anisotropic Chebyshev distance takes place on each CPU independently. Calculations and results on each CPU are identical at this point. Each CPU obtains the material distribution over the total numerical grid as the only input.

Before ray tracing starts, the number of tiles to be processed on each CPU is balanced. For this, the number of tiles located at a specific MPI partition is compared with the mean arithmetic tile number among all CPU’s. The CPU’s with an excessive number of tiles are treated as the tile donors, the rest as the tile acceptors. Further, the tile’s data is sent from the donor to the acceptor CPU’s in order to obtain a balanced tile distribution. Finally, each CPU has to process some number of own not exported and some number of imported ray bundles associated with each tile.

All rays on each CPU are dispatched independently of one another as described in Section 3.1. No MPI communication during tracing of a particular ray is required, since each CPU operates on the identical full numerical grid collected from all MPI partitions.

Radiation sources of own tiles are found, selected by means of the geometric and thermal adaptation as described in Section 2.2 and stored. Imported tiles on the CPU acceptors are processed next. The radiation sources, selected according to the adaptation procedure, are temporarily stored in a buffer.

When all tiles have been processed, the results for all imported tiles are sent backwards from the buffer by the CPU acceptors to the CPU donors. Each donor CPU may generally have several acceptors and each acceptor CPU may have several donors. Asynchronous MPI communication takes place.

5. Model applications
Three computed examples for heat treatment, investment and ingot casting are presented. They all were computed on a PC with 8 cores with default model settings. 5691 rays from 3 space angle refinement levels were traced for each tile. The used numerical grid, the numerical data of the radiation model and computational times are summarized in Table 1 below.

The results for a heat treatment of several castings in a furnace is shown in Figure 7, left, where 82,251,716 radiation sources were stored totally. A sample simulation of an investment casting is
shown in Figure 7, right. A large number of radiation tiles (554,905) after agglomeration was required due to the curved, complex surfaces involved. The total number of stored radiation sources was 156,429,337.

**Table 1.** Statistics of three representative radiation model applications. Column descriptions: the total number of used grid cells, the total number of radiation tiles after their agglomeration, the total number of traced rays, the mean number of radiative sources after geometrical and thermal adaptation, the computational time spent for the model set-up (ray tracing), total computational time with set-up for the transient thermal modelling and the corresponding time without radiation.

| Model Type          | Grid size (cells) | Tiles       | Total Rays     | Sources /Tile | Set-up Time | Total Time | Without Radiation |
|---------------------|-------------------|-------------|----------------|---------------|-------------|-------------|-------------------|
| Heat Treatment      | 9,755,200         | 237,702     | 1,352,762,082  | 346           | 147 sec.    | 516 sec.    | 53 sec.           |
| Investment Casting  | 45,006,208        | 554,905     | 3,157,964,355  | 281           | 365 sec.    | 35 min.     | 21 min.           |
| Ingot Casting       | 1,743,000         | 29,216      | 166,268,256    | 289           | 12.4 sec.   | 55 sec.     | 44 sec.           |

**Figure 7.** Left: example of surface-to-surface radiation model application, heat treatment of 5 cast pump housings in a furnace. Right: a thermal result for an investment casting project calculated using the radiation model.

A sample simulation from steel ingot production is shown in Figure 8. A symmetry boundary condition is applied for radiation and the thermal model, hence a total of 6 ingot blocks were effectively simulated. On average 2.4 radiating cell faces were joined to a single radiation tile after their agglomeration. The average number of radiation sources for the tile after thermal and geometrical adaptation was 289 – roughly 20 times less than at the finest level. The total number of stored radiation sources was 8,456,505. The parallel ray tracing performance was over 13 million traced rays per second.
Figure 8. An example from steel ingot casting. The radiation model was applied to simulation of pouring and solidification of multiple ingots. Left: temperature at the ingots' surface during solidification. Right: computed net radiation flux density at the radiating surfaces.

The computational times from table 1 were compared with reported performance for several state of the art methods reported in [9]. The computational times for the quickest reported method View3D-O for 6 different numbers of radiation tiles are extrapolated by means of the power law for the number of tiles obtained in all three of the benchmark cases in table 1. The reported and extrapolated results are plotted together with the results of the current method in figure 9. A huge acceleration of 477, 1407 and 2593, respectively, is obtained with respect to the View3D-O method for the benchmark cases of ingot casting, heat treatment and investment casting.

Figure 9. Computational efficiency compared with evaluation in [9]. Round symbols are times of the quickest method View3D-O, see table 5 in [9]. The symbols for number of tiles less than 10,000 are from [9]. Round symbols for three larger values are extrapolated times for the benchmark cases in Table 1. The extrapolation was done using the power law approximation time~N^{1.7238}. The result for the actual method in Table 1 are shown with rhomb symbols.

6. Summary

A high computational efficiency in a surface-to-surface radiation model was achieved by a complex optimization especially of the radiation model set-up. The particular computational time reductions were obtained by

- agglomeration of elementary radiating surfaces,
- adaptation of the space angle discretization by means of a hierarchical system,
- application of voxel based ray tracing with a direct utilization of a Cartesian grid,
- ray tracing “tunneling” method supported by the anisotropic Chebyshev distance,
- organization of the parallel ray tracing without any need for communication of the total grid among all CPU’s.

The developed model allows robust full-scale thermal simulations including thermally coupled radiative heat exchange in very complex three-dimensional geometries using conventional multi-core hardware. The computational cost of such simulations is not significantly higher than the reference case without radiation between surfaces.

References

[1] Modest M F 1993 *Radiative heat transfer* (McGraw-Hill series) ch 15 pp 541-71, ch 20 pp 703-58
[2] Siegel R and Howell J 2002 *Thermal radiation heat transfer* (Taylor & France 4th edition) ch 16 pp 663-706
[3] Siegel R and Howell J 2002 *Thermal radiation heat transfer* (Taylor & France 4th edition) ch 6 pp 207-66
[4] Modest M F 1993 *Radiative heat transfer* (McGraw-Hill series) ch 4 pp 155-92
[5] Fluent 6.3 user’s guide (Fluent Inc) ch 13.3
[6] Class M *CHAPARRAL 1995 - a library package for solving large enclosure radiation heat transfer problems* (Sandia National Laboratories, Albuquerque NM)
[7] Cohen M and Greenberg D 1985 *The hemi-cube: a radiosity solution for complex environments* Comp. Graphics 19 No 3 pp 31-40

[8] Abello J, Pardalos P and Mauricio G 2002 *Handbook of massive data sets* (Springer 2002)

[9] Walton G N 2002 *Calculation of obstructed view factors by adaptive integration* (NISTIR 6926)