Numerical Heat Transfer, Part B, 35:137–161, 1999

1040–7790/99 $12.00 + .00

PARALLELIZATION OF THE DISCRETE TRANSFER METHOD

P. J. Novo, P. J. Coelho, and M. G. Carvalho
Instituto Superior Técnico, Technical University of Lisbon, Mechanical Engineering Department, Av. Rovisco Pais, 1096 Lisboa Codex, Portugal

The discrete transfer method was parallelized and applied to the calculation of radiative heat transfer in 2D and 3D enclosures containing a gray emitting-absorbing and isotropically scattering medium. Two different parallel strategies were used, the ray domain decomposition and the spatial domain decomposition. Three test cases were considered, and the influence of the number of processors, angular and spatial discretizations, absorption coefficient of the medium and emissivity of the boundaries on the parallel performance was investigated. It is shown that the ray decomposition is the most efficient method, mainly because the convergence rate is not influenced by the number of processors.

INTRODUCTION

The discrete transfer method (DTM) [1] has been widely used in the calculation of radiative heat transfer in combustion chambers, especially when the simulation of the reactive fluid flow is alsc carried out [2–4]. The computational requirements of these calculations are quite high, and there is a general consensus about the benefits of using parallel computing to tackle these kinds of problems. Furthermore, parallelization techniques may be used together with other acceleration methods, such as the successive overrelaxation, synthetic acceleration, or mesh rebalance methods used in [5] in the framework of the discrete ordinates method, or the multigrid method. However, while the parallelization of fluid flow codes has been widely investigated, only limited attention has been given to the parallelization of radiation models [6], despite the impact that parallel computing may have in this field [7]. Nevertheless, several authors have already addressed the problem of parallelizing a radiative heat transfer method, such as the zone method [8], the Monte Carlo [9, 10], the discrete ordinates [11–16], and the finite-volume method [17].

As far as the DTM is concerned, besides a preliminary work presented by us [18], we are only aware of another paper addressing its parallelization [19]. But in [19] the attention was focused on the comparison of algorithms for the calculation of the radiative properties of the medium, and the application was restricted to a one-dimensional problem. In the present work an improved version of the code described in [18] was used and a much more detailed analysis of the parallel

Received 20 May 1998; accepted 11 September 1998.
The authors are indebted to the Council for the Central Laboratory of the Research Councils (CLRC) Daresbury Laboratory for permission to use their computational facilities.
Address correspondence to Prof. Pedro J. Coelho, Technical University of Lisbon, Instituto Superior Técnico, Mechanical Engineering Department, Av. Rovisco Pais, 1096 Lisboa Codex, Portugal. E-mail: coelho@vangogh.ist.utl.pt

NOMENCLATURE

E_a	difference between successive iterations of the absorbed radiative energy on the wall	S_p	speed-up
		t_{cm}	communication time
		t_e	execution time
E_i	difference between successive iterations of the incident radiative energy on the wall	t_p	execution time using p processors
		β	extinction coefficient
		δs	geometric path length within a control volume
E_p	efficiency		
G	incident radiation	ε	emissivity
H	irradiation onto a surface	κ	absorption coefficient
I	radiation intensity	σ	Stefan-Boltzmann constant
J	radiosity	σ_s	scattering coefficient
n_{iter}	number of iterations required to achieve convergence	τ	optical thickness
		ϕ	scattering phase function
n_p	number of iterations required to achieve convergence using p processors	ω	single scattering albedo
		Ω	solid angle
N_x, N_y	number of grid nodes along x and y directions	**Subscripts**	
N_θ, N_φ	number of polar and azimuthal angles per octant	b	blackbody value
		$n, n+1$	entry (n) into or exit ($n+1$) from a control volume
p	number of processors		
q	radiative heat flux	p	number of processors
s	geometric path length	P	point P
$\mathbf{s}$	unit vector along a given direction	w	wall

performance was carried out. The calculations were performed in two distributed memory computers, the Intel iPSC/860 and the IBM SP2, using up to 16 processors. Although a shared-memory computer would require fewer changes in the sequential code, massively parallel shared-memory computers are not presently available. Therefore, the applicability of the proposed parallelization methods would be limited if a shared-memory computer were used.

The parallelism in the solution of the radiative transfer equation (RTE) may be achieved using wavelength decomposition, angular or ray decomposition (RDP), or spatial domain decomposition (DDP). A combination of two or three decomposition methods is also feasible. In the case of gray media, as considered in the present work, only the last two options are available. The parallelization methods differ in the way the computational load is distributed among the processors. The RDP is the most straightforward approach, and it consists in splitting up the total number of radiation beams into a number of subsets equal to the number of processors. Every processor integrates the RTE along the directions of the radiation beams across the entire domain. In the DDP the domain is split up into subdomains, and each of them is assigned to one processor. All the radiation beams traveling in a subdomain are tracked by the processor assigned to that subdomain. Although the DDP does not fit the characteristics of the DTM as well as the RDP, it is a natural choice when the radiative heat transfer calculations are coupled to the simulation of a reactive fluid flow, since the spatial domain decomposition is the standard approach in computational fluid dynamics.

A short overview of the DTM is given in the next section to provide the background required for the detailed explanation of the parallel implementation which is presented next. Then, the results are presented and discussed. They include three test cases consisting of two- and three-dimensional enclosures with gray emitting-absorbing and isotropically scattering media. The influence of the number of processors, grid size, angular refinement, absorption coefficient of the medium, and emissivity of the walls on the parallel performance are investigated for both parallelization methods. Two different ray splitting strategies in the RDP are compared. Finally, the main conclusions are summarized.

THE SEQUENTIAL ALGORITHM AND THE PARALLELIZATION STRATEGIES

The Discrete Transfer Method

The main features of the DTM are described below in order to facilitate the discussion of the parallelization strategies. A complete description of the method is given elsewhere [1].

The DTM is based on the numerical solution of the RTE along specified directions. For a gray emitting-absorbing-scattering medium, as considered in this article, the RTE may be written as follows:

$$\frac{dI}{ds} = -\beta I + \kappa I_b + \frac{\sigma_s}{4\pi}\int_0^{4\pi} I(\mathbf{s}')\phi(\mathbf{s}', \mathbf{s})\, d\Omega' \tag{1}$$

Equation (1) is a statement of the principle of conservation of energy applied to a pencil of radiation traveling along direction $\mathbf{s}$. In this equation I is the radiation intensity, s is the coordinate measured along the direction $\mathbf{s}$, κ is the absorption coefficient of the medium, σ_s is the scattering coefficient, β is the extinction coefficient, and the subscript b refers to a blackbody. The ratio $\phi\ (\mathbf{s}', \mathbf{s})/4\pi$ represents the probability that radiation propagating in the direction $\mathbf{s}'$ and confined within the solid angle $d\Omega'$ is scattered through the angle $(\mathbf{s}', \mathbf{s})$ into the direction $\mathbf{s}$ confined within the solid angle $d\Omega$. The absorption, scattering and extinction coefficients are related to the single scattering albedo as follows:

$$\omega = \frac{\sigma_s}{\kappa + \sigma_s} = \frac{\sigma_s}{\beta} \tag{2}$$

In the DTM the physical domain is divided into control volumes. The temperature and the radiative properties of the medium are taken as constant in each of them. For all the control volumes adjacent to the boundary, the central points of the faces of the control volumes coincident to the boundary are determined. Let P be one such point. The hemisphere centered at P is discretized into a given number of solid angles. Each solid angle defines a direction along which the RTE is solved.

Hence, given a point P at the center of a cell face on the boundary, a radiation beam is fired from P for each of the directions selected above. The path

of a radiation beam is followed until it hits another boundary. Let Q_i be the impingement point. Although, in general, Q_i is not the central point of the boundary cell, it is assumed that the radiation intensity at Q_i and at the central point of the cell which contains Q_i are equal. This radiation intensity is either known from the boundary conditions or calculated based on the values of the previous iteration. Then, starting from Q_i, the path of the beam is followed back to the origin (point P), and the RTE is integrated analytically along this path, yielding [1]

$$I_{n+1} = I_n e^{-\beta\,\delta s} + \left[(1-\omega)I_b + \left(\frac{\omega}{4\pi}\right)\int_0^{4\pi} I(\mathbf{s}')\phi(\mathbf{s}',\mathbf{s})\,d\Omega'\right](1-e^{-\beta\,\delta s}) \quad (3)$$

In this equation the subscripts n and $n+1$ denote the points at the entry into or at the exit from a control volume, respectively, and δs is the distance between those two points. In the case of isotropic scattering, the integral in the in-scattering term is equal to the incident radiation:

$$\int_0^{4\pi} I(\mathbf{s}')\phi(\mathbf{s}',\mathbf{s})\,d\Omega' = G \quad (4)$$

The incident radiation is related to the radiation source (or sink) in a control volume by means of the conservation of radiative energy equation:

$$\nabla \cdot q = \kappa(4\sigma T^4 - G) \quad (5)$$

The radiation source (or sink) in a control volume is the integral of the divergence of the radiative heat flux over the control volume. It is obtained from the sum of the contributions of all the radiation beams which cross that control volume. This source may be calculated only after a converged solution has been achieved, provided that the temperature of the medium is prescribed and there is no scattering. If a radiative heat source rather than the temperature field is prescribed, the temperature field must be determined from the simultaneous solution of the energy conservation equation and the RTE. This may be accomplished as described in [20].

The incident radiative flux at point P, that is, the irradiation, H_P, is calculated by adding the contributions due to all the radiation beams that reach point P. The solution of the RTE requires the specification of the boundary conditions. If the wall temperature, T_w, is prescribed, the boundary condition for gray diffuse boundary surfaces may be written as follows:

$$J_P = \varepsilon_w \sigma T_w^4 + (1-\varepsilon_w)H_P \quad (6)$$

where J_P is the radiosity at point P, ε_w is the emissivity of the wall, and σ is the Stefan-Boltzmann constant. Other boundary conditions may be treated as described in [1].

The calculations described above are performed sequentially for all the cell faces on the boundary of the computational domain. The method is iterative,

unless $\varepsilon_w = 1$, because the radiation intensities leaving the points Q_i are not known a priori. They are guessed in the first iteration, and then computed from the values of the previous iteration, as stated above.

The convergence criterion used in this work demands that the difference between two successive iterations of the total radiative energy absorbed (or incident) on the boundary be smaller than a specified tolerance. If a radiative heat source is prescribed, in addition to the previous criterion, it is also required that the sum over all the control volumes of the difference (or the maximum difference) between the prescribed and the computed heat source decreases below a specified tolerance. Moreover, the sum over all the control volumes (or the maximum) temperature difference between successive iterations must be smaller than a specified tolerance.

Parallelization of the Discrete Transfer Method

Ray decomposition parallelization (RDP). In the RDP, the total number of radiation beams fired from the boundary of the domain is split into a number of subsets equal to the number of processors, and each subset is assigned to a processor. Each processor performs the calculations for the whole domain but deals only with its subset of radiation beams. This approach may be implemented in different ways depending on how the subsets of radiation beams are selected.

The strategy employed in the present work consists of the division of every boundary of the enclosure into a number of subregions equal to the number of processors, each subregion being assigned to a different processor. Two different methods were used to define these subregions. In method 1 the boundary is divided into equal subregions. If the division is performed along the x direction, denoted by N_x the number of grid nodes along that direction and by p the total number of processors, and assuming that N_x is a multiple of p, then the ith processor treats all the radiation beams fired from the boundary cell $N_x * (i - 1)/p + 1$ to the boundary cell $N_x * i/p$. All the radiation beams fired from the boundary cells in a subregion are treated by the processor assigned to that subregion. This method is illustrated in Figure 1*a* for a 2D boundary (8×8 boundary cells) of a 3D problem and assuming that 4 processors are used.

Method 1 may yield load imbalance among the processors, as will be shown in the results section of this article. It is worth emphasizing that this imbalance is due only to the way the radiation beams are split among the processors, and it is independent of the machine used. Another method was implemented in an attempt to reduce the load imbalance. In this second method the subregions are defined according to a chessboardlike pattern. The radiation beams fired from neighboring cells are assigned sequentially to different processors. Figure 1*b* illustrates this method for the example considered above.

The restriction of methods 1 and 2 to a 2D problem mapped using a uniform grid with 8×8 control volumes and 4 processors is shown in Figure 2. The path of all the radiation beams treated by processor 1 are drawn in this figure for the case $N_\theta = N_\varphi = 1$, N_θ and N_φ standing for the number of discrete polar and azimuthal angles per octant, respectively.

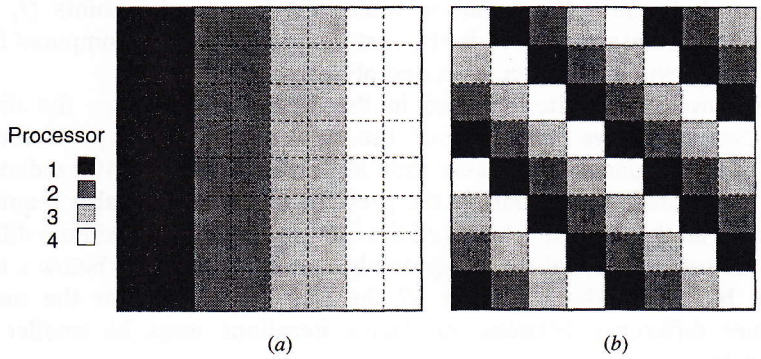


Figure 1. Splitting of the radiation beams among the processors in the ray decomposition parallelization method in the case of a 2D boundary (8×8 boundary cells) of a 3D problem and for 4 processors: (*a*) method 1; (*b*) method 2.

It has been assumed, for simplicity reasons, that the number of cells per boundary is a multiple of the number of processors. If this is not the case, there is a straightforward extension. In fact, it would be easy to split up the total number of radiation beams among the various processors regardless of the boundary cells. Since a total of $N_{x,y} * (4 * N_\theta * N_\varphi)$ radiation beams are fired from each one of the four boundaries in 2D problems, $N_{x,y}$ being N_x or N_y depending on the boundary, then each processor would deal with $N_{x,y} * (4 * N_\theta * N_\varphi)/p$ radiation beams.

In distributed memory platforms each processor has access to its own memory, but the access to a nonlocal memory requires message passing via interconnection networks. The communication among processors in the RDP is

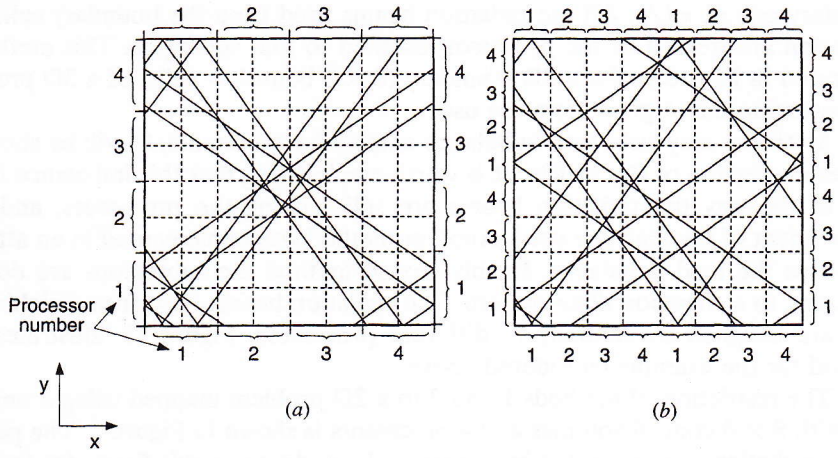


Figure 2. Path of all the radiation beams fired from processor 1 in the ray decomposition parallelization method in the case of a 2D problem (8×8 control volumes), $N_\theta = N_\varphi = 1$ and 4 processors: (*a*) method 1; (*b*) method 2.

restricted to global operations, e.g., a global sum. These global operations were performed using calls specific to the message-passing library of each machine.

During each iteration the walls of the enclosure are treated sequentially, and each processor deals with its own subregion. As an example, the bottom wall of the 2D enclosure shown in Figure 2 is treated simultaneously by all the processors. Then, the top, right, and left walls are treated similarly. After all the processors have finished the calculation of the incident fluxes on a wall, e.g., the bottom wall, for all the boundary cells assigned to them, they are able to compute the radiosity on those boundary cells using the most updated data. Then, every processor broadcasts the radiosities that have just been computed for their boundary cells. This is needed because the radiation beams fired from the boundary cells assigned to a processor do not necessarily hit, and in most cases will not hit, a cell that belongs to its own subregion. For example, the radiation beams fired by processor 1 in the positive x direction from the bottom boundary of the enclosure displayed in Figure 2*a* hit boundary cells assigned to processor 4. Since different processors have independent memories, every processor must communicate to all the others its most updated radiosities, because these radiosities are needed to calculate the radiation intensities leaving the walls. This data exchange occurs four times per iteration, following the treatment of each wall. The extension to the 3D case is straightforward.

Due to this frequent exchange of data, the number of iterations required to achieve convergence does not increase with the number of processors, as will be shown in the next section. Moreover, it will also be shown that for the computers used in this work, the communication time is small compared to the total time. As a consequence, good parallelization efficiencies are achievable. On the other hand, if the radiosities were exchanged only once per iteration, the calculations performed during one iteration would always be based on the radiosities calculated in the previous iteration. This would yield an increase of the number of iterations required to achieve convergence, with a consequent decrease of the efficiency.

The solution algorithm in each processor may be summarized as follows.

1. Initialize the data, such as grid data, angular discretization, boundary conditions, temperature, absorption and scattering coefficients of the medium, tolerance for the convergence criterion, number of processors. Set the iteration counter to 1.
2. Loop over all the boundaries, and for each boundary perform steps 3–5.
3. Set to zero the absolute value of the difference between successive iterations of the incident, E_i, and/or absorbed, E_a, energy on the wall boundary.
4. Loop over the boundary cells assigned to the processor; perform the following operations for each boundary cell.
   (*a*) Set the irradiation, H_P, on the boundary cell face to zero.
   (*b*) Loop over all the radiation beams fired from the central point of the cell face on the boundary; perform the following operations for each of them.
      (*i*) Track the path of the radiation beam until it hits an opposite boundary.

(*ii*) Get the radiosity at the hitting point and determine the radiation intensity of the radiation beams leaving the wall.
(*iii*) Integrate the RTE from the hitting point back to the firing point and determine the incident radiation at the firing point. Also compute and accumulate the contribution of the crossed control volumes to the radiation source of the energy equation.
(*iv*) Add the contribution of the radiation beam to the irradiation H_P.
(*c*) Apply the boundary conditions to determine the radiosity J_p.
(*d*) Update E_i and E_a by adding the contribution of the cell face.

5. Broadcast the radiosities at the boundary cells treated in step 4, as well as the values E_i and E_a.
6. If a radiative heat source is prescribed, broadcast the computed radiation source of the energy equation. Then, update the temperature field as reported in [20], and calculate the quantities required for the convergence criterion.
7. In the case of scattering media the incident radiation field is updated.
8. Check whether the convergence criterion is satisfied. If not, increase the iteration counter by one and return to step 2. Otherwise, print the results and stop.

In step 6 of the solution algorithm the temperature field may be updated in two different ways. In the first method, all the processors update the temperature field for the whole domain. In the second method, the domain is divided into subdomains and each processor updates the temperature field only in its own subdomain, as in the spatial domain decomposition parallelization strategy. Then, a global operation is needed such that all the processors know the temperature field in the whole domain. These two methods are also applicable to the update of the incident radiation field in step 7 of the solution algorithm. The second method was selected, since it was found that it is more efficient for the problems studied and the computers used in this work.

Spatial domain decomposition parallelization (DDP). In the DDP the domain is split up into a number of subdomains equal to the number of processors, and each subdomain is assigned to one processor. Each processor performs all the calculations for its subdomain. Radiation beams are fired both from the walls of the enclosure and from the boundary cells on the virtual boundaries, i.e., boundaries between neighboring processors. This parallelization strategy is illustrated in Figure 3, which shows a 2D domain mapped using a uniform grid with 8×8 control volumes, and one radiation beam fired per octant. Assuming that 4 processors are used, the path of all the radiation beams fired from the bottom wall and treated by processor 1 are drawn in Figure 3.

Knowledge of the radiation intensities of the radiation beams leaving the boundaries (walls and virtual boundaries) is required to integrate the RTE along the path of the beams. Therefore, the radiation intensities along the virtual boundaries are exchanged between neighboring processors at the end of each iteration.

To clarify this issue, an example is illustrated in Figure 4. It shows a domain split up into two subdomains, each assigned to one processor. Two radiation beams

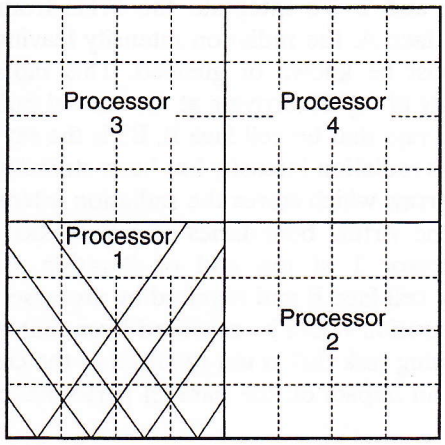


Figure 3. The spatial domain decomposition parallelization method.

are fired per octant ($N_\theta = 2$, $N_\varphi = 1$). At iteration number $n - 1$, when processor 2 is dealing with cell face B, on the virtual boundary, two radiation beams per octant are fired from the central point of cell face B. These radiation beams, B1 to B4, are tracked until they hit the boundaries of the subdomain treated by processor 2. These boundaries coincide with the boundaries of the enclosure, where the radiation intensity is known or guessed from the previous iteration. The RTE is integrated back to the central point of cell face B, along the path of each one of the four rays, allowing calculation of the radiation intensity incident on cell face B for each of the rays. These radiation intensities at the virtual boundaries are stored in an array, hereafter referred to as the RIVB array.

At iteration number n, when processor 1 is dealing with cell face A, two radiation beams per octant are fired from the central point of cell face A. These radiation beams, A1 to A4, are tracked until they hit the boundaries of the subdomain treated by processor 1. One of them, A1, hits cell face B, located on the

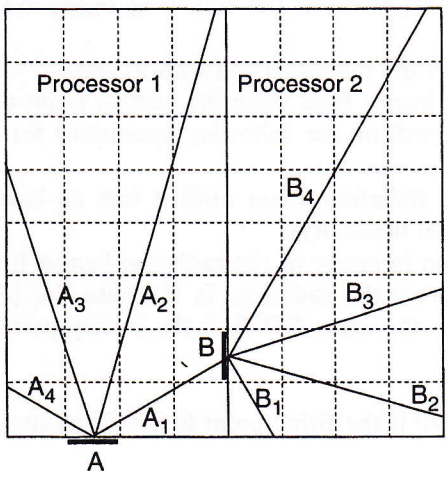


Figure 4. Treatment of the virtual boundaries in the spatial domain decomposition parallelization method.

virtual boundary between processors 1 and 2. To integrate the RTE from the hitting point to the central point of cell face A, the radiation intensity leaving the point where ray A1 hit cell face B must be known or guessed. This radiation intensity is taken as the radiation intensity of ray B3 arriving at the central point of cell face B in iteration $n - 1$ (among the rays that hit cell face B, B3 is the ray with a direction closest to that of ray A1). This radiation intensity has been stored in the array RIVB, as mentioned above. This array, which stores the radiation intensities of all the radiation beams arriving at the virtual boundaries of a processor, was transferred from processor 2 to processor 1 at the end of iteration $n - 1$. Therefore, the radiation intensity leaving cell face B and required by processor 1 to perform the calculations for ray A1 in iteration n can be retrieved from that array. This retrieval operation is a time-consuming task that is not required in the case of a single processor, and therefore it has an impact on the parallel performance, as will be shown in the results section.

The direction of radiation transport may change at interprocessor boundaries, but this possible distortion can be reduced as much as required by refining the angular discretization. An angular discretization refinement study should be carried out in the same way as grid refinement studies are performed, to ensure that this problem is overcome. This study is particularly important in the case of optically thin media with localized heat sources.

During the first iteration, the radiation intensities at the virtual boundaries were assumed to be equal to the radiation intensities at the boundaries of the enclosure. This assumption may influence the convergence rate and will be further investigated in the future.

The solution algorithm in each processor may be summarized as follows.

1. As in the RDP.
2. Loop over all the boundaries (walls or interfaces between neighboring processors), and for each boundary perform steps 3–5.
3. If part of the boundary is a wall, set to zero the absolute value of the difference between successive iterations of the incident, E_i, and/or absorbed, E_a, energy on the wall.
4. Loop over all the boundary cells; perform the following operations for each boundary cell.
   (*a*) Set the incident radiation on the boundary cell face to zero.
   (*b*) Loop over all the radiation beams fired from the central point of the cell face on the boundary; perform the following operations for each of them.
      (*i*) Track the path of the radiation beam until it hits an opposite boundary (wall or virtual boundary).
      (*ii*) Determine the radiation intensity of the radiation beams leaving that boundary either from the radiosity, in the case the hitting point is a wall, or from the array RIVB, if the hitting point is a virtual boundary.
      (*iii*) As in the RDP.
      (*iv*) Do either as in the RDP if the firing point is a wall, or store the

incident radiation in the array RIVB if the firing point is a virtual boundary.

(c, d) Do as in the RDP if the firing point is a wall; otherwise, skip these items.

5. Exchange the array RIVB between neighboring processors.
6. Broadcast E_i and E_a.
7. If a radiative heat source is prescribed, update the temperature field as reported in [20]. Calculate the quantities required for the convergence criterion and broadcast them.
8. In the case of scattering media the incident radiation field is updated.
9. Check whether the convergence criterion is satisfied. If not, increase the iteration counter by one and return to step 2. Otherwise, print the results and stop.

RESULTS AND DISCUSSION

The parallel strategies described in the previous section were applied to three test cases, and the results obtained are presented and discussed below. Most computations were performed on an Intel iPSC/860 using 1, 2, 4, and 16 processors, and using method 1 to split the rays among the processors in the RDP. In the last test case an IBM SP2 was used to show the generality of the parallelization methods, and the two ray splitting methods used in the RDP were compared. The parallel performance is evaluated by means of the efficiency and speed-up. The speed-up is defined as $S_p = t_1/t_p$, where t_1 and t_p are the wall clock execution time on 1 and p processors, respectively. The speed-up is closely related to the efficiency of the parallel implementation, defined as $E_p = S_p/p$. The efficiency per iteration is also important in the analysis of the results. It is defined as $E_p * n_p/n_1$, where n_1 and n_p are the number of iterations required to achieve convergence using 1 and p processors, respectively.

Square Domain with Prescribed Medium Temperature

In the first test case a two-dimensional square enclosure with cold, gray walls was studied. The enclosure contains a gray nonscattering medium maintained at an emissive power of unity. Preliminary calculations were performed for black walls and the results were compared with the analytical solution [21]. It was verified that the parallel and the sequential codes yield similar results, which are in close agreement with the analytical solution. However, these results are not presented here because the solution method is not iterative if the walls are black and their temperature is known. To examine the parallel performance of the algorithm we have considered gray walls, since the algorithm is iterative in this case. Although no analytical solution is available in the case of gray walls, the numerical solutions computed using the parallel code and a sequential one, which was thoroughly tested in the past, were systematically compared. It was found that the numerical solutions are identical.

The standard calculations were performed using a discretization with 16 solid angles per octant ($N_\theta = 4$, $N_\varphi = 4$), a grid with 128×128 control volumes, a medium with a unity optical thickness, and an enclosure with gray boundaries and an emissivity equal to 0.4. In the results presented below, one of these five parameters (number of processors, angular discretization, grid size, optical thickness, and emissivity of the boundaries) was varied, while the others were kept unchanged.

Figure 5 shows the efficiency, the speed-up, the number of iterations required to achieve convergence, n_{iter}, and the ratio of the communication time to the total elapsed time, t_{cm}/t_e, as a function of the number of processors, p. The efficiency decreases with p, as expected, but the decrease is relatively slow for the RDP, where $E_p = 91.3\%$ for $p = 16$, while it is quite fast for the DDP, where $E_p = 26.9\%$ for $p = 16$.

The significant difference in the efficiency between the two different parallel strategies is explained, to some extent, by the evolution of the number of iterations with p. In the RDP n_{iter} is independent of p, but in the DDP n_{iter} increases markedly with p. In the last case, the calculations performed in a subdomain at the nth iteration require knowledge of the radiative intensities at the boundaries of that subdomain. In the case of virtual boundaries, these radiation intensities have been calculated at the neighboring subdomains during the $(n-1)$th iteration,

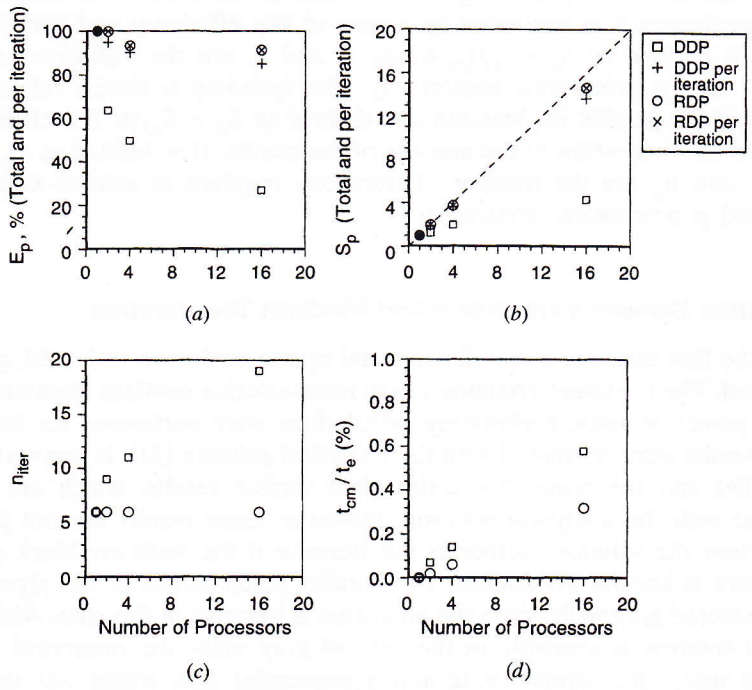


Figure 5. Influence of the number of processors on efficiency (*a*), speed-up (*b*), number of iterations (*c*), and ratio of communication to execution time (*d*) for test case 1.

stored in array RIVB, and transferred to the subdomain under consideration just before the end of that iteration, in step 5 of the solution algorithm. These local communications, inherent to the DDP strategy, prevent the information from traveling beyond the boundaries of a subdomain in one iteration, in contrast to the RDP, where the information is able to travel through the whole domain in one iteration. As a consequence, the DDP yields an increase of n_{iter} with p, in the same way as the DDP commonly used in computational fluid dynamics (CFD), although in radiation problems the percentage of increase of the number of iterations with p is larger than in CFD, as discussed in [22].

If the increase of n_{iter} with p were the only reason for the decrease of E_p with p, then the efficiency per iteration would remain constant regardless of the number of processors. However, the efficiency per iteration decreases with p, as shown in Figure 5*a*. The time required to transfer data among the processors increases with p, causing a decrease of the efficiency per iteration, but it has a minor role, as shown in Figure 5*d*. In fact, t_{cm}/t_e does not exceed 0.6%. Therefore, the cause for the decrease of the efficiency with p lies elsewhere, as explained below.

In the RDP the efficiency is 99.9% for $p = 2$, and the communication time is the only reason why E_p is smaller than 100%. However, if $p = 4$, then $E_p = 93.3\%$. In this case each wall of the enclosure is split into four regions, and the radiation beams fired from each of them are assigned to a different processor. The computational load associated to tracking the radiation beams and applying Eq. (3) along their paths depends on the number of cells crossed. Hence, the computational load for the two regions close to the corners is different from the computational load of the two central subregions. This is illustrated in Figure 6, which shows the total time required to perform the calculations for all the radiation beams fired from each region. The processors assigned to the corner regions have a lower computational load and must wait until the others have finished their calculations. Figure 5 shows that this load imbalance, hereafter referred to as the RDP load imbalance, also occurs for $p = 16$, but not for $p = 2$, for symmetry reasons. This explains why the efficiency is very close to 100% for $p = 2$, but decreases for $p = 4$ and $p = 16$. It was found that the RDP load imbalance is independent of the spatial and angular discretizations.

In the DDP the efficiency per iteration also decreases with p, and it is smaller than in the RDP. When the number of processors is increased, the number of radiation beams fired from the boundary of each processor is smaller, as well as the distance traveled by each ray until it hits an opposite boundary. Indeed, it was verified that the computational load required to track the radiation beams and to compute the radiation intensities via Eq. (3) is proportional to the number of control volumes per subdomain. However, there is also a computational load associated to the storage and retrieval of radiation intensities in the array RIVB. The computational load of these additional operations is proportional to the number of cell faces on the boundary of a subdomain. Therefore, the role of these additional operations increases with p, justifying the corresponding decrease of efficiency per iteration.

The influence of grid size on parallel performance is shown in Figure 7. In both parallelization strategies, and for a fixed number of processors, the number of

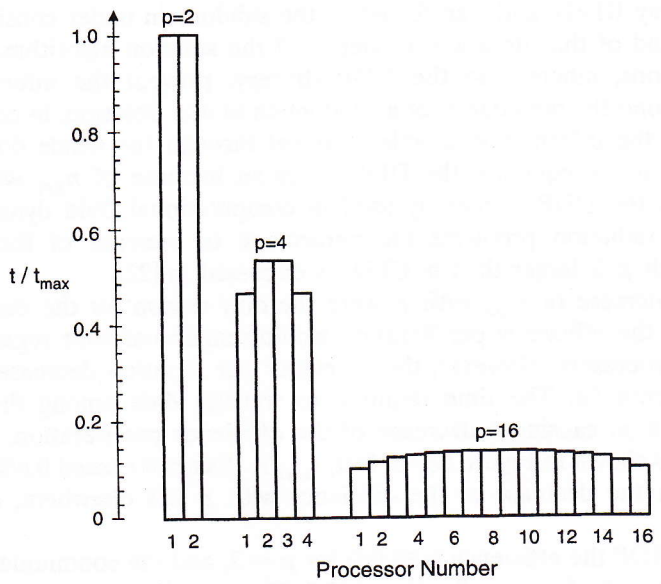


Figure 6. Distribution of computational load among the processors in the RDP for test case 1.

iterations is independent of the grid size, and the ratio t_{cm}/t_e decreases with grid size. Although this ratio is again small, it explains the small increase of the efficiency with p observed in the RDP, since the RDP load imbalance is not influenced by grid size. In the DDP the ratio t_{cm}/t_e explains only in part the observed increase of efficiency with grid size. For example, if $p = 16$ we have $E_p = 21.3\%$ for the coarser grid and $E_p = 26.9\%$ for the finer one, that is, E_p increases 5.6%. The efficiency per iteration increases even more. To explain this evolution fully it is necessary to take into account the additional operations in the array RIVB. As stated above, the computational load of these operations is proportional to the number of cell faces on the boundary of a subdomain, which increases linearly with grid size. But the computational load of the calculations is proportional to the number of control volumes in a subdomain, which increases quadratically with grid size ($N_x = N_y$). Therefore, the role of the additional storage and retrieve operations is attenuated with grid refinement, justifying the observed increase in efficiency.

Figure 8 shows the influence of angular discretization on parallel performance. It can be seen that n_{iter} is also independent of the angular refinement for both parallelization methods. Again, the communication time is very small compared to the execution time. However the ratio t_{cm}/t_e exhibits different trends according to the parallelization approach. In fact, the computation time is proportional to the number of radiation beams fired per boundary cell. In the RDP the communications appear in the broadcast operations in step 5 of the solution algorithm, and the time of those operations is independent of the angular discretization. Therefore, t_{cm}/t_e decreases for finer angular discretizations. In the

DDP the communications are associated with the local data transfer in step 5 of the solution algorithm, and with the broadcast operations in step 6. The time spent in the former task is proportional to the number of radiation beams fired per boundary cell, while the time of the last task is independent of the angular discretization. Since the former task is more time consuming, it turns out that t_{cm}/t_e is weakly dependent on N_θ and N_φ.

The efficiency of the RDP increases slightly with $N_\theta * N_\varphi$, due to the corresponding decrease of t_{cm}/t_e. The efficiency of the DDP exhibits more irregular behavior. In fact, it should not be influenced by the operations in the array RIVB, because the time involved both in such operations and in the solution of the RTE is proportional to $N_\theta * N_\varphi$. Therefore, the minor role of t_{cm}/t_e suggests a negligible dependence of the efficiency on the angular discretization. Figure 8 shows that this is not so, especially when $N_\theta = N_\varphi$ increases from 4 to 6. It is believed that the explanation lies on the cache effect. The efficiencies are determined on the basis of the execution time in one processor. If more than one processor is used, the size of the arrays diminishes accordingly, and the efficiencies may be higher than expected if the cache is used more efficiently (i.e., the arrays fit in the cache) for p processors than for one processor. This is what has been observed for $p = 4$ or $p = 16$, yielding a superlinear speed-up per iteration when $p = 4$.

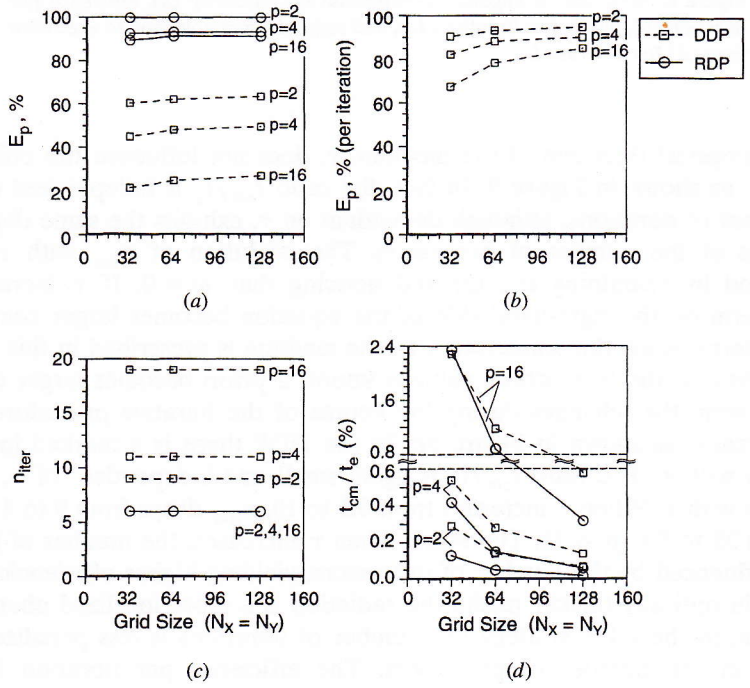


Figure 7. Influence of grid size on efficiency (*a*), efficiency per iteration (*b*), number of iterations (*c*), and ratio of communication to execution time (*d*) for test case 1.

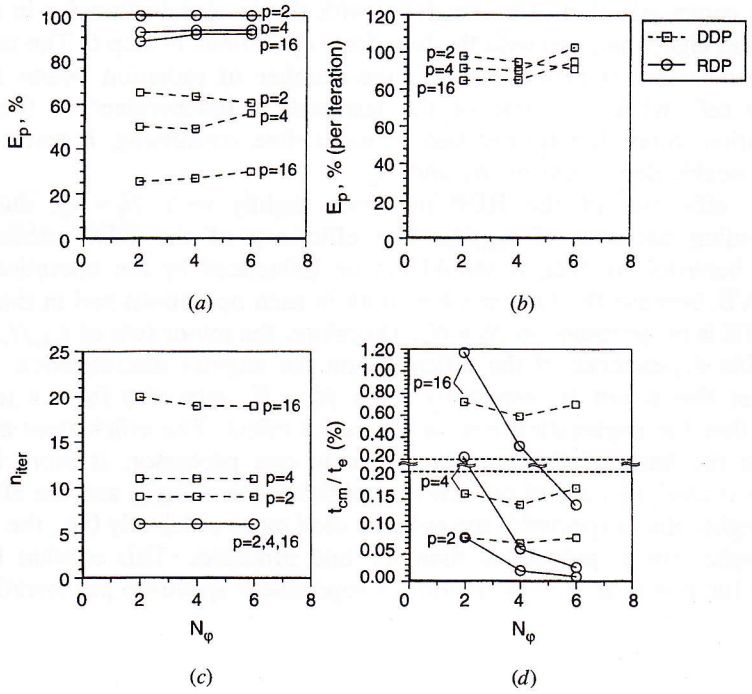


Figure 8. Influence of angular discretization on efficiency (*a*), efficiency per iteration (*b*), number of iterations (*c*), and ratio of communication to execution time (*d*) for test case 1.

The optical thickness of the medium, τ, does not influence the efficiency in the RDP, as shown in Figure 9. In fact, the ratio t_{cm}/t_e is independent of τ, and the number of iterations, although dependent on τ, exhibits the same dependence regardless of the number of processors. The evolution of n_{iter} with τ may be understood by examining Eq. (3) and noticing that $\omega = 0$. If τ increases, the second term on the right-hand side of the equation becomes larger compared to the first term. Since the temperature of the medium is prescribed in this test case, this means that the term whose value is known a priori becomes larger compared with the term that changes during the course of the iterative procedure. Hence, n_{iter} decreases as shown in Figure 9*c*. In the DDP there is a marked increase of efficiency with τ. The ratio t_{cm}/t_e is again small and independent of τ, and n_{iter} decreases with τ. When τ increases from 0.1 to 10, n_{iter} drops from 9 to 4 if $p = 1$, and from 35 to 5 if $p = 16$. Therefore, when τ increases, the number of iterations is less influenced by the number of processors, yielding higher efficiencies. This is because in optically thicker media the radiation is a more localized phenomenon. Therefore, in the DDP strategy the number of iterations is less penalized by the increase in the number of processors. The efficiency per iteration is weakly dependent on τ.

The influence of the emissivity of the boundary, ε_w, is displayed in Figure 10. The ratio t_{cm}/t_e, not shown in the figure, is small and independent of ε_w, while

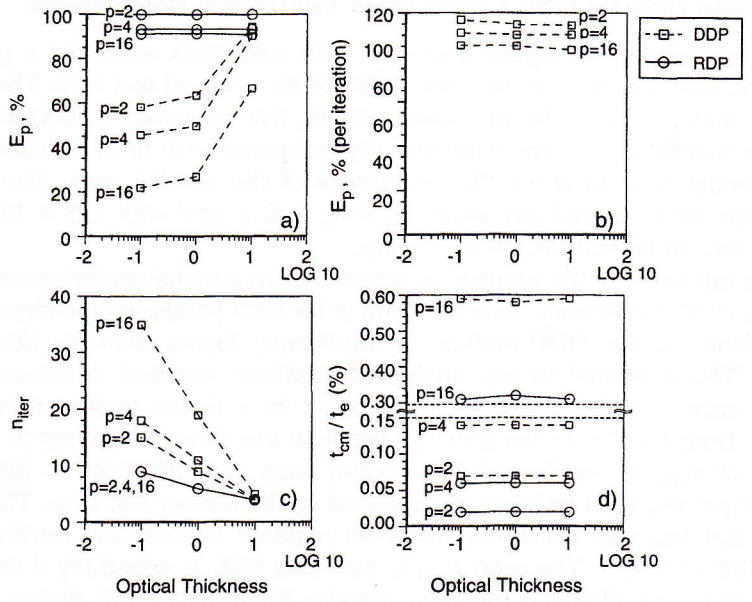
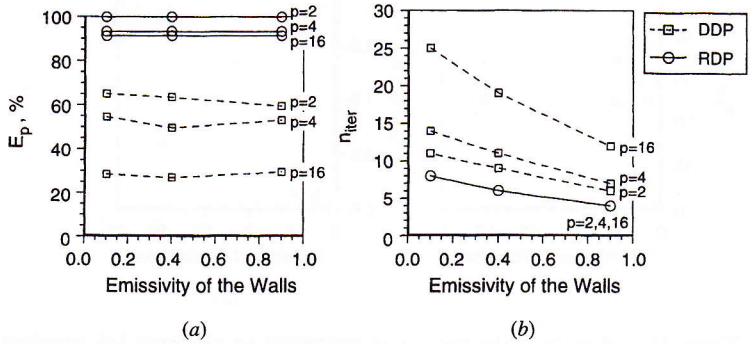


Figure 9. Influence of optical thickness on efficiency (*a*), efficiency per iteration (*b*), number of iterations (*c*), and ratio of communication to execution time (*d*) for test case 1.

n_{iter} decreases with ε_w. The efficiency is independent of ε_w if the RDP method is used, as explained above for the influence of τ. But now the efficiency is also weakly dependent on ε_w if the DDP is employed. This is due to the ratio of n_{iter} using p processors to n_{iter} using one processor, which is approximately independent of ε_w.



(*a*) (*b*)

Figure 10. Influence of emissivity of the walls on efficiency (*a*) and number of iterations (*b*) for test case 1.

Square Domain with a Prescribed Volumetric Heat Source

A two-dimensional square enclosure with cold black walls and a prescribed volumetric heat source (1 W/m^3) was selected as a second test case. The medium does not scatter. As in the previous problem, the numerical solutions obtained using the parallel code were systematically compared with those calculated using the sequential code to check the correctness of the parallel code. The standard calculations were carried out using $N_\theta = N_\varphi = 4$, a grid with 128×128 control volumes, and an optical thickness of unity.

The influence of the number of processors on parallel performance is shown in Figure 11. The efficiency decreases with p for both parallelization strategies, but the efficiency of the DDP method is consistently higher than for the previous problem. This is related to the number of iterations required to achieve convergence. Figure 11*c* shows that n_{iter} increases by a factor smaller than 2 as p increases from 1 to 16. In the previous problem this factor was about 3. Since the increase of n_{iter} is smaller in the present case, the efficiency is higher. The efficiency per iteration behaves as explained in the former test case. The reasons given in that test case concerning the time required to store and retrieve data in array RIVB still hold. The ratio t_{cm}/t_e increases with p, especially if the RDP is used. In this case, although the ratio remains small, it is much higher than that observed in the first test case. This is due to the global communication of the 2D

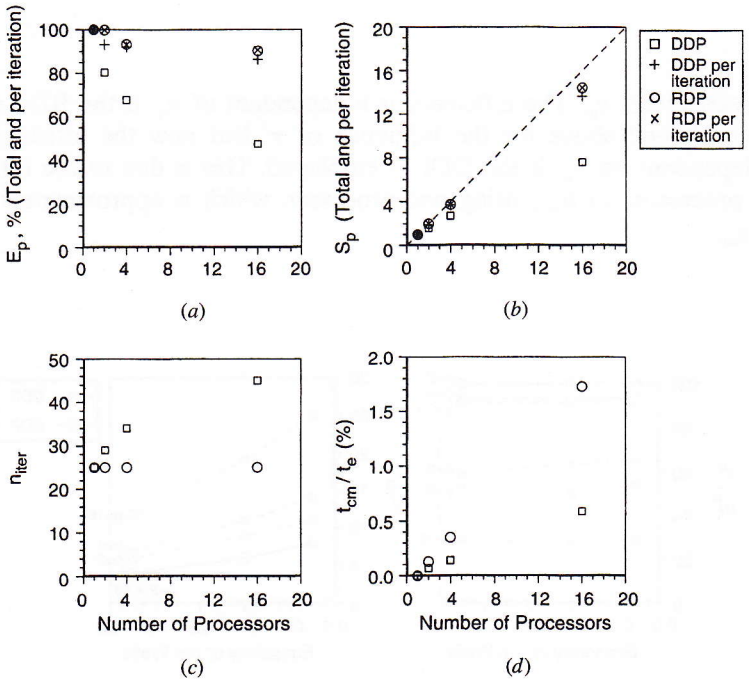


Figure 11. Influence of the number of processors on efficiency (*a*), speed-up (*b*), number of iterations (*c*), and ratio of communication to execution time (*d*) for test case 2.

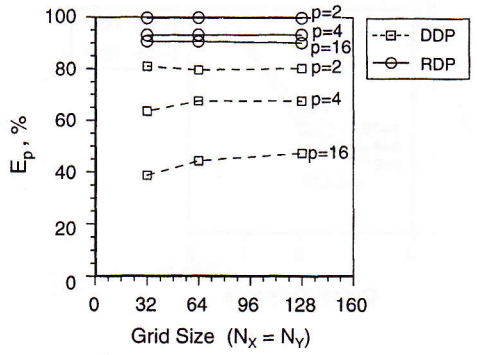


Figure 12. Influence of grid size on the number of processors for test case 2.

array of radiation sources of the energy equation in step 6 of the solution algorithm, which was not needed in that problem.

Figures 12 and 13 show the influence of the spatial and angular discretizations on the parallel performance. The evolutions are very similar to those described in the previous test case, as well as the evolutions of n_{iter} and t_{cm}/t_e, not displayed here. The reader is referred to the explanations given there for the interpretation of these results.

Finally, the evolution of E_p and n_{iter} with τ is shown in Figure 14. The efficiency is approximately independent of τ in the RDP, and increases with τ in the DDP, as already observed in the first test case. The ratio t_{cm}/t_e remains independent of τ, and too small to play a relevant role. However, contrary to the previous problem, now n_{iter} increases with τ. In the present problem the temperature field is not prescribed, and therefore I_b is calculated iteratively during the course of the solution algorithm via the energy equation. The second term on the right-hand side of Eq. (3) becomes dominant as τ increases, justifying the increase of n_{iter} with τ. But the key issue is that in the DDP the ratio of n_{iter} using p processors to n_{iter} using one processor decreases if τ increases, as in test case 1 (e.g., if $\tau = 0.1$, then $n_{\text{iter}} = 9$ for $p = 1$ and $n_{\text{iter}} = 20$ for $p = 16$, while if $\tau = 10$ then $n_{\text{iter}} = 337$ for $p = 1$ and $n_{\text{iter}} = 414$ for $p = 16$). This is why the efficiency exhibits the same dependence on the optical thickness in both problems, despite the opposite evolutions of n_{iter}.

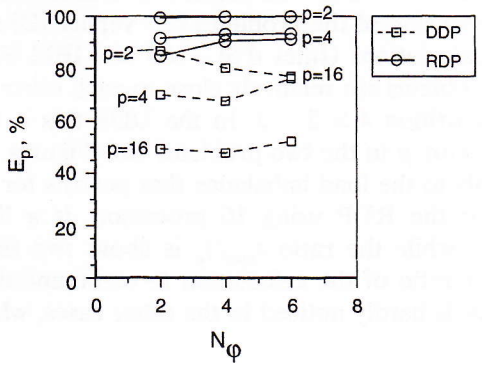
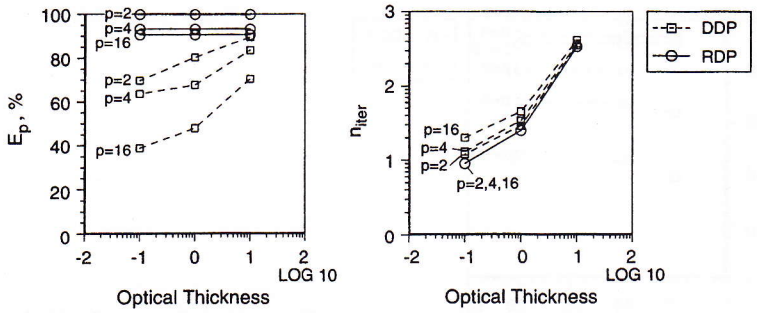


Figure 13. Influence of angular discretization on the number of processors for test case 2.



(*a*) (*b*)

Figure 14. Influence of optical thickness on efficiency (*a*) and number of iterations (*b*) for test case 2.

Three-Dimensional Rectangular Enclosure with a Prescribed Volumetric Heat Source

The idealized furnace studied in [23] constitutes the last test problem. It is a three-dimensional rectangular enclosure ($4.0 \times 2.0 \times 2.0$ m^3) containing an emitting, absorbing, and isotropically scattering medium. The emissivity of the four side walls (4.0×2.0 m^2 each) is 0.70, and the temperature is 900 K. The emissivities of the firing-end and exit-end walls are 0.86 and 0.70, respectively, and the corresponding wall temperatures are 1,200 K and 400 K, respectively. A uniform volumetric heat source equal to 5.0 kW/m^3 is prescribed. The predicted results closely follow those reported in [23], and it was checked that they are independent of the number of processors for both parallelization strategies. The calculations were carried out using $64 \times 64 \times 64$ control volumes and an angular discretization with $N_\theta = N_\varphi = 4$.

Figure 15 shows the results obtained for $\kappa = 0.5$ m^{-1} and $\sigma_s = 0$. In the case of 16 processors, two different partitions were considered in the DDP: $4 \times 2 \times 2$ and $4 \times 4 \times 1$. The former partition is more efficient than the latter, as a result of the faster convergence. The smaller ratio t_{cm}/t_e also contributes to the higher efficiency of the partition $4 \times 2 \times 2$, but it plays a minor role due to the low values of that ratio. The evolutions of the efficiency, speed-up, n_{iter}, and t_{cm}/t_e for both DDP and RDP closely follow the trends observed for the previous two-dimensional test problem. Despite the different dimension of the problems (2D versus 3D) and the different computers used in the calculations (Intel iPSC/860 and IBM SP2), the efficiencies obtained for the two problems are relatively close to each other for both the RDP and the DDP with a partition $4 \times 2 \times 2$. In the DDP this is due mainly to the similar variation of n_{iter} with p in the two problems (see Figures 11*c* and 15*c*). In the RDP this is due mainly to the load imbalance that persists for 3D problems. The efficiency obtained for the RDP using 16 processors is a little smaller than in the previous problem, while the ratio t_{cm}/t_e is about two times higher. This is explained by the higher ratio of the calculation to communication time in the IBM than in the Intel. This is hardly noticed in the other cases, where the ratio t_{cm}/t_e is too small.

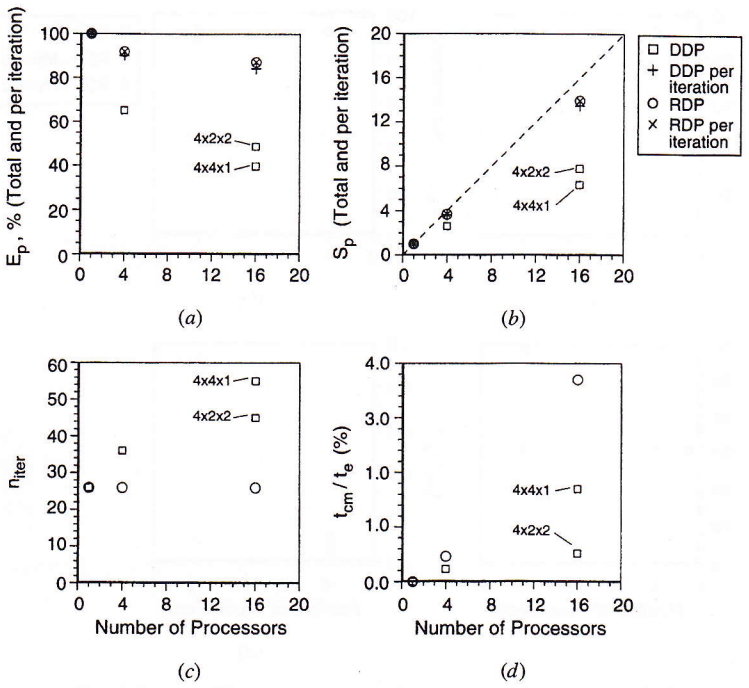


Figure 15. Influence of the number of processors on efficiency (*a*), speed-up (*b*), number of iterations (*c*), and ratio of communication to execution time (*d*) for test case 3 without scattering. In the DDP with 16 processors the results of two different partitions are shown: $4 \times 4 \times 1$ and $4 \times 2 \times 2$.

Further calculations were carried out for $\kappa = 0.15\ \mathrm{m}^{-1}$ and $\sigma_s = 0.35\ \mathrm{m}^{-1}$. The $4 \times 2 \times 2$ partition was selected for the DDP. The results, shown in Figure 16, reveal that for a fixed number of processors n_{iter} has increased compared to the nonscattering case, as expected. In the DDP the ratio of n_{iter} using p processors to n_{iter} using one processor is slightly higher in the present scattering case. This implies that the efficiency is slightly lower, since t_{cm}/t_e remains very small. However, the efficiency per iteration has increased a little due to the additional calculations associated with step 8 of the solution algorithm. In the RDP the efficiency is similar in the two cases, but marginally smaller for the scattering medium, as a result of the higher t_{cm}/t_e ratio. This higher ratio is due to the global communication of the incident radiation field in step 7 of the solution algorithm.

Figure 16 shows also the results obtained using method 2 for splitting the radiation beams among the processors in the RDP. In this problem method 2 significantly reduces the load imbalance, as revealed by Figure 17, which represents the distribution of the computational load among the processors for both methods. Therefore, better efficiency is obtained. Although method 2 is more likely to result in a better load balance than method 1, in a general case it is not guaranteed to be so. An optimum load balance might be achieved by introducing a preprocessing step at the expense of higher algorithmic complexity, but this is outside the scope of the present work.

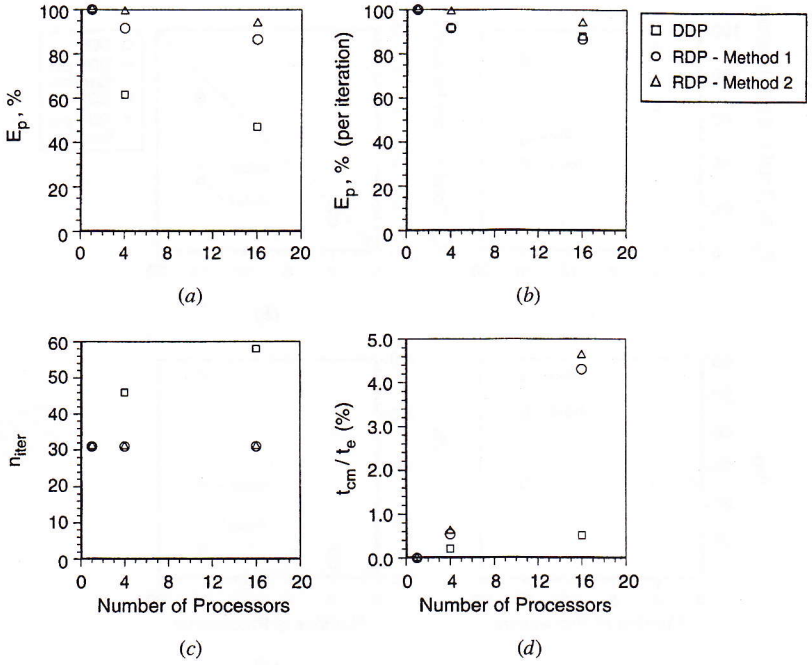


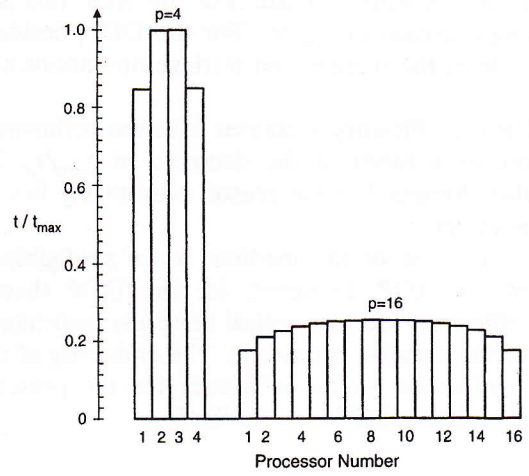
Figure 16. Influence of the number of processors on efficiency (*a*), efficiency per iteration (*b*), number of iterations (*c*), and ratio of communication to execution time (*d*) for test case 3 with isotropic scattering.

Some of the parallelization issues used in this study may be applied to other methods for the solution of the RTE. In particular, some ideas used in the present work were also used in the parallelization of the discrete ordinates method (DOM) reported in [15, 16]; e.g., the DDP technique developed for the DTM was used in a very similar way in the DOM.

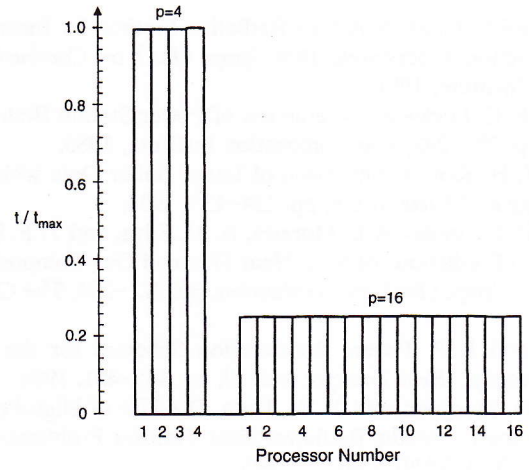
CONCLUSIONS

The DTM was parallelized using two different methods, the RDP and the DDP. In the first method each processor performs calculations for the whole domain and for a subset of the total number of rays. In the second case, the domain is split into subdomains and each processor performs the calculations for all the rays within its own subdomain. Two- and three-dimensional enclosures containing emitting-absorbing and isotropically scattering media were studied, either with a prescribed medium temperature or with a prescribed volumetric heat source. The calculations were performed on an Intel iPSC/860 and on an IBM SP2 using up to 16 processors. From the results obtained, the following conclusions may be drawn.

1. The efficiency decreases with the increase of p for both parallelization strategies. But while high efficiencies, generally above 90%, are achieved using the RDP, the efficiency of the DDP is low, often below 50%. The



(*a*)



(*b*)

Figure 17. Distribution of computational load among the processors in the RDP for test case 3: (*a*) method 1; (*b*) method 2.

efficiency of the DDP is limited by the increase of the number of iterations with p. This problem does not occur with the RDP, where the convergence rate is independent of p.

2. The efficiency per iteration also decreases with increase of p. For the RDP this is due to a load imbalance arising from the partition of the radiation beams among the processors. For the DDP this is due to the additional operations required to store and retrieve the radiation intensities at the virtual boundaries between neighboring processors.

3. Efficiency increases with grid size. For the RDP this is explained by the corresponding decrease of t_{cm}/t_e. For the DDP, besides the decrease of t_{cm}/t_e, the role of the storage and retrieve operations also decreases with grid size.
4. For the RDP the efficiency increases with the refinement of the angular discretization, as a result of the decrease of t_{cm}/t_e. For the DDP the efficiency also changes, but the reason presumably lies in the role played by the cache effect.
5. The optical thickness of the medium has a negligible influence on the efficiency of the RDP. However, for the DDP there is a significant increase of efficiency with the optical thickness associated with the convergence rate of the iterative procedure. The emissivity of the boundaries has a negligible influence on the efficiency for the prescribed temperature case and for both parallelization methods.

REFERENCES

1. F. C. Lockwood and N. G. Shah, A New Radiation Method for Incorporation in General Combustion Prediction Procedures, 18*th Symp.* (*Int.*) *on Combustion*, pp. 1405–1414, The Combustion Institute, 1981.
2. A. S. Abbas and F. C. Lockwood, Prediction of Power Station Boilers, 21*st Symp.* (*Int.*) *on Combustion*, pp. 285–292, The Combustion Institute, 1986.
3. R. K. Boyd and J. H. Kent, Comparison of Large Boiler Data with Combustion Boiler Predictions, *Energy and Fuels*, vol. 8, pp. 124–130, 1994.
4. M. G. Carvalho, P. J. Coelho, A L. Moreira, A. M. Silva, and T. F. Silva, Comparison of Measurements and Predictions of Wall Heat Flux and Gas Composition in an Oil-Fired Utility Boiler, 25*th Symp.* (*Int.*) *on Combustion*, pp. 227–234, The Combustion Institute, 1994.
5. W. A. Fiveland and J. P. Jessee, Acceleration Schemes for the Discrete Ordinates Method, *J. Thermophys. Heat Transfer*, vol. 10, pp. 445–451, 1996.
6. L. A. Gritzo, R. D. Skocypec, and T. W. Tong, The Use of High-Performance Computing to Solve Participating Media Radiative Heat Transfer Problems—Results of an NSF Workshop, Sandia Rep. SAND95-0225, 1995.
7. J. R. Howell, Thermal Radiation in Participating Media: the Past, the Present, and Some Possible Futures, *J. Heat Transfer*, vol. 110, pp. 1220–1229, 1988.
8. C. Saltiel and M. H. N. Naraghi, Parallel Processing Approach for Radiative Heat Transfer Prediction in Participating Media, *J. Thermophys. Heat Transfer*, vol. 7, pp. 739–742, 1993.
9. P. J. Burns and D. V. Pryor, Vector and Parallel Monte Carlo Radiative Heat Transfer Simulation, *Numer. Heat Transfer B*, vol. 16, pp. 97–124, 1989.
10. R. G. Minnich and D. V. Pryor, Radiative Heat Transfer on a SPARCStation Farm, *Concurrency: Practice and Experience*, vol. 5, pp. 345–357, 1993.
11. U. R. Hanebutte and E. E. Lewis, A Massively Parallel Algorithm for Radiative Transfer Calculations, ASME Paper 91-WA-HT-10, 1991.
12. A. Benmalek, T. W. Tong, and W. Li, Distributed-Memory Parallel Algorithm for the Solution of the Spectral Radiative Transfer Equation, AIAA Paper 96-0606, 1996.
13. S. P. Burns and M. A. Christon, Spatial Domain-Based Parallelism in Large Scale Participating-Media, Radiative Transport Applications, *Numer. Heat Transfer B*, vol. 31, pp. 401–422, 1997.

14. S. P. Burns, Application of Spatial and Angular Domain Based Parallelism to a Discrete Ordinates Formulation with Unstructured Spatial Discretization, in M. P. Mengüç (ed.), *Radiative Transfer—II, Proc. 2nd Int. Symp. on Radiative Heat Transfer*, pp. 173–193, Begell House, New York, 1998.
15. P. J. Coelho, J. Gonçalves, and P. Novo, Parallelization of the Discrete Ordinates Method: Two Different Approaches, in J. Palma and J. Dongarra (eds.), *Vector and Parallel Processing, VECPAR'96*, Springer-Verlag Lecture Notes in Computer Science, vol. 1215, pp. 222–235, 1997.
16. J. Gonçalves and P. J. Coelho, Parallelization of the Discrete Ordinates Method, *Numer. Heat Transfer B*, vol. 32, pp. 151–173, 1997.
17. J. Gonçalves and P. J. Coelho, Parallelization of the Finite Volume Method, in M. P. Mengüç (ed.), *Radiative Transfer—II, Proc. 2nd Int. Symp. on Radiative Heat Transfer*, pp. 209–219, Begell House, New York, 1998.
18. P. J. Novo, P. J. Coelho, and M. G. Carvalho, Parallelization of the Discrete Transfer Method: Two Different Approaches, *ASME HTD—vol.* 235, pp. 45–54, 1996.
19. N. W. Bressloff, Parallelisation of the Discrete Transfer Radiation Model, in D. R. Emerson, J. Periaux, A. Ecer, N. Satofuka, and P. Fox (eds.), *Parallel Computational Fluid Dynamics*, pp. 307–314, Elsevier, Amsterdam, 1998.
20. P. J. Coelho and M. G. Carvalho, A Conservative Formulation of the Discrete Transfer Method, *ASME J. Heat Transfer*, vol. 76, pp. 919–929, 1997.
21. N. G. Shah, New Method of Computation of Radiation Heat Transfer in Combustion Chambers, Ph.D. thesis, Imperial College of Science and Technology, London, 1979.
22. P. J. Coelho, P. J. Novo, and M. G. Carvalho, Modelling of a Utility Boiler Using Parallel Computing, *J. Supercomput.*, in press, 1999.
23. M. P. Mengüç and R. Viskanta, Radiative Transfer in Three-Dimensional Rectangular Enclosures Containing Inhomogeneous Anisotropically Scattering Media, *J. Quant. Spectrosc. Radiative Transfer*, vol. 33, pp. 533–549, 1985.