

Dynamic modelling of radiant heat from wildfires

J. E. Hilton ^a, Justin Leonard ^b, Raphaelae Blanche ^b, Glenn Newnham ^b, Kimberley Opie ^b,
Chris Rucinski ^a and William Swedosh ^a

^aData61, CSIRO, Clayton South, VIC 3169, Australia

^bLand & Water, CSIRO, Clayton South, VIC 3169, Australia
Email: Justin.Leonard@csiro.au

Abstract: Risk evaluation in wildfire prone areas requires the determination of the radiant heat incident on structures and other elements. The radiant heat flux profile can be used to determine the likelihood of failure of various elements through ignition or thermal degradation. The radiant heat flux incident at a particular point is, however, challenging to calculate as it requires evaluation of all possible lines of sight from an emitting (hot) surface to the point with possible obstruction and attenuation through smoke and vegetation. This is compounded by the behaviour of wildfires, which dynamically change over short time scales creating complex spatially and temporally varying emitting surfaces.

In this study we implemented a digital differential analyser (DDA) ray casting algorithm for calculation of radiant heat flux view factors on graphics processing units (GPUs). GPUs are a new generation of processing architecture that allow for massively parallel calculations on commonly available computer hardware. We show that implementation of a ray casting DDA algorithm on GPUs allows radiant heat fluxes to be calculated orders of magnitude faster than CPU-based implementations. The method can also handle factors such as obstructions and variable transmission coefficients.

The GPU-based DDA method is rapid enough to allow heat fluxes to be evaluated for dynamic fire spread scenarios. The method can be linked to any predefined scene or fire simulator. In this study it was coupled to Spark, a wildfire simulation toolkit (<http://research.csiro.au/spark/>), which was used to model the behaviour of small fires and create three-dimensional flame regions. An example of the maximum heat flux on two spherical receivers ('A' and 'B') from a simulation using the McArthur rate-of-spread model is shown in Fig. 1a. To demonstrate the ability of the method to handle shielding one receiver ('A') was partially shielded by an 'L' shaped obstruction. The profile in Fig. 1b shows the heat flux over time at the two receivers and demonstrates the reduction due to shielding. The method can be used to calculate accurate heat fluxes from dynamic fire scenarios to inform risk metrics and to trial and evaluate mitigation strategies for homes and infrastructure.

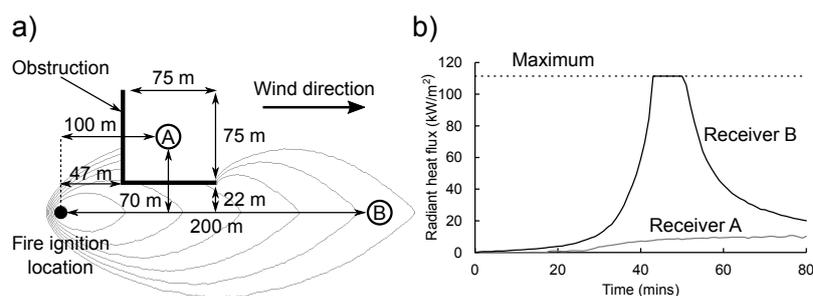


Figure 1. a) Schematic set up of dynamic test case with two receivers, one partially shielded ('A') and one in the path of the flame ('B'). Ten minute isochrones of the fire simulation are shown as contours. b) Plot of radiant heat flux at the two receivers against time. The maximum possible heat flux with a view factor $F = 1$ and the parameters used in the test case is plotted as horizontal dashed line.

Keywords: Wildfires, heat flux, modelling

1 INTRODUCTION

Safety and risk assessment of homes and infrastructure in areas impacted by wildfires require accurate evaluation of radiant heat fluxes under different fire scenarios. Although the physical relations governing radiative heat transfer are relatively straightforward, application to general real-world scenarios are challenging due to the calculations required. The heat flux at a particular point requires all possible lines of sight from all emission sources to be determined and checked for obstruction and attenuation. The complexity of these calculations currently limits widespread practical use. Safety standards such as the Australian AS3959 (2009), governing construction in wildfire prone areas, make extensive use of radiant heat flux estimations to determine bushfire attack levels (BAL). These BAL levels are used to determine the level of protection required in construction. However, the modelling techniques incorporated into the standard are necessarily limited in order to make calculations tractable. For example, flame regions are considered only as rectangles of a given width, height and inclination. The methods in AS3959 (2009) also do not take into account vegetation attenuation or dynamic effects from flame propagation. Even with these assumptions the resulting ‘full’ method to determine heat flux (method 2) is complex and time-consuming to calculate and results in a single estimate of peak radiant heat flux.

In this study we present an implementation of radiant heat flux calculation using a ray casting digital differential analyser (DDA) algorithm. The method can handle complex flame shapes, attenuation and obstruction of radiation. The method can also provide time based radiation heat flux profiles which can better inform the potential failure processes of building elements. The DDA algorithm is implemented in parallel on computational graphics processing (GPU) architecture. The speed of current generation GPUs allow accurate full scene ray casting at meter scale resolution. The method is extremely fast, requiring sub-second times to calculate the radiant heat flux from fires hectares in size at meter scale levels. The method is rapid enough to be used in conjunction with dynamic fire simulations to allow dynamic profiles of radiant heat flux to be determined. In the following sections, the methodology behind radiant heat flux calculation and our computational implementation method are presented. The method is then applied to a simple test case with an analytic solution before being demonstrated in a dynamic context.

2 METHODOLOGY

The radiant heat flux, q , with units Wm^{-2} on a infinitesimal receiver surface at ambient temperature T_a K from a surface at a temperature T K is given by:

$$q = F\sigma\epsilon(T^4 - T_a^4) \quad (1)$$

where F is the view factor from the surface to the infinitesimal receiver, σ is the Stefan-Boltzmann constant ($\approx 5.670 \times 10^{-8} \text{Wm}^{-2}\text{K}^{-1}$) and ϵ is the surface emissivity. The view factor is a geometric term representing the proportion of radiation leaving the emitting surface that strikes the receiver. This term is not straightforward to calculate, although complex closed form mathematical solutions exist for certain common shapes.

A geometric method for calculating the view factor suitable for numerical computation is the projected solid angle method, due to Nusselt (1928). This is shown schematically in Fig. 2a in two dimensions. The view factor is equal to the ratio of the length of the projection of the emitting surface onto a semicircle, a , and the diameter of the semicircle, b , $F = b/a$. This method can be discretised by casting a number of rays with direction vectors \mathbf{r}_i covering equal solid angles θ_i from the receiver location, shown schematically in Fig. 2b. The partial projected length a_i of any rays can be calculated as $a_i = c_i \cos(\theta_i) = c_i(\hat{\mathbf{n}} \cdot \mathbf{r}_i)$. The view factor can then be calculated for all rays where $(\hat{\mathbf{n}} \cdot \mathbf{r}_i) > 0$ using:

$$F = \frac{\sum_{i=1}^N s_i(\hat{\mathbf{n}} \cdot \mathbf{r}_i)}{\sum_{i=1}^N (\hat{\mathbf{n}} \cdot \mathbf{r}_i)} \quad (2)$$

where N is the total number of rays, and the switch function s_i has the value 1 if the emitting surface is intersected and 0 otherwise. Uniform attenuation can be taken into account by including a Lambert attenuation in Eq. (2) as $s_i = e^{-\mu l_i}$, where μ is the attenuation coefficient and l_i is the path length of ray i to the emission surface, or ∞ if the ray does not intersect with an emitting surface. Eq. (2) holds in three dimensions, but

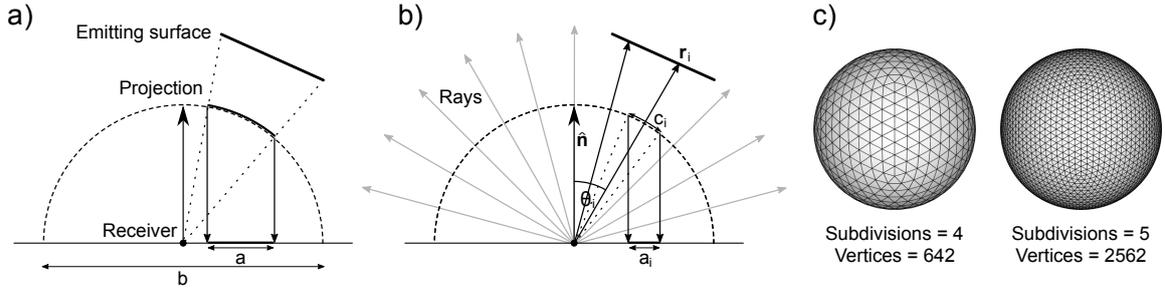


Figure 2. a) Nusselt analogue for calculating view factor (illustrated in two dimensions). The view factor is equal to the segment length a of the projected arc of the projected emitting surface divided by the semicircular diameter b . b) Discretised version of the Nusselt calculation, where a number of rays are cast from the receiver point at equally spaced angles. c) A smoothed geodesic sphere is used for the ray directions in three dimensions as no closed form expression exists for equal partitioning of angles over a sphere.

requires equally angularly spaced rays to be cast over a hemisphere. Unfortunately, the division of a sphere into equally spaced solid angles is an open mathematical problem. In this study an iteratively constructed geodesic sphere with approximated equally spaced vertices was used. An additional smoothing step was applied using a weighted nearest neighbour algorithm to improve the regularity of the spacing. Examples of smoothed geodesic spheres at different levels of iterative subdivision are shown in Fig. 2c. The ray direction vectors were set to the vector from the origin of the sphere to each of the vertices on the surface.

The ray casting method was implemented using a discretised three-dimensional Cartesian grid, the cells of which are called voxels. Voxels are either empty, have an attenuation factor, are solid (obstructions) or are emission sources (flame regions). The ray casting method uses an efficient digital differential analyser (DDA) algorithm by Amanatides *et al.* (1987), shown schematically in Fig. 3. The algorithm calculates all voxels traversed in order by a ray starting at the receiver point. The efficiency of the algorithm lies in the handling of the intersections of the ray with each of the horizontal and vertical cell boundaries. Rather than checking for collisions with every boundary, the algorithm keeps track of the distance along the ray from the starting point. At the initialisation stage of the algorithm, the path lengths between vertical and horizontal cell boundaries, Δ_x and Δ_y , are calculated (red and green bars, respectively, in Fig. 3). The algorithm keeps track of the next cell boundary intersection in each direction and steps between the nearest intersection in each direction. For the example shown in Fig. 3, the distance along the ray is set to the first intersection at point 'A'. From this point, the nearest intersection after taking a step one cell wide in the x-direction is point 'C' and the nearest for a step in the y-direction is point 'B'. The algorithm then jumps to the nearest point in the x or y direction, in this case point 'B'. The nearest distance in the y-direction is then incremented by Δ_y , taking it to point 'E'. The process then repeats, and the nearest point from point 'B' is 'C'. The nearest distance in the x-direction is then incremented by Δ_x , taking it to point 'D'. For the next step, point 'E' in the y-direction is still further away than point 'D' so the algorithm jumps to point 'D'. Finally, the algorithm reaches point 'E' and terminates at the boundary of the domain.

The DDA ray casting algorithm not only gives the correct ordering of voxels, but also calculates the path length through each voxel. The algorithm can also easily be extended to allow variable attenuation coefficients within each voxel as $\exp(-\mu l_i) \approx \exp(-\sum \mu_j \Delta_j)$, where μ_j and Δ_j are the attenuation coefficient and path segment, respectively, for ray i in voxel j . An outline of the algorithm for handling emitting voxels, domain boundaries and obstructing voxels with variable attenuation coefficients is:

1. If the next voxel is outside the domain set $F = 0$ and end.
2. If the next voxel is marked as a solid voxel ($\mu = \infty$) set $F = 0$ and end.
3. If the next voxel is marked as an emitter and end with current value of F .
4. Otherwise, set $F = F_{previous} \times \exp(-\mu \Delta)$, where μ and Δ are the attenuation coefficient and path length in current voxel, respectively.
5. Move to the next voxel.

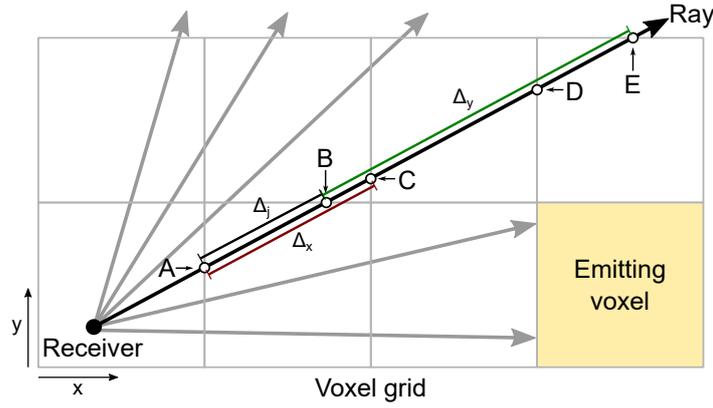


Figure 3. Ray casting DDA algorithm (illustrated in two dimensions). Rays are cast through a Cartesian grid, where cells are called voxels. The algorithm steps through voxels along the path of the ray in order and can identify termination conditions such as an incident emitting or obstructing voxel, or when the domain boundary has been reached.

The ray casting algorithm is highly suited to modern graphical processing unit (GPU) architectures. These are based on single instruction, multiple data (SIMD) designs in which a calculation can be run on many items in parallel. As each ray path can be calculated independently, the multiple rays required for the flux calculation can be calculated entirely in parallel taking full advantage of the GPU architecture. Recent GPU architectures can calculate thousands of items in parallel, providing computational speed-ups orders of magnitude greater than serial computation on a CPU.

The heat flux over a small spherical receiver can also be calculated efficiently in parallel. A spherical evaluation is useful to determine the direction in which most heat is received. This requires evaluation of the radiant heat flux from every direction on a sphere. For evaluations regarding the exposure of a flat surface of a predefined orientation the ray directions \mathbf{r}_i are normalised, and the heat flux can be determined from Eqs. (1, 2):

$$q_j = \sigma \epsilon (T^4 - T_a^4) \frac{\sum_{i=1}^N e^{-\mu l_i} (\mathbf{r}_j \cdot \mathbf{r}_i)}{\sum_{i=1}^N (\mathbf{r}_j \cdot \mathbf{r}_i)} \quad (3)$$

where each direction j can be evaluated in parallel. All GPU implementations within this study used the underlying GPU-based GeoStack module within the Spark framework (<http://research.csiro.au/spark/>).

3 APPLICATIONS

3.1 Analytic comparison

As a first test, the implementation was used to determine the heat flux as a function of distance, d , along the midline from a rectangular wall of flame 25 m high and 100 m wide. The set-up is shown schematically in inset of Fig. 4a. The flame temperature was set to 1200 Kelvin with an ambient temperature of 273.15 Kelvin and a flame emissivity $\epsilon = 0.95$. The plot in Fig. 4a shows values calculated from the computational method (crosses) to an analytic solution (solid line) for the view factor (Hottel, 1931):

$$F_{rectangle} = \frac{2}{\pi} \left[\frac{w}{\sqrt{4d^2 + w^2}} \tan^{-1} \left(\frac{h}{\sqrt{4d^2 + w^2}} \right) + \frac{h}{\sqrt{4d^2 + h^2}} \tan^{-1} \left(\frac{w}{\sqrt{4d^2 + h^2}} \right) \right] \quad (4)$$

where w is the width of the rectangle and h is the height. For this case the heat flux was evaluated in the direction of the emitting surface ($\phi = 0^\circ, \theta = 0^\circ$). As can be seen, the computational results closely match the analytical solution. The convergence to the analytic solution is also of interest for understanding the number of rays required for an accurate estimate of the radiant heat flux. A plot of the calculated radiant heat flux value at $d = 30$ m against number of rays is shown in Fig. 4b, with the analytic value plotted as a horizontal dashed line. A surprisingly small number of rays ($\sim 10^3$) are required for an accuracy of within 3% of the analytic value. It should be noted, however, that this will have a strong dependency on distance on receiver distance.

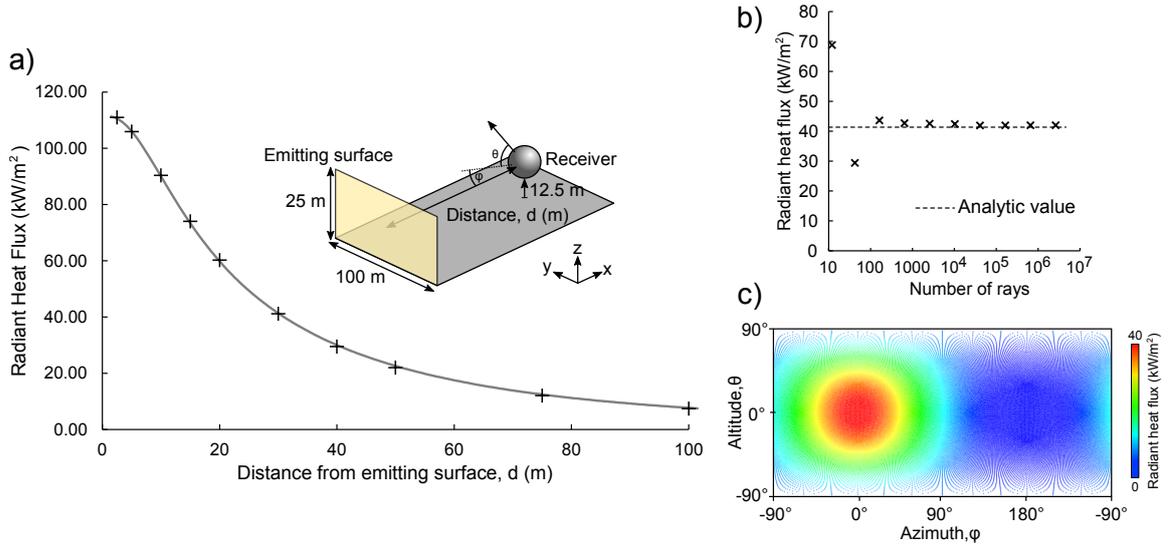


Figure 4. a) Comparison plot of analytic solution for the radiant heat flux against distance for a rectangular emitting surface (solid line) with the ray casting algorithm (crosses). Inset shows the schematic set-up. b) Convergence behaviour of the ray casting algorithm with the number of rays used, where the dashed line shows the analytic value. c) Colour shaded plot of the radiant heat flux value at each direction on a sphere as a function of altitude and azimuth angles.

To visualise the spatial distribution of the heat flux over the surface of a receiver sphere, Eq. (3) was evaluated to find the heat flux at all points on the receiver sphere. A plot of radiant heat flux over the surface as a function of altitude and azimuth angles is shown in Fig. 4c. As expected, the maximum heat flux is in the direction of the emitting surface. As an indication of computational time taken to compute the ray casting and evaluate the heat flux at each point on a sphere the highest number of rays used in the convergence analysis (2,621,442) took 0.11 s to calculate on a NVidia Quadro K6000 over a domain of $100 \times 50 \times 25$ voxels, where each voxel in the simulation was 1 m^3 .

3.2 Dynamic simulation

The method can readily be applied to dynamic scenarios where the emission surface is a region of flame dynamically generated from a wildfire simulation tool. In this study the Spark framework was used to model an example wildfire scenario. Spark is well-suited to using the ray casting method as the underlying level set solver is based on a two-dimensional (x, y) Cartesian grid. This two-dimensional grid can be extended into the third (z) dimension to build voxelised regions representing flame emission surfaces. To create the volumetric flame, the flame height h was dynamically calculated at each cell of the two-dimensional Spark simulation grid, $h[i, j]$, where i, j are the grid cell co-ordinates in the x, y directions, respectively. The initial height of the flame at the leading edge of the fire, h_i , was assumed to be linearly proportional to the rate of spread, r : $h_i = \gamma r$. The flame height was assumed to decay as $h(t) = h_i e^{-\beta t}$, where β is a decay constant and t the length of time the cell has been ignited. The flame height $h[i, j]$ was converted to a three-dimensional binary voxel value $V[i, j, k]$ at cell co-ordinates i, j, k in the x, y, z directions, respectively. This was either marked as an emitting voxel or empty space using the following criteria:

$$V[i, j, k] = \begin{cases} \text{Emitter} & k\Delta z < h \left[i - \left\lfloor \frac{k}{\tan(\alpha)} \right\rfloor, j \right] \\ \text{Empty space} & \text{Otherwise} \end{cases} \quad (5)$$

where Δz is the voxel length in the z -direction, α is the flame tilt angle (in radians).

The set-up is shown schematically in Fig. 1a. Two spherical receivers were used, one within an ‘L’ shaped opaque enclosure (receiver ‘A’) and one outside (receiver ‘B’) in the path of the fire. The receivers were positioned vertically at $z = 10 \text{ m}$. The McArthur (1967) empirical model was used for the rate of spread, r . The parameters used for the McArthur model were wind strength 40 m s^{-1} , air temperature 25° , relative

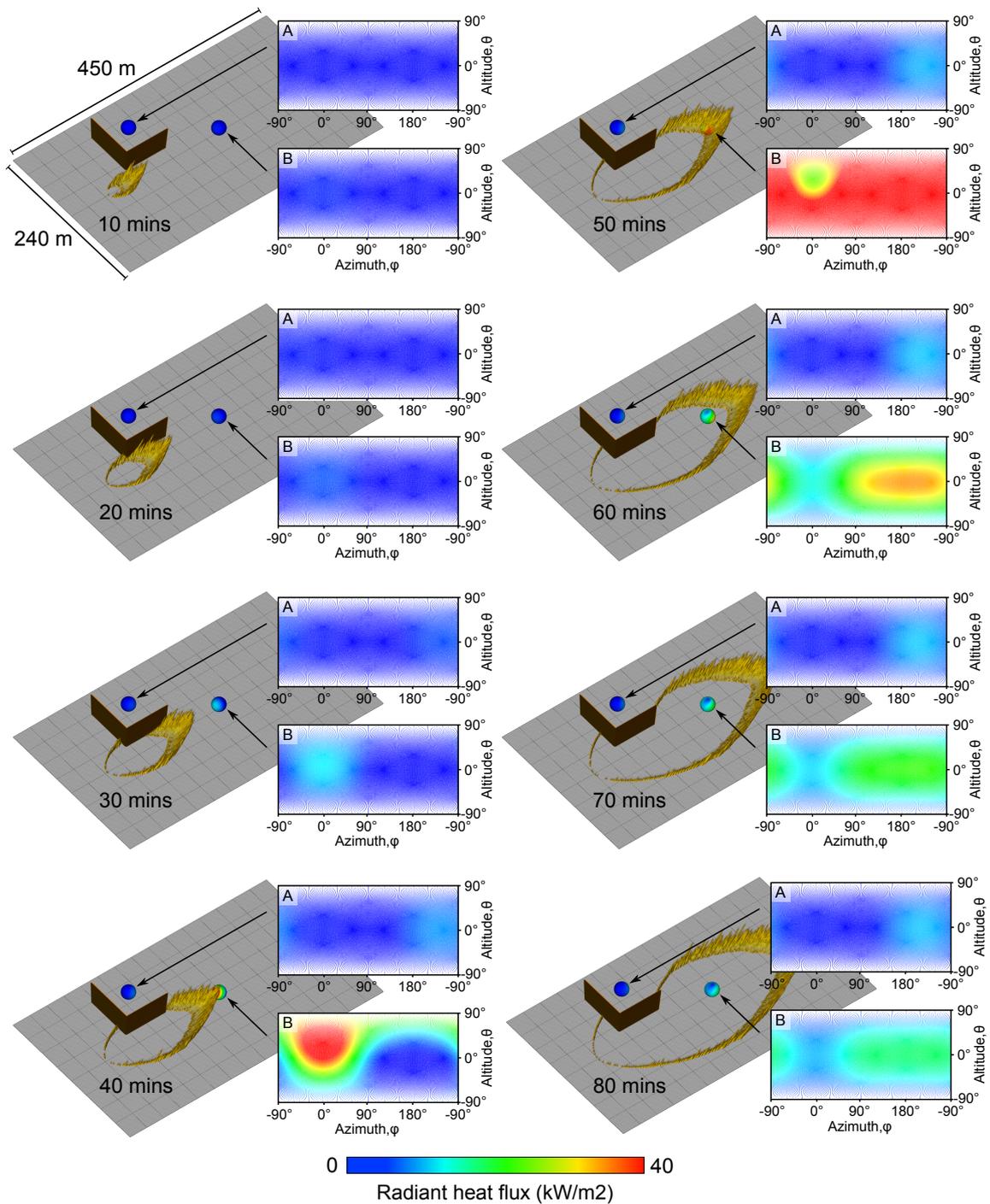


Figure 5. Radiant heat flux on two spherical receivers using flames generated from dynamic fire simulations using Spark with the McArthur rate-of-spread model. Each figure shows a visualisation of the flames and a colour shaded plot of the heat flux on two receivers as a function of altitude and azimuth angles at ten minute intervals. For demonstration receiver 'A' was situated behind an 'L' shaped opaque obstacle of height 40 m, whereas receiver 'B' was situated directly in the path of the fire.

humidity 30%, drought factor 5 and a fuel load of 20 t ha^{-1} . An additional component of the model imposed an elliptical shape on the fire with a backing ratio of 2% of the head fire speed and a backing ratio of 15% of the head fire speed. A random variation in wind direction around the mean was also imposed to include random wind fluctuations (Hilton *et al.*, 2015), where the wind direction was picked at each time step from a normal distribution of width 30° around the mean. The simulation used a cell resolution of 1 m, an circular ignition of radius 5 m and a simulation duration of 80 minutes. The flame temperature was set to 1200 K with an ambient temperature of 273.15 K and a flame emissivity $\epsilon = 0.95$. The flame was angled 60° in the wind direction to take the tilting effect from the wind into account. The constants for the flame height were chosen to be $\gamma = 300$ and $\beta = 0.01$. It should be emphasised that these values were chosen to illustrate the method for this example and any fuel specific values, or even different empirical relations, could be used in place of these.

An isometric visualisation of the test case at ten minute intervals along with maps of the radiant heat flux on the two receivers is shown in Fig. 5. The heat flux was calculated for all points on the spheres from Eq. (3). The spatial variation in heat flux and difference between the two receivers can clearly be seen, especially as the receiver in the path of the flame is engulfed. The maximum value over all points on the sphere for both receivers is shown in Fig. 1b. The radiant heat flux on receiver 'A' only started increasing when the fire passed the open end of the obstruction after approximately $t = 30$ mins. In contrast, the radiant heat flux on receiver 'B' in the path of the flame increased to the maximum possible ($F = 1$) when the receiver was completely engulfed, before decreasing as the fire passed.

4 CONCLUSIONS

New GPU architecture allows ray casting algorithms to be efficiently run in parallel for thousands of individual rays. Simulations using this parallel technique are of sufficient fidelity for radiant heat flux calculations to be calculated at meter scales for infrastructure at risk from fires. The processing time is also fast enough to allow radiant heat fluxes to be calculated for dynamic fire simulation. In this study we presented the methodology and application of this method, validated the results against a known analytic solution and demonstrated application to a dynamic wildfire scenario.

Although not included in the study, the method allows variable spatial attenuation coefficients which could be used to model intervening vegetation and structure. Information such as spatial LIDAR scans could be used to provide such volumetric structural information, and incorporation of such information is currently being implemented. The method could be used to revolutionise current radiant heat flux evaluation methods and to either directly or indirectly support assumption around the use of terrain, vegetation, fencing and other buildings to reduce the radiation reaching building elements.

REFERENCES

- Standards Australia AS 3959-2009. Construction of buildings in Bushfire Prone areas.
- Nusselt W. (1928), Graphische Bestimmung des Winkelverhältnisses bei der Wärmestrahlung, Zeitschrift des Vereines Deutscher Ingenieure, **72**, 673
- Amanatides, J., Woo, A., A Fast Voxel Traversal Algorithm for Ray Tracing, In G. Marechal, Proceedings of EUROGRAPHICS 87, pp 3-9, North-Holland. Elsevier Science Publishers B.V.
- Hottel, H.C. (1931) Radiant heat transmission between surfaces separated by non-absorbing media, Trans. ASME, **53**, 265-273
- McArthur, A.G. (1967), Fire behaviour in eucalypt forest. Commonwealth Department of National Development. Forestry Timber Bureau, Leaflet 107, Canberra, ACT.
- Hilton J.E., Miller C., Sullivan A.L., Rucinski C. (2015) Effects of spatial and temporal variation in environmental conditions on simulation of wildfire spread, *Environmental Modelling and Software*, **67**, 118-127