Abstract
The objective of this paper was to contribute to the development of fast yet precise building energy modelling in tools that are fit to conform to the dynamic processes of the early stage of building design. To this end, a new fast yet precise algorithm for solar heat gain through complex fenestration systems was developed. Documentation provided in this paper suggests that the new algorithm is relevant for any thermal simulation tool that seeks to provide the option of fast and precise hourly thermal simulation of dynamically controlled complex fenestration systems in the early stage of building design.

Introduction
Recent research on identification of user barriers for the adoption of building performance simulation (BPS) tools indicates that architects want to use the output from BPS to inform their decisions in the early design stage (Petersen et al., 2014). However, a crucial criterion for the uptake of BPS tools is minimal time required for modelling and runtime. A recent initiative to obtain this is the tool ICEbear (Lauridsen and Petersen, 2014) which is an implementation of the simple hourly method in EN 13790:2008 but with enhanced capabilities for representing different HVAC systems (Purup and Petersen, 2016). ICEbear currently works in Rhinoseros (2018) and has a runtime of 0.07 seconds for an annual thermal simulation. However, this does not include calculation of solar heat gain which currently is handled by Viper (DIVA-for-Rhino, 2014), a third party solar algorithm based on EnergyPlus rendering a somewhat complicated and time consuming simulation setup for the user. Furthermore, Viper and other available solar algorithms have limited capabilities for handling dynamic complex fenestration systems (CFS) for control of solar heat gains. Various sophisticated thermal tools like EnergyPlus and TRNSYS are able to include a fixed BSDF in thermal simulations using on/off control. However, to the knowledge of the authors, no current thermal algorithms takes into account dynamic control using BSDF. The ability to evaluate the thermal performance of CFS in the early design stage is a crucial issue because CFS have become standard elements in facade design of high performance buildings (Laouadi, 2009). CFS take many forms but a common denominator is that they can be activated, deactivated and adjusted, they have strong sun angle-dependent transmittances, and results in both inward and outward bidirectional scattering of the incoming solar radiation. The most common types of CFS are blinds and roller shades (Konstantoglou and Tsangrassoulis, 2016).

Current BPS tools for the early design stage handles the performance of CFS in different ways. A simple way is to have an algorithm calculate the optimal total solar shading factor between 1 and a user-defined minimum in each time step of the simulation, see e.g. Nielsen (2005). A more sophisticated way is to rely on uni-directional transmittance data, i.e. the solar shading factor of a specific CFS depending on the altitude of the sun as in Hviid et al. (2008). However, the state of the art transmittance data for CFS is in the form of bidirectional scattering distribution function (BSDF). BSDF data is accessible for an increasing number of CFS in the program WINDOW (LBNL, 2008). Research have indicated that the data representation of CFS is of crucial importance to obtain reliable results. TRNSYS results from Hauer et al. (2015) shows that simplified solar transmittance functions of an in-between blind leads to 12-50% difference in annual energy performance compared to simulations using BSDF. Hviid et al. (2008) finds an error of 10-35% in illuminance levels using uni-directional transmittance data to represent a blind compared to RADIANCE simulation, and therefore calls for more accurate characterization of the properties of CFS in BPS tools for the early design stage.

The purpose of this paper is therefore to report on a project that aimed at developing an open source solar heat gain algorithm fit for evaluating the dynamic thermal performance of CFS using BSDF data – bearing in mind that it should be sufficiently fast and precise enough for BPS-based decision support in the early design stage.

Method
The developed solar heat gain algorithm first calculates the different components of total solar energy onto the glazed part of a window, namely direct and sky diffuse solar irradiation, the shading effect of exterior surfaces, radiation reflected from exterior surfaces and ground. Then, the algorithm calculates the solar heat gain to the room or zone using data from WINDOW; simple data for simple glazing systems (SGS), and BSDF for CFS. The methodology of the development was to evaluate the precision and time use of a range of existing methods for calculating the previously mentioned components of the...
total solar energy onto a window and use this as basis for selection and further development of appropriate algorithms for early stage BPS tools. The chosen methods were programmed in C# and the precision of the new algorithms were verified by comparing output with EnergyPlus (EP) simulations using the model depicted in figure 1 and the EP weather data file for Copenhagen, Denmark (Copenhagen 061800 (IWEC)). Computational clock time was always measured on a newly rebooted MacBook Pro with Intel Core i7-3520M CPU running Windows 7. Each simulation is done 10 times to find a mean value, as the computational time can vary. Measurement of computational time was conducted to identify favourable trade-offs between precision and calculation time during the implementation. Simulation time is important if the simulation tool is to be used for generating design advice based on optimization where thousands of simulations must be conducted (Lauridsen and Petersen, 2014).

![Figure 1: Façade and window geometry used for verification of the new algorithm.](image)

Then, two individual algorithms for calculating the total solar heat gain through a SGS and CFS, respectively, was developed and verified by comparing to outputs from EP.

Solar algorithm

Direct solar radiation

Calculating the direct solar radiation on a (tilted) surface, $E_d$ (W/m²), requires the direct normal radiance for a given time and place (e.g. from a weather file) multiplied with cosine of the solar incidence angle of the same time and place. Finding the solar incidence angle requires information on solar position. There are several existing mathematical models for determining the solar position for a given time and place. For example, Grena (2012) provides an overview of the validity period, error (average and range), and relative computational cost of nine solar position algorithms. The algorithm SPA by Reda and Andreas (2004) is considered to be the most reliable with a maximum solar altitude error of 0.0003°. However, its computational time is 10-20 times slower than the eight alternatives.

The simplified tool BuildingCalc (BC) (Nielsen, 2005) uses the solar position method by Scharmer and Greiff (2000) combined with a simple sun declination formula provided by Duffie and Beckman (2013). Compared to SPA, it was found that the error (mean±std.dev.) of the BC algorithm was -0.39±0.32° for the mid-hourly sun altitude over a year. Table 1 shows that the difference in $E_d$ on a vertical surface when using the SPA solar position instead of the original BC solar position algorithm is neglectable. The time for the BC solar position algorithm to calculate the mid-hourly sun positions of a year (8760 hours) was 0.0043 seconds while SPA took 1.2 seconds. It was chosen to implement the original BC $E_d$ algorithm in the new solar algorithm for ICEbear due to a favorable trade-off between precision and calculation time. Table 1 also shows that the hourly difference between the new implementation of the BC solar algorithm and the solar algorithm in EP. The errors here are in general larger; errors are due to minor differences in calculation of local apparent time and equation of time. Consequently, the BC algorithm is more in agreement with SPA than EP.

**Table 1: Difference of mid-hourly $E_d$ over a year on vertical surfaces (Wh/m²). Mean±std.dev. [min;max].**

<table>
<thead>
<tr>
<th>Models</th>
<th>South</th>
<th>North</th>
<th>West</th>
<th>East</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPA minus BC</td>
<td>-0.2±1</td>
<td>0±0.1</td>
<td>0±0.4</td>
<td>0.1±0.4</td>
</tr>
<tr>
<td>New BC minus EP</td>
<td>0.6±11</td>
<td>-0.1±2</td>
<td>0±9</td>
<td>0.3±9</td>
</tr>
</tbody>
</table>

Sky diffuse solar radiation

There are several available methods for calculating the sky diffuse solar radiation on inclined surfaces, $E_s$ (W/m²). A popular and widely acknowledged method used in many BPS tools is provided by Perez et al. (1990). It was therefore merely chosen to implement the model by Perez et al. (1990) as it was done in EP, i.e. using updated circumsolar and horizon brightening coefficients. Table 2 shows that deviations may be up to 100 Wh/m² but the std.dev. indicates that the implementation is insignificantly off for the vast majority of time steps compared to EP. The deviations are probably caused by round-off errors throughout the algorithm.

**Table 2: Difference between mid-hourly diffuse solar radiation over a year on a vertical surface (Wh/m²). Mean±std.dev. [min;max].**

<table>
<thead>
<tr>
<th>Models</th>
<th>South</th>
<th>North</th>
</tr>
</thead>
<tbody>
<tr>
<td>New minus EP</td>
<td>0.2±6 [-100;67]</td>
<td>0.6±5 [-76;47]</td>
</tr>
</tbody>
</table>

Shading effect of external surfaces

This part of the new solar algorithm concerns how external surfaces might reduce direct and diffuse solar radiation on a (tilted) surface; shading of ground reflected solar energy is handled in the next section.

There are several ways of calculating the shading effect. For ICEbear it was chosen to replicate the method implemented in EP as it allows for evaluations of shading from rather arbitrary geometries. Our implementation was validated by calculating the direct and diffuse solar radiation on a south-facing window surface of the geometrical scenarios depicted in Figure 2 and compare them to results from EP.

Table 3 shows that the new implementation and EP renders quite similar results. Deviations are mainly...
because EP multiplies the shading factor by 0.25, 0.5 or 0.75 (depending on the season) if the calculated shading factor is equal to one in the first and last daily hour with sun, respectively.

Figure 2: Left: Simple overhang, Center: Simple overhang and offset, Right: Complex star and offset

Minor differences are also caused by round off errors. The computational time of the new algorithm is approx. 0.8, 1.6 and 3.2 seconds respectively for the three scenarios in Figure 2. This indicates that one generally have to accept a longer computational time when working with complex exterior shading geometries.

Table 3: Difference between new implementation of shading effect and the EP implementation (Wh/m²)

<table>
<thead>
<tr>
<th>Models vs BC</th>
<th>Overhang</th>
<th>Overhang+offset</th>
<th>Star+offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>4±4</td>
<td>4±4</td>
<td>3±2</td>
</tr>
<tr>
<td>BC</td>
<td>[-15;53]</td>
<td>[-13;47]</td>
<td>[-20;39]</td>
</tr>
</tbody>
</table>

It is noted that the new algorithm, just like EnergyPlus, is not able to handle shading effects of concave exterior surfaces. However, the shading effect of a concave exterior surface can be included by dividing the surface into convex subsurfaces.

**Ground reflected solar radiation**

Equation (1) is a general formula for solar radiation on a surface coming from reflections from the ground, $E_g$:

$$E_g = E \cdot \rho \cdot R_e$$  \hspace{1cm} (1)

where $E$ is the global horizontal radiance often provided in weather data files, $\rho$ is the reflectance of the ground (albedo), and $R_e$ is a transposition factor.

A simple approach from here is to assume that $E$ is isotropic (Liu and Jordan, 1962) resulting in:

$$R_e = \frac{1 - \cos(s)}{2}$$  \hspace{1cm} (2)

where $s$ is the tilt angle (degrees) of the receiving surface relative to the horizontal ground.

Assuming that $E$ is isotropic is a coarse assumption as it neglects that the orientation of the receiving surface affect $E_{\text{ground}}$ significantly – especially in terms of the amount of received reflected direct solar radiation. A more sophisticated method is used in EP (called “WithReflections”) where an hourly $R_e$ is determined separately for the direct radiation and for the diffuse radiation (assumed to be uniformly distributed, i.e. not divided into horizon, dome and circumsolar contributions,), respectively, using a one-bounce ray-tracing method. Figure 3 shows the annual ground reflected radiance on surfaces with different inclinations ($\beta$) relative to the horizontal ground.

There is a significant difference between the simple method (equation (2)) and the EP method at all inclinations. The isotropic assumption of the simple method means that especially the northern oriented inclined surfaces recieves an unrealistic amount of ground reflected direct solar radiation. It was therefore chosen to implement a method based on the EP model despite the fact that the simple method – all things being equal – would result in a lower calculation time. A difference in the implementation compared to EP is that for the sake of simplicity it was assumed that the receiving surface is a part of an infinite wide building. Any rays sent out from ground hit-points that does not reach the sun or sky because of the infinite wide building is neglected. However, shadows on the ground generated by the infinite wide building are not included in the calculation of ground reflected $E_g$.

We then investigated how the hemispherical division in the ray trace method would affect results and computational time of $E_g$. Figure 4 shows the annual solar radiance on the reference model facing north from a variation of evenly distributed azimuth divisions ($n=6-90$) for a ray trace hemisphere divided into eight altitude bands of even size (resulting in 48-720 rays). Figure 4 shows that the results deviates less from the mean the more rays that is used and stabilises around a mean value at approx. 70 azimuth divisions (560 rays). The hourly standard deviation for $n=12$ (96 rays) was 2.7 Wh/m², $n=24$ (192 rays) was 1.3 Wh/m², and $n=88$ (704 rays) was 0.9 Wh/m². The computational time was 0.8 seconds for $n=12$ while it was 1.6 seconds for $n=24$. Computational time for $n=88$ was not measured.

Figure 4: Annual radiance as a function of azimuth divisions, $\beta=90^\circ$. Red line: Mean of 70 to 90 divisions.

Figure 5 shows the same as figure 4 but for $\beta=120^\circ$, i.e. the receiving surface is tilted towards the ground. The
distribution of the results is more random than in figure 4, but the deviation still decreases as more rays are used for the calculation.

Figure 5: Annual radiance as a function of azimuth divisions, \(\beta=120^\circ\). Red line: Mean of 70 to 90 divisions.

The hourly standard deviation for \(n=12\) (96 rays) was 4.1 Wh/m², \(n=24\) (192 rays) was 2.0 Wh/m², and \(n=88\) (704 rays) was 0.8 Wh/m². The computational time was 1 second for \(n=12\) while it was 1.9 seconds for \(n=24\). Computational time for \(n=88\) was not measured. It was therefore chosen to use 12 evenly distributed azimuth divisions as it seems to be a favourable trade-off between computational time and precision. Figure 6 shows the results from a similar investigation on the effect of the altitude division using a fixed azimuth division of \(n=12\). The figure indicates that results are not significantly affected by the choice of altitude division (max. 0.8 and 10 kWh/year for \(\beta=90^\circ\) and \(\beta=120^\circ\), respectively) which is why it was chosen to use four divisions ending up with a total of 48 rays.

Figure 6: Annual radiance as a function of altitude division for different tilt angles of receiving surface.

Figure 3 also illustrates that the annual ground reflected radiation of the new algorithm is similar to the EP method. This is supported by the fact that the highest mid-hourly standard deviation over a year is 1.1 Wh/m² (north, \(\beta=120^\circ\)). Adding an opposite building in the calculation of ground reflected radiation (like in figure 7, right) did not increase the magnitude of error; however, calculation time for \(\beta=90^\circ\) was increased from approx. 0.3 to 1.1 seconds.

Reflected solar radiation from exterior surfaces

In the new algorithm, the same principles from the implementation of ground reflected radiation of the new algorithm was applied to represent reflected solar radiation from exterior surfaces onto a receiving surface, \(E_b\) (W/m²). However, the number of altitude divisions for the ray trace method was increased to eight resulting in a total of 96 rays. This leads to an approach which is somewhat different from the approach used in EP. Figure 7 depicts two scenarios used for verification of the implementation by comparing outcome to analogue models in EP. Scenario 1 was a window reveal of 0.3 m in depth, and scenario 2 was a long (infinite) opposite building with a height of 10 m placed 10 m from the building with the window.

Figure 7: Scenarios for verification of \(E_b\). Left: Window reveal (Scen. 1). Right: Opposite building (Scen. 2).

Figure 8 shows the annual \(E_b\) from the window reveal on a receiving surface with different inclinations (\(\beta\)). The relative difference in annual \(E_b\) is high; up to 52 % (\(\beta=120^\circ\)). On hourly basis, the highest standard deviation was 1.4 Wh/m² (\(\beta=120^\circ\)). The high relative difference in annual \(E_b\) is because of the relatively low absolute amount of annual \(E_b\). The annual contribution is therefore relatively sensitive to hourly differences. The hourly differences is probably caused by differences in the number and distribution of rays used in the ray-tracing process, as well as the difference in approach. As illustrated in figure 9, 96 rays may not be sufficient to represent reflections from a small surface close to the receiving surface. Only two rays intersect the single window reveal surface whereas five rays intersect the opposite building; in 3D the difference would be higher. This indicates that one could minimise the difference by making the distribution of rays less coarse. The current computational time for this scenario is approx. 0.4 seconds; however, increasing the number of rays will increase the calculation time significantly.

Figure 8: Annual reflected solar radiance from window reveal on a south-facing surface with inclination, \(\beta\).

Figure 10 shows the annual \(E_b\) from radiance reflected by the opposite building on a receiving surface with different inclinations (\(\beta\)). For \(\beta=90^\circ\), the new algorithm leads to slightly higher annual \(E_b\) than EP. On an hourly basis, the largest standard deviation is 3.6 Wh/m² (east), and the lowest is 0.4 Wh/m² (south). For \(\beta=120^\circ\) and \(\beta=60^\circ\), the new algorithm calculates a somewhat lower \(E_b\).
For $\beta=60^\circ$, the difference seems to be rather sensitive to orientation; for south the annual difference is 65 % while the difference is 11 % for north. On an hourly basis, largest standard deviation was 2.6 Wh/m$^2$ (south).

For $\beta=120^\circ$, the difference due to orientation is somewhat less; the annual difference for west is 35 % while the difference is 17 % for east. On an hourly basis, largest standard deviation was 4.5 Wh/m$^2$ (east). The computational time of this part of the algorithm for the specific scenario was approx. 1 second.

### Total solar radiation

Figure 11 recaps the annual precisions and magnitude of the individual contributions to the total solar radiation on a vertical surface for four orientations. For this purpose the verification model (Figure 1) was used and an opposite building as obstruction (Figure 7, right). The maximum difference is for the east orientation with a total annual difference in radiation of 1.4 % and an hourly standard deviation of 14 Wh/m$^2$.

The total calculation time is approx. 2 seconds hereof 1.1 seconds for the ground reflected part. The total calculation time would be reduced to less than 1 seconds if the opposite building was not a part of the model.

### Summary

Table 4 provides an overview of the various investigations made in the development of the new solar algorithm, the choices made, and the precision of the new algorithm. It was not possible to measure the computational time of the individual part in EP but the new algorithm as a whole has a computational time on par with EP.

<table>
<thead>
<tr>
<th>Part</th>
<th>Investigation</th>
<th>Choice</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dir. solar irradi.</td>
<td>Compare EP and BC to SPA</td>
<td>BC</td>
<td>-0.2±1 Wh/m$^2$</td>
</tr>
<tr>
<td>Sky solar irradi.</td>
<td>Implement Perez model</td>
<td>Perez</td>
<td>0.6±5 Wh/m$^2$</td>
</tr>
<tr>
<td>Shading effect of exterior surfaces</td>
<td>Implement approach similar to EP</td>
<td>EP</td>
<td>4±4 Wh/m$^2$</td>
</tr>
<tr>
<td>Rad. reflected from exterior surfaces</td>
<td>Proposed simplification of EP approach</td>
<td>Simple EP</td>
<td>Std. dev. ±4.1 Wh/m$^2$</td>
</tr>
<tr>
<td>Rad. reflected from ground</td>
<td>Proposed simplification of EP approach</td>
<td>Simple EP</td>
<td>Std. dev. ±2.7 Wh/m$^2$</td>
</tr>
</tbody>
</table>

### Solar heat gain through fenestration systems

The solar heat gain through a fenestration system for a given hour, $Q_s$, can be calculated according to Equation (3):

$$Q_s = A_g (SHGC_{dir} \cdot E_{dir} + SHGC_{dif} \cdot (E_s + E_g + E_b))$$

where $A_g$ is the glazing area (m$^2$), $SHGC_{dir}$ is the solar heat gain coefficient for the direct solar contribution on the glazing (W/m$^2$), and $SHGC_{dif}$ is the solar heat gain coefficient for diffuse solar contribution on the glazing (W/m$^2$).

The new algorithm for calculating solar heat gain through fenestration systems takes offset in Equation (3), and is divided into a part taking care of Simple Glazing System (SGS), i.e. glazings without shading systems, and a part taking care of Complex Fenestration Systems (CFS). Both algorithms make use of data from WINDOW. The data from WINDOW can also be used in EP which gives the advantage of direct comparison of results. Verification is made on the geometry shown in Figure 1. All fenestration data used for verification is from the International Glazing Database (IGDB) updated 14 December 2017 (LBNL, 2017a) and is compiled according to NFRC standards (LBNL, 2017b; 2017c). All verifications of the algorithms makes use of $E_{dir}$, $E_s$, and $E_g$ ($E_b$ is left out) from EP. This eliminates any variations due to differences in the solar radiation algorithms. The new SGS and CFS algorithms are explained and verified in the following sections.
Simple glazing algorithm

For SGS, the solar heat gain coefficient (SHGC) is only dependent of the altitude of the incoming solar radiation – not the azimuth. WINDOW can provide SHGCs for angles of incidence from 0° to 90° with an increment of 10° for any user-defined built-up of glazings. Furthermore, one hemispherical SHGC is given for diffuse solar radiation. The new algorithm runs a linear interpolation on the WINDOW data to determine SHGC$_{dir}$ belonging to the mid-hour solar angle of incidence of each hour in the weather data set with direct solar radiation larger than zero. The corresponding SHGC$_{dif}$ is assumed to be the hemispherical SHGC from the glazing database. $Q_s$ can now be calculated according to Equation (3).

Figure 12 shows that differences in annual total solar heat gain between the new algorithm and EP are small. On an hourly basis, largest standard deviation was 12 W ($\beta$=120°, north). The computational time for the whole year accumulated to approx. 0.05 seconds going from hourly solar irradiation on the outdoor surface to transmitted solar heat gain.

Complex fenestration systems algorithm

The SHGC of a fenestration system consists of a transmission factor (τ$_e$), and a factor expressing the inward emitted fraction of absorbed solar energy ($q_i$) (EN410:2001). WINDOW provides bi-directional transmittance distribution functions (BTDF) for calculation of τ$_e$ but not for $q_i$ as it should be calculated dynamically as a part of the thermal heat transfer through the CFS. A BTDF in WINDOW can be defined by 145 incoming beam directions distributed according to the sky model by Klems (LBNL, 2008). For each incoming direction there are 145 outgoing directions also distributed according to the Klems coordinate system. This leads to a 145x145 matrix of transmittance data.

The task is now to determine the BTDF for any hour with direct solar radiation. It is highly unlikely that the sun angle of any hour perfectly fits a coordinate in the Klems system and thereby a given set of BTDF. Figure 13 illustrates the three principle ways that the incoming direct solar radiation may intersect with the Klems coordinate system, i.e. somewhere between two, three, or four Klems coordinates. Instead, an approximation to a BTDF fitting the solar angle can be calculated as described by Petersen (2011) as a distance-weighted sum of the BTDF coordinates closest to the solar intersection coordinate. This weighted BTDF is then summed to a single value describing the transmittance for the direct solar radiance for this specific hour, τ$_{e,dir}$.

Next, the total transmitted diffuse solar radiation is calculated. The total calculated diffuse solar radiation on the window, $E_{dif}=E_d+E_g+E_b$, is distributed uniformly to the 145 incoming direction of the BTDF. The transmittance factor for $E_{dif}$ (τ$_{e,dif}$) can therefore be calculated according to Equation (4):

$$\tau_{e,dif} = \frac{1}{\pi} \sum_{i=1}^{145} BTDF(\theta_i, \varphi_i) \cdot \cos(\theta_i) \cdot d\Omega_i$$

where $\theta_i$ and $\varphi_i$ is Klems coordinate (altitude and azimuth, respectively) of the BTDF, and $d\Omega_i$ is the solid angle of patch $i$ in the Klems sky model.

Figure 14 shows a comparison of the annual transmitted total energy calculated with the new algorithm and EP. Maximum annual difference is at 70° (15%). On an hourly basis, the highest standard deviation was 9 W (slat angle of 0°). It is noted that compared to EP, the new algorithm in general calculates a slightly higher transmission of direct solar radiation, and a slightly lower transmission of diffuse solar radiation.

Figure 15 shows that the differences in annual transmitted total solar radiation for the slat angle of 10° in figure 14 (hourly standard deviation of 7 W) for different orientations, tilt angles, and opposite building are small. On an hourly basis, largest standard deviation was 14 W ($\beta$=60°, south).

Figure 14 and 15 shows results for certain slat angles that are fixed throughout an entire year.

However, CFS are often meant for dynamic control of solar gains.
Figure 15: Annual transmitted total solar radiation for a venetian blind slat angle of 10° (figure 14) at various orientations, tilt angles and opposite building.

In this case it would be by changing the slat angle of the venetian blind. There are several ways of setting up shading control strategies (Konstantoglou et al., 2013). One is to set the slat angle perpendicular to the sun altitude whenever solar shading is needed. This requires that there is a BTDF for the CFS that matches every possible solar altitude. The new algorithm handles this using BTDF for a few number of slat angles, e.g. from 0-90° in steps of 10°, and then linearly interpolates a new BTDF whenever the solar altitude is not fitting any of the BTDF. Figure 16 shows the transmitted total solar energy through a CFS with slat angle positions perpendicular on the sun altitude for two specific hours of the year (1 January 12.01 pm, solar altitude=11.4°; 5 August 12.01 pm, hour 5195, solar altitude=55.6°) using the BTDF interpolation of the new algorithm and the exact BTDF for the specific slat angle in EP. The difference is low; 13 W and 5 W for 5 August and 1 January, respectively.

Figure 16: Transmitted total solar radiation for a blind with slat angle perpendicular to the solar altitude.

Finally, calculating the inward emitted fraction of absorbed solar energy ($q_i$) is, as stated earlier, usually considered to be a part of the thermal heat transfer calculation as it in principle varies with temperature. However, in an effort to save calculation time, data from WINDOW was used to investigate whether it is reasonable to assume that $q_i$ is constant as stated by Klems and Kelley (1996). Figure 17 shows $q_i$ as a function of temperature difference between indoor and outdoor ($\Delta T$) for a double-layer clear glazing with air gap and external white venetian blinds in various slat angles. Furthermore, $q_i$ at standard temperature setting according to NFRC 100-2010, 32 °C and 24°C, are provided. Here, $q_i$ is rather stable and quite similar to the calculation using standard temperatures.

Figure 17: $q_i$ as a function of $\Delta T$ for a double-layer clear glazing with air gap and external white venetian blinds in various slat angles.

Based on this investigation, it was chosen to include the $q_i$ provided by WINDOW for standard temperature conditions as constant contribution to the total SHGC. Unfortunately it was not possible to compare this part of the new algorithm with outcome from EP as EP does not provide a separate output for $q_i$.

Figure 18 shows $q_i$ as a function of $\Delta T$ and Low-E coating position in a double-layer glazing with air gap and external white venetian blinds in a slat angle of 40°. Here, $q_i$ is also rather stable and rather similar to the calculation using standard temperatures.

Figure 18: $q_i$ as a function of $\Delta T$ and Low-E coating position in a double-layer glazing with air gap and external white venetian blinds in a slat angle of 40°.

The computational time for the new algorithm to go from hourly solar irradiation on the outdoor surface to solar heat gain for the CFS with a fixed slat angle accumulated to approx. 1 seconds for the whole year. Including an obstructing building increased computational time to approx. 2 seconds. It is noted that adding dynamic control of a CFS based on setting the slat angle perpendicular to the sun altitude whenever solar
shading is needed would require additional computational time for interpolation of the BTDF data.

**Conclusion**

In this paper, it has been documented that a newly developed algorithm for calculation of solar heat gain through simple as well as complex fenestration systems is able to calculate reliable hourly solar heat gains for an entire year within few seconds. To this end, a range of simplifications were made. The most critical simplification was that the inward emitted fraction of absorbed solar energy in the fenestration system was set to be constant (i.e. not to vary with temperature differences as would be the case in reality). The results suggests that this simplification seems to be reasonable for the purpose of the algorithm; however, future studies should investigate whether the simplification has any impact on simulated heating and cooling need of a thermal zone. The overall conclusion is that the new algorithm is fast (few seconds) and precise enough (max. std. dev. of 14 W in hourly transmitted total solar radiation) to be included in a thermal building simulation tool fit to conform to the dynamic processes of the early stage of building design.

**Acknowledgement**

We gratefully acknowledge Thorbjørn Basse for his analysis of direct and indirect solar radiation algorithms during his master thesis project.

**References**


Konstantoglou M, A. Kontadakis and A. Tsangrassoulis (2013). Counterbalancing daylighting, glare and view out: the role of an external shading system control strategy. 11th REHVA World Congress CLIMA, Prague, Czech Republic.


LBNL (2017b). About the IGDB - Window Optics, University of California.

LBNL (2017c). How to provide data to the IGDB - Window Optics, University of California.


