Faculteit Technische Natuurwetenschappen

TU Delft

Radiative fluxes in an Urban Heat Canyon

A study with a Monte Carlo method

Naam:	Victor Vertregt		
Studentnummer:	4003969		
Opleiding:	Bachelor Technische Natuurkunde		
Begeleider:	Dr. S.R. de Roode	**Sectie:**	MSP
2de beoordelaar:	Prof. H.J.J. Jonker		
Startdatum:	5 september 2011	**Einddatum:**	27 juni 2012
Is vertrouwelijkheid van toepassing?	Nee		

Abstract

In this Bachelors Thesis, the problem of the Urban Heat Canyon (UHC) has been addressed. The Urban Heat Canyon is a simplified model of an urban area, modelled as a two dimensional model of a street with uniform walls and an uniform street. This model is used to investigate the radiative component of heat balance and absorption, or the ratio between absorbed and incident radiation, of an urban area. The radiation balance in urban areas, modelled with the UHC, is such an interesting topic, because in urban areas, the walls absorb reflected solar radiation and emitted heat radiation from the street, and thus trap radiation. Several papers have been written already on the subject, in which the UHC has been researched with real life measurements, algebraic solutions and models on scale. Here the UHC has been addressed by using the Monte Carlo simulation, in order to simulate the statistics of a large number of photons, the quanta of radiation. The research did focus on finding the influences of the geometry, defined as the height to width ratio of the walls and street, on the absorption and the heat output, and the influence of the zenith angle, on the absorption. The main conclusions were that there is a positive relation between height to width ratio and the absorption, and a negative relation between the radiation output and the height to width ratio. Finally, a negative relation was found between the absorption of an Urban Heat Canyon and the zenith angle for direct radiation.

Contents

1. Introduction

Urban areas usually have a higher temperature than rural areas in the same atmospheric conditions. [Heusinkveld et al., 2010]. The main reason for this is supposed to be the difference in radiation balance, due to the buildings, which absorb reflected or emitted radiation from the street and other walls. Due to this effect, urban areas do not emit or reflect radiation as well as rural areas. This effect is called the Urban Heat Isle. It causes urban areas to cool down relatively slow, and to stay at a relative high temperature at night, especially in hot summers. This causes problems for human health and increases dead rates, especially for elderly people [Heusinkveld et al., 2010]. More knowledge about this subject is therefore import for general public health.

In order to learn more about this behaviour, a model has been invented: the Urban Heat Canyon. Basically, this is nothing more than a 2D street. Imagine a street with buildings on either side, while the sun shines on the whole with an angle θ, see Figure 1. This is a basic model for an Urban Heat Canyon.

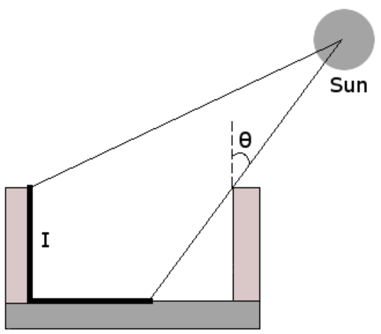

Figure 1 Basic model for a 2D street, with I is the area illuminated by the sun

It is clear at which part of the walls and street the solar radiation hits a surface, and easy to calculate how much radiation would be absorbed. Unfortunately, not all of the radiation is absorbed: some of it, depending on the surface texture and the materials used, is reflected . This reflection is Lambertian reflection, which means that each direction of reflection has an equal probability. The unpredictability of these reflections, makes it a hard problem to solve completely analytical.

The Urban Heat Canyon has been the subject of various papers, each with an own perspective. Harman et al. [2004] took the analytical approach, by considering probabilities up to two reflections, to this problem, which is confirmed by measurements for low buildings and wide streets and near black body materials. Voogt and Oke [1997] did measurements with a special vehicle in real streets in Vancouver. Aida and Gotoh [1982] built a model of a series of Urban Heat Canyons, and did measurements with that.

In this Bachelor Thesis therefore, a different approach will be used to investigate the principle of the Urban Heat Canyon (UHC), namely simulating by using the Monte Carlo method. With this method It is possible to simulate the behaviour of radiation in an UHC. The main questions addressed in this thesis will be: How does the height to width ratio of the canyon influence the absorption, ratio of absorbed to incident radiation, for both direct and indirect radiation, and how does it influence the ability to emit long wave radiation?

The properties of the use of the Monte Carlo method gives this research yet another purpose: if a proper program is made, more difficult problems, other than the standard 2D Urban Heat Canyon, can also be addressed.

In the chapter Theory, the principle of the Urban Heat Canyon and the Monte Carlo method will be explained. In Experimental Setup, the functioning of the various programs that were written for the simulations, will be reviewed. In Results, the results of the simulations will be presented, together with influences of several parameters. In the last chapter, Conclusion & Discussion, several conclusions will be drawn, but also the limitations of this research will be discussed.

2. Theory

2.1. Urban Heat Canyon

An Urban Heat Canyon is a model for the set-up of buildings in an urban area, usually a street with buildings on either side. In order to use this model, there need to be some simplifications. First of all, the model is two dimensional and only one street will be examined. Secondly, the buildings on either side of the street are of equal height and are made of the same material. Thirdly, the walls of the building are made of materials with homogeneous physical properties. Finally, we assume that the sun travels in the same plane as our canyon is in. The model is presented in Figure 2:

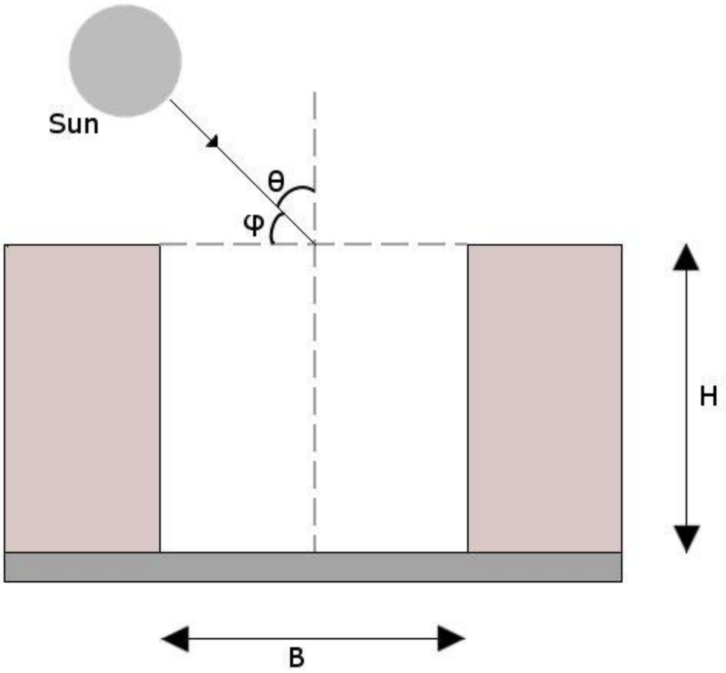

Figure 2 Image of the Urban Heat Canyon, with the different parameters zenith angle *ϑ*, angle *φ*, height *H* and width *B*.

H is the height of the buildings and *B* the width of the street. φ is the angle between the horizontal and the line from the sun to the centre of the top of the canyon, θ is the zenith angle. The parameters on which the thesis focusses the most, are the θ and the geometry, or the height to width ratio of the canyon. The height to width ratio will be abbreviated to the *H/B- ratio* or H_B in this thesis, so:

$$H_B = \frac{H}{B} \tag{1}$$

2.2. Absorption & Emissivity

The absorption A of an object or surface is defined in this thesis as ratio of absorbed radiation, $N_{absorbed}$ from an object to the radiation incident, $N_{\downarrow}$, upon the surface, so:

$$A = \frac{N_{absorbed}}{N_{\downarrow}} \tag{2}$$

The opposite quantiy, the ratio of $N_{reflected}$ and $N_{\downarrow}$ is called the albedo.

$$Albedo = \frac{N_{reflected}}{N_{\downarrow}} \tag{3}$$

The absorption and the albedo are a dimensionless number between zero and one. The absorption is zero for an object which doesn't reflect radiation at all, and one for an object which reflects all radiation. The absorption depends on the material of the object, its shape, the roughness of the surface, the incident angle and the wavelength of the radiation.

How well a certain material absorbs radiation, depends on a few properties: the absorption α and the reflection r are use to describe this phenomena. The absorption α is a number between 0 and 1, this is a material property which also strongly depends on the wavelength λ of the incident radiation. In this thesis α_λ will be used. A material with $\alpha_\lambda = 0$ does not absorb any radiation at all, it only reflects it. A material with $\alpha_\lambda = 1$ absorbs all the incident radiation, and reflects none. A material which has $\alpha_\lambda = 1$ for all λ, is called a black body.

Because a black body absorbs all of the incident long wave radiation, it reflects none. Its reflection r is therefore: $r = 0$. For other materials, r depends, as does the α, on the wavelength of the incident radiation. A material which reflects all the radiation for a certain wavelength, has $r_\lambda = 1$. So the relation between r_λ and α_λ is as follows:

$$\alpha_\lambda + r_\lambda = 1 \tag{4}$$

The wavelength and amount of emission of radiation of a material depends on the temperature T of the material, and its emissivity ε. The radiated power over all wavelengths depends on the Stefan-Boltzmann law:

$$P = \varepsilon \sigma A T^4 \tag{5}$$

P is the emitted power, σ is the Stefan-Boltzmann constant, A is the surface area from which the radiation is emitted and T the temperature of the emitting surface. The ε is called the emissivity, with a value between 1 and 0. The emissivity ε is linked with the absorption α, by Kirchhoff's Law '

$$\alpha_\lambda = \varepsilon_\lambda \tag{6}$$

A black body has ε=1, and is therefore a perfect emitter.

For the wavelength of the emitted radiation of a black body, there is Planck's law:

$$E_\lambda(T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1} \tag{7}$$

E_λ is the energy emitted in a certain wavelength, *h* is Planck's constant, *c* the speed of light, and k_B Boltzmann's constant. What is clear, is that the amount of energy emitted in a certain part of the wavelength spectrum depends on the temperature. Especially, the wavelength at which the intensity peaks is reciprocally proportional with the temperature of the emitter. This is neatly formulated in Wien's law:

$$\lambda_{\max} \cdot T = b \tag{8}$$

In this formula, *T* is the temperature of the emitter, λ_{max} is the wavelength at which the most energy is emitted and b is Wien's constant.

The important conclusion for this thesis is, that the temperature of an object influences the wavelength of the emitted radiation. Therefore the wavelength of the solar radiation and heat radiation from buildings is not the same, solar radiation is emitted in the UV and visible-light spectrum, heat radiation from buildings is infrared radiation. This is important, because also the absorption α and reflection *r* depend on this wavelength. So buildings and streets can have a very different absorption coefficient for the two different kinds of radiation.

Finally, if a surface has an r>0, it reflects part of the incident radiation. It depends on the roughness of the surface in which way the radiation will be reflected. On smooth, shiny surfaces, radiation will be reflected specular. This means that the angles of incidence and reflection with the normal, are equal. If the surface is rough, reflection will be diffuse. Walls and streets are often rough surfaces, windows and glass buildings excluded, on which incident radiation experiences diffuse reflection. Thus every photon incident on a wall or street has a certain chance of being absorbed and, when reflected a random angle of reflection. [Wolfson, 2007]. In this thesis, it will be assumed that this reflection is Lambertian: the angle of reflection is uniformly distributed, so every angle of reflection has an equal chance.

2.3. Monte Carlo

The Monte Carlo method is a simulation principle, which is very useful if probabilities are involved in a problem. The basic concept is the following: by running the simulation a large amount of times with the same parameter values and begin values and accumulating the results of all the simulations, one gets a probability density function of the outcome. Each time probabilities are involved, a random outcome is drawn. This way, there are no elaborate calculations needed with probabilities, but relative easy ones, with random outcomes. In this research, the Monte Carlo simulation will be used to simulate individual photons, and for directions of reflection, random values will be drawn. So the exact behaviour of individual photons in an UHC cannot be determined by this method, but the overall probability density can. In this research, the Monte Carlo method is a very useful tool, because of the behaviour of incident photons. By simulating the incidence of a photon over and over again, an approximation is made for the real life behaviour of radiation in an Urban Heat Canyon.

3. Experimental Method

As explained in the previous chapter, using the Monte Carlo method makes it possible to simulate the behaviour of photons, which are the quanta of radiation, in the Urban Heat Canyon. In order to this, a program was built that consisted of three main scripts. The first one simulates the behaviour of a single photon entering the canyon. The second script uses the first script to simulate not one, but a significant amount of photons with the same start values and keeps account of the position where they are finally absorbed. Finally, there are several independent scripts which repeat the second script while varying parameters such as the zenith angle θ or the H\B-ratio H_B.

In this chapter the functioning of the several scripts and programs will be explained. All of the programming was done in Matlab R2009b-SP1, the scripts are in the Appendices.

3.1. Path of the photons – Script I

In order to simulate the path of the incident photons, two important assumptions were made, according to the theory explained in the previous chapter:

1. If a photon hits a wall or street, it will be absorbed with a chance α or reflected with a chance r= 1-α. The α and r depend on whether the photon hit a street or a wall, α_S is used in case of a street, α_W in case of a wall.
2. If a photon is reflected, the angle of reflection is uniformly distributed and does not depend on the angle of incidence.

The main idea of this script is the following: it simulates the canyon as if it were a 2D box, with a photon somewhere on the edge. The height and width of the box are H and B. From the photon, three properties are known: its initial location is in Cartesian coordinates x and y, and the angle φ, see Figure 3:

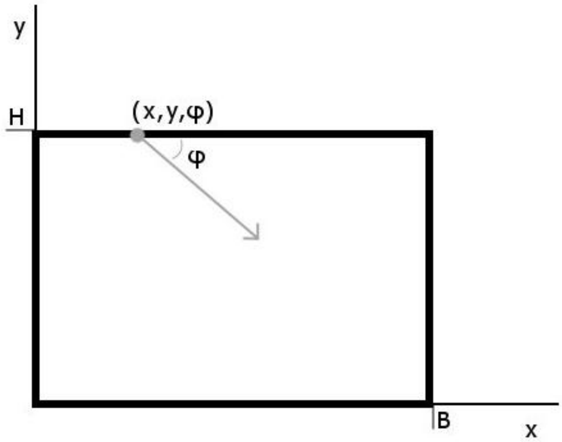

Figure 3 Simple representation of Urban Heat Canyon, as it is seen by the program: a box with four edges, and a moving photon with its initial position, in this case *x*, *y*=*H* and direction *φ*.

The script runs the same loop over and over, until the photon is absorbed or has escaped into the atmosphere. This loop consists out of the following steps:

1. The script starts with the position and the angle. In the first loop, these are the initial values. In the consecutive loops, these are the results of the previous loop. How the start values for the position and φ are determined, will be explained in the section 3.3.
2. After that, the next position must be determined where the photon will incident an edge of the box from Figure 3, so a street, wall, or the exit to the atmosphere. This can simply be calculated using geometry with the x, y and φ and the H and B of the box.
3. In case the photon exits the canyon, the script is stopped. In case of incidence at a street or wall, it needs to be determined whether the photon is absorbed or reflected. This is done by simulating a random number q between 0 and 1 and comparing that with r:

$$q = U[0,1] \rightarrow \begin{cases} q \le r & \rightarrow \text{ reflection} \\ q > r & \rightarrow \text{ absorption} \end{cases} \tag{9}$$

   In case of absorption, the script is stopped.

4. In case of reflection, a new angle φ is determined by again simulating a random number between -90° and 90° for the walls, and 0° and 180° for the street. After this, the loop starts again until the photon has been absorbed or escaped into the atmosphere.

The program records every loop by saving the end position after each loop, which is the begin position for the next loop, as a line in a matrix F. The first line in F are the initial values. The first column of this matrix are the x coordinates, the second are the y coordinates and the third the angles φ. Finally, there is a fourth column, called q, which can have three values: 0, 1 and 2. If this is 0, it means the photon has been absorbed, so the program has stopped. If it 2, it means the photon has escaped the canyon, so the program again was stopped. If it is 1, it means the photon hasn't been absorbed or is escaped, so the program continues to run the same loop. By programming it like this, it is very simple to find out later on what eventually happened to a certain photon. An imaginary example of matrix F would be:

x	y	ϕ	q
0.2	1	30	1
1	0.54	-60	1
0.69	0	-	0

Figure 4 Example of matrix *F* with *B*=*H*=1.

In this way, adapting the script for different situations is simply changing the H, B or α's, or imposing different conditions on the start position and direction of the photons.

3.2. Accumulating the results – script II

Of course the behaviour of one photon is completely random, and is it necessary to simulate the path of thousands of photons, before conclusions can be drawn. This is where the second part of the program comes in: to use the results for single photons in order to obtain relevant statistics

The functioning of this script is relatively simple. First the values of the height H, width B, and absorption of the street and walls, α_S and α_W are stored. After that, the script needs to have the start values of the photons. This can be exactly the same for each photon, but variations in position or direction are also possible. How these are to be chosen for different problems, is the subject of the next section.

With the information of the start value, Script II uses the Script I to simulate *N* photons with the same starting value. Not all the information from the first script is useful, only the last row of *F* contains important information, because the index *q* shows whether the photon was absorbed or escaped the canyon, and the *x* and *y* are the position of absorption. By storing these three variables as a row in a new matrix G, with *N* columns for the *N* photons, all the necessary information is in the same matrix, which is the result of script II. This matrix G contains thus for each of the N simulated photons just whether they were absorbed or escaped and where they were absorbed or where they escaped. An example:

x	y	q
0.5	1	2
0.8	0	0
0.6	0	0
0		0
0.31	1	2
0.3	1	2
0.2	0	0

Figure 5 An example of matrix *F*, *H*=*B*=1.

From this matrix G the absorption *A,* as defined in Formula (2), can be easily determined:

$$A = \frac{N(q=0)}{N_{total}} \tag{10}$$

For this simple case, the absorption would be *A*=4/7=0.57.

In order to use these results further, it is useful to sort the place of absorption not by photon, but by position of absorption. For this purpose, there is a second matrix, *W*, with *H·m* rows and *B·m* columns, where *m* is a natural number. The value in the elements of this matrix are the number of photons that were absorbed at that position. In the end, only the elements in the first and last column and the last row, corresponding to the walls and street in the canyon, will get a value not equal to zero. A simple example of this matrix W would be:

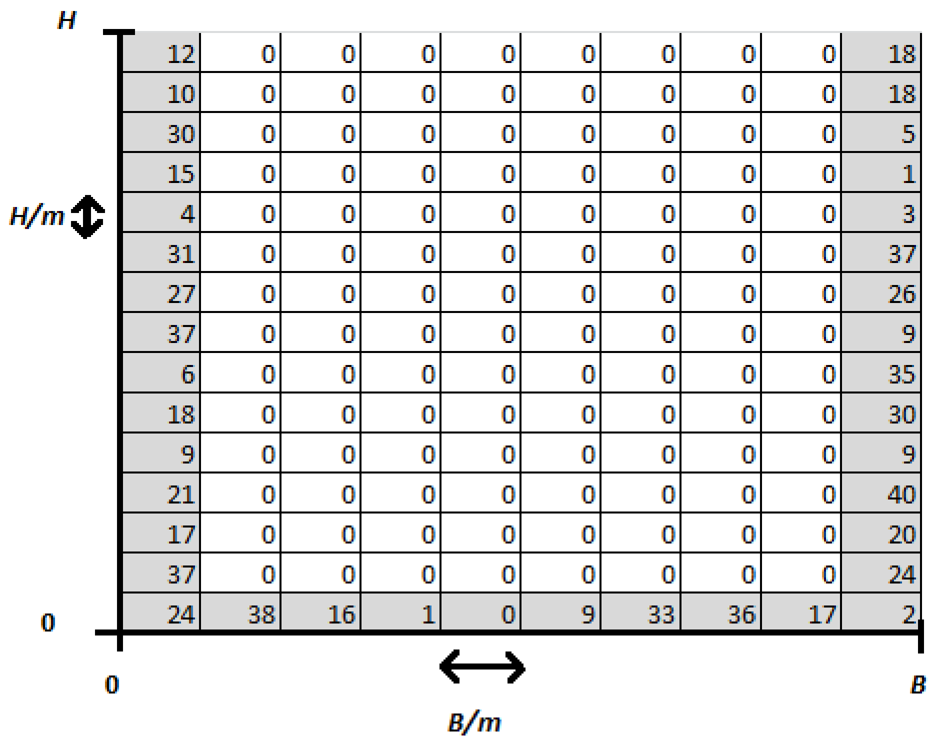

Figure 6 Visual representation of matrix W. The numbers in the elements indicate the number of photons absorbed in that area, with an length of *B/m* on the street, and *H/m* on the walls.

Script II sorts the absorbed photons in the following way to obtain the matrix W; it multiplies the x and y coordinates of the photons with m, then rounds them to round numbers. The coordinates of the absorbed photons can now serve as indices for the matrix W. In the end, the elements corresponding with the top of the canyon are set to zero.

In this way, the number in the elements of the first and last column and the last row of W correspond to the number of photons that where absorbed in that place. By making a 3D-plot from this matrix, it is possible to see where the most of the photons landed, and thus, most of the radiation was absorbed.

Also from matrix W, it is easy to subtract the absorption: by summing up all of the element in W, and dividing it by the total number of simulated photons, gives the absorption.

3.3. Projects

With the scripts described above, it is possible to research the problems described in the Introduction. For each new research question, a new script was built. First, the direct radiation from the sun was simulated, with varying zenith angle θ and H/B-ratio H_B. After that, the diffuse radiation from the atmosphere was simulated with varying H/B-ratio, and finally the radiation emitted from the walls and street of the canyon itself was simulated, in order to figure out the cooling down situation at night.

3.3.1. Direct radiation

In order to examine the behaviour of sunlight in an UHC, a beam of incoming photons will be simulated first. The main idea is to simulate for several H/B-ratios an incoming beam of photons, with a zenith angle θ from 0° to 90°, and calculate the absorption *A* for each situation by using formula (2). By making a plot of absorption versus the zenith angle θ for the various H/B ratio's H_B, the effect of the H/B-ratio and zenith angle can be evaluated simultaneously.

The script that was written to do this, uses a for-loop to repeat script II once for every combination of the zenith angle θ and the H/B-ratio, H_B. For the incident angle, the values θ=1,2,3..90° were taken, and for H/B-ratio H_B seven values between 0.1 and 10. Start values for the photons were $\varphi=\vartheta$, $y = H$ and x is a uniformly distributed number between zero and B, each combination of the angle ϑ and the H/B-ratio was simulated 10000 times, so N=10000.

This simulation did not take into account, that the amount of radiation which enters the canyon, is much less for higher values of θ. In order to take this behaviour in account, the number of simulated photons is dependent on the incident angle. In order to take this phenomenon in account, the number of simulated photons should be: $N(\theta)=N(0)\cos(\theta)$, where $N(0) = 10000$.

3.3.2. Diffuse radiation

Not all the light from the sun reaches the earth directly. A part of the radiation is reflected in the atmosphere. This causes the diffuse radiation that enters the Urban Heat Canyon. Diffuse radiation has a random direction of propagation. In order to simulate this kind of radiation, it is necessary to know the distribution of direction from which the diffuse photons enter the UHC. In this thesis, it was assumed that this diffuse radiation has a uniform distribution. From every angle, an even number of photons should enter the canyon. In order to simulate this, for every angle φ, N=100000 photons were simulated with a uniform distributed x-coordinate. This was done for several H_B-ratio's.

3.3.3. Heat output

Almost as important as how the Urban Heat Canyon absorbs radiation, is how it emits it. In order to simulate the heat output of an Urban Street Canyon , again script I and II are used with varying start values. In this case, $m\cdot H$ sources are evenly distributed on a wall of the canyon, and $m\cdot B$ sources on the street. From each of these sources, N photons with a random direction of propagations are simulated. This process was repeated for several H/B-ratios, so that the influences of the H/B ratio on the heat output of a canyon could be investigated.

4. Results

During all the simulations, the absorption coefficients of the walls and street have been α_S = 0.8 and α_w = *0.7*. These values are typical for grey cement concrete, which has an absorption between 0.6 and 0.8, higher values when weathered and darker [Levinson and Akbari, 2001].

4.1. Convergence of the total absorption

First of all, the convergence of the solutions was tested. Because the important results are often in terms of the absorption, defined in formula (2), the number of absorbed photons divided by the number of simulated photons, this is used to determine the convergence. In order to the determine the absorption, one particular situation was examined, with the number of photons, N, simulated varying from 1 to 100000 in steps of 100. In each simulation, the absorption was determined, as well as the absolute difference between the absorption from the simulation before. The situation was: direct radiation, θ=45°, H_B=0.5. The result is plotted in Figure 7.

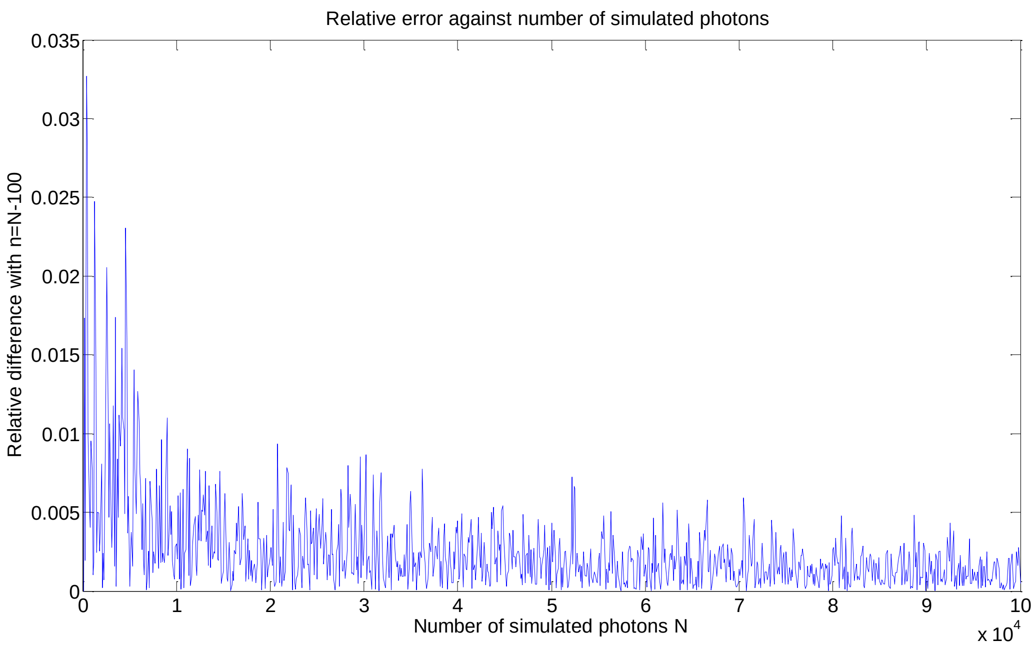

Figure 7 Relative difference between the measurements of the absorption

It is clear when more photons are simulated, the absorption has a smaller standard error. The most simulations in this thesis were done with N=1000, so with a standard error of ±1%. In these simulations, the absorption was the required output.

4.2. Direct radiation

In order to show the results of the calculations in a complete and structured way, the results will be presented as follows: first one photon, to illustrate the random behaviour. Secondly, the case of a few thousand photons will be presented, all with the same zenith angle θ and start conditions. At last, the results will be expanded to a varying θ and H_B.

First, the illustrating case of 1 photon. The zenith angle θ is 45° and the H_B is 0.5. As start positions of the incoming photons, the *x*-coordinate is a random number between 0 and the width of the canyon *B*, and the *y*-coordinate is the height *H*. The begin angle φ is just θ.

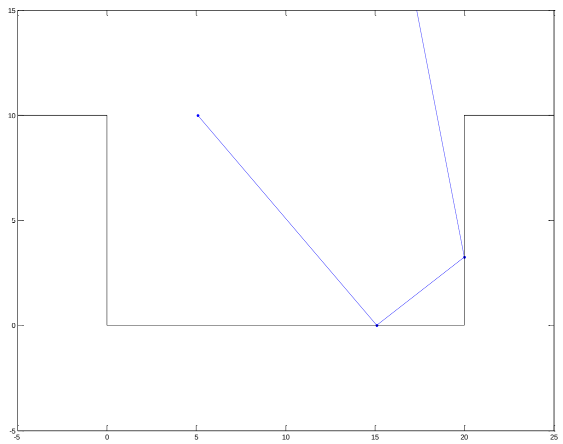

Figure 8 Path of a random photon, θ=45°, H_B= 0.5, α_S = 0.8 and α_w = 0.7.

This calculation was repeated *N* times, with *N*=100.000, in order to find a pattern in the behaviour of the photons, this time with H_B=1, in order to examine the effect of the UHC. The start position was the same for each photon, so the same y-coordinate and φ, but all with a random distributed x-coordinate between 0 and the *B*.

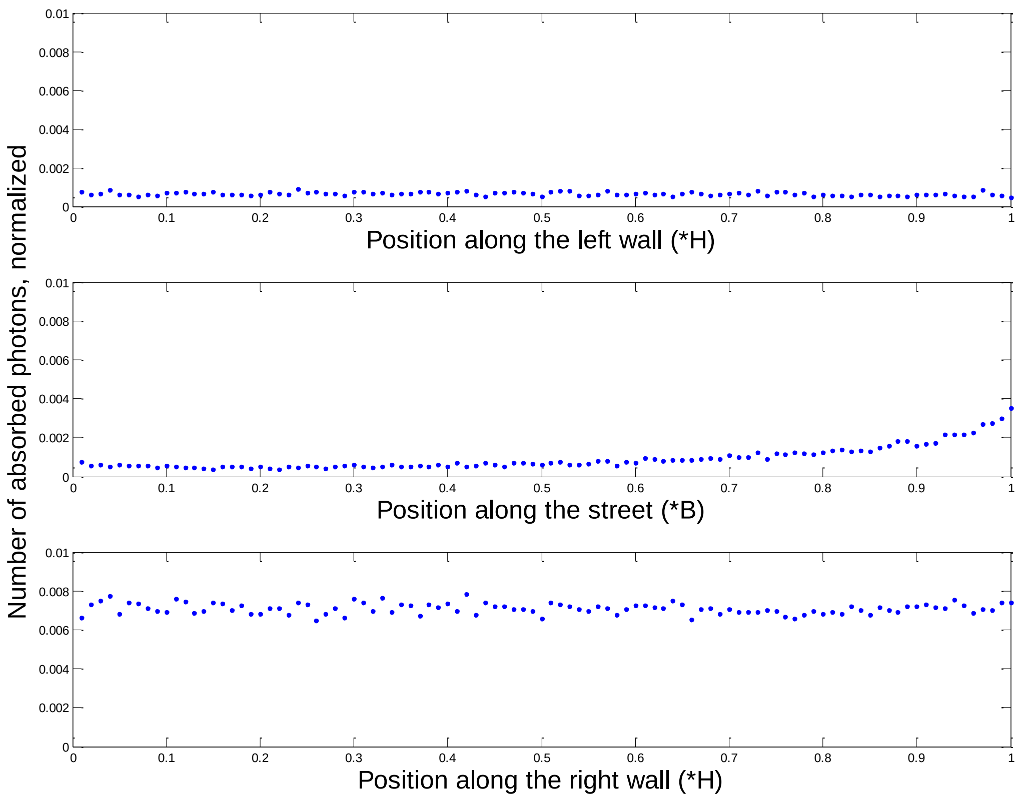

Figure 9 Location of absorption for direct radiation, θ=45°, H_B = 1

From Figure 9, it is clear that most photons are absorbed at the spot where they first incident, only few are reflected. As a result, areas which are directly lit, absorb far more radiation than areas which are not. However, also a significant number of photons are eventually absorbed at a position, that is not directly lit, such as the opposite wall or street. The absorption of these reflected photons, are the reason of the UHC effect.

In order to say something about the influence of the zenith angle and the geometry, only the absorption will be evaluated in the next part. In Figure 10 below, the absorption versus θ has been plotted, for several H_B. Each simulation of a combination of H_B and ϑ counted 100.000 photons, so N=100.000 and m=100. The absorption, following (2), is $N_{absorbed}/N_{\downarrow}$.

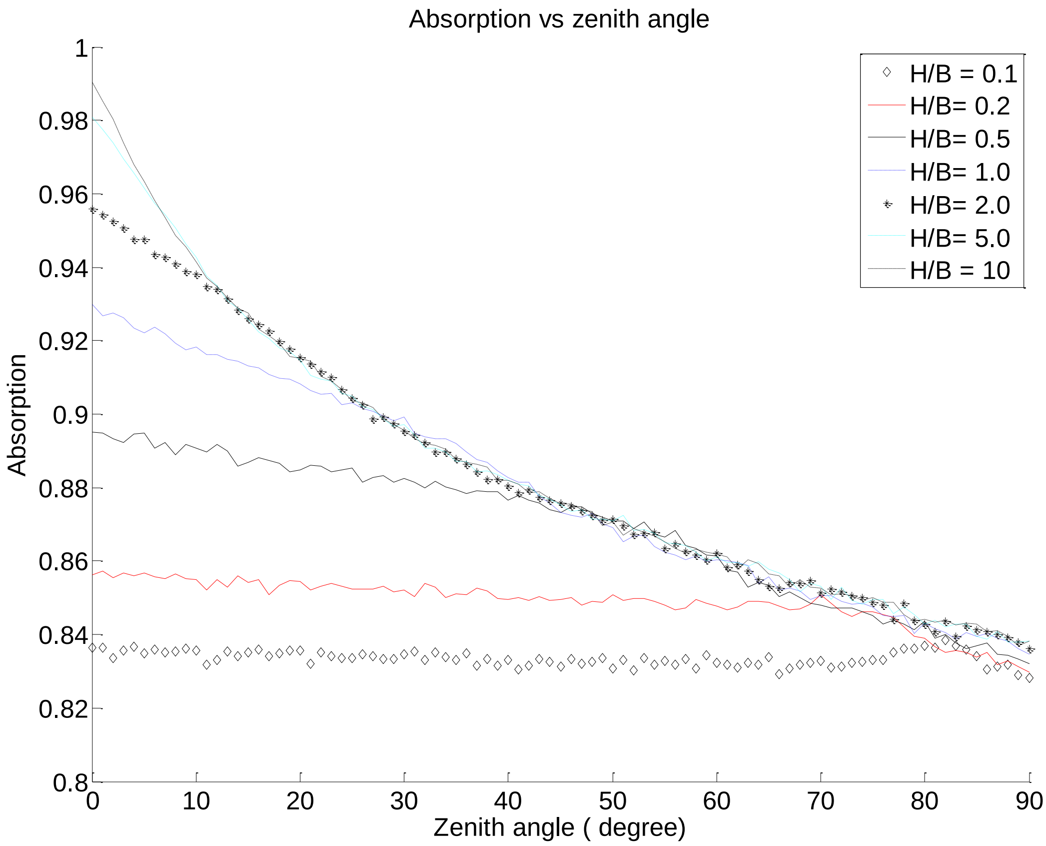

Figure 10 Absorption plotted against the zenith angle θ, for H_B = 0.1 ,0.2 ,0.5 ,1 ,2, 5, 10.

From the graphs in Figure 10 three things become clear: first, there is a positive relation between the absorption and the H_B of an Urban Heat Canyon, and second: how larger the H_B of a canyon, the larger the influence of the θ on the absorption. Finally it is clear that there is a negative relation between the absolute value of θ, |θ|, and the absorption.

These observations are quite intuitive: a positive relation between the absorption and the H_B is what one would expect: the "deeper" the canyon, the harder it is for incoming radiation to escape towards the atmosphere, and the more photons will be trapped. The second and third observation are also quite predictable: if |θ|≈90°, the radiation just hits the top of the walls of the canyon. The tops of the walls are the same for each H_B , so the absorption should be the same. When |θ|≈0°, the radiation hits the bottom of the canyon. In canyons with a high H_B, escape is not very likely, so the total number of absorbed photons and the absorption are relatively high. In canyons with a low H_B, this is easier, so the absorption is lower.

As explained in the Experimental Setup, it could also be useful to compare the normalized absolute amount of radiation that is absorbed in the Urban Heat Canyon. In the plot below, Figure 11, the normalized number of photons is the amount that enters the Urban Heat Canyon at θ=0°. The data is the same as used for Figure 10.

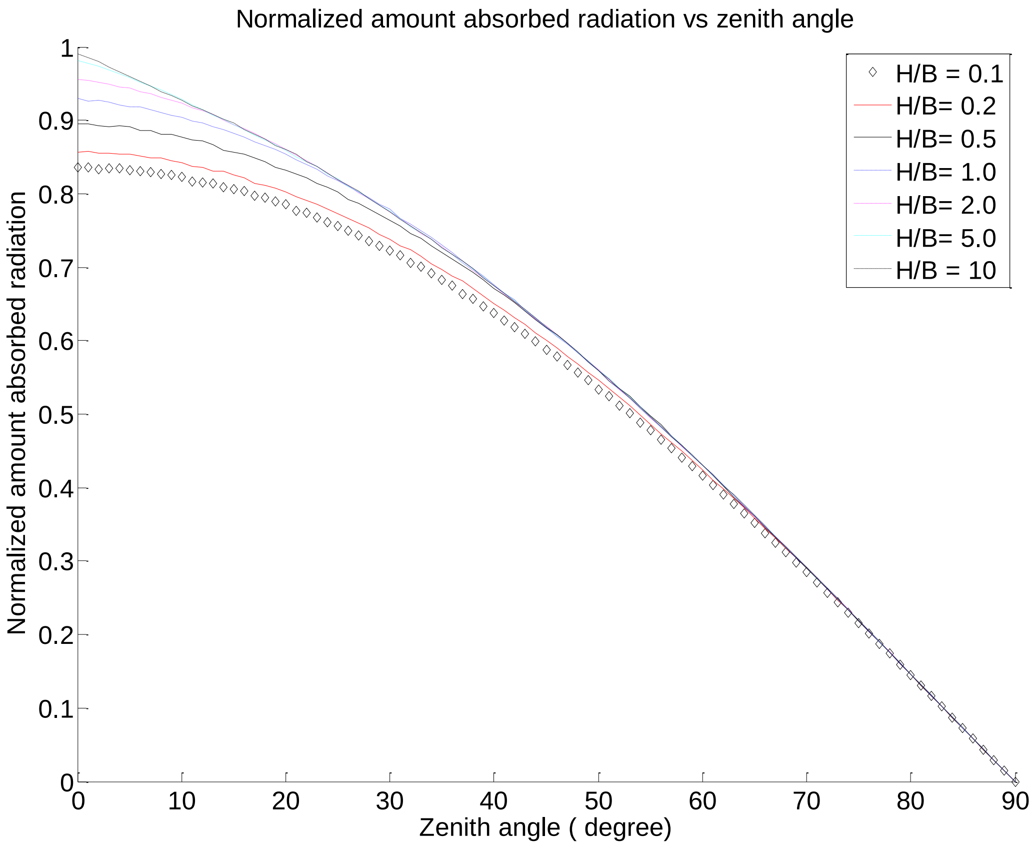

Figure 11 Number of absorbed photons, normalized with the total entering photons for ϑ=0, plotted against the zenith angle ϑ, for H_B = 0.1 ,0.2 ,0.5 ,1 ,2, 5, 10.

From the graphs in Figure 11, it is clear to see that the amount of radiation, which is absorbed by an Urban Heat Canyon, depends mainly on θ for higher |θ|. The influence of the H_B is only noticeable when the zenith angle θ is relatively close to 0.

4.3. Diffuse radiation

The effect of the H_B on the absorption, as defined in Formula (2), in case of diffuse radiation, was also simulated. In Figure 12, the absorption has been plotted against the H_B, for seven values of H_B.

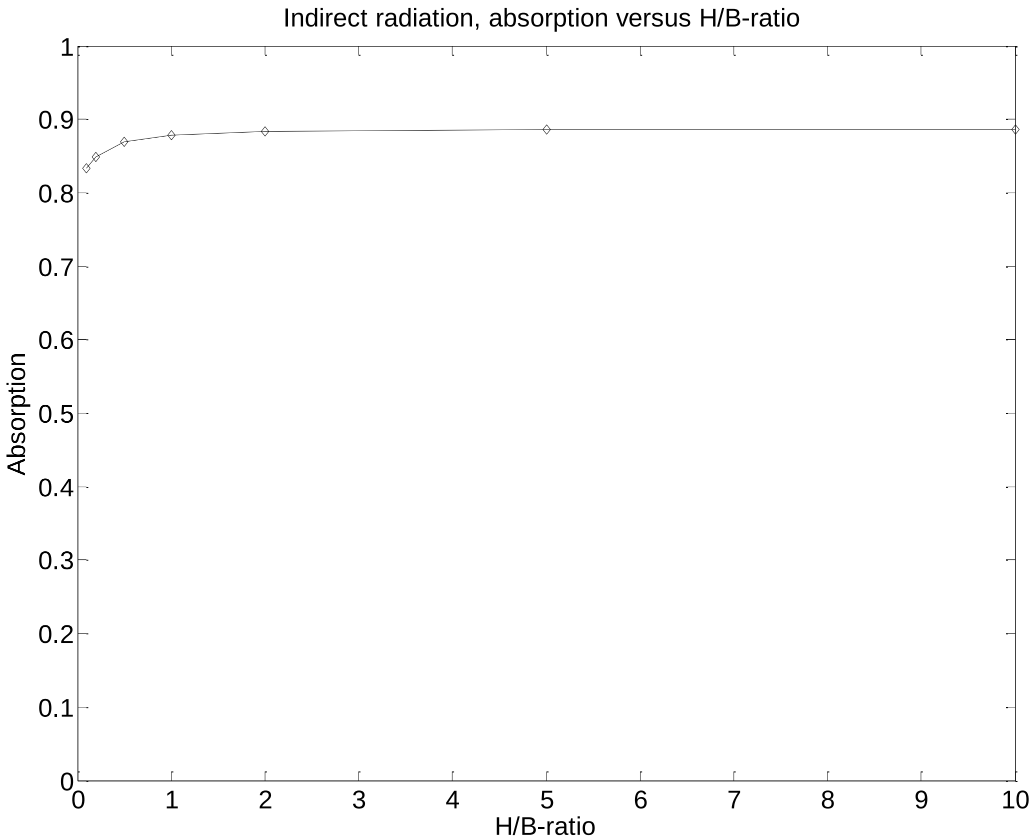

Figure 12 Plot of the absorption versus the H_B in the case of indirect radiation, which is simulated as uniform distributed over all possible angles of incident.

What is interesting in this plot, is that there is an relation between H_B and the absorption in the case of diffuse radiation. For $H_B \leq 1$, the absorption becomes larger when the H_B becomes larger. For $H_B > 1$, this characteristics reaches a limit.

4.4. Heat output

Finally, it was simulated how Urban Heat Canyons cool down by emitting radiation. As explained in the Experimental Setup, alongside the walls and street of the canyon, respectively *H·m* and *B·m* imaginary radiation sources are positioned. From each of the sources *N* photons are uniformly emitted in every direction. First, a plot for one such source, positioned at *y*=0. and *x*=0.3*B*, so on the street. H_B is 0.5, N=10000. A plot of the position of absorption of the *N* photons is in Figure 13.

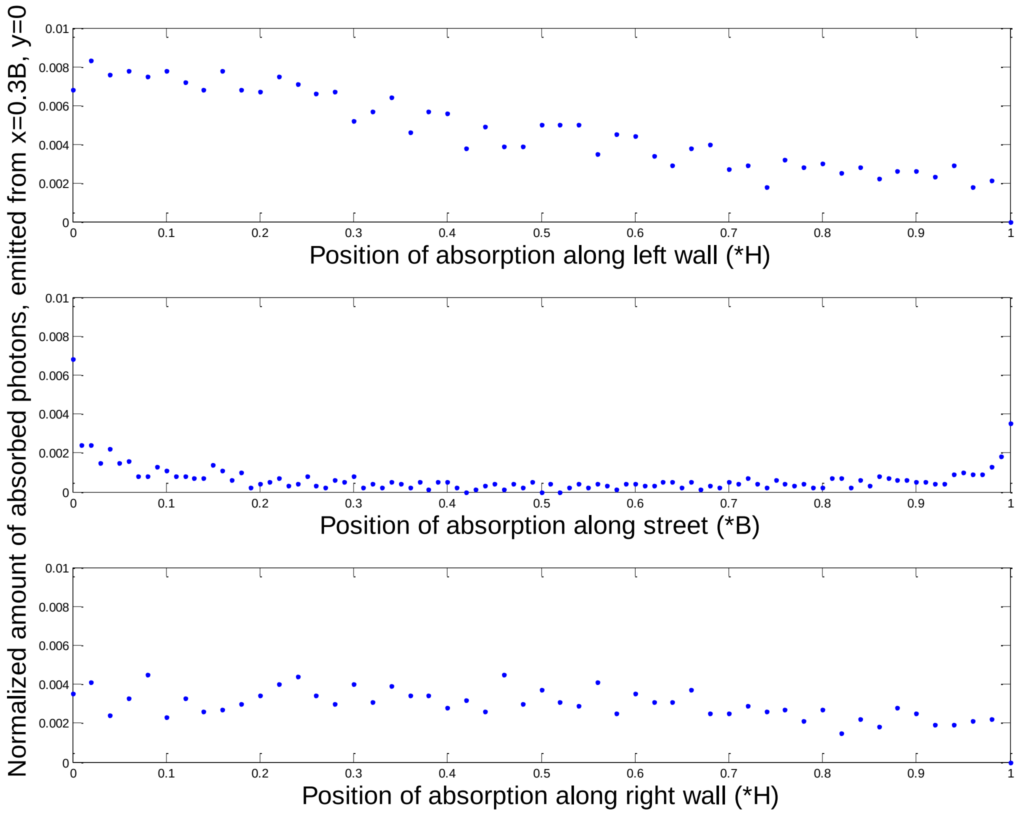

Figure 13 Plot of the heat output of one source at the street, x=0.3B and y=0. H_B =0.5, N=10000

The plots of Figure 13 illustrate very well the effect of the UHC on the emission of radiation. If there would have been no walls next to this street, all the radiation would have been emitted to the atmosphere. But in this case, 45% of all the radiation was again absorbed, by the walls or, after reflection, the street itself.

In order to look at the behaviour for all these sources, again not the location of absorption, but the amount of radiated photons which is eventually absorbed is evaluated.

In Figure 14 and Figure 15 it can be seen what part of emitted photons is eventually absorbed in the Urban Heat Canyon and what part eventually escapes. This is plotted against their normalized start position, for several H_B.

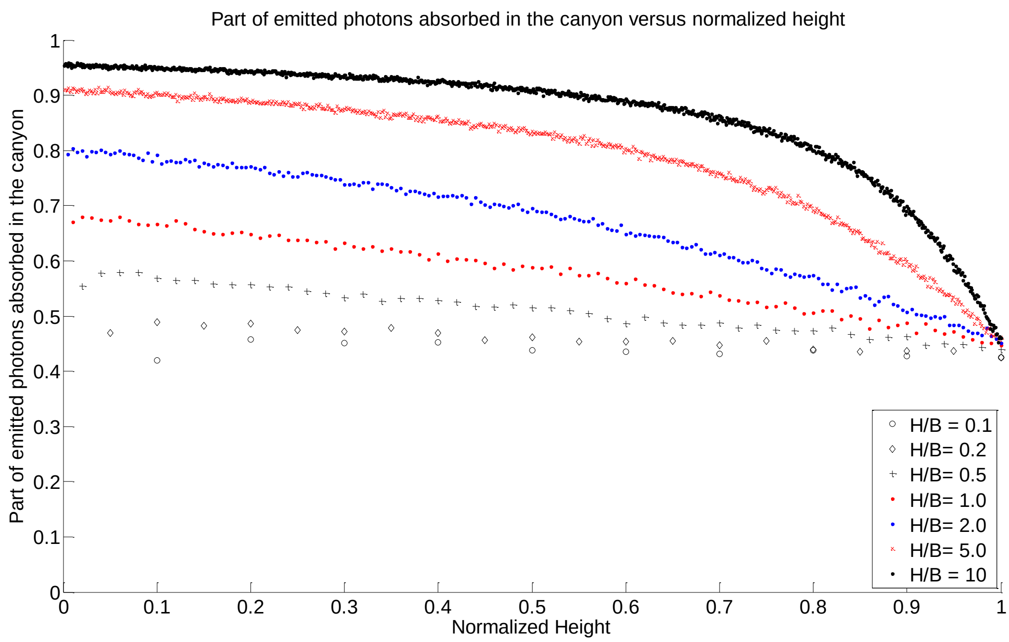

Figure 14 Fraction of the emitted photons absorbed in the canyon, versus the normalized height of the source of the wall

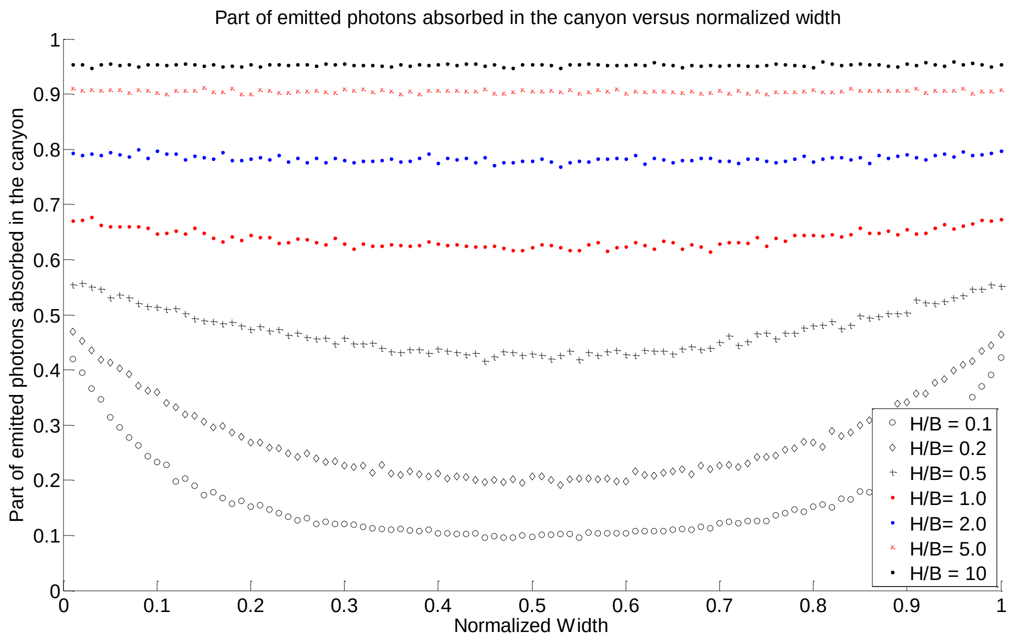

Figure 15 Fraction of the emitted photons absorbed in the canyon, versus the normalized width of the source of the street

The graphs have a few interesting characteristics. First in canyons with a high H_B, the chances for photons of escaping the canyon is much smaller than for canyons with lower H_B, except when looking at the top corners. Secondly, there is a gradation in the possibility of escape for photons: the lower in the canyon they are emitted from the wall, the less likely it is they will escape. This gradation is stronger in canyons with a high H_B.

For the street, another trend can be found. For $H_B < 1$, photons that are emitted from the centre of the street, are more likely to escape the canyon, than photons emitted near de corners. For $H_B > 1$, the begin position of a photon does not influence its chance of escaping.

5. Conclusion and discussion

5.1. Conclusion

In the introduction, two main research questions were introduced : How does the absorption of an ideal Urban Heat Canyon depend on the height to width ratio of the canyon and the zenith angle for both direct and indirect radiation? And: What is the influence of the height to width ratio of the canyon on emitting heat radiation? The quantity of interest was in all cases the absorption, defined by formula (2) as the ratio between the number of absorbed photons and the number of emitted/incident photons. In this chapter, these questions will be answered with the results from the chapter Results.

First, the case of an Urban Heat Canyon under influence of direct radiation was examined. The geometry, defined as the ratio between the width and the height of the canyon, H_B , clearly has a relation with the absorption: the higher this ratio, the higher the absorption. Next to that, the height to width ratio also has an effect on the influence of the zenith angle, θ : the higher the ratio, the larger the influence of the zenith angle. The zenith angle θ itself has a negative effect on the absorption: the lower the zenith angle, the higher the absorption.

Not just the absorption, but also the total amount of absorbed radiation was examined. The same relations hold true for the total amount of radiation as for the absorption, but it is very clear that the zenith angle has a greater impact than the height to width ratio. The height to width ratio only has influence on the total absorbed radiation when the zenith angle is between 20° and -20°.

Then the case of diffuse radiation: in this case, only the influence of the height to width ratio, $H_{B,}$ has been researched, because the zenith angle has no meaning for indirect radiation. From the results, it became clear that for H_B<1, there is a relation between the absorption and the height to width ratio: the larger the H_B, the larger the absorption For H_B>1, this behaviour reaches a limit, the absorption remains constant.

At last, the influence of the height to width ratio, H_B, on emitting heat radiation was researched. Overall, there was a positive relation between the H_B and the amount of emitted heat radiation that was absorbed again. There were, however, some interesting characteristics. First, the higher on the wall the radiation is emitted, the greater the chance it will escape the canyon. This effect is stronger for canyons with a higher H_B. Secondly, for H_B <1, radiation emitted from the centre of the street is more likely to escape than radiation emitted near the corners. This effect is stronger for lower H_B. For H_B>1, this effect is negligible.

Overall, one can say the following about the influences of the geometry and the zenith angle: Urban Heat Canyons with greater height-width ratios do absorb more radiation and are more difficult for heat radiation to escape from than canyons with a lower height-width ratio. In the case of direct radiation: the absorption is the highest when the zenith angle approaches zero, thus when the radiation enters the canyon vertically.

5.2. Discussion

This research has, although the conclusions are straightforward, a number of limitations. First of all, limitation to the 2D case makes it hard to translate the results to real life situations and compare them to actual measurements. Also the assumptions of rough but uninterrupted surface do not compare enough to the real life situation, but the effect of this is not investigated. Finally, the emissivity of the walls and the street is an issue. This is so for two reasons: firstly, hard values for this are hard to find, so a guess was used in this research. A possible extended research could look further into this emissivity, and see what the effect is on the absorption of the Urban Heat Canyon. Secondly, the direct and diffuse radiation from sun and atmosphere have a wavelength of another order than the heat radiation. In this research, the emissivity's used in both cases were the same, although this does not necessarily have to be so. Also in this case, more research is needed.

List of references

Aida, M. and Gotoh, K., Urban Albedo as a Function of the Urban Structure – A Two-Dimensional Numerical Simulation, *Boundary-Layer Meteorol*. **23** p.415-424, 1982

Harman, I. Best, M. and Belcher, S., Radiative Exchange in an Urban Street Canyon, *Boundary-Layer Meteorol*. **110** p.301-316, 2004

Heusinkveld, G. Van Hove, L.W.A Jacobs, C.M.J. Steeneveld, G.J. Elbers, J.A. Moors, E.J. and Holtslag, A.A.M., Use of a mobile platform for assessing urban heat stress in Rotterdam

Levinson, R. and Akbari, H., Effects of Composition and Exposure On the Solar Reflectance Of Portland Cement Concrete, *Lawrence Berkely National Laboratory Report,* LBNL-48334, 2001

Voogt, J. and Oke, T., Radiometric Temperatures of urban Canon Walls obtained from Vehicle Traverses, *Theor. Appl. Climatol*. **60**, p.199-217, 1998

Wolfson, R., *Essential University Physics*, Pearson Addison Wesley, 2007

Appendix

1. Alternative way of simulating Diffuse Radiation

One of the issues that came up while doing this thesis, was the nature of the diffuse radiation from the atmosphere. In the thesis, this problem was tackled by using a uniform distribution for the direction of the diffuse radiation. There was, however, another idea to simulate this. Diffuse radiation is solar radiation, reflected by random particles in the atmosphere. One could look at them as little sources of radiation. Instead of assuming random sources of this radiation throughout the whole atmosphere, it is assumed that all the sources are on a semicircle with a radius *R*, and the centre of the top of the canyon as centre. See Figure 16.

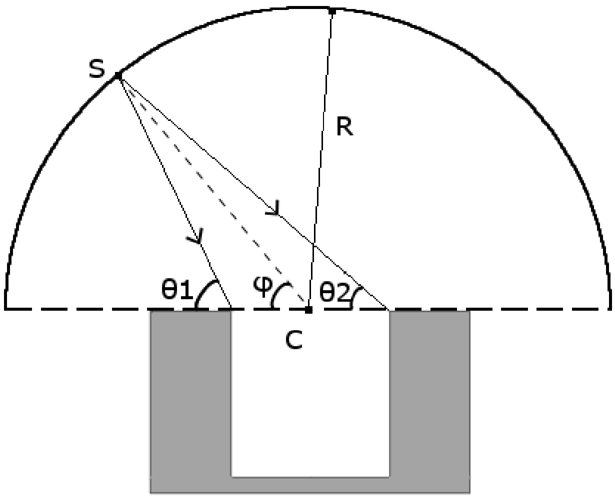

Figure 16 Visualization of the simulating steps used by the program to simulate the indirect radiation

The program to calculate the absorption in this situation simulates for each source *S*, with φ=1,2,3…180°, for each natural θ, from θ_1 to θ_2, 1000 photons. This was done for three values of R, the radius of the semicircle of sources: 5*B*, 10*B*, 20*B*, were *B* is the normalized width of the Urban Heat Canyon. The number of photons simulated from each source (180 sources were simulated on each semicircle)is found in Figure 17.

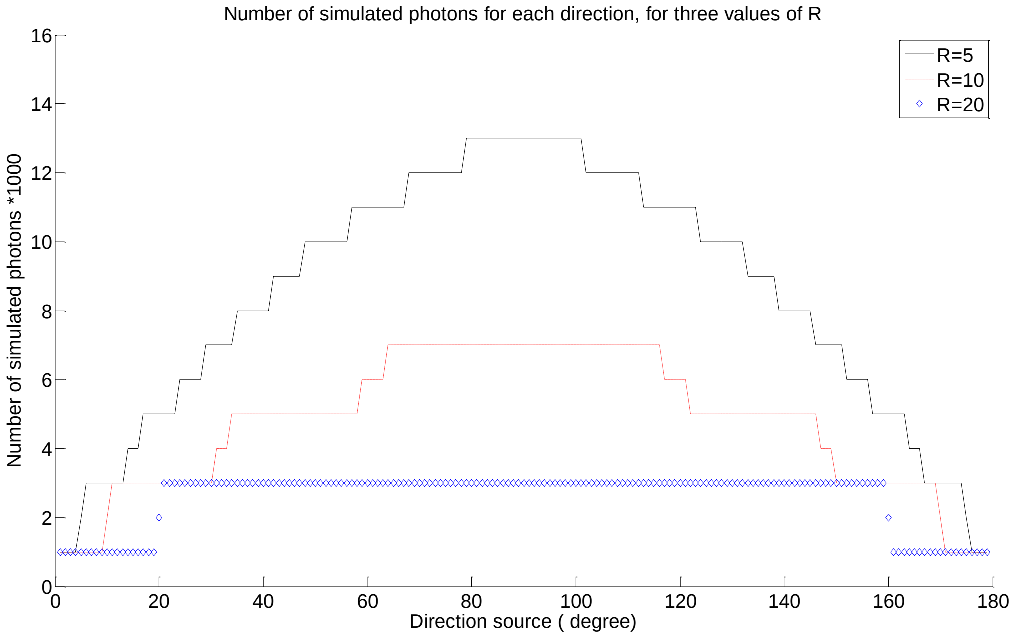

Figure 17 Number of simulated photons (·1000) plotted versus the direction of the source, in order to simulate indirect radiation

In order to correct for the fact that on each semicircle the same number of radiation sources were simulated, in further calculations the photons are weighed with a factor *R*. Accumulating the results from all these simulations gives a result for the absorption of the canyon for indirect radiation. This is done for several values of H_B.

Modelling the diffuse radiation in this manner, yielded the following results:

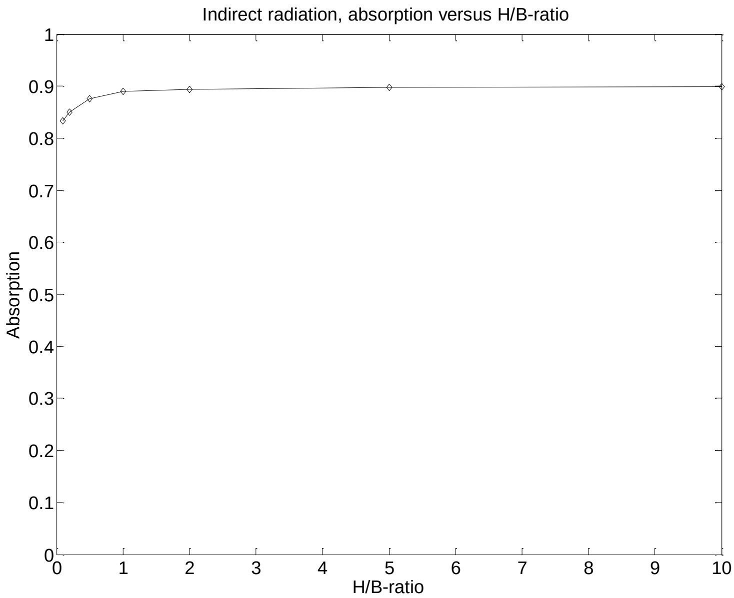

Figure 18 Absorption plotted versus H_B, in case of indirect radiation

From the plot in Figure 18 it becomes clear that for $H_B > 1$, the H_B has no influence on the absorption , in the situation of indirect radiation. For lower H_B there is a positive effect on the absorption.

It is interesting to notice, that this way of modelling the indirect radiation, yields almost the same results as the uniform distribution.

2. Matlab Codes

This entire thesis has been written using the results from various simulations in Matlab. As explained in the Experimental Setup, the simulations existed of three parts: the first one for simulating one photon, the second one for simulating N photons with the same start conditions, and the third one for simulation M times N photons, with varying H_B or θ. The first and second part were written as Matlab functions, used by the third part to produce result. In the table below are the names of the various parts, and how they are used:

Table 1 Names and purpose of the various Matlab scripts used to produce the results

Purpose	Simulating 1 photon	Simulating N photons	Varying conditions
Direct Radiation	**foton1**	**stralingzon3**	**directestraling**
Diffuse Radiation	**foton3**	**stralingindir2**	**indirectstraling**
Heat Output	**foton2**	**stralingcanyon**	**warmteafgifte**

In the various programs, also additional information in Dutch is given. Note that programming the various scripts was a process of trial-and-error, often the scripts could have been more simplified. Another important point is the simulating of the Diffuse Radiation. In earlier versions of this thesis, only the alternative model from the previous Appendix was used, so this for this version the Matlab codes are given. The model with the uniform distribution of diffuse radiation from section 3.3.2 was simulated, using the results from the direct radiation. The program used to do this is called **Indirectestralingmeth2**, and is the last program in this appendix.

Foton1

```
function [  F ] = foton1(B,H,theta,ew,es,E)

% Dit programma rekent uit wat er met een foton gebeurt, die vanaf de top
% van de canyon invalt. De beginpositie is dus op hoogte H met een zekere
% hoek. X-coordinaat wordt random gekozen.

% begin positie foton:

x = B*rand;
f=zeros(1,5);
f(1)=x; % xcoordinaat, 0-punt licht linksonder in de canyon
f(2)=H; % y-coordinaat, 0-punt is de bodem van de canyon
f(3)=theta; % hoek met de horizontaal, waarheen het deeltje weggaat
f(4)=1; % reflectievoorwaarde: 0= absorptie, 1= reflectie, 2 = weggevlogen naar
admosfeer
f(5)=E; % energie van het deeltje, zie 2 cellen naar boven

F(1,:)=f;

% translatie naar eerste rand/weg

fold=f;
fnow=zeros(1,4);
% fnow is nodig om precieze positie, mocht de straat geraakt worden, te bepalen

if fold(3) < 90
    fnow(1)=B;
    fnow(2)=fold(2)-tand(fold(3))*(B-fold(1));
elseif fold(3) > 90
    fnow(1)=0;
    fnow(2)=fold(2)-fold(1)*tand(180-fold(3));
else fnow(1)=fold(1);
    fnow(2)=0;
end

if (fnow(2) < 0) % Deeltje is op de weg gekomen
        f(2)=0;
        f(1)=fold(2)/tand(abs(fold(3)))+fold(1);
else f(2)=fnow(2); % Deeltje is beland op de tegenoverliggende muur
        f(1)=fnow(1);
end

if (f(2)==0) % Bepaling van de nieuwe hoek
        f(3)= rand*180;
        ref=rand;
    if (ref > es)
    f(4)=0;
    else f(4)=1;
    end

else f(3)= rand*180-90;
    ref=rand;
    if (ref > ew)
    f(4)=0;
    else f(4)=1;
    end
end



F(2,:)=f;

% Nu een systeem voor elke willekeurige botsing

    a=2;
```

```
while f(4)==1 % zolang als er reflectie, wordt onderstaand programma herhaald
    fold=f; % waarde uit de vorige iteratie

    a=a+1;

% Eerst geval waarbij deeltje op de rechterwand zit

if fold(1)==B;
    fnow(1)=0;
    fnow(2)=fold(2)+B*tand(fold(3));
    if fnow(2) < 0 % geval dat het deeltje op de weg is gekomen
        f(2)=0;
        f(1)=B-fold(2)/tand(abs(fold(3)));
    elseif fnow(2) > H % geval dat het deeltje de atmosfeer in is gevlogen
        f(2)=fnow(2);
        f(1)=fnow(1);
        fnow(4)=2;
    else f(2)=fnow(2); % het deeltje is op de tegenoverliggende muur geland
        f(1)=fnow(1);
    end

    if f(2)==0 % bepaling van de nieuwe hoek, afhankelijk van geland op weg/muur
        f(3)= rand*180;
    else f(3)= rand*180-90;
    end

    if fnow(4)==2; % bepaling van de nieuwe reflectievoorwaarde
        f(4)=2;
    elseif f(2)==0
        ref=rand;
        if ref > es
            f(4)=0;
        else f(4)=1;
        end
    else ref=rand;
        if ref > ew
            f(4)=0;
        else f(4)=1;
        end
    end

% Nu het geval waarin het deeltje zich op de linkerwand bevindt

elseif fold(1)==0
    fnow(1)=B;
    fnow(2)=fold(2)+B*tand(fold(3));
     if fnow(2) < 0 % Deeltje is op de weg gekomen
        f(2)=0;
        f(1)=fold(2)/tand(abs(fold(3)));
    elseif fnow(2) > H % Deeltje is de atmosfeer ingevlogen
        f(2)=fnow(2);
        f(1)=fnow(1);
        fnow(4)=2;
    else f(2)=fnow(2); % Deeltje is beland op de tegenoverliggende muur
        f(1)=fnow(1);
     end

     if f(2)==0 % Bepaling van de nieuwe hoek
        f(3)= rand*180;
    else f(3)= rand*180-90;
     end

     if fnow(4)==2;
        f(4)=2;
    elseif f(2)==0
        ref=rand;
```

```
        if ref > es
            f(4)=0;
        else f(4)=1;
        end
    else ref=rand;
        if ref > ew
            f(4)=0;
        else f(4)=1;
        end
     end

% Als laatste het geval waarbij het deeltje op de bodem van de canyon zit

else fnow(2)=H;
    fnow(1)=fold(1)-H/tand(fold(3));
    if fnow(1) < 0 % Deeltje landt op de linkerwand
        f(1)=0;
        f(2)=fold(1)*tand(fold(3));
    elseif fnow(1) > B % Deeltje landt op de rechterwand
        f(1)=B;
        f(2)=(B-fold(1))*tand(180-fold(3));
    else f(1)=fnow(1); % Deeltje vliegt de atmosfeer in
        f(2)=fnow(2);
        fnow(4)=2;
    end

    f(3)=rand*180-90;

    if fnow(4)==2;
        f(4)=2;
    else ref=rand;
        if ref > ew
            f(4)=0;
        else f(4)=1;
        end
    end

end

% Nu de resultaten opslaan

    F(a,:)=f;


end



end
```

Foton2

```
function [  F ] = foton2(x,y,B,H,theta,ew,es,E)

% Dit programma rekent uit wat een foton doet, die op een random plek
% vanuit de wand/vloer van de canyon vertrekt.
% begin positie foton:

f=zeros(1,5);
f(1)=x; % xcoordinaat, 0-punt licht linksonder in de canyon
f(2)=y; % y-coordinaat, 0-punt is de bodem van de canyon
f(3)=theta; % hoek met de horizontaal, waarheen het deeltje weggaat
f(4)=1; % reflectievoorwaarde: 0= absorptie, 1= reflectie, 2 = weggevlogen naar
admosfeer
f(5)=E; % energie van het deeltje, zie 2 cellen naar boven

F(1,:)=f;

% Nu een systeem voor elke willekeurige botsing

 fnow=zeros(1,4);
   fnow(4)=1;
    a=1;

while f(4)==1 % zolang als er reflectie, wordt onderstaand programma herhaald
    fold=f; % waarde uit de vorige iteratie
    a=a+1;

% Eerst geval waarbij deeltje op de rechterwand zit

if fold(1)==B;
    fnow(1)=0;
    fnow(2)=fold(2)+B*tand(fold(3));
    if fnow(2) < 0 % geval dat het deeltje op de weg is gekomen
        f(2)=0;
        f(1)=B-fold(2)/tand(abs(fold(3)));
    elseif fnow(2) > H % geval dat het deeltje de atmosfeer in is gevlogen
        f(2)=fnow(2);
        f(1)=fnow(1);
        fnow(4)=2;
    else f(2)=fnow(2); % het deeltje is op de tegenoverliggende muur geland
        f(1)=fnow(1);
    end

    if f(2)==0 % bepaling van de nieuwe hoek, afhankelijk van geland op weg/muur
        f(3)= rand*180;
    else f(3)= rand*180-90;
    end

    if fnow(4)==2; % bepaling van de nieuwe reflectievoorwaarde
        f(4)=2;
    elseif f(2)==0
        ref=rand;
        if ref > es
            f(4)=0;
        else f(4)=1;
        end
    else ref=rand;
        if ref > ew
            f(4)=0;
        else f(4)=1;
        end
    end

% Nu het geval waarin het deeltje zich op de linkerwand bevindt

elseif fold(1)==0
```

```
    fnow(1)=B;
    fnow(2)=fold(2)+B*tand(fold(3));
     if fnow(2) < 0 % Deeltje is op de weg gekomen
        f(2)=0;
        f(1)=fold(2)/tand(abs(fold(3)));
    elseif fnow(2) > H % Deeltje is de atmosfeer ingevlogen
        f(2)=fnow(2);
        f(1)=fnow(1);
        fnow(4)=2;
    else f(2)=fnow(2); % Deeltje is beland op de tegenoverliggende muur
        f(1)=fnow(1);
     end

     if f(2)==0 % Bepaling van de nieuwe hoek
        f(3)= rand*180;
    else f(3)= rand*180-90;
     end

     if fnow(4)==2;
        f(4)=2;
    elseif f(2)==0
        ref=rand;
        if ref > es
            f(4)=0;
        else f(4)=1;
        end
    else ref=rand;
        if ref > ew
            f(4)=0;
        else f(4)=1;
        end
     end

% Als laatste het geval waarbij het deeltje op de bodem van de canyon zit

else fnow(2)=H;
    fnow(1)=fold(1)-H/tand(fold(3));
    if fnow(1) < 0 % Deeltje landt op de linkerwand
        f(1)=0;
        f(2)=fold(1)*tand(fold(3));
    elseif fnow(1) > B % Deeltje landt op de rechterwand
        f(1)=B;
        f(2)=(B-fold(1))*tand(180-fold(3));
    else f(1)=fnow(1); % Deeltje vliegt de atmosfeer in
        f(2)=fnow(2);
        fnow(4)=2;
    end

    f(3)=rand*180-90;

    if fnow(4)==2;
        f(4)=2;
    else ref=rand;
        if ref > ew
            f(4)=0;
        else f(4)=1;
        end
    end
end

% Nu de resultaten opslaan

    F(a,:)=f;

end

end
```

Foton3

```
function [  F ] = foton3(x,B,H,theta,ew,es,E)

% Dit programma rekent uit wat er met een foton gebeurt, die vanaf de top
% van de canyon invalt. De beginpositie is dus op hoogte H met een zekere
% hoek. De x-coordinaat moet ook worden ingegeven.

% begin positie foton:

f=zeros(1,5);
f(1)=x; % xcoordinaat, 0-punt ligt linksonder in de canyon
f(2)=H; % y-coordinaat, 0-punt is de bodem van de canyon
f(3)=theta; % hoek met de horizontaal, waarheen het deeltje weggaat
f(4)=1; % reflectievoorwaarde: 0= absorptie, 1= reflectie, 2 = weggevlogen naar
admosfeer
f(5)=E; % energie van het deeltje, zie 2 cellen naar boven

F(1,:)=f;

% translatie naar eerste rand/weg

fold=f;
fnow=zeros(1,4);
% fnow is nodig om precieze positie, mocht de straat geraakt worden, te bepalen

if fold(3) < 90
    fnow(1)=B;
    fnow(2)=fold(2)-tand(fold(3))*(B-fold(1));
elseif fold(3) > 90
    fnow(1)=0;
    fnow(2)=fold(2)-fold(1)*tand(180-fold(3));
else fnow(1)=fold(1);
    fnow(2)=0;
end

if (fnow(2) < 0) % Deeltje is op de weg gekomen
        f(2)=0;
        f(1)=fold(2)/tand(abs(fold(3)))+fold(1);
else f(2)=fnow(2); % Deeltje is beland op de tegenoverliggende muur
        f(1)=fnow(1);
end

if (f(2)==0) % Bepaling van de nieuwe hoek
        f(3)= rand*180;
        ref=rand;
    if (ref > es)
    f(4)=0;
    else f(4)=1;
    end

else f(3)= rand*180-90;
    ref=rand;
    if (ref > ew)
    f(4)=0;
    else f(4)=1;
    end
end



F(2,:)=f;

% Nu een systeem voor elke willekeurige botsing

    a=2;
```

```
while f(4)==1 % zolang als er reflectie, wordt onderstaand programma herhaald
    fold=f; % waarde uit de vorige iteratie

    a=a+1;

% Eerst geval waarbij deeltje op de rechterwand zit

if fold(1)==B;
    fnow(1)=0;
    fnow(2)=fold(2)+B*tand(fold(3));
    if fnow(2) < 0 % geval dat het deeltje op de weg is gekomen
        f(2)=0;
        f(1)=B-fold(2)/tand(abs(fold(3)));
    elseif fnow(2) > H % geval dat het deeltje de atmosfeer in is gevlogen
        f(2)=fnow(2);
        f(1)=fnow(1);
        fnow(4)=2;
    else f(2)=fnow(2); % het deeltje is op de tegenoverliggende muur geland
        f(1)=fnow(1);
    end

    if f(2)==0 % bepaling van de nieuwe hoek, afhankelijk van geland op weg/muur
        f(3)= rand*180;
    else f(3)= rand*180-90;
    end

    if fnow(4)==2; % bepaling van de nieuwe reflectievoorwaarde
        f(4)=2;
    elseif f(2)==0
        ref=rand;
        if ref > es
            f(4)=0;
        else f(4)=1;
        end
    else ref=rand;
        if ref > ew
            f(4)=0;
        else f(4)=1;
        end
    end

% Nu het geval waarin het deeltje zich op de linkerwand bevindt

elseif fold(1)==0
    fnow(1)=B;
    fnow(2)=fold(2)+B*tand(fold(3));
     if fnow(2) < 0 % Deeltje is op de weg gekomen
        f(2)=0;
        f(1)=fold(2)/tand(abs(fold(3)));
    elseif fnow(2) > H % Deeltje is de atmosfeer ingevlogen
        f(2)=fnow(2);
        f(1)=fnow(1);
        fnow(4)=2;
    else f(2)=fnow(2); % Deeltje is beland op de tegenoverliggende muur
        f(1)=fnow(1);
     end

     if f(2)==0 % Bepaling van de nieuwe hoek
        f(3)= rand*180;
    else f(3)= rand*180-90;
     end

     if fnow(4)==2;
        f(4)=2;
    elseif f(2)==0
        ref=rand;
        if ref > es
```

```
                f(4)=0;
            else f(4)=1;
            end
        else ref=rand;
            if ref > ew
                f(4)=0;
            else f(4)=1;
            end
         end

% Als laatste het geval waarbij het deeltje op de bodem van de canyon zit

else fnow(2)=H;
    fnow(1)=fold(1)-H/tand(fold(3));
    if fnow(1) < 0 % Deeltje landt op de linkerwand
        f(1)=0;
        f(2)=fold(1)*tand(fold(3));
    elseif fnow(1) > B % Deeltje landt op de rechterwand
        f(1)=B;
        f(2)=(B-fold(1))*tand(180-fold(3));
    else f(1)=fnow(1); % Deeltje vliegt de atmosfeer in
        f(2)=fnow(2);
        fnow(4)=2;
    end

    f(3)=rand*180-90;

    if fnow(4)==2;
        f(4)=2;
    else ref=rand;
        if ref > ew
            f(4)=0;
        else f(4)=1;
        end
    end

end

% Nu de resultaten opslaan

    F(a,:)=f;


end



end
```

Stralingzon3

```
function [ canyon ]=stralingzon3(theta,H,B,ew,es,n,m)
%geeft een verdeling van fotonposities aan.

HB=H/B;
E=1;

P=zeros(n,3);
canyon=zeros(HB*m+1,m+1);

for b = 1:n
    [  F ]=foton1(B,H,theta,ew,es,E);
    P(b,1:3)=F(end, 1:3);
end

Pnow=P(:,1:3);
Pnow(:,1)=int32(round((P(:,1).*m./B))+1);
Pnow(:,2)=int32(round((P(:,2).*m./H.*HB))+1);

p = ( Pnow(:,2) <= (HB*m+1) );
Pnew=int32(Pnow(p,:));

for b=1:length(Pnew(:,1))
    canyonold=canyon;
    canyon(Pnew(b,2),Pnew(b,1))=canyonold(Pnew(b,2),Pnew(b,1))+1;
end

canyon(end,2:(m-1))=0;

end
```

Stralingindir2

```
function [ V ] = stralingindir2(x,B,H,theta,ew,es,E,n,m)

HB=H/B;

P=zeros(n,3);

for b = 1:n

    [  F ]=foton3(x,B,H,theta,ew,es,E);

    P(b,1:3)=F(end, 1:3);

end

canyon=zeros(HB*m+1,m+1);
Pnow=P(:,1:3);

Pnow(:,1)=int32(round((P(:,1).*m./B))+1);
Pnow(:,2)=int32(round((P(:,2).*m./H.*HB))+1);

p = ( Pnow(:,2) <= (HB*m+1) );

Pnew=int32(Pnow(p,:));

for b=1:length(Pnew(:,1))
    canyonold=canyon;
    canyon(Pnew(b,2),Pnew(b,1))=canyonold(Pnew(b,2),Pnew(b,1))+1;
```

```
end

canyon(end,2:(m-1))=0;

V=canyon;


end
```

Stralingcanyon

```
function [ W ]=stralingcanyon(x,y,B,H,ew,es,n,m)

HB=H/B;
E=1;


P=zeros(n,3);

for a=1:n
    if y==0
        phi=rand*180;
    else
        phi =rand*180-90;
    end
   [F]=foton2(x,y,B,H,phi,ew,es,E);
   P(a,1:3)=F(end,[1 2 5]);
end

canyon=zeros(HB*m+1,m+1);
Pnow=P(:,1:3);

Pnow(:,1)=int32(round((P(:,1).*m./B))+1);
Pnow(:,2)=int32(round((P(:,2).*m./H.*HB))+1);

p = ( Pnow(:,2) <= (HB*m) );

Pnew=Pnow(p,:);

for b=1:length(Pnew(:,1))
    canyonold=canyon;
    canyon(Pnew(b,2),Pnew(b,1))=canyonold(Pnew(b,2),Pnew(b,1))+1;
end

canyon(end,2:(m-1))=0;
W=zeros(1,2);
W(1)=sum(sum(canyon));
W(2)=W(1)/n;
end
```

Directestraling

```
% Dit programma, directestraling, gaat de directe straling doorrekenen.
% Voor diverse H/B verhoudingen, ew/es waarden en hoeken rekent hij uit
% waar fotonen uiteindelijk terecht komen in de canyon. Hiervoor gebruikt
% dit programma stralingzon3.m!

clc; clear all; close all;

B=1;
es=0.2;
ew=0.3;
n=100000;
m=100;

HB=[0.1 0.2 0.5 1 2 5 10];

for a=1:length(HB)
    H=HB(a)*B;
    for b=1:180;
        theta=b;
        V=stralingzon3(theta,H,B,ew,es,n,m);

name=sprintf('/home/victor/Documents/Resultaten/directestraling/HB=%g/theta=%g',HB(
a),theta');
        save(name,'n','m','V');
    end
end
```

Indirectestraling

```
Dit programma gaat de indirecte straling doorrekenen, voor diverse
% waarden van R en de H/B-verhouding. Voor elke R en H/B wordt de fotonverdeling in
% de canyon bewaart. Daarnaast moet er nog een soort manier bedacht worden
% om het totaal aantal fotonen dat ( eventueel per hoek ) doorberekend is.

close all; clear all; clc;

B=1;
es=0.2;
ew=0.3;
n=10000; % aantal fotonen per bron
m=100; % grootte van het grid
k=10; % aantal bronnen
E=1;

HB=[0.1 0.2 0.5 1 2 5 10];
RR=[5 10 20 50];

coord=zeros(179,5);

for a=1:length(HB)
    H=HB(a)*B;

    for b=1:length(RR)
        R=RR(b);

       for phi=1:179

            coord(phi,1)=phi; % hoek vanaf de bron naar het centrum van de ingang
van de canyon
            coord(phi,2)=-R*cosd(phi)+B/2; % xcoordinaat van de "bron"
            coord(phi,3)=R*sind(phi)+H; % ycoordinaat van de "bron"
            if coord(phi,2) < 0
```

```
                coord(phi,4)= int32(round(atand((coord(phi,3)-
H)/abs(coord(phi,2)))))); % hoek van de bron tot het linkereind van de canyon, tov
horizontaal
                coord(phi,5)=int32(round(atand((coord(phi,3)-
H)/(abs(coord(phi,2))+B)))); %hoek van bron tot rechtereind canyon, tov
horizontaal.
            elseif coord(phi,2) > B
                coord(phi,4)=int32(round(180-atand((coord(phi,3)-
H)/coord(phi,2)))); % hoek van de bron tot het linkereind van de canyon, tov
horizontaal
                coord(phi,5)=int32(round(180-atand((coord(phi,3)-H)/(coord(phi,2)-
B)))); %hoek van bron tot rechtereind canyon, tov horizontaal.
            else coord(phi,4)=int32(round(180-atand((coord(phi,3)-
H)/coord(phi,2))));
                 coord(phi,5)=int32(round(atand((coord(phi,3)-H)/(-
coord(phi,2)+B))));
            end

       end

       % Deze twee loops zitten niet aan elkaar, aangezien er hierdoor
       % een random fout in het programma kwam.

       C=zeros(H*m+1,m+1);
       info=zeros(2,179);

       for fi=1:179



           for theta=coord(fi,5):1:coord(fi,4)    % elke hoek inclusief de twee
grenshoeken gaan we doorrekenen
                X=coord(fi,2)+(coord(phi,3)-H)/tand(theta); % positie waar de
fotonen de canyon inkomen, is alleen erg onnauwkeurig door afronding van de hoeken
( voor erg lage en hoge hoeken )
                if X < 0
                    x=rand*B;
                elseif X > B;
                    x=rand*B;
                else x=X;
                end

                V = stralingindir2(x,B,H,theta,ew,es,E,n,m);

                C=C+V;




            end

            info(1,fi)=fi;
            info(2,fi)=length(coord(fi,5):1:coord(fi,4));

       end


name=sprintf('/home/victor/Documents/Resultaten/indirectestraling/HB=%g/R=%g',HB(a)
,R);
       save(name,'n','m','info','C');


    end

end
```

Warmteafgifte

```
% Dit programma rekent uit wat er gebeurd met straling die door de wanden
% en vloer van de put uitgezonden wordt. Eerst per positie op de wand,
% daarna iets anders bedenken.

clc;clear all; close all;

B=1;
HBB=[0.1 0.2 0.5 1 2 5 10];
ew=0.3;
es=0.2;
n=10000;
m=100;




for e=1:length(HBB);
    HB=HBB(e);
    H=HB*B;
    G=zeros(HB*m+1,m+1);

    for a=2:(HB*m+1)
        y=a/(HB*m+1)*H;
        W=stralingcanyon(0,y,B,H,ew,es,n,m);
        G(a,1)=W(2);
    end

    for c=1:m+1
        x=c/(m+1)*B;
        W=stralingcanyon(x,0,B,H,ew,es,n,m);
        G(1,c)=W(2);

    end

    for d=2:(HB*m+1)
        y=d/(HB*m+1)*H;
         W=stralingcanyon(B,y,B,H,ew,es,n,m);
         G(d,end)=W(2);

    end

    name=sprintf('/home/victor/Documents/Resultaten/warmteafgifte/HB=%g',HB);
    save(name,'n','m','G');
end
```

Indirectstralingmeth2

```
clc; clear all; close all;

HB=[0.1 0.2 0.5 1 2 5 10];
Grafiek=zeros(7,180);
Grafiek2=zeros(7,1);
for b=1:7

for a=1:180
    name=sprintf('HB=%g/theta=%g.mat',HB(b),a);
    load(name);
    V(end,end-1)=0;
    Grafiek(b,a)= sum(sum(V))/n;
end

Grafiek2(b)=sum(Grafiek(b,:))/180;

end

FS=20;
plot(HB,Grafiek2,'kd-');
axis([0 10 0 1]);

xlabel('H/B-ratio','FontSize',FS);
ylabel('Albedo','FontSize',FS);
title('Indirect radiation, albedo versus H/B-ratio ','FontSize',FS);
```