

Model voor het berekenen van de warmtestraling van elektrische voertuigbranden



Nederlandse Academie voor
Crisisbeheersing en Brandweezorg
Postbus 7010
6801 HA Arnhem
Kemperbergerweg 783, Arnhem
www.nipv.nl
info@nipv.nl
026 355 24 00

Colofon

© Nederlands Instituut Publieke Veiligheid (NIPV), 2024

Auteurs H. Brans en J. Reinders
Met medewerking van P. van Rede

Opdrachtgever Omgevingsdienst NL
Contactpersoon E. Graven

Datum 26 april 2024

Foto cover 112Groningen.nl

Wij hechten veel belang aan kennisdeling. Delen uit deze publicatie mogen dan ook worden overgenomen op voorwaarde van bronvermelding.

Het Nederlands Instituut Publieke Veiligheid is bij wet vastgelegd onder de naam Instituut Fysieke Veiligheid.

Inhoud

Inleiding	5
Aanleiding	5
Doelstelling en methode	5
Afbakening	6
Begeleidingscommissie	6
Leeswijzer	6
1 Onderzoeksmethode	7
1.1 Literatuurstudie	7
1.2 Opzetten van het rekenmodel	7
1.3 Uitwerken scenario's	7
2 Literatuuronderzoek naar voertuig- en batterijbranden	8
2.1 Inleiding	8
2.2 Bijdrage van batterijen aan een autobrand	8
2.3 Verband tussen HRR en energie-inhoud van een batterij	8
2.4 Brandbare materialen in auto's	9
2.5 Brandproeven met elektrische auto's	10
2.6 Brandvermogens van voertuigen	11
2.7 Beschouwing	13
3 Rekenmodel	14
3.1 Inleiding	14
3.2 Solid Flame Model	14
3.3 Geometrie van het voertuig	15
3.4 Vlamhoogte	16
3.5 Bepaling van de viewfactor	17
3.5.1 Viewfactor loodrecht op een plat vlak	17
3.5.2 Viewfactor in de hoekpunten	18
3.6 Transmissiviteit	19
3.7 Onzekerheidsanalyse	19
3.8 Beschouwing	20
4 Warmtestralingscontouren	22
4.1 Inleiding	22
4.2 Elektrische personenauto - SUV	22
4.3 Bus	24
4.4 Vrachtwagen met lading	26
4.5 Beschouwing	28
5 Conclusie	29
Literatuurlijst	30

Bijlage 1	32
Leden van de Begeleidingscommissie	32
Bijlage 2	33
Broncode rekenprogramma (Python)	33

Inleiding

Aanleiding

Branden van elektrisch (via Li-ion batterijen) aangedreven voertuigen kunnen lang aanhouden en zijn vaak moeizaam te bestrijden als het batterijpakket bij de brand betrokken is. Naast veel voorkomende brandoorzaken zoals brandstichting, kan een elektrische voertuigbrand ook het gevolg zijn van een thermal runaway. Een thermal runaway kan ontstaan als een batterijpakket oververhit, overladen of beschadigd raakt. Gedurende dit proces worden binnenin de batterij brandbare en giftige gassen gevormd, die op een gegeven moment explosief naar buiten treden en daarbij brand kunnen veroorzaken. Er bestaat een kans dat dit bij het opladen gebeurt. Laden vindt plaats op locaties waar voertuigen worden gestald of geparkeerd, in het algemeen in de nabijheid van andere activiteiten of voertuigen, en in dan wel nabij gebouwen.

Uit diverse gesprekken met onder meer de veiligheidsregio's, PGS-commissies en vergunningverleners is gebleken dat er om onder meer deze redenen behoefte bestaat aan inzicht in de te verwachten warmtestraling bij elektrische voertuigbranden. Wanneer een (elektrisch) voertuig in brand raakt, kunnen namelijk ook voorwerpen in de directe omgeving ontbranden of kunnen er slachtoffers vallen, hetzij direct door de batterijbrand (waarbij fakkels ontstaan), hetzij als gevolg van de warmtestraling van het brandende voertuig. Om deze reden is het van belang om een beeld te krijgen van de reikwijdte van de effecten (warmtestraling) van (elektrische) voertuigbranden.

Inzicht in de te verwachten warmtestraling tijdens elektrische voertuigbranden geeft bovendien mogelijkheden om vuistregels voor veiligheidsafstanden af te leiden. Deze kennis is relevant voor incidentbestrijding, maar ook voor ruimtelijke inrichting en te nemen beheersmaatregelen voor bijvoorbeeld parkeergarages, tunnels en voetgangerszones. Daarnaast kan deze informatie worden opgenomen in nieuwe les- en leerstof over de bestrijding van branden in elektrische voertuigen.

Data over warmtestraling van een brand in elektrische voertuigen of modellen om deze te berekenen zijn (zo goed als) niet aanwezig. Dit bleek uit een interne rondvraag bij het NIPV, bij een rondvraag bij het RIVM en uit een voorstudie door de veiligheidsregio's Gelderland-Zuid en Twente. Meer algemeen bleek deze informatie ook voor conventionele voertuigen niet beschikbaar te zijn. Om in de behoefte aan kennis hieromtrent te voorzien, is deze studie uitgevoerd.

Doelstelling en methode

Hoofddoelstelling van deze studie is om een rekenmodel op te stellen waarmee de warmtestraling van elektrische voertuigbranden kan worden berekend. Hierbij wordt gebruikgemaakt van bestaande inzichten (modellen) en van gemeten brandparameters uit gepubliceerde brandexperimenten. Ook zijn enkele voorbeeldberekeningen uitgevoerd. Hierbij is gebruikgemaakt van uit de literatuur verkregen (experimentele) data.

Om dit doel te bereiken is de studie in drie fasen uitgevoerd:

1. **Literatuurstudie.** Het bestuderen van de wetenschappelijke literatuur over brandexperimenten met voertuigen en rekenmethodiek(en) voor de bepaling van warmtestraling.
2. **Opzetten van het rekenmodel.** Op basis van de literatuurstudie is een rekenmodel opgezet en geprogrammeerd, waarmee warmtestralingsniveaus kunnen worden berekend.
3. **Uitwerken van enkele scenario's.** Via proefberekeningen zijn voor drie voertuigtypes warmtestralingscontouren getekend.

Afbakening

- > Het rekenmodel beschrijft uitsluitend de warmtestraling als functie van de afstand tot een brandend voertuig. Het model tracht niet om de batterijbrand op detailniveau te simuleren.
- > Andere effecten van elektrische-voertuigbranden dan de warmtestraling worden in deze studie niet behandeld.
- > Fakkels en wegschietende cellen vallen (derhalve) buiten de beschouwing. Door hun korte duur is van fakkels alleen via direct vlamcontact brandoverslag te verwachten en speelt warmtestraling geen rol. Ook wegschietende (hete) batterijcellen zullen alleen via direct contact brandpropagatie veroorzaken.

Begeleidingscommissie

Voor dit onderzoek is een begeleidingscommissie samengesteld met leden van het RIVM, de parkeerbranche, veiligheidsregio's en de Nationale Agenda Laadinfrastructuur. In elke fase van de studie is de begeleidingscommissie bijeengekomen om de tussenresultaten te bespreken en keuzes met betrekking tot de voortgang te maken.

Leeswijzer

In hoofdstuk 1 wordt de onderzoeksmethode nader toegelicht. In hoofdstuk 2 komt de literatuurstudie naar voertuig- en batterijbranden aan bod. In hoofdstuk 3 wordt het fysisch-mathematische rekenmodel beschreven, dat in hoofdstuk 4 wordt toegepast op drie voertuigtypes: een elektrische personenauto (SUV), een bus en een vrachtwagen. Warmtestralingscontouren rondom de voertuigen worden weergegeven, alsook het verloop van de warmtestraling als functie van de afstand tot de voertuigen. Hoofdstuk 5 bevat de conclusie. Bijlage 1 bevat de namen en functies van de leden van de begeleidingscommissie. De broncode van het rekenmodel is in Bijlage 2 toegevoegd.

1 Onderzoeksmethode

Het onderzoek bestond, zoals vermeld, uit drie fases:

- > een literatuurstudie
- > het opstellen en programmeren van een rekenmodel
- > het toepassen en testen van het rekenmodel en uitwerken van drie, in overleg met de begeleidingscommissie bepaalde, brandscenario's.

1.1 Literatuurstudie

De literatuurstudie was gericht op het verkrijgen van inzicht in bestaande rekenmodellen en het verzamelen van experimentele data over voertuigbranden (zowel elektrisch als fossiel aangedreven). Er is met name gezocht naar informatie over de volgende parameters:

- > warmteontwikkeling in de tijd
- > totale verbrandingsenergie
- > maximale warmteontwikkeling
- > type voertuig
- > relatieve bijdragen van brandstof, accu, carrosserie, interieur en bekleding
- > warmtestralingsniveau 's van (auto)branden, gemeten of anderszins bepaald.

1.2 Opzetten van het rekenmodel

Op basis van de literatuurstudie is een rekenmodel opgezet waarmee de stralingsniveaus van voertuigbranden op afstand van de brand kunnen worden berekend. Dit rekenmodel is geprogrammeerd in de programmeertaal Python. Door testberekeningen en het analyseren van tussenresultaten is de broncode getest en, waar passend, verbeterd.

Het model bevat ook de mogelijkheid om, via een onzekerheidsanalyse, de invloed van een variatie in invoerparameters op het resultaat te berekenen. Belangrijkste reden hiervoor is dat experimentele gegevens over warmtestraling van voertuigbranden zeer schaars bleken zijn. Dit geldt ook voor de benodigde invoergegevens waarvan het model gebruikmaakt. Dit betekent dat de met het model berekende warmtestralingscontouren met de nodige onzekerheid omgeven zijn. De via de onzekerheidsanalyse berekende foutenmarge geeft een beeld van de grootte van de onzekerheid.

1.3 Uitwerken scenario's

In deze studie zijn met het rekenmodel drie scenario's uitgewerkt. Het betreft voertuigbranden van een (wat grotere) personenauto (SUV), een bus en een vrachtwagen. Bij de berekening van de foutenmarge is de invloed van de onzekerheid van de meest dominante parameters op de warmtestraling verwerkt.

2 Literatuuronderzoek naar voertuig- en batterijbranden

2.1 Inleiding

Het aantal brandexperimenten met bruikbare brandparameters van elektrische voertuigen is beperkt. Dit geldt met name voor grotere voertuigen, zoals bussen en vrachtwagens. Wel wordt er in studies over tunnelbranden gesproken over het maximum brandvermogen om de grootte van de brand uit te drukken. In die studies is het brandvermogen een maat om de beheersbaarheid van de brand aan te geven, specifiek gericht op de mogelijkheden voor repressie.

Een elektrische voertuigbrand kan in principe worden beschouwd als een brandend batterijpakket en een brandend voertuigchassis (zie paragraaf 2.2). Om die reden is een beknopte literatuurstudie uitgevoerd naar het brandvermogen van batterijbranden en de (andere) brandbare materialen in auto's. Het voertuigchassis van een elektrische en conventionele auto zijn globaal gezien immers van vergelijkbare aard. Dat impliceert dat significante verschillen in het brandgedrag en brandvermogen gezocht dan wel verklaard dienen te worden in het batterijpakket of de brandstoftank. Om die reden is inzicht in de afzonderlijke componenten van belang.

2.2 Bijdrage van batterijen aan een autobrand

Volgens Zhang et al. (2023) is de totale warmteontwikkeling van een brand de som van de bijdrage van het batterijpakket en de bijdrage van de rest van het voertuig (chassis):

$$HRR_{\text{totaal}} = HRR_{\text{batterijpakket}} + HRR_{\text{chassis}} \quad (1)$$

Hierin staat HRR voor de Heat Release Rate, ofwel het brandvermogen: de hoeveelheid warmte per seconde die door het vuur wordt gegenereerd, uitgedrukt in J/s of W. In deze studie definiëren wij het voertuigchassis als alle delen van het voertuig met uitzondering van het batterijpakket of de brandstof. Hieronder vallen dus carrosserie, interieur en bekleding.

2.3 Verband tussen HRR en energie-inhoud van een batterij

In de wetenschappelijke literatuur zijn twee overzichtsstudies gevonden, waarin een verband wordt gelegd tussen de elektrische-energie-inhoud van de batterij (E) en het maximale brandvermogen. De verbanden, weergegeven in onderstaande formules (2) en (3) zijn gebaseerd op datasets van batterijbranden.

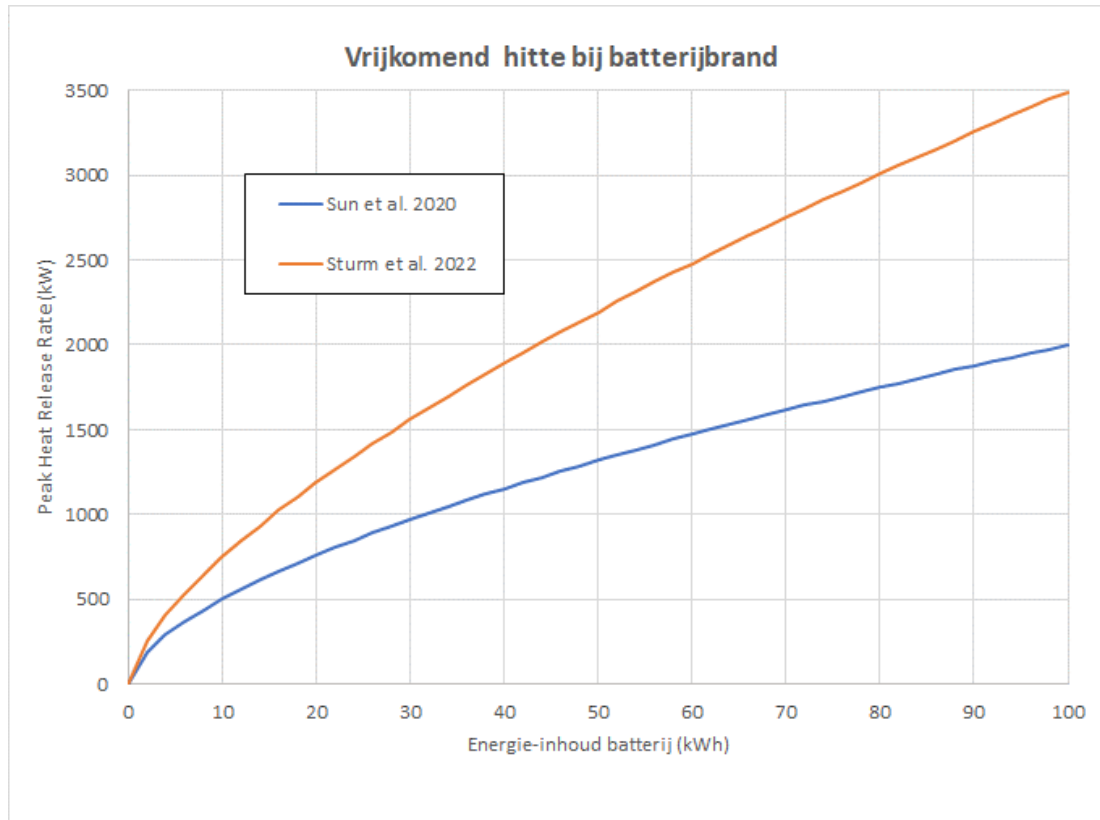
Sun et al. (2020):

$$\dot{Q}_p = 2 E^{0.6} \quad (2)$$

Sturm et al. (2022):

$$\dot{Q}_p = 1.56 E^{0.67} \quad (3)$$

Hierin is \dot{Q}_p de maximale waarde van het brandvermogen, ook wel aangeduid als de peak Heat Release Rate (pHRR). In Figuur 2.1 zijn beide verbanden grafisch weergegeven.



Figuur 2.1 Heat Release Rate van Li-ion batterijen

Merk op dat er een aanzienlijk verschil bestaat tussen beide verbanden. Het is niet duidelijk wat hiervan de oorzaak is, maar mogelijk wordt het veroorzaakt door een verschil in de gebruikte datasets. Ook wordt in de publicaties aangegeven dat er een grote mate van fluctuatie is in de data waarop de verbanden zijn gebaseerd.

2.4 Brandbare materialen in auto's

Volgens Euric (2020) bevat een moderne auto 12 - 15 massa% kunststoffen. Voor een auto van 1300 kg betekent dit 150 - 200 kg. Volgens ARN (2024) bevat een auto (elektrisch aangedreven of via een verbrandingsmotor) gemiddeld 208 kg plastic. In Tabel 2.1 staan de gebruikte kunststoffen vermeld. Ook zijn de verbrandingswaardes aangegeven. Op basis van deze gegevens is de totale verbrandingsenergie van (de kunststoffen van) een auto berekend, uitgaande van 200 kg kunststof. Ter vergelijking is ook de verbrandingsenergie van 40 liter benzine vermeld. Uit de tabel blijkt dat de brandstof circa 15 % uitmaakt van de totale verbrandingsenergie.

Tabel 2.1 Kunststoffen in auto's

Plasticsoort	Gewicht% in auto ¹	Massa bij 200 kg kunststof in auto	Verbrandingswaarde (MJ/kg)	Verbrandings-energie MJ
Polypropylene (PP)	35	70	45.8 ²	3206
Polyurethaan (PU)	19	38	28.9 ³	1098.2
Polyamide (PA)	11	22	26 ⁴	572
Polyvinylchloride (PVC)	9	18	17.95 ⁵	323.1
Acrylonitrile butadiene styrene (ABS)			39.84 ⁴	
Polystyrene (PS)			43.9 ⁴	
Polyethylene (PE)			47.74 ⁴	
Polyoxymethylene (POM)	36	52	17.4 ⁴	1794
Polycarbonate (PC)			31.3 ⁴	
Poly(methylmethacrylate) (PMMA)			26.8 ⁴	
Totaal aan kunststof				6993 (84%)
40 liter benzine	n.v.t.	30	43	1290 (16%)
Totaal aan brandbaar materiaal				8283

¹ [Euric \(2020\)](#)

² [stanford.edu](#)

³ [Kakae, 2004](#)

⁴ [Wiley Online Library, 2023](#)

⁵ [Polyvinyl chloride - Wikipedia](#)

2.5 Brandproeven met elektrische auto's

In Kang et al. (2023) worden brandproeven beschreven van een aantal personenauto's. De resultaten zijn samengevat in Tabel 2.2. Het betreft enkele varianten van één type SUV:

- > twee elektrische varianten met batterijpakketten van verschillende grootte (test 3 en 4)
- > één SUV met een conventionele verbrandingsmotor (test 5)
- > één SUV waarvan het batterijpakket is verwijderd (test 2)
- > het batterijpakket uit de SUV uit test 2 (test 1).

Tabel 2.2: Gegevens brandtest Kang et al. (2023)

Measured Parameters	Test 1	Test 2	Test 3	Test 4	Test 5
	64 kWh LIB pack	SUV body	SUV with 39 kWh LIB	SUV with 64 kWh LIB	ICE SUV
Peak heat release rate, pHRR (MW)	1.54	7.81	6.51	7.25	7.66
Time to pHRR (TTP) (min)	19 [#]	17	14 [*]	11 [*]	17
Total heat released, THR (GJ)	1.3	7.53	8.45	9.03	8.08
Peak heat fluxes during fully developed fire (kW/m ²)				40-60	40-60
Total weight loss (kg)	28	262	284	296	294

Test 1: brandtest van een voor 100 % opgeladen 64kWh Li-ion-batterijpakket uit de SUV van test 2.

Test 2: brandtest van de SUV zonder batterijpakket.

Test 3: Brandtest van hetzelfde type SUV met 100 % opgeladen 39kWh Li-ion-batterijpakket.

Test 4: brandtest van SUV incl. 100 % opgeladen 64kWh Li-ion-batterijpakket.

Test 5: brandtest van hetzelfde type SUV met verbrandingsmotor.

[#] meerdere pieken vanaf ca 12 min.

^{*} herhaaldelijk pieken zichtbaar waarbij HRR met 0,5- 1MW toenam.

Uit Tabel 2.2 blijkt dat de bijdrage van het 64 kWh Li-ion pakket aan de Total Heat Release circa 15 % bedraagt (1,30 / 9,03). Voor test 5 werd de hoeveelheid brandstof niet vermeld. Ervan uitgaande dat het verschil in THR tussen Test 2 en Test 5 een gevolg is van de brandstof in de tank, zou dit betekenen dat de tank circa 32 kg (38 liter) benzine of diesel heeft bevat, goed voor een THR van 1.4 GJ, ofwel (ook) zo'n 15 % van de totale GHR. Dit komt goed overeen met de berekeningen uit de vorige paragraaf. Dit betekent dat de invloed van de brandstof op het brandgedrag relatief klein is. Dit blijkt ook uit de waarden voor de peak Heat Release Rate (pHRR), waar de hoogste waarde werd gemeten bij het voertuig zonder brandstof. De waarden voor de Tijd tot pHRR (TTP) voor de voertuigen met batterijen liggen tussen de 11 en 14 minuten. De invloed van de batterijen was zichtbaar, doordat vanaf circa 11 minuten herhaaldelijk thermal-runaway-reacties plaatsvonden, die tot piekjes in de HRR leidden. Bij test 2 en 5, waar geen batterijen aanwezig waren, ontstonden deze piekjes niet.

De warmteflux werd gemeten in tests 4 en 5. Meetpunten bevonden zich op 1,27 m hoogte op 4 locaties rondom de auto: op 89 cm van de achterraut en beide zijritten en op 1,90 m van de voorruit. Gedurende de tijd dat de brand zich volledig ontwikkelde, werden piekwaarden van 40 - 60 kW/m² gemeten. Er zijn geen significante verschillen tussen de elektrische en benzineauto waargenomen.

De som van waarden voor de THR uit tests 1 en 2 komt redelijk overeen met de THR uit test 4, hetgeen in overeenstemming is met vergelijking (1).

Al met al lijkt er geen significant verschil te zijn tussen elektrische en conventioneel aangedreven (personen)auto's voor wat betreft de genoemde brandparameters THR, pHRR, TTP en warmteflux (warmtestraling). Alleen de herhaaldelijk optredende thermal-runaway-reacties zorgen er bij elektrisch aangedreven auto's voor dat de brand een minder geleidelijk verloop heeft en met tussenpozen pieken vertoont.

Voor grotere voertuigen zijn er weinig gegevens bekend. De fractie brandstof ten opzichte van de totale (brandbare) massa zal naar verwachting echter niet groter, en in vele gevallen (zoals bij goederentransport), zelfs lager zijn. Het is dan ook de verwachting dat ook voor grotere voertuigen de bijdrage van de brandstof (fossiel dan wel elektrisch) in het algemeen beperkt zal zijn, mogelijk zelfs kleiner dan voor personenauto's. Derhalve zal er in het vervolg van uit worden gegaan dat THR, pHRR en warmteflux van conventioneel aangedreven voertuigen ook representatief kunnen worden geacht voor elektrisch aangedreven voertuigen.

2.6 Brandvermogens van voertuigen

Experimenten met brandende voertuigen zijn veelal uitgevoerd in het kader van de veiligheid in of sterkte van tunnels, waarin HRR en TTP vaak belangrijke factoren zijn. Rijkswaterstaat hanteert de brandvermogens als gegeven in Tabel 2.3 (Rijkswaterstaat, 2007).

Tabel 2.3 Brandvermogens volgens (Rijkswaterstaat, 2007)

Type voertuig	pHRR (MW)
Kleine personenauto	2.5
Grote personenauto	5
Bestelbus	15
Bus	25
Vrachtauto leeg	25
Vrachtauto + brandbare goederen	50 of 100
Tankauto met benzine	200 - 300

In een eerdere literatuurstudie naar het brandgevaar van elektrische auto's (IFV, 2021) werd vermeld dat het brandvermogen van een personenauto gemiddeld tussen 4 tot 6 MW ligt en dat het bij elektrische auto's van een groter formaat kon oplopen tot 8 MW.

Xu et al. (2023) hebben uit een aantal brandtests met voertuigen gemiddelde waarden voor pHRR berekend. Deze waarden zijn weergegeven in Tabel 2.4.

Tabel 2.4 Gemiddelde waarden berekend uit Tabel 2.5

Types of vehicle	Range (min - max) van HRRmax (MW)	Average pHRR (MW)
Car	1,5 – 8,9	5,4
Bus	29 ~ 30	29,5
HGV (Heavy Goods Vehicle)	13 ~ 203	71,8

De enorme range voor vrachtauto's (HGV) is een gevolg van experimenten met verschillende hoeveelheden brandbare lading, zoals blijkt uit Tabel 2.5, waarin de gemeten waarden voor de HRR's van de brandexperimenten met vrachtauto's zijn weergegeven.

Tabel 2.5 Overzicht brandexperimenten met vrachtwagens (Xu et al. 2023)

Type of vehicles	Conditions (u is the longitudinal wind speed)	pHRR (MW)
Heavy Goods Vehicle (HGV)	Test of 10.9-ton HGV-trailer mock-up in Runehamar tunnel $u = 3$ m/s	203
	Test of 6.8-ton HGV-trailer mock-up in Runehamar tunnel $u = 3$ m/s	158
	Test of 8.5-ton HGV-trailer mock-up in Runehamar tunnel $u = 3$ m/s	125
	Test of 3.1-ton HGV-trailer mock-up in Runehamar tunnel $u = 3$ m/s	70
	Second Benelux tunnel test $u = 1 \sim 2$ m/s	25
	Second Benelux tunnel test $u = 0$ m/s	13
	Second Benelux tunnel test $u = 4 \sim 6$ m/s	19
	Second Benelux tunnel test $u = 6$ m/s	16
	EUREKA 499 test of mock-up HGV $u = 0.7$ m/s	17

De lading in de tests in de Runehamar-tunnel bestond uit houten pallets met een variatie aan brandbare koopmansgoederen. Een lineaire regressieanalyse van deze data levert de volgende vergelijking op:

$$pHRR = 15.3 M + 27 \quad (4)$$

Hierin is M de hoeveelheid (brandbare) lading in ton. Voor een lege trailer ($M=0$) leidt deze formule tot een pHRR van 27 MW. Het is niet duidelijk of de Runehamar mock-up representatief kan worden geacht voor een trailer met brandbare goederen, maar de waarde komt goed overeen met de pHRR van een lege vrachtauto volgens Rijkswaterstaat, te weten 25 MW (zie Tabel 2.3).

2.7 Beschouwing

Uit de literatuurstudie blijkt dat de batterijpakketten zo'n 15 % procent bijdragen aan de Total Heat Release. Dit is zo ongeveer dezelfde waarde als de bijdrage die een fossiele brandstof levert aan de THR van conventioneel aangedreven voertuigen. Invloed van de brandstof of batterijen op de peak Heat Release Rate (pHRR) kon in het geheel niet worden vastgesteld. Derhalve kan in principe voor de benadering van de (maximale) hittestraling afkomstig van voertuigen (ook) worden uitgegaan van conventioneel aangedreven voertuigen. Aangezien er geen experimenten van elektrische bussen of vrachtwagens zijn gevonden met bruikbare brandparameters zijn daarom gegevens van conventioneel aangedreven voertuigen gebruikt bij de modelberekeningen.

Verder bleek uit de studies dat het maximale brandvermogen voor een personenauto tussen de 5 en 8 MW ligt. Voor een bus is een gemiddelde waarde van 29,5 MW gevonden. Voor vrachtauto's werd een grote range gerapporteerd, voornamelijk als gevolg van de grote verschillen in de hoeveelheid lading. Op basis van experimenten in de Runehamar-tunnel kan het effect van de lading ingeschat worden. Dit zou een waarde van 27 MW voor een leeg Heavy Goods Vehicle opleveren. Deze waarde komt goed overeen met de 25 MW die Rijkswaterstaat hanteert voor een lege vrachtauto.

3 Rekenmodel

3.1 Inleiding

In dit hoofdstuk wordt een rekenmodel gepresenteerd waarmee de warmtestraling, of, meer precies, de radiatieve warmteflux (in kW/m²), als functie van de afstand tot een (voertuig-) brand kan worden bepaald. Het rekenmodel is uitgewerkt en geprogrammeerd in de programmeertaal Python, die sinds enkele jaren veel wordt gebruikt in het natuurwetenschappelijk onderwijs en onderzoek.

Voor de rekenmodellen is gebruikgemaakt van het *Gele Boek* (Yellow Book, 2005) en het *SFPE Handbook of Fire Protection Engineering* (SFPE, 2016). Beide boeken zijn een standaard voor effectmodellering. Daarnaast zijn twee wetenschappelijke artikelen gevonden waarin berekeningen voor de warmtestraling van brandende voertuigen staan beschreven. Deze informatie vormde het uitgangspunt voor de ontwikkeling van het rekenmodel.

Basis voor de rekenmethode is een semi-empirische benadering, waarbij gebruik is gemaakt van experimenteel bepaalde grootheden. Startpunt hierbij is de Heat Release Rate (HRR of, in dit hoofdstuk, $\dot{Q} = \frac{dQ}{dt}$), dat wil zeggen de energie (verbrandingswarmte) die per seconde vrijkomt bij een autobrand, ofwel het brandvermogen.

Om tot warmtestralingsfluxen te komen, zijn een aantal aanvullende parameters nodig, te weten:

- > Stralingsfractie: de fractie van de totale energie die als straling wordt afgegeven.
- > Transmissiviteit van lucht: het deel van de straling dat niet door de lucht wordt geabsorbeerd.
- > Viewfactor: een combinatie van vorm en grootte van het stralend oppervlak en de afstand tot het stralend oppervlak.
- > Eventueel een gemeten warmteflux ter validatie.

Voor de stralingsfractie en transmissiviteit is van waarden uitgegaan uit de literatuur. De viewfactor wordt in het rekenprogramma bepaald aan de hand van voertuig- en brandkenmerken. Hiervoor is gebruikgemaakt van het zogenaamde Solid Flame Model (zie ook: NIPV, 2024 en Yellow Book, 2005). Dat wil zeggen dat de brand wordt beschouwd als een uniform stralend oppervlak. Dit model is ook hier toegepast.

3.2 Solid Flame Model

De formules die de basis vormen voor het model zijn als volgt. Allereerst wordt het (theoretische) bronvermogen per oppervlakte-eenheid (SEP_{theor} , met SEP = Surface Emissive Power) bepaald aan de hand van de Heat Release Rate (\dot{Q}) en het oppervlak van het stralend lichaam (vlamlichaam) A_f :

$$SEP_{theor} = \frac{\dot{Q}}{A_f} \quad (5)$$

Het stralingsvermogen is een deel van het totale bronvermogen. De stralingsfractie χ_r is hierbij een (experimentele) waarde die bepaald dient te worden. Deze is van (grote) invloed op de uitkomsten van de studie. Voor koolwaterstoffen wordt vaak een waarde tussen de 0,3 en 0,4 gekozen (Yellow Book, 2005). In de wetenschappelijke literatuur over voertuigbranden, waarin een model van vergelijkbare aard wordt gepresenteerd, worden hiervoor (redelijk conservatieve) waarden gehanteerd tussen de 0,4 en 0,5 (Zhang et al. 2023; Hu et al. 2020). In de voorbeeldberekeningen in hoofdstuk 4 zal ook worden uitgegaan van een stralingsfractie tussen 0,4 en 0,5.

De bronstraling wordt vastgesteld aan de hand van het totale bronvermogen vermenigvuldigd met de stralingsfractie:

$$\dot{Q}_r = \chi_r \dot{Q} \quad (6)$$

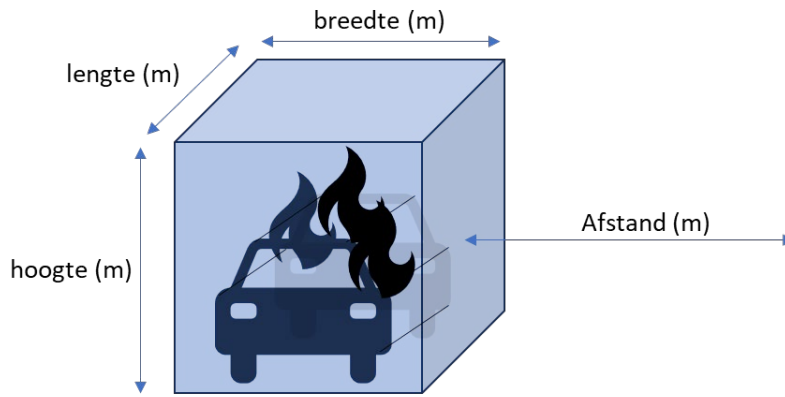
Om de warmtestraling op een afstand van de bron $\Phi_q(r)$ te bepalen, wordt de bronstraling vermenigvuldigd met de viewfactor $F(r)$ en de transmissiviteit van lucht $\tau_{air}(r)$. De viewfactor is de fractie van het gezichtsveld van een waarnemer op een punt \vec{r} van de bron die wordt ingenomen door het stralend oppervlak. De viewfactor wordt berekend met behulp van afstands- en vormafhankelijke formules, die zijn beschreven in de literatuur. In de praktijk wordt hierbij het vlamoppervlak gezien als het stralende oppervlak, bepaald door de vlamhoogte en de geometrie van het brandende object (hier dus het voertuig) (Yellow Book, 2005; Zhang et al. 2023; Hu et al. 2020). De transmissiviteit hangt ook af van de afstand tot de bron en tevens van de luchtvochtigheid.

$F(r)$ en $\tau_{air}(r)$ nemen af met groter wordende afstand. Hierdoor neemt de warmtestraling af als functie van de afstand. Het verband tussen de warmtestraling en de afstand tot het stralend (brandend) oppervlak is weergegeven in de volgende formule:

$$\Phi_q(\vec{r}) = \frac{\chi_r \dot{Q} F(\vec{r}) \tau_{air}(|\vec{r}|)}{A_f} \quad (7)$$

3.3 Geometrie van het voertuig

Het Solid Flame Model wordt vaak toegepast om de straling van vloeistofplasbranden te berekenen. Voor de bepaling van het vlamoppervlak wordt dan in het algemeen een cilinder of kegel als geometrische vorm gehanteerd. Voor een brandend voertuig is deze geometrie echter minder geschikt. In het model zullen de vlammen van het elektrisch voertuig worden beschouwd als een balk, gekarakteriseerd door te kiezen waarden voor lengte, hoogte en breedte (zie Figuur 3.1). De zijvlakken worden als stralende oppervlakken gezien.



Figuur 3.1 Beschouwing van elektrisch voertuig als balk ter illustratie

De lengte en breedte van de balk zijn in de studie gelijk genomen aan de voertuiglengte en de voertuigbreedte. De hoogte wordt benaderd met een formule voor de vlamhoogte, beschreven in de volgende paragraaf.

3.4 Vlamhoogte

Volgens Heskestad (1984) kan de vlamhoogte h als volgt worden berekend uit het grondoppervlak van de vlammen en de Heat Release Rate:

$$h = -1,02 D + a\dot{Q}^{2/5} \quad (8)$$

Hierin is D de zogeheten effectieve diameter van het grondoppervlak van de vlam (uitgelegd in het blauwe kader hieronder) en \dot{Q} de Heat Release Rate. Volgens Heskestad (1984) is deze formule bruikbaar voor plasbranden en voor brandbare goederen (genoemd worden onder meer een stapel van 4-4,5 m aan koopmansgoederen en 0,3-3 m hoge stapels houtpallets). De vlamhoogte wordt gemeten vanaf de basis van de vlammen. Heskestad adviseert een waarde voor a van 0,235, tenzij er sprake is van extreme omgevingscondities (temperatuur, druk). Hij geeft ook aan dat de waargenomen vlamhoogte wat kan afwijken als gevolg van wind, en indien de exacte vorm van het brandoppervlak sterk afwijkt van een cirkelvorm.

Combineren van formule 7 en 8, zoals uitgelegd in het blauwe kader hieronder, leidt tot onderstaande formule (14) voor het verband tussen de warmtestralingsflux $\phi_q(\vec{r})$ en de afstand \vec{r} tot het vlamoppervlak (hier tot het voertuig):

$$\phi_q(\vec{r}) = \frac{\chi_r \dot{Q} F(\vec{r}) \tau_{air}(\vec{r})}{2lb + 2(b+l) \left(a\dot{Q}^{2/5} - 1.02 \sqrt{\frac{alb}{\pi}} \right)} \quad (14)$$

In de formule van Heskestad is D de effectieve diameter van het grondoppervlak van de vlam. Dat wil zeggen: de diameter van een cirkel van gelijke oppervlakte als het gekozen grondoppervlak. Zodat geldt: $\frac{\pi D^2}{4} = A_{bodem}$. In deze situatie kiezen we een rechthoek als grondoppervlak, zodat $A_{bodem} = lb$. Hierbij is l de lengte van de rechthoek en b de breedte. De vergelijking oplossen levert:

$$D = \sqrt{\frac{4lb}{\pi}} \quad (9)$$

Ingevuld in de formule voor vlamhoogte, verkrijgen we:

$$h = -1,02 \sqrt{\frac{4lb}{\pi}} + a\dot{Q}^{2/5} \quad (10)$$

Het vlamoppervlak is het oppervlak van een balk:

$$A_f = 2lb + 2lh + 2hb \quad (11)$$

Met de zojuist geïntroduceerde vlamhoogte kan deze in termen van het brandvermogen en de lengte en breedte van het voertuig worden berekend.

$$A_f = 2lb + 2l \left(-1,02 \sqrt{\frac{4lb}{\pi}} + a\dot{Q}^{2/5} \right) + 2b \left(-1,02 \sqrt{\frac{4lb}{\pi}} + a\dot{Q}^{2/5} \right) \quad (12)$$

Ofwel:

$$A_f = 2lb + 2(b+l) \left(a\dot{Q}^{2/5} - 1,02 \sqrt{\frac{4lb}{\pi}} \right) \quad (13)$$

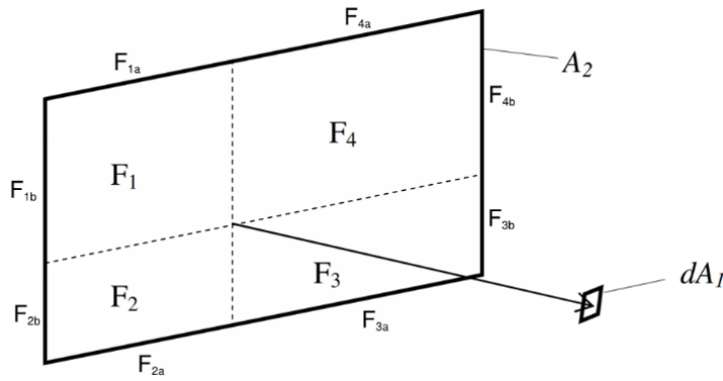
Waarmee we uiteindelijk een formule voor de warmtestralingsflux in termen van brandvermogen, voertuiglengte en voertuigbreedte hebben verkregen:

$$\phi_q(\vec{r}) = \frac{\chi_r \dot{Q} F(\vec{r}) \tau_{air}(\vec{r})}{2lb + 2(b+l) \left(a\dot{Q}^{2/5} - 1,02 \sqrt{\frac{4lb}{\pi}} \right)} \quad (14)$$

3.5 Bepaling van de viewfactor

3.5.1 Viewfactor loodrecht op een plat vlak

De gehanteerde viewfactor is gebaseerd op het *Handbook of Fire Protection Engineering*, 5th Edition (SFPE, 2016) en een spreadsheet uit ons archief. Hierin kan voor elk punt loodrecht op een groot vlak de viewfactor worden berekend, door het grote vlak in vier deelvlakken op te splitsen (zie Figuur 3.2). De viewfactoren van elk van de afzonderlijke deelvlakken worden daarna bij elkaar opgeteld. Het punt dA_1 is op 1 meter (leefhoogte) vanaf de grond gekozen.



Figuur 3.2 Bepaling van de viewfactor op een punt dA_1 loodrecht op een plat vlak A_2

De totale viewfactor wordt berekend door de bijdragen van elk van de deelvlakken bij elkaar op te tellen.

$$F = \sum_{i=1}^4 F_i \quad (15)$$

$$F = \sum_{i=1}^4 \frac{1}{2\pi} \left[\frac{X_i}{\sqrt{1+X_i^2}} \arctan\left(\frac{Y_i}{\sqrt{1+X_i^2}}\right) + \frac{X_i}{\sqrt{1+Y_i^2}} \arctan\left(\frac{Y_i}{\sqrt{1+Y_i^2}}\right) \right] \quad (16)$$

Waarbij geldt voor de deelvlakken 1 en 4:

$$X_1 = \frac{F_{1b}}{|dA_1 - A_2|}; Y_1 = \frac{F_{1a}}{|dA_1 - A_2|} \quad (17)$$

$$X_4 = \frac{F_{4b}}{|dA_4 - A_2|}; Y_4 = \frac{F_{4a}}{|dA_1 - A_2|} \quad (18)$$

En voor de deelvlakken 2 en 3:

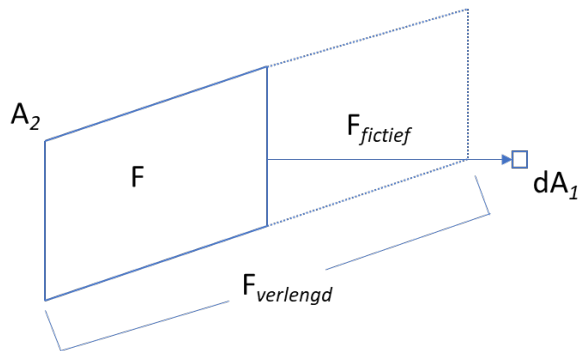
$$X_2 = \frac{F_{2a}}{|dA_1 - A_2|}; Y_2 = \frac{F_{2b}}{|dA_1 - A_2|} \quad (19)$$

$$X_3 = \frac{F_{3a}}{|dA_1 - A_2|}; Y_3 = \frac{F_{3b}}{|dA_1 - A_2|} \quad (20)$$

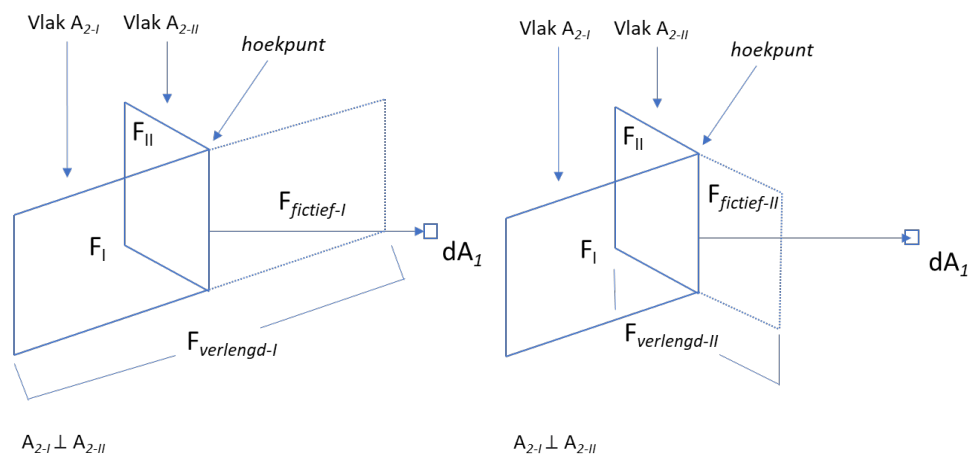
De afstand $|dA_1 - A_2|$ wordt verderop in het rapport ook als $|r|$ genoteerd en refereert naar de loodrechte afstand tot het stralend vlak.

3.5.2 Viewfactor in de hoekpunten

Om de warmtestraling in de hoekpunten (vertex) te bepalen, is het stralend vlak verlengd met een zogeheten fictief vlak, waardoor een verlengd stralend vlak ontstaat. Vervolgens wordt de viewfactor van het verlengde vlak berekend, waarna de viewfactor van het fictieve vlak wordt afgetrokken. Deze berekening wordt herhaald voor de korte en de lange zijde, die loodrecht op elkaar staan.



Figuur 3.3 Principe van een verlengd vlak en fictief vlak



Figuur 3.4 Bepaling van viewfactor in een punt dA_1 schuin voor twee loodrecht op elkaar staande vlakken A_{2-I} en A_{2-II}

$$F_{\text{hoekpunt}} = F_I + F_{II} \quad (21)$$

$$F_{\text{hoekpunt}} = (F_{\text{verlengd-I}} - F_{\text{fictief-I}}) + (F_{\text{verlengd-II}} - F_{\text{fictief-II}}) \quad (22)$$

3.6 Transmissiviteit

Voor de transmissiviteit van lucht is de (experimentele) formule uit Zhang et al. (2023) gekozen, die gelijk is aan die uit het *Gele Boek*. Dicht bij het vlamoppervlak (tot 0,8 m) is de transmissiviteit gelijk aan 1. De afstand tot het stralend oppervlak is $|\mathbf{r}|$ en ρ_w is de dampdruk van het water in de atmosfeer. Voor met waterdamp verzadigde lucht bij 25 °C geldt: $\rho_w = 3169$ Pa (CRC, 2004).

$$\tau_{\text{air}} = 2,02 (\rho_w |\mathbf{r}|)^{-0,09} \text{ als } |\mathbf{r}| > 0,8$$

$$\tau_{\text{air}} = 1 \text{ als } |\mathbf{r}| \leq 0,8$$

3.7 Onzekerheidsanalyse

Gebruikmakend van de, in zeer beperkte mate aanwezige, (experimentele) data zijn bij de doorrekening van de drie voertuigbranden (hoofdstuk 4) onzekerheidsberekeningen uitge-

voerd. Hiermee kan worden berekend hoe een onzekerheid in meetwaarden (invoerparameters) propageert tot de onzekerheid in de uitkomst. In deze onzekerheidsberekening zijn onzekerheden in de pHRR, de stralingsfractie en de vlamhoogte beschouwd. De onzekerheid is in de warmtestralingscontouren weergegeven als een onzekerheidsbalkje op het punt waar de warmtestraling maximaal is. Dat is in het midden van de lange zijde van het voertuig. Het geeft de onzekerheid in de afstand weer waarop een bepaald stralingsniveau aanwezig is. De wijze waarop de onzekerheid wordt berekend, is in onderstaand blauw kader weergegeven.

Voor een functie $G(x_1, \dots, x_N)$ die afhankelijk is van N parameters of variabelen x_i kan de algemene formule voor de propagatie van onzekerheden worden toegepast, zoals hieronder weergegeven. Hierin geeft $S[G]$ de onzekerheid weer in de uitkomst van de formule, waarbij $S[x_i]$ de onzekerheid in de gemeten (of ingevoerde) variabelen is.

$$S[G(x_1, \dots, x_N)] = \sqrt{\sum_{i=1}^N \left(\frac{\partial G}{\partial x_i}\right)^2 S[x_i]^2} \quad (23)$$

Nu vullen we voor G de formule voor de warmtestralingsflux in:

$$G = \phi_q(\vec{r}) = \frac{\chi_r \dot{Q} F(\vec{r}) \tau_{air}(\vec{r})}{A_f} = \frac{\chi_r \dot{Q} F(\vec{r}) \tau_{air}(\vec{r})}{2lb + 2(b+l) \left(a \dot{Q}^{2/5} - 1.02 \sqrt{\frac{4lb}{\pi}} \right)} \quad (24)$$

Aangezien we de propagatie van de onzekerheid in de stralingsfractie χ_r , het brandvermogen \dot{Q} en de Heskestad-constante a willen berekenen, vullen we die drie termen in.

$$S[\phi_q] = \sqrt{\left(\frac{\partial \phi_q}{\partial \chi_r}\right)^2 S[\chi_r]^2 + \left(\frac{\partial \phi_q}{\partial \dot{Q}}\right)^2 S[\dot{Q}]^2 + \left(\frac{\partial \phi_q}{\partial a}\right)^2 S[a]^2} \quad (25)$$

Dit ingevuld, berekend en herschreven in termen van A_f geeft de formule voor de onzekerheid in de warmtestralingsflux:

$$S[\phi_q] = \sqrt{\left(\frac{\dot{Q} F \tau_{air}}{A_f}\right)^2 S[\chi_r]^2 + \left(\frac{\chi_r F \tau_{air} (A_f^{-\frac{4}{5}} a (b+l) \dot{Q}^{2/5})}{A_f^2}\right)^2 S[\dot{Q}]^2 + \left(\frac{2(b+l) \chi_r F \tau_{air} \dot{Q}^{7/5}}{A_f^2}\right)^2 S[a]^2} \quad (26)$$

3.8 Beschouwing

In dit hoofdstuk is een formule afgeleid waarmee het warmtestralingsniveau (kW/m²) als functie van de afstand tot een brandend voertuig kan worden berekend. De invoergegevens bestaan uit de afmetingen van het voertuig (lengte en breedte), de Heat Release Rate (brandvermogen als het voertuig in brand staat in MW) en de stralingsfractie. Ook is een methode beschreven waarmee kan worden berekend hoe de onzekerheid in de invoergegevens doorwerkt op de berekende stralingsniveaus. Dit kan met name nuttig zijn bij het bepalen van de invloed van de onzekerheid in Peak Heat Release Rate en de stralingsfractie op de resultaten, aangezien voor deze parameters weinig (experimentele) data aanwezig zijn. Ook is er wat onzekerheid in de vlamhoogte. Deze is (zoals bepaald met formule (8)) enigszins afhankelijk van de aard van de brandende goederen, de vorm van het brandoppervlak en de omgevingscondities. Bedenk ook dat formule (8) is bepaald op basis

van ongehinderde vlammen, dus voor vlammen die niet worden gehinderd of afgebogen door obstructies, zoals bijvoorbeeld een (af)dak of plafond.

4 Warmtestralingscontouren

4.1 Inleiding

De berekening in het vorige hoofdstuk is in Python geprogrammeerd en herhaaldelijk uitgevoerd voor alle punten rondom het voertuig. Dit levert de warmtestralingscontouren op. De broncode hiervan staat in Bijlage 2. De gehanteerde constanten zijn redelijk conservatief en representeren in die zin een worst-case-scenario. De constanten kunnen in principe naar behoefte vrij worden aangepast om andere scenario's te berekenen. In dit hoofdstuk worden de resultaten van drie scenario's gepresenteerd. In overleg met de begeleidingscommissie is ervoor gekozen om de resultaten weer te geven in de vorm van warmtestralingscontouren in het xy-vlak (bovenaanzicht) rondom drie typen voertuigen: personenauto (SUV), stadsbus en vrachtwagen met lading.

Voor warmtestralingsniveaus worden in de omgevingsveiligheid vaak de grenzen van 35 kW/m², 10 kW/m² en 4 kW/m² gehanteerd. Uitgangspunt hierbij is dat een warmtestralingsniveau van 35 kW/m² of meer altijd tot letaal letsel zal leiden. Ook wordt ervan uitgegaan dat bij dit stralingsniveau secundaire branden ontstaan. Bij blootstelling aan stralingsniveaus van minder dan 10 kW/m² zal men in het algemeen nog in staat zijn om te vluchten, zodat geen letaal letsel zal optreden. Dit stralingsniveau is ook de grenswaarde voor het brandaandachtsgebied in het *Besluit kwaliteit leefomgeving* (Bkl) (PGS-29; RIVM). 4 kW/m² wordt als grens gehanteerd voor het optreden van de (bedrijfs)brandweer met beschermende brandweerkleding, die voldoet aan NEN-EN 469.

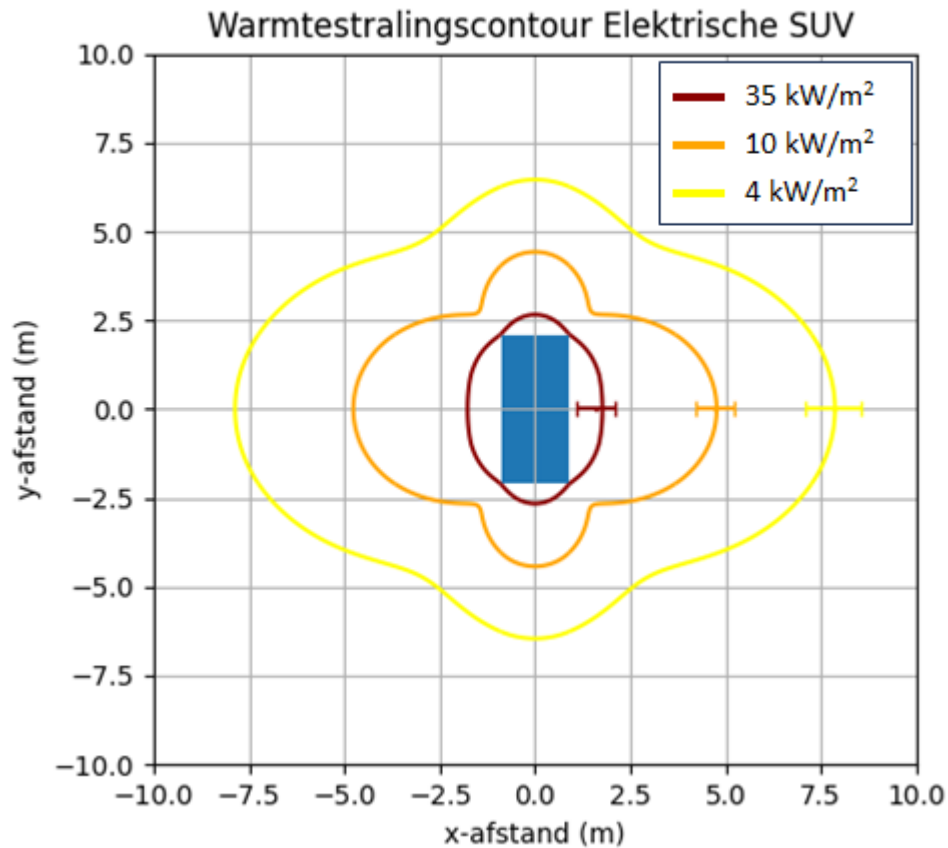
Voor alle berekeningen geldt:

- > Contouren zijn bepaald op 1 meter hoogte.
- > Contouren voor 35, 10 en 4 kW/m² zijn weergegeven.
- > De getoonde warmtestralingscontouren zijn gebaseerd op de maximale waarden van de HRR (pHRR) van de geselecteerde autobranden. Na de piek in het brandvermogen zal de warmtestraling afnemen in de tijd.
- > De parameters van elk scenario zijn weergegeven in een tabel onder elke warmtestralingscontour.
- > De onzekerheidsmarge is weergegeven met horizontale balkjes als één standaardafwijking op de stralingscontouren. Deze is gebaseerd op de spreiding in de parameters die in de bijbehorende tabel is weergegeven.

4.2 Elektrische personenauto - SUV

Voor de berekening van de warmtestralingscontouren van de personenauto is gebruikgemaakt van de HRR uit het brandexperiment van Kang et al. 2023. De warmtestralingscontouren van een elektrische personenauto (SUV) met een batterijpakket van 64 kWh zijn in Figuur 4.1 weergegeven. De onzekerheidsmarge in de pHRR is gebaseerd op de data uit de brandtesten van elektrische voertuigen van Kang et al. (2023) en Sturm et al. (2022). Voor de stralingsfractie is uitgegaan van een onzekerheidsmarge van $\pm 0,05$ rondom 0,45

(het gemiddelde van 0,4 en 0,5; zie paragraaf 3.2). Voor a is een foutenmarge van $\pm 10\%$ gehanteerd (zie paragraaf 3.4).

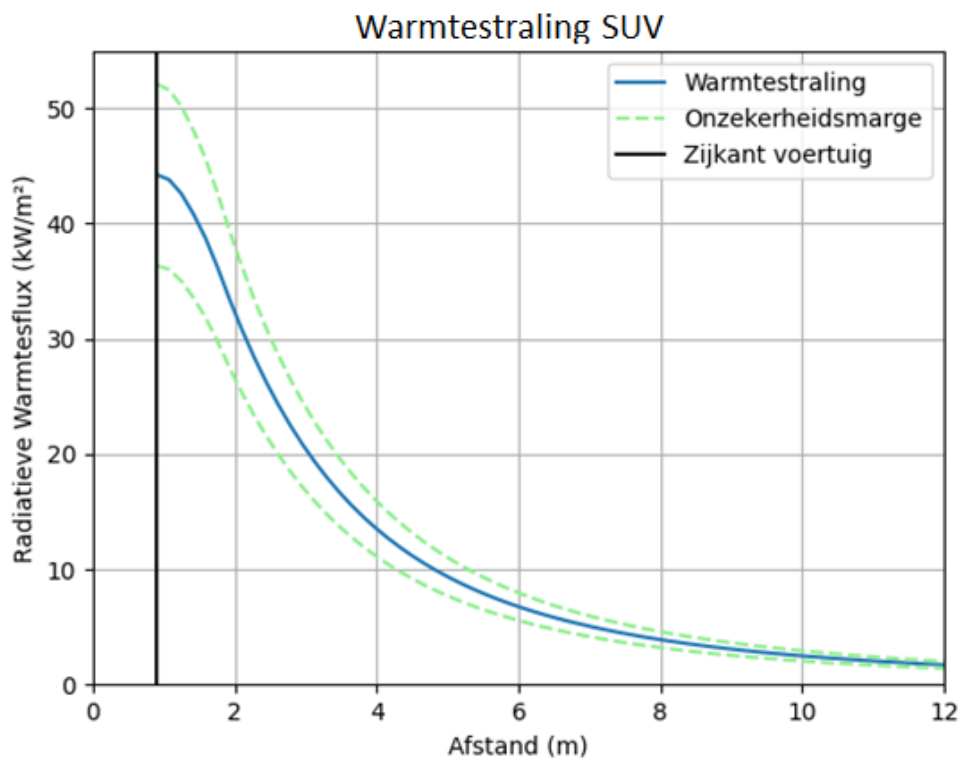


Figuur 4.1 Warmtestralingscontour elektrische SUV

Tabel 4.1 Parameters berekening elektrische SUV (Kang et al. 2023; Sturm et al. 2022)

Parameter	Weergegeven contour SUV \pm onzekerheidsmarge
Peak Heat Release Rate	$7,25 \pm 0,7$ MW
Stralingsfractie	$0,45 \pm 0,05$
a (zie formule (8))	$0,235 \pm 10\%$
Lengte SUV	4,2 m
Breedte SUV	1,8 m

In Figuur 4.2 is de warmtestraling als functie van de afstand weergegeven op het punt naast de auto waar de warmtestraling maximaal is, namelijk in het midden van de lange zijde. Het midden van de auto ligt op $x, y = 0,0$. De zijkant van de auto ligt op $x = 0,9$ m. De vlamlengte bedraagt circa 6 m.

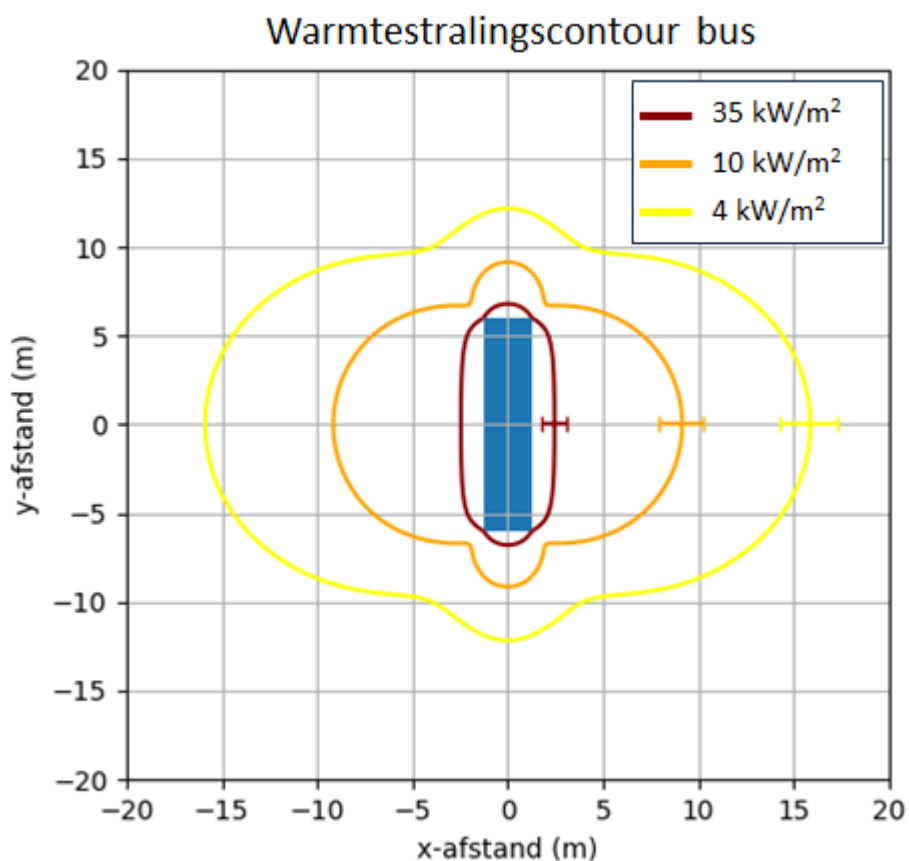


Figuur 4.2 Warmtestraling versus afstand voor de SUV

4.3 Bus

In Figuur 4.3 is de warmtestralingscontour van een bus weergegeven. De HRR in de bijbehorende Tabel 4.2 is afkomstig uit experimenten met conventionele stadsbussen uit Xu et al. (2023) (Tabel 2.4). In Figuur 4.4 is de warmtestraling als functie van de afstand weergegeven op het punt naast het voertuig waar de warmtestraling maximaal is, namelijk in het midden van de lange zijde. De zijkant van de bus ligt op $x = 1,275$ m.

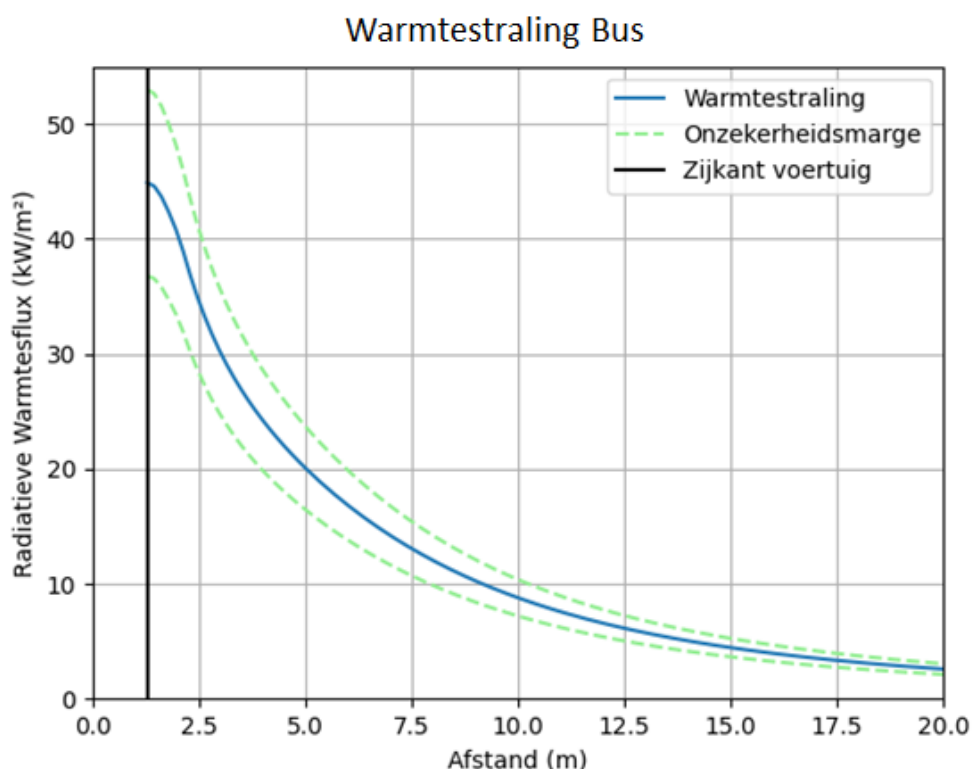
De onzekerheid in de pHRR is kleiner dan bij het scenario SUV uit de vorige paragraaf. Dit wordt veroorzaakt door het feit dat de onzekerheid gebaseerd is op de (slechts) twee, dicht bij elkaar liggende, waardes uit Xu et al. (2023) (Tabel 2.4). Op het moment van schrijven van voorliggend rapport waren er geen andere data van experimenten met busbranden voorhanden. Voor de stralingsfractie en a geldt hetzelfde als bij de SUV. De vlammen kunnen zo'n 10 m hoog worden.



Figuur 4.3 Warmtestralingscontour bus

Tabel 4.2 Parameters Berekening bus (Xu et al. 2023)

Parameter	Weergegeven contour bus ± onzekerheidsmarge
Peak Heat Release Rate	29,5 ± 0,500 MW
Stralingsfractie	0,45 ± 0,05
a (zie formule (8))	0,235 ± 10%
Lengte bus	12 m
Breedte bus	2,55 m

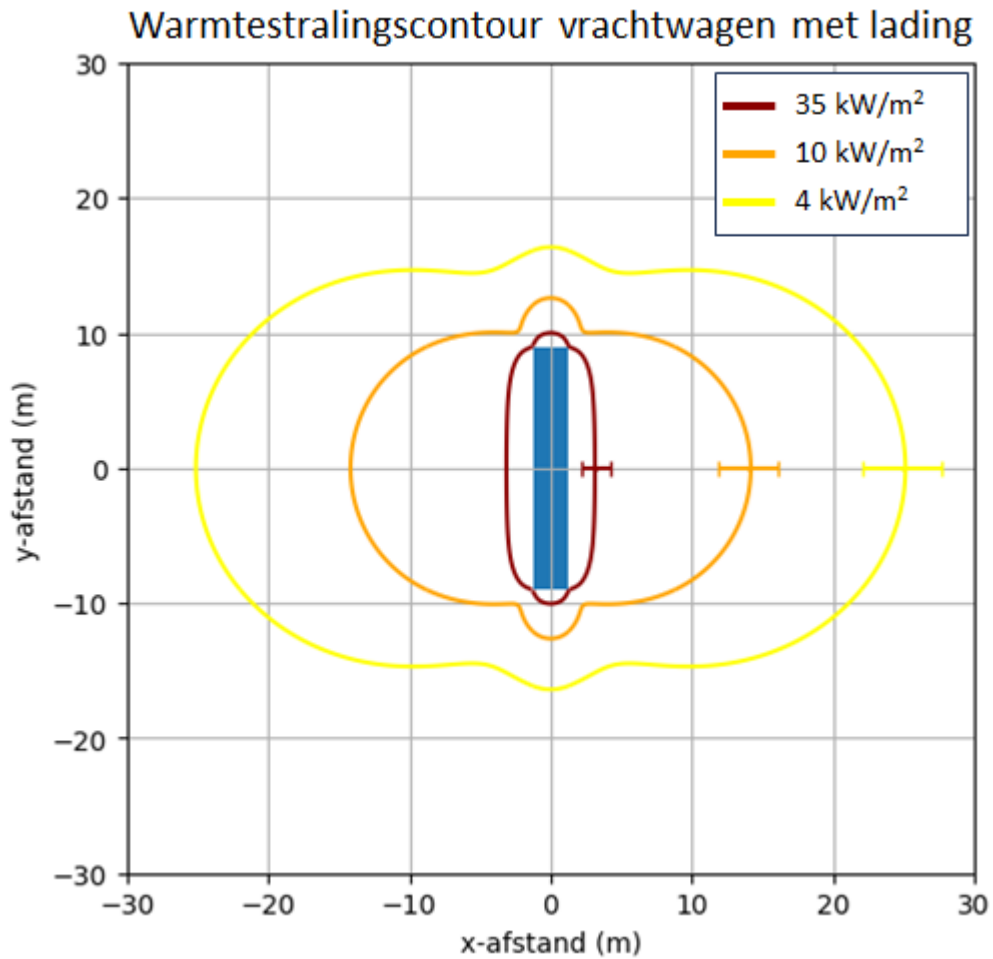


Figuur 4.4 Warmtestraling versus afstand bus

4.4 Vrachtwagen met lading

De warmtestralingscontouren voor de vrachtwagen met lading zijn weergegeven in Figuur 4.5. In Figuur 4.6 is de warmtestraling als functie van de afstand weergegeven op het punt naast het voertuig waar de warmtestraling maximaal is, namelijk in het midden van de lange zijde. De zijkant van de vrachtwagen ligt op $x = 1,275$ m.

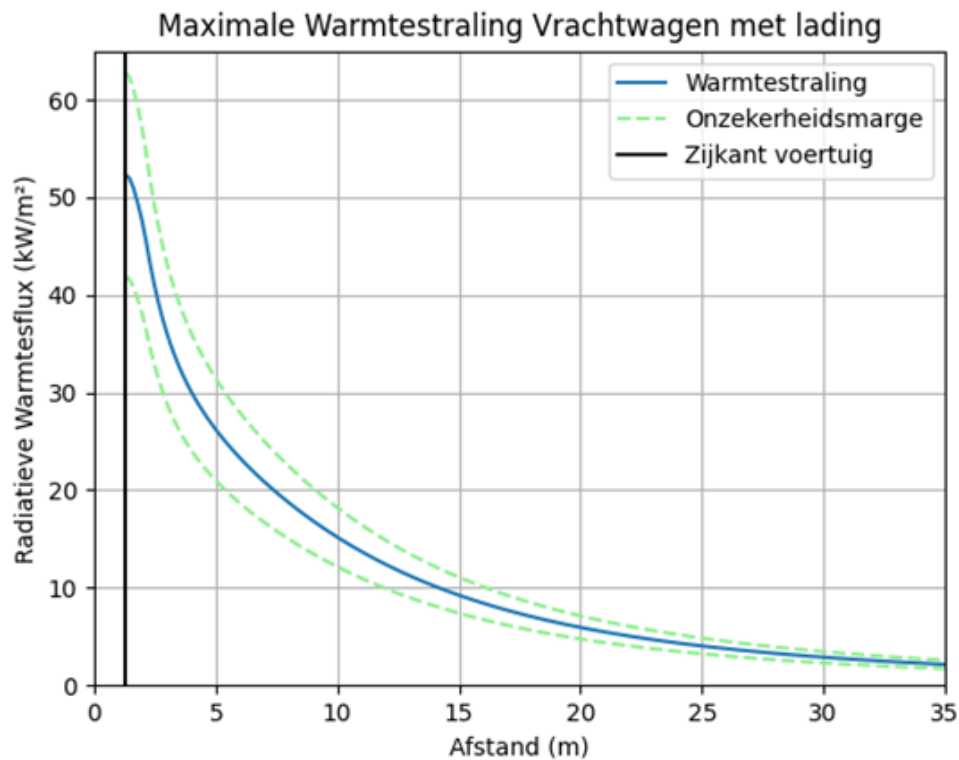
Voor het scenario van de vrachtwagen zijn brandexperimenten van Rijkswaterstaat en de studie van Xu et al. (2023) geraadpleegd. Hierin was 71,8 MW de gemiddelde waarde van de pHRR (Tabel 2.4). De vlamhoogte volgens formule (8) is dan ca 15 m. In de studie van Xu et al. 2023 staat echter een zeer grote spreiding vermeld. In dit geval wordt de spreiding met name veroorzaakt door de hoeveelheid en het type lading. In overleg met de begeleidingscommissie is besloten om één scenario van een vrachtwagen met lading te berekenen. Er is hier gekozen voor de gemiddelde waarde uit Xu et al. (2023) met een aangenomen onzekerheid van 20 %. Deze waarden liggen binnen de HRR's van Rijkswaterstaat, 2007. Voor het scenario van een lege vrachtwagen zijn de HRR's vergelijkbaar met die van een bus en kan paragraaf 4.3 als uitgangspunt dienen. Voor de stralingsfractie en a geldt hetzelfde als bij de SUV en de bus.



Figuur 4.5 Warmtestralingscontour vrachtwagen met lading

Tabel 4.3 Parameters Berekening vrachtwagen met lading (Xu et al. 2023; Rijkswaterstaat 2007)

Parameter	Weergegeven contour vrachtwagen met lading ± onzekerheidsmarge
Peak Heat Release Rate	71,8 ± 14,4 MW
Stralingsfractie	0,45 ± 0,05
<i>a</i> (zie formule (8))	0,235 ± 10%
Voertuiglengte	18 m
Voertuigbreedte	2,55 m



Figuur 4.6 Warmtestraling versus afstand Vrachtwagen met lading

4.5 Beschouwing

Met het ontwikkelde model zijn warmtestralingscontouren berekend voor een SUV, een bus, en een vrachtauto. De afstanden waarop deze waarden worden bereikt (op 1 m hoogte en ter hoogte van de hartlijn van de lange zijde van het voertuig) zijn weergegeven in Tabel 4.4. Ook de berekende range (min – max) in het maximale brandvermogen, de stralingsfractie en de vlamhoogteberekening die het gevolg is van de onzekerheden, is weergegeven (één standaardafwijking). Ten aanzien van de vrachtwagen moet worden opgemerkt dat de berekeningen gelden voor een ‘gemiddelde’ vrachtwagen met brandbare lading. Voor (grote) opleggers met veel brandbare lading kan deze afstand aanzienlijk groter zijn. Het is uiteraard mogelijk om berekeningen uit te voeren voor ander posities rondom het voertuig en voor andere hoogtes.

Tabel 4.4 Afstanden en onzekerheidsmarge (min - max) waarop bepaalde stralingsniveaus aanwezig zijn, vanaf de zijkant van het voertuig op 1 m hoogte

Afstanden in m	35 kW/m ²	10 kW/m ²	4 kW/m ²
SUV	0,9 (0,2 – 1,2)	3,9 (3,3 – 4,3)	7,0 (6,2 – 7,6)
Bus	1,2 (0,5 – 1,8)	7,9 (6,6 – 9,0)	14,6 (13,0 – 16,1)
Vrachtwagen met lading	1,9 (1,0 – 2,9)	12,9 (10,6 – 14,8)	23,8 (20,9 – 26,5)

5 Conclusie

In deze studie is een rekenmodel opgesteld waarmee de warmtestraling van voertuigbranden kan worden berekend. Het rekenmodel is gebaseerd op een combinatie van het Solid Flame Model en de Formule van Heskestad voor vlamhoogte. Hierbij is het voertuig vereenvoudigd tot een (rechthoekige) balk en wordt de vlam beschouwd als een uniform stralend vlak. De combinatie van deze twee modellen maakt het mogelijk om op basis van de Heat Release Rate (HRR) en voertuigafmetingen (lengte en breedte) de warmtestraling als functie van de afstand tot het voertuig te berekenen. Ook kunnen met het model warmtestralingscontouren rondom het voertuig worden afgebeeld. Er dient wel een stralingsfractie te worden aangenomen, omdat hiervoor geen eenduidige waarde voor voertuigbranden in de literatuur is gevonden. Ook het gebruik van de formule van Heskestad gaat met enige onzekerheid gepaard, aangezien deze enigszins wordt beïnvloed door omgevingscondities en de vorm van het brandoppervlak.

In de literatuur zijn wel experimentele gegevens van voertuigbranden gevonden, waarmee een beeld gevormd kan worden van realistische waarden van de HRR van elektrische voertuigen. Het blijkt dat er tussen elektrische en conventionele voertuigbranden van vergelijkbare grootte geen significant verschil is ten aanzien van het (maximale) brandvermogen of de totale verbrandingsenergie (THR). Circa 15 % van de THR bij voertuigbranden is afkomstig van de brandstof of het batterijpakket. De rest van het voertuig is dominant in het (piek)brandvermogen en de THR. Voor vrachtwagens geldt dat (eventuele) brandbare lading al gauw het brandvermogen domineert.

Het rekenmodel maakt het uitwerken van meerdere scenario's naar behoefte mogelijk, bijvoorbeeld als meer data beschikbaar komen. In deze studie zijn scenario's uitgewerkt voor een elektrische SUV, een bus en een vrachtwagen met lading. Voor de getoonde warmtestralingscontouren (35, 10 en 4 kW/m²) zijn de HRR-waarden op de piek van de brand gebruikt. Het model is echter toepasbaar voor elke waarde van de HRR. Ook contouren voor andere stralingsniveaus kunnen worden afgebeeld, afhankelijk van bijvoorbeeld blootstellings- of brandduur of relevante stralingswaarden die tot schade of verwondingen kunnen leiden.

Het model bevat ook een onzekerheidsanalyse waarmee de invloed van onzekerheden in de invoerparameters op het resultaat kan worden berekend. In de voorbeeldberekeningen is rekening gehouden met onzekerheden in het maximale brandvermogen, de stralingsfractie en de vlamhoogte. Overigens geldt de vlamhoogte op basis van de formule van Heskestad voor ongehinderde vlammen. Het rekenmodel is dan ook niet toepasbaar indien vlammen worden gehinderd of afgebogen door obstructies, zoals bijvoorbeeld een (af)dak of plafond.

Literatuurlijst

ARN (2024). Plastics in today's cars will still be used in car parts 20, 30 and even 40 years' time. <https://arn.nl/en/plastics-in-todays-cars-will-still-be-used-in-car-parts-20-30-and-even-40-years-time/>.

CRC (2004). *CRC Handbook of Chemistry and Physics*, 85th Edition, Volume 85.

Euric (2020). EuRIC Call for Recycled Content Plastics in Cars -Position Paper-. https://euric-aisbl.eu/images/Position-papers/EuRIC-Call_for_Recycled_ContentPlastics.pdf

Heskestad, G. (1984). Engineering Relations for Fire Plumes. *Fire Safety Journal*, 7, 25 – 32.

Hu, Y., Zhou, X., Cao, J. et al. (2020) Interpretation of Fire Safety Distances of a Minivan Passenger Car by Burning Behaviors Analysis. *Fire Technol.* 56, 1527–1553. <https://doi.org/10.1007/s10694-019-00938-1>

IFV. (2021). Brandveiligheid parkeergarages met elektrisch aangedreven voertuigen. <https://nipv.nl/wp-content/uploads/2022/03/20210715-IFV-Brandveiligheid-parkeergarages-met-elektrisch-aangedreven-voertuigen-v2021.pdf>

Kang, S., Kwon, M., Yoon Choi, J., & Choi, S. (2023). Full-scale fire testing of battery electric vehicles. *Applied Energy*, 332, 120497. <https://doi.org/10.1016/J.APENERGY.2022.120497>

NIPV (2024). Vos, J., Brans H., en Reinders, J. Literatuuronderzoek naar de brandeffecten van Lithium-ion Batterijbranden. Nog niet gepubliceerd.

SFPE (2016). *Handbook of Fire Protection Engineering*, 6th edition.

Sturm, P., Fößleitner, P., Fruhwirt, D. et al. (2022). Fire tests with lithium-ion battery electric vehicles in road tunnels, *Fire Safety Journal*, Volume 134, 103695, ISSN 0379-7112, <https://doi.org/10.1016/j.firesaf.2022.103695>.

Sun, P., Bisschop, R., Niu, H. et al. A Review of Battery Fires in Electric Vehicles. *Fire Technol* 56, 1361–1410 (2020). <https://doi.org/10.1007/s10694-019-00944-3>

Rijkswaterstaat (2007). Het RWS QRA-model voor wegtunnels - Deel 2: Achtergronddocument, Versie 2.0. <https://www.cob.nl/wp-content/uploads/2018/01/RWSQRA-deel-2.pdf>

RIVM (2024). Stappenplan bepalen brandaandachtsgebied. <https://www.rivm.nl/omgevingsveiligheid/handboek/stappenplannen/bepalen-afstanden-en-gebieden/brandaandachtsgebied>

Xu et al. 2023: Study on the key parameters of vehicle fires for the growth stage in road tunnels. Fire and Materials, 22 July 2023. <https://doi.org/10.1002/fam.3172>.

Yellow Book (2005). Methods for the calculation of Physical Effects Due to releases of hazardous materials (liquids and gases). TNO, 2nd edition, chapter 6.
<https://content.publicatiereeksgevaarlijkestoffen.nl/documents/PGS2/PGS2-1997-v0.1-physical-effects.pdf>

Zhang, D.; Huang, G.; Li, H.; Deng, Q.; Gao, X. (2023). A Study of the Factors Influencing the Thermal Radiation Received by Pedestrians from the Electric Vehicle Fire in Roadside Parking Based on PHRR. Appl. Sci., 13, 609. <https://doi.org/10.3390/app13010609>

Bijlage 1

Leden van de Begeleidingscommissie

RIVM

Jochem Wijten, onderzoeker omgevingsveiligheid

Nationale Agenda Laadinfrastructuur (NAL)

Suzan Reitsma, programmacoördinator elektrisch vervoer, RVO

Wout Benning, beleidsadviseur, RAI-vereniging (Rijwiel en Automobiel Industrie)

Veiligheidsregio's

Bart Slot, adviseur industriële veiligheid, Veiligheidsregio Twente

Jamie Mutsaerts, adviseur industriële veiligheid, Veiligheidsregio Gelderland-Zuid

Vexpan, platform voor parkeren in Nederland

Marcel Laeven, corporate manager innovations, Q-park

Bijlage 2

Broncode rekenprogramma (Python)

```

1  ##Henk Brans (NIPV), Warmtestralingscontouren elektrische voertuigen, 23 april 2024
2  import numpy as np
3  import matplotlib.pyplot as plt
4  from math import pi, atan, acos, sqrt, tan
5  from matplotlib.patches import Rectangle
6  from scipy.stats import norm
7
8  #Constants
9  X_r = 0.45 #Radiative fraction of SEP
10 lifeheight = 1 #Height of plotted contour (meter)
11 alpha = 0.235 #Hesketad Constant
12 aerr = 0.0235 #Hesketad Error
13
14 def calc_viewfactor_middle(r, planelength, planeheight):
15     flb = planeheight - lifeheight #("Enter height of the emitting panel F1, F_1b: "), leefhoogte
16     fla = planelength/2 #("$"Enter length of the emitting panel F1, F_1a: ")
17     f2b = lifeheight #("Enter height of the emitting panel F2, F_2b: ")
18     f2a = planelength/2 #("Enter length of the emitting panel F2, F_2a: ")
19     f3b = lifeheight# ("Enter height of the emitting panel F3, F_3b: ")
20     f3a = planelength/2 #("Enter length of the emitting panel F3, F_3a: ")
21     f4b = planeheight - lifeheight #("Enter height of the emitting panel F4, F_4b: ")
22     f4a = planelength/2# ("Enter length of the emitting panel F4, F_4a: ")
23     c = r #meter, Enter distance of the plane element dA1 from the radiating plane parallel element F: ")
24     X1 = flb/c
25     Y1 = fla/c
26     F1 = (0.5/pi)*(((X1/((1+X1**2)**0.5))*atan(Y1/((1+X1**2)**0.5)))+\
27                 ((Y1/((1+Y1**2)**0.5))*atan(X1/((1+Y1**2)**0.5))))
28     X2 = f2a/c
29     Y2 = f2b/c
30     F2 = (0.5/pi)*(((X2/((1+X2**2)**0.5))*atan(Y2/((1+X2**2)**0.5)))+\
31                 ((Y2/((1+Y2**2)**0.5))*atan(X2/((1+Y2**2)**0.5))))
32     X3 = f3a/c
33     Y3 = f3b/c
34     F3 = (0.5/pi)*(((X3/((1+X3**2)**(0.5)))*atan(Y3/((1+X3**2)**(0.5)))+\
35                 ((Y3/((1+Y3**2)**(0.5)))*atan(X3/((1+Y3**2)**(0.5)))))
36     X4 = f4b/c
37     Y4 = f4a/c
38     F4 = (0.5/pi)*(((X4/((1+X4**2)**0.5))*atan(Y4/((1+X4**2)**0.5)))+\
39                 ((Y4/((1+Y4**2)**0.5))*atan(X4/((1+Y4**2)**0.5))))
40     F = F1+F2+F3+F4
41     return F
42
43 def calc_viewfactor_plane(r, x, planelength, planeheight):
44     flb = planeheight - lifeheight #("Enter height of the emitting panel F1, F_1b: "), leefhoogte
45     fla = x #("$"Enter length of the emitting panel F1, F_1a: ")
46     f2b = lifeheight #("Enter height of the emitting panel F2, F_2b: ")
47     f2a = x #("Enter length of the emitting panel F2, F_2a: ")
48     f3b = lifeheight# ("Enter height of the emitting panel F3, F_3b: ")
49     f3a = planelength - x #("Enter length of the emitting panel F3, F_3a: ")
50     f4b = planeheight - lifeheight #("Enter height of the emitting panel F4, F_4b: ")
51     f4a = planelength - x #("Enter length of the emitting panel F4, F_4a: ")

```

```

51 f4a = planelength - x# ("Enter length of the emitting panel F4, F_4a: ")
52 c = r #meter, Enter distance of the plane element dA1 from the radiating plane parallel element F: ")
53 X1 = f1b/c
54 Y1 = f1a/c
55 F1 = (0.5/pi)*(((X1/((1+X1**2)**0.5))*(atan(Y1/((1+X1**2)**0.5))))+\
56             ((Y1/((1+Y1**2)**0.5))*(atan(X1/((1+Y1**2)**0.5))))))
57 X2 = f2a/c
58 Y2 = f2b/c
59 F2 = (0.5/pi)*(((X2/((1+X2**2)**0.5))*(atan(Y2/((1+X2**2)**0.5))))+\
60             ((Y2/((1+Y2**2)**0.5))*(atan(X2/((1+Y2**2)**0.5))))))
61 X3 = f3a/c
62 Y3 = f3b/c
63 F3 = (0.5/pi)*(((X3/((1+X3**2)**(0.5)))*(atan(Y3/((1+X3**2)**(0.5))))+\
64             ((Y3/((1+Y3**2)**(0.5)))*(atan(X3/((1+Y3**2)**(0.5))))))
65 X4 = f4b/c
66 Y4 = f4a/c
67 F4 = (0.5/pi)*(((X4/((1+X4**2)**0.5))*(atan(Y4/((1+X4**2)**0.5))))+\
68             ((Y4/((1+Y4**2)**0.5))*(atan(X4/((1+Y4**2)**0.5))))))
69 F = F1+F2+F3+F4
70 return F
71
72 def calc_airtransmittance(r):
73     if r < 0.8:
74         tau_air = 1
75     else:
76         P_w = 3169# Pa vapour pressure of water, Wiki 25 degC
77         tau_air = 2.02 *(P_w * r)**(-0.09) #air transmittance
78     return tau_air
79
80 def calc_errorbar(Qdot, Qerr, Xerr, carlength, carwidth, heatflux):
81     r = np.linspace(0.01, 30, num = 500) #meters
82
83
84     Flamebottom = carlength*carwidth
85     aerodynamic_D = sqrt( 4*Flamebottom/pi )
86
87     carheight = -1.02*aerodynamic_D + alpha*Qdot**(2/5)
88
89     Area = 2*carheight*carlength + 2*carheight*carwidth + 2*carwidth*carlength
90     SEP = X_r*Qdot / Area #kw/m2 bronstraling
91
92     viewfactor_list = np.array([calc_viewfactor_middle(i, carlength, carheight) for i in r ])
93     airtransmittance_list = np.array([calc_airtransmittance(i) for i in r])
94
95     multiplied_factors = np.multiply(airtransmittance_list, viewfactor_list)
96     rad_heat_received = np.array([i * SEP for i in multiplied_factors])
97
98     find_nearest = np.asarray(rad_heat_received)
99     idx = (np.abs(find_nearest - heatflux)).argmin()
100
101     meters = r[idx]
102

```

```

102 #Errorcalculation, i = F*tau_air
103 #errorlist = np.array([sqrt( ( ( X_r*i * ( Area - (23/125)*(carwidth+carlength)*Qdot**(2/5) ) ) \
104 # / Area**2 )**2 * Qerr**2 + ( Qdot*i/Area )**2 * Xerr**2 ) for i in multiplied_factors])
105
106
107 errorlist = np.array([sqrt( ( ( X_r*i * ( Area - (23/125)*(carwidth+carlength)*Qdot**(2/5) ) ) \
108 / Area**2 )**2 * Qerr**2 + ( Qdot*i/Area )**2 * Xerr**2 + ( (2*(carwidth+carlength)*X_r*i*Qdot**(7/5)) / Area**2 )**2 * aerr**2 ) \
109 for i in multiplied_factors])
110
111 rad_heat_error = np.subtract(rad_heat_received, errorlist) #De warmteflux minus de onzekerheid.
112
113 find_nearestlow = np.asarray(rad_heat_error) #converting, warmteflux minus onzekerheid
114 idx_low = (np.abs(find_nearestlow - heatflux)).argmin() #Vind de index van het element dat dichtbij de heatflux ligt.
115
116 rad_heat_error_high = np.add(rad_heat_received, errorlist)
117
118 find_nearesthigh = np.asarray(rad_heat_error_high)
119 idx_high = (np.abs(find_nearesthigh - heatflux)).argmin()
120 print('distance [m], low, high to ',heatflux, 'kW/m2 =' , meters, r[idx_low], r[idx_high])
121 return meters, r[idx_low], r[idx_high]
122
123 def calc_heatflux_at_point_longside(r, x, carlength, carheight, SEP):
124 F = calc_viewfactor_plane(r, x, carlength, carheight)
125 tot_rad_emitted = SEP #enter SEP here in kW/m2
126 rad_heat_received = F * tot_rad_emitted * calc_airtransmittance(r)
127 return rad_heat_received
128
129 def calc_heatflux_at_point_shortside(r, x, carwidth, carheight, SEP): #Q is 120 in dit geval
130 F = calc_viewfactor_plane(x, r, carwidth, carheight)
131 tot_rad_emitted = SEP #enter SEP here in kW/m2
132 rad_heat_received = F * tot_rad_emitted * calc_airtransmittance(x)
133 return rad_heat_received
134
135 def calc_heatflux_at_point_vertex(c, d, carlength, carwidth, carheight, SEP, plotsize):
136 #x = afstand vanaf onderkant/beginkant vlak, r = loodrechte afstand tot vlak
137
138 Fbig = calc_viewfactor_plane(c, d+carlength,plotsize+carlength, carheight) # verlengde deelvlak, c= afstand loodrecht
139 Ffictive = calc_viewfactor_plane(c, d, plotsize, carheight)
140 Fvertex = Fbig-Ffictive
141
142 Fbig_short = calc_viewfactor_plane(d, c+carwidth, plotsize+carwidth, carheight)
143 Ffictive_short = calc_viewfactor_plane(d, c, plotsize, carheight)
144 Fvertex_short = Fbig_short - Ffictive_short
145
146 Fvertex = Fvertex + Fvertex_short
147
148 tot_rad_emitted = SEP #enter SEP here in kW/m2
149 rad_heat_received = Fvertex * tot_rad_emitted * calc_airtransmittance(sqrt(c**2+d**2))
150 return rad_heat_received
151
152 def plot_contour():
153 carwidth = 1.8 #meter

```

```

154 carlength = 4.2 #meter
155
156 Flamebottom = carlength*carwidth
157 aerodynamic_D = sqrt( 4*Flamebottom/pi )
158 Qdot = 7250 #kW Enter Peak Heat Release Rate here
159 carheight = -1.02*aerodynamic_D + alpha *Qdot**(2/5)
160
161 print('flameheight [m]=' , carheight)
162 Area = 2*carheight*carlength + 2*carheight*carwidth + 2*carwidth*carlength
163 SEP = X_r * Qdot / Area #kw/m2 bronstraling
164 print('SEP [kw/m2]=' ,SEP)
165 plotsize = 10
166
167 #lange zijde
168 u = np.linspace(carwidth/2+0.0001, plotsize, num = 200)
169 v = np.linspace(0.0001, carlength, num = 200)
170 x, y = np.meshgrid(u,v)
171 Z = np.sqrt(y*0+x) # dummy values
172 Y = x*0 + y
173 X = y*0 + x
174
175 #korte zijde
176 p = np.linspace(carlength/2+0.0001, plotsize, num = 200)
177 q = np.linspace(0.0001, carwidth, num = 200)
178 s, t = np.meshgrid(q,p)
179 W = np.sqrt(s*0+t) # dummy values
180 S = t*0 + s
181 T = s*0 + t
182
183 #hoekpunt
184 a = np.linspace(carwidth/2+0.0001, plotsize, num = 200)
185 b = np.linspace(carlength/2+0.0001, plotsize, num = 200) #y=2 tot 8
186 c, d = np.meshgrid(a,b)
187 G = np.sqrt(c*0+d) # dummy values
188 C = d*0 + c #x-cord
189 D = c*0 + d #y-cord
190
191 fig, ax = plt.subplots()
192 ax.add_patch(Rectangle((-carwidth/2, -carlength/2), carwidth, carlength)) #linkerpunt cord, breedte, hoogte
193
194 for i in range(len(Z)):
195     for j in range(len(Z)):
196         Z[i][j] = calc_heatflux_at_point_longside(X[i][j]-carwidth/2 , Y[i][j],carlength ,carheight , SEP)
197 for i in range(len(W)):
198     for j in range(len(W)):
199         W[i][j] = calc_heatflux_at_point_shortside(S[i][j], T[i][j] -carlength/2, carwidth , carheight, SEP)#gekanteld vlak
200 for i in range(len(G)):
201     for j in range(len(G)):
202         G[i][j] = calc_heatflux_at_point_vertex(C[i][j]-carwidth/2, D[i][j]-carlength/2, carlength, carwidth, carheight, SEP, plotsize)
203
204 plt.contour(x,y -carlength/2, Z, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])

```

```

205 #plt.contour(-x,y-carlength/2,Z, levels = [1, 3, 4.5, 10, 35], colors=['blue', 'lightblue', 'yellow', 'orange', 'darkred'])
206 plt.contour(-x,y-carlength/2,Z, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
207 plt.contour(s-carwidth/2,t, W, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
208 plt.contour(s-carwidth/2,-t, W, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
209
210 plt.contour(a,b, G, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
211 plt.contour(a,-b,G, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
212 plt.contour(-a,-b,G, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
213 plt.contour(-a,b,G, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
214
215 #Errorbar calculation
216 errorbar_values35 = calc_errorbar(7250, 706, 0.05, carlength, carwidth, 35)
217 errormiddle35 = (errorbar_values35[1]+errorbar_values35[2])/2 + carwidth/2
218 plt.errorbar( errormiddle35, 0, xerr= carwidth/2+ errorbar_values35[2]-errormiddle35, fmt = 'o', markersize =1, color = 'darkred',capsize=3)
219
220 errorbar_values10 = calc_errorbar(7250,706, 0.05, carlength, carwidth, 10)
221 errormiddle10 = (errorbar_values10[1]+errorbar_values10[2] +carwidth)/2
222 plt.errorbar( errormiddle10, 0, xerr= carwidth/2+ errorbar_values10[2]-errormiddle10, fmt = 'o', markersize =1, color = 'orange',capsize=3)
223
224 errorbar_values4 = calc_errorbar(7250, 706, 0.05,carlength, carwidth, 4)
225 errormiddle4 = (errorbar_values4[1]+errorbar_values4[2]+carwidth)/2
226 plt.errorbar( errormiddle4, 0, xerr= carwidth/2+ errorbar_values4[2]-errormiddle4, fmt = 'o', markersize =1, color = 'yellow',capsize=3)
227 plt.grid()
228
229 plt.title('Warmtestralingscontour Elektrische SUV ')
230
231 plt.xlabel('x-afstand (m)')
232 plt.ylabel('y-afstand (m)')
233 cbar = plt.colorbar()
234 cbar.set_label('Radiatieve Hitteflux (kW/m\u00b2)')
235 cbar.lines[0].set_linewidth(10)
236
237 plt.show()
238
239 def plot_contourbus():
240     carwidth = 2.55
241     carlength = 12
242
243     Qdot = 29500 #kW Enter Peak Heat Release Rate here
244
245     Flamebottom = carlength*carwidth
246     aerodynamic_D = sqrt( 4*Flamebottom/pi )
247
248     carheight = -1.02*aerodynamic_D + 0.235*Qdot**(2/5)
249
250     print('car/flameheight=',carheight)
251     Area = 2*carheight*carlength + 2*carheight*carwidth + 2*carwidth*carlength
252     SEP = X_r * Qdot / Area #kw/m2 bronstraling
253     print('SEP=',SEP)
254
255     plotsize = 20

```

```

256 #lange zijde
257 u = np.linspace(carwidth/2+0.0001, plotsize, num = 200)
258 v = np.linspace(0.0001, carlength, num = 200)
259 #u = np.append(u,v)
260 x, y = np.meshgrid(u,v)
261 Z = np.sqrt(y*0+x) # dummy values
262 Y = x*0 + y
263 X = y*0 + x
264
265 #korte zijde
266 p = np.linspace(carlength/2+0.0001, plotsize, num = 200)
267 q = np.linspace(0.0001, carwidth, num = 200)
268
269 s, t = np.meshgrid(q,p)
270 W = np.sqrt(s*0+t) # dummy values
271 S = t*0 + s
272 T = s*0 + t
273
274 #hoekpunt
275 a = np.linspace(carwidth/2+0.0001, plotsize, num = 200)
276 b = np.linspace(carlength/2+0.0001, plotsize, num = 200) #y=2 tot 8
277 c, d = np.meshgrid(a,b)
278 G = np.sqrt(c*0+d) # dummy values
279 C = d*0 + c #x-cord
280 D = c*0 + d #y-cord
281
282 fig, ax = plt.subplots()
283 ax.add_patch(Rectangle((-carwidth/2, -carlength/2), carwidth, carlength)) #linkerpunt cord, breedte, hoogte
284
285 for i in range(len(Z)):
286     for j in range(len(Z)):
287         Z[i][j] = calc_heatflux_at_point_longside(X[i][j]-carwidth/2 , Y[i][j],carlength ,carheight , SEP)
288 for i in range(len(W)):
289     for j in range(len(W)):
290         W[i][j] = calc_heatflux_at_point_shortside(S[i][j], T[i][j] -carlength/2, carwidth , carheight, SEP)#gekanteld vlak
291 for i in range(len(G)):
292     for j in range(len(G)):
293         G[i][j] = calc_heatflux_at_point_vertex(C[i][j]-carwidth/2, D[i][j]-carlength/2, carlength, carwidth, carheight, SEP, plotsize)
294
295 plt.contour(x,y -carlength/2, Z, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
296 #plt.contour(-x,y-carlength/2,Z, levels = [1, 3, 4.5, 10, 35], colors=['blue', 'lightblue', 'yellow', 'orange', 'darkred'])
297 plt.contour(-x,y-carlength/2,Z, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
298 plt.contour(s-carwidth/2,t, W, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
299 plt.contour(s-carwidth/2,-t, W, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
300
301 plt.contour(a,b, G, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
302 plt.contour(a,-b,G, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
303 plt.contour(-a,-b,G, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
304 plt.contour(-a,b,G, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
305
306 #Errorbar calculation
307 errorbar values35 = calc_errorbar(29500, 500, 0.05, carlength, carwidth, 35)

```

```

307 errorbar_values35 = calc_errorbar(29500, 500, 0.05, carlength, carwidth, 35)
308 errormiddle35 = (errorbar_values35[1]+errorbar_values35[2]+carwidth)/2
309 plt.errorbar( errormiddle35, 0, xerr= carwidth/2+ errorbar_values35[2]-errormiddle35 , fmt = 'o', markersize =1, color = 'darkred',capsize=3)
310
311 errorbar_values10 = calc_errorbar(29500,500, 0.05, carlength, carwidth, 10)
312 errormiddle10 = (errorbar_values10[1]+errorbar_values10[2] +carwidth)/2
313 plt.errorbar( errormiddle10, 0, xerr= carwidth/2+ errorbar_values10[2]-errormiddle10, fmt = 'o', markersize =1, color = 'orange',capsize=3)
314
315 errorbar_values4 = calc_errorbar(29500, 500, 0.05,carlength, carwidth, 4)
316 errormiddle4 = (errorbar_values4[1]+errorbar_values4[2]+carwidth)/2
317 plt.errorbar( errormiddle4, 0, xerr= carwidth/2+ errorbar_values4[2]-errormiddle4, fmt = 'o', markersize =1, color = 'yellow',capsize=3)
318
319 plt.grid()
320
321 plt.title('Warmtestralingscontour Stadsbus ')
322
323 plt.xlabel('x-afstand (m)')
324 plt.ylabel('y-afstand (m)')
325 cbar = plt.colorbar()
326 cbar.set_label('Radiatieve Hitteflux (kW/m\u00b2)')
327 cbar.lines[0].set_linewidth(10)
328 plt.show()
329
330 def plot_contourtruck():
331     carwidth = 2.55
332     carlength = 18
333
334     Qdot = 71800 #kW Enter Peak Heat Release Rate here
335
336     Flamebottom = carlength*carwidth
337     aerodynamic_D = sqrt( 4*Flamebottom/pi )
338
339     carheight = -1.02*aerodynamic_D + alpha*Qdot**(2/5)
340
341     print('car/flameheight=',carheight)
342     Area = 2*carheight*carlength + 2*carheight*carwidth + 2*carwidth*carlength
343     SEP = X_r * Qdot / Area #kw/m2 bronstraling
344     print('SEP=',SEP)
345     plotsize = 30
346
347     #lange zijde
348     u = np.linspace(carwidth/2+0.0001, plotsize, num = 200)
349     v = np.linspace(0.0001, carlength, num = 200)
350     x, y = np.meshgrid(u,v)
351     Z = np.sqrt(y*0+x) # dummy values
352     Y = x*0 + y
353     X = y*0 + x
354
355     #korte zijde
356     p = np.linspace(carlength/2+0.0001, plotsize, num = 200)
357     q = np.linspace(0.0001, carwidth, num = 200)
358

```



```

359 s, t = np.meshgrid(q,p)
360 W = np.sqrt(s*0+t) # dummy values
361 S = t*0 + s
362 T = s*0 + t
363
364 #hoekpunt
365 a = np.linspace(carwidth/2+0.0001, plotsize, num = 200)
366 b = np.linspace(carlength/2+0.0001, plotsize, num = 200) #y=2 tot 8
367 c, d = np.meshgrid(a,b)
368
369 G = np.sqrt(c*0+d) # dummy values
370 C = d*0 +c #x-cord
371 D = c*0 + d #y-cord
372
373 fig, ax = plt.subplots()
374 ax.add_patch(Rectangle((-carwidth/2, -carlength/2), carwidth, carlength)) #linkerpunt cord, breedte, hoogte
375 for i in range(len(Z)):
376     for j in range(len(Z)):
377         Z[i][j] = calc_heatflux_at_point_longside(X[i][j]-carwidth/2 , Y[i][j],carlength ,carheight , SEP)
378 for i in range(len(W)):
379     for j in range(len(W)):
380         W[i][j] = calc_heatflux_at_point_shortside(S[i][j], T[i][j] -carlength/2, carwidth , carheight, SEP)#gekanteld vlak
381 for i in range(len(G)):
382     for j in range(len(G)):
383         G[i][j] = calc_heatflux_at_point_vertex(C[i][j]-carwidth/2, D[i][j]-carlength/2, carlength, carwidth, carheight, SEP, plotsize)
384
385 plt.contour(x,y -carlength/2, Z, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
386 #plt.contour(-x,y-carlength/2,Z, levels = [1, 3, 4.5, 10, 35], colors=['blue', 'lightblue', 'yellow', 'orange', 'darkred'])
387 plt.contour(-x,y-carlength/2,Z, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
388 plt.contour(s-carwidth/2,t, W, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
389 plt.contour(s-carwidth/2,-t, W, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
390
391 plt.contour(a,b, G, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
392 plt.contour(a,-b,G, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
393 plt.contour(-a,-b,G, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
394 plt.contour(-a,b,G, levels = [4, 10, 35], colors=['yellow', 'orange', 'darkred'])
395
396 #Errorbar calculation
397 errorbar_values35 = calc_errorbar(71800, 14360, 0.05, carlength, carwidth, 35)
398 errormiddle35 = (errorbar_values35[1]+errorbar_values35[2]+carwidth)/2
399 plt.errorbar( errormiddle35, 0, xerr=carwidth/2+ errorbar_values35[2]-errormiddle35, fmt = 'o', markersize =1, color = 'darkred',capsize=3)
400
401 errorbar_values10 = calc_errorbar(71800, 14360, 0.05, carlength, carwidth, 10)
402 errormiddle10 = (errorbar_values10[1]+errorbar_values10[2] +carwidth)/2
403 plt.errorbar( errormiddle10, 0, xerr= carwidth/2+ errorbar_values10[2]-errormiddle10, fmt = 'o', markersize =1, color = 'orange',capsize=3)
404
405 errorbar_values4 = calc_errorbar(71800, 14360, 0.05,carlength, carwidth, 4)
406 errormiddle4 = (errorbar_values4[1]+errorbar_values4[2]+carwidth)/2
407 plt.errorbar( errormiddle4, 0, xerr= carwidth/2+ errorbar_values4[2]-errormiddle4, fmt = 'o', markersize =1, color = 'yellow',capsize=3)
408 plt.grid()
409 plt.title('Warmtestralingscontour \n Vrachtwagen met lading ')

```

```

410 plt.xlabel('x-afstand (m)')
411 plt.ylabel('y-afstand (m)')
412 cbar = plt.colorbar()
413 cbar.set_label('Radiatieve Hitteflux (kW/m\u00b2)')
414
415 cbar.lines[0].set_linewidth(10)
416 plt.show()
417
418 def plot_heatflux():
419     carwidth = 1.8 #meter
420     carlength = 4.2 #meter
421     r = np.linspace(carwidth/2+0.001, 35, num = 200) #meters
422     Flamebottom = carlength*carwidth
423     aerodynamic_D = sqrt( 4*Flamebottom/pi )
424     Qdot = 7250 #kW Enter Peak Heat Release Rate here
425     Qerr = 706
426     Xerr = 0.05
427     aerr = 0.0235
428
429     carheight = -1.02*aerodynamic_D + alpha*Qdot**(2/5)
430
431     print('flameheight [m]=' , carheight)
432     Area = 2*carheight*carlength + 2*carheight*carwidth + 2*carwidth*carlength
433     SEP = X_r * Qdot / Area #kw/m2 bronstraling
434
435     viewfactor_list = np.array([calc_viewfactor_middle(i-carwidth/2, carlength, carheight) for i in r ])
436     airtransmittance_list = np.array([calc_airtransmittance(i-carwidth/2) for i in r])
437     multiplied_factors = np.multiply(airtransmittance_list, viewfactor_list)
438
439     rad_heat_received = [i * SEP for i in multiplied_factors]
440
441     errorlist = np.array([sqrt( ( ( X_r*i * ( Area - (23/125)*(carwidth+carlength)*Qdot**(2/5) ) ) \
442         / Area**2 )**2 * Qerr**2 + ( Qdot*i/Area )**2 * Xerr**2 + ( (2*(carwidth+carlength)*X_r*i*Qdot**(7/5)) / Area**2 )**2 * aerr**2 ) \
443         for i in multiplied_factors])
444
445     rad_heat_low = np.subtract(rad_heat_received, errorlist) #De warmteflux minus de onzekerheid.
446
447     rad_heat_high = np.add(rad_heat_received, errorlist)
448
449     plt.xlabel('Afstand (m)')
450     plt.ylabel('Radiatieve Warmtesflux (kW/m\u00b2)')
451     plt.title('Maximale Warmtestraling Elektrische SUV')
452     plt.plot(r, rad_heat_received, label = 'Warmtestraling')
453     plt.plot(r, rad_heat_high, color = 'lightgreen', linestyle = 'dashed', label = 'Onzekerheidsmarge')
454     plt.plot(r, rad_heat_low, color = 'lightgreen',linestyle = 'dashed')
455     plt.grid()
456
457     plt.axvline(x = carwidth/2, color = 'black', label = 'Zijkant voertuig')
458
459     leg = plt.legend(loc='upper right')
460
461     plt.vlim(0,55)

```

```

461 plt.xlim(0,12)
462 #x = [0,1,2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
463 #plt.xticks(x)
464 plt.show()
465
466 def plot_heatflux_bus():
467     carwidth = 2.55 #meter
468     carlength = 12 #meter
469     r = np.linspace(carwidth/2+0.001, 35, num = 200) #meters
470     Flamebottom = carlength*carwidth
471     aerodynamic_D = sqrt( 4*Flamebottom/pi )
472     Qdot = 29500 #kW Enter Peak Heat Release Rate here
473     Qerr = 500
474     Xerr = 0.05
475
476     carheight = -1.02*aerodynamic_D + alpha*Qdot**(2/5)
477
478     print('flameheight [m]=' , carheight)
479     Area = 2*carheight*carlength + 2*carheight*carwidth + 2*carwidth*carlength
480     SEP = X_r * Qdot / Area #kw/m2 bronstraling
481
482     viewfactor_list = np.array([calc_viewfactor_middle(i-carwidth/2, carlength, carheight) for i in r ])
483     airtransmittance_list = np.array([calc_airtransmittance(i-carwidth/2) for i in r])
484     multiplied_factors = np.multiply(airtransmittance_list, viewfactor_list)
485
486     rad_heat_received = [i * SEP for i in multiplied_factors]
487
488     errorlist = np.array([sqrt( ( ( X_r*i * ( Area - (23/125)*(carwidth+carlength)*Qdot**(2/5) ) ) \
489         / Area**2 )**2 * Qerr**2 + ( Qdot*i/Area )**2 * Xerr**2 + ( (2*(carwidth+carlength)*X_r*i*Qdot**(7/5)) / Area**2 )**2 * aerr**2 ) \
490         for i in multiplied_factors])
491
492     rad_heat_low = np.subtract(rad_heat_received, errorlist) #De warmteflux minus de onzekerheid.
493
494     rad_heat_high = np.add(rad_heat_received, errorlist)
495
496     plt.xlabel('Afstand (m)')
497     plt.ylabel('Radiatieve Warmtesflux (kW/m\u00b2)')
498     plt.title('Maximale Warmtestraling Stadsbus')
499     plt.plot(r, rad_heat_received, label = 'Warmtestraling')
500     plt.plot(r, rad_heat_high, color = 'lightgreen', linestyle = 'dashed', label = 'Onzekerheidsmarge')
501     plt.plot(r, rad_heat_low, color = 'lightgreen',linestyle = 'dashed')
502     plt.grid()
503
504     plt.axvline(x = carwidth/2, color = 'black', label = 'Zijkant voertuig')
505
506     leg = plt.legend(loc='upper right')
507     plt.ylim(0,55)
508     plt.xlim(0,20)
509     plt.show()
510
511 def plot_heatflux_truck():

```

```

512
513 carwidth = 2.55 #meter
514 carlength = 18 #meter
515 r = np.linspace(carwidth/2+0.001, 35, num = 200) #meters
516 Flamebottom = carlength*carwidth
517 aerodynamic_D = sqrt( 4*Flamebottom/pi )
518 Qdot = 71800 #kW Enter Peak Heat Release Rate here
519 Qerr = 14360
520 Xerr = 0.05
521
522 carheight = -1.02*aerodynamic_D + alpha*Qdot**(2/5)
523
524 print('flameheight [m]=' , carheight)
525 Area = 2*carheight*carlength + 2*carheight*carwidth + 2*carwidth*carlength
526 SEP = X_r * Qdot / Area #kw/m2 bronstraling
527
528 viewfactor_list = np.array([calc_viewfactor_middle(i-carwidth/2, carlength, carheight) for i in r ])
529 airtransmittance_list = np.array([calc_airtransmittance(i-carwidth/2) for i in r])
530 multiplied_factors = np.multiply(airtransmittance_list, viewfactor_list)
531
532 rad_heat_received = [i * SEP for i in multiplied_factors]
533
534 errorlist = np.array([sqrt( ( ( X_r*i * ( Area - (23/125)*(carwidth+carlength)*Qdot**(2/5) ) ) \
535     / Area**2 )**2 * Qerr**2 + ( Qdot*i/Area )**2 * Xerr**2 + ( 2*(carwidth+carlength)*X_r*i*Qdot**(7/5) / Area**2 )**2 * aerr**2 ) \
536     for i in multiplied_factors])
537
538 rad_heat_low = np.subtract(rad_heat_received, errorlist) #De warmteflux minus de onzekerheid.
539
540 rad_heat_high = np.add(rad_heat_received, errorlist)
541
542 plt.xlabel('Afstand (m)')
543 plt.ylabel('Radiatieve Warmtesflux (kW/m\u00b2)')
544 plt.title('Maximale Warmtestraling Vrachtwagen met lading')
545 plt.plot(r, rad_heat_received, label = 'Warmtestraling')
546 plt.plot(r, rad_heat_high, color = 'lightgreen', linestyle = 'dashed', label = 'Onzekerheidsmarge')
547 plt.plot(r, rad_heat_low, color = 'lightgreen',linestyle = 'dashed')
548 plt.grid()
549
550 plt.axvline(x = carwidth/2, color = 'black', label = 'Zijkant voertuig')
551
552
553 leg = plt.legend(loc='upper right')
554 plt.ylim(0,65)
555 plt.xlim(0,30)
556 plt.show()

```