**MODELLEREN EN SIMULEREN HOOFDSTUK 3: Simuleren van (bio)systemen**

1. Modelsimulatie

1.1 inleiding

* Vorige H’s: hoe model voor een systeem kan worden opgesteld
* In dit H: het simuleren met behulp van een model
* Modeleren = model bouwen
* Simuleren
  + = model gebruiken om scenario’s door te rekenen
  + = ‘virtueel’ experimenteren
  + = het oplossen vh model naar zijn outputs en/of toestanden (soms output =toestand)
    - Modellen gebruiken om oplossingen te krijgen vd vglen
  + Vb: geïnteresseerd in wat er gebeurt wnnr de beginsituatie wijzigt of wnnr waarden voor bep. parameters veranderen
    - => met gevalideerd model kan dit nagegaan worden zonder experimenten

1.2 Analytische oplossingen

* Analytische oplossing
  + Bestaat als het model niet te complex is
  + Analytische oplossing vinden = het oplosbaar zijn van de (set van) DV

1.3 Numerieke oplossingen

* Numerieke oplossing
  + Als er geen analytische oplossing is
    - => numerieke technieken gebruiken om set van DV op te lossen
  + Biosysteemmodellen bestaan uit set van **gekoppelde niet lineaire DV** (complex)
    - => geen analytische oplossing voor
  + Vb numerieke methodes: Euler methode,….

2. Parameterschatting

2.1 Inleiding

* Parameterschatting
  + Parameters in model vaak onbekend => waarden parameters schatten
  + = het bepalen vd optimale waarden voor de parameters door het model te vglen met experimentele data
    - Veronderstelling: structuur vh model & dus relaties tssn parameters en variabelen zijn gekend => model moet kloppen!
  + = modelkalibratie of het fitten van een model tov experimentele data
  + Gebruikt in statistische context:
    - Reden: schatting is niet 100% correct => er is een statistische afwijking
    - Geschatte parameter = een gem. waarde met een statistische variabiliteit
      * Variabiliteit gekwantificeerd door de standaardafwijking SD of standaardfout SE of door betrouwbaarheidsinterval
    - Bias = de afwijking vd parameter tov verwachte werkelijke waarde

2.2 De doelfunctie

* Doelfunctie
  + => gebruikt om de parameterschatting uit te voeren
  + = kwantificatie vd afwijking van (of het verschil tussen) de modelpredictie & de experimentele data voor een bepaalde parameterset  **(vector!)**
  + = waarden afhankelijk van parameterset
  + = J**)** metde parameterset
    - De waarde vd doelfunctie w bepaald door de set parameters vh model
  + Minimalisatie vd doelfunctie
    - Door aanpassen vd waarden van 1 of meer modelparameters
    - Minimalisatie gebeurt ahdv een optimalisatiealgoritme
* Som van de kwadratische afwijkingen (residuals) **SSE** (sum of squared errors)
  + = meest gebruikte doelfunctie voor parameterschatting
  + => gaat de experimentele data vergelijken met de modelpredictie waarden, om zo de meest correcte ligging van de rechte lijn te bekomen (zie fig)
  + Doelfunctie in geval van **1 variabele** waarvoor **n observaties zijn**

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

* + - yexp, i = de (n) experimentele observaties
    - y^i ( = de modelpredicties voor gegeven parameterset
    - figuur ppt 154
      * lijn = modelpredictie = wat model voorspelt adhv parameters
      * punten = experimentele observaties
      * => goed model want punten liggen ong op de lijn
  + Doelfunctie voor **meer dan 1 variabele** 
    - Vb: substraatconcentratie en hoeveelheid biomassa in bioreactor
    - Als de fouten op de versch. variabelen van dezelfde grootte orde zijn dan is de doelfunctie:

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

* + - Als de fouten op de versch. variabelen van versch. grootte orde zijn dan is de doelfunctie de **gewogen som van de kwadratische afwijkingen WSSE**
      * Maw niet alle variabelen zijn even betrouwbaar

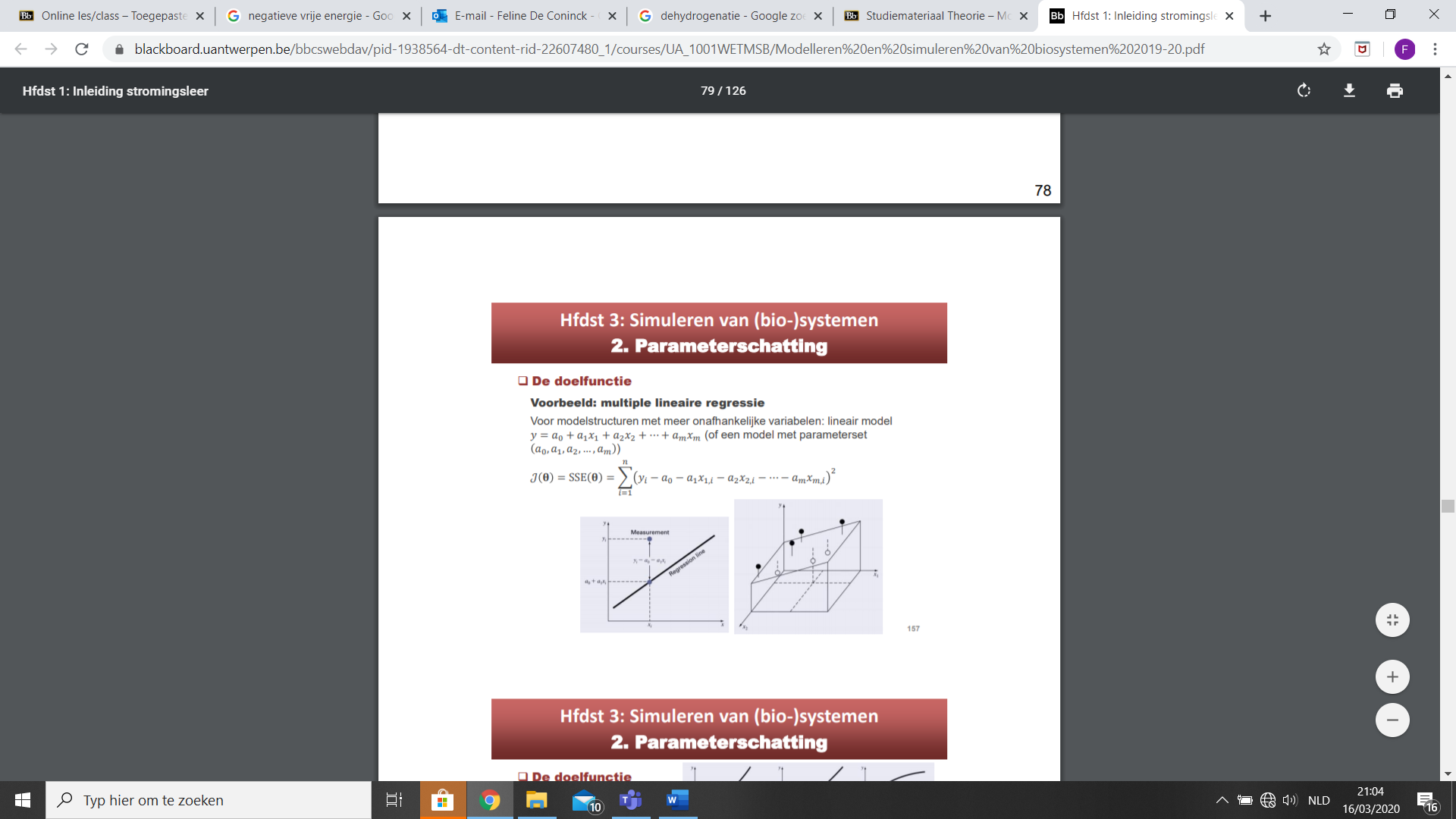
Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

* + - * Waarbij i = de standaardafwijking vd dataset yexp,1,i (n waarden)
      * Grotere standaardafwijking i zal kleinere bijdrage hebben
* **Voorbeeld:** lineaire regressie
  + SSE ook gebruiken voor lineaire regressie
  + Lineaire regressie
    - = voorbeeld van parameterschatting
    - = het fitten van een eenvoudig lineair model y= a0 + a1x tov de experimentele data, maw een model met parameterset (a0,a1)
      * Output = y= a0 + a1x
    - LR w gebruikt om na te gaan of de respons y lineair afh. is vd waarde x
    - De doelfunctie:

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

* + - * Best fittende rechte verkrijgen (de regressierechte) door de doelfunctie te minimaliseren mbv een minimalisatiealgoritme
        + Conclusie: LR = vb voor SSE gebruik als doelfunctie waarbij SSE geminimaliseerd w
* **Voorbeeld**: Multiple lineaire regressie
  + Multiple lineaire regressie
    - = Uitbreiding voor modelstructuren met meer onafhankelijke variabelen
    - = het fitten van een lineair model y=a0 + a1x1 + a2x2 + …. + amxm tov de experimentele data, maw een model met parameterset (a0,a1,a2,…am))
      * Output = lineaire combo van meer variabelen: y=a0 + a1x1 +.. +amxm
    - De doelfunctie voor m variabelen:
      * Doelfunctie wordt geminimaliseerd mbv minimalisatiealgoritme
    - Ppt 157 fig links: 1 variabele
    - Ppt 157 fig rechts: voorbeeld voor 2 variabelen
      * Weergave de beste fit en residuals bij multiple lineaire regressie
      * => zoeken vlak dat beste fit
* **Voorbeeld:** linearisatie van niet-lineaire modellen
  + Linearisatie van niet-lineaire modellen
    - Parameterschatting voor niet lineair modellen
    - 2 manieren fitten
      * 1) Niet lineair model kan door transformatie gelineariseerd worden => dan SSE toepassen
      * 2) Rechtstreeks via niet lineair model: kwadratenverschil tssn experimentele data & waarden op NL grafiek
    - Ppt p158 voorbeelden
      * 1) Exp. functie, … om data te fitten
      * 2) Linearisatie exp. functie, machtsfunctie & hyperbolische functie
    - Opmerking: toepassen van lineaire transformatie => 2 problemen:
      * 1) niet alle vglen kunnen getransformeerd worden
      * 2) beste fit voor het lineaire model is niet noodzakelijk de beste fit voor het originele, niet lineaire model
        + => parameters bekomen door beide methoden ku verschillen
        + => Voorbeeld ppt159

Experimentele data waarbij 2 modellen w gebruikt voor het schatten van parameters

Curven = model waarbij de parameters zijn geoptimaliseerd via optimalisatiealgoritme

Volle lijn = originele machtsfunctie

Stippel lijn = gelineariseerde functie

=> hier geeft originele functie beste overeenkomst

* **Voorbeeld:** niet lineaire regressie
  + Niet lineaire regressie
    - = fitting door een hogere orde polynoom (hogere graadsvergelijking)
    - = polynomiale regressie
      * Versch. mogelijkheden om niet-lineaire fit uit te voeren door gebruik v/e hogere orde polynoom
    - Doelfunctie voor m-de orde polynoom:

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

* + - * De som vd kwadratische fouten/ doelfunctie wordt geminimaliseerd
    - Ppt p160 fig links: 1ste orde polynoom/ rechte = slecht
    - Ppt p160 fig rechts: 2de orde polynoom = goed

2.3 . Polynomiale modellen

* Polynomiale modellen
  + Gebruikt om experimentele data te fitten
    - Model heeft hierbij geen fysische betekenis => black box model
    - Gevolg: Onbetrouwbaar bij extrapolatie
  + Vb: 3de graads vgl, 4de graadsvgl,..
  + **Voorbeeld:** stel experimentele data van microbiële groeisnelheid µin functie vd substraatconcentratie S is beschikbaar
    - 3 modelstructuren w gefit tov de experimentele data:
      * Monod model, Tessier model, Polynomiaal blackbox model
    - Monod & Tessier: dezelfde modelcomplexiteit
      * Modellen worden beschreven door 2 parameters en
      * Modellen hebben fysische interpretatie
    - Polynomiaal blackbox model
      * Model beschreven door 6 parameters
      * Model heeft geen fysische intepretatie
    - Ppt p161 foto
      * = experimentele data van µ in functie van S met het gesimuleerd Monod, Teissier en polynoom model
      * => fitten van de 3 modellen met de experimentele data levert modelparameters
      * => elk van de 3 modellen geeft goede beschrijving vd data weer
    - Ppt p162 foto rechts
      * = modellen geëxtrapoleerd voor een meer uitgebreide substraatrange / bereik
      * => **bereik verhogen S > 500**
      * => polynomiaal model (black box) geeft geen correcte weergave data buiten bij extrapolatie naar hoger bereik
        + buiten bereik waarin ze gefit is (500) slechte weergave
      * => Monod & Tessier geven goede voorspelling ook buiten range waarin ze gefit zijn (500)
  + Problemen polynomiaal model
    - 1) Extrapolatie kan probleem geven (bij grotere range/bereik)
    - 2) Accuraatheid vd geschatte coeff of parameters bij hogere orde polynomen
      * Polynomiale modellen gevoelig aan accuraatheid parameters
        + Accuraatheid polyn. modellen ~ accuraatheid parameters
      * Ppt p163: voorbeeld van 6de orde polynoom
        + Bovenste fig: geschatte parameters met accuraatheid 16dec

= zeer nauwkeurige parameters

* + - * + Onderste fig: geschatte parameters met accuraatheid 8dec
      * Oplossing: hogere orde polynomen vermijden
        + Door bereik te beperken: opsplitsen van hogere orde polynomen in lagere orde polynomen
        + Ppt p163 rechts: 6de orde polynoom => 2 2de orde polynomen

2.4. Identificeerbaarheid van een model

* Identificeerbaarheid van een model
  + = voorwaarde om aan parameterschatting te kunnen voldoen
* Identificeerbaar model
  + = model (ook modelstructuur en modelparameters) is identificeerbaar voor een bepaalde experimentele dataset indien aan alle parameters een unieke waarde kan w toegekend ⬄ niet-identificeerbaar model

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

* + Gevolg niet-identificeerbaar model: het gelijktijdig schatten van alle parameters resulteert dan in lage kwaliteit vd schatters & dus groot betrouwbaarheidsinterval
  + Theoretische of structurele identificeerbaarheid
    - = gekoppeld aan de modelstructuur: bij perfecte data ku problemen met identificeerbaarheid enkel gelegen zijn aan de modelstructuur
  + Praktische identificeerbaarheid
    - = gekoppeld aan de kwaliteit vd data: een model kan theoretisch identificeerbaar zijn maar toch niet praktisch identificeerbaar (kwaliteit vd data is ontoereikend) door vb ruis op data => data bevat weinig info
* Voorbeeld 1
  + Afbeelding met schermafbeelding

    Automatisch gegenereerde beschrijvingModel 1: y=ax1 +bx2
  + Toestandsvariabelen x1 en x2
  + Parameters a en b zijn beiden theoretisch identificeerbaar: indien output y partieel afgeleid w naar beide toestandsvariabelen, ku ze onafh van elkaar bepaald w
  + MAAR
    - Indien uit experimentele data blijkt dat x1 en x2 proportioneel zijn (x1=αx2) => dan ku de parameters niet uniek geschat w
      * Dan bestaan versch parametercombinaties (a,b) die bij de geg. Toestand dezelfde output geven
      * Dan is de beschikbare experimentele data niet voldoende informaties om aan de modelparameters unieke waarden te geven
      * Conclusie: parameters zijn wel theoretisch ident., maar niet praktisch
    - Indien nieuwe data verzameld w => inzien dat x1 en x2 toch niet proportioneel zijn => dan ook praktische identificeerbaarheid
* Voorbeeld 2

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

2.5 Voorbereidende stappen voor parameterschatting

* 1) Geschikte dataset selecteren
  + Een beschikbare dataset wordt opgedeeld in:
    - Dataset voor parameterschatting (kalibratie)
    - Dataset voor validatie na parameterschatting
  + Beide datasets moeten voldoende info bevatten => dus voldoende breed spectrum aan experimentele condities hebben
  + Beide datasets moeten zo onafhankelijk mogelijk zijn => voor betrouwbare validatie
* 2) Parameters selecteren
  + Beslissen welke parameters wel & welke niet geschat moeten worden
  + Belangrijk: ook begin- en randvwden van toestandsvariabelen & inputs ku als parameters beschouwd (en geschat) worden
* 3) Initiele schatting
  + Altijd nodig indien een iteratieve methode voor parameterschatting gebruikt w
  + Door een goede initiele schatting van parameters => kan sneller het optimalisatiecriterium bereikt w (sneller convergeren)
* 4) Opgeven van grenzen voor parameterwaarden
  + Sommige parameters ku slechts waarden binnen een bep. interval aannemen => grenzen opleggen
  + Vb: massa of concentratie = steeds positief

2.6 Minimaliseren vd doelfunctie

* Minimaliseren vd doelfunctie gaat hij niet in de theorie vragen, dus lees de ppt gwn

2.7 Minimalisatiealgoritmen

2.7.1 Inleiding

* Verschillen tssn de minimalisatiealgoritmen kennen (concept kennen)
  + Maar vergelijkingen/formules nooit vragen bij theorie
* Minimalisatiealgoritmen
  + => minimaliseren vd doelfunctie
  + Vertrekken van een initiele schatting vanwaar iteratief gezocht w naar de parameterwaarden van die de doelfunctie minimaliseert
    - Iteratief = stapsgewijs
  + Op basis vd initiele schatting => modelvoorspellingen berekenen & vervolgens de doelfunctie berekenen door voorspellingen te confronteren met de data
  + Op basis van regels die verschillend zijn voor elk minimalisatiealgoritme w vervolgens
    - 1) een nieuwe parameterset voorgesteld waarmee opnieuw de modelvoorspellingen & de doelfunctie berekend kan w
    - 2) of indien aan bep criteria voldaan is => de parameterwaarden gebruiken als beste schattingen
  + Criteria kunnen zijn
    - 1) maximaal aantal iteraties bereikt
    - 2) geen verdere vermindering vd doelfunctie bij aanpassen of verfijnen vd parameters
* Probleem bij niet-lineaire minimalisatie: het bestaan van lokale minima in de doelfunctie
  + Lokaal minimum = laagste waarde voor doelfie in een bep regio in parameterruimte
  + ⬄ globaal minimum = laagste waarde voor de doelfie in volledig parameterdomein
  + Fig onder ppt 176: topografie vd doelfunctie voor een niet-lineair optimalisatieprobleem met 2 te schatten parameters
    - => voorkomen van lokale minima en globaal minimum in parameterruimte
  + Fig rechts ppt 176: parameterruimte met 1 parameter (doorsnede)
    - => onderscheid tssn lokaal minimum & globaal minimum
  + Oplossing: starten vanuit verschillende initiele schattingen
    - => indien ze hetzelfde minimum aangeven => dit is het globaal min.

2.7.1 Minimalisatiealgoritmen op basis van de gradient

* 1) Methode van de steilste helling
  + Elke stap vd zoektocht gebeurt in de richting vd steilste helling (grootste gradient) vd doelfunctie en dit zolang de doelfunctie afneemt
  + Als minimum in bep. richting bereikt is => wordt opnieuw de richting vd steilste helling gezocht => zoektocht gaat verder in deze richting => tot globaal min bereikt is
  + Procedure: voor elke iteratie k
    - 1) Bepaal de richting sk vd steilste helling = steilste neg. gradient
      * = de Jacoberiaan = **vector** met daarin de 1ste orde partiele afgeleiden vd doelfunctie naar de parameters
    - 2) Minimaliseren in deze richting waarbij nieuwe parameterset wordt gevonden: met de stapgrootte
      * De optimale stapgrootte w gevonden door de doelfie in die richting te bepalen & hiervan min. bepalen
  + = **lineaire benadering**: Maakt gebruik vd 1ste orde partiele afgeleiden om zoekrichting te bepalen
  + OPM: In praktijk kan deze methode zeer lang duren wnnr het pad vd steilste helling zigzagt naar de bodem v/e smalle vallei => groot aantal iteraties nodig
    - => methode zelden gebruikt
* 2) Methode van Newton
  + Bij de zoektocht wordt niet enkel de Jacoberiaan beschouwd, maar w ook rekening gehouden met de 2de orde afgeleiden dmv de Hessiaan H
    - Hessiaan = **matrix** met alle 2de orde partiele afgeleiden vd doelfie naar parameters ⬄ Jacoberiaan = vector
  + = **kwadratische benadering**
  + Houdt rekening met de **vorm** vd doelfunctie in bep. richting
    - Fig: ppt p183: verschil tssn lineaire benadering volgens de methode vd steilste helling & de kwadratische methode van Newton
  + Voordeel: gaat **gerichter het minimum** zoeken => vraagt minder iteraties => sneller dan de steile helling methode **normaal** gezien
  + Nadelen:
    - Gevoeliger voor lokale minima
    - Gebruikt kleinere stappen waardoor de convergentie vaak **toch** langer duurt
* 3) De Levenberg-Marquardt methode
  + = en hybride vd Newton en de steile helling methode (combinatie)
    - Ze interpoleert tssn beide methodes => zo de snelheid vd Newton methode t combineren met betere kans op convergentie
  + Ver vh minimum => steile helling methode gebruiken
  + Dichter bij het minimum (kleinere gradienten) => overgaan naar Newton methode
  + Voordeel: veel robuuster dan Newton & steile helling methode => w veel gebruikt

2.7.2 Minimalisatiealgoritmen niet op basis vd gradient

* 1) Powell en Brent methode
  + = de richtingverzamelingmethodes
  + = gebaseerd op herhaalde combinatie van 1D zoektochten langs een set van versch richtingen
  + Voordeel: in minder iteraties het minimum bereiken
  + Powell: maakt gebruik van een vaste set richtingen vb: afwisselend langs de eenheidsvectoren in de parameterruimte
    - Zie fig ppt p185: beginnen bij initiele schatting => 2 vaste richtingen gebruiken als 2 parameters => afwisselend langs eenheidsvector => 1 parameter constant veronderstellen & andere varieren => stopt als afname doelfunctie stopt
    - Aantal richtingen ~ aantal parameters
  + Brent: de richtingen worden na elke iteratie aangepast volgens aantal regels

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

* 2) Het simplex algoritme
  + = gebaseerd op het geometrische concept **simplex**: een geometrie bestaande uit p+1 punten in p dimensies en al hun verbindende lijnen en vlakken
    - P=2: driehoek
    - P=3: tetrahedron
  + De zoektocht w gestart van een initiele simplex => dan minimum gezocht door de doelfie te evalueren op elk vd hoekpunten & het hoekpunt met de hoogste waarde vervangen door een ander punt
  + Voorbeeld: parameterruimte met 2 dimensies
    - 1) Bepaal een intiele simplex (1,2,3)
    - 2) 1ste iteratie:
      * Doelfunctie evalueren voor elk hoekpunt
      * Hoekpunt met hoogste waarde vd doelfunctie (hier punt 3) w gereflecteerd (3->4) volgens de as 1-2
      * Resultaat: nieuwe simplex (1,2,4)
    - 3) 2de iteratie:
      * Opnieuw hoogste waarde vd doelfunctie bepalen (punt 1)
      * Reflecteren punt (1->5) volgens de 2-4 as
      * Resultaat: nieuwe simplex (2,4,5)
    - 4) Iteratie n:
      * Wnnr op het einde vh iteratieproces eenzelfde hoekpunt gedurende een aantal iteraties deel uitmaakt vd simplex => wordt de simplex gecontraheerd => waardoor het zoektochtdomein verfijnd wordt
  + Buiten reflecties & contracties kunnen ook expansies w gebruikt om het proces te versnellen, of krimpen om het zoekdomein te verkleinen
    - Reflectie, contractie, expansie, krimpen