

MEMO BvtH/M09.039
Datum 29-05-2009
Auteur(s) Bas van 't Hof
Onderwerp Gebruik van de DuD-methode voor calibratie van
 riviermodellen

Documentinformatie

Versie	Auteur	Datum	Opmerkingen	Review
0.1	BvtH	29-05-2009	Concept	
Bestandslocatie:		/v3/E05q_bo_simona/c89243-costa-calibriv/report		

Inhoudsopgave

Samenvatting	1
1 Inleiding	1
2 Wiskundige representatie van een afregelproces	2
3 Kwadratische benadering van de boetefunctie	3
3.1 Taylor reeks ontwikkeling	3
3.2 Linearisatie van prediction	4
4 De DuD-methode	4
5 Modellen met een beperkte afhankelijkheid	5
6 Een DuD-methode voor modellen met een beperkte afhankelijkheid	6
6.1 Berekening van een stukje afgeleide	6
6.2 Hergebruik van afhankelijkheidspatronen	7
6.3 Betrouwbaarheid van de benadering	7
6.4 Opgeven van de afhankelijkheidsstructuur	7

1 Inleiding

Binnen Rijkswaterstaat wordt het programma Calibriv gebruikt om riviermodellen af te regelen. Dat is een beetje een ongelukkige situatie, omdat een algemene taak (afregelen van

modellen) wordt uitgevoerd met een gereedschap (Calibriv) dat alleen geschikt is voor een specifiek geval (namelijk alleen voor rivieren).

Er wordt al enige tijd gewerkt aan het maken van algemene gereedschappen bijvoorbeeld voor data-assimilatie. De pakketten Costa en OpenDa, die sterk verwant zijn, bieden een grote hoeveelheid data-assimilatiemethoden aan, waarbij het niet meer uitmaakt wat voor model wordt afgeregeld.

Het is gewenst om het programma Calibriv te vervangen door een koppeling met een algemeen gereedschap. De 'DuD-methode' ('does not use derivatives') van OpenDa lijkt vrij sterk op de methode die wordt gebruikt in Calibriv. Er zijn echter enkele uitbreidingen nodig aan deze DuD-methode voordat deze voor het afregelen van riviermodellen even efficiënt is als het programma Calibriv.

In dit memo wordt onderzocht welke mogelijkheden er zijn om de DuD-methode uit te breiden zodat riviermodellen even efficiënt kunnen worden afgeregeld als Calibriv dat kan. Daarbij wordt er op gelet dat de nieuwe methode een bepaalde mate van algemene toepasbaarheid heeft. Verder moet de nieuwe methode zodanig krachtig zijn voor de uitbreidingen van de afregeling van riviermodellen die binnenkort verwacht worden.

In Sectie 4 wordt de huidige DuD-methode beschreven. In Sectie 5 wordt uitgelegd welke speciale eigenschappen van de riviermodellen een uitbreiding van de DuD-methode noodzakelijk maken. In Sectie 6 wordt een dergelijke uitbreiding van de DuD-methode gepresenteerd.

2 Wiskundige representatie van een afregelproces

Bij het afregelen van een model worden geschikte waarden gekozen een aantal parameters $\mathbf{par} \in \mathbb{R}^{n_{\mathbf{par}}}$ die in het model een rol spelen. Het gaat hierbij bijvoorbeeld om de bodemruwheid in bepaalde gedeelten van een rivier.

Voor een gekozen waarde van de parameters kan door een simulatie een aantal resultaten $\mathbf{pred} = \text{Simu}(\mathbf{par}) \in \mathbb{R}^{n_{\text{obs}}}$ worden verkregen. Deze resultaten worden vergeleken met metingen $\mathbf{obs} \in \mathbb{R}^{n_{\text{obs}}}$.

Het afregelproces bestaat uit het systematische zoeken naar waarden voor \mathbf{par} waarvoor de rekenresultaten \mathbf{pred} zo goed mogelijk lijken op de metingen \mathbf{obs} . Daarvoor moeten de aanpassingen aan de parameters zo klein mogelijk blijven, want de oorspronkelijke waarden zijn vaak al vrij goed ingevuld.

De hierboven gebruikte termen 'zo goed mogelijk lijken' en 'zo klein mogelijke aanpassingen' moeten gepreciseerd worden. Hiervoor worden de onzekerheden in de parameters en de metingen beschreven. Hiervoor worden de 'fouten' ϵ_{obs} en ϵ_{par} geïntroduceerd:

$$\begin{aligned}\mathbf{obs} &= \mathbf{pred} + \mathbf{L}_{\text{obs}}\epsilon_{\text{obs}} \\ \mathbf{par} &= \mathbf{par}_{\text{orig}} + \mathbf{L}_{\text{par}}\epsilon_{\text{par}},\end{aligned}\tag{1}$$

waarbij de L-matrices aangeven hoe 'erg' een verschil tussen metingen en rekenresultaten of een aanpassing in de parameters wordt gevonden. De fouten ϵ kunnen vrij eenvoudig worden

berekend uit (1):

$$\begin{aligned}\epsilon_{obs} &= (\mathbf{L}_{obs}^T \mathbf{L}_{obs})^{-1} \mathbf{L}_{obs}^T (\text{pred} - \text{obs}) \\ \epsilon_{par} &= (\mathbf{L}_{par}^T \mathbf{L}_{par})^{-1} \mathbf{L}_{par}^T (\text{par} - \text{par}_{orig}).\end{aligned}\quad (2)$$

Wanneer de L -matrices vierkant zijn, wordt (2) nog een beetje eenvoudiger:

$$\begin{aligned}\epsilon_{obs} &= \mathbf{L}_{obs}^{-1} \mathbf{L}_{obs}^{-T} \mathbf{L}_{obs}^T (\text{pred} - \text{obs}) = \mathbf{L}_{obs}^{-1} (\text{obs} - \text{pred}) \\ \epsilon_{par} &= \mathbf{L}_{par}^{-1} (\text{par} - \text{par}_{orig}).\end{aligned}\quad (3)$$

De *boetefunctie* J die wordt geminimaliseerd wordt verkregen door de sums-of-squares van de beide fout-termen ϵ :

$$\begin{aligned}J(\text{par}) &= |\epsilon_{obs}|^2 + |\epsilon_{par}|^2 \\ &= (\text{pred} - \text{obs})^T \mathbf{L}_{obs} (\mathbf{L}_{obs}^T \mathbf{L}_{obs})^{-2} \mathbf{L}_{obs}^T (\text{pred} - \text{obs}) + \\ &\quad (\text{par} - \text{par}_{orig})^T \mathbf{L}_{par} (\mathbf{L}_{par}^T \mathbf{L}_{par})^{-2} \mathbf{L}_{par}^T (\text{par} - \text{par}_{orig}). \\ &:= (\text{Simu}(\text{par}) - \text{obs})^T \mathbf{C}_{obs} (\text{Simu}(\text{par}) - \text{obs}) + \\ &\quad (\text{par} - \text{par}_{orig})^T \mathbf{C}_{par} (\text{par} - \text{par}_{orig}),\end{aligned}\quad (4)$$

waar de \mathbf{C} -matrices symmetrisch zijn en positief definitiet.

3 Kwadratische benadering van de boetefunctie

3.1 Taylor reeks ontwikkeling

Een kwadratische benadering van J is:

$$J(\text{par} + \text{dp}) = J(\text{par}) + \nabla^T J \text{ dp} + \text{dp}^T \nabla \nabla^T J \text{ dp} + \mathcal{O}(|\text{dp}|^3). \quad (5)$$

De gradient van de boetefunctie is dan:

$$(\nabla^T J)_i = 2(\text{Simu}(\text{par}) - \text{obs})^T \mathbf{C}_{obs} \frac{\partial \text{Simu}}{\partial \text{par}_i} + 2(\text{par} - \text{par}_{orig})^T (\mathbf{C}_{par})_{:,i}. \quad (6)$$

De Hessiaan is:

$$(\nabla \nabla^T J)_{ij} = 2 \frac{\partial \text{Simu}^T}{\partial \text{par}_j} \mathbf{C}_{obs} \frac{\partial \text{Simu}}{\partial \text{par}_i} + 2(\text{Simu}(\text{par}) - \text{obs})^T \mathbf{C}_{obs} \frac{\partial^2 \text{Simu}}{\partial \text{par}_i \partial \text{par}_j} + 2(\mathbf{C}_{par})_{j,i}. \quad (7)$$

Nabij een minimum van de boetefunctie is de Hessiaan $\nabla \nabla^T J$ positief definit. De zoekrichting dp , gegeven door

$$\text{dp} = -(\nabla \nabla^T J)^{-1} \nabla J, \quad (8)$$

is dus een dalende richting, want

$$(\nabla J)^T \text{dp} = -(\nabla J)^T (\nabla \nabla^T J)^{-1} \nabla J < 0. \quad (9)$$

Deze kwadratische benadering levert dus alleen nabij een (lokaal) minimum een dalende richting op.

3.2 Linearisatie van prediction

Een alternatieve kwadratische benadering voor de boetefunctie is het gebruik van een lineaire benadering van de predictor:

$$\text{Simu}(\text{par} + \text{dp}) = \text{Simu}(\text{par}) + \nabla^T \text{Simu} \text{ dp} + \mathcal{O}(|\text{dp}|^2). \quad (10)$$

Hiermee krijgen we de volgende kwadratische benadering van de boetefunctie:

$$J(\text{par} + \text{dp}) = J(\text{par}) + \nabla^T J \text{ dp} + \text{dp}^T \text{H}_J \text{ dp} + \mathcal{O}(|\text{dp}|^2), \quad (11)$$

waarbij de benaderde Hessiaan H_J gegeven wordt door

$$\text{H}_J := 2(\nabla \text{Simu})^T \text{C}_{obs} \nabla \text{Simu} + 2\text{C}_{par}. \quad (12)$$

Merk op dat deze benadering slechts tweede orde nauwkeurig is. De matrix H_J is echter, net zoals de echte Hessiaan (nabij een minimum), positief definit. Daarom is ook de zoekrichting $\tilde{\text{dp}}$, die uit deze kwadratische benadering volgt, een dalende richting. De zoekrichting $\tilde{\text{dp}}$ wordt gegeven door

$$\tilde{\text{dp}} = -\text{H}_J^{-1} \nabla J. \quad (13)$$

Deze kwadratische benadering is nauwkeurig als de tweede afgeleiden van de predictor-functie (die immers zijn verwaarloosd) niet zo belangrijk zijn. Deze benadering leidt wel vrij robuust naar een (lokaal) minimum, want de gevonden zoekrichting is altijd dalend.

4 De DuD-methode

De DuD-methode ('does not use derivatives') gaat uit van de benadering (11) voor de boetefunctie. Om een kwadratische benadering te krijgen, is dus slechts een benadering nodig van de gradient van de predictie-operator Simu .

In de berekening wordt een aantal sets parameters bewaard, alsmede de bijbehorende waarden van de predicties:

$$\text{P} = [\text{par}_0, \dots, \text{par}_N] \quad , \quad \text{S} = [\text{Simu}(\text{par}_0), \dots, \text{Simu}(\text{par}_N)]. \quad (14)$$

Hierbij zijn de kolommen in volgorde van oplopende boetefunctie geordend, zodat

$$J(\text{par}_0) \leq \dots \leq J(\text{par}_N). \quad (15)$$

Vervolgens worden differenties genomen van de matrices P en S :

$$\begin{aligned} \text{dP} &= [\text{par}_1 - \text{par}_0, \dots, \text{par}_N - \text{par}_0], \\ \text{dS} &= [\text{Simu}(\text{par}_1) - \text{Simu}(\text{par}_0), \dots, \text{Simu}(\text{par}_N) - \text{Simu}(\text{par}_0)] \\ &\approx \nabla \text{Simu} \text{ dP}. \end{aligned} \quad (16)$$

Een eenvoudige manier om een benadering voor de gradient ∇Simu te krijgen zou nu zijn om een stelsel op te lossen:

$$\nabla \text{Simu} \approx \text{dS}(\text{dP})^{-1}. \quad (17)$$

Vervolgens moet deze benadering van ∇Simu gebruikt worden om de benaderde Hessiaan (12) te berekenen, waarna de zoekrichting $\tilde{\text{d}}\mathbf{p}$ wordt berekend met behulp van (13). Zowel stap (17) als (13) vergt het oplossen van een stelsel vergelijkingen: bij stap (17) wordt zelfs een matrix-matrix stelsel opgelost.

De methode kan daarom nog een klein beetje efficiënter gemaakt worden. Hiervoor wordt de oplossing $\tilde{\text{d}}\mathbf{p}$ gezocht in het opspannel van dP (wat doorgaans de gehele zoekruimte is):

$$\begin{aligned} \tilde{\text{d}}\mathbf{p} &= \text{dP} \mathbf{x} \stackrel{(12,17)}{\Rightarrow} \\ -(\text{dS}^T \mathbf{C}_{obs} \text{dS} + \text{dP}^T \mathbf{C}_{par}) \text{dP} \mathbf{x} &= \text{dS}^T \mathbf{C}_{obs} (\text{Simu}(\text{par}_0) - \text{obs}) \\ &+ \text{dP}^T \mathbf{C}_{par} (\text{par}_0 - \text{par}_{orig}) \end{aligned} \quad (18)$$

Wanneer dit stelsel is opgelost, kan de zoekrichting $\tilde{\text{d}}\mathbf{p}$ worden berekend via

$$\tilde{\text{d}}\mathbf{p} = \text{dP} \mathbf{x}. \quad (19)$$

Er hoeft dus maar één stelsel te worden opgelost.

Deze rekenkundige handigheid is niet essentieel voor de methode, omdat de veronderstelling is dat het uitvoeren van een enkele simulatie veel meer rekentijd vergt dan het oplossen van deze (redelijk kleine) stelsels.

5 Modellen met een beperkte afhankelijkheid

Bij het afregelen van riviermodellen hebben we te maken met een predictor-functie Simu waarin we vantevoren al weten dat veel afhankelijkheden erg klein zijn: de waterstanden op een bepaalde plek hangen eigenlijk alleen af van de ruwheden net stroomopwaarts (of was het juist stroomafwaarts?) van dat station. De gradient ∇Simu is daarom ongeveer een diagonaalmatrix.

Er wordt al gebruik gemaakt van modellen waarin deze diagonale structuur niet helemaal klopt: hierin zijn er hier en daar ook off-diagonals aanwezig. In het algemeen hebben we dan ook te maken met *ijlheidsstructuren*: een groot aantal entries van de afgeleide ∇Simu zijn verwaarloosbaar, en enkele anderen niet.

In sommige gevallen wordt er nog een iets ingewikkelder structuur gebruikt, waarbij een algebraïsche vergelijking wordt gegeven voor een bepaalde afhankelijkheid:

$$(\nabla \text{Simu})_{ij} = \sum_{kl} \alpha_{ij,kl} (\nabla \text{Simu}_{kl}). \quad (20)$$

Dergelijke eigenschappen van een model kunnen ook wel in een DuD-methode worden verwerkt, maar in eerste instantie bestuderen we alleen het eenvoudigere geval van ijle matrix-structuren.

6 Een DuD-methode voor modellen met een beperkte afhankelijkheid

6.1 Berekening van een stukje afgeleide

In het geval van modellen met een beperkte afhankelijkheid kunnen we ongeveer net zo te werk gaan als in het geval van volledige afhankelijkheid.

We gaan weer uit van de vergelijking (17), waaruit we de afgeleide ∇Simu zullen proberen te berekenen. Omdat de afgeleide ∇Simu de onbekende is, schrijven we (17) nu als volgt:

$$dP^T \nabla\text{Simu}^T \approx dS^T \quad (21)$$

Wanneer we slechts één predictie (de i -de) beschouwen, gaat dat over een enkele rij van de matrix ∇Simu :

$$dP^T \nabla\text{Simu}(i,:)^T \approx dS(i,:)^T. \quad (22)$$

De notaties zijn sterk door Matlab geïnspireerd.

In de i -de rij komen alleen nonzeros voor in de afgeleide ∇Simu op de kolommen $\text{nnz}\{i\}$. Laten we deze nullen uit de vergelijking weg, dan krijgen we

$$dP(\text{nnz}\{i\},:)^T \nabla\text{Simu}(i,\text{nnz}\{i\})^T \approx dS(i,\text{nnz}\{i\})^T. \quad (23)$$

Hieruit kunnen we de gezochte afgeleiden berekenen, wanneer de rang van $dP(\text{nnz}\{i\},:)$ voldoende is. Is deze meer dan voldoende, dan kunnen we kiezen of we het meeste vertrouwen hebben in een kleinste kwadraten-oplossing of in de oplossing waarin enkele rijen van $dP(\text{nnz}\{i\},:)$ worden verwaarloosd. Voorlopig is de kleinste kwadraten-benadering denk ik het eenvoudigst, en mogelijk ook het meest robuust.

Op deze manier kan de afgeleide worden berekend, als een klein aantal parameter-keuzen is doorgerekend. Als we er van uitgaan dat er geen lineaire afhankelijkheden optreden in de te inverteren matrices $dP(\text{nnz}\{i\},:)$, dan is het aantal parameterkeuzen $n\text{Try}$ dat moet worden doorgerekend (en onthouden) gelijk aan

$$n\text{Try} = \max(\text{length}(\text{nnz}\{i\})) \quad (24)$$

De afgeleide ∇Simu , die immers een ijle matrix is, kan efficiënt worden geïnverteerd. Omdat, naar verwachting, veruit de meeste rekentijd nodig is voor het uitvoeren van simulaties, zal echter gebruik worden gemaakt van volle matrix-inversie voor het berekenen van de zoekrichting.

6.2 Hergebruik van afhankelijkheidspatronen

Wanneer er veel observaties zijn en weinig parameters, bijvoorbeeld wanneer de observaties de vorm van *tijdreeksen* aannemen, dan komt het nonzero-patroon $\text{nnz}\{i\}$ vele malen voor. In dat geval kunnen we beter kijken naar *alle* predicties die het patroon $\text{nnz}\{i\}$ hebben. De bijbehorende indices noemen we $\text{ipred}\{i\}$. Nu kunnen we een veel groter deel van de afgeleide ineens berekenen:

$$dP(\text{nnz}\{i\}, :)^T \nabla \text{Simu}(\text{ipred}\{i\}, \text{nnz}\{i\})^T \approx dS(\text{ipred}\{i\}, \text{nnz}\{i\})^T. \quad (25)$$

6.3 Betrouwbaarheid van de benadering

In sommige gevallen zal de benaderde afgeleide niet nauwkeurig zijn. Dat gebeurt in de volgende gevallen:

- Het matrix-blokje $dP(\text{nnz}\{i\}, :)$ is (bijna) singulier.
Dat kan bijvoorbeeld het geval zijn wanneer de parameters die verantwoordelijk zijn voor een groep predicties al enkele iteraties correct bekend zijn: ze zijn dan al enkele iteraties niet aangepast, dus de bijbehorende entries van dP zijn nul.
- Het matrix-blokje $dP(\text{nnz}\{i\}, :)$ is klein.
Een groot aantal afhankelijkheden zijn verwaarloosd. Wanneer het verwaarloosde gedeelte toch groter is dan het niet-verwaarloosde gedeelte, kunnen onnauwkeurigheden optreden.

Wanneer deze problemen optreden, kunnen ze worden vermeden door de benaderde afgeleide opnieuw te berekenen, met een geheel nieuwe, en slim gekozen, matrix dP (en bijbehoren de dS).

Een verstandige keuze voor dP kan worden opgesteld, maar we wachten hier nog even mee totdat dit nodig blijkt te zijn.

6.4 Opgeven van de afhankelijkheidsstructuur

Voor het gebruik van de ijle DuD-methode zal de gebruiker moeten kunnen aangeven welke afhankelijkheden kunnen worden verwaarloosd. Dit zal moeten gedaan worden door middel van een XML-file. Het opgeven van een dergelijke afhankelijkheidsstructuur kan op twee manieren: rij-gewijs en kolomsgewijs. In Tabel 1 wordt een idee gegeven van de manier waarop de afhankelijkheden kunnen worden gespecificeerd. Er kan zowel worden opgegeven dat een bepaalde predictie afhangt van bepaalde parameters, als dat bepaalde parameters een invloed hebben op bepaalde observaties. Alle afhankelijkheden die niet genoemd zijn, worden verwaarloosd.

```
<nonzeros>
  <obs id="<obs name 1>">
    <depends_on>
      <param id="<param name 1>"/>
      <param id="<param name 2>"/>
    </depends_on>
  </obs>
  <obs id="<obs name 2>">
    <depends_on>
      <param id="<param name 1>">
      <param id="<param name 2>">
      <param id="<param name 3>">
    </depends_on>
  </obs>
  <param id="<param name 3>">
    <influences>
      <obs id="<obs name 1>"/>
      <obs id="<obs name 2>"/>
      <obs id="<obs name 3>"/>
    </influences>
  </param>
</nonzeros>
```

Tabel 1: *stukje XML-code, om te illustreren hoe de afhankelijkheden kunnen worden opgegeven.*

Een dergelijke manier van specificeren wordt een probleem wanneer er heel veel observaties zijn, bijvoorbeeld wanneer de observaties tezamen een tijdreeks vormen. In dat geval moeten de namen van observaties een groep waarnemingen kunnen aanduiden (bijvoorbeeld alle observaties van een bepaald station), of moeten wildcard-characters worden toegestaan in de observatie-namen.