

**Comment on 'P. Xin et al., *Advances in Water Resources* 91 (2016) 1–10'**

Zaadnoordijk, Willem J.

**DOI**

[10.1016/j.advwatres.2016.08.003](https://doi.org/10.1016/j.advwatres.2016.08.003)

**Publication date**

2016

**Document Version**

Accepted author manuscript

**Published in**

*Advances in Water Resources*

**Citation (APA)**

Zaadnoordijk, W. J. (2016). Comment on 'P. Xin et al., *Advances in Water Resources* 91 (2016) 1–10'. *Advances in Water Resources*, 96, 438-439. <https://doi.org/10.1016/j.advwatres.2016.08.003>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

in Water Resources

Elsevier Editorial System(tm) for Advances

Manuscript Draft

Manuscript Number:

Title: Comment on 'P. Xin et al./Advances in Water Resources 91 (2016) 1-10'

Article Type: Discussion

Keywords: storage coefficient  
specific yield  
porosity

Corresponding Author: Dr. Willem Jan Zaadnoordijk, Ph.D.

Corresponding Author's Institution:

First Author: Willem Jan Zaadnoordijk, Ph.D.

Order of Authors: Willem Jan Zaadnoordijk, Ph.D.

**Comment on 'P. Xin et al./Advances in Water Resources 91 (2016) 1-10'**

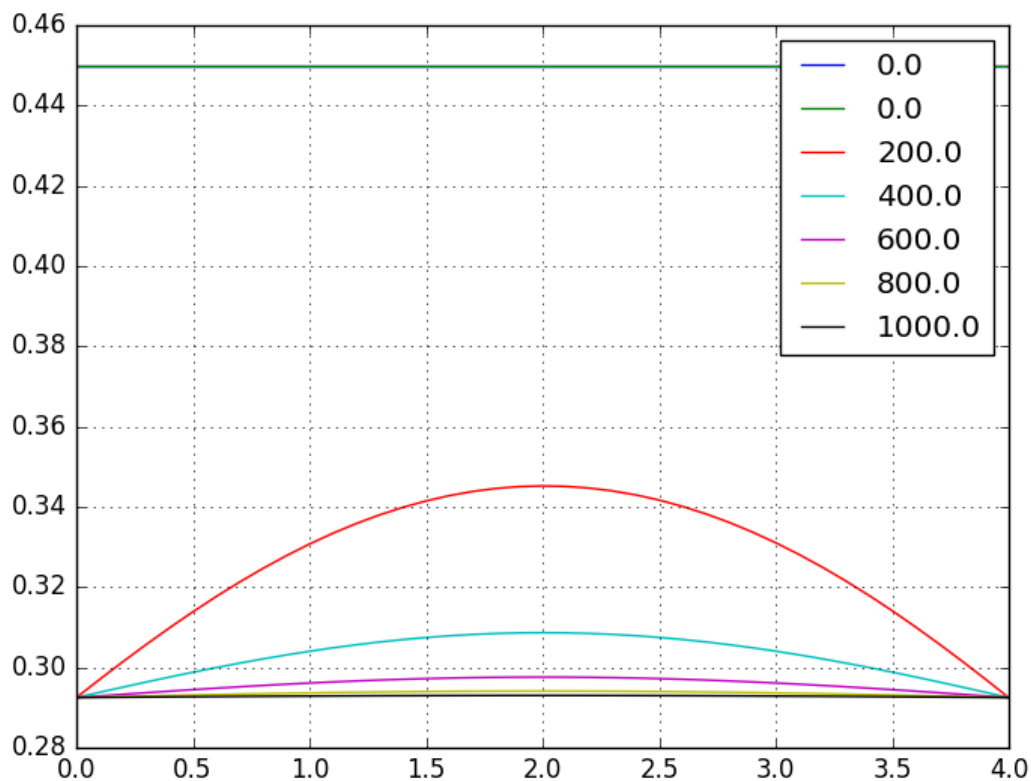
1 Willem J. Zaadnoordijk, KWR Watercycle Research Institute, Nieuwegein, the Netherlands (until 1 July  
2 2016), TNO-Geological Survey of the Netherlands, Utrecht, the Netherlands (starting 1 July 2016);  
3 also at: Water Resources Section, Faculty of Civil Engineering and Geosciences, Delft University of  
4 Technology, Delft, Netherlands  
5

6 [wjz\\_rdam@yahoo.co.uk](mailto:wjz_rdam@yahoo.co.uk)  
7

8 keywords: storage coefficient, specific yield, porosity  
9

10  
11 Xin et al. (2016) present an approximate analytical solution to a variable saturated formulation of a  
12 drainage problem. They compared their solution to a saturated flow-based model and found a large  
13 discrepancy. However, the discrepancy is due to the value of the storage coefficient used and not to  
14 the model formulation.  
15

16  
17 Xin et al. (2016) used the total porosity as the storage coefficient in the saturated flow-based model,  
18 while water table storage (often referred to as specific yield) usually is smaller (see a textbook such  
19 as Fitts, 2013). They present results from experiments in a sand flume and report a water release of  
20 12% of the volume between the initial and final water table. Using this value of 0.12 as storage  
21 coefficient in a 1-dimensional calculation based on the Dupuit Forchheimer assumption gives a result  
22 (figure below) which is very similar to the outcome of the analytical solution presented by Xin et al.  
23 (2016) in their Figure 5a.  
24  
25 (2016) in their Figure 5a.  
26



58 Note: Figure 5a of Xin et al. (2016) shows the initial groundwater table erroneously at 0.4 m instead  
59 of the correct value of 0.45 m.  
60  
61  
62  
63  
64  
65

## References

Charles R. Fitts (2013) Groundwater Science, second edition, Academic Press, Waltham MA, USA.

Pei Xin, Han-Cheng Dan, Tingzhang Zhou, Chunhui Lu, Jun Kong, Ling Li (2016) An analytical solution for predicting the transient seepage from a subsurface drainage system, Advances in Water Resources, , <http://dx.doi.org/10.1016/j.advwatres.2016.03.006>.

## Appendix Python script used to produce figure

```
14 # explicit finite differences
15 # problem of Xin et al.,2016
16
17 import numpy as np
18
19 h0 = 0.45
20 k = 3.8e-3 # m/s
21 S = 0.12
22 hBnd = 0.2925
23
24 L = 4.0
25 n = 40
26 dx = L/n
27
28 dt = 0.1
29 tEnd = 1000.
30 nt = int(tEnd/dt)
31 tPlot = [0., 200., 400., 600., 800., 1000.]
32 iPlot = 1
33 nPlot = len(tPlot)
34
35 x = np.arange(0.0, L+dx, dx)
36 nx = len(x)
37 h = np.ones(nx)*h0 # initial heads
38
39 hTimeT = []
40 hTimeT.append( h.copy() )
41 tCurrent = 0.0
42 h[0] = hBnd # boundary head t>0
43 h[-1] = hBnd # boundary head t>0
44 while iPlot<nPlot :
45     hl = h[0:-2]
46     hm = h[1:-1]
47     hr = h[2:]
48     ql = k*hm*(hl-hm)/dx
49     qr = k*hr*(hr-hm)/dx
50     dh = (ql+qr)*dt/S/dx
51     hn = hm+dh
52     h[1:-1] = hn
53     tCurrent += dt
54     if tCurrent >= tPlot[iPlot] :
55         hTimeT.append(h.copy() )
56         iPlot += 1
57
58 import matplotlib.pyplot as plt
59 plt.hold('on')
60 for i in range(nPlot) :
61     plt.plot(x, hTimeT[i], label=str(tPlot[i]) )
62 plt.legend()
63 plt.grid(True)
64 plt.savefig('efd.png')
65 plt.show()
```

Figure  
[Click here to download high resolution image](#)

