Technical Report 2008/02:

# Pressure Driven Demand Extension for EPANET (EPANETpdd)

M.S.Morley & C.Tricarico

## Introduction

Predominantly, Demand-Driven hydraulic simulators such as EPANET used in optimization processes are configured to deliver water even when there is insufficient pressure to do so – Demand-Driven network solver (as EPANET – Rossman, 2000). In the analysis of structurally inadequate systems, however, recent studies [Germanopoulos, 1985, Hayuti & Burrows, 2004, Soares et al., 2003], have highlighted limitations related to the use of such demand-driven solvers.

Initially, the sole requirement for the PDD extension was for it to be able to determine more accurately the non-revenue water unsupplied in a pressure-deficient network in order to better estimate the network's Economic Level of Reliability [Tricarico et al., 2006]. A logical extension of that work required that the PDD simulator should also be able to operate in an EPS mode. As well as EPS, the application of the simulator to the Neptune project introduced two further requirements. PD demand nodes need to be able to exist in parallel with EPANET's conventional emitters and the ability to specify emitter exponents on an individual rather than global basis. This functionality is required to simulate bursts in networks: PDD nodes will be used to observe the effects on demand nodes whilst EPANET's standard emitters will be used to simulate unconstrained bursts, which will be represented by different emitter characteristics.

## Implementation

The **EPANETpdd** extension has been derived from two existing modifications to the core EPANET library: **OOTEN** (Object Oriented Toolkit for EPANET) [van Zyl et al. 2003], provided by the University of Johannesburg and a revised PDD version of **EPANET** obtained from its author, Lewis Rossman.

Version 2.0 of **EPANET** already includes a pressure driven modelling element, the "emitter", ordinarily used for modelling outflow owing to leakage or through hydrants etc. Both the **OOTEN** and **EPANET** implementations of PDD replace the standard emitter device with one that can simulate PDD. Consequently, the standard emitter representations are no longer available with either extension. These standard emitters have an outflow described by the following equation:

$$Q_{demand} = S \cdot (H_{available})^{exp} \qquad \text{i)}$$

| | | | |
|---|---|---|---|
| $H_{available}$ | Available Head at node | $S$ | Emitter coefficient |
| $Q_{demand}$ | Demand outflow from the node | $exp$ | Emitter exponent |

As can be seen from Equation i), whilst the standard emitter is well suited to modelling leakage or other purely pressure-driven outflow, it is of little use for PDD modelling given that it has no upper bound for the outflow to peg it to the required demand. Moreover, in the event of negative pressure at this node, the standard emitter is deemed to have a negative demand – i.e. supplying water to the network.

## OOTEN

The initial implementation of the simulator was derived from PDD code [Cheung et al., 2005] extracted from the **OOTEN** library. **OOTEN** introduces a term, $H_{critical}$ ($H_{desired}$ in some literature), which describes a pressure at which 100% of the required demand can be considered to be delivered – for the purposes of this library, this "Critical" pressure is fixed across the network. The standard emitter object is replaced with one that has a capped maximum flow equal to the required demand, $Q_{required}$. In addition, in the event of negative or zero available head, the flow is constrained to zero – avoiding the back-flow effects seen with the standard emitter behaviour in EPANET.

Outside of the constrained flow behaviours, the emitter behaves as a standard EPANET emitter - Table 1 shows the behaviour of the PDD emitter as used in **OOTEN**. Thus, the emitter coefficient, *S*, defined in the input file is some term related to the magnitude of demand at the node:

$$S = \frac{Q_{required}}{H_{critical}{}^{exp}} \qquad \text{ii)}$$

**Table 1: OOTEN PDD Emitter behaviour**

| Condition | Demand |
|---|---|
| $\boldsymbol{H_{available} < Elevation}$ | $Q_{demand} = 0$ |
| $\boldsymbol{Elevation \le H_{available} < H_{critical}}$ | $Q_{demand} = Q_{required} \cdot \left(\frac{H_{available}}{H_{critical}}\right)^{exp}$ |
| $\boldsymbol{H_{available} \ge H_{critical}}$ | $Q_{demand} = Q_{required}$ |

The **OOTEN** code was tested on a number of networks and was found to function acceptably on most networks. However, in some situations, notably the Piedemonte San Germano network models, the model would fail to converge – apparently because of the highly redundant nature of this model, coupled with its extreme low flow scenarios. When coupled to Least Cost Design (LCD) optimization software [Kapelan et al., 2003], the **OOTEN** code exhibited further issues when attempting to solve many pressure-deficient networks – a common requirement in the early stages of an evolutionary optimization process. Consequently, detection routines were added to the LCD optimizer to allow it to disregard solutions where the solver had apparently failed to converge – although not all failure conditions could be identified. This allowed the preparation of the results presented by de Marinis et al. [2007] – albeit with some uncertainty as to their accuracy.

**OOTEN** introduces new code in parallel with the normal EPANET hydraulic solver routines, for its replacement emitters. However, it would appear that in duplicating a number of the solver routines, some instability has been introduced into the library – causing the convergence problems witnessed. Extensive analysis revealed that tweaking some of the initialisation conditions for the PDD side of the solver allowed some networks to converge, which previously would not. However, no rationale was identified for these changes having this effect and it did not prove possible to develop code that could automatically adjust the initialisation conditions to promote convergence.

## EPANET

An alternative PDD extension to **EPANET** by Lewis Rossman was made available to the authors and this proved to provide a consistently robust solver under all conditions and for all network topologies. The revised **EPANET** code provided introduces new states for emitters, *OPEN*, in which the outflow is constrained to the required demand for that node and *ACTIVE*, where the emitter behaves as a standard **EPANET** emitter. A further state, *CLOSED*, is used when the available head is lower than the nodal elevation – preventing the negative demand issue encountered with the standard emitter. Table 2 shows the revised emitter states. The iterative nature of the hydraulic simulation allows the emitter states to be formulated in terms of two different variables (available head and available flow) with the simulator switching states as necessary to converge the solution:

**Table 2: Rossman's revised EPANET emitter states**

| Condition | Emitter State | Demand |
|---|---|---|
| $\boldsymbol{H_{available}} < Elevation$ | CLOSED | $Q_{demand} = 0$ |
| ***All other conditions*** | ACTIVE | $Q_{demand} = S \cdot (H_{available})^{exp}$ |
| $\boldsymbol{Q_{available} \geq Q_{required}}$ | OPEN | $Q_{demand} = Q_{required}$ |

$Q_{available}$ Available Flow at node

Rossman's updated EPANET capabilities were merged into the current version of EPANET (the PDD extensions being based on the former version) and were found to solve reliably all network configurations tested, including those that would not converge with **OOTEN**. As with the **OOTEN** implementation, the demand specification for a node is defined in terms of the emitter coefficient, *S.*

## EPANETpdd

Owing to the absence of the Critical Pressure concept in Rossman's code it was decided to attempt to merge his solver with the Emitter behaviour of **OOTEN**. Accordingly, **EPANETpdd** uses modified state transition conditions (Table 3) with the more robust hydraulic simulation facilitated by Rossman's modified emitters.

**Table 3: EPANETpdd emitter states: non-zero $H_{critical}$**

| Condition | Emitter State | Demand |
|---|---|---|
| $\boldsymbol{H_{available} < H_{min}}$ | CLOSED | $Q_{demand} = 0$ |
| $\boldsymbol{H_{min} \leq H_{available} < H_{critical}}$ | ACTIVE | $Q_{demand=} \begin{cases} a) & Q_{required} \cdot \left(\frac{H_{available} - H_{min}}{H_{critical} - H_{min}}\right)^{exp} \\ b) & Q_{required} \cdot \sin^2\left(\pi \frac{H_{available} - H_{min}}{2(H_{critical} - H_{min})}\right) \\ c) & Q_{required} \cdot \left(\frac{(H_{available} - H_{minimum})^2(3H_{critical} - 2H_{available} - H_{min})}{(H_{critical} - H_{min})^3}\right) \end{cases}$ |
| $\boldsymbol{H_{available} \geq H_{critical}}$ | OPEN | $Q_{demand} = Q_{required}$ |

$H_{min}$ Head at node for which demand begins to be delivered.

$H_{critical}$ Head at node for which 100% of demand can be considered satisfied.

**EPANETpdd** introduces the term $H_{min}$ which ordinarily represents the elevation of the node. However, $H_{min}$ is treated as an independent variable internally and can be used to represent, for example, the elevation of an outlet attached to a demand node where that outlet is above the nominal elevation for the node.

By default, **EPANETpdd** uses equation *a)* [Wagner et al., 1988] for specifying the node/head relationship for the *ACTIVE* emitter state. In addition two other, similar, representations are available: *b)* [Tucciarelli et al., 1999] and *c)* [Fujiwara et al., 1998] by recompiling the code accordingly.

In the event that the critical pressure, $H_{critical}$, is zero the ACTIVE state is no longer available as demand is deemed fulfilled as soon as there is any available pressure above $H_{min}$ (Table 4) – behaviour that matches the results obtained using Rossman's updated EPANET emitters. This representation can be seen as curve e) in Figure 1.

**Table 4: EPANETpdd emitter states: $H_{critical}$ = 0**

| Condition | Emitter State | Demand |
|---|---|---|
| $\boldsymbol{H_{available} < H_{min}}$ | CLOSED | $Q_{demand} = 0$ |
| $\boldsymbol{H_{available} \geq H_{min}}$ | OPEN | $Q_{demand} = Q_{required}$ |

Whilst **OOTEN** and **EPANETpdd** effectively use the same emitter states, there are several important differences in the way they are implemented. Firstly, **EPANETpdd** allows for per-node specification of the $H_{critical}$, $H_{min}$ and *exp* variables allowing for greater flexibility in the description of a hydraulic network problem as well as permitting the concurrent use of the standard EPANET emitters. In addition, EPANETpdd does not require the specification of the outflow in terms of the emitter coefficient, *S*. Instead, the coefficient is calculated dynamically according to the demand specified in the input file – making it much simpler to convert a model between demand-driven and pressure-driven modes.

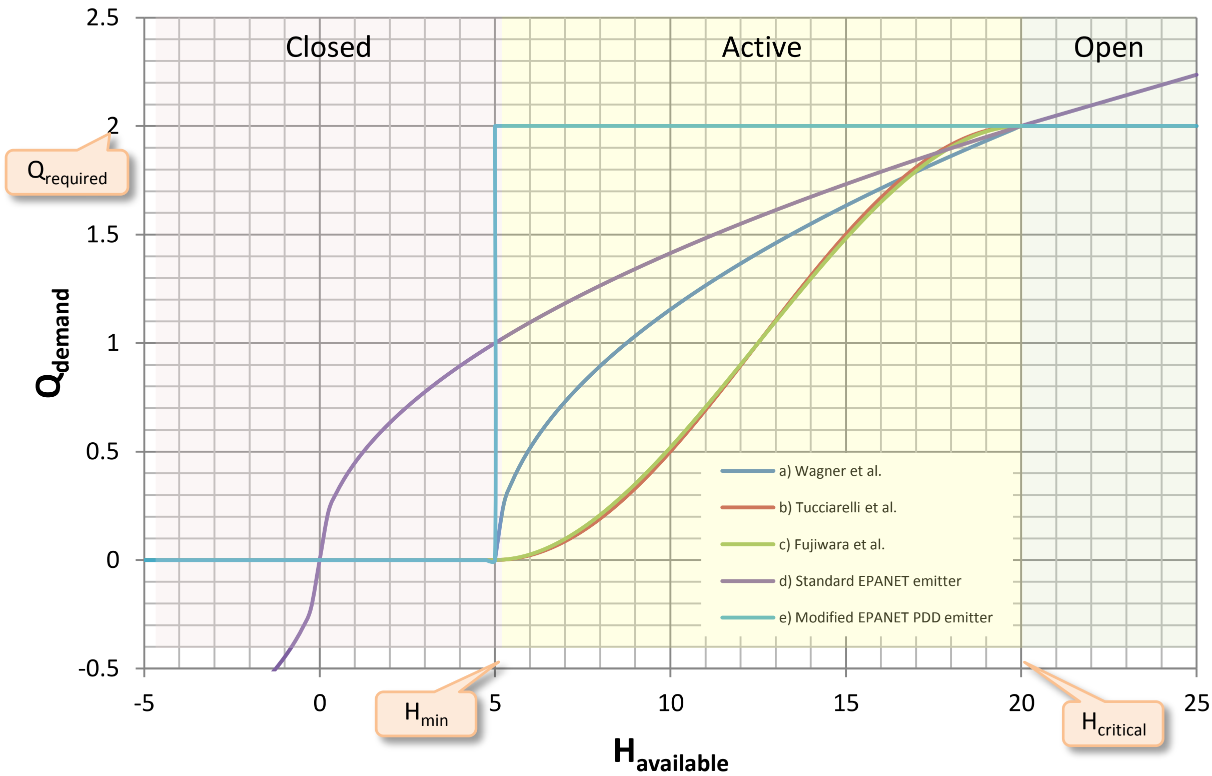


**Figure 1: Head/Flow relationship for a Pressure Driven Demand node as implemented in EPANETpdd**

**Error! Reference source not found.** shows an example of PDD node/flow relationships for each of the three formulations *(a-c)* used for an active PDD emitter as well as the behaviour of a standard EPANET emitter *(d)* and the modified emitter developed by Rossman *(e)*,equivalent to a PDD node with $H_{critical}$=0.0). The node has a $Q_{required}$ (nominal demand) of 2 l/s, a critical pressure of 20m and a minimum pressure of 5m. As can be seen from curve *e)*, the standard EPANET emitter does not have the ability to specify an $H_{min}$ value and so attempts to deliver demand as soon as there is available pressure. This curve also illustrates the negative demand (i.e. supply) situation that occurs when the available pressure is below the elevation of a node.

### Extended Period Simulation

The functionality of the **EPANETpdd** code has been further extended to incorporate Extended Period Simulation – something that neither the **OOTEN** nor revised **EPANET** algorithms facilitated. This is accommodated through the dynamic computation of demand ahead of each timestep. The nodal demand is then converted into an appropriate emitter coefficient and applied to the node accordingly.

## Validation

In order to verify the performance of the **EPANETpdd** solver it has been compared with other PDD results found in the literature. In each case the results obtained with the OOTEN solver are identical, whilst those...

### Example 1

Ang & Jowitt [2006] demonstrate their PDD methodology on three problems. The first is a simple serial network first described by Gupta & Bhave [1996]. This network consists of a single reservoir feeding four nodes in series. The four nodes are modelled as PD Demand Nodes with a critical pressure of 0.0. Table 5 shows a comparison of the results between Ang & Jowitt's methodology and **EPANETpdd**:

**Table 5: Results for Serial Network: Ang & Jowitt [A] vs. EPANETpdd [B]**

| Head at source *(m)* | | Outflow at Node *(m³/min)* / Head *(m)* | | | | | | | | Total Supply *(m³/min)* | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | | 2 | | 3 | | 4 | | | |
| | | A | B | A | B | A | B | A | B | A | B |
| **85.00** | Outflow | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.01 |
| | Head | 85.00 | 85.00 | 85.00 | 85.00 | 85.00 | 85.00 | 85.00 | 85.00 | | |
| **89.08** | Outflow | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 2.73 | 2.73 | 2.73 | 2.73 |
| | Head | 88.71 | 88.71 | 88.00 | 88.00 | 86.50 | 86.50 | 85.00 | 85.00 | | |
| **90.98** | Outflow | 0.00 | 0.00 | 2.00 | 2.00 | 0.00 | 0.00 | 2.73 | 2.74 | 4.73 | 4.74 |
| | Head | 89.96 | 89.96 | 88.00 | 88.00 | 86.50 | 86.50 | 85.00 | 85.00 | | |
| **91.03** | Outflow | 0.00 | 0.00 | 2.00 | 2.00 | 0.00 | 0.00 | 2.75 | 2.75 | 4.75 | 4.75 |
| | Head | 90.00 | 90.00 | 88.03 | 88.03 | 86.52 | 86.52 | 85.00 | 85.00 | | |
| **91.97** | Outflow | 2.00 | 2.00 | 2.00 | 2.00 | 0.00 | 0.00 | 2.75 | 2.75 | 6.75 | 6.75 |
| | Head | 90.00 | 90.00 | 88.03 | 88.03 | 86.52 | 86.52 | 85.00 | 85.00 | | |
| **96.82** | Outflow | 2.00 | 2.00 | 2.00 | 2.00 | 0.00 | 0.00 | 4.00 | 4.00 | 8.00 | 8.00 |
| | Head | 94.11 | 94.12 | 91.08 | 91.09 | 88.04 | 88.06 | 85.00 | 85.02 | | |
| **98.78** | Outflow | 2.00 | 2.00 | 2.00 | 2.00 | 0.00 | 0.00 | 4.00 | 4.00 | 8.00 | 8.00 |
| | Head | 96.08 | 96.08 | 93.04 | 93.04 | 90.00 | 90.00 | 86.96 | 86.97 | | |
| **100.00** | Outflow | 2.00 | 2.00 | 2.00 | 2.00 | 0.40 | 0.40 | 4.00 | 4.00 | 8.40 | 8.40 |
| | Head | 97.04 | 97.05 | 93.62 | 93.62 | 90.00 | 90.00 | 86.96 | 86.97 | | |
| **109.86** | Outflow | 2.00 | 2.00 | 2.00 | 2.00 | 3.00 | 3.00 | 4.00 | 4.00 | 11.00 | 11.00 |
| | Head | 104.99 | 105.00 | 98.55 | 98.57 | 90.00 | 90.02 | 86.96 | 86.98 | | |

As can be seen, there is a very good correlation between the two sets of results.

Cheung et al. [2005] demonstrate their PDD solver (i.e. that found in **OOTEN**) on a similar version of this network with the same basic network parameters, but with a Critical Pressure of 20m and source-head of 100m. Note that the format and units of the results presented in Table 6 have been changed to match the representation used by Ang & Jowitt in their analysis of the network in order that direct comparison may be made with Table 5.

**Table 6: Results for Serial Network (with Critical Pressure 20m): Cheung et al. [A] vs EPANETpdd [B]**

| Node | Normal | | | | Fire Flow | | | |
|---|---|---|---|---|---|---|---|---|
| | Head *(m)* | | Demand *(m³/min)* | | Head *(m)* | | Demand *(m³/min)* | |
| | A | B | A | B | A | B | A | B |
| **1** | 98.81 | 98.81 | 1.33 | 1.33 | 98.29 | 98.29 | 1.28 | 1.28 |
| **2** | 97.50 | 97.51 | 1.38 | 1.38 | 96.16 | 96.16 | 1.28 | 1.28 |
| **3** | 96.30 | 96.30 | 1.69 | 1.69 | 93.56 | 93.55 | 1.26 | 1.27 |
| **4** | 96.20 | 96.16 | 0.75 | 0.75 | 92.36 | 92.35 | 2.42 | 2.42 |
| | | | 5.15 | 5.15 | | | 6.24 | 6.25 |

Comparing Table 6 with the appropriate (100m head) row from Table 5 clearly shows the influence of the Critical Pressure. Ang & Jowitt's solver, in common with Rossman's updated **EPANET**, attempt to deliver maximum demand as soon is there is available pressure at the node. Whereas **OOTEN** and **EPANETpdd** will not do so until the Critical Pressure is attained – thus explaining the higher flows and lower pressures seen in Table 5. Surprisingly, there are minor differences between the results using **OOTEN**'s methodology, replicated by the authors, and those obtained by Cheung et al. This is likely to be due to the underlying EPANET solver used by EPANETpdd having been modified to use double-precision numbers throughout.

### Example 2

The second example network comprises a single reservoir feeding six demand nodes through eight pipes in a looped configuration. Two scenarios are examined: fire-flow in one of the distal nodes of the network and the failure of one of the pipes for each of nine levels of the reservoir. Again, the demand nodes are modelled in **EPANETpdd** as PD demand nodes with a Critical Pressure of 0.0. Table 7 and Table 8 show the results obtained by **EPANETpdd** in comparison with those reported by Ang & Jowitt (2006):

**Table 7: Results of Single Source Network: Ang & Jowitt [A] vs. EPANETpdd [B] – Fire-Flow Scenario**

| Head at source *(m)* | Outflow at Node *(l/s)* | | | | | | | | | | | | Total Supply *(l/s)* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | | |
| | A | B | A | B | A | B | A | B | A | B | A | B | A | B |
| **86** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.29 | 10.35 | 10.15 | 10.09 | 20.45 | 20.44 |
| **88** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 18.62 | 18.65 | 18.38 | 18.34 | 37.00 | 37.00 |
| **90** | 0.00 | 0.00 | 0.00 | 0.00 | 9.87 | 9.86 | 0.00 | 0.00 | 22.77 | 22.79 | 22.72 | 22.69 | 55.36 | 55.35 |
| **92** | 0.00 | 0.00 | 0.00 | 0.00 | 25.00 | 25.00 | 9.04 | 9.03 | 22.98 | 22.93 | 22.86 | 22.90 | 79.88 | 79.86 |
| **94** | 25.00 | 25.00 | 0.00 | 0.00 | 25.00 | 25.00 | 14.61 | 14.60 | 23.27 | 23.27 | 22.91 | 22.90 | 110.79 | 110.77 |
| **96** | 25.00 | 25.00 | 7.04 | 7.03 | 25.00 | 25.00 | 24.15 | 24.15 | 24.36 | 24.42 | 22.91 | 22.84 | 128.46 | 128.44 |
| **98** | 25.00 | 25.00 | 20.99 | 20.98 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 24.38 | 24.38 | 145.38 | 145.36 |
| **100** | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 29.93 | 29.92 | 154.93 | 154.92 |
| **117.56** | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 75.00 | 75.00 | 200.00 | 200.00 |

**Table 8: Results of Single Source Network: Ang & Jowitt [A] vs. EPANETpdd [B] – Pipe Failure Scenario**

| Head | Outflow at Node *(l/s)* | Total Supply |
|---|---|---|

| at source *(m)* | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | A | B | A | B | A | B | A | B | A | B | A | B |
| **86** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 9.63 | 9.60 | 8.27 | 8.29 | 17.89 | 17.89 |
| **88** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 17.42 | 17.43 | 14.97 | 14.95 | 32.38 | 32.38 |
| **90** | 0.00 | 0.00 | 0.00 | 0.00 | 0.90 | 0.72 | 0.00 | 0.00 | 22.77 | 22.71 | 19.55 | 19.61 | 43.21 | 43.03 |
| **92** | 25.00 | 25.00 | 1.52 | 1.52 | 10.97 | 10.96 | 0.00 | 0.00 | 22.77 | 22.72 | 19.55 | 19.60 | 79.81 | 79.79 |
| **94** | 25.00 | 25.00 | 24.12 | 24.11 | 17.25 | 17.23 | 0.00 | 0.00 | 22.72 | 22.62 | 19.59 | 19.69 | 108.68 | 108.67 |
| **96** | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 0.00 | 0.00 | 24.29 | 24.28 | 20.94 | 20.95 | 120.23 | 120.22 |
| **98** | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 2.56 | 2.55 | 25.00 | 25.00 | 25.00 | 25.00 | 127.56 | 127.55 |
| **100** | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 10.38 | 10.37 | 25.00 | 25.00 | 25.00 | 25.00 | 135.38 | 135.37 |
| **104.27** | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 150.00 | 150.00 |

### Example 3

The final example related by Ang & Jowitt is a twin reservoir system feeding nine demand nodes through fourteen pipes in a looped configuration. Partial results are reported in the literature and Table 9 illustrates the equivalent results obtained with **EPANETpdd**.

**Table 9: Selected Results of Multiple Source Network: Ang & Jowitt [A] vs. EPANETpdd [B] – Fire-Flow Scenario**

| Head at source *(m)* | Outflow at Node *(l/s)* | | | | | | Total Supply *(l/s)* | |
|---|---|---|---|---|---|---|---|---|
| | 2 | | 6 | | 9 | | | |
| | A | B | A | B | A | B | A | B |
| **86** | 4.63 | 4.62 | 21.16 | 21.15 | 54.26 | 54.26 | 230.05 | 230.03 |

## Use

Use of the EPANETpdd extension is straightforward – the only requirement being to configure the input files accordingly. No internal changes to existing software should be necessary. To convert a network to Pressure-Driven usage, it is simply necessary to add an entry in the input file under the [EMITTERS] section for each of the demand nodes (see PDD emitter type below). Whilst it is possible to run the EPANETpdd model with a mixture of Demand-Driven and Pressure-Driven nodes, this scenario is not recommended and the results are undefined. Having created an EMITTER entry for each of the demand nodes, it is easy to revert to a Demand-Driven model by commenting out each relevant line in the [EMITTERS] section.

It is also possible to incorporate EPANET's standard emitters in a PD model – though it is important to remember the negative-pressure behaviour of these devices: a loosely constrained PD node may be a better alternative for modelling.

### Emitter Types

This section details the types of emitter that can be added to an **EPANETpdd** model and its syntax in the input file.

**Normal** EPANET's standard emitters may be defined as usual. Although EPANET allows a junction to be both a demand node and an emitter, it is important that an emitter has no demand associated with it as this is how normal emitters and PDD emitters are distinguished. In the example below, the emitter exponent is retrieved from the global setting in the [OPTIONS] section of the input file.

```
[JUNCTIONS]
; Node ID      Elevation
10             117.9

[EMITTERS]
; Node ID      Coefficient
10             0.286
```

**Normal Exponent** To define a standard EPANET emitter with a custom exponent, the exponent is added as an additional parameter in the emitter specification, thus:

```
[JUNCTIONS]
; Node ID      Elevation
10             117.9

[EMITTERS]
; Node ID      Coefficient   Exponent
10             0.286         0.54
```

**PDD** A PDD demand node can be defined by specifying a demand node as normal and then making an entry for it in the emitters section. The emitter coefficient is computed automatically from the supplied demands whether they are specified in the [JUNCTIONS] or [DEMANDS] section or as patterns for EPS models. Instead of the emitter coefficient, the value for $H_{critical}$ (the Critical Pressure at which demand is considered to be satisfied) is specified. If no Critical Pressure is desired then it should be given the value 0.0. Optionally, a value may be supplied for $H_{min}$, the pressure at which supply from the node is deemed to become available. If omitted, a value of 0.0 is assumed – i.e. that supply will begin when total head exceeds the nodal elevation.

```
[JUNCTIONS]
; Node ID      Elevation     Demand
10             117.9         1.158
20             112.0         0.435

[EMITTERS]
; Node ID      Hcritical     Hminimum
10             20.0
20             20.0          1.25
```

**PDD Exponent** To apply a custom exponent to a PDD node, the exponent value should be appended to the emitter specification. If using a custom exponent it is also required to specify the value of $H_{min}$, even if it is zero, thus:

```
[JUNCTIONS]
; Node ID      Elevation     Demand
10             117.9         1.158

[EMITTERS]
; Node ID      Hcritical     Hminimum      Exponent
10             20.0          0.0           0.54
```

## Extensions to EPANET DLL

In order to accommodate the PDD elements added to the hydraulic solver, a number of additional return values have been made available through the EPANET DLL's API functions.

**Table 10: Additional options/variations for ENgetnodevalue/ENsetnodevalue**

| C/C++ Constant name | Value | Read/Write | Purpose |
|---|---|---|---|
| **EN_DEMAND** | 9 | Read | Returns nodal demand (available only after running model). For PDD models this will be the actual demand delivered, $Q_{demand}$. For EPS models, this value is the demand for the current timestep only. For DD models, this will always be the demand specified in the input file. |
| **EN_COORDINATEX** | 100 | Read/Write | Returns X coordinate of node |
| **EN_COORDINATEY** | 101 | Read/Write | Returns Y coordinate of node |
| **EN_CALCULATEDDEMAND** | 110 | Read | Returns aggregated demand for the node. This value is available prior to the model being run. For EPS models, this value is the demand for the current timestep only. For PDD models, this value is the required demand, $Q_{required}$. |
| **EN_MINIMUMPRESSURE** | 120 | Read/Write | The minimum pressure, $H_{minimum}$ for PDD nodes |
| **EN_CRITICALPRESSURE** | 121 | Read/Write | The critical pressure, $H_{critical}$ for PDD nodes |
| **EN_EMITTERTYPE** | 122 | Read | Returns the type of emitter represented by this node:<br>0 Normal node<br>1 Standard emitter<br>2 Standard emitter with custom exponent<br>3 PDD demand node<br>4 PDD demand node with custom exponent |
| **EN_EMITTERSTATUS** | 123 | Read | Returns the operational status of an emitter:<br>0 Emitter Closed<br>1 Emitter Active<br>2 Emitter Open |
| **EN_EMITTEREXPONENT** | 124 | Read/Write | The emitter exponent – if not specified for a node then the global emitter exponent defined in the [OPTIONS] section of the input file is used instead. |

**Table 11: Additional options for ENgetlinkvalue/ENsetlinkvalue**

| C/C++ Constant name | Value | Read/Write | Purpose |
|---|---|---|---|
| **EN_VERTEXCOUNT** | 100 | Read | Returns the number of vertices in a given link. |
| **EN_VERTEXX** | 101 | Read | Returns the X coordinate of a given vertex – pass the required vertex index into the *ENgetlinkvalue* function. |
| **EN_VERTEXY** | 102 | Read | Returns the Y coordinate of a given vertex – pass the required vertex index into the *ENgetlinkvalue* function. |

## Conclusions

EPANETpdd implements a Pressure Driven Demand hydraulic simulation that offers considerable flexibility in defining the network configuration as well as being straightforward in use.

## References


ANG, W.K. & JOWITT, P.W. 2006. Solution for Water Distribution Systems under Pressure-Deficient Conditions. *Journal of Water Resources Planning and Management - ASCE*, **132**(3) *pp175-182*.

CHEUNG, P.B., VAN ZYL, J.E. & REIS, L.F.R. 2005. Extension of EPANET for Pressure Driven Demand Modeling in Water Distribution System. In: SAVIĆ, D.A., WALTERS, G.A., KING, R. & KHU, S.T. (eds.) *Proceedings of the Eighth International Conference on Computing and Control for the Water Industry (CCWI2005).* University of Exeter, UK. *Vol 1, pp311-316.*

DE MARINIS, G., GARGANO, R., KAPELAN, Z., MORLEY, M.S., SAVIĆ, D.A. & TRICARICO, C. 2007. The Influence of the Hydraulic Simulator in Water Distribution System Rehabilitation Analysis. In: ULANICKI, B., VAIRAVAMOORTHY, K., BUTLER, D., BOUNDS, P.L.M. & MEMON, F.A. (eds.) *Supplementary Proceedings of the Combined International Conference of Computing and Control for the Water Industry (CCWI2007) and Sustainable Urban Water Management (SUWM2007)* de Montford University, Leicester, UK. *pp7-14.*

FUJIWARA, O. & LI, J. 1998. Reliability Analysis of Water Distribution Networks in Consideration of Equity, Redistribution and Pressure-Dependent Demand. *Water Resources Research*, **34**(7) *pp1843-1850.*

GERMANOPOULOS, G. 1985. A technical note on the inclusion of pressure dependent and leakage terms in water supply network models, *Civil Engineering Systems*, **2**(3), *pp171-179.*

GUPTA, R. & BHAVE, P.R. 1996. Comparison of Methods for Predicting Deficient-Network Performance. *Journal of Water Resources Planning and Management - ASCE*, **122**(3), *pp214-217.*

KAPELAN, Z., SAVIĆ, D.A. & WALTERS, G.A. 2003. Robust Least Cost Design of Water Distribution System Using GAs. In: MAKSIMOVIĆ, C., BUTLER, D. & MEMON, F.A. (eds.) *Proceedings of the Seventh International Conference on Computing and Control for the Water Industry (CCWI2003).* A.A.Balkema, London, UK. *pp147-155.*

ROSSMAN, L.A. 2000. *EPANET 2 User's Manual.* United States Environmental Protection Agency, Cincinnati, U.S.A.

TRICARICO, C., GARGANO, R., KAPELAN, Z., SAVIĆ, D.A. & DE MARINIS, G. 2006. Economic Level of Reliability for the Rehabilitation of Hydraulic Networks, *Journal of Civil Engineering and Environmental Systems.* **23**(3) *pp191-207.*

TUCCIARELLI, T., CRIMINISI, A. & TERMINI, D. 1999. Leak Analysis in Pipeline Systems by Means of Optimal Valve Regulation. *Journal of Hydraulic Engineering*, **125**(3) *pp277-285.*

VAN ZYL, J.E., BORTHWICK, J. & HARDY, A. 2003. OOTEN: An Object-Oriented Programmer's Toolkit for EPANET. In: MAKSIMOVIĆ, C., BUTLER, D. & MEMON, F.A. (eds.) *Proceedings of the Seventh International Conference on Computing and Control for the Water Industry (CCWI2003).* Imperial College London, UK. *Supplementary Paper.*

WAGNER, J.M., SHAMIR, U. & MARKS, D.H. 1988. Water Distribution Reliability: Simulation Methods. *Journal of Water Resources Planning and Management - ASCE*, **114**(3) *pp276-294.*