Temas de análisis respecto a MGH

**Nombres de nudo y tramo**

Se requiere incluir en los archivos de la red además del identificador de nudo y de tramo un nombre en forma de un código para relacionar cada elemento con su correspondiente en la vida real. Para el modelo es necesario nombrar a los nudos con un identificador [id] de tipo *integer*, contando desde 0, porque se debe acceder a la posición de cada elemento en una matriz de 1 o 2 dimensiones. Con ese [id] relacionado con un [nombre] se puede leer el archivo de entrada y salida con un formato más apto para los humanos *(human readable*)

**Reposición de estado de presiones negativas**

Cuando existan presiones negativas en la red, en los nodos con carga negativa, el caudal asignado es incongruente con la presión negativa. Entonces el caudal debe ser menor, pero no se sabe cuánto. La propuesta de algoritmo es buscar el nudo con la presión más negativa, y reducir la demanda del nodo un valor ∆q, proporcional a la demanda qi (por ejemplo ∆q =qi\*0.1), de manera que el nuevo q sea: qi+1 = 0.9\*qi.

Con esa nueva demanda en el nudo se vuelve a correr la red. En esta nueva corrida necesariamente variarán las presiones en general de la red, deben ser mayores, por lo que uno o varios nudos que tenían carga negativa deben llegar a cargas positivas.

Si siguen existiendo presiones negativas, vuelve a seleccionar el nudo de la mínima presión y vuelve a ajustar el caudal demandado factorizándolo de la manera explicada anteriormente: qi+1 = 0.9\*qi.

Se prosigue en ese procedimiento de ajuste de demandas hasta que no haya nudos con presión negativa o hasta que el caudal de dichos nodos afectados sea cero.

Aquellos nudos que lleguen a demanda cero qi=0 serán consistentes con “desabastecimiento” espontáneo, es decir no causado por un cierre de tuberías por mantenimiento o falla.

En el proceso, el archivo de entrada va a sufrir una variación para que el modelo realice dicha reposición; por lo tanto, es necesario que en este proceso de reposición, el archivo de entrada y salida iniciales sean guardados intactos y se construya un nuevo archivo con las iteraciones de los nuevos datos de entrada y salidas.

Este proceso no debe ser automático, sino que debe ser solicitado por el usuario, ya que dependiendo de las necesidades puede ser requerido no hacerlo.

Se propone que se pueda ejecutar esta rutina de reposición por medio de un modo de ejecución:

# python3 mgh.py archivo\_red.mgh -r

-r modo de ejecución con rutina de reposición por presiones negativas

Este modo de ejecución arroja los resultados de la corrida normal, pero finalmente arroja un archivo adicional con las demandas y presiones reales máximas posibles y el estado de caudales en los tramos, que permiten tener una estimación de los niveles de desabastecimiento y/o reducción de la satisfacción de la demanda en cada nudo

**Modos de ejecución**

Algunos de los modos de ejecución propuestos para el programa ya han sido implementados, sin embargo, se requieren nuevos modos de ejecución y combinaciones de ellos. A continuación una lista de los modos propuestos:

|  |  |  |
| --- | --- | --- |
| **Modo** | **Uso** | **Estado** |
| -v | Detallado: imprime todos los datos de iteraciones, matrices y resultados | OK |
| -q | Silencioso: solo muestra vectores de Q, H y P | OK |
| -n | Normal: imprime tabla de datos y resultados de nudos y tramos | OK |
| -i | Interactivo: permite pasar los datos de la red por *terminal input* | OK |
| -r | Reposición: Reponerse de presiones negativas en nudos | WIP |
| -j | JSON: salida de datos en forma de archivo json. Función de API | OK |
| -x | Extendido: ejecución en tiempo extendido, en tractos de 1 hora | Espera |

**Separar archivos de entrada de datos**

Los datos de entrada del modelo de una red se compones de al menos 2 partes principales:

* Datos topológicos de la red
* Parámetros hidráulicos que serán usados en la simulación

Es conveniente separar ambos componentes en 2 archivos independientes.

Uno contendrá los parámetros de estado de la red tales como niveles de tanque, demandas, con sus respectivas fecha y hora y junto con los patrones o curvas de demanda, en un archivo de entrada independiente, de modo que pueda ser construido con datos de un SCADA, digitados o por medio de un módulo API, para hacer corridas con diferentes escenarios en una red existente. Los datos que se incluirían en este archivo serían:

|  |  |
| --- | --- |
| **Tipo de elemento** | **Parámetros a incluir en archivo de entrada** |
| Nudos de carga fija | [id] [carga, fecha, hora] |
| Nudos de demanda | [id] [demanda, fecha, hora] [curva de demanda] |
| Nudos tipo naciente | [id] [producción, fecha, hora] [curva de producción] |
| Nudos tipo emisor | [id] [ecuación] [estado, fecha hora] |
| Tramos simples | [id] [estado, fecha, hora] |
| Tramos tipo válvula | [id] [consigna, fecha, hora] |
| Tramos tipo bomba o booster | [id] [estado, fecha,hora] [opciones, fecha, hora] |

El otro archivo contendrá los valores intrínsecos de la topología, tales como:

* Nudos: identificador, tipo, cota topográfica y nombre
* Tramos: desde, hasta, L, D, ks, kL, tipo

Actualmente el tipo de tramo se escoge en el campo de [Estado] porque inicialmente se pensó en que una tubería podía estar abierta o cerrada, podía tener estado=TA (tubería abierta) y estado=TC (tubería cerrada). Sin embargo al aparecer otros tipos de tubería, ya el [estado] se debe cambiar por [tipo] y en caso de requerir que la tubería esté abierta o cerrada se debe indicar en las opciones, tal y como se propone .

**Siendo congruentes con la separación de archivos de entrada en topología y parámetros**

**Tubería en canal abierto**

A

**Vaciado de tanques**

El nivel de tanque puede ser negativo, e ir más abajo del nivel de piso, descendiendo por la tubería a medida que se va vaciando la red, con el fin de simular el hecho de que la tubería se vacía

**Tubería cerrada**

TC

**Válvula de retención (check)**

CK

**Ejecución en tiempo extendido**

**Curvas de demanda**

**Emisores**

Hidrante

Fuga

**Activar/Desactivar Bomba**

Debe existir la capacidad de que durante la ejecución de un modelo en tiempo extendido, se pueda simular el encendido y apagado de una bomba. Esto es posible mediante la asignación del estado del tramo intercambiándolo de bomba a check (BO -> CB, con CB en lugar de CK para diferenciarlo de un Check normal, que puede volver a funcionar como bomba en dado caso que vuelva a arrancar., o talvez más ordenado, incluir en la lista de opciones de la bomba un estado en la forma de un bit 0=apagado, 1=Encendido. Este bit se concatenaría con los datos de la curva de la bomba:

* "opciones": "-3125.0 187.5 77.5 1" => α=-3125, β=187.5, γ=77.5 ON
* "opciones": "-3125.0 187.5 77.5 0" => α=-3125, β=187.5, γ=77.5 OFF