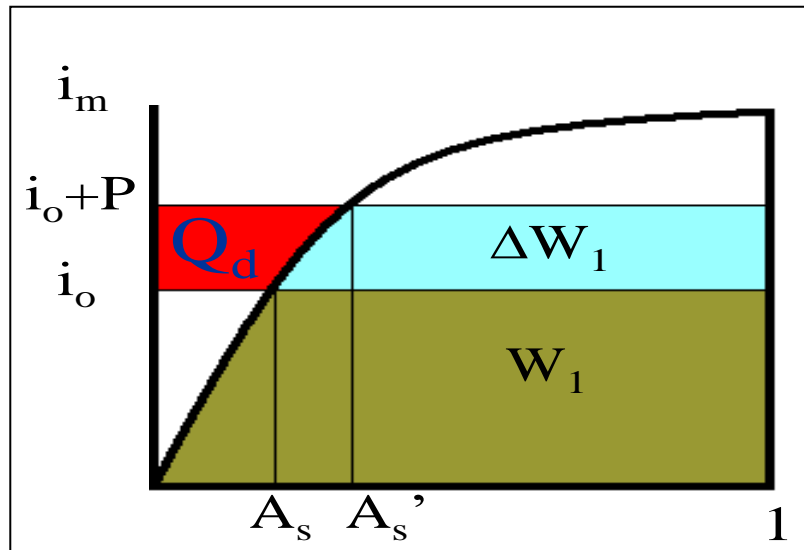


**Coordinación de Hidrológica
Subcoordinación de Hidrometeorología
Subcoordinación de Hidrología y Mecánica de Ríos**

Proyecto TH1004.1



**Calibración y validación de un modelo acoplado
lluvia-escorrentía MM5-VIC**

Informe Final

Participantes:

M. en C. Olivia Rodríguez López

Dr. René Lobato Sánchez

M. I. Esteban Pardo García

M. I. Jorge Humberto Salgado
Rabadán

Dr. Francisco Muñoz Arriola

L. I. José Guadalupe Rosario de la Cruz

Diciembre de 2010

1.1.a.1.1.1 Contenido

2	INTRODUCCIÓN	3
3	ANTECEDENTES	4
1.	ACTUALIZACIÓN DE BASE DE DATOS	6
1.1	Cuencas	6
1.2	Hidrometría	7
1.3	Meteorología	9
1.4	Infraestructura	11
a)	Presas.....	11
b)	Centrales hidroeléctricas	12
1.5	Determinación de los escurrimientos naturalizados.....	13
a)	Series históricas de las centrales hidroeléctricas	13
b)	Tránsito de avenidas.....	14
2.	SIMULACIONES CON PERIODOS DE TIEMPO MÁS LARGOS PARA INTEGRARLO AL MODELO ACOPLADO	16
3.	CALIBRACIÓN DEL MODELO HIDROLÓGICO (VIC)	22
3.1	Capacitación	22
3.2	Modelo hidrológico	23
3.2.1	Archivos de entrada de VIC	24
a)	Forzantes meteorológicos	24
b)	Parámetros de suelo	25
c)	Parámetros de vegetación.....	26
3.2.2	Simulaciones con VIC.....	27
3.2.3	Archivos de salida de VIC.....	28
3.2.4	Procesamientos con el modelo ROUT	29
3.2.5	Archivos de entrada de ROUT.....	30
3.2.6	Correr ROUT	32
3.2.7	Archivos de salida de ROUT	33
3.2.8	Scripts para sintetizar archivos	33
a)	Activate_basin.pl.....	34
b)	Agg_space.pl.....	34
c)	Agg_time.pl	35
3.2.9	Calibración del modelo VIC	35

a)	Definición de parámetros de calibración	35
b)	Grupos de parámetros	36
c)	Simulación de VIC con un primer juego de parámetros	36
3.2.10	Optimización de parámetros de VIC	38
a)	Modelo de optimización PEST	38
b)	Archivos de entrada y salida de PEST	38
c)	Archivo template: in.tpl	39
d)	Archivo de parámetros (in.par)	39
e)	Archivo de observaciones (Mesure.dat)	40
f)	Archivo de instrucciones (Mesure.ins)	41
g)	Verificación de los archivos con PEST	41
h)	Interfaz para crear los archivos de PEST	43
i)	Archivo de control PEST	44
j)	Acoplamiento VIC-ROUT con PEST	52
k)	Optimización de VIC-ROUT con PEST	57
l)	Pasos para realizar una corrida de PEST	57
m)	Salidas de PEST y resultados	61
4.	COMPARACIÓN DE RESULTADOS Y VALIDACIÓN DEL MODELO	64
4.1	Series obtenidas por antitránsito	iError! Marcador no definido.
n)	Funcionamiento de vaso	¡Error! Marcador no definido.
4.2	Simulación VIC con periodo de XX años	iError! Marcador no definido.
4.3	Resultados y validación	iError! Marcador no definido.
5.	MODELOS ACOPLADOS	65
6.	CONCLUSIONES Y RECOMENDACIONES	67
6.1	Conclusiones	68
6.2	Recomendaciones	69
4	REFERENCIAS	70
5	ANEXOS	71

2 Introducción

El presente informe corresponde a las actividades del proyecto TH1004.1 denominado **"Calibración y validación de un modelo acoplado lluvia-escurrimiento MM5-VIC"**, es un proyecto desarrollado en el marco de la investigación aplicada dentro del campo de estudio de las ciencias exactas y naturales, el cual fue desarrollado con recursos fiscales. Este proyecto es la continuación del proyecto **"Acoplamiento de un modelo lluvia-escurrimiento para una cuenca piloto en México"** realizado durante el año 2009.

El objetivo del proyecto planteado en el anexo técnico es: Probar la eficacia del modelo MM5-VIC para su posible implementación operativa con dos eventos hidrometeorológicos extremos ocurridos en la cuenca del alto Grijalva.

Para el cumplimiento de los objetivos la problemática se aborda con base en la siguiente estrategia.

El modelo hidrológico se calibró comparando los resultados con escurrimientos vírgenes medidos a la salida de la cuencas utilizando un periodo de 4 años comprendido dentro del periodo comprendido del 01 de enero de 1960 al 31 de diciembre de 1963, periodo en el cual no se iniciaba aún la construcción de ninguna obra hidráulica en la zona, el primer almacenamiento inició su llenado en el año de 1966.

La validación del modelo hidrológico se realizó con la reproducción de los hidrogramas de los años posteriores a ese periodo, antes del año de inicio de la construcción de la infraestructura hidráulica en la zona (1966) de esta manera se determinó la capacidad del modelo representar los "picos" en el hidrograma simulado.

Finalmente se realizan simulaciones con los modelos acoplados MM5-VIC y se determina su eficacia en el pronóstico de avenidas a partir de un pronóstico meteorológico.

3 Antecedentes

En la cuenca del río Grijalva se encuentra un sistema de generación hidroeléctrica que comprende actualmente la operación de cuatro aprovechamientos en cascada. En la cuenca alta del Grijalva se encuentra la presa Venustiano Carranza (La Angostura), aguas abajo se encuentra la presa Manuel Moreno Torres (Chicoasén), a continuación se localiza la presa Netzahualcóyotl (Malpaso), por último se encuentra la presa Ángel Albino Corzo (Peñitas).

Aunque la cuenca del río Grijalva está controlada tiene un aporte de caudal importante en las llanuras inundables. Asimismo, en esta cuenca es donde se tiene mayor número de estaciones hidrométricas y climatológicas con periodos de registros importante. Las aportaciones de esta cuenca, la de Ríos de la Sierra y la del río Usumacinta, impactan en las afectaciones por inundación en la parte baja de la cuenca. Por tal motivo, se inició la modelación hidrometeorológica de la cuenca utilizando el modelo acoplado MM5-VIC.

El modelo MM5 es un modelo atmosférico de mesoescala, de escala regional que utiliza un sistema de ecuaciones primitivas con configuración no hidrostática (*Grell et al 1993; Gill 1992*). Permite el uso de dominios múltiples y anidados, está desarrollado en coordenadas sigma que sigue la superficie terrestre y resuelve ecuaciones de diferencias finitas con un esquema de paso de tiempo *Chen y Dudhia (2001)*. El MM5 incorpora las modificaciones del modelo de superficie de la Universidad del estado de Oregón, para capturar las variaciones de los flujos y temperatura superficial, así como la evolución estacional de la evaporación y humedad del suelo (*Chen et al 1996, Chen and Mitchell 1999*).

El sistema acoplado MM5-VIC fue desarrollado en la Universidad de Washington y permite que VIC establezca en primera instancia las condiciones iniciales sujetas a parámetros como vegetación humedad de suelo y nieve, generando los flujos para momentum, calor y humedad correspondientes a las condiciones iniciales en la frontera inferior (cercana a la superficie) de MM5. El sistema produce una gama de variables de estado hidrometeorológicas que van desde la precipitación, evapotranspiración, hasta la humedad de suelo, flujo base y escurrimiento.

El IMTA de manera conjunta con la Universidad de Washington realizaron el proyecto denominado **"Acoplamiento de un modelo lluvia-escurrimiento para una cuenca piloto en México"** en el año 2009, en el que se implementó un sistema acoplado con los modelos numérico atmosférico de mesoescala MM5 e hidrológico semidistribuido VIC para fines de pronóstico a corto y mediano plazo, aplicado a la cuenca del río Grijalva.

En esta primera etapa se realizó la delimitación del área de estudio con base en la información hidrometeorológica disponible. Asimismo, se realizaron actividades relativas a la comunicación de archivos de datos entre ambos modelos en donde se conjugan las capacidades de cada uno de ellos.

Con el desarrollo de estos modelos se pretende generar una herramienta que nos permita predecir los flujos superficiales en un horizonte de tiempo de horas, días y semanas; y proporcione soporte a la toma de decisiones en la operación de la infraestructura que permita sortear el embate de los fenómenos extremos en la parte baja de la cuenca.

1. Actualización de base de datos

1.1 Cuencas

Se delimitó el área de estudio con base en la información hidrológica disponible. La cuenca se delimitó inicialmente hasta el punto de bifurcación de los ríos Samaria y Carrizal (Figura 1), en donde se tiene tres estaciones hidrométricas con registros que se describen en la tabla 1.

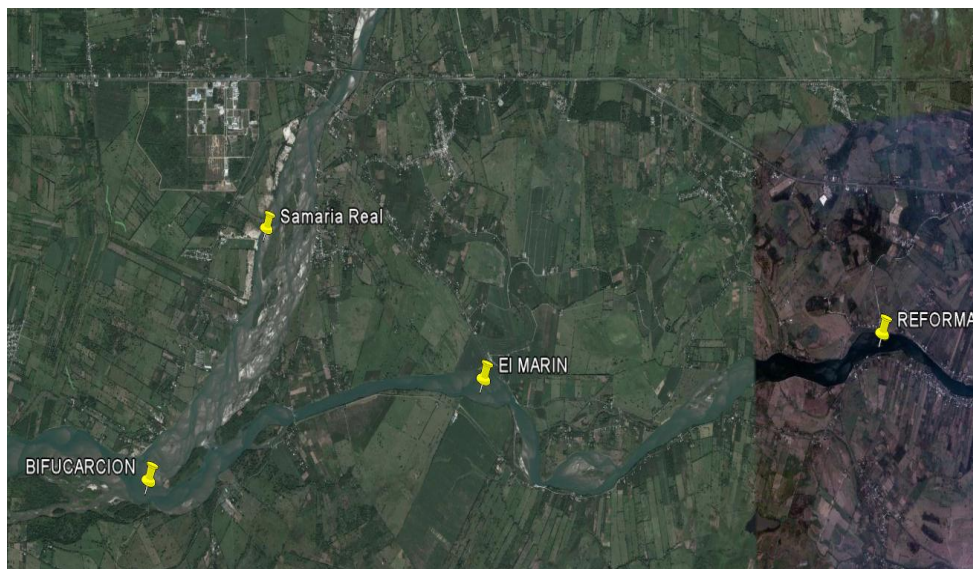


Figura 1. Delimitación de la cuenca del río Grijalva

Nombre	Clave	Coordenadas	
		Latitud N	Longitud W
Samaria Real	30005	17.9792	93.2861
El Marín	30012	17.9606	93.2500
Reforma	30051	17.9661	93.1833

Tabla 1. Estaciones hidrométricas en el sitio considerado

Debido a la incompatibilidad de periodos registrados que se tienen en estas estaciones y a la imprecisión que implicaría obtener un hidrograma por diferencias, se recorrió el punto de salida de la cuenca hasta el punto donde se encuentra la presa Ángel Albino Corzo (Peñitas), como puede observarse en la figura 2.

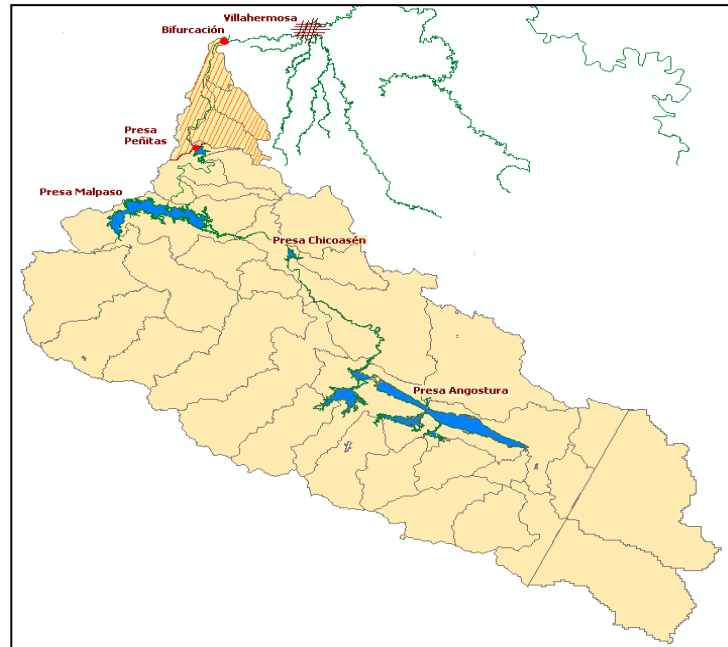


Figura 2. Delimitación del río Grijalva hasta la presa Peñitas

Se solicitó a la Comisión Federal de Electricidad (CFE) registros hidrométricos en este punto obteniendo información de la estación hidrométrica Peñitas con clave 30015 y registros comprendidos en el periodo de 1950 a 2001.

Asimismo, se completó la información cartográfica de la parte de la cuenca que pertenece a Guatemala.

1.2 Hidrometría

Se identificaron 69 estaciones hidrométricas en la cuenca completa del río Grijalva hasta su desembocadura. Se utilizó el BANDAS (Ref. 1.7) para obtener los registros de cada estación. (Figura 3).

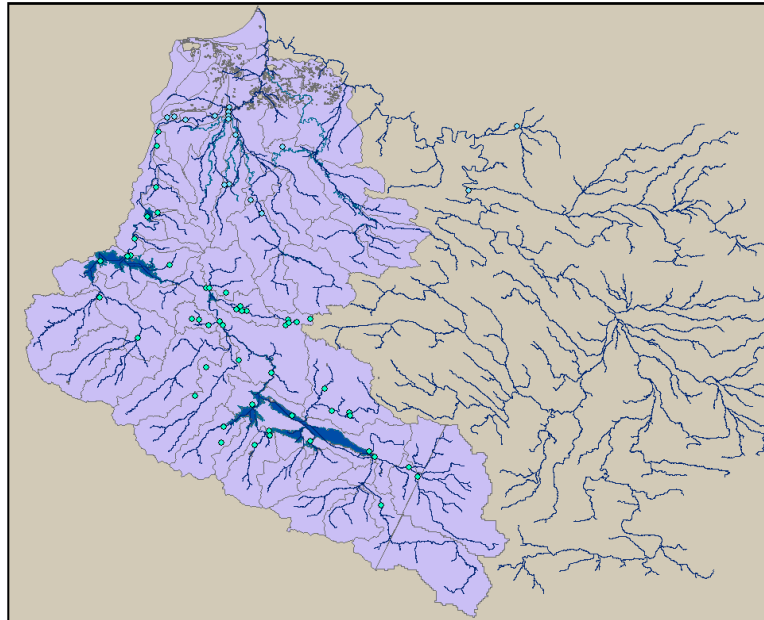


Figura 3. Estaciones hidrométricas en la cuenca del río Grijalva hasta su desembocadura

Se extrajeron los registros exportándolos a una hoja de cálculo con el fin de ordenar los registros por año y calcular los porcentajes de información. Se cuenta con registros desde 1950 hasta el 2001.

Se observó que la información tiene pocos años de registros completos, en el Anexo 2 se pueden consultar los registros y en la figura 4 se muestra la densidad de datos hidrométricos en la cuenca, en donde se aprecia que se tiene un máximo en la medición en el año 1965, a partir de éste decae la medición hidrométrica.

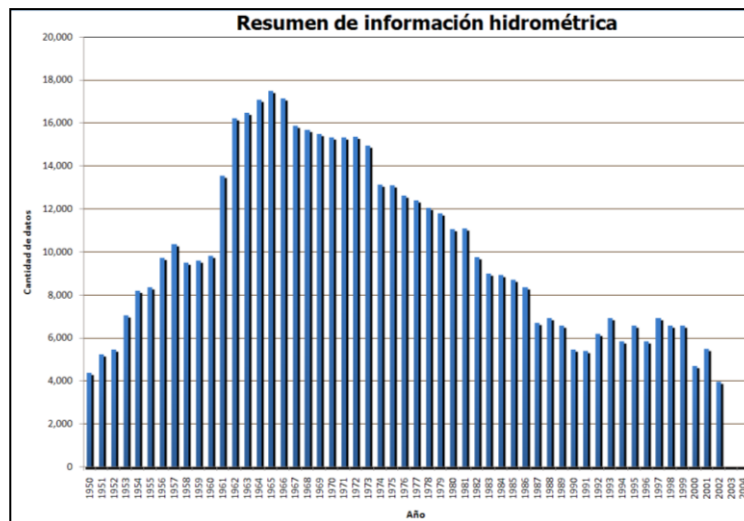


Figura 4. Densidad de registros en las estaciones hidrométricas

En la cuenca delimitada hasta la bifurcación de los ríos Samaria y Carrizal, se localizan ocho estaciones hidrométricas, los registros pueden consultarse en el Anexo 2.

Con base en esta información se realizó una segunda delimitación de la cuenca, hasta la presa Ángel Albino Corzo (Peñitas), debido a que existe una estación con registros completos de un periodo de 51 años (1950-2001), como puede apreciarse en la tabla 2.

Nombre	Clave	Río	Periodo de registros
Samaria Real	30005	Samaria	1951-1986;1993-2002
El Marín	30012	Carrizal (Grijalva)	1951-1953
Las Peñitas	30015	Mezcalapa	1950-2001
Reforma	30051	Carrizal (Grijalva)	1954-1973; 1975-1985
Malpaso I	30064	Grijalva	1959-1962
Malpaso II	30076	Grijalva	1962-1982
Grijalva	30098	Grijalva	1965-2000
La Angostura	30128	Grijalva	1963-1977

Tabla 2 Estaciones hidrométricas

Se utilizaron los registros hidrométricos de la estación Las Peñitas para la calibración del modelo. Se tomó un periodo de cinco años (1960-1964), antes del inicio de la construcción de la infraestructura hidráulica que vendría a modificar el régimen de escurrimientos naturales, que inició en 1966 con el llenado de la presa Netzahualcóyotl (Malpaso), la primera presa del construida en la cuenca.

1.3 Meteorología

En la zona de estudio se identificaron 151 estaciones climatológicas, se obtuvo la precipitación, temperatura máxima y temperatura mínima. En Anexo 3 se pueden consultar los registros de las estaciones seleccionadas.

La información se extrajo del Eric III (Ref. 1.8), cuya información se exportó a archivos con extensión XLS para su análisis por medio de una macro, la cual ordena los registros de cada estación por año. La Figura 5 muestra la distribución espacial.



Figura 5. Estaciones meteorológicas

En el Anexo 3 se pueden consultar los registros de precipitación y en la figura 6 se muestra la densidad de datos meteorológicos de las estaciones localizadas, en donde se aprecia una mayor densidad de registros en el periodo de 1972 a 1999.

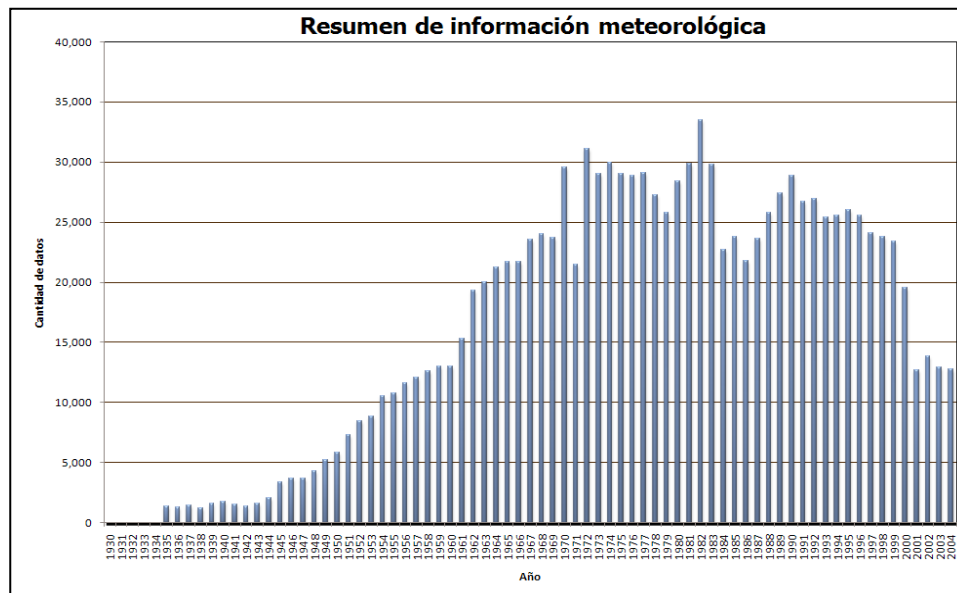


Figura 6. Densidad de registros en las estaciones meteorológicas

1.4 Infraestructura

La cuenca del río Grijalva tiene un sistema de generación hidroeléctrica que comprende actualmente la operación de cuatro aprovechamientos en cascada. En la cuenca alta del Grijalva se encuentra la presa Venustiano Carranza que cuenta con la central hidroeléctrica La Angostura, aguas abajo se encuentra la presa Manuel Moreno Torres equipada con la central Chicoasén, a continuación se localiza la presa Netzahuacoyotl con su central Malpaso, por último se encuentra la presa Ángel Albino Corzo con la central Peñitas. La Figura 7 muestra un diagrama conceptual del sistema.

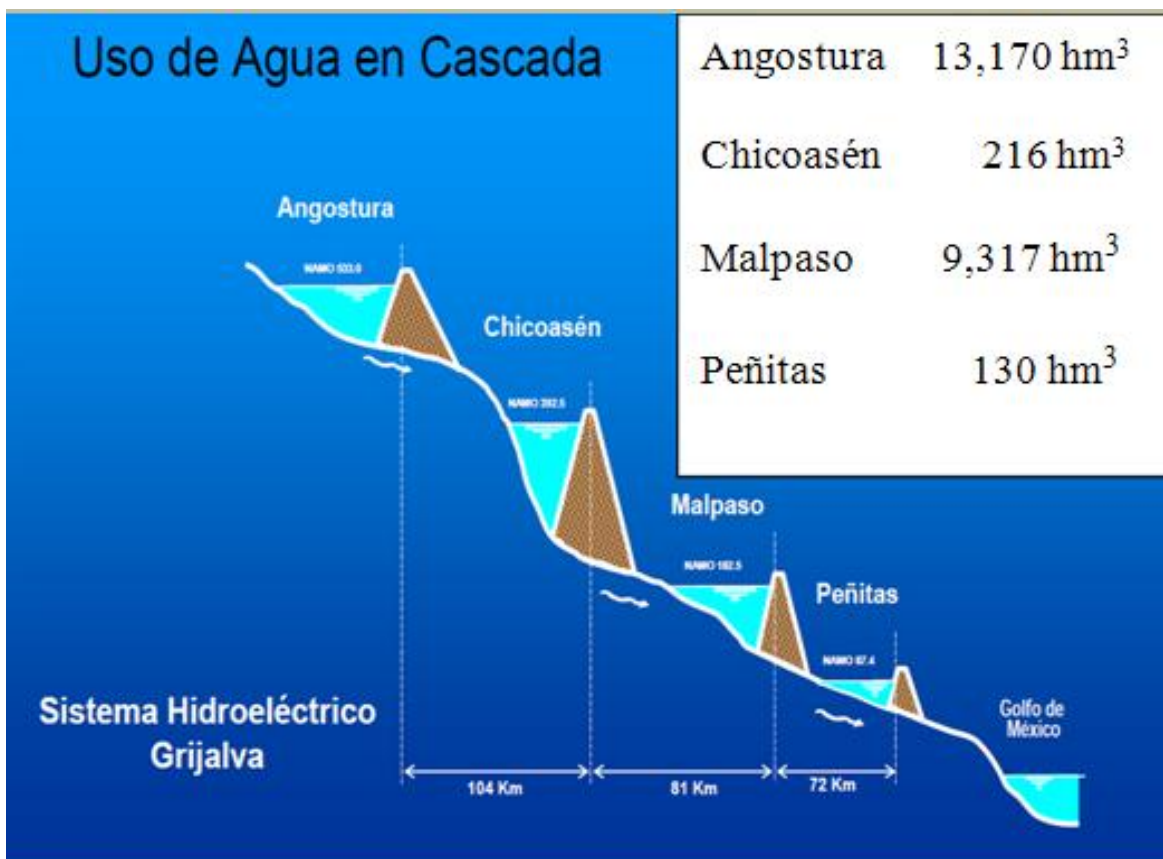


Figura 7. Centrales hidroeléctricas

a) Presas

En la cuenca de río Grijalva se localizan cuatro aprovechamientos (Tabla 3), con base en el programa de *Google Earth 3.0* se determinó su ubicación (Ref. 1.9).

Embalse	Latitud	Longitud
Presa Venustiano Carranza	92° 45' 00"	16° 28' 00"
Presa Manuel Moreno Torres	93° 06' 00"	16° 59' 00"
Presa Netzahualcóyotl	93° 35' 54"	17° 10' 41"
Presa Ángel Albino Corzo	93° 27' 35"	17° 26' 52"

Tabla 3 Localización de embalses

Las características generales con las que cuenta cada embalse fueron proporcionadas por la CFE (Ref. 1.10), en la Tabla 4 se muestran los datos obtenidos.

Característica	Venustiano Carranza	Manuel Moreno Torres	Netzahualcóyotl	Ángel Albino Corzo
NAME (msnm)	539.50	395.00	188.00	95.50
NAMO (msnm)	533.00	392.50	182.50	87.40
NAMINO (msnm)	500.00	380.00	144.00	85.00
Capacidad del vertedor (m ³ /s)	519.60	373.00	167.64	76.50
Volumen útil (hm ³)	13,170.00	216.00	9,317.39	130.00

Tabla 4 Características de embalses existentes

b) Centrales hidroeléctricas

Las centrales hidroeléctricas con que cuenta el sistema se localizaron a través del programa *Google Earth 3.0* (Ref. 1.9). En la Tabla 5 se enumeran las centrales hidroeléctricas y su localización.

Central Hidroeléctrica	Latitud	Longitud
C. H. Angostura	92° 45' 00"	16° 28' 00"
C. H. Chicoasén	93° 06' 00"	16° 59' 00"
C. H. Malpaso	93° 35' 54"	17° 10' 41"
C. H. Peñitas	93° 27' 35"	17° 26' 52"

Tabla 5 Localización de las centrales hidroeléctricas existentes

Los datos de las características generales de las centrales hidroeléctricas (Tabla 6), fueron proporcionados por la CFE (Ref. 1.12 y 1.15).

Característica	Venustiano Carranza	Manuel Moreno Torres	Netzahualcóyotl	Ángel Albino Corzo
Inicio de operación	1978	1980	1966	1987
Capacidad instalada (kW.)	900	1,500	1,080	420
Potencia de unidades (Mw.)	180	300	180	105
Cons.Esp. m ³ /kWh	4.36	2.2001	5.05	11.85
Caudal de diseño (m ³ /s)	1,090.00	1,493.60	1,440.00	1,440.00
Nivel medio de desfogue (msnm)	421.50	203.00	84.50	53.00

Tabla 6 Características de las centrales hidroeléctricas

1.5 Determinación de los escurrimientos naturalizados

Los requerimientos de información para la calibración del modelo hidrológico semidistribuido son los escurrimientos naturalizados, los cuales representan el hidrograma de salida de la cuenca con el cual se comparan los resultados obtenidos de la simulación.

a) Series históricas de las centrales hidroeléctricas

Se integró información sobre la operación de las presas y centrales proporcionada por la CFE (Ref. 1.16). Se obtuvieron series horarias de generación, volumen turbinado y nivel de embalse. En la Tabla 7 se muestran los periodos que se obtuvieron.

Concepto	Venustiano Carranza	Manuel Moreno Torres	Netzahualcóyotl	Ángel Albino Corzo
Series de generación (kwh)	Diaria: 2003-2005 Mensual: 1967-2004 Anual: 1990-2004	Diaria: 2003-2005 Mensual: 1967-2004 Anual: 1990-2004	Diaria: 2003-2005 Mensual: 1967-2004 Anual: 1990-2004	Diaria: 2003-2005 Mensual: 1967-2004 Anual: 1990-2004
Series de volumen de agua turbinada (hm ³)	Diaria: 2003-2005 Mensual: 1967-2004 Anual: 1990-2004	Diaria: 2003-2005 Mensual: 1967-2004 Anual: 1990-2004	Diaria: 2003-2005 Mensual: 1967-2004 Anual: 1990-2004	Diaria: 2003-2005 Mensual: 1967-2004 Anual: 1990-2004
Series de nivel de embalse (msnm)	Diaria: 2003-2005	Diaria: 2003-2005	Diaria: 2003-2005	Diaria: 2003-2005

Tabla 7 Periodos obtenidos de las series históricas

Con el fin de validar los datos obtenidos de las series, se organizaron en hojas de cálculo, para identificar inconsistencias tales como discontinuidad en las fechas, valores fuera de rango, registros faltantes, entre otras (Anexos 2, 3, 4 y 5).

Una vez validada la información proporcionada, se elaboró un archivo Excel llamado *Series históricas* en donde se concentraron las series de funcionamiento horario de tres años (2006-2008).

b) Tránsito de avenidas

Para conocer la evolución del hidrograma a medida que transcurre a lo largo del cauce y determinar el hidrograma a la salida de la cuenca, se aplicó el método de Muskingum para el tránsito de avenidas.

$$S = K [x I + (1-x) O]$$

Donde:

S : Almacenamiento en el tramo considerado de cauce

K : Parámetro de almacenamiento

x : Factor de ponderación entradas-salidas

I : Caudal de entrada en el tramo

O : Caudal de salida en el tramo

La aportación por cuenca propia en los almacenamientos se determinó aplicando el tránsito inverso en cada uno de ellos, de tal manera que se obtuvo el hidrograma a la entrada de cada uno de los embalses y por diferencia de éstas ya las extracciones de la presa de aguas arriba se obtiene la aportación por cuenca propia (Fig. 8).

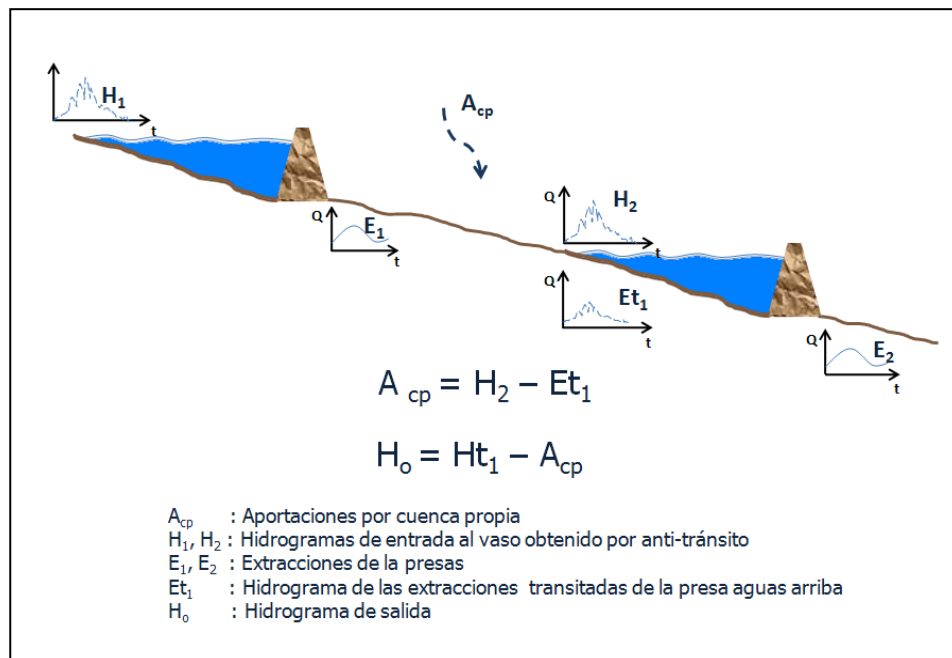


Figura 8. Diagrama del tránsito de hidrogramas

De esta manera se obtuvo el hidrograma naturalizado a la salida de la cuenca, el cual se compara con el hidrograma obtenido como resultado del modelo hidrológico. En la

figura 9 se muestra el hidrograma naturalizado obtenido con el hidrograma transitado, para el periodo 2006-2008.

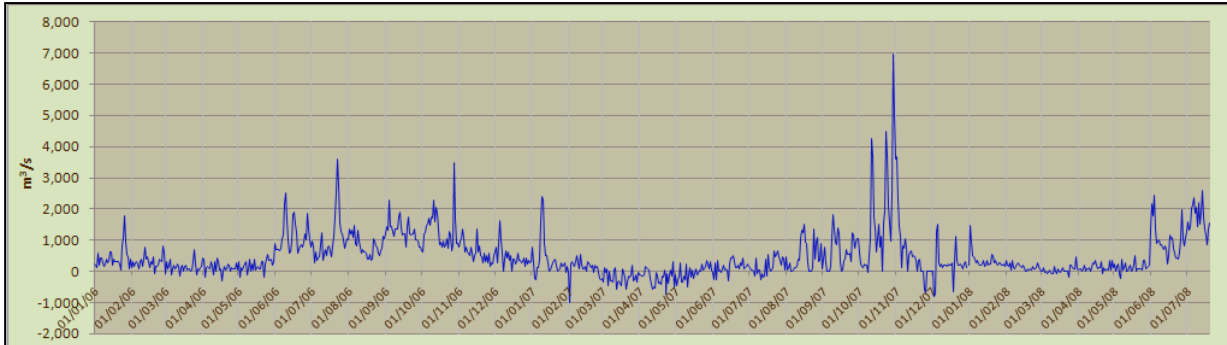


Figura 9. *Hidrograma transitado*

Los cálculos se realizaron en archivos con formato .xls que pueden consultarse en el anexo 4.

2. Simulaciones con periodos de tiempo más largos para integrarlo al modelo acoplado

Para el presente estudio se realizaron corridas con el modelo meteorológico en primera instancia para después usar esta información como condiciones de entrada para correr el modelo acoplado para periodos más largos, es decir, se corrió el MM5 para los eventos de interés. A continuación se describe lo que se hizo:

Descripción del Mesoescala MM5

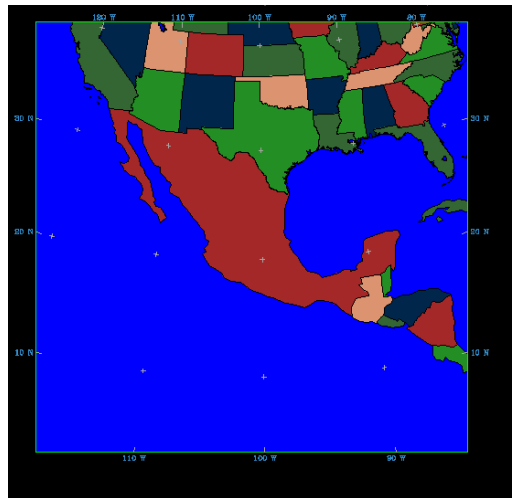
El modelo MM5 es un modelo de escala regional que utiliza un sistema de ecuaciones primitivas con configuración no hidrostática (Grell et al., 1993; Gill, 1992), que permite el uso de dominios múltiples y anidados, está desarrollado en coordenadas sigma que siguen la superficie terrestre permitiendo así modelar en regiones con topografía compleja, además que resuelve las ecuaciones de diferencias finitas con un esquema de paso de tiempo, usando el operador "leapfrog", permite 23 niveles en la vertical y tiene las siguientes opciones de proyección geográfica: Estereográfica polar, Lambert conformal y Mercator (Cox et al., 1997). Cuenta con opciones adicionales para las parametrizaciones de convección, capa límite planetaria, radiación, procesos de superficie y condensación de gran escala (Silva, 2000).

El modelo MM5 se divide en los siguientes módulos:

- TERRAIN
- REGRID
- INTERPF
- MM5

1.- TERRAIN

Permite determinar la configuración del modelo de mesoescala: donde toma lugar la malla, el tamaño de la malla, que resolución de datos usar para generar la elevación del terreno, las categorías de uso de suelo y otra serie de datos como tipo de suelo, fracción de vegetación. Interpola horizontalmente la elevación del terreno y los datos de tipo de suelo para los dominios de mesoescala elegidos. Estos archivos de terreno serán utilizados en el programa REGRID. El tipo de suelo se separa en 13 categorías para cada punto de malla. Estos 13 tipos de suelo determinan el albedo, rugosidad, emisividad, capacidad de calentamiento del suelo y disponibilidad de humedad. En general este programa hace una configuración del dominio y crea los campos terrestres (tutorial MM5, <http://www.mmm.ucar.edu/mm5/documents/tutorial-v3-notes.html>).



Área de estudio

Nuestra área de estudio:

- Latitud 2.43 a 42.2464
- Longitud -127.97 a -74.0252
- Resolución 15 Kilómetros
- Periodo 04 al 29 de septiembre de 1998
25 de octubre al 20 de noviembre de 2007
- Proyección Lambert-conformal
- Salida de datos Cada 6 horas
- Tiempo de ejecución 88 horas

2. BAJADA DE DATOS

Los datos de reanálisis que se utilizaron para la realización de la corrida, fueron bajados automáticamente con el código del programa llamado bajadatos.sh, de acuerdo a la fecha del caso de estudio:

```
#!/bin/csh -f
set opts = "-N"
set passwd = "xxxxxxdx"
wget $cert_opt -O /dev/null --save-cookies auth.dss_ucar_edu --postdata="
email=xxxxxx@yahoo.com&passwd=$passwd&action=login" https://dss.ucar.edu/cgi-
bin/login
wget $cert_opt $opts --load-cookies auth.dss_ucar_edu
http://dss.ucar.edu/datazone/dsszone/ds090.0/pgbf00_grb2d/1998/A20203-
9808.pgb.f00
wget $cert_opt $opts --load-cookies auth.dss_ucar_edu
http://dss.ucar.edu/datazone/dsszone/ds090.0/pgbf00_grb2d/1998/A20206-9808.grb2d
wget $cert_opt $opts --load-cookies auth.dss_ucar_edu
```

```
http://dss.ucar.edu/datazone/dsszone/ds090.0/pgbf00_grb2d/1998/A20223-9809.pgb.f00  
wget $cert_opt $opts --load-cookies auth.dss_ucar_edu  
http://dss.ucar.edu/datazone/dsszone/ds090.0/pgbf00_grb2d/1998/A20226-9809.grb2d  
rm auth.dss_ucar_edu
```

3.-REGRID

Se crean los campos meteorológicos en la primera malla previamente determinada en el programa TERRAIN, aquí se selecciona el período del tiempo para cada caso, REGRID tiene dos programas secuenciales: *pregrid* y *regridder*, el programa *pregrid* contiene datos de nivel de presión de los campos meteorológicos, obtenidos de fuentes como el NCAR y NCEP, y pone los datos en un formato intermedio, el programa *regridder* toma los datos en formato intermedio y los archivos de salida del TERRAIN creando la primera aproximación. Para los programas subsecuentes. Aquí se interpolan horizontalmente los datos del análisis del punto de malla de los análisis globales, a la malla del modelo de mesoescala. Estos datos son escritos sobre una superficie de presión para luego ser interpolados por el programa INTERPF.

4.-INTERPF

Este programa toma el nivel de presión de los campos meteorológicos producidos por el REGRID y las definiciones del usuario de los niveles sigma e interpola los datos del nivel de presión a niveles sigma. Esto exige interpolar los datos sobre la malla vertical y horizontal de superficies de presión a las superficies de sigma. Esta generación de datos es necesaria para que el modelo reciba los datos para las condiciones iniciales y de frontera.

5.-MM5 (Multiprocesamiento)

Este es el modulo que realiza la corrida, el cual recibe como parámetros de entrada:

Del módulo de **INTERPF**: *MMINPUT_DOMAIN1*, *BDYOUT_DOMAIN1*, and *LOWBDY_DOMAIN1*.

Del módulo de **TERRAIN**: *TERRAIN_DOMAIN1*

En esta sección se describe cómo se logra la corrida del MM5 en multiprocesamiento:

1. El compilador que se utilizó fue el LAN de INTEL, el cual fue configurado con Fortran y C de Intel.
2. La configuración quedó de la siguiente manera:

LAM/MPI: 7.1.4

Prefix: /usr/local/lam_intel

Architecture: x86_64-unknown-linux-gnu
Configured by: modelo
Configured on: Tue Sep 30 08:09:25 CDT 2008
Configure host: vortex.imta.mx
Memory manager: ptmalloc2
C bindings: yes
C++ bindings: yes
Fortran bindings: yes
C compiler: icc
C++ compiler: icpc
Fortran compiler: ifort
Fortran symbols: underscore
C profiling: yes
C++ profiling: yes
Fortran profiling: yes
C++ exceptions: no
Thread support: yes
ROMIO support: yes
IMPI support: no
Debug support: no
Purify clean: no
SSI boot: globus (API v1.1, Module v0.6)
SSI boot: rsh (API v1.1, Module v1.1)
SSI boot: slurm (API v1.1, Module v1.0)
SSI coll: lam_basic (API v1.1, Module v7.1)
SSI coll: shmemp (API v1.1, Module v1.0)
SSI coll: smp (API v1.1, Module v1.2)
SSI rpi: crtcp (API v1.1, Module v1.1)
SSI rpi: lamd (API v1.0, Module v7.1)
SSI rpi: sysv (API v1.0, Module v7.1)
SSI rpi: tcp (API v1.0, Module v7.1)
SSI rpi: usysv (API v1.0, Module v7.1)
SSI cr: self (API v1.0, Module v1.0)

3. Modificar el archivo configure.user realizamos dos modificaciones a las secciones de Multiprocesamiento y opciones de compilación de acuerdo al compilador.

SECCION MULTIPROCESAMIENTO

MPP options

#

For general information and updated "helpdesk" information see

<http://www.mmm.ucar.edu/mm5/mpp>

<http://www.mmm.ucar.edu/mm5/mpp/helpdesk>

```

#
#-----
#
# Presently, of the MPP platforms only the "sp2"
# is supplied with the "make deck" capability.
#
# MPP Software Layer
MPP_LAYER=RSL
#MPP_LAYER=NNTSMS
#
# PROCMIN_NS - minimum number of processors allowed in N/S dim
#
PROCMIN_NS = 2
#
# PROCMIN_EW - minimum number of processors allowed in E/W dim
#
PROCMIN_EW = 4
#
# ASSUME_HOMOGENEOUS_ENVIRONMENT - on a machine with a heterogeneous
# mix of processors (different speeds) setting this compile time
# constant to 0 (zero) allows the program to detect the speed of each
# processor at the beginning of a run and then to attempt to come up with
# an optimal (static) mapping. Set this to 0 for a heterogeneous
# mix of processors, set it to 1 for a homogeneous mix. Unless you
# are certain you have a heterogeneous mix of processors, leave this
# set to 1. Currently, this option is ignored on platforms other
# than the IBM SP.
#
ASSUME_HOMOGENEOUS_ENVIRONMENT = 1

```

NOTA: En nuestro caso contamos con 8 procesadores, por lo tanto, este número debe ser el resultado de la multiplicación de PROCMIN_NS y PROCMIN_EW.

SECCION DE OPCIONES DE COMPILACIÓN

```

#-----
# 7g3. Linux PCs. Need INTEL and LAM/MPI.
#-----
RUNTIME_SYSTEM = "linux"
MPP_TARGET=$(RUNTIME_SYSTEM)
# edit the following definition for your system
LINUX_MPIHOME = /usr/local/lam_intel
MFC = $(LINUX_MPIHOME)/bin/mpif77
MCC = $(LINUX_MPIHOME)/bin/mpicc

```

```
MLD = $(LINUX_MPIHOME)/bin/mpif77
FCFLAGS = -O3 -convert big_endian -pc32 -axT
LDOPTIONS = -O3 -convert big_endian -pc32 -axT
LOCAL_LIBRARIES = -L$(LINUX_MPIHOME)/lib -llamf77mpi -lmpi -llam
MAKE = make -i -r
AWK = awk
SED = sed
CAT = cat
CUT = cut
EXPAND = /usr/bin/expand
M4 = m4
CPP = /lib/cpp -C -P
CPPFLAGS = -debug -traditional -DMPI -Dlinux -D2cFortran -axT
CFLAGS = -debug -DMPI -I/usr/local/lam_intel/include
ARCH_OBJS = milliclock.o
IWORDSIZ = 4
RWORDSIZ = 4
LWORDSIZ = 4
```

4. Se corre un programa llamado COMPILA.CSH que realiza lo siguiente:
 - Make uninstall Elimina cualquier mpp.exe que se haya creado anteriormente
 - Make mpclean Elimina los archivos ejecutables que ayudaron a crear mpp.exe
 - Make clean
 - Make mpp Genera el archivo MPP.exe basándose en la configuración del archivo configure.
5. Levantamos los demonios que van a controlar los 8 procesadores con la siguiente línea.
 - lamboot -v -ssi boot rsh lamhosts

En donde lamhost es un archivo que contiene el nombre de nuestra máquina y también contiene el número de procesadores con lo que cuenta.

6. Modificamos el archivo .deck de acuerdo a nuestra corrida, por ejemplo TIMAX,TISTEP,TAPFRQ,NESTIX,NESTJX,NESTI,NESTJ,IOVERW.

NOTA: Al final del archivo .deck debemos agregar la siguiente línea:

```
/usr/local/lam_intel/bin/mpirun -np 8 mm5.mpp
```

Establece que voy a correr el programa mm5.deck en modo multiprocesamiento con 8 procesadores.

7. Todo lo anterior se corre con la opción nohup archivo.deck&
(Corre el archivo .deck en modo background usando los 8 procesadores)

En el anexo 6 se pueden observar algunas de las salidas en modo de grafica de los resultados arrojados por el modelo MM5 para los periodos completos de estudio, como para compararlos una vez que se agregue el modelo hidrológico.

3. Calibración del modelo hidrológico (VIC)

La calibración del modelo hidrológico se realizó utilizando series de escurrimiento en la salida de la cuenca en estudio (Fig. 1), esta cuenca tiene su punto de salida en la central hidroeléctrica Peñitas.

Inicialmente se tenía considerado calibrar el modelo con un hidrograma obtenido aplicando métodos de antitránsito, descrito en el capítulo anterior, en el punto de salida de la presas Peñitas. Posteriormente se obtuvo información de la estación hidrométrica Las Peñitas, con registros en el periodo de 1950-2001, de la cual se utilizó un periodo de cinco años (1060-1064) para la calibración del modelo hidrológico VIC.

3.1 Capacitación

Se tuvo una semana de capacitación para operar el modelo hidrológico VIC. La capacitación fue realizada por el Dr. Francisco Muñoz Arriola, adscrito al Departamento de Ingeniería Civil y Ambiental de la Universidad de Washington liderada por el Dr. Dennis P. Lettenmaier.

El objetivo fue recibir instrucción sobre la estructura de archivos de entrada y salida del modelo fisiográfico y de ruta, asimismo de las variables de calibración. Al final de la semana de instrucción se revisaron y precisaron los resultados de una primera simulación con valores ajustados, identificando las variables más sensibles para después en forma iterativa ajustar los valores hasta lograr resultados aceptables en la calibración del modelo hidrológico VIC.

En la semana de capacitación se realizaron las siguientes actividades:

- Delimitación de la cuenca
- Revisión de las direcciones de flujo
- Revisión de los archivos de datos meteorológicos y fisiográficos
- Modificación de valores en las variables de calibración
- Revisión de la estructura de los algoritmos y archivos de datos y resultados

3.2 Modelo hidrológico

El modelo VIC (*Variable Infiltration Capacity*), es un modelo hidrológico de superficie a macroescala diseñado tanto para uso fuera de línea como para usarse en modelos acoplados suelo-atmosfera. VIC se distingue por la parametrización de los efectos de variabilidad espacial en la capacidad de infiltración del suelo, topografía y vegetación (*Liang et al, 1994*). Las unidades hidrológicas del modelo son celdas de tamaño de 1/8 de grado $\approx 12.5 \times 12.5$ kilómetros.

VIC permite realizar balances superficiales de agua y energía. Es un modelo de 3 capas de suelo y vegetación que funciona como se muestra en la figura 10

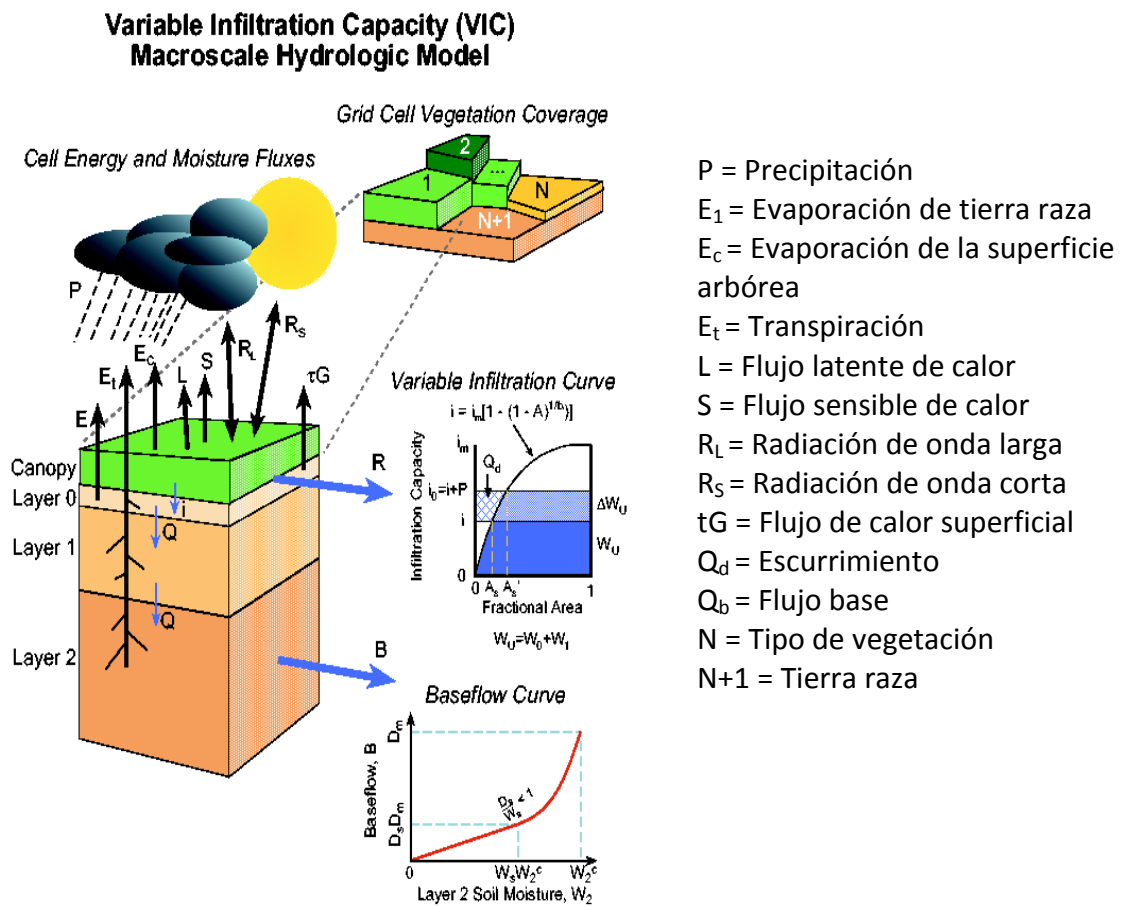


Figura 10. Balance de agua y energía en el modelo VIC

VIC se desarrolló dentro de un ambiente UNIX, se compila y ejecuta en ese ambiente. El proceso de calibración se inició identificando los archivos de entrada, de proceso y de

resultados. Los archivos de entrada contienen las características fisiográficas y meteorológicas de la cuenca, se propusieron valores iniciales para los parámetros fisiográficos, con los cuales se hicieron iteraciones comparando los resultados con series de escurrimiento medidas en el punto de salida de la cuenca.

3.2.1 Archivos de entrada de VIC

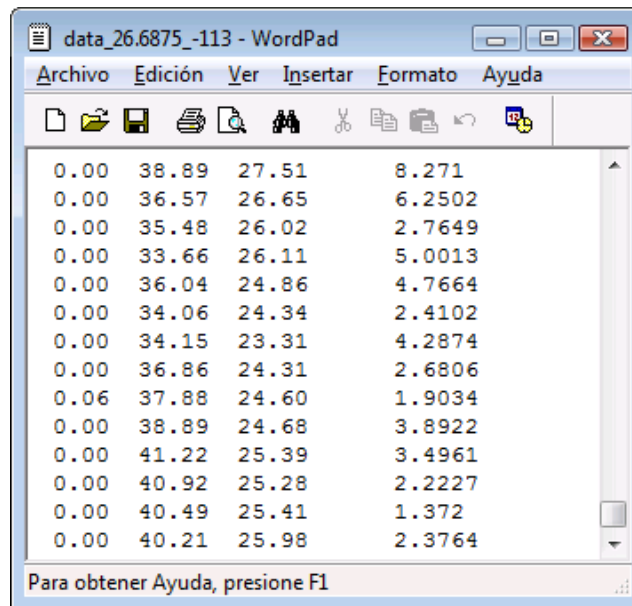
VIC utiliza los archivos de forzantes meteorológicos y parámetros fisiográficos, características del suelo, vegetación y nieve. Estos datos dentro de la estructura de VIC, se encuentran en dos archivos independientes.

a) Forzantes meteorológicos

Dentro de los forzantes meteorológicos se tienen los siguientes datos.

- Precipitación (mm)
- Temperatura máxima diaria (°C)
- Temperatura mínima diaria (°C)
- Velocidad de viento (m/s)

En esta carpeta hay un archivo por cada celda, que tiene el nombre de *data_lon_lat: data_14.5625_-92.3125*. Cada archivo tiene cuatro columnas con los datos mencionados. Estas series de tiempo inician el día 01/01/1925 y en nuestro caso, terminar el día 31/10/2004, la figura 2 muestra la estructura de los datos que conservan el mismo orden.

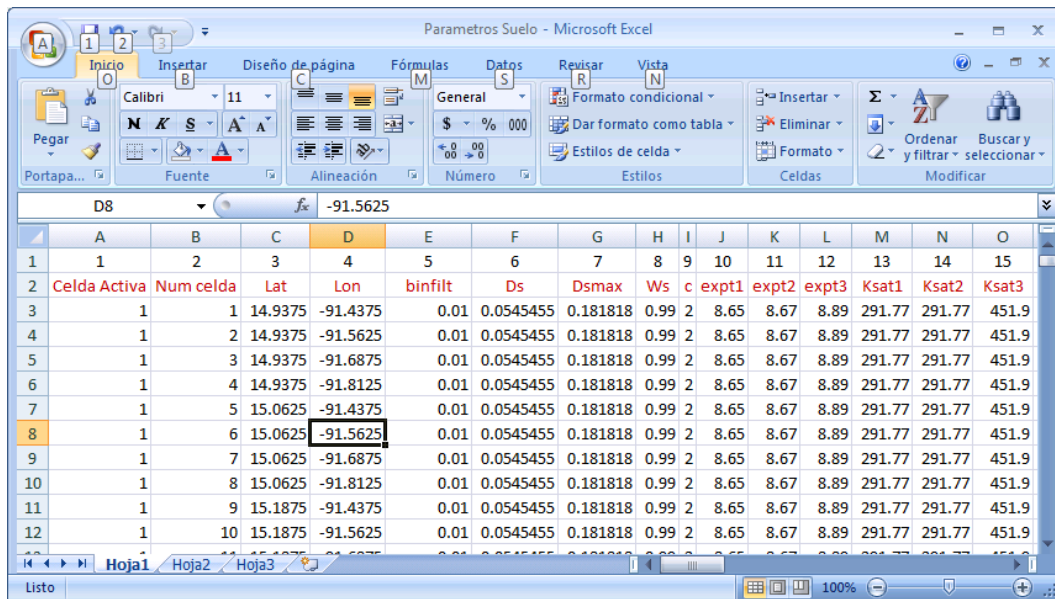


Precipitación	Temperatura máxima diaria	Temperatura mínima diaria	Velocidad de viento
0.00	38.89	27.51	8.271
0.00	36.57	26.65	6.2502
0.00	35.48	26.02	2.7649
0.00	33.66	26.11	5.0013
0.00	36.04	24.86	4.7664
0.00	34.06	24.34	2.4102
0.00	34.15	23.31	4.2874
0.00	36.86	24.31	2.6806
0.06	37.88	24.60	1.9034
0.00	38.89	24.68	3.8922
0.00	41.22	25.39	3.4961
0.00	40.92	25.28	2.2227
0.00	40.49	25.41	1.372
0.00	40.21	25.98	2.3764

Figura 11. Archivo de forzantes, Precipitación, Temperatura máxima diaria, Temperatura mínima diaria y Velocidad de viento

b) Parámetros de suelo

El archivo de suelo es un archivo general que contiene 53 parámetros por cada celda, algunos de los parámetros son por ejemplo: longitud, latitud, profundidad de las capas, humedad del suelo, etc., figura 12.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	Celda Activa	Num celda	Lat	Lon	binfilt	Ds	Dsmax	Ws	c	expt1	expt2	expt3	Ksat1	Ksat2	Ksat3
3	1	1	14.9375	-91.4375	0.01	0.0545455	0.181818	0.99	2	8.65	8.67	8.89	291.77	291.77	451.9
4	1	2	14.9375	-91.5625	0.01	0.0545455	0.181818	0.99	2	8.65	8.67	8.89	291.77	291.77	451.9
5	1	3	14.9375	-91.6875	0.01	0.0545455	0.181818	0.99	2	8.65	8.67	8.89	291.77	291.77	451.9
6	1	4	14.9375	-91.8125	0.01	0.0545455	0.181818	0.99	2	8.65	8.67	8.89	291.77	291.77	451.9
7	1	5	15.0625	-91.4375	0.01	0.0545455	0.181818	0.99	2	8.65	8.67	8.89	291.77	291.77	451.9
8	1	6	15.0625	-91.5625	0.01	0.0545455	0.181818	0.99	2	8.65	8.67	8.89	291.77	291.77	451.9
9	1	7	15.0625	-91.6875	0.01	0.0545455	0.181818	0.99	2	8.65	8.67	8.89	291.77	291.77	451.9
10	1	8	15.0625	-91.8125	0.01	0.0545455	0.181818	0.99	2	8.65	8.67	8.89	291.77	291.77	451.9
11	1	9	15.1875	-91.4375	0.01	0.0545455	0.181818	0.99	2	8.65	8.67	8.89	291.77	291.77	451.9
12	1	10	15.1875	-91.5625	0.01	0.0545455	0.181818	0.99	2	8.65	8.67	8.89	291.77	291.77	451.9

Figura 12. Parámetros de suelo

Algunos de estos parámetros son utilizados en la calibración y por tanto en el proceso de optimización.

c) Parámetros de vegetación

Los parámetros de vegetación usan dos archivos, uno es la librería de la vegetación o base de datos que contiene los tipos de vegetación y el otro despliega el concentrado de tipos de suelo que hay en cada una de las celdas. Por cada tipo de suelo hay diversas variables como son: la fracción de un tipo de vegetación, profundidad de las raíces en cada capa, índice foliar para cada mes, etc., figura 13.

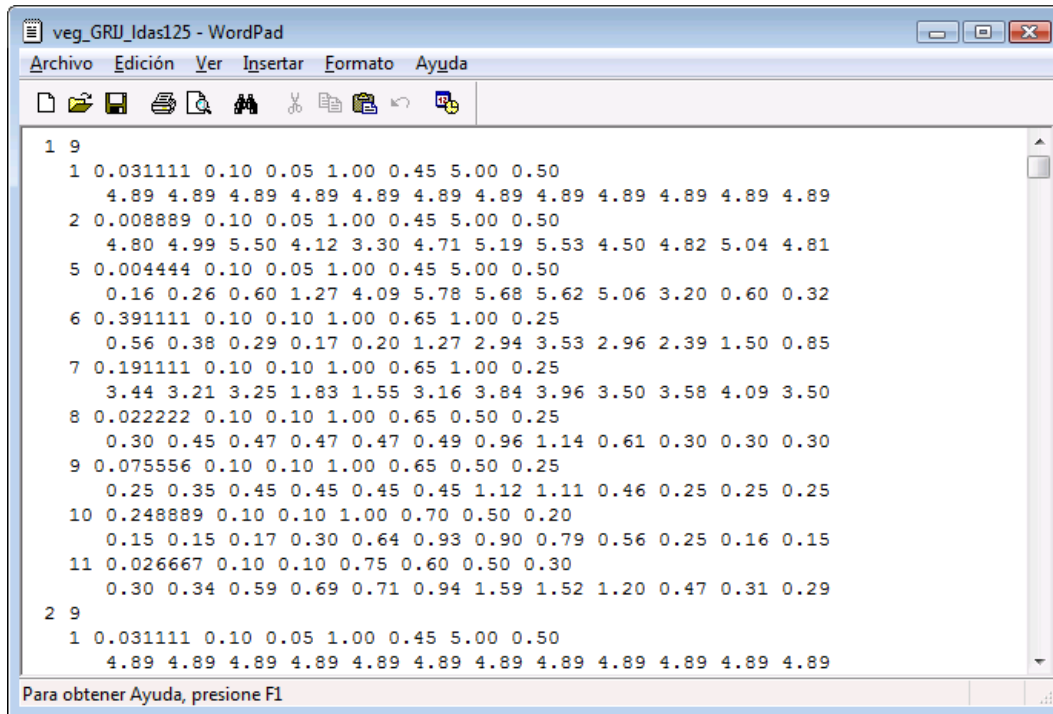
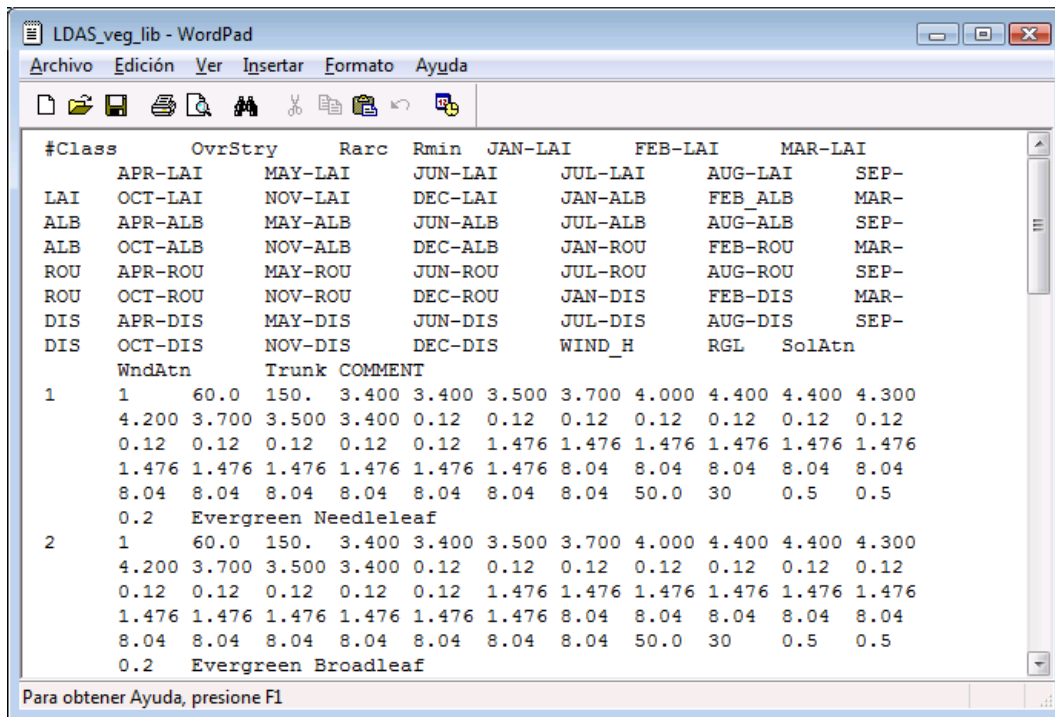


Figura 13. Archivo de vegetación para cada celda

Este archivo está acompañado de otro que contiene la librería de la vegetación, figura 14.



```

#Class      OvrStry      Rarc      Rmin      JAN-LAI      FEB-LAI      MAR-LAI
LAI      APR-LAI      MAY-LAI      JUN-LAI      JUL-LAI      AUG-LAI      SEP-
LAI      OCT-LAI      NOV-LAI      DEC-LAI      JAN-ALB      FEB-ALB      MAR-
ALB      APR-ALB      MAY-ALB      JUN-ALB      JUL-ALB      AUG-ALB      SEP-
ALB      OCT-ALB      NOV-ALB      DEC-ALB      JAN-ROU      FEB-ROU      MAR-
ROU      APR-ROU      MAY-ROU      JUN-ROU      JUL-ROU      AUG-ROU      SEP-
ROU      OCT-ROU      NOV-ROU      DEC-ROU      JAN-DIS      FEB-DIS      MAR-
DIS      APR-DIS      MAY-DIS      JUN-DIS      JUL-DIS      AUG-DIS      SEP-
DIS      OCT-DIS      NOV-DIS      DEC-DIS      WIND_H      RGL      SolAtn
WndAtn      Trunk COMMENT
1      1      60.0      150.      3.400      3.400      3.500      3.700      4.000      4.400      4.400      4.300
4.200      3.700      3.500      3.400      0.12      0.12      0.12      0.12      0.12      0.12      0.12      0.12
0.12      0.12      0.12      0.12      0.12      1.476      1.476      1.476      1.476      1.476      1.476      1.476
1.476      1.476      1.476      1.476      1.476      1.476      8.04      8.04      8.04      8.04      8.04      8.04
8.04      8.04      8.04      8.04      8.04      8.04      8.04      50.0      30      0.5      0.5
0.2      Evergreen Needleleaf
2      1      60.0      150.      3.400      3.400      3.500      3.700      4.000      4.400      4.400      4.300
4.200      3.700      3.500      3.400      0.12      0.12      0.12      0.12      0.12      0.12      0.12      0.12
0.12      0.12      0.12      0.12      0.12      1.476      1.476      1.476      1.476      1.476      1.476      1.476
1.476      1.476      1.476      1.476      1.476      1.476      8.04      8.04      8.04      8.04      8.04      8.04
8.04      8.04      8.04      8.04      8.04      8.04      8.04      50.0      30      0.5      0.5
0.2      Evergreen Broadleaf

```

Para obtener Ayuda, presione F1

Figura 14. Archivo de librería de la vegetación

VIC también permite calcular la nieve por medio de un archivo que contiene parámetros relativos a ella; este archivo al igual que para el suelo, contiene información por celda.

3.2.2 Simulaciones con VIC

Una vez definidos los parámetros de entrada, para realizar una corrida VIC necesita un archivo que contiene todos los parámetros de control de la simulación, así como las rutas de los archivos de entrada y salida. Figura 15.

Figura 15. Archivo Global File

Este archivo también permite guardar o leer un archivo de estado, es decir; se graba un archivo para una fecha precisa durante una simulación y éste se usa como archivo de entrada para otra simulación.

En este archivo también se definen los rangos de datos de tiempo para los archivos de entrada, por ejemplo: definir los datos de meteorología desde el 01/01/1925. Se define también los modos de operación de la simulación: balance energético completo o balance de agua, simulación con o sin bandas de nieve.

Una vez especificado este archivo, se corre VIC. El proceso es de la siguiente manera: (se puede copiar el ejecutable compilado dentro de la carpeta donde se encuentra el archivo global):

Nombre_del_ejecutable_de_VIC -g nombre_del_archivo_global.

Por ejemplo: **vicN1 -g globalfile**

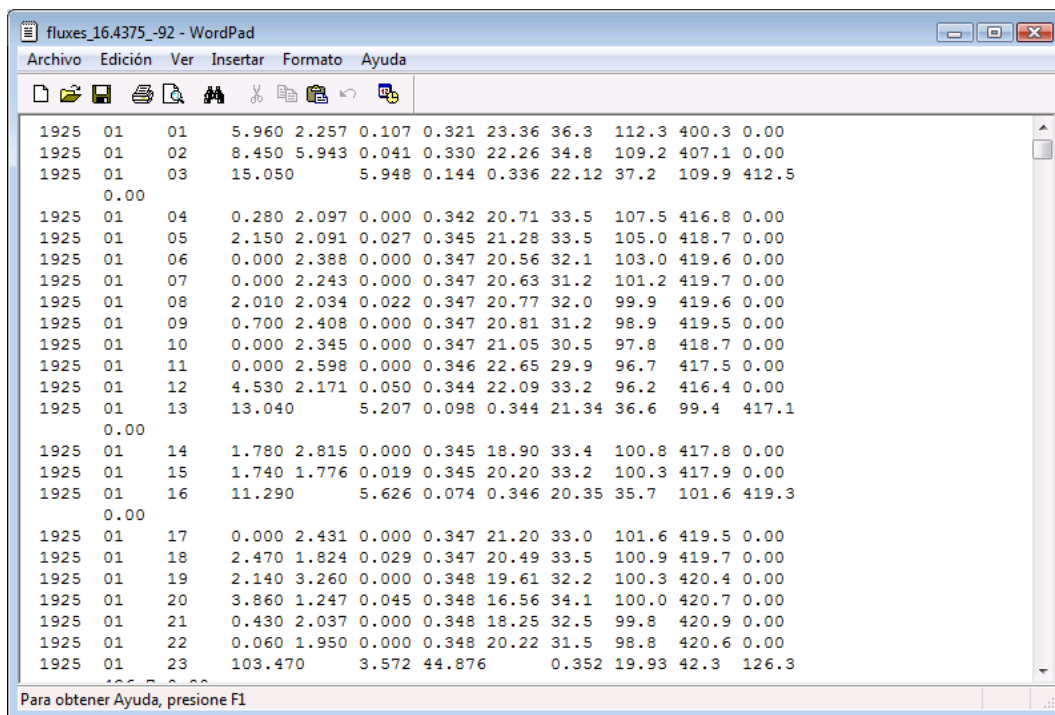
3.2.3 Archivos de salida de VIC

El modelo VIC corre en modo "vector", lo cual significa que para cada celda calcula la serie de tiempo total; por lo tanto cada celda es independiente. VIC produce flujos de humedad, energía y archivos de estado por cada celda.

Después de la corrida, los resultados se registran en la carpeta proporcionada por el usuario dentro del archivo global. Los resultados se muestran en archivos por cada una de las celdas de la red hidrográfica.

Cada archivo tiene por nombre: fluxes_longitud_latitud y contiene varios parámetros; como el año de simulación, el mes, el día, la precipitación, el escurrimiento superficial, el flujo base, etc.

En el código fuente de VIC, el usuario tiene la posibilidad de agregar o quitar los parámetros que se requieren obtener como resultado. Figura 16.



Year	Month	Day	1	2	3	4	5	6	7	8	9	10	11	12	13
1925	01	01	5.960	2.257	0.107	0.321	23.36	36.3	112.3	400.3	0.00				
1925	01	02	8.450	5.943	0.041	0.330	22.26	34.8	109.2	407.1	0.00				
1925	01	03	15.050	5.948	0.144	0.336	22.12	37.2	109.9	412.5	0.00				
1925	01	04	0.280	2.097	0.000	0.342	20.71	33.5	107.5	416.8	0.00				
1925	01	05	2.150	2.091	0.027	0.345	21.28	33.5	105.0	418.7	0.00				
1925	01	06	0.000	2.388	0.000	0.347	20.56	32.1	103.0	419.6	0.00				
1925	01	07	0.000	2.243	0.000	0.347	20.63	31.2	101.2	419.7	0.00				
1925	01	08	2.010	2.034	0.022	0.347	20.77	32.0	99.9	419.6	0.00				
1925	01	09	0.700	2.408	0.000	0.347	20.81	31.2	98.9	419.5	0.00				
1925	01	10	0.000	2.345	0.000	0.347	21.05	30.5	97.8	418.7	0.00				
1925	01	11	0.000	2.598	0.000	0.346	22.65	29.9	96.7	417.5	0.00				
1925	01	12	4.530	2.171	0.050	0.344	22.09	33.2	96.2	416.4	0.00				
1925	01	13	13.040	5.207	0.098	0.344	21.34	36.6	99.4	417.1	0.00				
1925	01	14	1.780	2.815	0.000	0.345	18.90	33.4	100.8	417.8	0.00				
1925	01	15	1.740	1.776	0.019	0.345	20.20	33.2	100.3	417.9	0.00				
1925	01	16	11.290	5.626	0.074	0.346	20.35	35.7	101.6	419.3	0.00				
1925	01	17	0.000	2.431	0.000	0.347	21.20	33.0	101.6	419.5	0.00				
1925	01	18	2.470	1.824	0.029	0.347	20.49	33.5	100.9	419.7	0.00				
1925	01	19	2.140	3.260	0.000	0.348	19.61	32.2	100.3	420.4	0.00				
1925	01	20	3.860	1.247	0.045	0.348	16.56	34.1	100.0	420.7	0.00				
1925	01	21	0.430	2.037	0.000	0.348	18.25	32.5	99.8	420.9	0.00				
1925	01	22	0.060	1.950	0.000	0.348	20.22	31.5	98.8	420.6	0.00				
1925	01	23	103.470	3.572	44.876	0.352	19.93	42.3	126.3						

Figura 16. Ejemplo de un archivo de salida de VIC para una celda

3.2.4 Procesamientos con el modelo ROUT

Para obtener el hidrograma de salida y los promedios mensuales de los flujos de energía y agua, se usa un modelo independiente llamado ROUT que sirve para llevar a cabo el cálculo de avenidas.

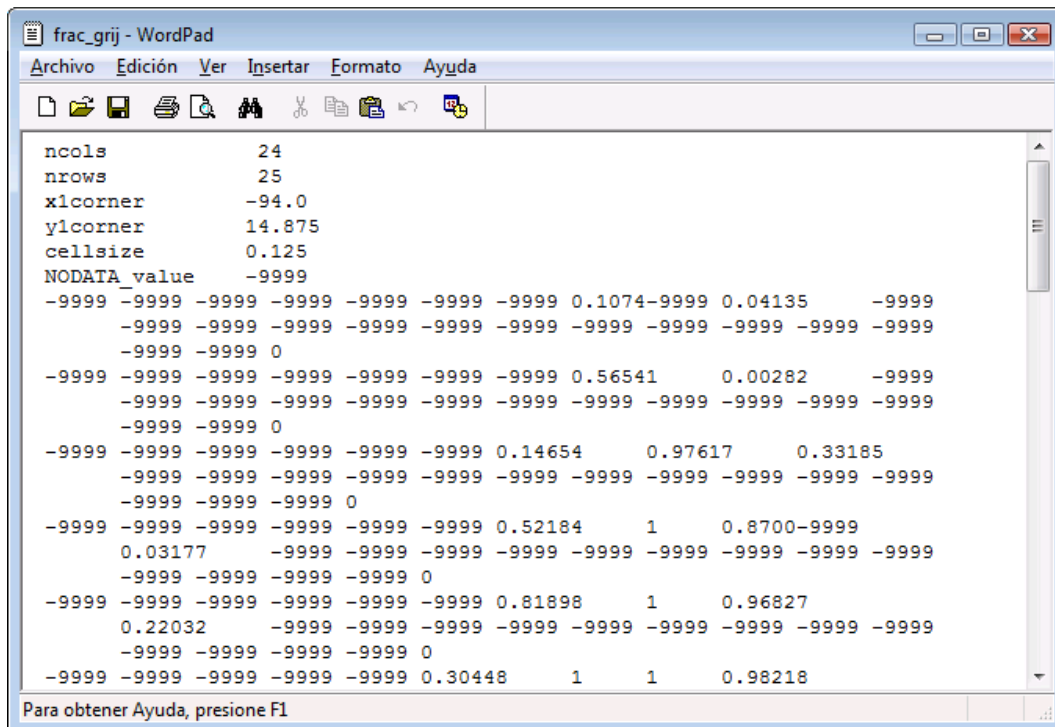
El modelo ROUT toma los flujos de VIC en todas las celdas de una cuenca, lee el escurrimiento y el flujo base diarios y combina escurrimiento y flujo base con la función de respuesta del hidrograma unitario; además adiciona la respuesta del hidrograma local al flujo desde flujo aguas-arriba y se propaga el flujo aguas-abajo.

3.2.5 Archivos de entrada de ROUT

ROUT necesita varios archivos de entrada.

- Archivos de salida diarios de VIC (deben ser en código ASCII y contiene en las columnas 1-3: año, mes y día; y en la columna 6 y 7 contiene el escurrimiento y flujo base)
- Archivo que indica la fracción de cada celda en la cuenca
- Archivo de dirección de flujo en cada celda
- Archivo del hidrograma unitario: forma del hidrograma unitario
- Archivo de ubicación de estaciones hidrométricas

El formato de los diferentes archivos se observan en las figuras 17 a 20.



```

frac_grij - WordPad
Archivo Edición Ver Insertar Formato Ayuda
ncols      24
nrows     25
xlcorner  -94.0
ylcorner   14.875
cellsize   0.125
NODATA_value -9999
-9999 -9999 -9999 -9999 -9999 -9999 -9999 0.1074-9999 0.04135 -9999
-9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999
-9999 -9999 0
-9999 -9999 -9999 -9999 -9999 -9999 -9999 0.56541 0.00282 -9999
-9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999
-9999 -9999 0
-9999 -9999 -9999 -9999 -9999 -9999 0.14654 0.97617 0.33185
-9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999
-9999 -9999 -9999 0
-9999 -9999 -9999 -9999 -9999 -9999 0.52184 1 0.8700-9999
0.03177 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999
-9999 -9999 -9999 -9999 0
-9999 -9999 -9999 -9999 -9999 -9999 0.81898 1 0.96827
0.22032 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999
-9999 -9999 -9999 -9999 0
-9999 -9999 -9999 -9999 -9999 0.30448 1 1 0.98218
Para obtener Ayuda, presione F1

```

Figura 17. Fracción de cada celda en la cuenca

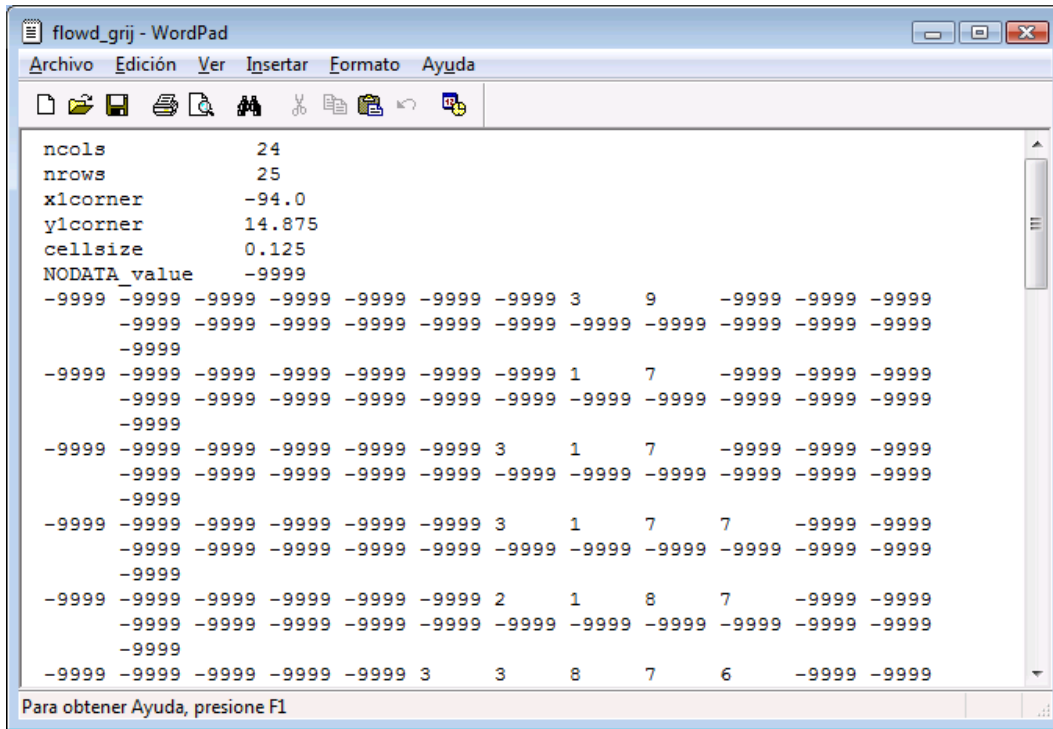


Figura 18. Dirección de flujo

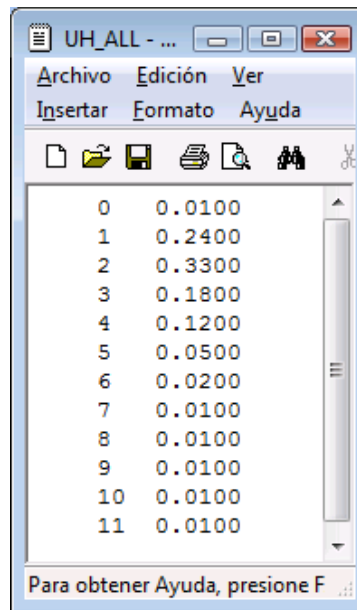


Figura 19. Hidrograma unitario

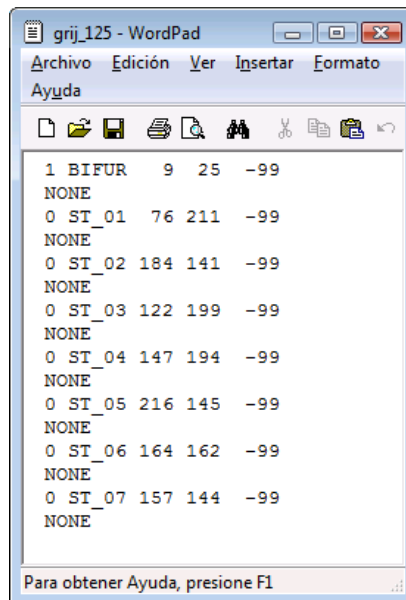


Figura 20 Ubicación de estaciones hidrométricas

3.2.6 Correr ROUT

Al igual que VIC, ROUT necesita un archivo general para ubicar los archivos de entrada y salida del modelo, así como las salidas del modelo VIC, y los años que se quieren representar dentro de las simulaciones que se hicieron en VIC, como se observa en la figura 21.

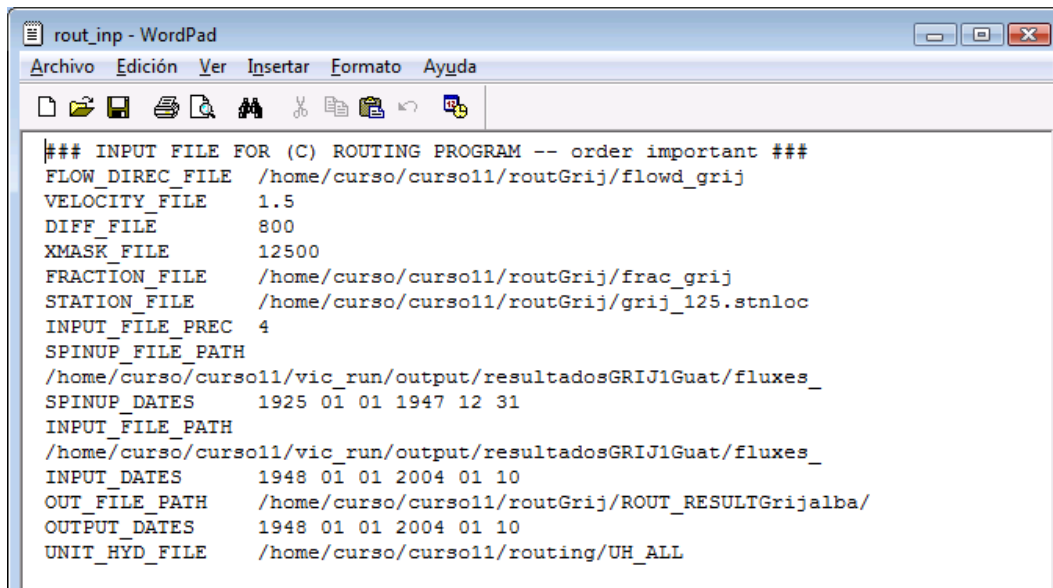


Figura 21. Archivo global para ROUT

Para correr el archivo se procede de la siguiente manera: Se copia el archivo ejecutable en la misma carpeta que el archivo global de la siguiente manera.

Ejecutable_ROUT archivo_global

Ejemplo: *routv4l routinp.grij*

3.2.7 Archivos de salida de ROUT

ROUT entrega 3 tipos de archivos: diario, mensual y anual (Fig. 22), por defecto ROUT los da en pies cúbicos por segundo (ft^3/s). Para tener m^3/s se aplica el factor de conversión (0.028).

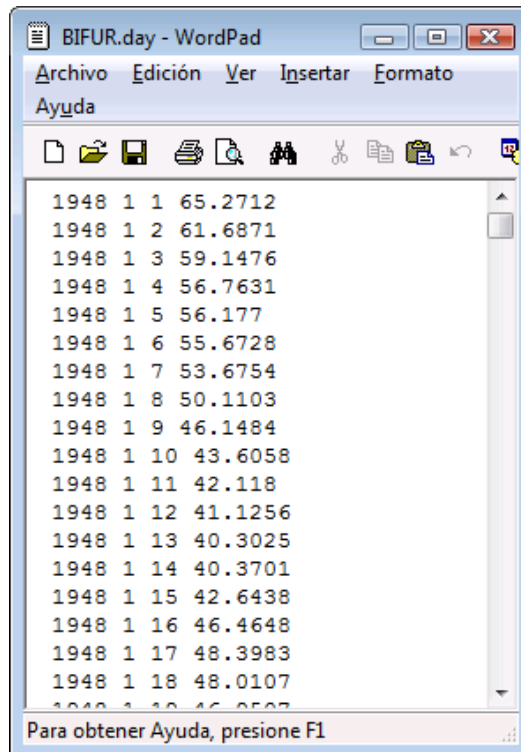


Figura 22. Ejemplo del archivo de salida de ROUT diario en m^3/s

3.2.8 Scripts para sintetizar archivos

VIC contiene algunas subrutinas llamadas scripts, que permiten crear o modificar archivos de VIC.

a) Activate_basin.pl

Este script se usa para activar o desactivar una parte de una cuenca o toda la cuenca.

El archivo de suelos de una simulación, se define en las celdas activas con el número 1 y no activas con el número 0. Para modificar el archivo de suelos y activar o desactivar celdas, se sigue el siguiente proceso:

Crear un archivo que contenga para cada celda su ubicación geográfica (longitud y latitud). Se puede hacer a partir del archivo de suelo, recuperando las 2 columnas de longitud y latitud, por ejemplo; usando el lenguaje de programa interno de unix awk:

```
awk '{print $4, $3} archivo_suelo > archivo_lon_lat Activate_basin.pl archivo_suelo_anterior  
archivo_suelo_nuevo lonlat
```

Cuando se proporciona un archivo con longitud en la primera columna y latitud en la segunda, el programa reconoce que debe desactivar las celdas; cuando es un archivo con latitud y longitud es para activarlas (se usa de la misma manera).

b) Agg_space.pl

Este script se usa para sacar sumas de todas las celdas de los archivos de salida de VIC de forma temporal.

Para correr se necesita un archivo que contiene las direcciones, o rutas de los resultados de VIC.

Dentro del ambiente UNIX se puede sacar de la manera siguiente:

```
ls ruta/fluxes_* > listaRutas
```

El programa agg_space.pl obtiene la suma de todas las celdas en la serie de tiempo que corresponde a la simulación. Se puede sacar un promedio horario, diario, mensual o anual como enseguida se explica:

```
Agg_space.pl -i archivo_listaRutas -r resolucion -o nombre_archivo_salida -in tipo_promedio
```

Un ejemplo con una resolución a 1/8 grado = 0.125, y una salida diaria:

```
Agg_space.pl -i listaRutas -r 0.125 -o arch_salida -in daily
```

c) Agg_time.pl

Script que permite sacar promedios mensuales o anuales de un archivo diario de salida de VIC (para una sola celda).

De la manera siguiente:

```
Agg_time.pl -i archivo_salida_VIC -o nombre_archivo_salida -in tipo_entrada -out tipo_salida
```

Enseguida un ejemplo:

```
Agg_time.pl -i fluxes_32.6875_-115.3125 -o salidamensual -in daily -out monthly
```

3.2.9 Calibración del modelo VIC

En los estudios realizados por el equipo que desarrolló el programa VIC se definieron algunos parámetros para la calibración. Las conclusiones a las que llegaron permitieron definir cuáles son los parámetros que tienen más influencia en la calibración.

De ahí que los parámetros que necesitan ser calibrados son principalmente los parámetros del archivo de suelo y que son los siguientes:

- Parámetro de Infiltración (b_i)
- Ancho de las dos primeras capas de suelo (Z_1, Z_2, Z_3)
- Velocidad máxima del flujo base (D_m)
- Fracción máxima de flujo base (D_s)
- Fracción máxima de la humedad de suelo en la tercera capa (W_s)

Por ello se puso especial interés en estos parámetros para realizar la calibración del modelo VIC.

a) Definición de parámetros de calibración

Para hacer una simulación y calibración necesitamos definir el periodo de simulación y un juego de parámetros que queremos calibrar.

El periodo de simulación que elegimos es: de 01/01/1960 a 12/31/1963. Y los parámetros a calibrar son el coeficiente de infiltración (B_{inf}), y las profundidades de cada capa de suelo ($DepthSoil_1$, $DepthSoil_2$ y $DepthSoil_3$).

b) Grupos de parámetros

En el archivo de suelo tiene 57 parámetros, y en nuestro caso (de Grijalva) tenemos 258 celdas. Si queremos calibrar 4 parámetros para cada celda son realmente 1,032 parámetros, demasiados para el proceso de calibración.

Por tal motivo se hicieron agrupaciones de celdas en función de la altura.

- Grupo 1 para las celdas que tienen una altura de 0 a 479 m
- Grupo 2 para las celdas que tienen una altura de 480 a 959 m
- Grupo 3 para las celdas que tienen una altura de 960 a 1439 m
- Grupo 4 para las celdas que tienen una altura de 1440 a 1919 m
- Grupo 5 para las celdas que tienen una altura superior a 1920m

De esa manera tenemos que optimizar 20 parámetros, 4 para cada grupo.

c) Simulación de VIC con un primer juego de parámetros

Para aproximar el modelo VIC a los datos históricos para el periodo escogido, se prueban varios juegos de parámetros hasta tener una calibración aceptable.

Después de varias iteraciones se logró tener una buena calibración. Los valores obtenidos del juego de parámetros se presentan en la tabla 8.

	B_{inf}	$DepthSoil_1$	$DepthSoil_2$	$DepthSoil_3$
Grupo 1	0.01	0.10	0.20	0.50
Grupo 2	0.03	0.20	0.30	1.0
Grupo 3	0.05	0.20	0.40	0.60
Grupo 4	0.07	0.30	0.70	0.80
Grupo 5	0.10	0.10	0.90	1.00

Tabla 8. *Primera aproximación de parámetros calibrados.*

Posteriormente se modifica el archivo de suelo con este juego valores de los parámetros como datos de entrada para VIC.

Para hacer esa simulación tenemos que hacer una primera con años anteriores para crear un archivo de estado que será un archivo de entrada. Ese archivo se crea un día antes de la simulación (31/12/1959). Este mismo permite tener una humedad inicial.

En la figura 23 se muestran los resultados obtenidos, en la cual se muestran los resultados comparándolos con los datos históricos de la estación hidrométrica Peñitas.

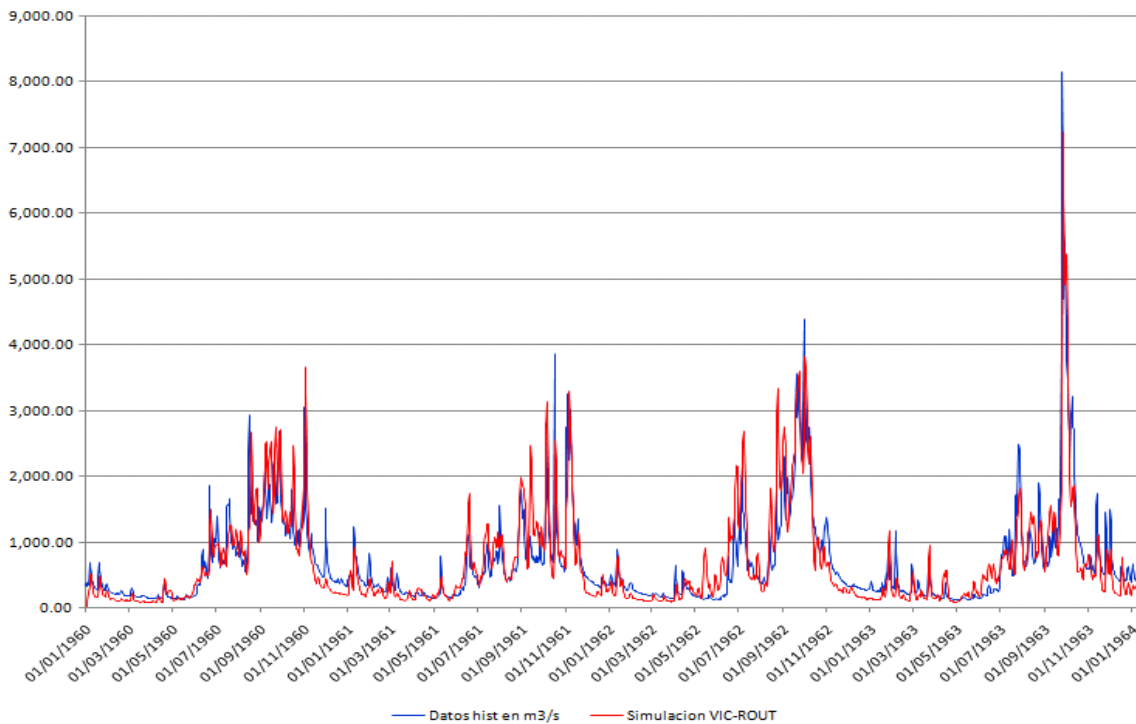


Figura 23. Resultados de la calibración.

En esta simulación podemos ver que ya tenemos un resultado muy cerca del histórico. Para ver de qué magnitud es el acercamiento podemos calcular el valor de *Nash Sutcliffe*, definido de la manera siguiente:

$$Nash = 1 - \frac{\sum_{i=1}^n (Qs_i - Qo_i)^2}{\sum_{i=1}^n (\overline{Qo} - Qo_i)^2}$$

En este caso, el valor de Nash es igual a 0.708. Todo este proceso de calibración nos permite crear los archivos de control de VIC y ROUT.

3.2.10 Optimización de parámetros de VIC

El proceso de calibración se realizó de manera iterativa, realizando varias simulaciones modificando cada vez, los valores de los parámetros, este proceso requiere de mucho tiempo, por lo que se optó por aplicar un método de optimización de parámetros para automatizar el proceso de calibración, para lo cual se aplicó el método PEST.

a) Modelo de optimización PEST

PEST es un paquete de estimación de parámetros no lineal, permite realizar simulación inversa y funciona con un modelo matemático, en nuestro caso VIC-ROUT. El método de optimización que usa PEST es *Gauss Leverberg Marquardt*. Es un método iterativo de optimización no lineal para solucionar problemas de ajuste de mínimos cuadrados de modelos no lineales.

PEST puede ser utilizado para estimar los parámetros de cualquier modelo existente, corre un modelo tantas veces ajustando los parámetros hasta que las diferencias cuadráticas entre las salidas del modelo y las observaciones del modelo se reducen a un mínimo.

Es una alternativa del método de Gauss Newton para buscar el mínimo de la función objetivo que es la suma de los cuadrados de funciones no lineales.

$$F(x) = \frac{1}{2} \sum_{i=1}^m [f_i(x)]^2$$

b) Archivos de entrada y salida de PEST

PEST requiere diferentes archivos de entrada:

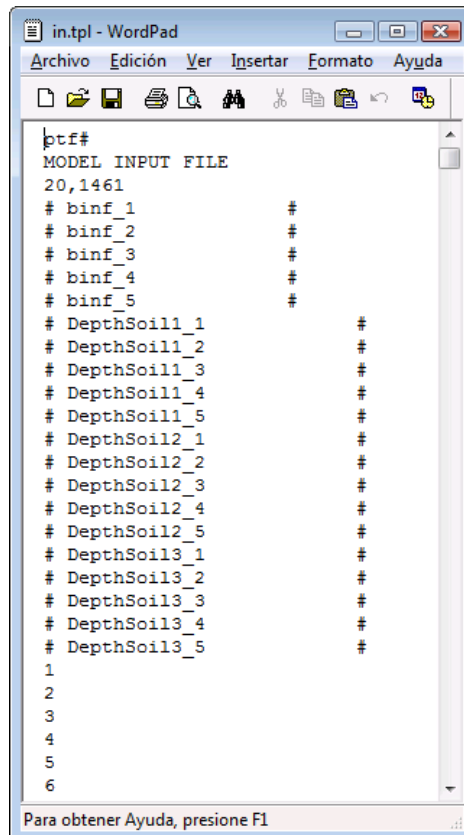
- Archivos *template*
- Archivos de instrucciones
- Archivo PEST Control File.

Estos archivos contienen datos de los parámetros a optimizar, las observaciones contra las cuales se va a comparar los resultados y un archivo de control de PEST.

Por lo que se necesita crear a mano cuatro archivos para comenzar el proceso de PEST.

c) Archivo template: in.tpl

Este archivo contiene el número de parámetros y el número de observaciones así como el nombre de cada parámetro y una lista ligada con un número que nos permite recuperar cada valor de las observaciones.

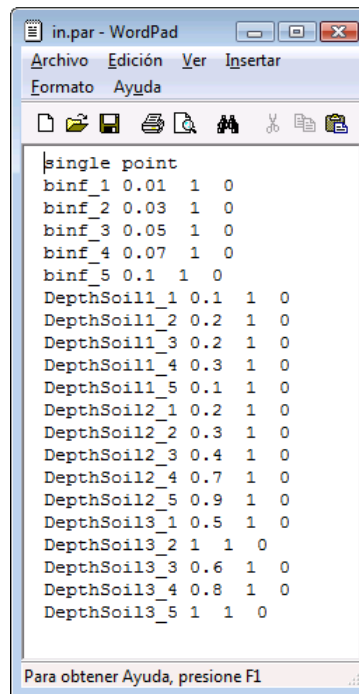


```
ptf#
MODEL INPUT FILE
20,1461
# binf_1           #
# binf_2           #
# binf_3           #
# binf_4           #
# binf_5           #
# DepthSoil1_1     #
# DepthSoil1_2     #
# DepthSoil1_3     #
# DepthSoil1_4     #
# DepthSoil1_5     #
# DepthSoil2_1     #
# DepthSoil2_2     #
# DepthSoil2_3     #
# DepthSoil2_4     #
# DepthSoil2_5     #
# DepthSoil3_1     #
# DepthSoil3_2     #
# DepthSoil3_3     #
# DepthSoil3_4     #
# DepthSoil3_5     #
1
2
3
4
5
6
```

Figura 24. Ejemplo del archivo de formato con 20 parámetros y 1461 observaciones: in.tpl

d) Archivo de parámetros (in.par)

Este archivo contiene dos variables que corresponden a la precisión de un parámetro (simple o doble precisión) y si es decimal o no ("point" o "nopoint"). Contiene también el nombre del parámetro y su valor, y 2 parámetros que corresponden a la escala y el desplazamiento. En nuestro caso estos parámetros valen 1 cuando están activos y 0 cuando están inactivos, figura 25.



```

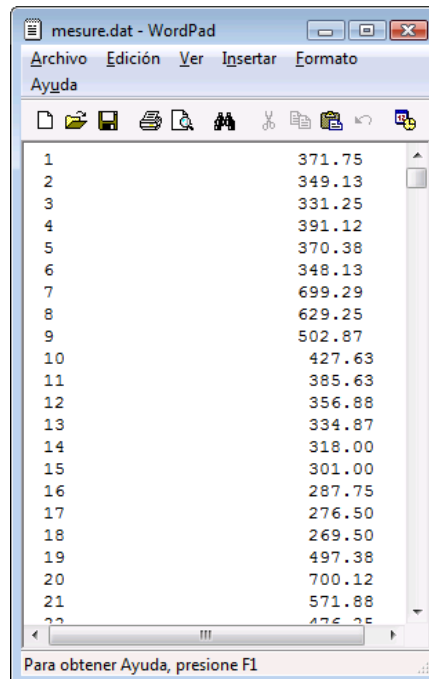
single point
binf_1 0.01 1 0
binf_2 0.03 1 0
binf_3 0.05 1 0
binf_4 0.07 1 0
binf_5 0.1 1 0
DepthSoil1_1 0.1 1 0
DepthSoil1_2 0.2 1 0
DepthSoil1_3 0.2 1 0
DepthSoil1_4 0.3 1 0
DepthSoil1_5 0.1 1 0
DepthSoil2_1 0.2 1 0
DepthSoil2_2 0.3 1 0
DepthSoil2_3 0.4 1 0
DepthSoil2_4 0.7 1 0
DepthSoil2_5 0.9 1 0
DepthSoil3_1 0.5 1 0
DepthSoil3_2 1 1 0
DepthSoil3_3 0.6 1 0
DepthSoil3_4 0.8 1 0
DepthSoil3_5 1 1 0

```

Figura 25. Ejemplo del archivo de parámetros: *in.par*

e) Archivo de observaciones (Measure.dat)

Este archivo tiene un número que corresponde a cada observación, así como los valores de cada observación contra las cuales se van a comparar las salidas del modelo ROUT, figura 26.



```

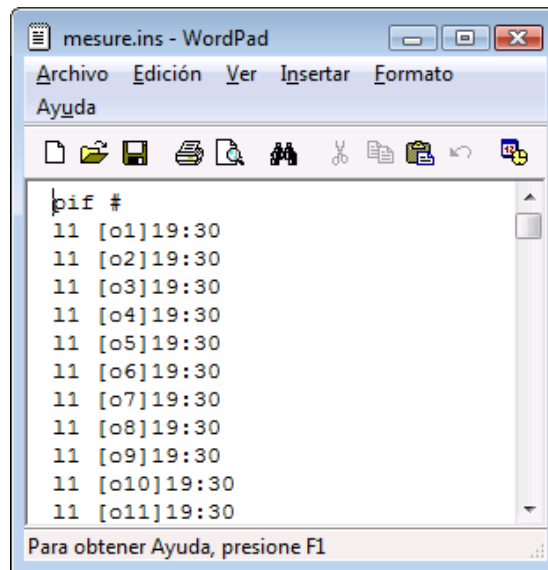
1 371.75
2 349.13
3 331.25
4 391.12
5 370.38
6 348.13
7 699.29
8 629.25
9 502.87
10 427.63
11 385.63
12 356.88
13 334.87
14 318.00
15 301.00
16 287.75
17 276.50
18 269.50
19 497.38
20 700.12
21 571.88

```

Figura 26. Ejemplo del archivo de observaciones con valores: *measure.dat*

f) Archivo de instrucciones (Measure.ins)

Este archivo contiene un número que corresponde a cada observación así como un formato que permite a PEST saber dónde leer la salida del modelo. Este archivo puede permitir leer directamente la salida del modelo, o como en nuestro caso un archivo parecido a *measure.dat* que contendrá los valores de salida del modelo, figura 27. Para este trabajo, el archivo contiene un valor para cada observación así como donde leer el valor contra el cual se va a comparar el modelo contra las mediciones.



```
pif #
11 [o1]19:30
11 [o2]19:30
11 [o3]19:30
11 [o4]19:30
11 [o5]19:30
11 [o6]19:30
11 [o7]19:30
11 [o8]19:30
11 [o9]19:30
11 [o10]19:30
11 [o11]19:30
```

Figura 27. Ejemplo del archivo de observaciones con formato de lectura: *measure.ins*

g) Verificación de los archivos con PEST

Una vez creados los archivos dentro de Windows, se tienen que transformar a formato *Unix* donde se va a hacer la optimización. Se convierten a *Unix* con el programa *dos2unix*.

PEST tiene diferentes funciones que le permiten verificar los archivos creados por el usuario y crear otros archivos necesarios para la corrida. Para verificar si el archivo *template* se creó de manera correcta, PEST pone a nuestra disposición un programa que se llama *tempchek*. Al ejecutar esta aplicación, si no tenemos errores es porque nuestro archivo se creó de manera correcta, figura 28.

Tempchek in.tpl

```
[curso11@galaxy ~/pest]$ tempchek in.tpl
TEMPCHEK Version 12.0. Watermark Numerical Computing.

Errors in file in.tpl ----->
No errors encountered.

20 parameters identified in file in.tpl: these are listed in file in.pmt.
```

Figura 28. Verificación con tempchek del archivo template

Podemos ver que en este caso PEST encontró 20 parámetros y creó un archivo in.pmt (este archivo solo contiene los nombres de los parámetros).

Para verificar el archivo in.par, podemos hacer lo mismo agregando el archivo in.dat, figura 29:

Tempchek in.tpl modfile in.par

```
[curso11@galaxy ~/pest]$ tempchek in.tpl modfile in.par
TEMPCHEK Version 12.0. Watermark Numerical Computing.

Errors in file in.tpl ----->
No errors encountered.

Errors in parameter value file in.par ----->
No errors encountered.

Writing model input file modfile ----->
File modfile written ok.
```

Figura 29. Verificación con tempchek del archivo de parámetros

Para verificar el archivo de instrucciones, PEST pone a nuestra disposición el programa inschek, figura 30.

Inschek mesure.ins

```
[curso11@galaxy ~/pest]$ inschek mesure.ins
INSCHEK Version 12.0. Watermark Numerical Computing.

Errors in file mesure.ins ----->
No errors encountered.

1461 observations identified in file mesure.ins: these are listed in file
mesure.obf.
```

Figura 30. Verificación con inschek del archivo de instrucciones

Podemos ver que PEST encontró 1461 observaciones y creó el archivo mesure.obf que contiene el nombre de cada observación, figura 31.

Para verificar el archivo `measure.dat` se verifica de la misma manera:

inschek measure.ins measure.dat

```
[curso11@galaxy ~/pest]$ inschek measure.ins measure.dat
INSCHEK Version 12.0. Watermark Numerical Computing.

Errors in file measure.ins ----->
No errors encountered.

Reading model output file measure.dat ----->
No errors encountered.

1461 observations identified in file measure.ins: these are listed in file
measure.obf together with their values as read from file measure.dat.
```

Figura 31. Verificación con *inschek* del archivo de datos

Podemos ver que cambió el archivo `measure.obf` creado por PEST; se agregaron valores a cada nombre de observación.

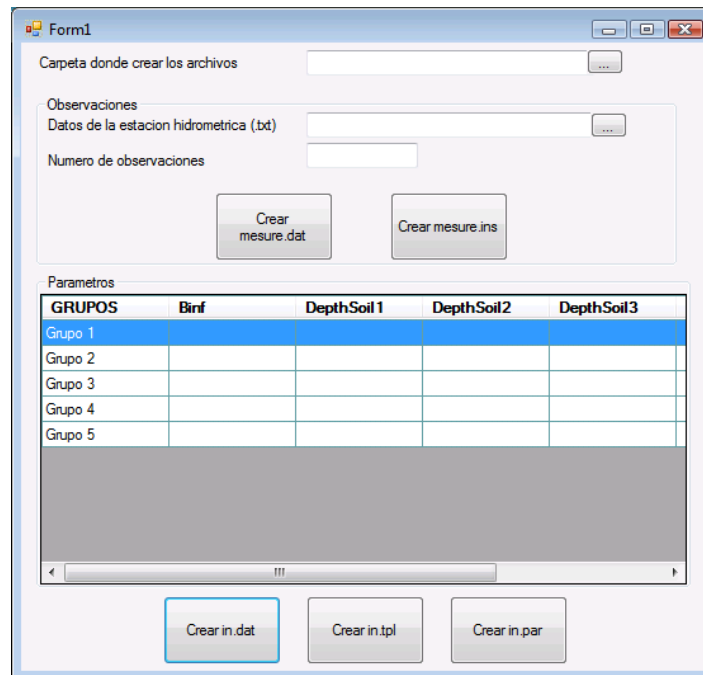
h) Interfaz para crear los archivos de PEST

Para crear de manera automática los archivos necesarios para hacer una corrida con PEST, se realizó una interfaz. La interfaz tiene 2 partes, una parte para los archivos que corresponden a las observaciones y otra que corresponde a los parámetros.

Para la primera parte es necesario tener un archivo que contiene los datos que corresponden a los años de simulación de la estación hidrométrica (en nuestro caso esas son las observaciones). En la segunda parte, se llena la tabla que contiene los parámetros que queremos optimizar. Para las 2 partes se pide al usuario el número de datos (de observaciones).

Esta interfaz crea los archivos siguientes: *measure.dat*, *measure.ins*, *in.tpl*, *in.dat* e *in.par*.

El archivo *in.dat* no es directamente necesario pero lo vamos a necesitar para el modelo de intercambio entre PEST y VIC. Es un archivo que contiene solamente los valores de cada parámetro así como el número que corresponde a cada observación, figura 32 y 33.



Form1

Carpeta donde crear los archivos

Observaciones
Datos de la estación hidrometrica (.txt)

Numero de observaciones

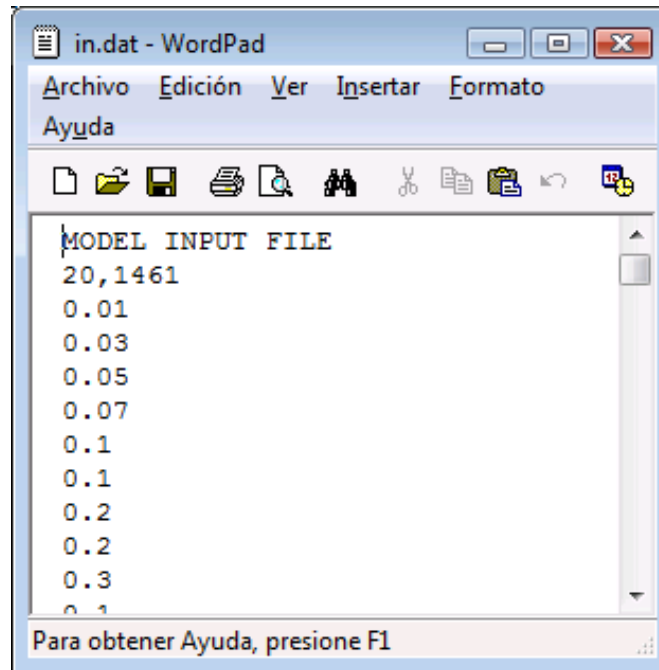
Crear measure.dat Crear measure.ins

Parametros

GRUPOS	Birf	DepthSoil 1	DepthSoil2	DepthSoil3
Grupo 1				
Grupo 2				
Grupo 3				
Grupo 4				
Grupo 5				

Crear in.dat Crear in.tpl Crear in.par

Figura 32. Interfaz para crear los archivos de entrada de PEST



in.dat - WordPad

Archivo Edición Ver Insertar Formato
Ayuda

```

MODEL INPUT FILE
20,1461
0.01
0.03
0.05
0.07
0.1
0.1
0.2
0.2
0.3
0.1

```

Para obtener Ayuda, presione F1

Figura 33. Ejemplo del archivo de datos de parámetros: in.dat

i) Archivo de control PEST

Pest trabaja con un archivo que contiene todos los datos necesarios para la optimización. Este se llama "*Pest control File*". El archivo se puede generar de manera automática por PEST, aunque se tendrá que modificar con el nombre del modelo, los nombres de archivos de entrada y salida, los límites de los parámetros y otros parámetros específicos a PEST que el usuario podría desear cambiar. Para eso, PEST necesita 2 archivos: el archivo ***par* y el archivo que contiene las observaciones con los datos *measure.obf*.

Se crea el archivo de control con *pestgen*. Por ejemplo si llamamos *ejemplo1* el archivo de control, PEST pone por defecto la extensión ***pst*, figura 34 y 35.

Pestgen ejemplo1 in.par measure.obf

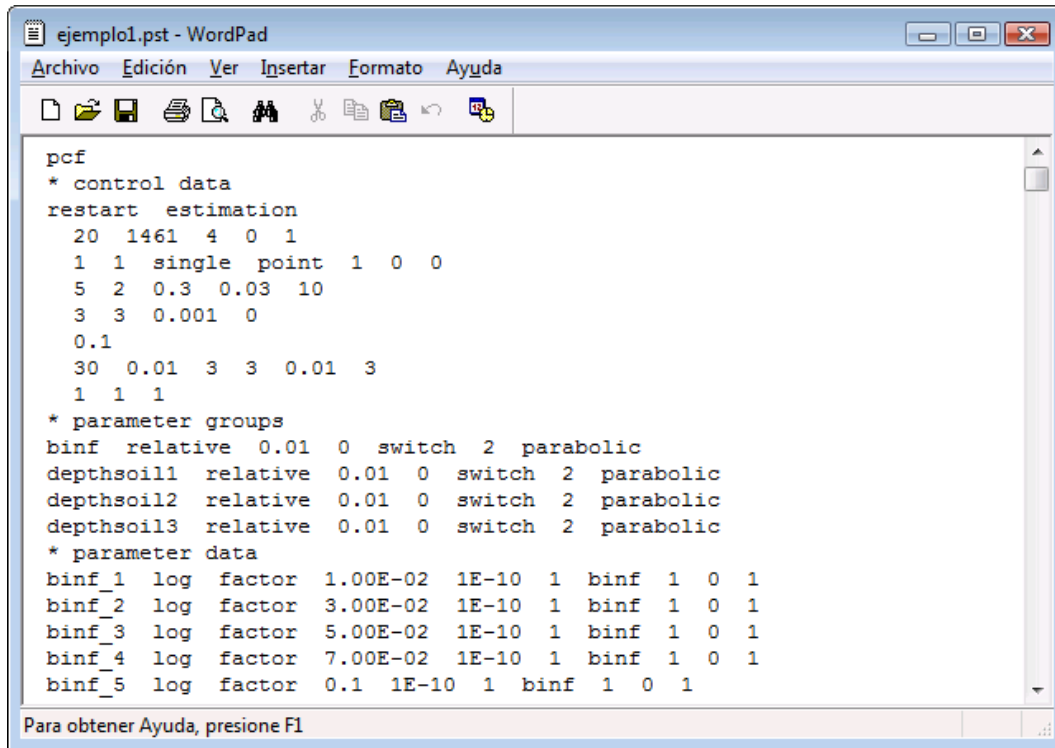
```
[curs011@galaxy ~/pest]$ pestgen ejemplo1 in.par measure.obf
PESTGEN Version 12.0. Watermark Numerical Computing.

Checking parameter value file in.par ----->
No errors encountered.

Checking observation value file measure.obf ----->
No errors encountered.

Writing PEST control file ejemplo1.pst ----->
File written ok.
```

Figura 34. Creación del archivo de control: *ejemplo1.pst*



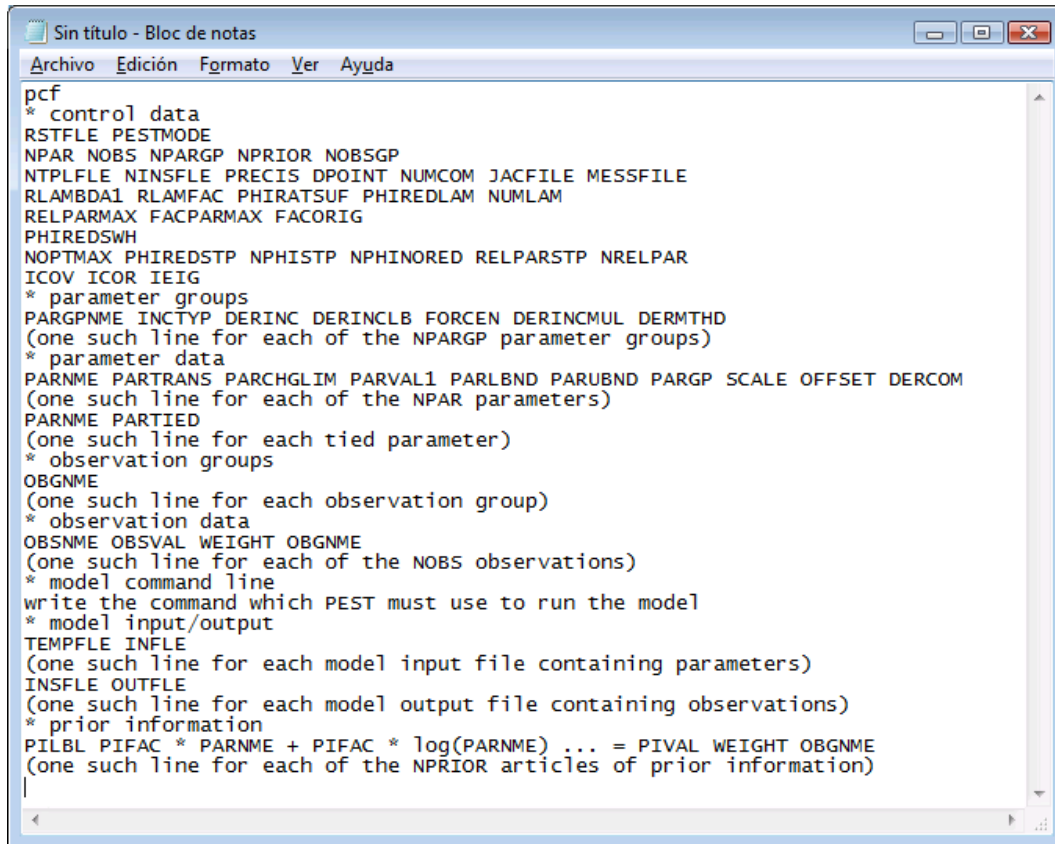
```

ejemplo1.pst - WordPad
Archivo Edición Ver Insertar Formato Ayuda
pcf
* control data
restart estimation
 20 1461 4 0 1
 1 1 single point 1 0 0
 5 2 0.3 0.03 10
 3 3 0.001 0
 0.1
 30 0.01 3 3 0.01 3
 1 1 1
* parameter groups
binf relative 0.01 0 switch 2 parabolic
depthsoil1 relative 0.01 0 switch 2 parabolic
depthsoil2 relative 0.01 0 switch 2 parabolic
depthsoil3 relative 0.01 0 switch 2 parabolic
* parameter data
binf_1 log factor 1.00E-02 1E-10 1 binf 1 0 1
binf_2 log factor 3.00E-02 1E-10 1 binf 1 0 1
binf_3 log factor 5.00E-02 1E-10 1 binf 1 0 1
binf_4 log factor 7.00E-02 1E-10 1 binf 1 0 1
binf_5 log factor 0.1 1E-10 1 binf 1 0 1
Para obtener Ayuda, presione F1

```

Figura 35. Ejemplo de un archivo de control de PEST: ejemplo1.pst

Este archivo tiene el formato que se muestra en la figura 36.



```

pcf
* control data
RSTFLE PESTMODE
NPAR NOBS NPARGP NPRIOR NOBSGP
NTPLFLE NINSFLE PRECIS DPOINT NUMCOM JACFILE MESSFILE
RLAMBDA1 RLAMFAC PHIRATSUF PHIREDLAM NUMLAM
RELPARMAX FACPARMAX FACORIG
PHIREDSWH
NOPTMAX PHIRE DSTP NPHISTP NPHINORED RELPARSTP NRELPAR
ICOV ICOR IEIG
* parameter groups
PARGP NME INCTYP DERINC DERINCLB FORCEN DERINCMUL DERMTHD
(one such line for each of the NPARGP parameter groups)
* parameter data
PARNME PARTRANS PARCHGLIM PARVAL1 PARLBNB PARUBND PARGP SCALE OFFSET DERCOM
(one such line for each of the NPAR parameters)
PARNME PARTIED
(one such line for each tied parameter)
* observation groups
OBSNME
(one such line for each observation group)
* observation data
OBSNME OBSVAL WEIGHT OBSNME
(one such line for each of the NOBS observations)
* model command line
write the command which PEST must use to run the model
* model input/output
TEMPFLE INFLE
(one such line for each model input file containing parameters)
INSFLE OUTFLE
(one such line for each model output file containing observations)
* prior information
PIBL PIFAC * PARNME + PIFAC * log(PARNME) ... = PIVAL WEIGHT OBSNME
(one such line for each of the NPRIOR articles of prior information)

```

Figura 36. Estructura del archivo de control: ejemplo1.pst

Podemos ver que ese archivo tiene muchos parámetros que el usuario puede cambiar. Por defecto *PEST* pone valores a cada uno de esos parámetros.

Vamos a presentar de manera breve cada uno de esos parámetros:

RSTFLE: "restart" o "no restart". Este parámetro permite a *PEST* correr de nuevo si la ejecución se terminó antes.

PESTMODE: "estimation", "prediction" o "regularisation". Es el modo con el cual *PEST* puede correr. En nuestro caso, *PEST* estima parámetros así que elegimos "estimation".

NPAR: Número de parámetros.

NOBS: Número de observaciones.

NPARGP: Número de grupos de parámetros.

NPRIOR: Número de líneas para la información prioritaria.

NOBSGP: Número de grupos de observaciones.

NTPLFLE: Número de archivos de entrada al modelo.

NINSFLE: Número de archivos de instrucciones.

PRECIS: Precisión para el valor de 1 parámetro, "single", 13 caracteres o "doble" 23 caracteres.

DPOINT: "point" o "nopoint" parámetro decimal o no decimal.

NUMCOM, **JACFILE** and **MESSFILE**: Esas variables son usadas por *PEST* para obtener las derivadas directamente del modelo. Para una operación normal se usa 1, 0, 0.

RLAMBDA1: *PEST* trabaja con el método *Gauss Leverberg Marquardt*. Ese método calcula una función objetivo *PHI* con un valor numérico lambda. Después de un primer cálculo de la función objetivo, *PEST* evalúa otra lambda inferior a la lambda inicial, si la función objetivo *PHI* es más pequeña, lambda disminuye y si *PHI* aumenta, lambda también se aumenta en el mismo factor. Se calcula una nueva función objetivo *PHI*, etc. Normalmente se prueban entre 1 y 4 lambda. El valor inicial de lambda puede variar entre 1 y 10. Por defecto esta en 5.

RLAMFAC1: Factor de aumentación de lambda, tiene que ser superior a 1. Por defecto su valor esta en 2. Es el valor con el cual se va a multiplicar o dividir lambda al momento de cambiarlo.

PHIRATSUF: Proporción suficiente de phi para la iteración de la optimización. Si entre 2 cambios de lambda esa proporción de las funciones objetivos es menor a ese valor, *PEST* considera que la iteración actual termina y pasa a la siguiente iteración de la optimización. Por defecto su valor esta en 0.3.

PHIREDLAM: Si el criterio de *PHIRATSUF* no se alcanza, *PEST* necesita saber si debe pasar a la optimización siguiente. Si la reducción de la función objetivo entre 2 lambdas consecutivas es inferior a *PHIREDLAM*, *PEST* considera que puede pasar a la iteración siguiente y puede probar otra lambda. Por defecto este valor esta en 0.01.

NUMLAM: Numero de lambda a probar. Para los casos dónde los parámetros pueden ser muy cercanos a sus límites superiores o inferiores la experiencia ha mostrado que un valor de 10 es más apropiado que 5 porque esto da una oportunidad mayor de ajustar los parámetros. Por defecto se usa un valor de 10.

RELPARMAX: Valor máximo permitido por el cambio "relativo" de un parámetro.

RELPARMAX (o *r*) representa el cambio máximo relativo permitido de un parámetro, *r* debe ser positivo. Si b_0 es el valor de un parámetro "relativo" al principio de una iteración de la optimización, el valor *b* de este mismo parámetro de la siguiente iteración de la optimización será:

$$\frac{|b - b_0|}{b_0} \leq r$$

FACPARMAX: Valor máximo permitido por el cambio "factor". Si la transformación de los parámetros esta en log, el cambio debe ser "factor limited".

FACPARMAX (o f) representa el factor de cambio máximo permitido de un parámetro, f debe ser mayor que 1. Si b_0 es el valor de un parámetro con un "factor" límite al principio de una iteración de la optimización, el valor b de este mismo parámetro de la siguiente iteración de la optimización será:

$$\frac{b_0}{f} \leq b \leq fb_0 \text{ si } b_0 \geq 0$$

$$fb_0 \leq b \leq \frac{b_0}{f} \text{ si } b_0 \leq 0$$

En este caso el parámetro no puede cambiar de signo. Por defecto se usan valores de 3 para cada uno.

FACORIG: Si durante el proceso de optimización un parámetro se encuentra muy pequeño la elección del factor límite o relativo puede obstaculizar severamente el crecimiento del parámetro y volver la convergencia de la función objetivo muy lenta. Si el valor absoluto de un parámetro es inferior a *FACORIG*, entonces se sustituye b_0 en la ecuación relativa por *FACORIG*. Para parámetros factor-limitados, se realiza la misma cosa con la ecuación correspondiente. *FACORIG* no se utiliza para ajustar los límites de parámetros log-transformados. Un valor de 0.001 es bueno. En nuestro caso no se usa porque todos nuestros parámetros son log-transformados.

PHIREDSWH: Si la reducción de la función objetivo entre dos iteraciones de optimización sucesivas es menor a *PHIREDSWH*, PEST cambia su modo de calcular las derivadas a un cálculo de derivada. Este parámetro se usa solamente si la variable *FORCEN* que es el modo de calcular las derivadas es "SWITCH".

NOPTMAX: Número máximo de optimización. Es un criterio de parada para *PEST*. 20 o 30 son valores apropiados.

PHIREDSTP and NPHISTP: Son criterios de parada de *PEST*.

PHIREDSTP: Reducción relativa de phi, indica la convergencia.

$$\frac{\Phi_i - \Phi_{min}}{\Phi_i} \leq PHIREDSTP$$

Φ_i : Función objetivo al fin de la iteración i ,

Φ_{min} : Mínimo de la función objetivo.

NPHISTP: Número de iteración de la optimización donde la relación anterior se cumple. Cuando esta relación se verifica por el número de iteraciones especificadas, *PEST* considera que el proceso está terminado. Valores de 0.005 and 4 son apropiados. El valor de *NPHISTP* no debe ser muy pequeño si los valores de los parámetros pueden ser muy cerca de sus límites.

NPHINORED: Número máximo consecutivo de pruebas para obtener una función objetivo más pequeña. Si *PEST* no puede bajar la función objetivo después de *NPHINORED* operaciones, *PEST* cesa la ejecución. El valor es de 3 o 4.

RELPARSTP: Cambio mínimo para indicar la convergencia.

NRELPAR: Número máximo de iteraciones consecutivas con un cambio de parámetros mínimo. Si el cambio máximo relativo de un parámetro entre diferentes iteraciones de la optimización es menor a *RELPARSTP* después de *NRELPAR* iteraciones sucesivas, *PEST* cesa la ejecución.

ICOV, ICOR and IEIG: Para cada optimización *PEST* escribe un archivo con una matriz covariancia y una matriz de correlación con los vectores propios y valores propios de la matriz de covariancia basados en los valores en curso de la optimización de los parámetros. Esos valores son para decir a *PEST* si el usuario quiere que grave esos datos o no en un archivo.

PARGPNME: Nombre del grupo de parámetro.

INCTYP-DERINC: Esas variables permiten hacer un incremento de parámetro. *INCTYP* tiene diferentes valores, que tienen que ver con un valor dado de *DERINC*:

- **Absolute**: se da el mismo incremento para todos los parámetros del grupo (incrementados con *DERINC*). El incremento se calcula como una fracción del valor del parámetro actual, esta fracción es igual a *DERINC*.
- **Relative**: *DERINC* es multiplicado por el valor absoluto corriente del parámetro para determinar el incremento del parámetro.
- **Rel_to max**: el incremento para cualquier parámetro del grupo se calcula como una fracción del parámetro del grupo con el valor absoluto más alto, esta fracción es igual a *DERINC*.

DERINCLB: Si un incremento del parámetro se calcula cómo "relative" o "rel_to_max", es posible que pueda hacerse demasiado pequeño si el parámetro se hace muy pequeño. Un incremento del parámetro se vuelve "demasiado pequeño" si no permite calcular derivadas confiables con respecto a ese parámetro debido a errores en la diferencia de los valores. Para evitar esa posibilidad, se puede poner un límite inferior sobre los incrementos de parámetro; este límite inferior será el mismo para todos los miembros del grupo, es la variable *DERINCLB*. Si no se desea colocar un límite inferior sobre incrementos de parámetro, *DERINCLB* es igual 0.0. Si *INCTYP* es "absoluto", *DERINCLB* es ignorado.

FORCEN: *FORward/CENtral*. Esta variable determina el método utilizado para el cálculo de las derivadas. Uno de los dos o los dos (delante: "always_2" y central: "always_3"). La mayor parte del tiempo se emplea "switch", significa que *PEST* elige el mejor método.

DERINCMUL: *DERINCMUL* permite aumentar el incremento. Cada vez que el método central se utiliza para el cálculo de derivadas, *DERINCMUL* se multiplica a *DERINC*, así mismo si *INCTYP* es "absolute", "relative" o "rel_to_max", y si *FORCEN* es "always_3" o "switch".

DERMTHD: "parabolic", "best_fit" or "outside_pts" si es el cálculo de derivadas está en "always_3" or "switch" ese parámetro permite saber qué tipo de diferencias 3 puntos usar. Por defecto es *parabolic*.

PARNME: Nombre del parámetro.

PARTRANS: Tipo de transformación de los parámetros:

- "untransformed": sin transformación,
- "log-transformed": transformación logarítmica,
- "fixed": parámetro fijo,
- "tied": parámetro ligado con otro parámetro.

La transformación logarítmica de algunos parámetros puede tener un profundo efecto de estabilidad en el proceso de estimación de parámetros. Si un parámetro es fijo, no tiene ningún papel en el proceso de la optimización. Si un parámetro tiene una relación con otro parámetro, es un parámetro dependiente.

PARCHGLIM: Modo de cambio, factor o relativo.

PARVAL1: Valor inicial del parámetro.

PARLBND: Límite inferior de un parámetro.

PARUBND: Límite superior de un parámetro.

PARGP: Nombre del grupo al cual pertenece el parámetro.

SCALE: Escala.

OFFSET: Desplazamiento. Antes de poner el valor de un parámetro en el modelo, *PEST* lo multiplica por la escala y suma el desplazamiento que le proporciona el usuario.

DERCOM: Mientras se usa la función de derivadas externas a *PEST*, ese valor tiene que ser 1.

OBGNME: Nombre del grupo de observaciones.

OBSNME: Nombre de cada observación.

OBSVAL: Valor de cada observación.

WEIGHT: Peso que se puede asignar a cada observación.

TEMPFLE: Nombre del archivo *template*.

INFLE: Nombre del archivo de entrada del modelo.

INSFLE: Nombre del archivo de instrucciones.

OUTFLE: Nombre del archivo de salida del modelo.

j) Acoplamiento VIC-ROUT con PEST

Para hacer una optimización del modelo *VIC-ROUT* con *PEST*, se tienen que realizar unos programas que ligan los 2 modelos.

PEST debe ser capaz de leer los archivos de salida de *VIC-ROUT* para poder hacer sus comparaciones y proponer otro juego de parámetro, y también debe ser capaz de proponer otro archivo de entrada a *VIC-ROUT*.

Este acoplamiento es el modelo que *PEST* pide dentro del archivo de control.

Los parámetros que vamos a optimizar están principalmente en un solo archivo de entrada de *VIC*: el archivo de suelo.

Las salidas del modelo *VIC-ROUT* son en realidad las salidas del programa *ROUT*, en nuestro caso vamos a usar salidas diarias, archivo llamado *BIFUR.day*, figura 37.

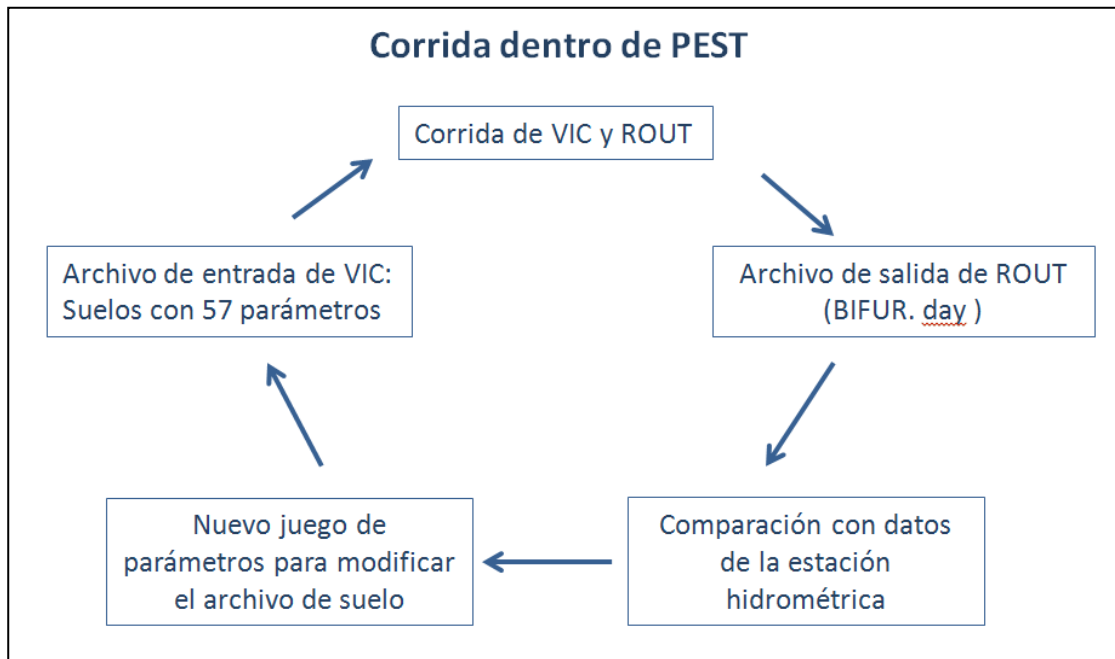
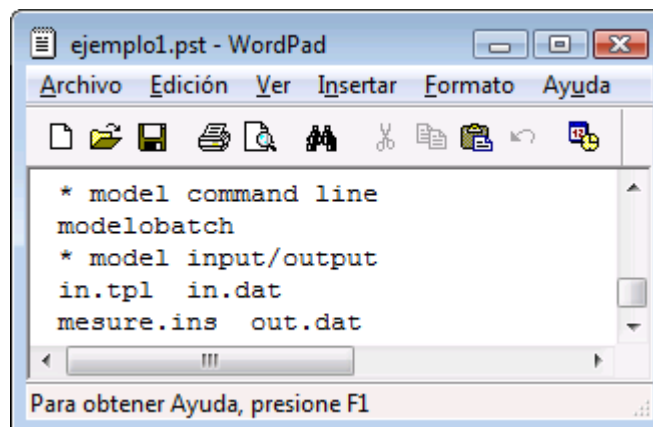


Figura 37. Diagrama de acoplamiento VIC-ROUT con PEST.

Para lograr esto se necesitan 2 algoritmos (2 programas realizados dentro del ambiente UNIX).

El primer programa lee las salidas de ROUT (BIFUR.day) y crea un archivo llamado out.dat; archivo proporcionado como archivo de salida dentro del archivo de control de PEST, figura 38.

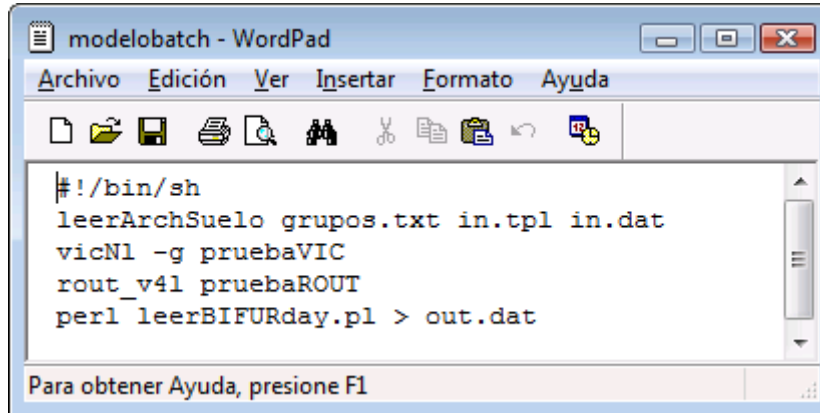


```

ejemplo1.pst - WordPad
Archivo Edición Ver Insertar Formato Ayuda
* model command line
modelobatch
* model input/output
in.tpl in.dat
mesure.ins out.dat
Para obtener Ayuda, presione F1
  
```

Figura 38. Parte del modelo a optimizar dentro del archivo de control de PEST con salidas y entradas

En este caso los archivos `in.tpl` y `measure.ins` son archivos fijos que contienen un formato de lectura. El archivo `out.dat` es un archivo que tiene que dar el modelo llamado aquí "*modelobatch*" y *PEST* proporciona a ese modelo un archivo llamado *in.dat*, figura 39.



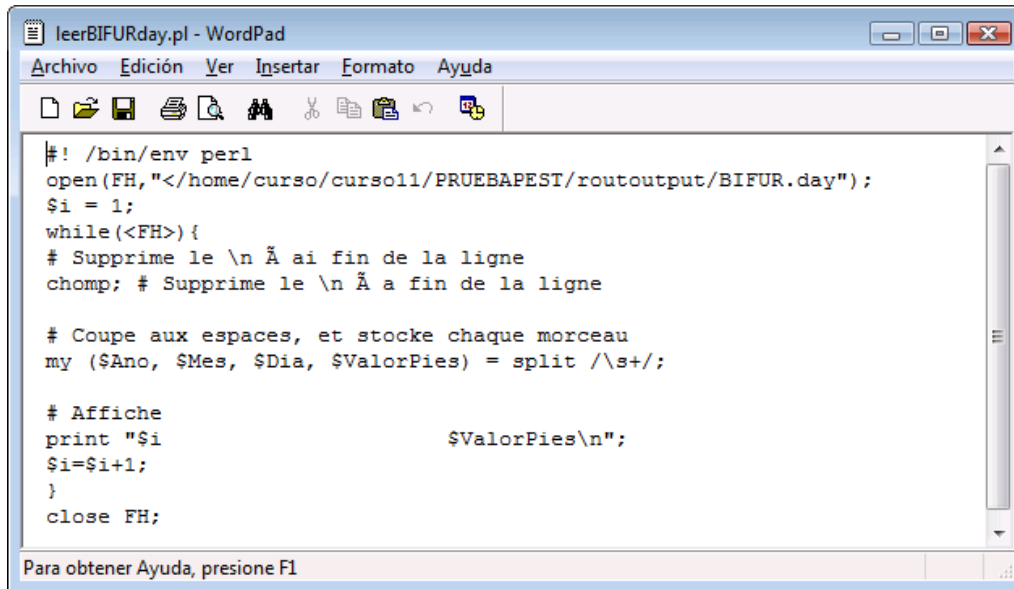
```
#!/bin/sh
leerArchSuelo grupos.txt in.tpl in.dat
vicN1 -g pruebaVIC
rout_v41 pruebaROUT
perl leerBIFURday.pl > out.dat
```

Figura 39. Modelo batch: modelo general que permite hacer la liga entre VIC-ROUT y PEST

Este modelo contiene 4 partes:

- Lectura de parámetros (que van a ser las salidas de *PEST*) y modificación dentro del archivo de entrada de *VIC* (en este caso el archivo de suelo). Este programa se llama leer *ArchSuelo* y pide 3 archivos de entrada
- Corrida de *VIC*
- Corrida de *ROUT* leyendo los datos de salida de *VIC*
- Lectura del archivo de salida de *ROUT* y creación de un archivo que lee *PEST* para comparar con las observaciones (*leerBIFURday*)

En este modelo general vemos los 2 programas que se realizaron: *leerArchSuelo* y *leerBIFURday*. El primero se hizo en *PERL* y el segundo en *awk*, figuras 40 y 41.



```

#! /bin/env perl
open(FH, "</home/curso/cursol1/PRUEBAPEST/routoutput/BIFUR.day");
$i = 1;
while(<FH>){
# Supprime le \n Ã ai fin de la ligne
chomp; # Supprime le \n Ã a fin de la ligne

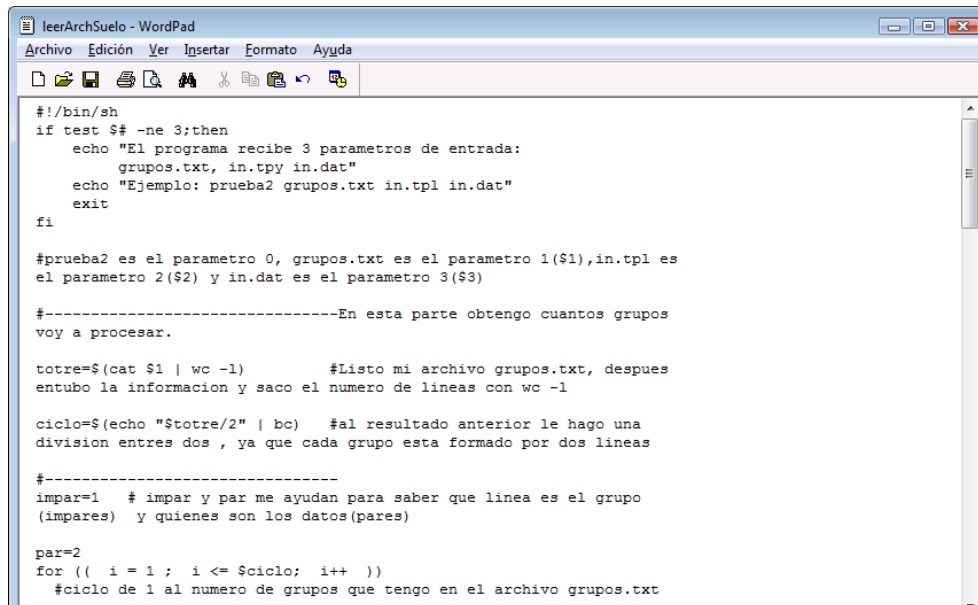
# Coupe aux espaces, et stocke chaque morceau
my ($Año, $Mes, $Dia, $ValorPies) = split /\s+/;

# Affiche
print "$i                $ValorPies\n";
$i=$i+1;
}
close FH;

```

Para obtener Ayuda, presione F1

Figura 40. Programa en PERL leerBIFURday.pl



```

#!/bin/sh
if test $# -ne 3;then
    echo "El programa recibe 3 parametros de entrada:
    grupos.txt, in.tpy in.dat"
    echo "Ejemplo: prueba2 grupos.txt in.tpl in.dat"
    exit
fi

#prueba2 es el parametro 0, grupos.txt es el parametro 1($1),in.tpl es
el parametro 2($2) y in.dat es el parametro 3($3)

#-----En esta parte obtengo cuantos grupos
voy a procesar.

totre=$(cat $1 | wc -l)          #Listo mi archivo grupos.txt, despues
entubo la informacion y saco el numero de lineas con wc -l

ciclo=$(echo "$totre/2" | bc)    #al resultado anterior le hago una
division entres dos , ya que cada grupo esta formado por dos lineas

#-----
impar=1 # impar y par me ayudan para saber que linea es el grupo
(impares) y quienes son los datos(pares)

par=2
for (( i = 1 ; i <= $ciclo; i++ ))
    #ciclo de 1 al numero de grupos que tengo en el archivo grupos.txt

```



```
do
grupo=$(cat $1 | sed -n $impar'p'|awk '{split($1,a,"");print a[2]}')

#obtengo el numero de grupo, si es 1, 2, 3...etc.
#listo grupos.txt con cat $1
#obtengo la linea donde esta grupo de acuerdo al valor de impar con
sed -n linea'p', en este caso serian 1,3,5,7,9
#de la linea obtengo la primera columna($1) que en nuestro caso
tendra grupo_1, divido la linea cada vez que encuentre el caracter
"_" y lo meto en una variable a que se comporta como un arreglo
obtengo a en su posicion 2 que es donde se almacena el numero,
porque a[1] almacena la palabra grupo.

linea=$(awk '{for (j=1; j<=NF; j++) split($2,a,"");if (a[2]
=="$grupo') print NR_"a[1]'} $2)

#linea almacena el o los numeros de linea donde se encuentra el
grupo que estoy buscando en el archivo in.tpl y le agrega la
variable que se ca a tratar, en este ejemplo, me dira que 1 se
encuentra en in.tpl en las lineas 4_binf,9_DepthSoil1,1
_DepthSoil2,19_DepthSoil3.

for r in ${linea[@]}
#Realizo un ciclo para tratar cada uno de los datos guardado en
linea(ejemplo:4_binf,9_DepthSoil1,14_DepthSoil2,19_DepthSoil3)
do
l=$(echo $r|awk '{split($1,a,"");print a[1]}')

#toma el valor de r, dividiendolo por el "_", y asigna a l,
el valor de la linea. (de 4_binf, se queda con el 4)

posdat=$(echo "$l-1" | bc)
# al valor de l, le resto 1, y esta es
la linea que me va a dar el valor en in.dat, en este caso
posdat valdria 3 y lo hago de esta manera

var=$(echo $r|awk '{split($1,a,"");print a[2]}')
#en el caso de l me traje el numero de linea, en este caso
var almacena la variable, binf, DepthSoil1.

val=$(cat $3 | sed -n $posdat'p' | awk '{print $1}')
#me traigo el valor que tenga la linea correspondiente a
posdat en el archivo in.dat

case $var in
#de acuerdo a val, v almacenara en valor de que columna va a
ser modificada.
"De") v=6;;
"Dsmax") v=7;;
"Wa") v=8;;
"binf") v=5;;
"DepthSoil1") v=23;;
"DepthSoil2") v=24;;
"DepthSoil3") v=25;;
esac

cadena=$(cat $1 | sed -n $par'p')
#cadena va almacenar todos los valores del grupo(las lineas
pares, en este caso una por una las lineas 2,4,6,8,10)

for n in ${cadena[@]}
#para cada uno de los grupos almacenados en cadena se va a
realizar el cambio de valor de acuerdo a la columna.

do
awk '{for (j=1; j<=NF; j++) if (NR=="$n") {$v="$val";
print }}' essaiBi00101m011.txt > kk
#lee de el archivo de suelo de la linea l hasta
el final y checa, si la linea en que va es la que
corresponde a cada uno de los valores del grupo, si es
asi, modifica la columna que corresponda de acuerdo al
valor de v y actualiza con el valor de val, una vez
hecho esto, vacia linea por linea a kk.....

mv kk essaiBi00101m011.txt
#cambio el nombre de kk por el del nombre del archivo
de suelo para no perder los cambios

done #termina do n in $@

done #termina do de r in ${linea[@]}

impar=$(echo "$par+1" | bc)
#incremento impar para empezar a jugar con el otro grupo

par=$(echo "$impar+1" | bc)
#incremento par para tomar los numeros correspondientes al valor
del grupo

done # termina do i<=$ciclo

Para obtener Ayuda, presione F1
```

Figura 41. Programa en *awk* LeerArchSuelo

En esta parte solamente se presenta de manera general estos programas, en la parte siguiente vamos a detallar cada uno de los archivos de entrada salida de esos programas al momento de explicar cómo realizar una optimización.

k) Optimización de VIC-ROUT con PEST

Una vez realizados los archivos de entrada de *PEST* así como los modelos para hacer la liga entre *PEST* y *VIC-ROUT*, se puede hacer una optimización.

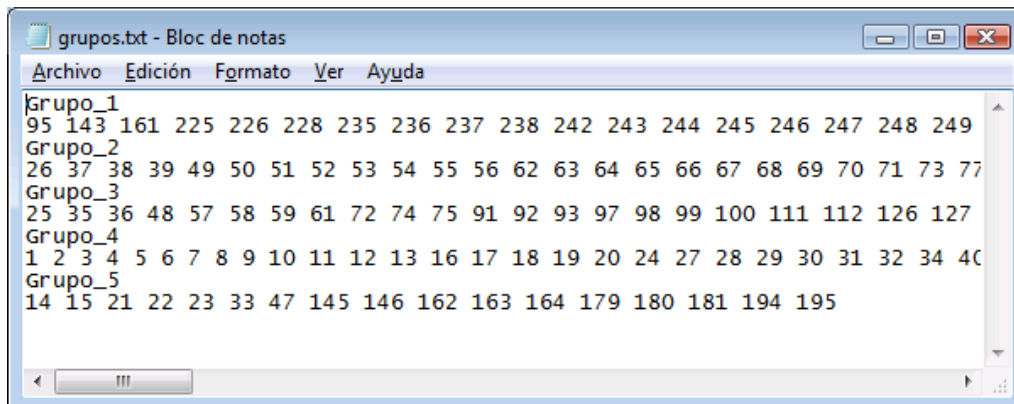
Primero, debemos de realizar unas corridas de *VIC* previas para preparar el modelo. Eso permitirá lograr mejores resultados con *PEST*, porque de esa manera puede encontrar un mínimo más rápidamente de su función objetivo y también evita que caiga en un mínimo local y no logre encontrar un juego de parámetros.

l) Pasos para realizar una corrida de PEST

Debido a la doble utilización del sistema Windows y Unix, la preparación de una corrida de *PEST* suela ser un poco laboriosa. En esta parte vamos a detallar todo lo que tiene que hacer el usuario para realizar una optimización con *PEST*.

Crear un archivo de grupos

Primeramente el usuario debe tener un archivo que define los diferentes grupos de celdas que va a usar. Este archivo contiene el nombre del grupo así como las celdas contenidas en este grupo, figura 42.



```
grupos.txt - Bloc de notas
Archivo Edición Formato Ver Ayuda
Grupo_1
95 143 161 225 226 228 235 236 237 238 242 243 244 245 246 247 248 249
Grupo_2
26 37 38 39 49 50 51 52 53 54 55 56 62 63 64 65 66 67 68 69 70 71 73 77
Grupo_3
25 35 36 48 57 58 59 61 72 74 75 91 92 93 97 98 99 100 111 112 126 127
Grupo_4
1 2 3 4 5 6 7 8 9 10 11 12 13 16 17 18 19 20 24 27 28 29 30 31 32 34 40
Grupo_5
14 15 21 22 23 33 47 145 146 162 163 164 179 180 181 194 195
```

Figura 42. Ejemplo de un archivo de grupos

En este ejemplo tenemos 5 grupos y para cada uno una fila que define cuales son las celdas que pertenecen al mismo.

Una vez creado este archivo en Windows se pasa a Unix dentro de la carpeta de PEST.

Crear un archivo con datos de la estación hidrométrica

La interfaz que crea los archivos de entrada para PEST necesita un archivo que contiene los datos (en m^3/s) correspondientes al periodo de simulación de la estación hidrométrica contra la cual se va a comparar.

En nuestro caso vamos a considerar el periodo 1960 – 1963, el formato del archivo se muestra en la figura 43.

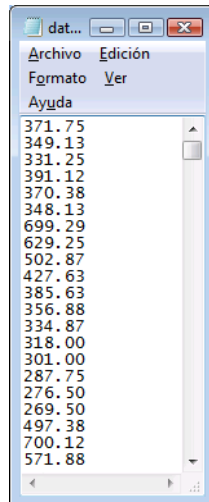
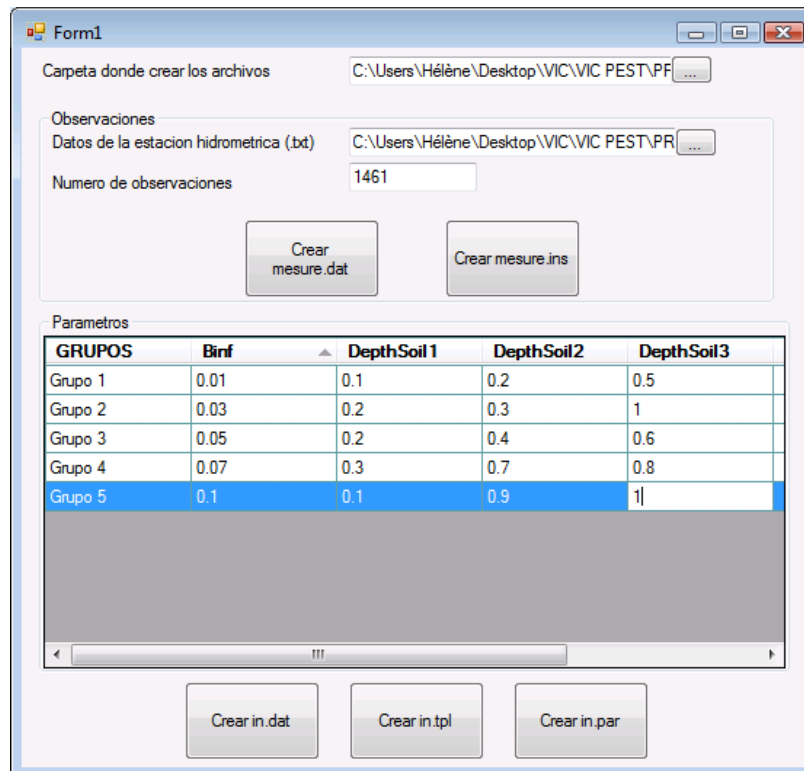


Figura 43. *Ejemplo de un archivo de datos de la estación hidrométrica*

Crear los archivos de entrada de PEST

Con la ayuda de la interfaz vamos a crear los archivos de entrada de PEST. El usuario tiene que conocer el número de observaciones, dar el archivo de datos y llenar los parámetros de entrada que desea optimizar, figura 44.



Form1

Carpeta donde crear los archivos C:\Users\Hélène\Desktop\VIC\VIC PEST\PF...

Observaciones

Datos de la estación hidrometrica (.txt) C:\Users\Hélène\Desktop\VIC\VIC PEST\PR...

Numero de observaciones 1461

Crear measure.dat Crear mesure.ins

Parametros

GRUPOS	Binf	DepthSoil1	DepthSoil2	DepthSoil3
Grupo 1	0.01	0.1	0.2	0.5
Grupo 2	0.03	0.2	0.3	1
Grupo 3	0.05	0.2	0.4	0.6
Grupo 4	0.07	0.3	0.7	0.8
Grupo 5	0.1	0.1	0.9	1

Crear in.dat Crear in.tpl Crear in.par

Figura 44. Ejemplo para llenar la interfaz de creación de archivos de entrada de PEST

Esto nos permite tener 5 archivos *measure.dat*, *measure.ins*, *in.dat*, *in.tpl*, *in.par* que vamos a pasar a UNIX dentro de la carpeta de PEST.

Dentro de *UNIX* tenemos que copiar los archivos en la carpeta que contiene todas las funciones de *PEST* para poder verificar estos archivos (si tenemos un error hay que convertir los archivos con *dos2unix*).

```
Tempchek in.tpl
Inshek mesure.ins
Tempchek in.tpl modfile in.par
Inshek mesure.ins mesure.obf
```

Para crear los archivos *measure.obf* e *in.pmt*, se crea un primer archivo de control de PEST con la función siguiente:

```
Pestgen ejemplo1 mesure.obf in.par
```

Ya tenemos el archivo de control *ejemplo1.pst* que vamos a tener que modificar. Se pasa a Windows para modificarlos parámetros necesarios.

Modificación del archivo de control de PEST

Se cambian varios valores dentro del archivo de control, inicialmente los límites de los parámetros.

- Límites inferiores: B_{inf} , $DepthSoil_1$, $DepthSoil_2$, $DepthSoil_3 = 0.0000001$
- Límites superiores: $B_{inf} = 1$, $DepthSoil_1$, $DepthSoil_2$, $DepthSoil_3 = 3$

Cambiamos la transformación de cada parámetro a logaritmo porque eso nos permite tener mejores resultados (de "none" a "log").

Se hacen solamente 4 grupos de parámetros: B_{inf} , $DepthSoil_1$, $DepthSoil_2$ y $DepthSoil_3$ para agrupar los parámetros por tipo. Eso permite hacer el cálculo de las derivadas más rápidamente.

Se cambia el nombre del modelo a *modelobatch* y los nombres de parámetros *in.tpl* *in.dat* para la entrada y *measure.ins* y *out.dat* para la salida.

También se pueden cambiar parámetros relativos a PEST directamente si se necesita, Una vez cambiados esos valores se pasa el archivo a UNIX y se verifica con:

pestchek ejemplo1.pst

Corrida de PEST

Antes de hacer una corrida se verifica lo siguiente: las salidas de VIC tienen que ser las entradas para el modelo ROUT.

En el modelo *modelobatch* tenemos que leer las salidas del modelo ROUT. Todos los resultados de ROUT son en ft^3/s , así se calcula la diferencia de la función objetivo.

Para correr PEST, necesitamos tener los archivos siguientes dentro de la carpeta de PEST (que contiene las funciones relativas a PEST):

- In.dat, in.tpl, in.par, in.pmt
- Measure.dat, measure.ins, measure.obf
- Ejemplo1.pst
- Programas de entrada y salida de PEST : LeerBIFURday, LeerArchSuelo, modelobatch
- Los archivos de control de VIC y ROUT

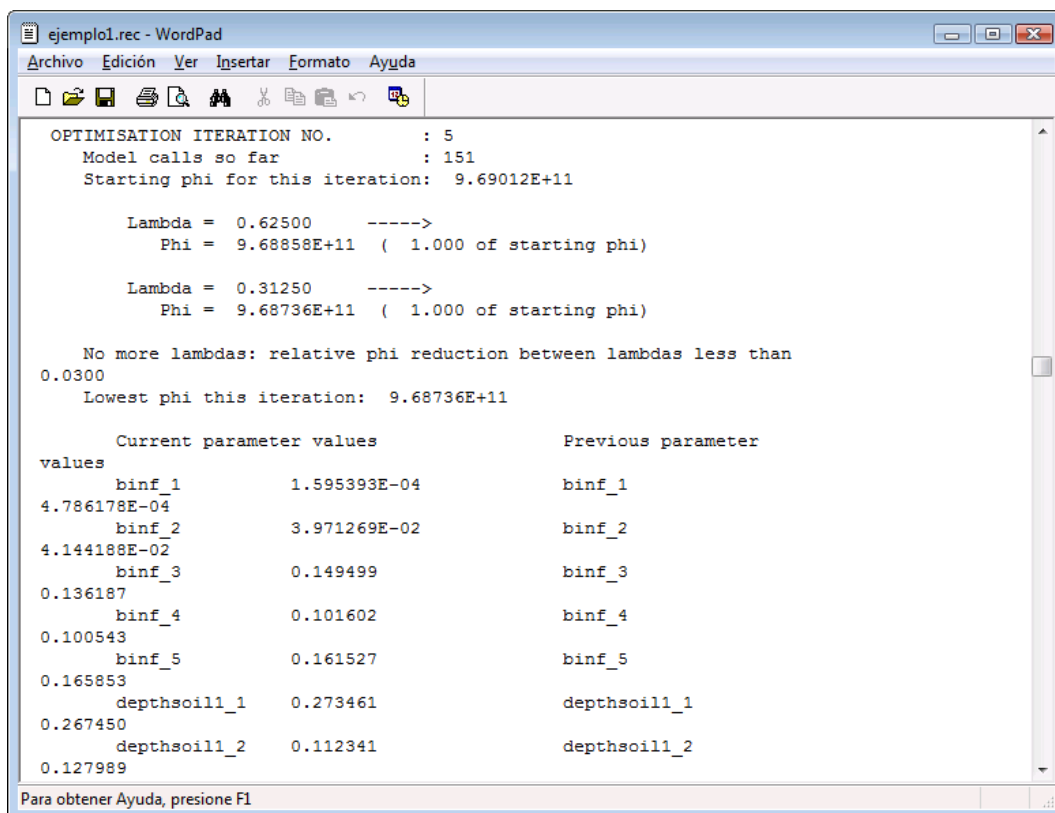
Ya teniendo todo listo se corre PEST: *pest ejemplo1.pst*

m) Salidas de PEST y resultados

Se crean varios archivos de salida de *PEST* teniendo el mismo nombre ejemplo1. Esos archivos nos permiten ver la evolución de la optimización como el archivo *.rec* que contiene un resumen de la optimización con los valores de entrada, el cálculo de la función objetivo a cada paso de la optimización y el número de llamadas al modelo, así como los cambios en los parámetros.

Al final de la simulación este archivo proporciona una parte con los resultados propios de la simulación: el juego final de parámetros, las diferencias entre los datos observados y simulados.

Este archivo contiene todos los resultados de la simulación, y cada uno está detallado en otros archivos como el desglose de los parámetros al transcurso de las optimizaciones, figura 45.



```

ejemplo1.rec - WordPad
Archivo Edición Ver Insertar Formato Ayuda

OPTIMISATION ITERATION NO.      : 5
Model calls so far              : 151
Starting phi for this iteration: 9.69012E+11

    Lambda = 0.62500    ----->
    Phi = 9.68858E+11   ( 1.000 of starting phi)

    Lambda = 0.31250    ----->
    Phi = 9.68736E+11   ( 1.000 of starting phi)

No more lambdas: relative phi reduction between lambdas less than
0.0300
Lowest phi this iteration: 9.68736E+11

Current parameter values      Previous parameter
values
    binf_1          1.595393E-04    binf_1
4.786178E-04
    binf_2          3.971269E-02    binf_2
4.144188E-02
    binf_3          0.149499        binf_3
0.136187
    binf_4          0.101602        binf_4
0.100543
    binf_5          0.161527        binf_5
0.165853
    depthsoil1_1    0.273461        depthsoil1_1
0.267450
    depthsoil1_2    0.112341        depthsoil1_2
0.127989

Para obtener Ayuda, presione F1

```

Figura 45. Ejemplo de un archivo de salida de *PEST .rec*

El último juego de parámetros se encuentra entonces en el archivo *.rec*. En nuestro caso *PEST* logro obtener los parámetros que se muestran en la figura 46.

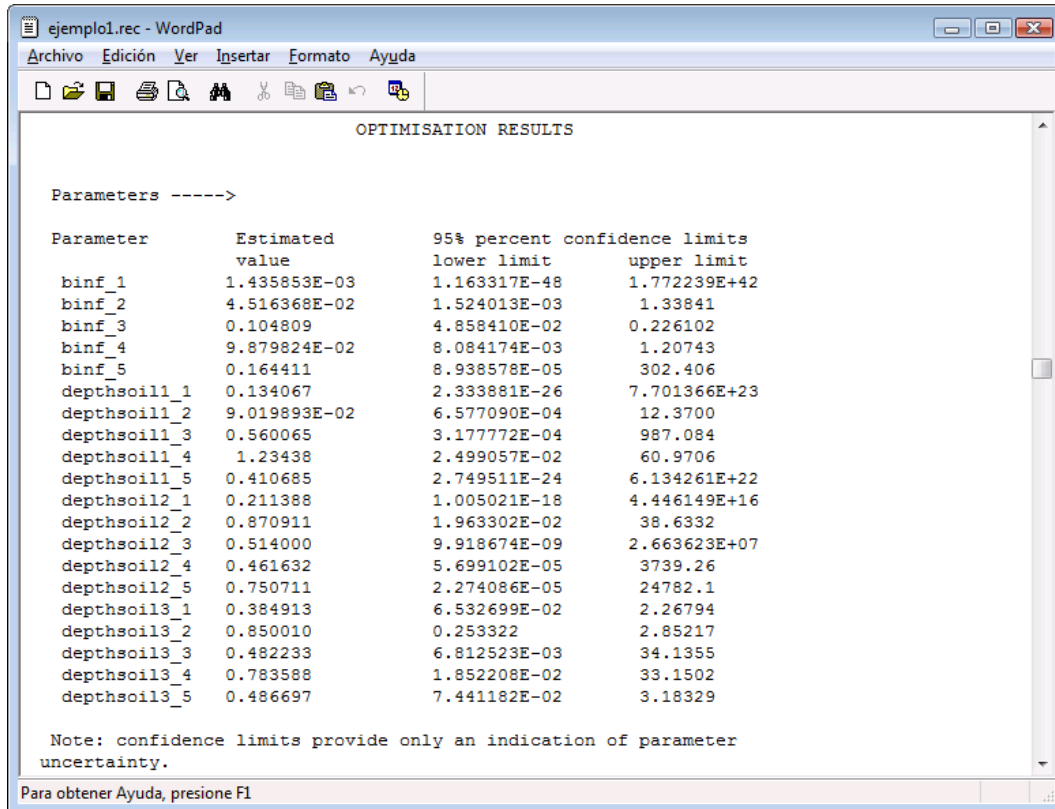


Figura 46. Ejemplo Resultados: juego de parámetros obtenido por PEST

Juego de parámetros inicial:

	B_{inf}	DepthSoil ₁	DepthSoil ₂	DepthSoil ₃
Grupo 1	0.01	0.1	0.2	0.5
Grupo 2	0.03	0.2	0.3	1
Grupo 3	0.05	0.2	0.4	0.6
Grupo 4	0.07	0.3	0.7	0.8
Grupo 5	0.1	0.1	0.9	1.0

Juego de parámetros después de una optimización:

	B_{inf}	DepthSoil ₁	DepthSoil ₂	DepthSoil ₃
Grupo 1	1.435853E-03	0.134067	0.211388	0.384913
Grupo 2	4.516368E-02	9.019893E-02	0.870911	0.850010
Grupo 3	0.104809	0.560065	0.514000	0.482233
Grupo 4	9.879824E-02	1.23438	0.461632	0.783588
Grupo 5	0.164411	0.410685	0.750711	0.486697

El archivo de salida de *ROUT: BIFFUR.day* contiene los resultados de la última simulación. Éste se copia a Windows y podemos realizar una gráfica de los resultados de la simulación así como calcular el *Nash Sutcliffe*.

Esta optimización nos permite tener un *Nash Sutcliffe* de 0.74, mejorando el acercamiento entre los datos históricos y la simulación, figura 47.

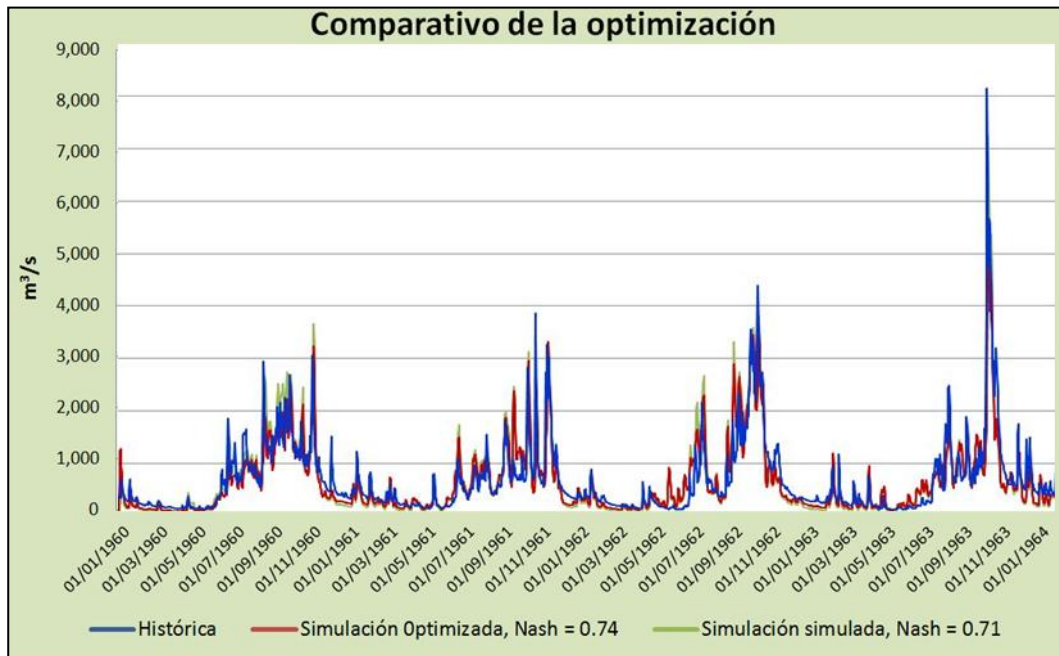


Figura 47. Comparación con la serie histórica y las simulaciones antes y después de PEST

4. Comparación de resultados y validación del modelo

El modelo calibrado (NASH = 0.74) se sometió a un proceso de validación. El cual consistió en realizar una simulación y compararla con registros recientes de escurrimientos naturales, debido a que estas condiciones no se tienen en la zona, se procedió a comparar las simulaciones en dos periodos de tiempo: uno de diez años antes del periodo utilizado en la calibración y otro de tres años recientes utilizando un hidrograma transitado.

4.1 Validación VIC con periodo de 10 años

La validación del modelo se realizó simulando un periodo de 10 años de 1950 a 1960, previos al periodo de calibración (1960-1964), Los resultados obtenidos tienen un valor de NASH de 0.73, (Figura 48).

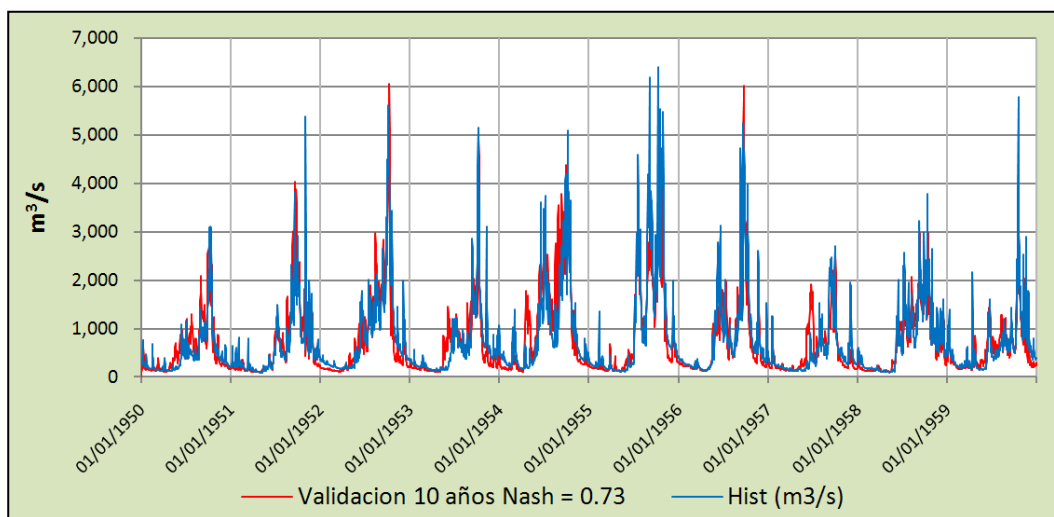


Figura 48. Validación con un periodo anterior de 10 años: 1950-1960

4.2 Simulación de un periodo reciente de 3 años

Para la simulación con registros relativamente recientes, se utilizó el hidrograma transitado descrito en el inciso 1.5.2, el cual se comparó con una simulación de tres años comprendidos de 2006 a 2008, periodo en el cual se dieron eventos extremos en

la zona en estudio. El resultado es un valor de NASH de 0.42, como se muestra en la figura 49.

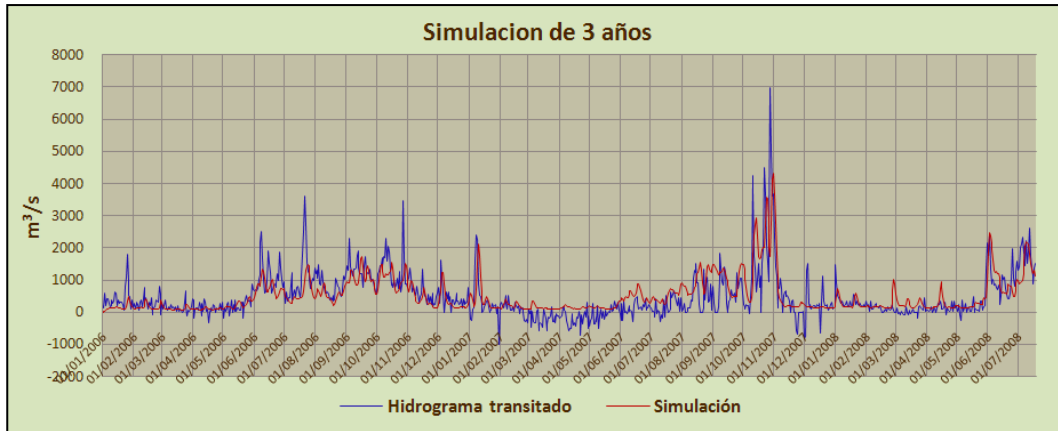


Figura 49. Comparativo de la serie de calibración con el hidrograma transitado: 2006–2008

5. Modelos acoplados

En general, la modelación juega un papel muy importante en el estudio del sistema terrestre, resuelve preguntas específicas, sirve como potente herramienta de diagnóstico, sugiere mediciones de campo, evalúa el comportamiento de un sistema ante posibles escenarios y es útil para la predicción de condiciones futuras. Un modelo detallado de interacción suelo-atmósfera debe lograr vincular los procesos hidroclimáticos con la cobertura vegetal y las características del suelo.

El acoplamiento MM5-VIC lo realizó Liang et al. (2005) de manera similar a la utilizada en el acoplamiento MM5-OSU por Chen y Dudhia (2001). El acoplado de estos modelos (MM5-VIC) se logra mediante el uso de VIC como esquema de superficie terrestre por MM5, lo cual permite una definición a nivel sub-celda de la variabilidad de las condiciones superficiales. El modelo desarrollado por la Universidad de Washington permite que VIC establezca primeramente las condiciones iniciales sujetas a parámetros como vegetación, humedad de suelo y nieve generando los flujos para momento, calor y humedad correspondientes a las condiciones iniciales en la frontera más baja de MM5.

Para correr el modelo acoplado MM5-VIC se corre los pasos del 1 al 5 referenciados en la sección 2 de este informe, más lo siguiente:

6.-MM5-VIC (Multiprocesamiento)

Se realizan los mismos pasos que cuando se corre el punto 5(MM5), y se realizan una serie de modificaciones a dos subrutinas que se encuentran en **MM5/MPP/build/**, las cuales son:

1. **RDINIT.F** Inicializa los parámetros y modelos de estado
2. **VICSURF.C** Drivers for VIC model

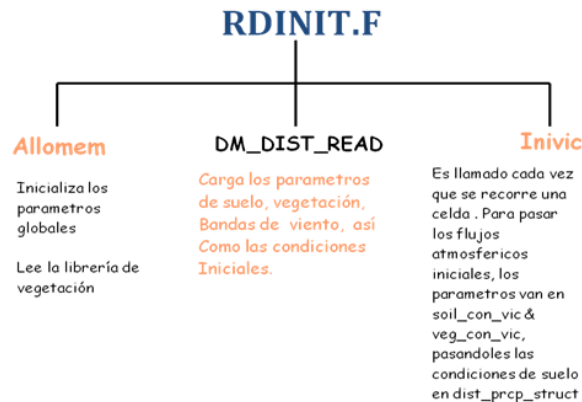


Fig. 50 Modificaciones al archivo RDINIT.F

Teniendo como resultado para cada una de las salidas correspondientes a los eventos corridos: MMOUT_DOMAIN1 y MMOUT_DOMAIN2; los números solo indica a los eventos. Posteriormente con el MM5toGrADS se pueden leer los resultados y generar en Excel las gráficas que ilustran los resultados (figura 51 y 52) ilustrando la comparación entre lo generado por el modelo acoplado (mm5-vic); modelo hidrológico (vic) y las observaciones.

Se puede apreciar que para el evento del 25 de octubre al 20 de noviembre de 2007 (figura 51) el modelo vic coincide con las observaciones, sin embargo, el modelo acoplado muestra como un retraso en la señal (defasamiento) pero aún continúa mostrando una sobre estimación respecto a lo registrado. Esto podría ser causado por el modelo atmosférico ya que se ha demostrado en otros estudios que ante eventos extremos pudiera estar subestimando o sobrestimando este modelo.

Para el evento del 04 al 29 de septiembre de 1998 (figura 52) ninguno de los modelos coincide con los observados, sin embargo, el modelo hidrológico representa la tendencia de lo mostrado en los observados aunque con variación en la intensidad pero el modelo mm5-vic no representada nada en este evento.

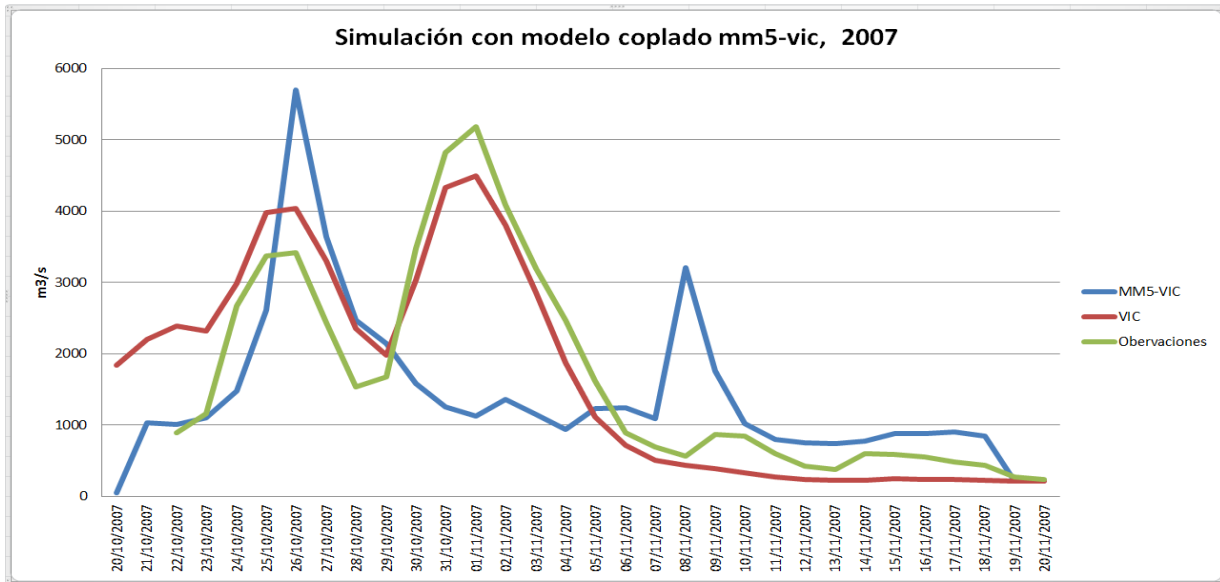


Figura 51.- Comparación de resultados entre el modelo acoplado mm5-vic, modelo hidrológico vic y observaciones del periodo 25 octubre al 20 de noviembre 2007.

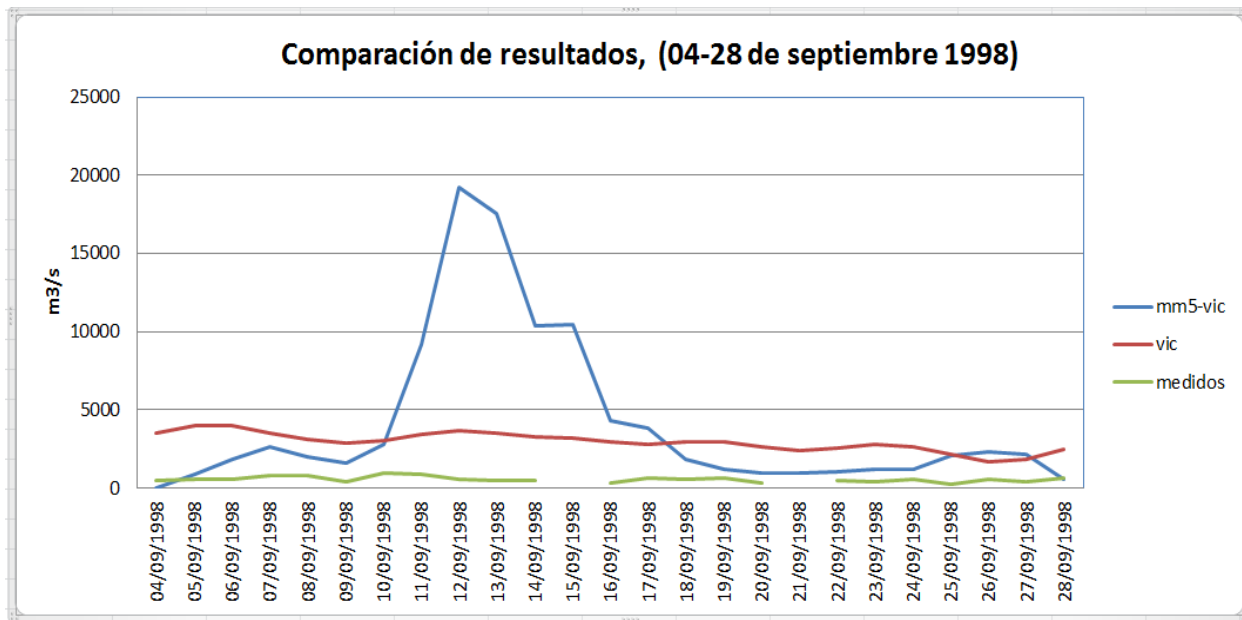


Figura 52.- Comparación de resultados entre el modelo acoplado mm5-vic, modelo hidrológico vic y observaciones del periodo 04al 28 de septiembre de 1998.

6. Conclusiones y recomendaciones

6.1 Conclusiones

Se comprobó la efectividad una herramienta que utiliza las variables físicas medidas espacialmente y el pronóstico meteorológico que complementan series de tiempo de datos de precipitación pluvial procesados mediante modelos matemáticos que simulan los procesos hidrológicos y de lluvia-escurrimiento.

El modelo del sistema hidroeléctrico del río Grijalva, una vez calibrado (1960-1965) se validó simulando un periodo de 10 años previo al de calibración. Los resultados obtenidos en el proceso de validación fueron aceptables (Nash=0.73). Posteriormente, se simuló un periodo reciente de tres años (2006-2008) el cual se comparó con una serie hidrométrica transitada en donde se obtuvieron resultados que subestiman el escurrimiento (Nash=0.42), sin embargo se aprecia que representa muy bien los picos del hidrograma es decir tiene una buena respuesta a las avenidas máximas. Cabe hacer mención que existe una diferencia de 45 años entre las condiciones fisiográficas con las que fue calibrado el modelo y las existentes en el periodo que se simuló.

Por otro lado, el modelo VIC debido a que es un modelo semidistribuido permite simular cuencas muy grandes del orden de varias decenas de miles de kilómetros cuadrados con un tiempo de simulación relativamente corto. Desde el punto de vista de su funcionalidad, es un modelo que carece de una interfaz amigable y aún se tienen que ligar archivos en forma manual en donde se requiere un cierto grado de *expertis* de la estructura de los algoritmos por parte del usuario.

Respecto al método de optimización PEST, éste logra mayor precisión en los valores de los parámetros de calibración, obteniendo buenos resultados en la misma.

Se observó que el pronóstico obtenido con el modelo acoplado MM5-VIC y la comparación con hidrogramas naturales medidos en la salida de la cuenca es razonablemente preciso, sin embargo, cuando se usa solo el modelo hidrológico se observan muy buenos resultado causando ya conflicto cuando se agrega el modelo atmosférico como se mostró en los resultados, debido a que el modelo mm5 en ocasiones cuando se encuentra con eventos extremos subestima o sobrestima, se podrían hacer más ejercicios para encontrar si es que existiera un patrón.

6.2 Recomendaciones

El modelo hidrológico (VIC) simula escurrimientos vírgenes, y es aplicable a cuencas que no tienen infraestructura hidráulica, si se aplica a cuencas con este tipo de infraestructura los resultados deben considerarse con mucha reserva. Si las cuencas son del orden de miles de kilómetros cuadrados es recomendable aplicar un modelo distribuido.

En el caso del modelo acoplado (MM5-VIC) se recomienda que se calibre el modelo atmosférico para ver si mejoran los resultados además de hacer una corrida mucho más largas, es decir, tener un mes de arranque del modelo para ver si llega a estabilizarse cuando se llegue a la fecha del evento de interés.

4 Referencias

- 1.1 Environmental Systems Research Institute, ESRI. Software Geográfico, ArcGis, ArcView V.9. USA 2005.
- 1.2 USGS. Modelos Digitales de Elevación del *Shuttle Radar Topography Mission*, USA, 2005.
- 1.3 INEGI. Continuo de Elevaciones Mexicano escala 1:50,000. México, 2005.
- 1.4 INEGI. Carta Hidrológica de Aguas Superficiales, escala 1:520,000 México.
- 1.5 INEGI. Conjunto de datos vectoriales escala 1:250,000. Discos Compactos. México 2003.
- 1.6 IMTA. Banco Nacional de Datos. BANDAS. México 2000.
- 1.7 Banco Nacional de Datos de Aguas Superficiales (Bandas) Versión actualizada a 2002. Instituto Mexicano de Tecnología del Agua (IMTA). Jiutepec, Mor.
- 1.8 Extracción rápido de información climatológica (Eric III) Versión 2.0. Instituto Mexicano de Tecnología del Agua (IMTA). Jiutepec, Mor.

5 Anexos

1. Listado de Municipios.
2. Estaciones hidrométricas
3. Estaciones meteorológicas
4. Tránsito de avenidas
5. Resultados gráficos de las corridas de periodos más largos