

SIMUSOL: EXTENSIONES, TABLAS EXTERNAS Y DINÁMICAS, SERIES SINTÉTICAS DE RADIACIÓN BASADAS EN PROCESOS ESTOCÁSTICOS CON CADENAS DE MÁRKOV, Y EN MÉTODOS AUTOREGRESIVOS GAUSSIANOS

Diego Saravia.

Depto. De Física. Universidad Nacional de Salta. Av. Bolivia 5150. CP 4400,
Salta, Argentina Fax : 54-387-4255489, E-mail: diego.saravia@gmail.com

Recibido 09/08/18, aceptado 24/09/18

RESUMEN: Se presentan avances en el software SIMUSOL/SCEPTRE que permiten el uso de módulos o EXTENSIONES desarrolladas en lenguaje c. Junto a dos de ellas: Tabular, que lee tablas desde archivos de texto, durante el funcionamiento del Sceptre, careciendo de límites de tamaño prefijados y Radia, que calcula series sintéticas de radiación. Radia incorpora diversos métodos de días de cielo despejado, y métodos estocásticos, uno basado en matrices de Markov, otro autoregresivo gaussiano. Combinados generan series de radiación para un intervalo de tiempo arbitrario con resolución horaria.

Palabras clave: Sceptre, Simusol, Radiación, Markov, Tabular

INTRODUCCIÓN.

Simusol (Alía D. et al., 2002) ha sido usado en numerosos trabajos de investigación y desarrollo. Era necesario habilitar la incorporación de funciones mediante módulos o EXTENSIONES que no requieran modificar el código de Simusol, ni del Sceptre. De ésta forma cualquier desarrollador puede aportar código en forma independiente.

Estas EXTENSIONES pueden redactarse en cualquier lenguaje en el que finalmente las funciones se compilen contra un programa Fortran (The IBM Mathematical Formula Translating System). Lo más práctico es hacer una librería en c. También es conveniente dotar a ésta librería de una interfaz Perl y ejemplos de uso ejecutables en éstos lenguajes, independientemente de Simusol.

En éste trabajo presentamos el mecanismo para que Simusol trabaje con EXTENSIONES y dos de las mismas: Tabular y Radia, junto con sus funciones.

Simusol necesitaba una forma de incorporar grandes tablas, en particular para calcular series temporales de radiación para períodos de tiempo largos.

También era necesario facilitar el uso de diagramas con radiación. Radia incluye algoritmos para calcular con precisión la posición del Sol, radiación de cielo despejado, métodos estocásticos para calcular la radiación diaria y horaria y la absorción sobre una superficie inclinada con cubierta. En el futuro se irán incorporando a éste módulo otras funciones presentadas en trabajos anteriores (Saravia, 2016).

MÉTODOS Y DATOS

Uso de EXTENSIONES

Se ha incorporado a los diagramas (.dia), un cuadro, el EXTENSIONES, donde se deben colocar los nombres de los módulos o extensiones con los que se quiere enlazar el ejecutable de Sceptre. Dichos módulos deben estar instalados como librerías, tanto en sus versiones en c, como en Fortran. Esto implica tener archivos del tipo libXXX.so.* y libXXXfor.so.*, en los directorios apropiados. XXX es el nombre del módulo, tal como se indica en el cuadro EXTENSIONES. Los mecanismos de instalación de cada extensión deben ubicar estos archivos donde son buscados.

Un módulo puede necesitar a otros para funcionar, por ejemplo Radia, requiere Tabular en algunos casos.

Las librerías y ejemplos en Fortran para usar los módulos c se distribuyen con el propio Sceptre-Simusol, pero la idea es que en el futuro se integren en la distribución propia de cada módulo. Se encuentran en el directorio “extras”, junto a los tradicionales “phase1”, “phase2” y “auxpro”, bajo el directorio “src”, en las fuentes del paquete Sceptre-Simusol.

Tabular

Es una librería de software, conteniendo funciones, que permite leer tablas, interpolarlas y extrapolarlas, sea en forma lineal o con curvas de Bezier spline (Wikipedia, 2018). Las tablas se leen de un archivo de texto con datos en columnas, o son cargadas en forma “virtual” desde el programa que invoca la librería mediante funciones.

Las tablas leídas por Tabular desde archivos no tienen otro límite de tamaño que la capacidad de memoria de la computadora donde se ejecute.

Está redactada en c. Puede usarse:

- En programas en c. Se usa como encabezado el archivo "simusol_table.h", se enlaza en gcc con el argumento -lsimusol_table.
- Como función de ExprTk (C++) (Partow 2015), así se puede usar en Siru, alternativa al Sceptre, en desarrollo (Saravia, 2014).
- En programas redactados en Fortran.
- En programas Perl como módulo, con el nombre Simusol::Tabular.
- El código de Tabular junto con un software denominado “Terminal” se ha comenzado a usar para conducir la ejecución de Siru. El analizador léxico de Tabular produce “tokens” o sílabas que se almacenan en una tabla virtual y se pueden interpretar con “parsers”, o analizadores sintácticos. Se ha comenzado a usar “Marpa” como tal (Savage, 2018) . Esto simplifica y abstrae varias capas de desarrollo de Siru.
- Como EXTENSION de Simusol/Sceptre

Funciones

En lo esencial Tabular se usa con una función denominada **interpolate()** a la que se le dan todos los datos necesarios y produce el valor de la tabla interpolado. Desde Simusol/Sceptre se usa la función **ETABLE()** que adapta mínimamente los argumentos al uso con Simusol/Sceptre en Fortran. Como Sceptre no permite pasar argumentos que no sean numéricos, se usan números para indicar los archivos y las columnas a usar. También se pueden usar funciones para cargar tablas virtuales, en los casos donde las mismas no provengan de archivos.

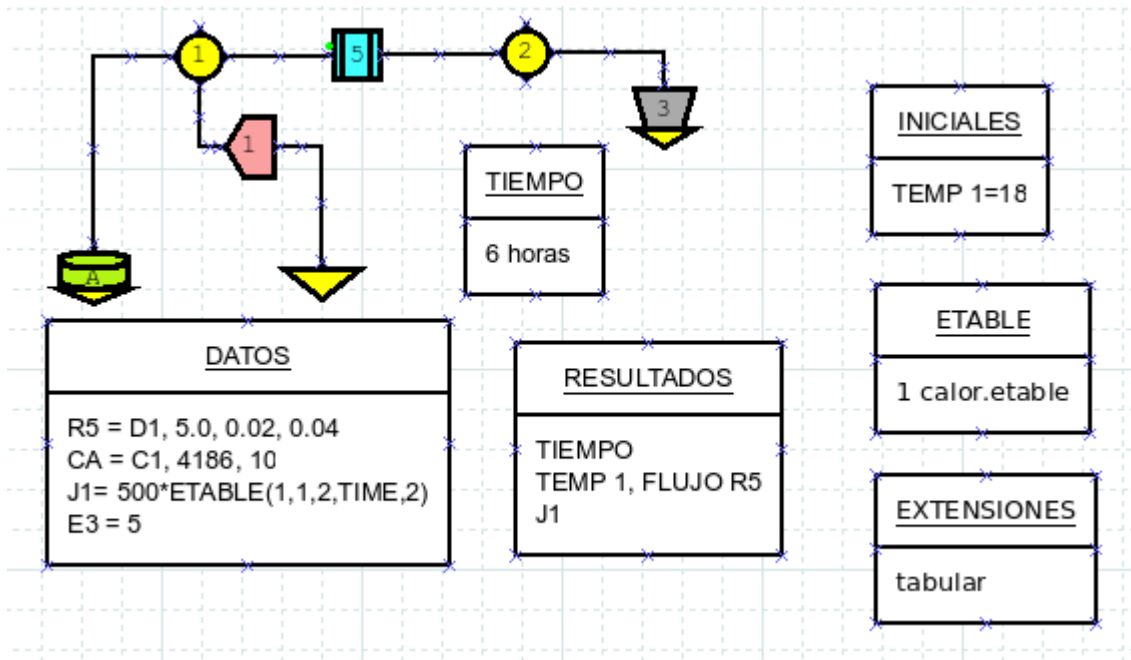


Figura 1: Diagrama que usa la extensión Tabular (tanqueT.dia en ejemplos de Simusol)

La función *ETABLE()*

Valor_de_Y =ETABLE(Número_de_tabla, Columna_X, Columna_Y, Valor_X, lineal (1)/spline (2))

Las columnas se especifican con números.

En el ejemplo: ETABLE(1,1,2,TIME,2)

- El primer argumento es el número que corresponde al archivo en el cuadro ETABLE, donde cada línea contiene el número y el nombre del archivo conteniendo la tabla. El nombre del archivo es relativo al directorio donde se ejecuta Simusol.
- El segundo argumento es el número de la columna de la variable a buscar en la tabla.
- El tercero, es el número de columna de la variable resultado en la tabla.
- El cuarto es la variable del Sceptre cuyo valor se debe encontrar en la tabla.
- El quinto: si es 1 interpola en forma lineal, con 2 usa spline

La función *interpolate()*

Al usarlo con c, Fortran, Perl, o ExprTk se usa la función interpolate similar a la ETABLE pero con argumentos alfanúmericos, y el resultado se obtiene con un puntero al tipo "double float", ejemplo:

```
interpolate("archivo", "UNO", "DOS", 550., 2, &result);
```

"Archivo" es una cadena de caracteres con la ubicación y nombre de un archivo, "UNO" es una cadena de caracteres con el nombre la columna que hace de dominio, y "DOS" de la columna que hace de codominio. Ambos nombres deben estar en la

primera línea del archivo de datos. En este ejemplo, 550. es el valor que queremos buscar en la columna "UNO" y en "&result", un puntero a un valor, se obtiene el resultado. El número entero 2 indica que se usa interpolación de Bezier.

Radia

Radia es una librería de software que provee funciones para calcular la posición del Sol, la radiación de cielo despejado, series estocásticas de radiación estimadas con diferentes métodos a partir de datos medios mensuales, y la radiación sobre superficies inclinadas cubiertas. Es decir todo la información de radiación que necesita Simusol para simular equipos que aprovechen la energía solar.

Está redactada en c. Al igual que Tabular, puede accederse a Radia desde un programa en c, en Perl, o bien desde Fortran para ser usada por Simusol/Sceptre como EXTENSION.

En Radia están disponibles diversos mecanismos para realizar cálculos, y se siguen agregando:

- Software SPA (Reda, 2003) y alternativos, para calcular la posición del Sol. SPA no es software libre, si bien es posible usarlo registrándose y descargando el código en forma individual, no podemos redistribuirlo junto a Simusol. Por lo que se proveen métodos alternativos menos precisos, para que Simusol pueda usarse en forma completamente libre. Coincido con el pedido a NREL para la completa liberación de SPA.
- Cálculo de radiación de Cielo despejado, a partir de datos del tiempo, latitud y longitud.

Éstas se calculan en cada momento de la simulación y no plantean ninguna consideración especial.

- Cálculo de radiación por métodos estocásticos, que permiten generar intervalos arbitrarios de resolución horaria de radiación que simule la real en cualquier lugar de la Tierra, a partir de diversos datos.

En éste caso se calculan las series al comienzo de la simulación, para horas específicas a intervalos fijos, – toda la serie –, y se construyen tablas virtuales con los resultados, usando Tabular.

Sceptre a veces va para adelante y otras para atras en el tiempo, en la resolución de las ecuaciones diferenciales. Entonces hay que recordar los valores aleatorios previos y futuros, ya calculados, e interpolar para un momento cualquiera.

- Uso de datos externos de radiación, suministrados por el usuario, contenidos en archivos, generalmente tipo TMY (Wilcox, 2008).

En éste casos también se usan tablas leídas a partir de archivos.

Dichos métodos facilitan el uso de Simusol para usuarios y desarrolladores de sistemas basados en energía solar. También contiene otras funciones útiles. Esta pendiente incorporar las funciones generadoras de las series sintéticas reducidas de radiación solar (SSR) (Saravia, 2016) a esta librería.

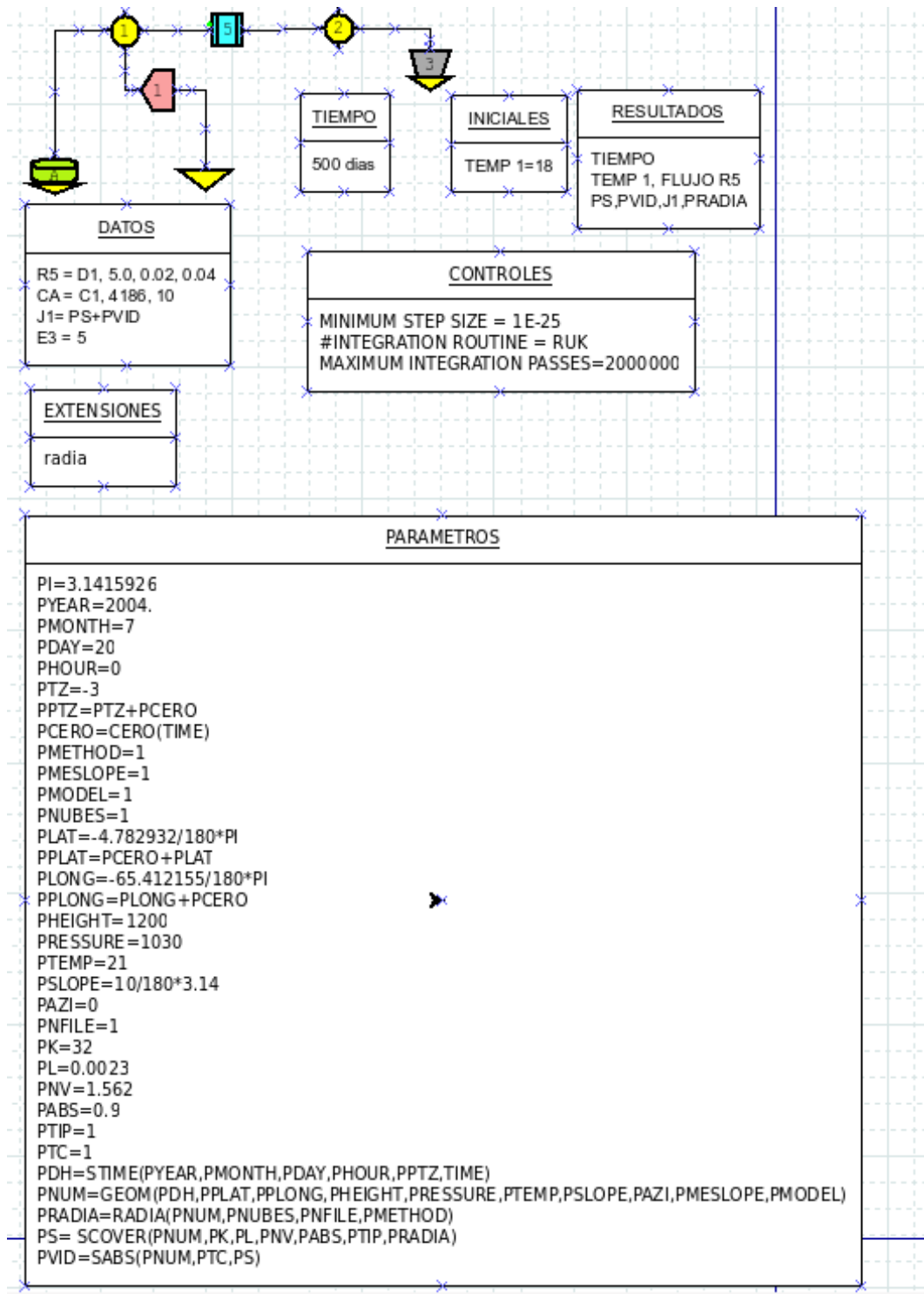


Figura 2: Diagrama que usa la extensión Radia.

Funciones:

- Tiempo:
 $PDHOUR = STIME(PYEAR, PMONTH, PDAY, PHOUR, PPTZ, TIME)$
 Calcula el tiempo: PYEAR, un mes: PMONTH, un día: PDAY, una hora PHOUR, una zona geográfica: PPTZ, y un tiempo adicional: TIME, dando en PDHOUR el resultado.
- Posición:

```
PNUM=GEOM(PD HOUR , PPLAT , PPLONG , PHEIGHT , PRESSURE , PTEMP , PSL
OPE ,
PAZI , PMESLOPE , PMODEL)
```

Calcula la ubicación del sol en coordenadas locales dada PDHOUR, latitud: PPLAT, longitud: PPLONG, altura: PHEIGHT, presión atmosférica: PRESSURE, temperatura: PTEMP, ángulo de la superficie con relación al piso: PSLOPE, ángulo de la superficie con relación al eje vertical: PAZI, modelo para el cálculo de inclinación: PMESLOPE, modelo: PMODEL, el mejor disponible es el software SPA.

PNUM es un número arbitrario que entrega la función para que el Simusol, vea que produce resultados y no deje de calcularla, en realidad el módulo Fortran usa variables globales que almacenan los resultados y argumentos de una función cuando calcula todas ellas.

- Radiación de cielo despejado:
`PRADIA=RADIA(PNUM, PNUBES, PNFIL, PMETHOD)`
 Kt: PNUBES, Método usado: PNFIL y PMETHOD.
- También se usan las funciones SCOVER y SVID, que calculan la radiación sobre una superficie inclinada y la absorbida por el vidrio, respectivamente.

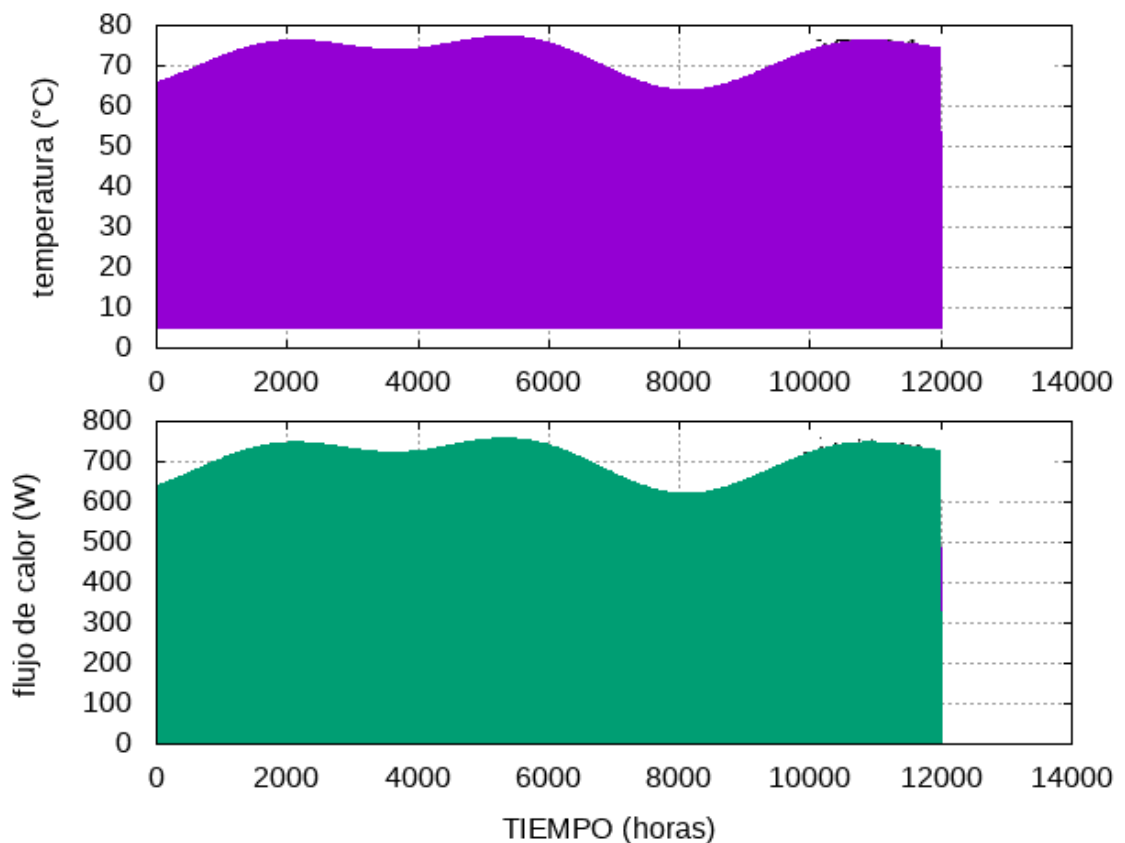


Figura 3: Resultados de la simulación del diagrama de la Figura 2, los datos parecen plenos, porque las líneas con los mismos suben y bajan todos los días.

MODELOS ESTOCÁSTICOS

Se usan dos métodos aleatorios para generar una serie que brinde radiación horaria en intervalos largos de tiempo.

Esta serie se calcula al inicio de la simulación, se cargan sus valores en una tabla virtual y se accede a la misma mediante las funciones de Tabular. El cálculo puede hacerse con una semilla prefijada por el usuario, a los efectos de poder repetir simulaciones, o bien usando una producida en forma aleatoria con una fuente de entropía provista por fuentes físicas en la computadora.

Generación de secuencias sintéticas de radiación global diaria. Matrices de Markov (MTM)

Se generan secuencias de radiación diaria, para cualquier lugar, a partir de la radiación media mensual. Las secuencias son indistinguibles estadísticamente de las reales (Aguar, R. J. et al., 1988)

El método se basa en que:

- a) hay un correlación significativa solo entre los valores de radiación para días consecutivos y
- b) la probabilidad de la ocurrencia de los valores de radiación es la misma para los meses con la misma Kt, (índice de claridad).

El método emplea matrices de transición de Markov, cada una correspondiente a un intervalo específico en Kt.

En la simulación se parte del valor medio mensual de Kt para la ubicación y tiempo que se quiere calcular. Se suministran los valores de Kt para la ubicación y para todos los meses que se necesiten (ejemplo en Tabla 1).

1. Una vez seleccionado el mes y teniendo en cuenta su Kt, se usa la matriz de transición de Markov que corresponda, entre 10 pre-calculadas, cada una para un rango de Kt (ejemplo en Tabla 3).
2. Para el primer día de la serie, se toma el valor de Kt medio del mes anterior. Se obtiene el primer valor para comenzar los cálculos. Para el siguiente:
3. A partir de la Tabla 2, que tiene valores mínimos y máximos de Kt, para cada Matriz de transición de Markov y usando esos valores se establecen 10 intervalos. Con el valor obtenido para el día anterior se verifica a cuál intervalo corresponde y según eso se elige una de las 10 filas de la matriz.
4. Se genera un número aleatorio uniforme entre cero y uno.
5. Se van sumando los valores de la fila obtenida en 3, columna por columna, empezando desde la primera, hasta que el valor de la suma sea superior al número aleatorio.
6. El número de columna en que se terminó de sumar se usa para seleccionar los valores máximo y mínimos correspondientes a la Tabla 2. Se interpola linealmente y se usa el promedio como Kt del día. Así se obtiene un nuevo valor.
7. Se repite desde el punto 3.

Mes	1	2	...	11	12
Kt	0,435	0,345	...	0,464	0,452

Tabla 1. Vector de Kt medios mensuales

.031	.705
.058	.694
.051	.753
.052	.753
.028	.807
.053	.856
.044	.818
.085	.846
.010	.842
.319	.865

Tabla 2 K_t mínimos y máximos por cada Matriz de Transición de Markov.

.229	.167	.250	.158	.211	.125	.040	.000	.000	.000
.333	.319	.250	.237	.053	.125	.240	.250	.250	.000
.208	.194	.091	.158	.211	.250	.080	.000	.000	.000
.042	.139	.136	.263	.158	.188	.120	.125	.125	.000
.083	.097	.091	.026	.053	.063	.080	.000	.250	.000
.042	.028	.046	.053	.053	.125	.080	.125	.000	.000
.042	.042	.046	.079	.158	.000	.120	.125	.250	.500
.021	.000	.023	.026	.105	.125	.120	.250	.000	.250
.000	.014	.068	.000	.000	.000	.080	.063	.000	.000
.000	.000	.000	.000	.000	.000	.040	.063	.125	.250

Tabla 3. Matriz de Markov para el intervalo $K_t > 0 - K_t < 0.30$

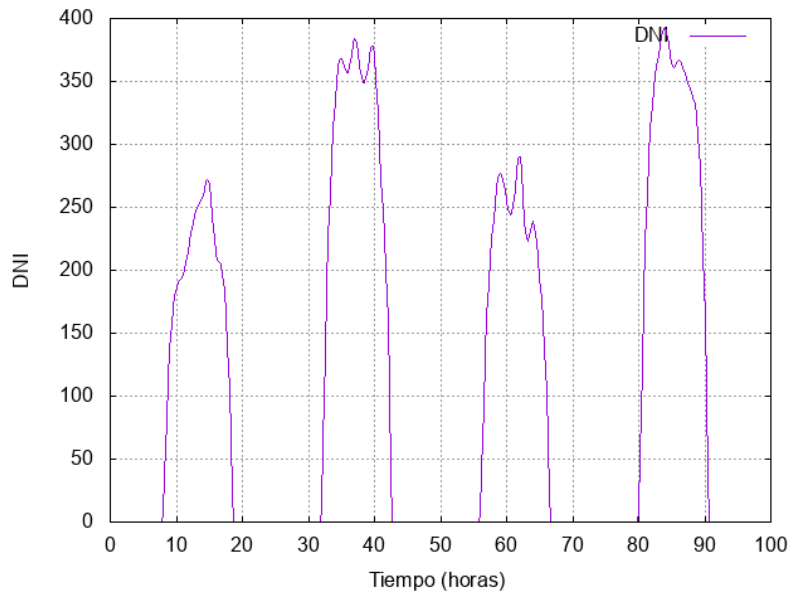


Figura 4: Resultados en una serie de radiación de 4 días, combinando ambos métodos. Generación de secuencias sintéticas de radiación global horaria. Time-dependent Autoregressive Gaussian (TAG)

Se generan secuencias diarias de valores de radiación horaria, en el plano horizontal, para cualquier lugar, teniendo como dato el índice diario de claridad K_t . El modelo asume que para cada K_t y hora solar, la densidad de probabilidad del índice horario kt

es una función gaussiana truncada. Se usa una variable normalizada independiente de la hora solar, en reemplazo de kt , donde interviene el índice diario Kt . Se usa el valor de la hora anterior y se le suma una variable que resulta del azar generado con una distribución gaussiana de desviación estándar definida. Si se superan los valores máximos y mínimos (0) de kt , se reinicia el cálculo para la hora en cuestión.

En (Aguiar 1992) se puede estudiar el método en todo su detalle, incluyendo como se comprueba su validez.

El índice diario Kt , se obtiene de los valores obtenidos con el método MTM.

La figura 4 muestra una serie de 4 días calculados combinando ambos métodos.

RESULTADOS Y CONCLUSIONES

Se ha desarrollado:

1. Una infraestructura informática para dotar a Simusol/Sceptre de mecanismos de EXTENSION dinámicos. Las EXTENSIONES o módulos pueden usarse además desde programas independientes Perl, c y Fortran.
2. Una EXTENSION denominada Tabular, para leer, parsear, interpolar, y generar tablas.
3. Otra EXTENSION denominada Radia para construir series de radiación para ejecutar Simusol. Se incorporaron funciones para determinar la posición del Sol, la radiación de día despejado y modelos estocásticos como MTM y TAG, junto a funciones para calcular la radiación sobre una placa inclinada con superficie vidriada o similar.

Todo este software se irá haciendo disponible en futuras versiones de Simusol, como es habitual en éste proyecto.

REFERENCIAS

Aguiar R., Collares-Pereira, M. y Conde J. P. (1988). A simple procedure for generating sequences of daily radiation values using a library of Markov transition matrices. *Solar Energy* 40, 3, 269-279.
<https://www.sciencedirect.com/science/article/pii/0038092X88900497>

Aguiar R. y Collares-Pereira, M. (1992). TAG: A time-dependent, autoregressive, Gaussian model for generating synthetic hourly radiation, *Solar Energy* 49, 167-174.
<https://www.sciencedirect.com/science/article/pii/0038092X9290068L>

Alía, D., Saravia L. y Saravia D. (2002). Avances introducidos en la capacidad del simulador de sistemas solares térmicos SIMUTERM (SIMUSOL). *Avances en Energías Renovables y Medio Ambiente. AVERMA* 6, 2, 8.31-8.36.
<http://www.cricyt.edu.ar/asades/modulos/averma/trabajos/2002/2002-t008-a006.pdf>

Mellit A., Benghanem M., Hadj Arab A. y Guessoum A. (2005). A simplified model for generating sequences of global solar radiation data for isolated sites: Using artificial neural network and a library of Markov transition matrices approach. *Solar Energy* 79, 5, 469-482.
<https://www.sciencedirect.com/science/article/pii/S0038092X05000204>

- Ngoko B., Sugihara H. y Funaki T. (2014). Synthetic generation of high temporal resolution solar radiation data using Markov models. *Solar Energy* 103, 160-170
- Paoli C., Voyant C, Muselli M. y Nivet M. (2010). Forecasting of preprocessed daily solar radiation time series using neural networks. *Solar Energy*. 84, 12, 2146-2160
<https://www.sciencedirect.com/science/article/pii/S0038092X10002793>
- Partow, A. (2015). C++ Mathematical Expression Toolkit Library (ExprTk). Sitio web personal.
<http://www.partow.net/programming/exprtk/>
- Reda, I. y Andreas, A. (2003). Solar Position Algorithm for Solar Radiation Applications. NREL Report No. TP-560-34302.
<https://www.nrel.gov/docs/fy08osti/34302.pdf>
- Saravia, D. (2014). SIMUSOL: Simulación de diagramas de enlace o Bond-Graphs. *AVERMA*, 18, 08.43-08.50.
<http://www.cricyt.edu.ar/asades/modulos/averma/trabajos/2014/2014-t008-a006.pdf>
- Saravia, D. (2016). Caracterización de radiación solar y construcción de series sintéticas reducidas de radiación solar y construcción de series sintéticas reducidas de radiación solar (SSRRS) para simular equipos con SIMUSOL. *AVERMA* 20, 08.13-08.22.
<http://www.cricyt.edu.ar/asades/modulos/averma/trabajos/2016/2016-t008-a004.pdf>
- Saravia L., Saravia, D. (2000). Simulación de sistemas solares térmicos con un Programa de cálculo de circuitos eléctricos de libre disponibilidad. *AVERMA* 4, 4 8.17-8.22.
<http://www.asades.org.ar/modulos/averma/trabajos/2000/2000-t008-a004.pdf>
- Savage, R. (2018). Sitio de Marpa
<https://savage.net.au/Marpa.html>
- Wilcox S. y Marion W. (2008) Users Manual for TMY3 Data Sets. Technical Report NREL/TP-581-43156
<https://www.nrel.gov/docs/fy08osti/43156.pdf>
- Wikipedia (2018). Curva de Bezier
https://es.wikipedia.org/wiki/Curva_de_Bezier

ABSTRACT

Advances are presented in Simusol/Sceptre software that allow external EXTENSIONS developed in c. In particular there are two extensions: Tabular, to read tables from text files, during Sceptre operation. It has no size limits set in the program. Radia, to calculate synthetic radiation series. It incorporates various methods of clear days, and stochastic methods, one based on Markov matrices, another on Gaussian autoregressive model. Both combined generate radiation for an arbitrary time interval with hourly time resolution.

Keywords: Sceptre, Simusol, Radiation, Markov, Tabular