





Notación:

-  Módulo principal del mét. numérico (preimplementado, no programable por el usuario).
-  Módulo del método numérico programable por el usuario.
-  Módulo del método numérico preimplementado (no programable por el usuario)
-  Modelo de datos (constantes, esquema numérico, etc.)

sample: variable de entrada a un módulo.

sample: variable de salida de un módulo.

dim1×dim2: dimensiones de una variable (matriz, vector o escalar)

Resumen de variables:

- xnode**: matriz de pares (x,y) representando cada nodo de la malla.
- icone**: matriz de conectividad. Todos los nodos se conectan formando elementos rectangulares (pero el tratamiento general del método es por celdas).
- neighb**: matriz de vecindad (conectividad de cada celda con 2, 3 o 4 celdas vecinas).
- cells**: struct con la información física y geométrica de cada celda de la malla.
- K**: matriz del sistema (difusión + reacción).
- F**: vector de flujo térmico.
- PHI**: vector solución del método numérico (escalar).
- DIR**: matriz de celdas frontera tipo Dirichlet.
- NEU**: matriz de celdas frontera tipo Neumann.
- ROB**: matriz de celdas frontera tipo Robin.
- model**: struct con todos los datos del modelo (constantes, esquema numérico, etc.)
- Q**: vector/matriz de flujo de calor.
- k**: conductividad térmica del material. Es un vector que permite representar $k(x,y)$.
- c**: constante de reacción del material. Es un vector que permite representar $c(x,y)$.
- G**: fuente de calor. Es un vector que permite representar $G(x,y)$.

Resumen de dimensiones de variables:

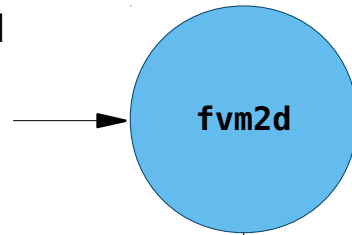
- nnodes**: cantidad total de nodos de la malla.
- ncells**: cantidad total de celdas de la malla.
- ndir**: cantidad de nodos frontera tipo Dirichlet.
- nneu**: cantidad de pares de nodos (lado de un elemento) frontera tipo Neumann.
- nrob**: cantidad de pares de nodos (lado de un elemento) frontera tipo Robin.
- nit**: cantidad de iteraciones alcanzadas por el esquema temporal.



model

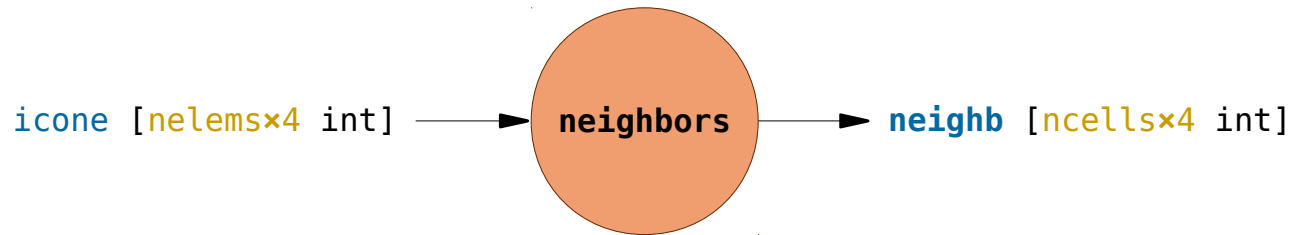
- `ncells` [1x1 int] - Cantidad de celdas totales de la malla.
- `ts` [1x1 double] - Espesor de la malla.
- `k` [nnodesx1 double] - Conductividad térmica del material. Permite representar una distribución $k(x,y)$ en toda la malla.
- `c` [nnodesx1 double] - Constante de reacción del sistema. Representa $c(x,y)$. Permite representar una distribución $c(x,y)$ en toda la malla.
- `G` [nnodesx1 double] - Fuente de calor volumétrica. Permite representar una distribución $G(x,y)$ en toda la malla.
- `ts` [1x1 int] - Parámetro de selección de esquema temporal: (0) Esquema Explícito, (1) Esquema Implícito y cualquier otro valor para estado estacionario.
- `rho` [1x1 double] - Densidad del material.
- `cp` [1x1 double] - Calor específico a presión constante del material.
- `maxit` [1x1 int] - Cantidad máxima de iteraciones (esquemas temporales).
- `tol` [1x1 double] - Tolerancia de error relativo entre iteraciones (esquemas temporales)
- `dt` [1x1 double] - Paso temporal de Esquema Implícito (arbitrario).
- `PHI_n` [nnodesx1 double] - Condición inicial para esquemas temporales. Es un vector que asigna un valor arbitrario inicial a cada nodo de la malla.

```
xnode [nnodes×2 double]
icone [nelem×4 int]
  DIR [ndir×1 double]
  NEU [nneu×1 double]
  ROB [nrob×1 double]
model [1×1 struct]
```



```
PHI [ncells×nit sparse double]
Q   [ncells×(2×nit) double]
```

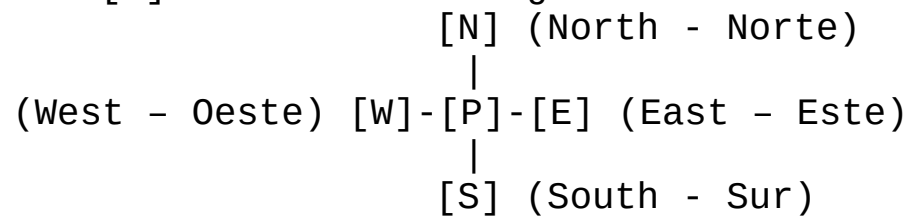
```
% Armado de la matriz de vecindad
1. [neighb] = fvm2d_neighbors(icone);
% Inicialización de variables principales del sistema
2. [K,F,cells] = fvm2d_initialize(xnode,icone,neighb,...
                                model.th,model.k);
% Ensamble de celdas que no tienen lados en la frontera
3. [K,F] = fvm2d_gen_system(K,F,neighb,cells,model.c,model.G);
% Ensamble de celdas con lados en frontera Neumann
4. [F] = fvm2d_neumann(F,cells,NEU);
% Ensamble de celdas con lados en frontera Robin
5. [K,F] = fvm2d_robin(K,F,cells,ROB);
% Ensamble de celdas con lados en frontera Dirichlet
6. [K,F] = fvm2d_dirichlet(K,F,cells,DIR);
% Resolución del sistema lineal de ecuaciones
7. [PHI,Q] = fvm2d_solve(K,F,neighb,cells,model);
```



Descripción: módulo para armar la matriz de vecindad (**neighb**). La malla del dominio se encuentra formada por elementos rectangulares. Cada uno de los 4 nodos que forman el elemento se enumeran en sentido antihorario empezando con la esquina inferior izquierda. Cada elemento conforma una celda. A partir de esta conectividad (almacenada en **icone**) se arma la matriz de vecindad indicando para cada celda sus celdas vecinas. Si una celda no tiene algún vecino, el índice correspondiente se guarda como -1. Existen 3 tipos de celdas:

- Celdas esquina: los cuales se conectan con otras 2 celdas.
- Celdas de borde: los cuales se conectan con otras 3 celdas.
- Celdas interiores: los cuales se conectan con otras 4 celdas.

Para una celda cualquiera [P] se define la siguiente vecindad:

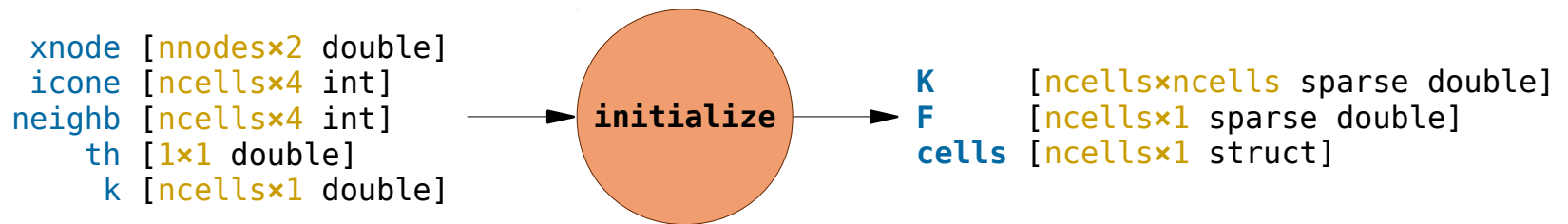


Entrada:

- **icone**: matriz de conectividad. Cada fila de la matriz indica la conectividad de un elemento rectangular, comenzando por el extremo inferior izquierdo y recorriendo el elemento en sentido antihorario.

Salida:

- **neighb**: matriz de vecindad.



Descripción: módulo para inicializar las variables principales del sistema (*sparse* para optimizar el rendimiento) y las celdas de la malla. Cada celda se modela como un *struct* con la siguiente información:

- `ds`, `de`, `dn`, `dw`: las distancias a los centroides vecinos. Para celdas de borde es la distancia del centroide al borde.
- `as`, `ae`, `an`, `aw`: las áreas por lado de la celda (South, East, North, West).
- `ks`, `ke`, `kn`, `kw`: conductividad térmica por lado de la celda. En caso de celdas con lados interfaz con otra celda, se utiliza la media armónica de ambos valores.
- `ts`, `te`, `tn`, `tw`: las temperaturas en el centro de cada lado de la celda.
- `tc`: la temperatura en el centroide de la celda.
- `dx`, `dy`: dimensiones de la celda en sentido eje-x y eje-y, respectivamente.
- `cx`, `cy`: posición (x,y) del centroide de la celda.
- `v`: volumen de la celda (depende del espesor de la placa).

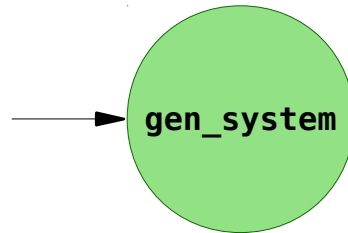
Entrada:

- `xnode`: matriz de pares (x,y) representando cada nodo de la malla.
- `icone`: matriz de conectividad. Cada fila de la matriz indica la conectividad de un elemento rectangular, comenzando por el extremo inferior izquierdo y recorriendo el elemento en sentido antihorario.
- `th`: espesor de la malla.
- `k`: conductividad térmica del material. Es un vector que permite representar $k(x,y)$.

Salida:

- `K`: matriz del sistema (difusión + reacción).
- `F`: vector de flujo térmico.
- `cells`: vector de celdas.

```
K [ncells×ncells sparse double]
F [ncells×1 sparse double]
neighb [ncells×4 int]
cells [ncells×1 struct]
c [1×1 double]
G [ncells×1 double]
```



```
K [ncells×ncells sparse double]
F [ncells×1 sparse double]
```

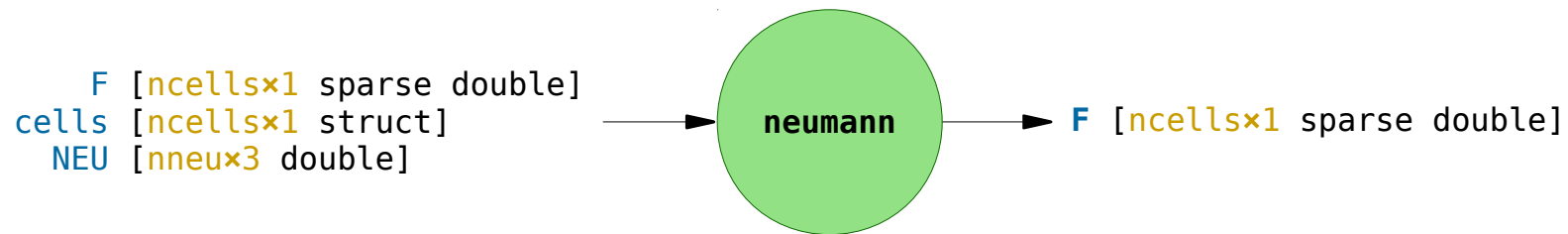
Descripción: módulo para ensamblar los términos difusivo, reactivo y fuente de todas las celdas de la malla, generando el *stencil* adecuado dependiendo de si alguna de sus caras pertenece o no a la frontera.

Entrada:

- **K**: matriz del sistema (difusión + reacción).
- **F**: vector de flujo térmico.
- **neighb**: matriz de vecindad.
- **cells**: vector de celdas.
- **c**: constante del término reactivo. Es un vector que permite representar $c(x,y)$.
- **G**: fuente volumétrica. Es un vector que permite representar $G(x,y)$.

Salida:

- **K**: matriz del sistema (difusión + reacción) con modificaciones luego del ensamble.
- **F**: vector de flujo térmico con modificaciones luego del ensamble.



Descripción: módulo para calcular y ensamblar las contribuciones de celdas con caras pertenecientes a fronteras de tipo Neumann.

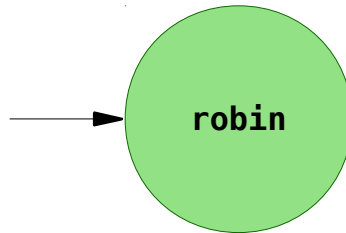
Entrada:

- `F`: vector de flujo térmico.
- `cells`: vector de celdas.
- `NEU`: matriz con la información sobre la frontera de tipo Neumann.
 - Columna 1: índice de la celda donde se aplica la condición de borde.
 - Columna 2: valor de flujo térmico (q) asociado al lado del elemento.
 - Columna 3: dirección y sentido del flujo:
 - 1) Flujo en dirección eje-y, sentido negativo (S - South - Sur)
 - 2) Flujo en dirección eje-x, sentido positivo (E - East - Este)
 - 3) Flujo en dirección eje-y, sentido positivo (N - North - Norte)
 - 4) Flujo en dirección eje-x, sentido negativo (W - West - Oeste)

Salida:

- `F`: vector de flujo térmico con modificaciones luego de aplicar la condición de borde.

K [**ncells**×**ncells** sparse double]
F [**ncells**×1 sparse double]
cells [**ncells**×1 struct]
ROB [**nrob**×4 double]



K [**ncells**×**ncells** sparse double]
F [**ncells**×1 sparse double]

Descripción: módulo para calcular y ensamblar las contribuciones de nodos pertenecientes a fronteras de tipo Robin.

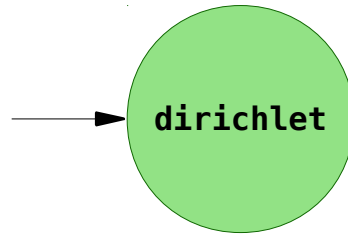
Entrada:

- **K**: matriz del sistema (difusión + reacción)
- **F**: vector de flujo térmico.
- **cells**: vector de celdas.
- **ROB**: matriz con la información sobre la frontera de tipo Robin.
 - Columna 1: índice de la celda donde se aplica la condición de borde.
 - Columna 2: valor de coeficiente de calor (**h**)
 - Columna 3: valor de temperatura de referencia (**phi_inf**).
 - Columna 4: dirección y sentido del flujo:
 - 1) Flujo en dirección eje-y, sentido negativo (S - South - Sur)
 - 2) Flujo en dirección eje-x, sentido positivo (E - East - Este)
 - 3) Flujo en dirección eje-y, sentido positivo (N - North - Norte)
 - 4) Flujo en dirección eje-x, sentido negativo (W - West - Oeste)

Salida:

- **K**: matriz del sistema (difusión + reacción) con modificaciones luego de aplicar la condición de borde.
- **F**: vector de flujo térmico con modificaciones luego de aplicar la condición de borde.

K [**ncells**×**ncells** sparse double]
F [**ncells**×1 sparse double]
cells [**ncells**×1 struct]
DIR [**ndir**×4 double]



K [**ncells**×**ncells** sparse double]
F [**ncells**×1 sparse double]

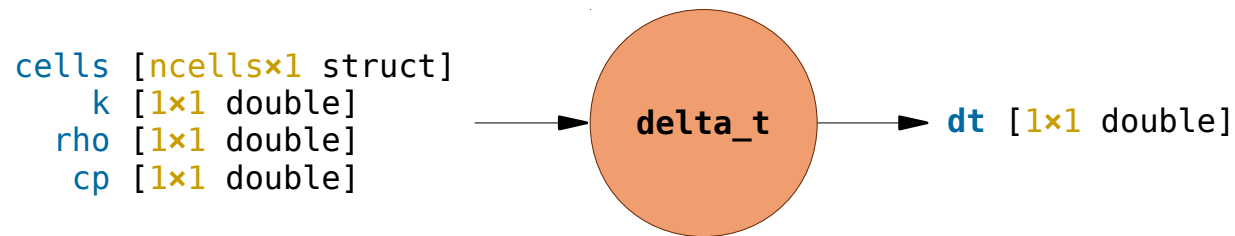
Descripción: módulo para calcular y ensamblar las contribuciones de celdas pertenecientes a fronteras de tipo Dirichlet.

Entrada:

- **K**: matriz del sistema (difusión + reacción)
- **F**: vector de flujo térmico.
- **cells**: vector de celdas.
- **DIR**: matriz con la información sobre la frontera de tipo Dirichlet.
 - Columna 1: índice de la celda donde se aplica la condición de borde.
 - Columna 2: valor en la cara de la celda (escalar)
 - Columna 3: cara a la que se aplica la condición de borde:
 - 1)S - South - Sur
 - 2)E - East - Este
 - 3)N - North - Norte
 - 4)W - West - Oeste

Salida:

- **K**: matriz del sistema (difusión + reacción) con modificaciones luego de aplicar la condición de borde.
- **F**: vector de flujo térmico con modificaciones luego de aplicar la condición de borde.



Descripción: módulo para calcular el paso temporal crítico para esquema temporal explícito a partir de las constantes del modelo y las dimensiones de los elementos de la malla.

Entrada:

- `cells`: vector de celdas.
- `k`: vector de conductividad de la malla. Hay una entrada por cada celda de la malla.
- `rho`: densidad del material.
- `cp`: calor específico del material.

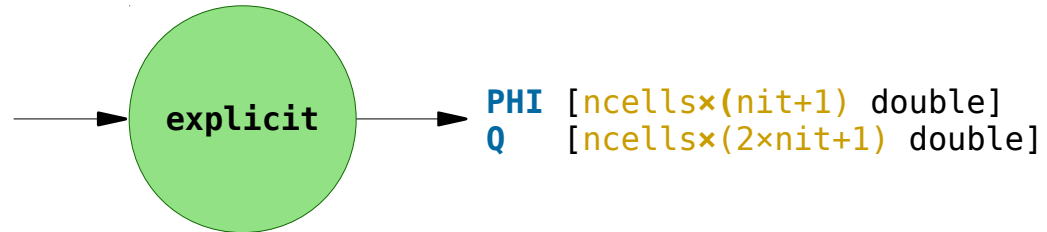
Salida:

- `dt`: paso temporal crítico para método explícito.

```

    K [ncells×ncells sparse double]
    F [ncells×1 sparse double]
    cells [ncells×1 struct]
    neighb [ncells×4 int]
    model [1×1 struct]
    dt [1×1 double]

```



Descripción: módulo para resolver el sistema lineal de ecuaciones utilizando esquema temporal *explícito*. El primer valor (primer columna) es la condición inicial.

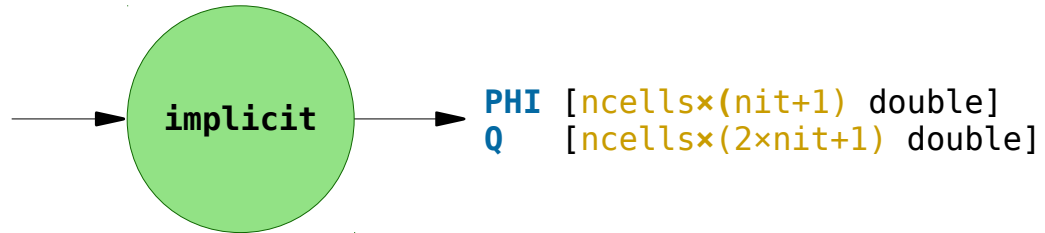
Entrada:

- **K**: matriz del sistema (difusión + reacción)
- **F**: vector de flujo térmico.
- **cells**: vector de celdas.
- **neighb**: matriz de vecindad.
- **model**: struct con todos los datos del modelo (constantes, esquema numérico, etc.)
- **dt**: paso temporal.

Salida:

- **PHI**: vector solución. Cada elemento del vector representa un valor escalar asociado al centroide de cada celda de la malla, y su posición dentro del vector depende de cómo se especificó cada celda en **icone**. Se devuelve un resultado por cada iteración del método (**nit** columnas).
- **Q**: vector de flujo de calor. Se forma de una componente en sentido x, Qx, y una componente en sentido y, Qy. Se devuelve un resultado por cada iteración del método (**2×nit** columnas).

```
K [ncells×ncells sparse double]
F [ncells×1 sparse double]
cells [ncells×1 double]
neighb [ncells×4 int]
model [1×1 struct]
dt [1×1 double]
```



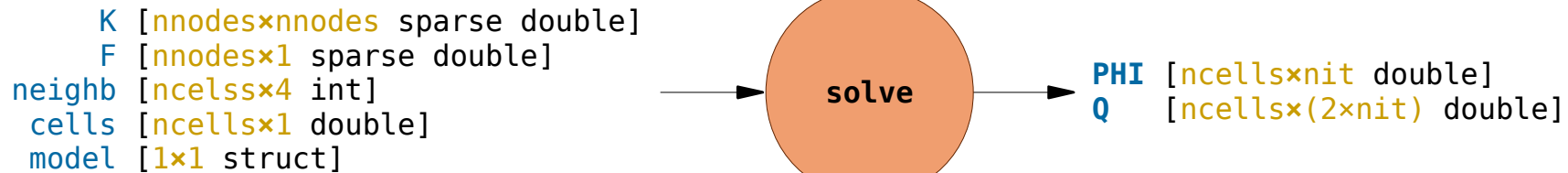
Descripción: módulo para resolver el sistema lineal de ecuaciones utilizando esquema temporal *implícito*. El primer valor (primer columna) es la condición inicial.

Entrada:

- **K**: matriz del sistema (difusión + reacción)
- **F**: vector de flujo térmico.
- **cells**: vector de celdas.
- **neighb**: matriz de vecindad.
- **model**: struct con todos los datos del modelo (constantes, esquema numérico, etc.)
- **dt**: paso temporal.

Salida:

- **PHI**: vector solución. Cada elemento del vector representa un valor escalar asociado al centroide de cada celda de la malla, y su posición dentro del vector depende de cómo se especificó cada celda en **icone**. Se devuelve un resultado por cada iteración del método (**nit** columnas).
- **Q**: vector de flujo de calor. Se forma de una componente en sentido x, Q_x , y una componente en sentido y, Q_y . Se devuelve un resultado por cada iteración del método (**2×nit** columnas).



Descripción: módulo para resolver el sistema lineal de ecuaciones. En este módulo se realizan los cálculos para obtener la solución propia del método numérico. Dicha solución se obtiene por dos vías:

- Sin la aplicación de esquemas temporales, es decir, la solución del sistema en estado estacionario. Resolución por método directo.
- Aplicación de esquemas temporales, a saber: método explícito y método implícito. Se evalúa la evolución temporal del sistema desde un estado inicial conocido hasta un determinado instante de tiempo. Resolución por método iterativo.

Entrada:

- `K`: matriz del sistema (difusión + reacción)
- `F`: vector de flujo térmico.
- `neighb`: matriz de vecindad.
- `cells`: vector de celdas.
- `model`: struct con todos los datos del modelo (constantes, esquema numérico, etc.)

Salida:

- `PHI`: vector solución. Cada elemento del vector representa un valor escalar asociado al centroide de cada celda de la malla, y su posición dentro del vector depende de cómo se especificó cada celda en `icone`. Se devuelve un resultado por cada iteración del método (`nit` columnas).
- `Q`: vector de flujo de calor. Se forma de una componente en sentido x, `Qx`, y una componente en sentido y, `Qy`. Se devuelve un resultado por cada iteración del método (`2×nit` columnas).



Descripción: módulo para actualizar los valores de temperaturas de cada celda. Luego de resolver el sistema de ecuaciones, en **PHI** se obtiene la temperatura en el centroide de cada celda. Con esta información y con la información existente sobre las celdas de frontera, para cada celda (P) se recalculan o sobrescriben los siguientes valores:

- **cells(P).ts**: temperatura en la cara S de la celda P (South - Sur)
- **cells(P).te**: temperatura en la cara E de la celda P (East - Este)
- **cells(P).tn**: temperatura en la cara N de la celda P (North - Norte)
- **cells(P).tw**: temperatura en la cara W de la celda P (West - Oeste)
- **cells(P).tc**: temperatura en el centroide de la celda P.

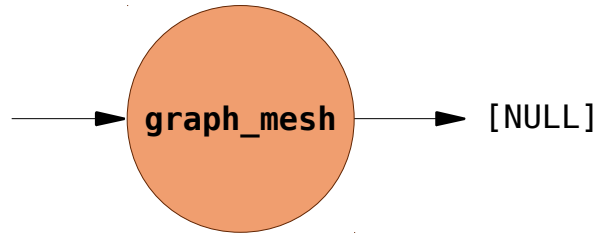
Entrada:

- **cells**: vector de celdas.
- **DIR**: matriz con la información sobre la frontera de tipo Dirichlet.
- **NEU**: matriz con la información sobre la frontera de tipo Neumann.
- **ROB**: matriz con la información sobre la frontera de tipo Robin.
- **PHI**: vector solución. Cada elemento del vector representa un valor escalar asociado al centroide de cada celda de la malla, y su posición dentro del vector depende de cómo se especificó cada celda en **icone**.

Salida:

- **cells**: vector de celdas actualizado.

```
PHI [ncells×nit double]
Q [ncells×(2×nit) double]
xnode [nnodes×2 double]
icone [nelems×4 int]
neighb [ncells×4 int]
cells [ncells×1 double]
DIR [ndir×1 double]
NEU [nneu×1 double]
ROB [nrob×1 double]
model [1×1 struct]
mode [1×1 int]
graph [1×1 int]
```

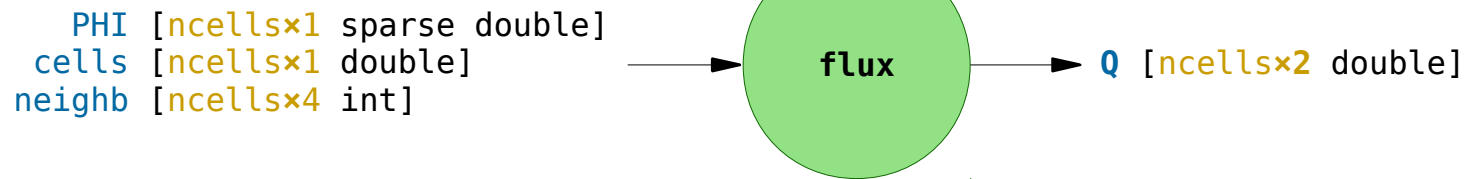


Descripción: módulo para graficar la solución del método numérico. Posee distintas formas de operación.

Entrada:

- **PHI**: matriz solución (**nit** columnas).
- **Q**: vector de flujo de calor (**2×nit** columnas).
- **xnode**: matriz de nodos con pares (x,y) representando las coordenadas de cada nodo de la malla.
- **icone**: matriz de conectividad.
- **cells**: vector de celdas.
- **DIR**: matriz con la información sobre la frontera de tipo Dirichlet.
- **NEU**: matriz con la información sobre la frontera de tipo Neumann.
- **ROB**: matriz con la información sobre la frontera de tipo Robin.
- **model**: modelo de datos.
- **mode**: modo de visualización: 2D, 3D, con o sin malla.
- **graph**: tipo de gráfica.

Salida: ninguna.



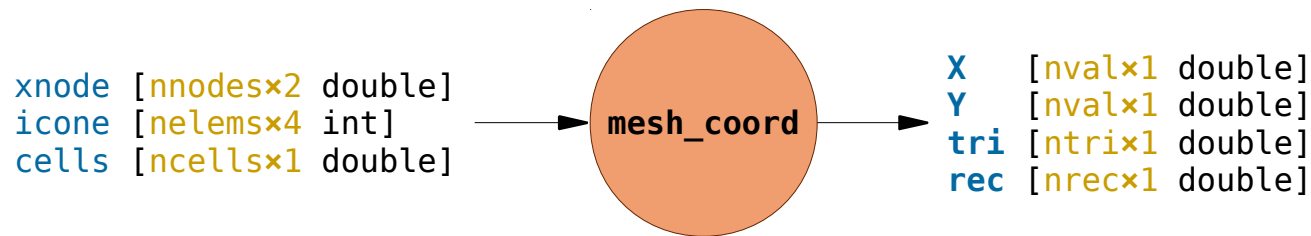
Descripción: módulo calcular el flujo de calor en todo el dominio. Se aplica la Ley de Fourier y se evalúa como fluye el calor en todos los centros de celdas del dominio.

Entrada:

- `PHI`: vector solución. Cada elemento del vector representa un valor escalar asociado a cada celda de la malla, y su posición dentro del vector depende de cómo se especificó en `icone`.
- `neighb`: matriz de vecindad.
- `cells`: vector de celdas.

Salida:

- `Q`: vector de flujo de calor. Se forma de una componente en sentido x, Q_x , y una componente en sentido y, Q_y .



Descripción: módulo ubicar las coordenadas (x,y) de todos los puntos que integrarán la malla para su graficación. Estas posiciones dependen exclusivamente de la malla y se componen de:

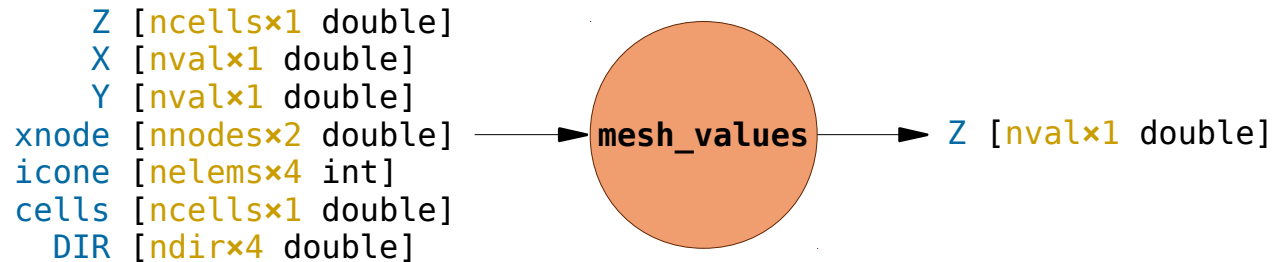
- La ubicación de los centroides de celdas.
- La ubicación de los centros de caras de todas las celdas de borde.
- La ubicación de los nodos que conforman esquinas, es decir, puntos donde se unen dos caras de borde de una misma celda.

Entrada:

- `xnode`: matriz de nodos con pares (x,y) representando las coordenadas de cada nodo de la malla.
- `icone`: matriz de conectividad. Cada fila de la matriz indica la conectividad de un elemento rectangular, comenzando por el extremo inferior izquierdo y recorriendo el elemento en sentido antihorario.
- `cells`: vector de celdas.

Salida:

- `X`: coordenada x de cada valor a graficar.
- `Y`: coordenada y de cada valor a graficar.
- `tri`: conectividad de todos los triángulos en que se particiona la malla. Estos triángulos se utilizan para graficar la superficie (3D) que representa la solución de la ecuación diferencial. Se triangula entre centroides de celdas y entre centros de caras y centroides de celdas.
- `rec`: conectividad de todos los rectángulos en que se particiona la malla. Estos rectángulos se utilizan para resaltar sobre la superficie (3D) la ubicación de todos los puntos utilizados para graficar.



Descripción: módulo para calcular los valores finales de la malla para su graficación. Estos valores dependen exclusivamente de la malla y se componen de:

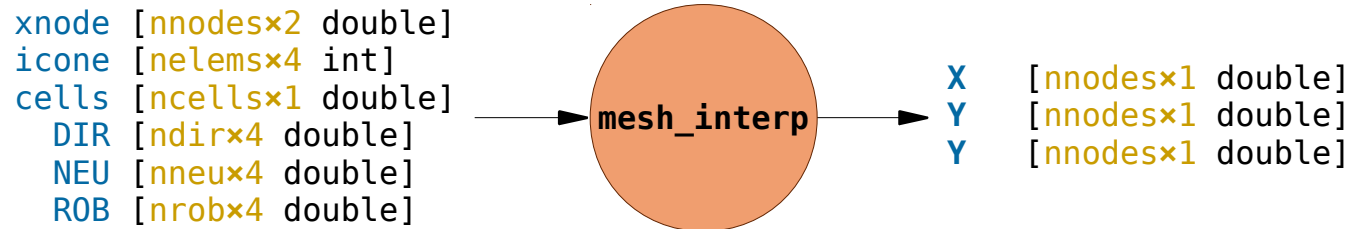
- Valores en los centroides de celdas.
- Valores en los centros de caras de todas las celdas de borde. Estos valores son interpolados utilizando las ecuaciones de FVM para las condiciones de borde.
- Valores interpolados para completar las esquinas, es decir, nodos en donde se unen las dos caras de borde de una celda.

Entrada:

- `Z`: vector solución. Cada elemento del vector representa un valor escalar asociado a cada centroide de celda.
- `X`: coordenada x de cada valor a graficar.
- `Y`: coordenada y de cada valor a graficar.
- `xnode`: matriz de nodos con pares (x,y) representando las coordenadas de cada nodo de la malla.
- `icone`: matriz de conectividad. Cada fila de la matriz indica la conectividad de un elemento rectangular, comenzando por el extremo inferior izquierdo y recorriendo el elemento en sentido antihorario.
- `cells`: vector de celdas.
- `DIR`: matriz con la información sobre la frontera de tipo Dirichlet. Dado que los valores de frontera se dan en las caras de las celdas, esta información permite usar directamente los valores fijados y no hace falta interpolar.

Salida:

- `Z`: vector de valores de la malla. Cada elemento del vector representa un valor escalar asociado a un par (x,y) según la ubicación dada por los vectores `X` y `Y`.



Descripción: módulo para interpolar/extrapolar la solución dada por el método numérico para llevar dicha solución a los nodos de la malla, es decir, se expresan los valores de la solución en cada nodo al igual que en FDM o FEM. Se realizan 3 tipos de interpolaciones/extrapolaciones:

- Bilineal, para hallar el valor de un nodo rodeado de 4 centroides de celdas.
- Baricéntrica, para hallar el valor de un nodo rodeado de 3 centroides de celdas.
- Lineal, para hallar el valor de un nodo rodeado de dos centros de cara de celdas.

Entrada:

- `xnode`: matriz de nodos con pares (x,y) representando las coordenadas de cada nodo de la malla.
- `icone`: matriz de conectividad. Cada fila de la matriz indica la conectividad de un elemento rectangular, comenzando por el extremo inferior izquierdo y recorriendo el elemento en sentido antihorario.
- `cells`: vector de celdas.
- `DIR`: matriz con la información sobre la frontera de tipo Dirichlet.
- `NEU`: matriz con la información sobre la frontera de tipo Neumann.
- `ROB`: matriz con la información sobre la frontera de tipo Robin.

Salida:

- `X`: coordenada x de cada nodo de la malla (proviene de `xnode`).
- `Y`: coordenada y de cada nodo de la malla (proviene de `xnode`).
- `Z`: valor escalar en el punto (x,y) (solución interpolada/extrapolada al nodo de la malla).