





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 nodos)
- neighb**: matriz de vecindad (conectividad de cada nodo con 2, 3 o 4 nodos vecinos)
- K**: matriz del sistema (difusión + reacción)
- F**: vector de flujo térmico.
- PHI**: vector/matriz solución del método numérico (escalar)
- DIR**: matriz de nodos frontera tipo Dirichlet.
- NEU**: matriz de pares de nodos frontera tipo Neumann.
- ROB**: matriz de pares de nodos 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:

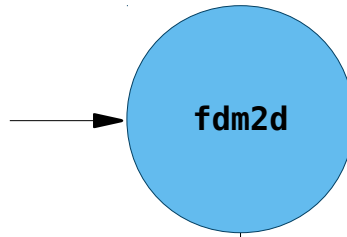
- nnodes**: cantidad total de nodos de la malla.
- nelem**: cantidad total de elementos de la malla.
- ndir**: cantidad de nodos frontera tipo Dirichlet.
- nneu**: cantidad de pares de nodos (lado de un elemento) frontera tipo Neumann.
- nrob**: cantidad de pares de nodos (lado de un elemento) frontera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



model

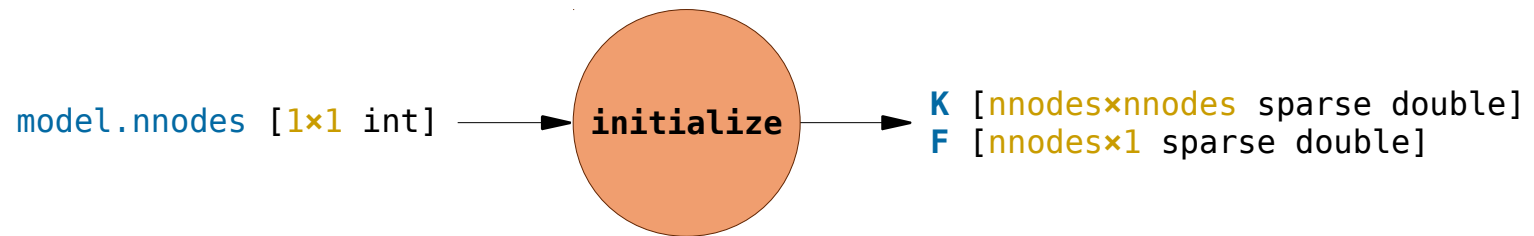
- `nnodes` [1x1 int] - Cantidad de nodos totales de la malla.
- `k` [nnodesx1 double] - Conductividad térmica del material. Permite representar una distribución  $k(x,y)$  en toda la malla.
- `c` [nnodesx1 double] - Constante de reacción del sistema. Representa  $c(x,y)$ . Permite representar una distribución  $c(x,y)$  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 [nnodes×nit sparse double]
Q [nnodes×(2×nit) double]
```

```
% Inicialización de variables principales del sistema
1. [K,F] = fem2d_initialize(model.nnodes);
% Armado de la matriz de vecindad
2. [neighb] = fem2d_neighbors(icone);
% Ensamble de coeficientes del sistema
3. [K,F] = fem2d_gen_system(K,F,xnode,neighb,model);
% Ensamble de nodos frontera Neumann
4. [K,F] = fem2d_neumann(K,F,xnode,neighb,NEU,model);
% Ensamble de nodos frontera Robin
5. [K,F] = fem2d_robin(K,F,xnode,neighb,ROB,model);
% Ensamble de nodos frontera Dirichlet
6. [K,F] = fem2d_dirichlet(K,F,DIR);
% Resolución del sistema lineal de ecuaciones
7. [PHI,Q] = fem2d_solve(K,F,xnode,neighb,DIR,model);
```



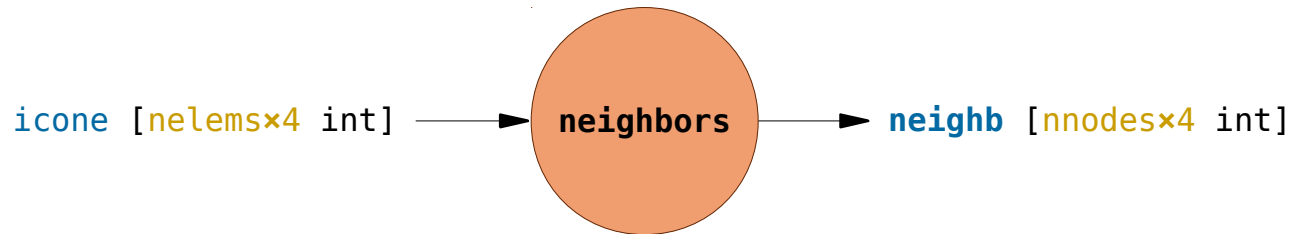
Descripción: módulo para inicializar las variables principales del sistema, las cuales se utilizarán para almacenar los datos calculados y ensamblados por el método numérico. Se inicializan como *sparse* para optimizar el rendimiento general del método numérico.

Entrada:

- `model.nnodes`: cantidad de nodos de la malla.

Salida:

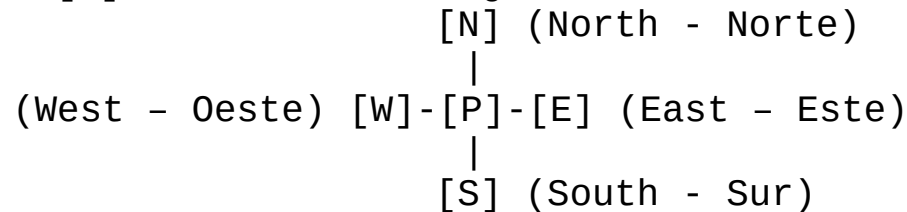
- `K`: matriz del sistema (difusión + reacción)
- `F`: vector de flujo térmico.



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. A partir de esta conectividad (almacenada en **icone**) se arma la matriz de vecindad indicando para cada nodo sus nodos vecinos. Si un nodo no tiene algún vecino, el índice correspondiente se guarda como -1. Existen 3 tipos de nodos:

- Nodos esquina: los cuales se conectan con otros 2 nodos.
- Nodos de borde: los cuales se conectan con otros 3 nodos.
- Nodos interiores: los cuales se conectan con otros 4 nodos.

Para un nodo cualquiera [P] se define la siguiente vecindad:



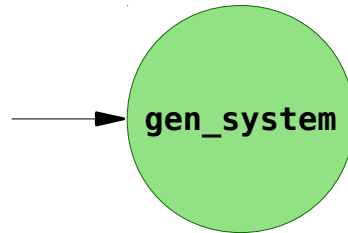
Entrada:

- **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.

```
K [nnodes×nnodes sparse double]
F [nnodes×1 sparse double]
xnode [nnodes×2 double]
neighb [nnodes×4 int]
k [nnodes×1 double]
c [nnodes×1 double]
G [nnodes×1 double]
```



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

Descripción: módulo para ensamblar los términos difusivo, reactivo y fuente de todos los nodos de la malla, generando el *stencil* adecuado dependiendo de si es un nodo interior o de frontera.

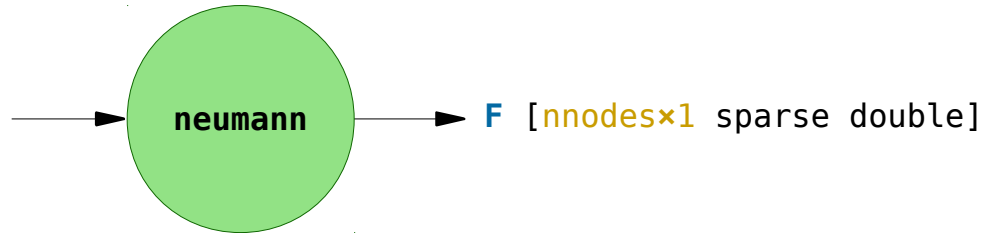
Entrada:

- **K**: matriz del sistema (difusión + reacción)
- **F**: vector de flujo térmico.
- **xnode**: matriz de pares (x,y) representando cada nodo de la malla.
- **neighb**: matriz de vecindad.
- **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  $G(x,y)$ .

Salida:

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

```
F [nnodes×1 sparse double]
xnode [nnodes×2 double]
neighb [nnodes×4 int]
NEU [nneu×3 double]
```



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

Entrada:

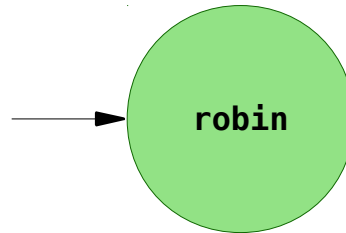
- **F**: vector de flujo térmico.
- **xnode**: matriz de nodos con pares (x,y) representando las coordenadas de cada nodo de la malla.
- **neighb**: matriz de vecindad.
- **NEU**: matriz con la información sobre la frontera de tipo Neumann.
  - Columna 1: índice del nodo donde se aplica la condición 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 [nnodes×nnodes sparse double]
F [nnodes×1 sparse double]
xnode [nnodes×2 double]
neighb [nnodes×4 int]
ROB [nrob×5 double]
```



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

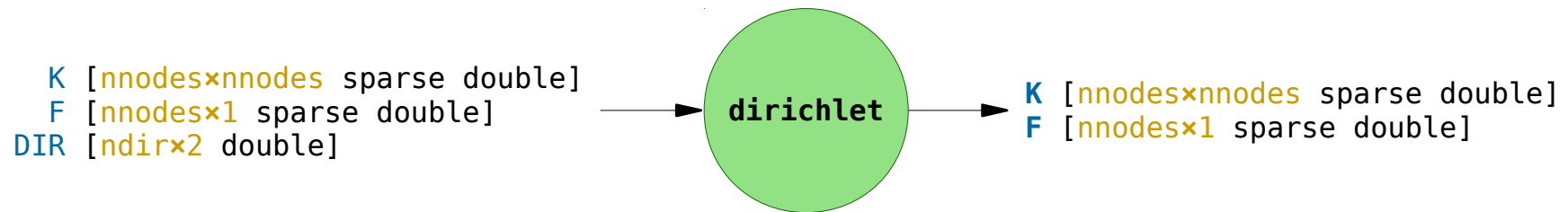
Descripción: módulo para calcular y ensamblar las contribuciones de nodos pertenecientes a fronteras de tipo Robin.

Entrada:

- **K**: matriz del sistema (difusión + reacción)
- **F**: vector de flujo térmico.
- **xnode**: matriz de nodos con pares (x,y) representando las coordenadas de cada nodo de la malla.
- **neighb**: matriz de vecindad.
- **ROB**: matriz con la información sobre la frontera de tipo Robin.
  - Columna 1: índice del nodo 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.



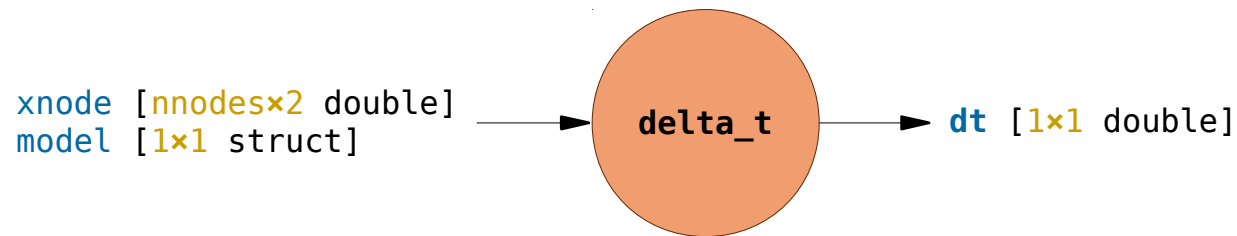
Descripción: módulo para calcular y ensamblar las contribuciones de nodos pertenecientes a fronteras de tipo Dirichlet.

Entrada:

- **K**: matriz del sistema (difusión + reacción)
- **F**: vector de flujo térmico.
- **DIR**: matriz con la información sobre la frontera de tipo Dirichlet.
  - Columna 1: número de nodo.
  - Columna 2: valor en ese nodo (escalar)

Salida:

- **K**: matriz del sistema (difusión + reacción) luego de realizar las simplificaciones que surgen de aplicar la condición de borde Dirichlet.
- **F**: vector de flujo térmico luego de realizar las simplificaciones que surgen de aplicar la condición de borde Dirichlet.



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.

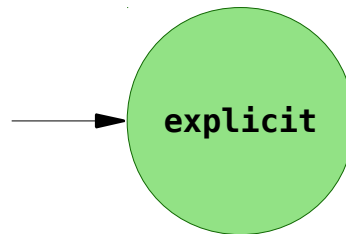
Entrada:

- `xnode`: matriz de pares (x,y) representando cada nodo de la malla.
- `model`: struct con todos los datos del modelo (constantes, esquema numérico, etc.).

Salida:

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

```
K [nnodes×nnodes sparse double]
F [nnodes×1 sparse double]
xnode [nnodes×2 double]
neighb [nnodes×4 int]
model [1×1 struct]
dt [1×1 double]
```



```
PHI [nnodes×(nit+1) sparse double]
Q [nnodes×(2×nit+1) double]
```

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

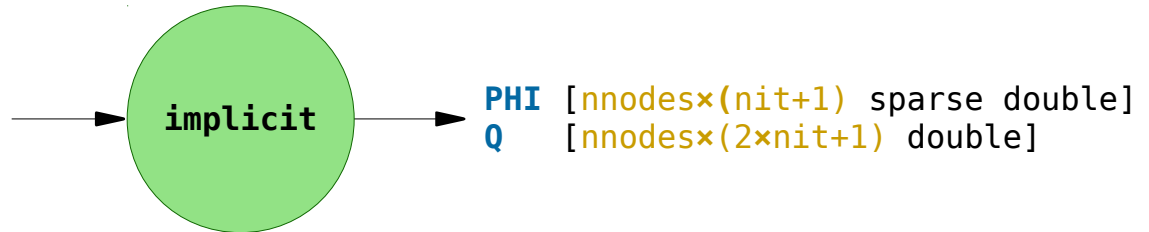
Entrada:

- **K**: matriz del sistema (difusión + reacción)
- **F**: vector de flujo térmico.
- **xnode**: matriz de nodos con pares (x,y) representando las coordenadas de cada nodo de la malla.
- **neighb**: matriz de vecindad.
- **model**: struct con todos los datos del modelo (constantes, esquema numérico, etc.)
- **dt**: paso temporal *crítico* para método explícito.

Salida:

- **PHI**: matriz solución. Cada elemento del vector representa un valor escalar asociado a cada nodo de la malla, y su posición dentro del vector depende de cómo se especificó cada nodo en **xnode**. Cada columna representa una iteración del esquema temporal (en total **nit** columnas).
- **Q**: matriz de flujo de calor. Para cada nodo se halla un vector bidimensional de flujo de calor, representado por un par (Qx,Qy). Cada par de columnas representa una iteración del esquema temporal (en total **2×nit** columnas).

```
K [nnodes×nnodes sparse double]
F [nnodes×1 sparse double]
xnode [nnodes×2 double]
neighb [nnodes×4 int]
model [1×1 struct]
dt [1×1 double]
```



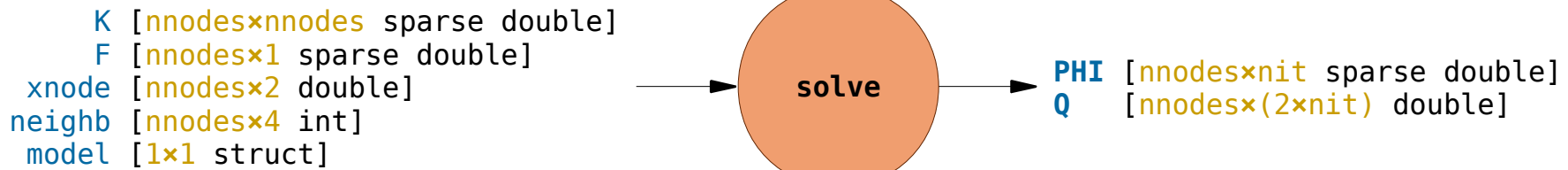
Descripción: módulo para resolver el sistema lineal de ecuaciones utilizando esquema temporal *implícito*. El primer valor (primer columna) es la condición inicial.

Entrada:

- **K**: matriz del sistema (difusión + reacción)
- **F**: vector de flujo térmico.
- **xnode**: matriz de nodos con pares (x,y) representando las coordenadas de cada nodo de la malla.
- **neighb**: matriz de vecindad.
- **model**: struct con todos los datos del modelo (constantes, esquema numérico, etc.)
- **dt**: paso temporal *arbitrario* para método explícito.

Salida:

- **PHI**: matriz solución. Cada elemento del vector representa un valor escalar asociado a cada nodo de la malla, y su posición dentro del vector depende de cómo se especificó cada nodo en **xnode**. Cada columna representa una iteración del esquema temporal (en total **nit** columnas).
- **Q**: matriz de flujo de calor. Para cada nodo se halla un vector bidimensional de flujo de calor, representado por un par (Qx,Qy). Cada par de columnas representa una iteración del esquema temporal (en total **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:

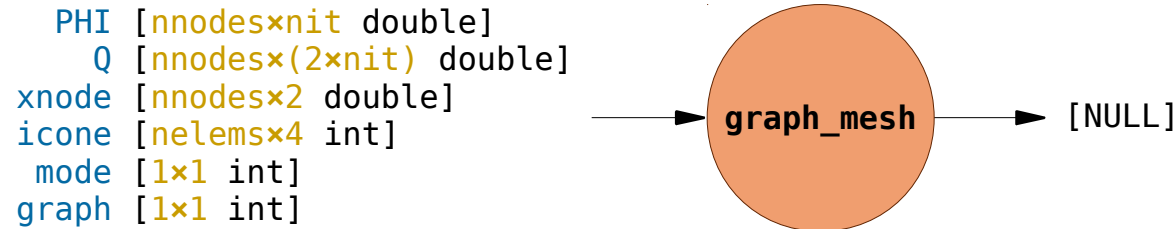
- Sin la aplicación de esquemas temporales, es decir, la solución del sistema en estado estacionario. Resolución por método directo.
- Aplicación de esquemas temporales, a saber: método explícito y método implícito. Se evalúa la evolución temporal del sistema desde un estado inicial conocido hasta un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

Entrada:

- `K`: matriz del sistema (difusión + reacción).
- `F`: vector de flujo térmico.
- `xnode`: matriz de nodos con pares (x,y) representando las coordenadas de cada nodo de la malla.
- `neighb`: matriz de vecindad.
- `model`: struct con todos los datos del modelo (constantes, esquema numérico, etc.).

Salida:

- `PHI`: matriz solución. Cada elemento del vector representa un valor escalar asociado a cada nodo de la malla, y su posición dentro del vector depende de cómo se especificó cada nodo en `xnode`. Cada columna representa una iteración del esquema temporal (en total `nit` columnas).
- `Q`: matriz de flujo de calor. Para cada nodo se halla un vector bidimensional de flujo de calor, representado por un par ( $Q_x, Q_y$ ). Cada par de columnas representa una iteración del esquema temporal (en total `2×nit` columnas).

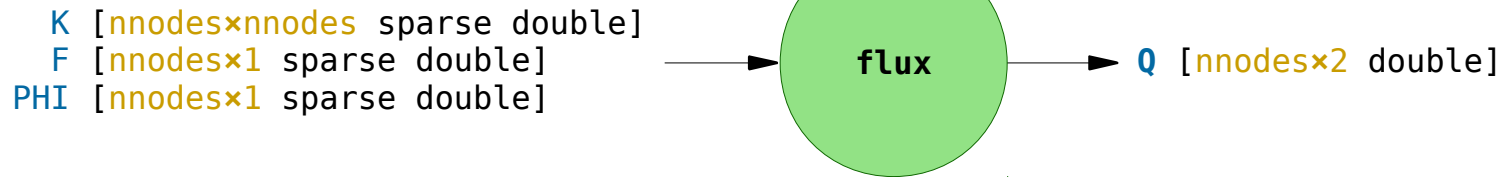


Descripción: módulo para graficar la solución del método numérico. Posee distintas formas de operación y diferentes tipos de gráficas.

Entrada:

- **PHI:** matriz solución. Cada elemento del vector representa un valor escalar asociado a cada nodo de la malla, y su posición dentro del vector depende de cómo se especificó cada nodo en `xnode`. Cada columna representa una iteración del esquema temporal (en total `nit` columnas).
- **Q:** matriz de flujo de calor. Para cada nodo se halla un vector bidimensional de flujo de calor, representado por un par ( $Q_x, Q_y$ ). Cada par de columnas representa una iteración del esquema temporal (en total `2×nit` columnas).
- **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.
- **mode:** modo de visualización: 2D, 3D, con o sin malla.
- **graph:** tipo de gráfica:
  - Temperatura (escalar)
  - Flujo de calor (vectorial)
  - Flujo eje-x (escalar)
  - Flujo eje-y (escalar)
  - Magnitud de flujo (escalar).

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 puntos (nodos) del dominio.

Entrada:

- `K`: matriz del sistema (difusión + reacción)
- `F`: vector de flujo térmico.
- `PHI`: vector solución. Cada elemento del vector representa un valor escalar asociado a cada nodo de la malla, y su posición dentro del vector depende de cómo se especificó cada nodo en `xnode`.

Salida:

- `Q`: vector de flujo de calor. Para cada nodo se halla un vector bidimensional de flujo de calor, representado por un par ( $Q_x, Q_y$ )