ilu(0) Algorithm

georg.may

April 2023

Algorithm 1 ILU(0) in MSR format. No zero elements on diagonal.

Input: V, JM (Matrix in MSR Format. Size of both arrays: nz)
Output: PV,PJM (Preconditioner in MSR format. Size of both arrays: nz)
JD (Size: n. Pointer to first non-zero element in upper triangle for each row)

```
1:  JR(1:n) = 0                          ▷ Initialize reverse pointers. (Column index to index in PJM,PV)
2:  JD(1:n) = 0                          ▷ Initialize pointers to upper triangle
3:  PJM(1:nz) = JM(1:nz)                 ▷ Copy column pointers. These don't change
4:  PV(1:nz) = V(1:nz)                   ▷ Copy data pointer. Will be modified in-place below
5:  for i=1,n do                         ▷ Loop over Rows
6:      JR(i) = i                        ▷ Reverse pointer is identity for diagonal
7:      for j=JM(i),JM(i+1)-1 do         ▷ Loop over non-zero off-diagonal elements in row i
8:          jc := JM(j)                  ▷ Column index: PV(j) = a(i,jc)
9:          JR(jc):= j                   ▷ Reverse pointer (column index to index in PJM,PV)
10:         if  jc > i and JD(i) = 0 then
11:             JD(i) = j                ▷ Pointer to upper triangle (i^th row)
12:         end if
13:     end for
14:     if JD(i) = 0 then
15:         JD(i) = j                    ▷ If no elements in upper triangle
16:     end if
17:     for j=JM(i),JD(i)-1 do           ▷ non-zero elem. in lower triangle of row i
18:         jc:=JM(j)                    ▷ column index, P(j)=a(i,jc). Note: jc<i
19:         PV(j) ← PV(j)/PV(jc)         ▷ P(jc)=a(jc,jc). Overwrite with L-factor
20:         for jj=JD(jc),JM(jc+1)-1 do  ▷ all non-zero a(jc,k) (jc < k=JM(jj))
21:             jk := JR(JM(jj))         ▷ Points back to a(i,k) (k=JM(jj)
22:             if jk ≠ 0 then           ▷ Could be zero, as a(i,k)=0 is possible
23:                 PV(jk) ← PV(jk) - PV(j)*PV(jj)   ▷ PV(jj)=a(jc,k) (k=JM(jj))
24:             end if
25:         end for
26:     end for
27:     JR(i) = 0                        ▷ Reset reverse pointers
28:     for j=JM(i),JM(i+1)-1 do
29:         JR(JM(j)) = 0
30:     end for
31: end for
```