

Algoritm

```

conf(dom, pcte, F) :=
  (
    (
      x0
      xN
      y0
      yN
    ) ← dom
    (
      nh
      mh
      nv
      mv
    ) ← pcte
    (
      pashg
      pashf
      pasvg
      pasvf
    ) ← (
      round( (yN - y0) / mh, 5 )
      round( (xN - x0) / nh, 5 )
      round( (xN - x0) / mv, 5 )
      round( (yN - y0) / nv, 5 )
    )
    for v ∈ 0..mh - 0
      |
      g ← y0 + pashg·v
      for h ∈ 0..nh - 0
        |
        f ← x0 + pashf·h
        Z ← F(f + gi)
        (
          Xhh,v
          Yhh,v
        ) ← (
          Re(Z)
          Im(Z)
        )
    for h ∈ 0..mv - 0
      |
      g ← x0 + pasvg·h
      for v ∈ 0..nv - 0
        |
        f ← y0 + pasvf·v
        Z ← F(g + fi)
        (
          Xvv,h
          Yvv,h
        ) ← (
          Re(Z)
          Im(Z)
        )
    (
      Xh
      Yh
      Xv
      Yv
    )
  )
  
```

Note Algoritm

- $x0...$ parametrii ce cuprind valori limita ce stabilesc domeniul
- $nh..$ număr de linii și număr de puncte pe linie, atât pentru grila de linii orizontale cât și pentru grila de linii verticale
- *pashf și pasvf* (pas fin) - pas pentru punctele ce construiesc o linie verticală sau orizontală
- *pashg și pasvg* (pas grosier) - pas sau distanța între linii
- se utilizează $mh-0, nh-0..$ la toate *for-le* ca să traseze și ultima linie
- *for h și for v* trasează linii orizontale
- *for h* construiește linia orizontală, iar *for v* trece de la o linie la alta
- *f și g* - variabile cu contor ce fac să se parcurgă domeniul complex
- Z și $F(z)$, transformare domeniul complex prin funcția F
- se reține valorile variabilei complexe Z în perechea de vectori $Xh Yh$
- *for v și for h* trasează linii verticale
- *for h* construiește linia verticală iar *for v* trece de la o linie la alta
- *f și g* - variabile cu contor ce fac să se parcurgă domeniul complex
- Z și $F(z)$, transformare domeniul complex prin funcția F
- se reține valorile variabilei complexe Z în perechea de vectori $Xv Yv$