Fonction de répartition gaussienne et fonctions annexes

par Charles Hubert

Fonction de répartition gaussienne

Pour calculer la probabilité qu'une variable aléatoire gaussienne soit située dans un intervalle, ou pour faire le calcul inverse, on doit utiliser la fonction de répartition de cette variable. La fonction de répartition d'une variable gaussienne normalisée X (moyenne = 0, écart-type = 1) est

$$\mathrm{prob}(X \le x) = \mathrm{Gaus}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp \frac{-t^2}{2}\, dt$$

En principe on peut exprimer cette fonction à partir de la fonction d'erreur

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} \exp(-t^2)\,dt$$

mais alors il faut introduire un coefficient sur la variable et un autre sur la fonction. Pour éviter ces deux coefficients, on peut utiliser la fonction d'erreur modifiée

$$\mathrm{Errm}(x) = \frac{1}{\sqrt{2\pi}} \int_{0}^{x} \exp \frac{-t^2}{2}\, dt \quad \Rightarrow \quad \mathrm{Gaus}(x) = \frac{1}{2} + \mathrm{Errm}(x)$$

Ces deux fonctions s'utilisent par les instructions

```
y←Errm x          y←Gaus x
```

La fonction Errm est très précise pour x voisin de 0 ; la fonction Gaus est très précise pour x très négatif.

La fonction Gaus n'est pratiquement utilisable que dans l'intervalle approximatif [-37.519, 8.29] ; en dehors de cet intervalle les valeurs qu'elle devrait fournir ne sont plus représentables dans le format des nombres réels du PC :

```
      0 0 ¯1 ¯1 +Gaus ¯37.7 ¯37.519 8.29 8.3
0 2.256992114E¯308 ¯1.110223025E¯16 0
```

On a donc créé la fonction

```
y←LogGaus x
```

qui calcule le logarithme népérien de Gaus, ce qui permet d'élargir l'intervalle précédent, si on sait se débrouiller avec ce logarithme :

```
      LogGaus ¯50 10
¯1254.831361 ¯7.619853024E¯24
```

Calculons

```
      y←(0,⍟10)⊤LogGaus ¯50
```

alors, si le PC avait une dynamique plus large, il trouverait pour Gaus(-50)

```
      (⍕*y[1]),'E',⍕y[0]  ⍝ ⎕io=0
1.080597947E¯545
```

Par ailleurs

$$\mathrm{Gaus}(10) = 1-7.619853024\times 10^{-24}$$

Ces trois fonctions s'aident de la fonction d'erreur auxiliaire définie par

$$\mathrm{Errx}(x) = \frac{1}{\sqrt{2\pi}} \exp\frac{x^2}{2} \int_x^{\infty} \exp\frac{-t^2}{2}\,dt \qquad \text{pour} \qquad x > 0$$

et

$$\mathrm{Errx}(0) = 0 \qquad \mathrm{Errx}(-x) = -\,\mathrm{Errx}(x)$$

Ces quatre fonctions, non exprimables par un nombre fini d'opérations avec des fonctions usuelles, sont calculées à partir de développements en séries limités à un nombre suffisant de termes.

Elles acceptent des tableaux gigognes ne contenant que des nombres (bien sûr) et les traitent à la manière des fonctions scalaires.

Les fonctions inverses

Ces fonctions inverses sont définies par

```
y = Agaus(x)      <=>  x = Gaus(y)
y = Aerrm(x)      <=>  x = Errm(y)
y = AlogGaus(x)   <=>  x = LogGaus(y)
```

Chacune de ces trois fonctions est calculée en résolvant numériquement l'équation de définition (à droite ci-dessus) par la méthode d'approximations successives de Newton, en démarrant l'itération avec une valeur donnée par une formule approchée convenable.

Comme les fonctions directes elles acceptent des tableaux gigognes ne contenant que des nombres (bien sûr) et les traitent à la manière des fonctions scalaires. Exemples :

```
      DISP y←Aerrm x← 0.41 0 ¯1E¯7 (¯0.37 0.104)
┌→─────────────────────────────────────────────────────────────┐
│ 1.340755034 0 ¯2.506628275E¯7 ┌→──────────────────────────┐ │
│                               │¯1.126391129 0.2637143982  │ │
│                               └~──────────────────────────┘ │
└≡─────────────────────────────────────────────────────────────┘
      DISP x-Errm y
┌→───────────────────────────┐
│ 0 0 0 ┌→──────────────────┐ │
│       │0 ¯1.387778781E¯17 │ │
│       └~──────────────────┘ │
└≡───────────────────────────┘
      DISP y←Agaus x← 0.73 0.38 1E¯8 (0.26 0.11)
┌→──────────────────────────────────────────────────────────────────────────┐
│ 0.612812991 ¯0.3054807881 ¯5.612001244 ┌→──────────────────────────────┐ │
│                                        │¯0.6433454054 ¯1.22652812     │ │
│                                        └~──────────────────────────────┘ │
└≡──────────────────────────────────────────────────────────────────────────┘
      DISP x-Gaus y
┌→──────────────────────────────────────────┐
│ 0 0 2.64697796E¯23 ┌→───────────────────┐ │
│                    │¯1.110223025E¯16 0  │ │
│                    └~───────────────────┘ │
└≡──────────────────────────────────────────┘
      DISP y←AlogGaus x←- 100 0.38 0.3 (0.25 1E¯20)
┌→───────────────────────────────────────────────────────────────────────┐
│ ¯13.88847603 0.4785241616 0.6458699862 ┌→─────────────────────────┐ │
│                                        │0.7681493953 9.26234009   │ │
│                                        └~─────────────────────────┘ │
└≡───────────────────────────────────────────────────────────────────────┘
      DISP x-LogGaus y
┌→─────────────────────────────────────────────────────────────┐
│ ¯1.421085472E¯14 0 0 ┌→─────────────────────────────────┐ │
│                      │5.551115123E¯17 6.770847461E¯35   │ │
│                      └~─────────────────────────────────┘ │
└≡─────────────────────────────────────────────────────────────┘
```

Les fonctions

Conçues pour APL*PLUS elles gèrent les erreurs par ⎕elx et ⎕error ; pour les utiliser avec un autre APL il faut peut-être réadapter les instructions correspondantes.

```
    ∇  x←Errm x;⎕elx;b;y;z
[1]    ⍝∇                 x     2
[2]    ⍝∇          1     ⌠      t
[3]    ⍝∇  Errm(x) = ───  │ exp ── dt
[4]    ⍝∇         ____   ⌡     -2
[5]    ⍝∇        √2π    0
[6]     ⎕elx←'⎕error((⎕dm⍳⎕tcnl)-⎕io)⍴⎕dm'
[7]     b←0≥1-|y←10⌊¯10⌈∊x
[8]     (b/∊x)←(0.5××z)-(*z×z×¯0.5)×Errx z←b/y
[9]     b←~b
[10]    (b/∊x)←z×(z×z←b/y)⊥¨⊂ ¯9.027165E¯11 2.22712404E¯9
       ¯4.117317843E¯8 6.6593178058E¯7 ¯9.44463986185E¯6
       1.1543468324926E¯4 ¯1.18732821481152E¯3 9.97355700998374E¯3
       ¯0.06649038006690387 0.39894228040143272
    ∇
```

```
    ∇  x←Gaus x;⎕elx
[1]    ⍝∇                          x        2
[2]    ⍝∇                    1    ⌠          t
[3]    ⍝∇  Gaus(x) =  ─────  ⎮  exp ── dt
[4]    ⍝∇                  ____  ⌡          -2
[5]    ⍝∇                ╲╱2π   -∞
[6]     ⎕elx←'⎕error((⎕dm⍳⎕tcnl)-⎕io)⍴⎕dm'
[7]     x←(0.5×1+×x)-(Errx x)×*x×¯0.5×x←¯38⌈38⌊x
    ∇
  ∇  l←LogGaus x;⎕elx;b;y
[1]    ⍝∇  LogGaus(x) = log Gaus(x)
[2]     ⎕elx←'⎕error((⎕dm⍳⎕tcnl)-⎕io)⍴⎕dm'
[3]     l←x ⋄ (∊l)←⍟0.5
[4]     y←y÷¯2+y←(*¯0.5×(38⌊y)*2)×Errx y←(b←0<x)/x←∊x
[5]    ⍝ (b/∊l)←¯7○y
[6]     (b/∊l)←y×(y×y)⍳¨⊂ 0.16175895405034462 0.0879491364854064
    0.12002262767455414 0.13314013605953826 0.1538557837098355
    0.18181789128857964 0.22222222723694516 0.28571428567047512
    0.40000000000014776 0.66666666666666648 2
[7]     y←(b←0>x)/x
[8]     (b/∊l)←(⍟-Errx y)-y×y×0.5
  ∇
  ∇  x←Errx x;⎕elx;b;y;z
[1]    ⍝∇                                  2   ∞       2
[2]    ⍝∇                        1         x  ⌠        t
[3]    ⍝∇  x > 0 => Errx(x) = ───── exp ── ⎮ exp ── dt
[4]    ⍝∇                      ____      2   ⌡       -2
[5]    ⍝∇                    ╲╱2π            x
[6]    ⍝∇
[7]    ⍝∇  Errx(0) = 0     Errx(-x) = - Errx(x)
[8]     ⎕elx←'⎕error((⎕dm⍳⎕tcnl)-⎕io)⍴⎕dm'
[9]     b←0<4-|y←∊x
[10]    (b/∊x)←(×z)×(¯2+|z←b/y)⍳¨⊂ ¯1.0437777E¯17 6.3975234E¯17
    ¯1.24862712E¯16 7.5241523E¯16 ¯7.32851775E¯15 4.207750656E¯14
    ¯2.1885531062E¯13 1.2132393092E¯12 ¯6.6651885768E¯12
    3.5424725101E¯11 ¯1.83930297498E¯10 9.33994477264E¯10
    ¯4.627383889682E¯9 2.233184547739E¯8 ¯1.0481909826082E¯7
    4.7761899139435E¯7 ¯2.10824858044462E¯6 8.9926881034069E¯6
    ¯3.69596131373338E¯5 1.458607306196509E¯4 ¯5.50438753299678E¯4
    1.976041890443334E¯3 ¯6.704277547367718E¯3 0.02131272265649356
    ¯0.06273827795509149 0.1681020012231706
[11]    b←~b
[12]    (b/∊x)←((z*¯2)⍳¨⊂ ¯1.0327011918E14 6.26643636334E13
    ¯1.779711897546E13 3160779839356 ¯397116306434.5 38083757132.75
    ¯2965955821.1292 199857452.80903 ¯12525059.185852
    789375.7952537 ¯53677.2565421 4144.9016248783 ¯376.98699707507
    41.888881684539 ¯5.9841340536772 1.19682684099072
    ¯0.3989422804013132 0.39894228040143264)÷z←b/y
  ∇
  ∇  x←Aerrm x;⎕elx;⎕io;c;e;g;i;m;y;z
[1]    ⍝∇  Aerrm(y) = x  =>  y = Errm(x)
[2]     ⎕io←0 ⋄ ⎕elx←'⎕error(⎕dm⍳⎕tcnl)⍴⎕dm'
[3]     m←7 ⋄ c←(○2)*0.5
[4]     y←(¯2×⍟0.5-|g←∊x)*0.5
[5]     y←y+(÷y)⍳¨⊂ ¯23.6132 85.304 ¯128.026 103.886 ¯50.2113 15.7268
    ¯4.38859 0
[6]     y←y××g
[7]     i←⍳⍴g
[8]    A:y[i]←z+e←c×(*z×z×0.5)×g[i]-Errm z←y[i]
[9]     i←((|e)>1E¯14×|z)/i
[10]    →((0≤m←m-1)∧0<⍴i)⍴A ⋄ (∊x)←y
  ∇
```

```
      ∇  x←Agaus x;⎕elx;⎕io;c;e;g;i;m;y;z
 [1]    ⍝∇  Agaus(y) = x  =>  y = Gaus(x)
 [2]     ⎕io←0 ⋄ ⎕elx←'⎕error(⎕dm⍳⎕tcnl)⍴⎕dm'
 [3]     m←7 ⋄ c←(○2)*0.5 ⋄ g←∊x
 [4]     y←(¯2×⍟g⌊1-g)*0.5
 [5]     y←y+(÷y)⊥¨⊂ ¯82.637 273.36 ¯366.86 257.22 ¯102.05 24.332
        ¯4.9237 0
 [6]     y←y××g-0.5
 [7]     i←⍳⍴g
 [8]    A:y[i]←z+e←(c×*z×z×0.5)×g[i]-Gaus z←y[i]
 [9]     i←((|e)>1E¯14×1⌈|z)/i
 [10]    →((0≤m←m-1)∧0<⍴i)⍴A ⋄ (∊x)←y
      ∇
    ∇  x←AlogGaus x;⎕elx;⎕io;c;d;e;i;m;n;p;w;y;z
[1]    ⍝∇  AlogGaus(y) = x  =>  y = log Gaus(x)
[2]     ⎕io←0 ⋄ ⎕elx←'⎕error(⎕dm⍳⎕tcnl)⍴⎕dm'
[3]     m←7 ⋄ c←(○2)*0.5 ⋄ y←w←∊x
[4]     p←~n←w<⍟0.5
[5]     z←(¯2×⍟-p/w)*0.5
[6]     (p/y)←z+(÷z)⊥¨⊂ 8.3947 ¯31.194 44.562 ¯31.621 13.321 ¯4.3468 0
[7]     z←(¯2×n/w)*0.5
[8]     (n/y)←((÷z)⊥¨⊂ 88.002 ¯290.17 387.76 ¯270.29 106.33 ¯25.008
       4.9628 0)-z
[9]     i←⍳⍴w
[10]   A:p←~n←0>z←d←y[i]
[11]    (p/d)←(Gaus p/z)×*0.5×(p/z)*2 ⋄ (n/d)←Errx-n/z
[12]    y[i]←z+e←c×d×w[i]-LogGaus z
[13]    i←((|e)>1E¯16×1⌈|z)/i
[14]    →((0≤m←m-1)∧0<⍴i)⍴A ⋄ (∊x)←y
    ∇
```