Solutions de l'équation $z^x = x^p$ quelque soit p > 0.

Soit $f(x) = x^{px}$. Soient $\Phi_0(x)$ et $\Phi(x)$ les fonctions réciproques de f(x) pour $0 < x < \frac{1}{e}$ et pour $x > \frac{1}{e}$.

On a : $f(x) = x^{px} = e^{px\ln(x)}$ => $\ln(f(x)) = px\ln(x)$
En fonction réciproque : $\ln(x) = p\Phi(x)\ln(\Phi(x))$

Soient W(x) et $W_{-1}(x)$ les deux fonctions bijectives de Lambert définies par l'équivalence : $z = we^w$ <==> $w = W(z)$.

Soient $w = \ln(\Phi(x))$ et $z = \ln(x)/p$
=> $\ln(x)/p = z = \Phi(x)\ln(\Phi(x)) = \ln(\Phi(x))e^{\ln(\Phi(x))} = we^w$
=> $w = W(z)$
=> $\ln(\Phi(x)) = w = W(z) = W(\ln(x)/p)$
=> $\Phi(x) = \ln(x)/(p\ln(\Phi(x))) = \ln(x)/(pW(\ln(x)/p))$

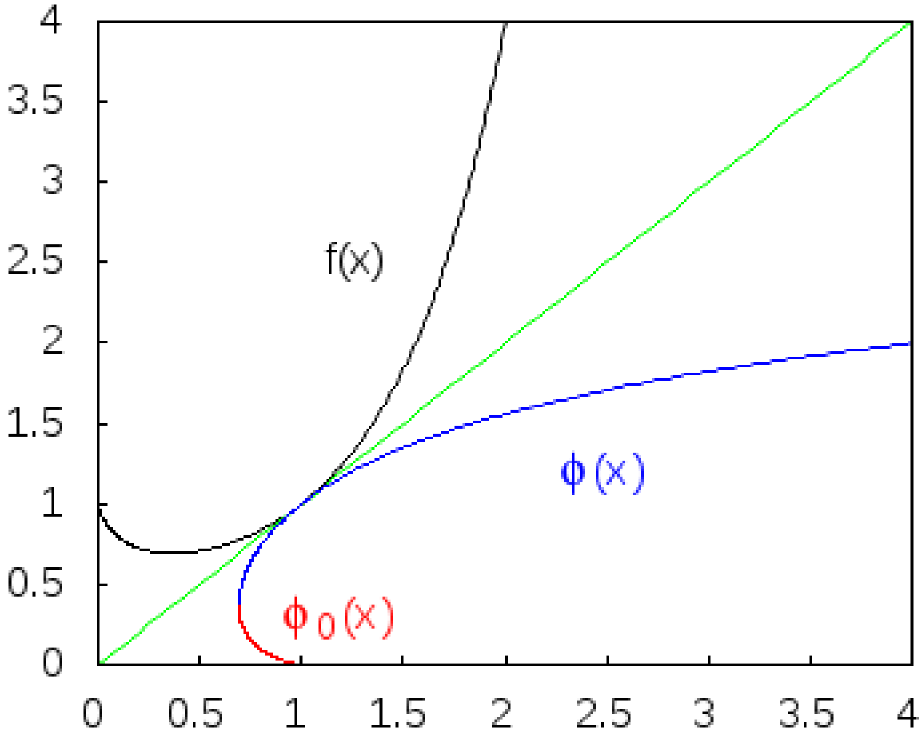
Et de même : $\Phi_0(x) = \ln(x)/(pW_{-1}(\ln(x)/p))$

On a : $x = \Phi(x)^{p\Phi(x)}$ => $\Phi(1/z)^{p\Phi(1/z)} = 1/z$
=> $z = 1/\Phi(1/z)^{p\Phi(1/z)} = (1/\Phi(1/z))^{p\Phi(1/z)}$
Soit $x = 1/\Phi(1/z)$ => $z = x^{p\Phi(1/z)} = x^{p/x}$ => $z^x = x^p$

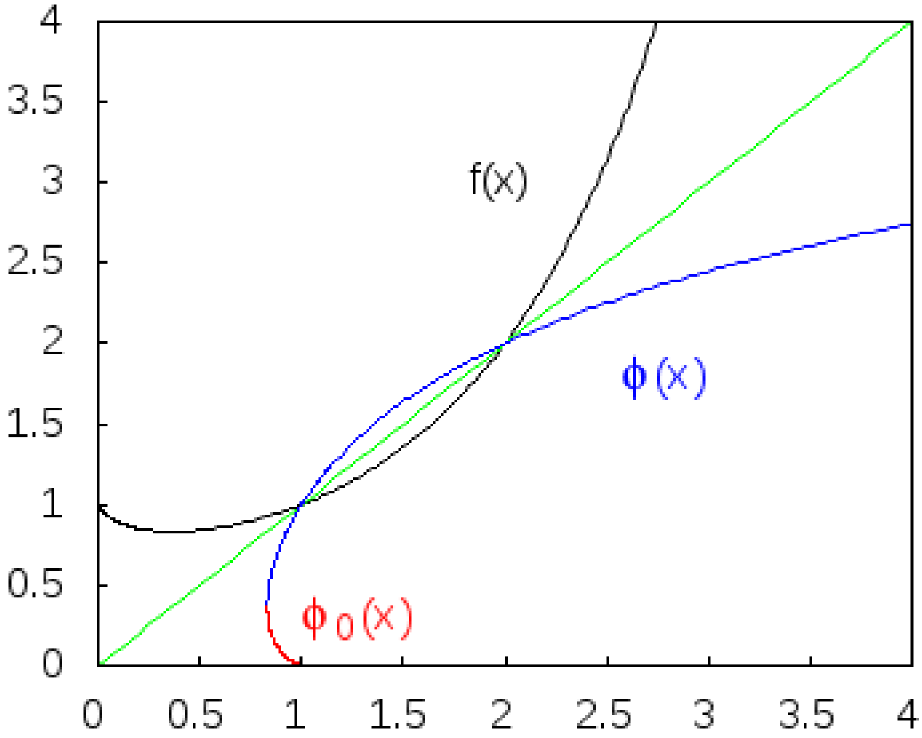
Et de même : $x_0 = 1/\Phi_0(1/z)$ => $z^{x_0} = (x_0)^p$

Pour $(\frac{1}{e})^{\frac{p}{e}} < x < 1$: $\Phi_0(x)$ est décroissante de $\frac{1}{e}$ à 0.
Pour $x > (\frac{1}{e})^{\frac{p}{e}}$: $\Phi(x)$ est croissante de $\frac{1}{e}$ à ∞.

Voici les fonctions f(x), $\Phi_0(x)$ et $\Phi(x)$ pour p = 1.



Voici les fonctions f(x), $\Phi_0(x)$ et $\Phi(x)$ pour p = 0.5.



Soit $n = (e^{p})^{\frac{1}{e}}$.

Pour $0 < z < 1$ l'équation $z^x = x^p$ a une solution : $x = 1/\Phi(1/z)$
Pour $1 < z < n$ l'équation $z^x = x^p$ a les deux solutions x et x_0.
Pour $z > n$: $1/z < \left(\frac{1}{e}\right)^{\frac{p}{e}}$ => l'équation $z^x = x^p$ n'a aucune solution réelle.

Pour p = 0.5 : n ≈ 1.20194337 et pour p = 2 : n ≈ 2.08706523.
Pour p = 0.5 et z = 1.1 : les deux solutions de l'équation $z^x = x^p$ sont : x = 1.2751596577217708 et x_0 = 13.750075318060432.
On a toujours : $x < e < x_0$ sauf pour z = n : $x = x_0 = e$.

Avec z et x donnés on a : $p = \frac{\ln(z^x)}{\ln(x)}$
Avec p et x donnés on a : $z = (x^p)^{\frac{1}{x}}$
Avec p et z donnés on a : $x = 1/\Phi(1/z)$ et $x_0 = 1/\Phi_0(1/z)$

Maintenant que l'on sait calculer, quand elles existent, la ou les deux solutions de $z^x = x^p$: pour utiliser les fonctions $\Phi_0(x)$ et $\Phi(x)$ on peut utiliser un logiciel de calcul formel qui connaît déjà les fonctions W(x) et $W_{-1}(x)$ de Lambert ou bien les programmer en C++ avec des variables de type double.

En allant à : http://www.dma.ufv.br/maxima/index.php avec les instructions Maxima suivantes où y(x) = $\Phi(x)$ et y0(x) = $\Phi_0(x)$ on peut écrire par exemple :

```
[e : 2.7182818284590452536];
[y(x) := log(x)/(p*lambert_w(log(x)/p))];
[y0(x) := log(x)/(p*generalized_lambert_w(-1,log(x)/p))];
[p : 2];
[n : (e^p)^(1/e)];
[z : 2.0];
[x : 1/y(1/z)];
```

cabs(z^x - x^p);
cabs(y(x)^(p*y(x)) - x);
[x0 : 1/y0(1/z)];
cabs(z^x0 - x0^p);
[x0 : 0.85];
cabs(y0(x0)^(p*y0(x0)) - x0);

On peut ensuite modifier : [p : 2]; ou [z : 2.0];.

Et voici en C++ un petit programme très simple qui calcule quand elles existent la ou les solutions de l'équation $z^x = x^p$ en utilisant pour $\Phi(x)$ et $\Phi_0(x)$ une méthode de dichotomie.

```
#include <iostream>
#include <iomanip>
#include <cmath>

//  Calcul de y(p,x) = m  <==>  m^pm = x
//  xmin = (1/e)^(p/e)
//  ymin = y(p,xmin) = 1/e = 0.36787944117144232 = amin
//  bmax = 143.0 <==> xmax = bmax^bmax voisin de : 1.633e+308
double y(double p, double x) {
    double e = 2.718281828459045235;
    if(x < pow(1.0/e, p/e)) return 0.0;
    double a = 0.36787944117144232;
    double b = 143.0;
    double m, mm, n=0.0;
    do {
        m = (a+b)/2.0;
        if(m == n) break;
        mm = pow(m,p*m);
        if(mm > x) b = m;
        if(mm < x) a = m;
        n = m;
    }while((abs(mm-x) > 0.0) && (b-a > 0.0));
    return m;
}

//  Calcul de y0(p,x) = m  <==>  m^pm = x
//  xmin = (1/e)^(p/e)
//  ymax = y0(xmin) = 1/e = 0.36787944117144232
//  ymin = y0(1) = 0.0
//  amin doit être très petit et bmax = 2.0/e = 0.735758882342885
double y0(double p,double x) {
    double e = 2.718281828459045235;
    if(x < pow(1.0/e, p/e) || x >= 1.0) return 0.0;
    double a = 0.0000000000000000000000001;
    if(pow(a,a) < x) return 0.0;
    double b = 0.735758882342885;
    double m, mm, n=0.0;
```

```
    do {
        m = (a+b)/2.0;
        if(m == n) break;
        mm = pow(m,p*m);
        if(mm > x) a = m;
        if(mm < x) b = m;
        n = m;
    }while((abs(mm-x) > 0.0) && (b-a > 0.0));
    return m;
}

int main() {
    double p;
    std::cout << std::endl << "Entrez p : ";
    std::cin >> p;
    if(p <= 0.0) return 0;
    double e = 2.718281828459045235;
    double n = pow(e,p/e);
    std::cout << std::endl << "=> n = " << std::setprecision(15)
              << n << std::endl;
    while(true) {
        double z;
        std::cout << std::endl << "Entrez z : ";
        std::cin >> z;
        if(z <= 0.0) break;
        double x = 1.0/y(p,1.0/z);
        double x0 = 1.0/y0(p,1.0/z);
        if(z < 1.0) std::cout << std::setprecision(15) << std::endl
                              << "x = " << x << std::endl;
        else if(z <= n) std::cout << std::setprecision(15) << std::endl
                        << "x = " << x << " et x0 = " << x0 << std::endl;
        else std::cout << std::endl << "z > n : aucune solution r\202elle"
                       << std::endl;
    }
    return 0;
}
```

Ce programme écrit en C++ permet d'obtenir l'exemple suivant :