Fun Facts First:

the HP32E uses two pettily different versions of the constant $\frac{1}{\sqrt{2\pi}}$,

`0,398942280444` and
`0,398942280385`.

Q^{-1} Function of HP32E

The HP32E is one of the few models that offers besides the Q function, the cumulative distribution function (CDF) of the standard normal distribution, also its reversal, Q^{-1}, the inverse cumulative distribution function, aka *probit*, or the quantile function associated with the standard normal distribution. (For its graph click here.)

In other words, for input x...

$p = \Phi(z)$ on 32E: Q(x)

$z_p = \Phi^{-1}(p)$ on 32E: $Q^{-1}(x)$ or: $x \stackrel{!}{=} Q(z)$

The Details

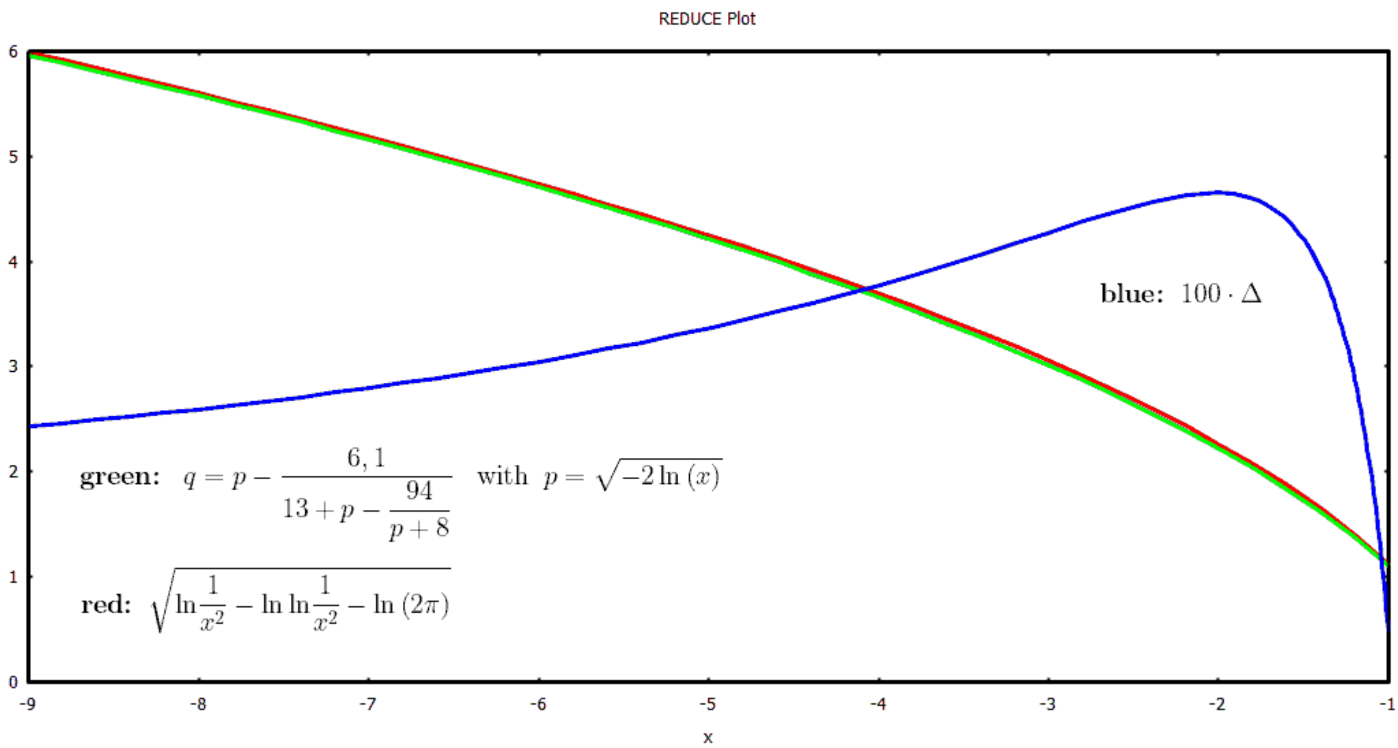
Analysing trace logs made with Tony's HP Classic Calculator Emulator Plus I found, the HP32E uses its approximations of Q to solve numerically $x - \mathrm{Q(z)} = 0$ for z, thus Q-1(x). The Q^{-1} function is split in three parts (special cases input<0, =0, =1, and >1 are sorted out first), one for the input range]0..0,1[(part 1) and next from [0,1..0,9[(part 2). The third range [0,9..1[is "folded down" to part 1 using the function's symmetry. (The limits 0,1 and 0,9 correspond to those $\pm 1{,}28$ of Q, dividing the scope correspondingly.)

Q^{-1} Part 1 and 3

Prepare first loop...
if $0 < x < 0,1$ => part 1,
if $0,9 <= x < 1$ => part 3, remember for later
if part 3 set $x := 1 - x$
$p = \sqrt{(-\ln(x^2))}$

$$q = p - \frac{6,1}{13 + p - \frac{94}{p+8}} \approx \sqrt{\ln\frac{1}{x^2} - \ln\ln\frac{1}{x^2} - \ln(2\pi) + o(1)}$$

— it's *amazing* close (or just sufficing?) to the asymptotic expansion (for small input).



But then, alas:

$z_0 = q^2/2$ the initial guess, **an error**, should be q
$h_0 = e^{z_0}$ when righting above error use $e^{q^2/2}$
begin...
  r_n as in "Q part 2" but computed with z_{n-1} instead of $|x|$
  $s_n = 1/(z_{n-1} + b_2 + r_n)$ with $b_2 =$ -3,8052E-8
  $t_n = x \cdot h_{n-1}/a_1$ with $a_1 =$ 0,398942280444
  $k_n = z_{n-1}(t_n - s_n)$ or: $\sqrt{2\pi} \cdot h \cdot z \cdot (x - Q(z))$
  $h_n = h_{n-1}/(1 + k_n)$
  $z_n = \sqrt{2\ln(h_n)}$ or: $\sqrt{z^2 - 2\ln(1+k)}$
loop while $|k_n| >$ 1E-10
if part 3 set $z := -z$

Part 2

For input $0,1 <= x < 0,9$ it goes like this:

Prepare first loop...
$s = x - 0,5$
$z_0 = s/b_1$ the initial guess, $b_1 =$ 0,398942280385
begin...
  $t_n = z_{n-1}^2/2$
  $u_n = Q(z_{n-1}) - 0,5$
  $w_n = (u_n - s) \cdot e^{t_n}/b_1$
  $z_n = z_{n-1} - w_n$
loop while $|w_n| >$ 1E-10

Lessons Learned?

Trace logs enable to detect the inbuilt algorithms rather mechanically, but to recognize the maths principle behind a procedure needs a bit more than just know-how, some know-why would help.

Part 2 is Newton's method, completely as in every maths standard textbook. e^{t_n}/b_1 is the reciprocal value of the Q function's derivative. Got it, ticked it.

For part 1 – sorry, no idea (yet). I am able to spot the $(x - Q(z))$ step, I am able to copy the procedure to another system (and it works), I did find an error in first valuation of z (what is ironed out with one or two more iterations), but in contrast to part 2 I do miss the eureka moment.