Then the answer is

\[
\sqrt{c + id} = \begin{cases} 
0 & w = 0 \\
 w + i \left( \frac{d}{2w} \right) & w \neq 0, c \geq 0 \\
 \frac{|d|}{2w} + iw & w \neq 0, c < 0, d \geq 0 \\
 \frac{|d|}{2w} - iw & w \neq 0, c < 0, d < 0 
\end{cases} \quad (5.4.7)
\]

Routines implementing these algorithms are listed in Appendix C.

CITED REFERENCES AND FURTHER READING:
(Reading, MA: Addison-Wesley) [see solutions to exercises 4.2.1.16 and 4.6.4.41].

5.5 Recurrence Relations and Clenshaw's Recurrence Formula

Many useful functions satisfy recurrence relations, e.g.,

\[
(n + 1)P_{n+1}(x) = (2n + 1)xP_n(x) - nP_{n-1}(x) \quad (5.5.1)
\]

\[
J_{n+1}(x) = \frac{2n}{x}J_n(x) - J_{n-1}(x) \quad (5.5.2)
\]

\[
nE_{n+1}(x) = e^{-x} - xE_n(x) \quad (5.5.3)
\]

\[
\cos n\theta = 2\cos \theta \cos(n-1)\theta - \cos(n-2)\theta \quad (5.5.4)
\]

\[
\sin n\theta = 2\cos \theta \sin(n-1)\theta - \sin(n-2)\theta \quad (5.5.5)
\]

where the first three functions are Legendre polynomials, Bessel functions of the first kind, and exponential integrals, respectively. (For notation see [1].) These relations are useful for extending computational methods from two successive values of \( n \) to other values, either larger or smaller.

Equations (5.5.4) and (5.5.5) motivate us to say a few words about trigonometric functions. If your program’s running time is dominated by evaluating trigonometric functions, you are probably doing something wrong. Trig functions whose arguments form a linear sequence \( \theta = \theta_0 + n\delta, n = 0, 1, 2, \ldots \), are efficiently calculated by the following recurrence,

\[
\cos(\theta + \delta) = \cos \theta - [\alpha \cos \theta + \beta \sin \theta] \\
\sin(\theta + \delta) = \sin \theta - [\alpha \sin \theta - \beta \cos \theta]
\quad (5.5.6)
\]
where $\alpha$ and $\beta$ are the precomputed coefficients

$$\alpha \equiv 2 \sin^2 \left( \frac{\delta}{2} \right) \quad \beta \equiv \sin \delta \quad (5.5.7)$$

The reason for doing things this way, rather than with the standard (and equivalent) identities for sums of angles, is that here $\alpha$ and $\beta$ do not lose significance if the incremental $\delta$ is small. Likewise, the adds in equation (5.5.6) should be done in the order indicated by square brackets. We will use (5.5.6) repeatedly in Chapter 12, when we deal with Fourier transforms.

Another trick, occasionally useful, is to note that both $\sin \theta$ and $\cos \theta$ can be calculated via a single call to $\tan$:

$$t \equiv \tan \left( \frac{\theta}{2} \right) \quad \cos \theta = \frac{1 - t^2}{1 + t^2} \quad \sin \theta = \frac{2t}{1 + t^2} \quad (5.5.8)$$

The cost of getting both $\sin$ and $\cos$, if you need them, is thus the cost of $\tan$ plus 2 multiplies, 2 divides, and 2 adds. On machines with slow trig functions, this can be a savings. However, note that special treatment is required if $\theta \to \pm \pi$. And also note that many modern machines have very fast trig functions; so you should not assume that equation (5.5.8) is faster without testing.

### Stability of Recurrences

You need to be aware that recurrence relations are not necessarily stable against roundoff error in the direction that you propose to go (either increasing $n$ or decreasing $n$). A three-term linear recurrence relation

$$y_{n+1} + a_n y_n + b_n y_{n-1} = 0, \quad n = 1, 2, \ldots \quad (5.5.9)$$

has two linearly independent solutions, $f_n$ and $g_n$, say. Only one of these corresponds to the sequence of functions $f_n$ that you are trying to generate. The other one $g_n$ may be exponentially growing in the direction that you want to go, or exponentially damped, or exponentially neutral (growing or dying as some power law, for example). If it is exponentially growing, then the recurrence relation is of little or no practical use in that direction. This is the case, e.g., for (5.5.2) in the direction of increasing $n$, when $x < n$. You cannot generate Bessel functions of high $n$ by forward recurrence on (5.5.2).

To state things a bit more formally, if

$$f_n / g_n \to 0 \quad \text{as} \quad n \to \infty \quad (5.5.10)$$

then $f_n$ is called the minimal solution of the recurrence relation (5.5.9). Nonminimal solutions like $g_n$ are called dominant solutions. The minimal solution is unique, if it exists, but dominant solutions are not — you can add an arbitrary multiple of $f_n$ to a given $g_n$. You can evaluate any dominant solution by forward recurrence, but not the minimal solution. (Unfortunately it is sometimes the one you want.)

Abramowitz and Stegun (in their Introduction) give a list of recurrences that are stable in the increasing or decreasing directions. That list does not contain all
possible formulas, of course. Given a recurrence relation for some function \( f_n(x) \) you can test it yourself with about five minutes of (human) labor: For a fixed \( x \) in your range of interest, start the recurrence not with true values of \( f_j(x) \) and \( f_{j+1}(x) \), but (first) with the values 1 and 0, respectively, and then (second) with 0 and 1, respectively. Generate 10 or 20 terms of the recursive sequences in the direction that you want to go (increasing or decreasing from \( j \)), for each of the two starting conditions. Look at the difference between the corresponding members of the two sequences. If the differences stay of order unity (absolute value less than 10, say), then the recurrence is stable. If they increase slowly, then the recurrence may be mildly unstable but quite tolerably so. If they increase catastrophically, then there is an exponentially growing solution of the recurrence. If you know that the function that you want actually corresponds to the growing solution, then you can keep the recurrence formula anyway e.g., the case of the Bessel function \( Y_n(x) \) for increasing \( n \), see §6.5; if you don’t know which solution your function corresponds to, you must at this point reject the recurrence formula. Notice that you can do this test before you go to the trouble of finding a numerical method for computing the two starting functions \( f_j(x) \) and \( f_{j+1}(x) \): stability is a property of the recurrence, not of the starting values.

An alternative heuristic procedure for testing stability is to replace the recurrence relation by a similar one that is linear with constant coefficients. For example, the relation (5.5.2) becomes

\[
y_{n+1} - 2\gamma y_n + y_{n-1} = 0
\]

(5.5.11)

where \( \gamma \equiv n/x \) is treated as a constant. You solve such recurrence relations by trying solutions of the form \( y_n = a^n \). Substituting into the above recurrence gives

\[
a^2 - 2\gamma a + 1 = 0 \quad \text{or} \quad a = \gamma \pm \sqrt{\gamma^2 - 1}
\]

(5.5.12)

The recurrence is stable if \(|a| \leq 1\) for all solutions \( a \). This holds (as you can verify) if \(|\gamma| \leq 1\) or \( n \leq x \). The recurrence (5.5.2) thus cannot be used, starting with \( J_0(x) \) and \( J_1(x) \), to compute \( J_n(x) \) for large \( n \).

Possibly you would at this point like the security of some real theorems on this subject (although we ourselves always follow one of the heuristic procedures). Here are two theorems, due to Perron [2]:

**Theorem A.** If in (5.5.9) \( a_n \sim an^\alpha, b_n \sim bn^\beta \) as \( n \to \infty \), and \( \beta < 2\alpha \), then

\[
g_{n+1}/g_n \sim -an^\alpha, \quad f_{n+1}/f_n \sim -(b/a)n^{\beta-\alpha}
\]

(5.5.13)

and \( f_n \) is the minimal solution to (5.5.9).

**Theorem B.** Under the same conditions as Theorem A, but with \( \beta = 2\alpha \), consider the characteristic polynomial

\[
t^2 + at + b = 0
\]

(5.5.14)

If the roots \( t_1 \) and \( t_2 \) of (5.5.14) have distinct moduli, \( |t_1| > |t_2| \) say, then

\[
g_{n+1}/g_n \sim t_1 n^\alpha, \quad f_{n+1}/f_n \sim t_2 n^\alpha
\]

(5.5.15)
and \( f_n \) is again the minimal solution to (5.5.9). Cases other than those in these two theorems are inconclusive for the existence of minimal solutions. (For more on the stability of recurrences, see [3].)

How do you proceed if the solution that you desire is the minimal solution? The answer lies in that old aphorism, that every cloud has a silver lining: If a recurrence relation is catastrophically unstable in one direction, then that (undesired) solution will decrease very rapidly in the reverse direction. This means that you can start with any seed values for the consecutive \( f_j \) and \( f_{j+1} \) and (when you have gone enough steps in the stable direction) you will converge to the sequence of functions that you want, times an unknown normalization factor. If there is some other way to normalize the sequence (e.g., by a formula for the sum of the \( f_n \)'s), then this can be a practical means of function evaluation. The method is called Miller's algorithm. An example often given [1,4] uses equation (5.5.2) in just this way, along with the normalization formula

\[
1 = J_0(x) + 2J_2(x) + 2J_4(x) + 2J_6(x) + \cdots \tag{5.5.16}
\]

Incidentally, there is an important relation between three-term recurrence relations and continued fractions. Rewrite the recurrence relation (5.5.9) as

\[
\frac{y_n}{y_{n-1}} = -\frac{b_n}{a_n + y_{n+1}/y_n} \tag{5.5.17}
\]

Iterating this equation, starting with \( n \), gives

\[
\frac{y_n}{y_{n-1}} = -\frac{b_n}{a_n - a_{n+1} + \cdots} \tag{5.5.18}
\]

Pincherle's Theorem [2] tells us that (5.5.18) converges if and only if (5.5.9) has a minimal solution \( f_n \), in which case it converges to \( f_n/f_{n-1} \). This result, usually for the case \( n = 1 \) and combined with some way to determine \( f_0 \), underlies many of the practical methods for computing special functions that we give in the next chapter.

**Clenshaw's Recurrence Formula**

Clenshaw's recurrence formula [5] is an elegant and efficient way to evaluate a sum of coefficients times functions that obey a recurrence formula, e.g.,

\[
f(\theta) = \sum_{k=0}^{N} c_k \cos k\theta \quad \text{or} \quad f(x) = \sum_{k=0}^{N} c_k P_k(x)
\]

Here is how it works: Suppose that the desired sum is

\[
f(x) = \sum_{k=0}^{N} c_k F_k(x) \tag{5.5.19}
\]

and that \( F_k \) obeys the recurrence relation

\[
F_{n+1}(x) = \alpha(n, x)F_n(x) + \beta(n, x)F_{n-1}(x) \tag{5.5.20}
\]
for some functions $\alpha(n,x)$ and $\beta(n,x)$. Now define the quantities $y_k$ $(k = N, N-1, \ldots, 1)$ by the following recurrence:

\begin{equation}
\begin{aligned}
y_{N+2} &= y_{N+1} = 0 \\
y_k &= \alpha(k,x)y_{k+1} + \beta(k+1,x)y_{k+2} + c_k \quad (k = N, N-1, \ldots, 1)
\end{aligned}
\end{equation}

If you solve equation (5.5.21) for $c_k$ on the left, and then write out explicitly the sum (5.5.19), it will look (in part) like this:

\[
f(x) = \cdots + [y_8 - \alpha(8,x)y_9 - \beta(9,x)y_{10}]F_k(x) \\
+ [y_7 - \alpha(7,x)y_8 - \beta(8,x)y_9]F_7(x) \\
+ [y_6 - \alpha(6,x)y_7 - \beta(7,x)y_8]F_6(x) \\
+ [y_5 - \alpha(5,x)y_6 - \beta(6,x)y_7]F_5(x) \\
+ \cdots + [y_2 - \alpha(2,x)y_3 - \beta(3,x)y_4]F_2(x) \\
+ [y_1 - \alpha(1,x)y_2 - \beta(2,x)y_3]F_1(x) \\
+ [c_0 + \beta(1,x)y_2 - \beta(1,x)y_2]F_0(x)
\]

Notice that we have added and subtracted $\beta(1,x)y_2$ in the last line. If you examine the terms containing a factor of $y_8$ in (5.5.22), you will find that they sum to zero as a consequence of the recurrence relation (5.5.20); similarly all the other $y_k$'s down through $y_2$. The only surviving terms in (5.5.22) are

\begin{equation}
f(x) = \beta(1,x)F_0(x)y_2 + F_1(x)y_1 + F_0(x)c_0
\end{equation}

Equations (5.5.21) and (5.5.23) are Clenshaw's recurrence formula for doing the sum (5.5.19): You make one pass down through the $y_k$'s using (5.5.21); when you have reached $y_2$ and $y_1$ you apply (5.5.23) to get the desired answer.

Clenshaw's recurrence as written above incorporates the coefficients $c_k$ in a downward order, with $k$ decreasing. At each stage, the effect of all previous $c_k$'s is "remembered" as two coefficients which multiply the functions $F_{k+1}$ and $F_k$ (ultimately $F_0$ and $F_1$). If the functions $F_k$ are small when $k$ is large, and if the coefficients $c_k$ are small when $k$ is small, then the sum can be dominated by small $F_k$'s. In this case the remembered coefficients will involve a delicate cancellation and there can be a catastrophic loss of significance. An example would be to sum the trivial series

\begin{equation}
J_{15}(1) = 0 \times J_0(1) + 0 \times J_1(1) + \ldots + 0 \times J_{14}(1) + 1 \times J_{15}(1)
\end{equation}

Here $J_{15}$, which is tiny, ends up represented as a canceling linear combination of $J_0$ and $J_1$, which are of order unity.
5.6 Quadratic and Cubic Equations

The roots of simple algebraic equations can be viewed as being functions of the equations’ coefficients. We are taught these functions in elementary algebra. Yet, surprisingly many people don’t know the right way to solve a quadratic equation with two real roots, or to obtain the roots of a cubic equation.

There are two ways to write the solution of the quadratic equation

\[ ax^2 + bx + c = 0 \]  

(5.6.1)

with real coefficients \(a, b, c\), namely

\[ x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]  

(5.6.2)