computing the value of the expression, the Maple model regards each number as being padded with zeros indefinitely. This is important for rounding.

For Maple 7, floating-point division has been sped up relative to previous Maple releases. As well as such efficiency improvements, Maple's floating point now conforms to the IEEE standards. Maple also supports a signed zero, as advocated by Kahan. This paper describes some further improvements in floating-point performance.

2 Inverse functions and iterative schemes

We shall describe here the evaluation of the square root function and the natural logarithm function, but the same methods can be applied to many other functions, for example, division, nth roots, and the Lambert W function. All inverting schemes are variations on, or extensions to, Newton iteration. We therefore start by giving a uniform treatment of these schemes.

2.1 General iteration formulae

Consider solving the equation $f(x) = 0$, given an initial estimate x_0 for the solution. We expand $f(x)$ as a Taylor series around x_0.

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \tfrac{1}{2}(x - x_0)^2 f''(x_0) + \frac{1}{6}(x - x_0)^3 f'''(x_0) + \ldots \tag{1}$$

Setting $h = x - x_0$ and assuming $f(x) = 0$, we can solve for h by either the Lagrange inversion theorem, or, equivalently, series reversion. Abbreviating $f(x_0), f'(x_0), f''(x_0)$ to f, f', f'' for clarity, we write

$$h = -\frac{f}{f'}$$

Square root. Let x_n be an approximation to $x = A^{1/2}$. Write

$$\begin{aligned} x &= \left(x_n^2 + A - x_n^2\right)^{1/2} = x_n\left(1 + A/x_n^2 - 1\right)^{1/2} \\ &= x_n \sum_{k=0}^{\infty} \binom{1/2}{k}\left(\frac{A}{x_n^2} - 1\right)^k . \end{aligned} \tag{5}$$

This is an example of an *expansion by residuals* according to Popov and Hare [4].

Taking any finite number of terms, we obtain an iteration formula for x_{n+1} from the exact formula for x. The first two terms reproduce the Newton iteration formula, also called Heron's rule by Kahan [5]. Taking the first three terms gives Chebyshev's method [6] or Halley's method, after conversion to a fraction as described above. Higher-order methods are obviously available by taking more terms in the series.

Markstein [1, 2] was apparently the first to propose that a faster way to compute $x = \sqrt{A}$ is to compute first the reciprocal $y = 1/\sqrt{A}$, followed by $x = Ay$. Letting y_n be an approximation to y, we write

$$y = y_n(1 + (Ay_n^2 - 1))^{-1/2} = y_n \sum_{k=0} \binom{-1/2}{k} (Ay_n^2 - 1)^k \tag{6}$$

$$= y_n\left(1 + \frac{1}{2}(1 - Ay_n^2) + \frac{3}{8}(1 - Ay_n^2)^2 + \frac{5}{16}(1 - Ay_n^2)^3 \ldots\right) . \tag{7}$$

Since the potential advantage of this series is the fact that there is no division, there is no point in forcing this into Halley's form. Notice that in either base 2 or in base 10, the coefficients can be expressed exactly as multiplying factors. It is obvious that (6) is a convergent series for $|Ay_n^2 - 1| < 1$.

Logarithm function.

Theorem. Given y_0 correct to $d > 1$ digits (rounded), a Newton improvement is correct to $2d - 1$ digits and a Chebyshev step is correct to $3d - 2$ digits.
Proof. Let $y_1 = y_0 + \frac{1}{2}y_0(1 - Ay_0^2)$ be the result of the Newton step. The forward error is then

$$\frac{|y - y_1|}{y} = \frac{|y - y_0 - \frac{1}{2}y_0(1 - Ay_0^2)|}{y} = \frac{|2y^3 - 3y_0y^2 + y_0^3|}{2y^3} = \frac{(y - y_0)^2(2y + y_0)}{2y^3} .$$

So

$$\frac{|y - y_1|}{y} \approx \frac{3(y - y_0)^2}{2y^2} \leq \tfrac{3}{2}\varphi^2\, 10^{2-2d} = \left(\tfrac{3}{2}\varphi^2\right) 10^{1-(2d-1)} .$$

Therefore in the worst case of $\varphi = 1/2$, y_1 is correct to $2d - 1$ rounded digits. In the case of a result y_0 correct to d truncated digits, then y_1 is accurate to $2d - 2$ digits.

After a Chebyshev step, we have $y_1 = y_0 + \frac{1}{2}y_0(1 - Ay_0^2) + \frac{3}{8}y_0(1 - Ay_0^2)^2$. Then

$$\frac{|y - y_1|}{}$$

3.4 Fine-grain precision control

From the point of view of optimizing the computation, the basic e□ects are that a higher-order method requires fewer steps, but requires more computation per step. However, this section will show that, with a fine-grained control of precision, the relevant considerations are not the obvious ones. In fact we shall see that the total number of iterations used is not a critical factor in the overall time, but only the characteristics of the final loop.

In Maple, a reasonably accurate model of the cost of arithmetic is that two numbers with m and n digits can be multiplied in $O(mn)$ time. We combine this model with some observations concerning the required precision at each step. For the sake of definiteness, we conduct the discussion in terms of square root.

Consider a Newton step in the inverse square root iteration. We write it as a pair of equations:

$$\begin{aligned} r &= 1 - Ay_n^2\,, \\ y_{n+1} &= y_n + {}^1 \end{aligned} \tag{11}$$

4 Implementation notes

4.1 Expected input data

The number of digits in the input data can be quite di erent from the number of digits in the output value. Thus there is a need in computer algebra systems for the square root of a small, exactly known number, such as an integer. Therefore the performance of an algorithm depends upon two parameters: the size (length) of the input and the size of the the requested output. The case in which both lengths are equal is the obvious case, but the case in which the input is short is significant.

4.2 Order of evaluation

The residual $1 - Ax^2$